cloudera®

Cluster Management Guide

Cloudera, Inc.
395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com

Release Information

Version: Cloudera Enterprise 6.0.x
Date: October 18, 2021

Table of Contents

About Cloudera Administration

The *Cluster Management Guide* describes how to configure and manage clusters in a Cloudera Enterprise deployment using Cloudera Manager. Cloudera Enterprise Hadoop Administrators manage resources, hosts, high availability, and backup and recovery configurations. The Cloudera Manager Admin Console is the primary tool administrators use to monitor and manage clusters. You can also use the Cloudera Manager API for cluster management tasks.

Cloudera Manager

Cloudera Manager is an end-to-end application for managing CDH clusters. Cloudera Manager provides granular visibility into and control over every part of the CDH cluster—empowering operators to improve performance, enhance quality of service, increase compliance, and reduce administrative costs. With Cloudera Manager, you can easily deploy and centrally operate the complete CDH stack and other managed services. The application automates the installation process, reducing deployment time from weeks to minutes; gives you a cluster-wide, real-time view of hosts and services running; provides a single, central console to enact configuration changes across your cluster; and incorporates a full range of reporting and diagnostic tools to help you optimize performance and utilization. Cloudera Manager also provides an API you can use to automate cluster operations.
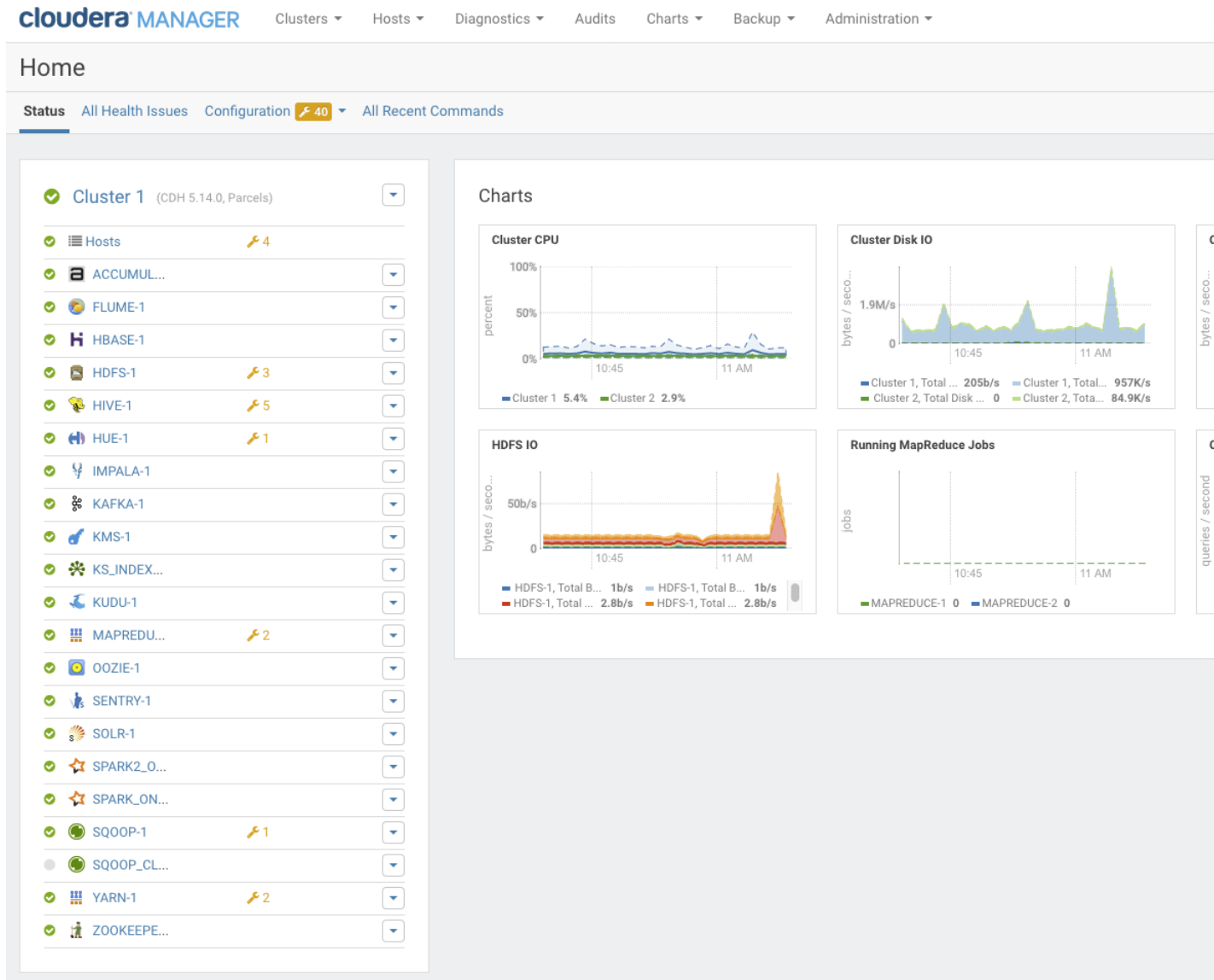
Figure 1: Cloudera Manager Admin Console

The following topics from the core Cloudera Enterprise documentation library can help you understand Cloudera Manager concepts and how to use Cloudera Manager to manage, monitor, and upgrade your deployment. They are listed by broad category:

Learn about Cloudera Manager

- [Cloudera Manager Overview](#)
- [Cloudera Manager Frequently Asked Questions](#) on page 22
- [Cloudera Manager Admin Console](#) on page 16
- [Cloudera Manager API](#) on page 24

Installing and Upgrading

You can use Cloudera Manager to upgrade Cloudera Manager and the CDH software running on your cluster.

- [Installation](#)
- [Upgrading Cloudera Manager](#)

Managing CDH using Cloudera Manager

Cloudera Manager provides tools you can use to manage the services, hosts, and applications deployed in your cluster.

- [Cloudera Manager Administration](#) on page 29
- [Cluster Configuration Overview](#) on page 59
- [Managing Clusters](#) on page 83
- [Managing Hosts](#) on page 169
- [Managing Services](#) on page 193
- [Managing Roles](#) on page 209
- [Maintenance Mode](#) on page 186
- [Resource Management](#) on page 367
- [High Availability](#) on page 421
- [Backup and Disaster Recovery](#) on page 483
- [Cloudera Manager Configuration Properties Reference](#) on page 82 (Reference)
- [Managing Services](#) on page 88

Monitoring CDH using Cloudera Manager

You can use Cloudera Manager to monitor many aspects of the performance of the services and hosts in your cluster. You can define alerts that send notifications in response to conditions you specify.

- [Introduction to Cloudera Manager Monitoring](#) on page 214
- [Monitoring Clusters](#) on page 230
- [Monitoring Services](#) on page 231
- [Monitoring Hosts](#) on page 243
- [Monitoring Activities](#) on page 248
- [Logs](#) on page 327

Managing CDH using the Cloudera Manager API

The Cloudera Manager API provides configuration and service lifecycle management, service health information and metrics, and allows you to configure Cloudera Manager itself. The API is served on the same host and port as the [Cloudera Manager Admin Console](#) on page 16, and does not require an extra process or extra configuration. The API supports HTTP Basic Authentication, accepting the same users and credentials as the Cloudera Manager Admin Console.

- [Quick Start](#)

- [Cloudera Manager API tutorial](#)
- [Cloudera Manager REST API documentation](#)
- [Python Client](#) (deprecated)
- [Python Client](#) (Swagger-based)
- [Java Client](#) (Swagger-based)
- [Java SDK Reference](#)
- [Using the Cloudera Manager API for Cluster Automation](#) on page 27

Cloudera Manager Admin Console

Cloudera Manager Admin Console is the web-based UI that you use to configure, manage, and monitor CDH.

If no services are configured when you log into the Cloudera Manager Admin Console, the Cloudera Manager installation wizard displays. If services have been configured, the Cloudera Manager top navigation bar:
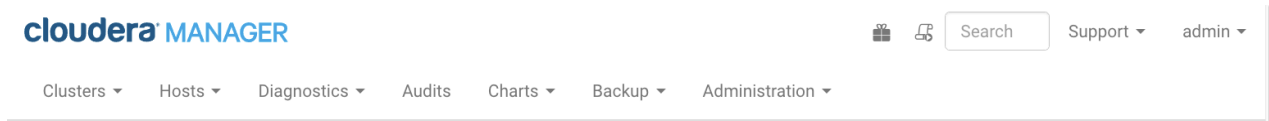
and [Home](#) page display. The Cloudera Manager Admin Console top navigation bar provides the following tabs and menus:

- **Clusters** > *cluster_name*

    - **Services** - Display individual services, and the Cloudera Management Service. In these pages you can:

        - View the status and other details of a service instance or the role instances associated with the service
        - Make configuration changes to a service instance, a role, or a specific role instance
        - Add and delete a service or role
        - Stop, start, or restart a service or role.
        - View the commands that have been run for a service or a role
        - View an audit event history
        - Deploy and download client configurations
        - Decommission and recommission role instances
        - Enter or exit maintenance mode
        - Perform actions unique to a specific type of service. For example:

            - Enable HDFS high availability or NameNode federation
            - Run the HDFS Balancer
            - Create HBase, Hive, and Sqoop directories

    - **Cloudera Manager Management Service** - Manage and monitor the Cloudera Manager Management Service. This includes the following roles: Activity Monitor, Alert Publisher, Event Server, Host Monitor, Navigator Audit Server, Navigator Metadata Server, Reports Manager, and Service Monitor.
    - **Cloudera Navigator** - Opens the Cloudera Navigator user interface.
    - **Hosts** - Displays the hosts in the cluster.
    - **Reports** - Create reports about the HDFS, MapReduce, YARN, and Impala usage and browse HDFS files, and manage quotas for HDFS directories.
    - **Utilization Report** - Opens the **Cluster Utilization Report.** displays aggregated utilization information for YARN and Impala jobs.
    - *MapReduce_service_name* **Jobs** - Query information about MapReduce jobs running on your cluster.
    - *YARN_service_name* **Applications** - Query information about YARN applications running on your cluster.
    - *Impala_service_name* **Queries** - Query information about Impala queries running on your cluster.
    - **Dynamic Resource Pools** - Manage dynamic allocation of cluster resources to YARN and Impala services by specifying the relative weights of named pools.

- **Static Service Pools** - Manage static allocation of cluster resources to HBase, HDFS, Impala, MapReduce, and YARN services.

- **Hosts** - Display the hosts managed by Cloudera Manager.

  - **All Hosts** - Displays a list of manage hosts in the cluster.
  - **Roles** - Displays the roles deployed on each host.
  - **Host Templates** - Create and manage **Host Templates**, which define sets of role groups that can be used to easily expand a cluster.
  - **Disks Overview** - Displays the status of all disks in the cluster.
  - **Parcels** - Displays parcels available in the cluster and allows you to download, distribute, and activate new parcels.

  In this page you can:

  - View the status and a variety of detail metrics about individual hosts
  - Make configuration changes for host monitoring
  - View all the processes running on a host
  - Run the Host Inspector
  - Add and delete hosts
  - Create and manage host templates
  - Manage parcels
  - Decommission and recommission hosts
  - Make rack assignments
  - Run the host upgrade wizard

- **Diagnostics** - Review logs, events, and alerts to diagnose problems. The subpages are:

  - **Events** - Search for and displaying events and alerts that have occurred.
  - **Logs** - Search logs by service, role, host, and search phrase as well as log level (severity).
  - **Server Log** -Display the Cloudera Manager Server log.

- **Audits** - Query and filter audit events across clusters, including logins, across clusters.
- **Charts** - Query for metrics of interest, display them as charts, and display personalized chart dashboards.
- **Backup** - Manage replication schedules and snapshot policies.
- **Administration** - Administer Cloudera Manager. The subpages are:

  - **Settings** - Configure Cloudera Manager.
  - **Alerts** - Display when alerts will be generated, configure alert recipients, and send test alert email.
  - **Users** - Manage Cloudera Manager users and user sessions.
  - **Security** - Generate Kerberos credentials and inspect hosts.
  - **License** - Manage Cloudera licenses.
  - **Language** - Set the language used for the content of activity events, health events, and alert email messages.
  - **AWS Credentials** - Configure S3 connectivity to Cloudera Manager.

-

  Parcel Icon - link to the **Hosts** > **Parcels** page.
- **Running Commands Indicator** - displays the number of commands currently running for all services or roles.
- **Search** - Supports searching for services, roles, hosts, configuration properties, and commands. You can enter a partial string and a drop-down list with up to sixteen entities that match will display.
- **Support** - Displays various support actions. The subcommands are:

  - **Send Diagnostic Data** - Sends data to Cloudera Support to support troubleshooting.
  - **Support Portal (Cloudera Enterprise)** - Displays the Cloudera Support portal.
  - **Mailing List (Cloudera Express)** - Displays the Cloudera Manager Users list.

- **Scheduled Diagnostics: Weekly** - Configure the frequency of automatically collecting diagnostic data and sending to Cloudera support.
  - The following links open the latest documentation on the Cloudera web site:
    - **Help**
    - **Installation Guide**
    - **API Documentation**
    - **Release Notes**

  - **About** - Version number and build details of Cloudera Manager and the current date and time stamp of the Cloudera Manager server.

- **Logged-in User Menu** - The currently logged-in user. The subcommands are:
  - **Change Password** - Change the password of the currently logged in user.
  - **Logout**

Starting and Logging into the Admin Console

1. In a web browser, go to `http://<server_host>:7180`, where *<server_host>* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

2. Log into Cloudera Manager Admin Console using the credentials assigned by your administrator. User accounts are assigned roles that constrain the features available to you.

   Note: You can configure the Cloudera Manager Admin Console to automatically log out a user after a configurable period of time. See Automatic Logout on page 21.

Cloudera Manager Admin Console Home Page

When you start the Cloudera Manager Admin Console on page 16, the **Home** > **Status** tab displays.

You can also go to the **Home** > **Status** tab by clicking the Cloudera Manager logo in the top navigation bar.

Status

The Status tab contains:

- **Clusters** - The clusters being managed by Cloudera Manager. Each cluster is displayed either in summary form or in full form depending on the configuration of the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property. When the number of clusters exceeds the value of the property, only cluster summary information displays.

  - **Summary Form** - A list of links to cluster status pages. Click **Customize** to jump to the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property.
  - **Full Form** - A separate section for each cluster containing a link to the cluster status page and a table containing links to the Hosts page and the status pages of the services running in the cluster.

  Each service row in the table has a menu of actions that you select by clicking

  ▼

  and can contain one or more of the following indicators:

Indicator	Meaning	Description
❶ 2	Health issue	Indicates that the service has at least one health issue. The indicator shows the number of health issues at the highest severity level. If there are Bad health test results, the indicator is red. If there are no Bad health test results, but Concerning test results exist, then the indicator is yellow. No indicator is shown if there are no Bad or Concerning health test results. **❗ Important:** If there is one Bad health test result and two Concerning health results, there will be three health issues, but the number will be one. Click the indicator to display the **Health Issues** pop-up dialog box. By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show _n_ concerning issue(s)** link.Click the link to display the Status page containing with details about the health test result.
✂ 4	Configuration issue	Indicates that the service has at least one configuration issue. The indicator shows the number of configuration issues at the highest severity level. If there are configuration errors, the indicator is red. If there are no errors but configuration warnings exist, then the indicator is yellow. No indicator is shown if there are no configuration notifications. **❗ Important:** If there is one configuration error and two configuration warnings, there will be three configuration issues, but the number will be one. Click the indicator to display the **Configuration Issues** pop-up dialog box. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show _n_ warning(s)** link.Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.See Managing Services on page 193.

Indicator	Meaning	Description
⏻ Restart Needed 🔃 Refresh Needed	Configuration modified	Indicates that at least one of a service's roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. Click the indicator to display the Stale Configurations on page 77 page.To bring the cluster up-to-date, click the Refresh or Restart button on the Stale Configurations page or follow the instructions in Refreshing a Cluster on page 85, Restarting a Cluster on page 85, or Restarting Services and Instances after Configuration Changes on page 64.
🗎	Client configuration redeployment required	Indicates that the client configuration for a service should be redeployed. Click the indicator to display the Stale Configurations on page 77 page.To bring the cluster up-to-date, click the Deploy Client Configuration button on the Stale Configurations page or follow the instructions in Manually Redeploying Client Configuration Files on page 80.

- **Cloudera Management Service** - A table containing a link to the Cloudera Manager Service. The Cloudera Manager Service has a menu of actions that you select by clicking

  ▼   .

- **Charts** - A set of charts (dashboard) that summarize resource utilization (IO, CPU usage) and processing metrics.

  Click a line, stack area, scatter, or bar chart to expand it into a full-page view with a legend for the individual charted entities as well more fine-grained axes divisions.

  By default the time scale of a dashboard is 30 minutes. To change the time scale, click a duration link

  30m 1h 2h 6h 12h 1d 7d 30d   at the top-right of the dashboard.

  To set the dashboard type, click 🖉▼ and select one of the following:

  - **Custom** - displays a custom dashboard.
  - **Default** - displays a default dashboard.
  - **Reset** - resets the custom dashboard to the predefined set of charts, discarding any customizations.

All Health Issues

Displays all health issues by cluster. The number badge has the same semantics as the per service health issues reported on the Status tab.

- By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show *n* concerning issue(s)** link.
- To group the health test results by entity or health test, click the buttons on the **Organize by Entity/Organize by Health Test** switch.
- Click the link to display the Status page containing with details about the health test result.

All Configuration Issues

Displays all configuration issues by cluster. The number badge has the same semantics as the per service configuration issues reported on the Status tab. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show *n* warning(s)** link. Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.

All Recent Commands

Displays all commands run recently across the clusters. A badge   indicates how many recent commands are still running. Click the command link to display details about the command and child commands. See also [Viewing Running and Recent Commands](#) on page 241.

Starting and Logging into the Cloudera Manager Admin Console

1. In a web browser, go to `http://<server_host>:7180`, where *<server_host>* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

2. Log into Cloudera Manager Admin Console using the [credentials](#) assigned by your administrator. User accounts are assigned [roles](#) that constrain the features available to you.

   > **Note:** You can configure the Cloudera Manager Admin Console to automatically log out a user after a configurable period of time. See [Automatic Logout](#) on page 21.

Displaying the Cloudera Manager Server Version and Server Time

To display the version, build number, and time for the Cloudera Manager Server:

1. Open the Cloudera Manager Admin Console.
2. Select **Support** > **About**.

Displaying Cloudera Manager Documentation

To display Cloudera Manager documentation:

1. Open the Cloudera Manager Admin Console.
2. Select **Support** > **Help, Installation Guide, API Documentation, or Release Notes**. By default, the Help and Installation Guide files from the Cloudera web site are opened. This is because local help files are not updated after installation. You can configure Cloudera Manager to open either the latest Help and Installation Guide from the Cloudera web site (this option requires Internet access from the browser) or locally-installed Help and Installation Guide by configuring the **Administration** > **Settings** > **Support** > **Open latest Help files from the Cloudera website** property.

Automatic Logout

For security purposes, Cloudera Manager automatically logs out a user session after 30 minutes. You can change this session logout period.

To configure the timeout period:

1. Click **Administration** > **Settings**.
2. Click **Category** > **Security**.
3. Edit the **Session Timeout** property.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

When the timeout is one minute from triggering, the user sees the following message:

Automatic Logout for Your Protection   ×

Due to inactivity, your current work session is about to expire. For your security,
Cloudera Manager sessions automatically end after 30 minutes of inactivity.

Your current session will expire in **1 minute**.
Press any key or click anywhere to continue.

If the user does not click the mouse or press a key, the user is logged out of the session and the following message
appears:

Automatic Log Out Due to Inactivity

You are now logged out of your account.

We hadn't heard from you for about 30 minute(s), so
for your security Cloudera Manager automatically
logged you out of your account. Log back in below
to continue.

admin

•••••

Log In

☐ Remember me

Cloudera Manager Frequently Asked Questions

This guide answers frequently asked questions about Cloudera Manager.

General Questions

What are the new features of Cloudera Manager?

For a list of new features in Cloudera Manager, see Cloudera Enterprise 6 Release Guide.

What operating systems are supported?

See CDH and Cloudera Manager Supported Operating Systems for more detailed information on which operating
systems are supported.

What databases are supported?

See Database Requirements for more detailed information on which database systems are supported.

What version of CDH is supported for Cloudera Manager?

See Cloudera Manager and CDH Compatibility for detailed information.

What are the differences between the Cloudera Express and the Cloudera Enterprise versions of Cloudera Manager?

Cloudera Express includes a free version of Cloudera Manager. The Cloudera Enterprise version of Cloudera Manager provides additional functionality. Both the Cloudera Express and Cloudera Enterprise versions automate the installation, configuration, and monitoring of CDH on an entire cluster. See the data sheet at Cloudera Enterprise Datasheet for a comparison of the two versions.

The Cloudera Enterprise version of Cloudera Manager is available as part of the Cloudera Enterprise subscription offering, and requires a license. You can also choose a Cloudera Enterprise Trial that is valid for 60 days.

If you are not an existing Cloudera customer, contact Cloudera Sales using this form or call 866-843-7207 to obtain a Cloudera Enterprise license. If you are already a Cloudera customer and you need to upgrade from Cloudera Express to Cloudera Enterprise, contact Cloudera Support to obtain a license.

Are there different types of Cloudera Enterprise licenses?

Cloudera Enterprise is available on a subscription basis in five editions, each designed around how you use the platform:

- **Essentials Edition** provides superior support and advanced management for core Apache Hadoop.
- **Data Science and Engineering Edition** for programmatic data preparation and predictive modeling.
- **Operational Database Edition** for online applications with real-time serving needs.
- **Data Warehouse Edition** for BI and SQL analytics.
- **Enterprise Data Hub Edition** provides for complete use of the platform.

All editions are available in your environment of choice: cloud, on-premise, or a hybrid deployment. For more information, see the Cloudera Enterprise Data Sheet.

Can I upgrade CDH using Cloudera Manager?

You can upgrade CDH from within the Cloudera Manager Admin Console using parcels. Furthermore, once you have installed or upgraded CDH using parcels, you can perform rolling upgrades on your CDH services. If you have HDFS high availability configured and enabled, you can perform a rolling upgrade on your cluster without taking the entire cluster down.

What version of CDH does Cloudera Manager install?

Cloudera Manager 5.x allows you to install any version of CDH 5 with the same minor version or lower as Cloudera Manager. Cloudera Manager 6.0.0 and higher allows you to install CDH 5.7 or higher. For more information, see Cloudera Enterprise 6 Requirements and Supported Versions.

Where are CDH libraries located when I distribute CDH using parcels?

With parcel software distribution, the path to the CDH libraries is `/opt/cloudera/parcels/CDH/lib/` instead of the usual `/usr/lib/`.

What upgrade paths are available for Cloudera Manager, and what's involved?

For instructions about upgrading, see Upgrading Cloudera Manager.

How do I install Cloudera Manager in a walled-off environment (no Internet access)?

You can set up a local repository and use it in the installer. For instructions, see Custom Installation Solutions.

Do worker hosts need access to the Cloudera public repositories for an install with Cloudera Manager?

You can perform an installation or upgrade using the parcel format and when using parcels, only the Cloudera Manager Server requires access to the Cloudera public repositories. Distribution of the parcels to worker hosts is done between the Cloudera Manager Server and the worker hosts. See Parcels for more information. If you want to install using the traditional packages, hosts only require access to the installation files.

For both parcels and packages, it is also possible to create local repositories that serve these files to the hosts that are being upgraded. If you have established local repositories, no access to the Cloudera public repository is required. For more information, see Using an Internal Package Repository.

Can I use the service monitoring features of Cloudera Manager without the Cloudera Management Service?

No. To understand the desired state of the system, Cloudera Manager requires the global configuration that the Cloudera Management Service roles gather and provide. The Cloudera Manager Agent doubles as both the agent for supervision and for monitoring.

Can I run the Cloudera Management Service and the Hadoop services on the host where the Cloudera Manager Server is running?

Yes. This is especially common in deployments that have a small number of hosts.

Does Cloudera Manager Support an API?

Yes. A comprehensive set of APIs for the various features is supported in this version of Cloudera Manager. For more information about the Cloudera Manager API, see Cloudera Manager API on page 24. You can download this Cloudera Manager API example that shows how to integrate with Nagios or other systems.

Cloudera Manager API

The Cloudera Manager API provides configuration and service lifecycle management, service health information and metrics, and allows you to configure Cloudera Manager itself. The API is served on the same host and port as the Cloudera Manager Admin Console on page 16, and does not require an extra process or extra configuration. The API supports HTTP Basic Authentication, accepting the same users and credentials as the Cloudera Manager Admin Console.

You can also access the Cloudera Manager Swagger API user interface from the **Cloudera Manager Admin Console**. Go to **Support** > **API Explorer** to open Swagger.

API Documentation Resources

- Quick Start
- Cloudera Manager API tutorial
- Cloudera Manager REST API documentation
- Python Client (deprecated)
- Python Client (Swagger-based)
- Java Client (Swagger-based)
- Java SDK Reference
- Using the Cloudera Manager API for Cluster Automation on page 27

Obtaining Configuration Files

1. Obtain the list of a service's roles:

```
http://cm_server_host:7180/api/v30/clusters/clusterName/services/serviceName/roles
```

2. Obtain the list of configuration files a process is using:

```
http://cm_server_host:7180/api/v30/clusters/clusterName/services/serviceName/roles/roleName/process
```

3. Obtain the content of any particular file:

```
http://cm_server_host:7180/api/v30/clusters/clusterName/services/serviceName/roles/roleName/process/
configFiles/configFileName
```

For example:

```
http://cm_server_host:7180/api/v30/clusters/Cluster%201/services/OOZIE-1/roles/
OOZIE-1-OOZIE_SERVER-e121641328fcb107999f2b5fd856880d/process/configFiles/oozie-site.xml
```

Retrieving Service and Host Properties

To update a service property using the Cloudera Manager APIs, you'll need to know the name of the property, not just the display name. If you know the property's display name but not the property name itself, retrieve the documentation by requesting any configuration object with the query string `view=FULL` appended to the URL. For example:

```
http://cm_server_host:7180/api/v30/clusters/Cluster%201/services/service_name/config?view=FULL
```

Search the results for the display name of the desired property. For example, a search for the display name **HDFS Service Environment Advanced Configuration Snippet (Safety Valve)** shows that the corresponding property name is `hdfs_service_env_safety_valve`:

```
{
    "name" : "hdfs_service_env_safety_valve",
    "require" : false,
    "displayName" : "HDFS Service Environment Advanced Configuration Snippet (Safety
Valve)",
    "description" : "For advanced use onlyu, key/value pairs (one on each line) to be
inserted into a roles
    environment. Applies to configurations of all roles in this service except client
configuration.",
    "relatedName" : "",
    "validationState" : "OK"
}
```

Similar to finding service properties, you can also find host properties. First, get the host IDs for a cluster with the URL:

```
http://cm_server_host:7180/api/v30/hosts
```

This should return host objects of the form:

```
{
    "hostId" : "2c2e951c-aaf2-4780-a69f-0382181f1821",
    "ipAddress" : "10.30.195.116",
    "hostname" : "cm_server_host",
    "rackId" : "/default",
    "hostUrl" :
"http://cm_server_host:7180/cmf/hostRedirect/2c2e951c-adf2-4780-a69f-0382181f1821",
    "maintenanceMode" : false,
    "maintenanceOwners" : [ ],
    "commissionState" : "COMMISSIONED",
    "numCores" : 4,
    "totalPhysMemBytes" : 10371174400
}
```

Then obtain the host properties by including one of the returned host IDs in the URL:

```
http://cm_server_host:7180/api/v30/hosts/2c2e951c-adf2-4780-a69f-0382181f1821?view=FULL
```

Backing Up and Restoring the Cloudera Manager Configuration

You can use the Cloudera Manager REST API to export and import all of its configuration data. The API exports a JSON document that contains configuration data for the Cloudera Manager instance. You can use this JSON document to back up and restore a Cloudera Manager deployment.

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

Exporting the Cloudera Manager Configuration

1. Log in to the Cloudera Manager server host as the `root` user.
2. Run the following command:

```
# curl -u admin_uname:admin_pass "http://cm_server_host:7180/api/v30/cm/deployment" >
path_to_file/cm-deployment.json
```

Where:

- `admin_uname` is a username with either the Full Administrator or Cluster Administrator role.
- `admin_pass` is the password for the *admin_uname* username.
- `cm_server_host` is the hostname of the Cloudera Manager server.
- `path_to_file` is the path to the file where you want to save the configuration.

Redacting Sensitive Information from the Exported Configuration

The exported configuration may contain passwords and other sensitive information. You can configure redaction of the sensitive items by specifying a JVM parameter for Cloudera Manager. When you set this parameter, API calls to Cloudera Manager for configuration data do not include the sensitive information.

> **!** **Important:** If you configure this redaction, you cannot use an exported configuration to restore the configuration of your cluster due to the redacted information.

To configure redaction for the API:

1. Log in the Cloudera Manager server host.
2. Edit the `/etc/default/cloudera-scm-server` file by adding the following property (separate each property with a space) to the line that begins with `export CMF_JAVA_OPTS`:

```
-Dcom.cloudera.api.redaction=true
```

For example:

```
export CMF_JAVA_OPTS="-Xmx2G -Dcom.cloudera.api.redaction=true"
```

3. Restart Cloudera Manager:

```
sudo service cloudera-scm-server restart
```

Restoring the Cloudera Manager Configuration

> **!** **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See Managing Licenses on page 37 for more information.

Using a previously saved JSON document that contains the Cloudera Manager configuration data, you can restore that configuration to a running cluster.

1. Using the Cloudera Manager Administration Console, stop all running services in your cluster:

   a. On the **Home** > **Status** tab, click

   ▼

   to the right of the cluster name and select **Stop**.

   b. Click **Stop** in the confirmation screen. The **Command Details** window shows the progress of stopping services.

When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.

> ⚠️ **Warning:** If you do not stop the cluster before making this API call, the API call will stop *all* cluster services before running the job. Any running jobs and data are lost.

2. Log in to the Cloudera Manager server host as the `root` user.
3. Run the following command:

```
curl -H "Content-Type: application/json" --upload-file path_to_file/cm-deployment.json
 -u admin:admin
http://cm_server_host:7180/api/v30/cm/deployment?deleteCurrentDeployment=true
```

Where:

- `admin_uname` is a username with either the Full Administrator or Cluster Administrator role.
- `admin_pass` is the password for the *admin_uname* username.
- `cm_server_host` is the hostname of the Cloudera Manager server.
- `path_to_file` is the path to the file containing the JSON configuration file.

4. Restart the **Cloudera Manager Server**.

   RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo systemctl restart cloudera-scm-server
```

   RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

```
sudo service cloudera-scm-server restart
```

Using the Cloudera Manager API for Cluster Automation

> 📝 **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

One of the complexities of Apache Hadoop is the need to deploy clusters of servers, potentially on a regular basis. If you maintain hundreds of test and development clusters in different configurations, this process can be complex and cumbersome if not automated.

Cluster Automation Use Cases

Cluster automation is useful in various situations. For example, you might work on many versions of CDH, which works on a wide variety of OS distributions (RHEL 6, Ubuntu Precise and Lucid, Debian Wheezy, and SLES 11). You might have complex configuration combinations—highly available HDFS or simple HDFS, Kerberized or non-secure, YARN or MRv1, and so on. With these requirements, you need an easy way to create a new cluster that has the required setup. This cluster can also be used for integration, testing, customer support, demonstrations, and other purposes.

You can install and configure Hadoop according to precise specifications using the Cloudera Manager REST API. Using the API, you can add hosts, install CDH, and define the cluster and its services. You can also tune heap sizes, set up HDFS HA, turn on Kerberos security and generate keytabs, and customize service directories and ports. Every configuration available in Cloudera Manager is exposed in the API.

The API also provides access to management functions:

- Obtaining logs and monitoring the system
- Starting and stopping services

- Polling cluster events
- Creating a disaster recovery replication schedule

For example, you can use the API to retrieve logs from HDFS, HBase, or any other service, without knowing the log locations. You can also stop any service with no additional steps.

Use scenarios for the Cloudera Manager API for cluster automation might include:

- OEM and hardware partners that deliver Hadoop-in-a-box appliances using the API to set up CDH and Cloudera Manager on bare metal in the factory.
- Automated deployment of new clusters, using a combination of Puppet and the Cloudera Manager API. Puppet does the OS-level provisioning and installs the software. The Cloudera Manager API sets up the Hadoop services and configures the cluster.
- Integrating the API with reporting and alerting infrastructure. An external script can poll the API for health and metrics information, as well as the stream of events and alerts, to feed into a custom dashboard.

Java API Example

This example covers the Java API client.

To use the Java client, add this dependency to your project's `pom.xml`:

```
<project>
  <repositories>
    <repository>
      <id>cdh.repo</id>
      <url>https://repository.cloudera.com/artifactory/cloudera-repos</url>
      <name>Cloudera Repository</name>
    </repository>
    …
  </repositories>
  <dependencies>
    <dependency>
      <groupId>com.cloudera.api</groupId>
      <artifactId>cloudera-manager-api</artifactId>
      <version>4.6.2</version>        <!-- Set to the version of Cloudera Manager you use
-->
    </dependency>
    …
  </dependencies>
  ...
</project>
```

The Java client works like a proxy. It hides from the caller any details about REST, HTTP, and JSON. The entry point is a handle to the root of the API:

```
RootResourcev30 apiRoot = new ClouderaManagerClientBuilder().withHost("cm.cloudera.com")
.withUsernamePassword("admin", "admin").build().getRootv30();
```

From the root, you can traverse down to all other resources. (It's called "v30" because that is the current Cloudera Manager API version, but the same builder will also return a root from an earlier version of the API.) The tree view shows some key resources and supported operations:

- `RootResourcev30`

  - `ClustersResourcev30` - host membership, start cluster

    - `ServicesResourcev30` - configuration, get metrics, HA, service commands

      - `RolesResource` - add roles, get metrics, logs
      - `RoleConfigGroupsResource` - configuration

    - `ParcelsResource` - parcel management

- `HostsResource` - host management, get metrics
- `UsersResource` - user management

For more information, see the Javadoc.

The following example lists and starts a cluster:

```
// List of clusters
ApiClusterList clusters = apiRoot.getClustersResource().readClusters(DataView.SUMMARY);
for (ApiCluster cluster : clusters) {
  LOG.info("{}: {}", cluster.getName(), cluster.getVersion());
}

// Start the first cluster
ApiCommand cmd = apiRoot.getClustersResource().startCommand(clusters.get(0).getName());
while (cmd.isActive()) {
    Thread.sleep(100);
    cmd = apiRoot.getCommandsResource().readCommand(cmd.getId());
}
LOG.info("Cluster start {}", cmd.getSuccess() ? "succeeded" : "failed " +
cmd.getResultMessage());
```

Python Example

You can see an example of automation with Python at the following link: Python example. The example contains information on the requirements and steps to automate a cluster deployment.

Cloudera Manager Administration

Starting, Stopping, and Restarting the Cloudera Manager Server

To start the Cloudera Manager Server:

```
sudo service cloudera-scm-server start
```

You can stop (for example, to perform maintenance on its host) or restart the Cloudera Manager Server without affecting the other services running on your cluster. Statistics data used by activity monitoring and service monitoring will continue to be collected during the time the server is down.

To stop the Cloudera Manager Server:

```
sudo service cloudera-scm-server stop
```

To restart the Cloudera Manager Server:

```
sudo service cloudera-scm-server restart
```

Configuring Cloudera Manager Server Ports

Minimum Required Role: Full Administrator

1. Select **Administration** > **Settings**.
2. Under the **Ports and Addresses** category, set the following options as described below:

Setting	Description
HTTP Port for Admin Console	Specify the HTTP port to use to access the Server using the Admin Console.
HTTPS Port for Admin Console	Specify the HTTPS port to use to access the Server using the Admin Console.
Agent Port to connect to Server	Specify the port for Agents to use to connect to the Server.

3. Click **Save Changes**.
4. [Restart the Cloudera Manager Server](#).

Moving the Cloudera Manager Server to a New Host

You can move the Cloudera Manager Server if either the Cloudera Manager database server or a current [backup](#) of the Cloudera Manager database is available.

To move the Cloudera Manager Server:

1. Identify a new host on which to install Cloudera Manager.
2. Install Cloudera Manager on a new host, using the method described under [Install the Cloudera Manager Server Packages](#).

> ⓘ **Important:**
> - The Cloudera Manager version on the destination host *must match* the version on the source host.
> - Do not install the other components, such as CDH and databases.

3. Copy the entire content of `/var/lib/cloudera-scm-server/` on the old host to that same path on the new host. Ensure you preserve permissions and all file content.
4. If the database server is not available:

   a. Install the database packages on the host that will host the restored database. This could be the same host on which you have just installed Cloudera Manager or it could be a different host. If you used the embedded PostgreSQL database, install the PostgreSQL package as described in [Managing the Embedded PostgreSQL Database](#). If you used an external MySQL, PostgreSQL, or Oracle database, reinstall following the instructions in [Step 4: Install and Configure Databases](#).

   b. Restore the backed up databases to the new database installation.

5. Update `/etc/cloudera-scm-server/db.properties` with the database name, database instance name, username, and password.
6. Do the following on all cluster hosts:

   a. In `/etc/cloudera-scm-agent/config.ini`, update the `server_host` property to the new hostname.

   b. If you are replacing the Cloudera Manager database with a new database, and you are not using a backup of the original Cloudera Manager database, delete the `/var/lib/cloudera-scm-agent/cm_guid` file.

   c. Restart the agent using the following command:

```
$ sudo service cloudera-scm-agent restart
```

7. Stop the Cloudera Manager server on the source host by running the following command:

```
service cloudera-scm-server stop
```

8. Copy any Custom Service Descriptor files for add-on services to the configured directory on the new Cloudera Manager host. The directory path is configured by going to **Administration** > **Settings** and editing the **Local Descriptor Repository Path** property. The default value is `/opt/cloudera/csd`. See [Add-on Services](#) on page 195.

9. Start the Cloudera Manager Server on the new (destination) host. Cloudera Manager should resume functioning as it did before the failure. Because you restored the database from the backup, the server should accept the running state of the Agents, meaning it will not terminate any running processes.

The process is similar with secure clusters, though files in `/etc/cloudera-scm-server` must be restored in addition to the database. See [Cloudera Security](#).

Managing the Cloudera Manager Server Log

Viewing the Log

To help you troubleshoot problems, you can view the Cloudera Manager Server log. You can view the logs in the Logs page or in specific pages for the log.

Viewing Cloudera Manager Server Logs in the Logs Page

1. Select **Diagnostics** > **Logs** on the top navigation bar.
2. Click **Select Sources** to display the log source list.
3. Uncheck the **All Sources** checkbox.
4. Click ▶ to the left of Cloudera Manager and select the **Server** checkbox.
5. Click **Search**.

For more information about the Logs page, see

Viewing the Cloudera Manager Server Log

1. Select **Diagnostics** > **Server Log** on the top navigation bar.

> **Note:** You can also view the Cloudera Manager Server log at
> `/var/log/cloudera-scm-server/cloudera-scm-server.log` on the Server host.

Setting the Cloudera Manager Server Log Location

By default the Cloudera Manager Server log is stored in `/var/log/cloudera-scm-server/`. If there is not enough space in that directory, you can change the location of the parent of the log directory:

1. Stop the Cloudera Manager Server:

```
sudo service cloudera-scm-server stop
```

2. Set the `CMF_VAR` environment variable in `/etc/default/cloudera-scm-server` to the new parent directory:

```
export CMF_VAR=/opt
```

3. Create `log/cloudera-scm_server` and `run` directories in the new parent directory and set the owner and group of all directories to `cloudera-scm`. For example, if the new parent directory is `/opt/`, do the following:

```
$ sudo su
$ cd /opt
$ mkdir log
$ chown cloudera-scm:cloudera-scm log
$ mkdir /opt/log/cloudera-scm-server
$ chown cloudera-scm:cloudera-scm log/cloudera-scm-server
$ mkdir run
$ chown cloudera-scm:cloudera-scm run
```

4. Restart the Cloudera Manager Server:

```
sudo service cloudera-scm-server start
```

Cloudera Manager Agents

The Cloudera Manager Agent is a Cloudera Manager component that works with the Cloudera Manager Server to manage the processes that map to role instances.

In a Cloudera Manager managed cluster, you can only start or stop role instance processes using Cloudera Manager. Cloudera Manager uses an open source process management tool called <u>supervisord</u>, that starts processes, takes care of redirecting log files, notifying of process failure, setting the effective user ID of the calling process to the right user, and so on. Cloudera Manager supports automatically restarting a crashed process. It will also flag a role instance with a bad health flag if its process crashes repeatedly right after start up.

The Agent is started by `init.d` at start-up. It, in turn, contacts the Cloudera Manager Server and determines which processes should be running. The Agent is monitored as part of Cloudera Manager's host monitoring. If the Agent stops heartbeating, the host is marked as having bad health.

One of the Agent's main responsibilities is to start and stop processes. When the Agent detects a new process from the Server heartbeat, the Agent creates a directory for it in `/var/run/cloudera-scm-agent` and unpacks the configuration. It then contacts `supervisord`, which starts the process.

cm_processes

To enable Cloudera Manager to run scripts in subdirectories of `/var/run/cloudera-scm-agent`, (because `/var/run` is mounted `noexec` in many Linux distributions), Cloudera Manager mounts a <u>tmpfs</u>, named `cm_processes`, for process subdirectories.

A `tmpfs` defaults to a max size of 50% of physical RAM but this space is not allocated until its used, and tmpfs is paged out to swap if there is memory pressure.

The lifecycle actions of `cmprocesses` can be described by the following statements:

- Created when the Agent starts up for the first time with a new supervisord process.
- If it already exists without `noexec`, reused when the Agent is started using `start` and not recreated.
- Remounted if Agent is started using `clean_restart`.
- Unmounting and remounting cleans out the contents (since it is mounted as a tmpfs).
- Unmounted when the host is rebooted.
- Not unmounted when the Agent is stopped.

Starting, Stopping, and Restarting Cloudera Manager Agents

Starting Agents

To start Agents, the `supervisord` process, and *all managed service processes*, use the following command:

- **Start**

```
sudo service cloudera-scm-agent start
```

Stopping and Restarting Agents

To stop or restart Agents *while leaving the managed processes running*, use one of the following commands:

- **Stop**

```
sudo service cloudera-scm-agent stop
```

- **Restart**

```
sudo service cloudera-scm-agent restart
```

Hard Stopping and Restarting Agents

> ⚠ **Warning:** The `hard_stop` and `hard_restart` commands kill all running managed service processes on the host(s) where the command is run.

> 📝 **Note:** The procedures in this section require you to stop all roles on the host. If it is not possible to stop all roles immediately, you must do so within 60 days of the hard stop or hard restart.

To stop or restart Agents, the `supervisord` process, and *all managed service processes*, use one of the following commands:

- **Hard Stop**

    1. Stop all roles running on the host. See [Stopping All the Roles on a Host](#) on page 183.

       If it is not possible to stop all roles immediately, you must do so within 60 days of the hard stop.

    2. Run the following command:

       - **RHEL 7, SLES 12, Debian 8, Ubuntu 16.04**

       ```
       sudo systemctl stop supervisord
       ```

       RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

       ```
       sudo service cloudera-scm-agent hard_stop
       ```

- **Hard Restart**

    1. Stop all roles running on the host. See [Stopping All the Roles on a Host](#) on page 183.

       If it is not possible to stop all roles immediately, you must do so within 60 days of the hard restart.

    2. Run the following command:

       RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

       ```
       sudo systemctl stop supervisord
       sudo systemctl start cloudera-scm-agent
       ```

       RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

       ```
       sudo service cloudera-scm-agent hard_restart
       ```

    Hard restart is useful for the following situations:

    1. You are upgrading Cloudera Manager and the `supervisord` code has changed between your current version and the new one. To properly do this upgrade you need to restart supervisor too.
    2. `supervisord` freezes and needs to be restarted.
    3. You want to clear out all running state pertaining to Cloudera Manager and managed services.

Checking Agent Status

To check the status of the Agent process, use the command:

```
sudo service cloudera-scm-agent status
```

Configuring Cloudera Manager Agents

Minimum Required Role: Full Administrator

Cloudera Manager Agents can be configured globally using properties you set in the Cloudera Manager Admin Console and by setting properties in Agent configuration files.

Configuring Agent Heartbeat and Health Status Options

You can configure the Cloudera Manager Agent heartbeat interval and timeouts to trigger changes in Agent health as follows:

1. Select **Administration** > **Settings**.
2. Under the **Performance** category, set the following option:

Property	Description
Send Agent Heartbeat Every	The interval in seconds between each heartbeat that is sent from Cloudera Manager Agents to the Cloudera Manager Server. Default: 15 sec.

3. Under the **Monitoring** category, set the following options:

Property	Description
Set health status to Concerning if the Agent heartbeats fail	The number of missed consecutive heartbeats after which a **Concerning** health status is assigned to that Agent. Default: 5.
Set health status to Bad if the Agent heartbeats fail	The number of missed consecutive heartbeats after which a **Bad** health status is assigned to that Agent. Default: 10.

4. Click **Save Changes**.

Configuring the Host Parcel Directory

> **Important:** If you modify the parcel directory location, make sure that all hosts use the same location. Using different locations on different hosts can cause unexpected problems.

To configure the location of distributed parcels:

1. Click **Hosts** in the top navigation bar.
2. Click the **Configuration** tab.
3. Select **Category** > **Parcels**.
4. Configure the value of the **Parcel Directory** property. The setting of the `parcel_dir` property in the Cloudera Manager Agent configuration file overrides this setting.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart the Cloudera Manager Agent on all hosts.

Agent Configuration File

The Cloudera Manager Agent supports different types of configuration options in the
`/etc/cloudera-scm-agent/config.ini` file. You must update the configuration on each host. After changing a
property, restart the Agent:

```
sudo service cloudera-scm-agent restart
```

Section	Property	Description
[General]	server_host, server_port, listening_port, listening_hostname, listening_ip	Hostname and ports of the Cloudera Manager Server and Agent and IP address of the Agent. Also see Configuring Cloudera Manager Server Ports on page 29 and Ports Used by Cloudera Manager and Cloudera Navigator.
		The Cloudera Manager Agent configures its hostname automatically. You can also manually specify the hostname the Cloudera Manager Agent uses by updating the listening_hostname property. To manually specify the IP address the Cloudera Manager Agent uses, update the listening_ip property in the same file.
		To have a CNAME used throughout instead of the regular hostname, an Agent can be configured to use listening_hostname=*CNAME*. In this case, the CNAME should resolve to the same IP address as the IP address of the hostname on that machine. Users doing this will find that the host inspector will report problems, but the CNAME will be used in all configurations where that's appropriate. This practice is particularly useful for users who would like clients to use namenode.*mycluster*.company.com instead of machine1234.*mycluster*.company.com. In this case, namenode.*mycluster* would be a CNAME for machine1234.*mycluster*, and the generated client configurations (and internal configurations as well) would use the CNAME.
	lib_dir	Directory to store Cloudera Manager Agent state that persists across instances of the agent process and system reboots. The Agent UUID is stored here. Default: /var/lib/cloudera-scm-agent.
	local_filesystem_whitelist	The list of local filesystems that should always be monitored. Default: ext2,ext3,ext4.
	log_file	The path to the Agent log file. If the Agent is being started using the init.d script, /var/log/cloudera-scm-agent/cloudera-scm-agent.out will also have a small amount of output (from before logging is initialized). Default: /var/log/cloudera-scm-agent/cloudera-scm-agent.log.
	max_collection_wait_seconds	Maximum time to wait for all metric collectors to finish collecting data. Default: 10 sec.

Section	Property	Description
	`metrics_url_timeout_seconds`	Maximum time to wait when connecting to a local role's web server to fetch metrics. Default: 30 sec.
	`parcel_dir`	Directory to store unpacked parcels. Default: `/opt/cloudera/parcels`. **⊟ Important:** If you modify the parcel directory location, make sure that all hosts use the same location. Using different locations on different hosts can cause unexpected problems. If you want to change this, Cloudera recommends following the procedure documented in Changing the Parcel Directory to change this for all hosts, rather than setting it in each host `config.ini` file. This property overrides the setting in Cloudera Manager. To use the recommended procedure, you must make sure that this property is commented out in each host `config.ini` file.
	`supervisord_port`	The supervisord port. A change takes effect the next time supervisord is restarted (not when the Agent is restarted). Default: 19001.
	`task_metrics_timeout_seconds`	Maximum time to wait when connecting to a local TaskTracker to fetch task attempt data. Default: 5 sec.

`[Security]`	`use_tls,verify_cert_file,`
	`client_key_file,`
	`client_keypw_file,`
	`client_cert_file`
		Security-related configuration. See
		• Configuring TLS Encryption for Cloudera Manager
		• Adding a Host to the Cluster on page 174

`[Cloudera]`	`mgmt_home`	Directory to store Cloudera Management Service files. Default: `/usr/share/cmf`.

`[JDBC]`	`cloudera_mysql_connector_jar,`
	`cloudera_oracle_connector_jar,`
	`cloudera_postgresql_jdbc_jar`
		Location of JDBC drivers. See Step 4: Install and Configure Databases.
		
		Default:
		• MySQL - `/usr/share/java/mysql-connector-java.jar`
		• Oracle - `/usr/share/java/oracle-connector-java.jar`
		• PostgreSQL - `/usr/share/cmf/lib/postgresql-`*version-build*`.jdbc4.jar`

Managing Cloudera Manager Agent Logs

Viewing Agent Logs

To help you troubleshoot problems, you can view the Cloudera Manager Agent logs. You can view the logs in the Logs page or in specific pages for the logs.

Viewing Cloudera Manager Agent Logs in the Logs Page

1. Select **Diagnostics** > **Logs** on the top navigation bar.
2. Click **Select Sources** to display the log source list.
3. Uncheck the **All Sources** checkbox.
4. Click ▶ to the left of Cloudera Manager and select the **Agent** checkbox.
5. Click **Search**.

For more information about the Logs page, see

Viewing the Cloudera Manager Agent Log

1. Click the **Hosts** tab.
2. Click the link for the host where you want to see the Agent log.
3. In the **Details** panel, click the **Details** link in the **Host Agent** field.
4. Click the **Agent Log** link.

You can also view the Cloudera Manager Agent log at `/var/log/cloudera-scm-agent/cloudera-scm-agent.log` on the Agent hosts.

Setting the Cloudera Manager Agent Log Location

By default the Cloudera Manager Agent log is stored in `/var/log/cloudera-scm-agent/`. If there is not enough space in that directory, you can change the location of the log file:

1. Set the `log_file` property in the Cloudera Manager Agent configuration file:

```
log_file=/opt/log/cloudera-scm-agent/cloudera-scm-agent.log
```

2. Create `log/cloudera-scm_agent` directories and set the owner and group to `cloudera-scm`. For example, if the log is stored in `/opt/log/cloudera-scm-agent`, do the following:

```
$ sudo su
$ cd /opt
$ mkdir log
$ chown cloudera-scm:cloudera-scm log
$ mkdir /opt/log/cloudera-scm-agent
$ chown cloudera-scm:cloudera-scm log/cloudera-scm-agent
```

3. Restart the Agent:

```
sudo service cloudera-scm-agent restart
```

Configuring Network Settings

Minimum Required Role: Full Administrator

To configure a proxy server thorough which data is downloaded to and uploaded from the Cloudera Manager Server, do the following:

1. Select **Administration** > **Settings**.
2. Click the **Network** category.
3. Configure proxy properties.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Managing Licenses

Minimum Required Role: Full Administrator

When you install Cloudera Manager, you can select among the following editions: Cloudera Express (no license required), a 60-day Cloudera Enterprise Cloudera Enterprise trial license, or Cloudera Enterprise (which requires a license). To obtain a Cloudera Enterprise license, fill in this form or call 866-843-7207.

Cloudera Manager

A Cloudera Enterprise license is required for the following features:

- LDAP and SAML authentication
- Configuration history
- Alerts delivered as SNMP traps and custom alert scripts
- Backup and disaster recovery
- Operational reports
- Cloudera Navigator
- Commands such as Rolling Restart, History and Rollback, and Send Diagnostic Data
- Cluster Utilization Reports on page 400
- Role Based Access Control and Cluster-specific roles

For details see What's the Difference Between Cloudera Express and Cloudera Enterprise?.

Accessing the License Page

To access the license page, select **Administration** > **License**.

If you have a license installed, the license page indicates its status (for example, whether your license is currently valid) and displays the license details: the license owner, the license key, and the expiration date of the license, if there is one.

At the right side of the page a table shows the usage of licensed components based on the number of hosts with those products installed. You can move the cursor over the

to see an explanation of each item.

Cloudera offers the following two types of licenses:

- **Cloudera Express**

  A free license that provides access to CDH, Cloudera's Apache Hadoop distribution, and a subset of cluster management features available with Cloudera Manager for up to 100 nodes. You can begin a trial or upgrade to Cloudera Enterprise with a Cloudera Express license.

- **Cloudera Enterprise**

  Cloudera Enterprise is available on a subscription basis in five editions, each designed around how you use the platform:

  - **Essentials Edition** provides superior support and advanced management for core Apache Hadoop.
  - **Data Science and Engineering Edition** for programmatic data preparation and predictive modeling.
  - **Operational Database Edition** for online applications with real-time serving needs.
  - **Data Warehouse Edition** for BI and SQL analytics.
  - **Enterprise Data Hub Edition** provides for complete use of the platform.

  All editions are available in your environment of choice: cloud, on-premise, or a hybrid deployment. For more information, see the Cloudera Enterprise Data Sheet.

License Expiration

Before a license expires, the Cloudera Manager Admin Console displays a message that indicates the number of days left on a license, starting at 60 days before expiration and counting down to 30, 14, and 0 days.

When a Cloudera Enterprise license expires, the following occurs:

- Cloudera Enterprise Cloudera Enterprise Trial - Enterprise features are disabled.
- Cloudera Enterprise - All services will continue to run as-is. The Cloudera Manager Admin Console will be disabled, meaning that you will not be able to view or modify clusters. Key Trustee KMS and the Key Trustee server will continue to function on the cluster, but you cannot change any configurations or add any services. On license expiration, the only action you can take in the Cloudera Manager Admin Console is to upload a new, valid license.

Trial Licenses

You can use a trial license only once; when the 60-day trial period expires or you have ended the trial, you cannot restart the trial. With the trial license, you can upgrade to a Cloudera Enterprise license or downgrade to an express license.

When a trial ends, enterprise features immediately become unavailable. However, data or configurations associated with the disabled functions are not deleted, and become available again once you install a Cloudera Enterprise license.

> **Note:** Trial licenses are not available for any of the Cloudera encryption products.

Ending a Cloudera Enterprise Cloudera Enterprise Trial

If you are using the trial edition the License page indicates when your license will expire. However, you can end the trial at any time (prior to expiration) as follows:

1. On the License page, click **End Trial**.
2. Confirm that you want to end the trial.
3. Restart the Cloudera Management Service, HBase, HDFS, and Hive services to pick up configuration changes.

Upgrading from Cloudera Express to a Cloudera Enterprise Cloudera Enterprise Trial

To start a trial, on the License page, click **Try Cloudera Enterprise Cloudera Enterprise for 60 Days**.

1. Cloudera Manager displays a pop-up describing the features enabled with Cloudera Enterprise Cloudera Enterprise. Click **OK** to proceed. At this point, your installation is upgraded and the Customize Role Assignments page displays.
2. Under **Reports Manager** click **Select a host**. The pageable host selection dialog box displays.

   The following shortcuts for specifying hostname patterns are supported:

   • Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   • IP addresses
   • Rack name

3. Select a host and click **OK**.
4. When you are finished with the assignments, click **Continue**.
5. Choose the database type:

   • Keep the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Record the auto-generated passwords.

- Select **Use Custom Databases** to specify the external database host and enter the database type, database name, username, and password for the custom database.
- If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged to improve performance, reduce database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See Configuring Oozie Data Purge Settings Using Cloudera Manager.

6. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

   The **Cluster Setup Review Changes** screen displays.

7. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

   > ⚠ **Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

8. At this point, your installation is upgraded. Click **Continue**.
9. Restart Cloudera Management Services and audited services to pick up configuration changes. The audited services will write audit events to a log file, but the events are not transferred to the Cloudera Navigator Audit Server until

you add and start the Cloudera Navigator Audit Server role as described in <u>Adding Cloudera Navigator Roles</u> on page 55. For information on Cloudera Navigator, see <u>Cloudera Navigator Data Management Overview</u>.

Upgrading from a Cloudera Enterprise Cloudera Enterprise Trial to Cloudera Enterprise

1. Purchase a Cloudera Enterprise license from Cloudera.
2. On the License page, click **Upload License**.
3. Click the **Select License File** field.
4. Browse to the location of your license file, click the file, and click **Open**.
5. Click **Upload**.

Upgrading from Cloudera Express to Cloudera Enterprise

1. Purchase a Cloudera Enterprise license from Cloudera.
2. On the License page, click **Upload License**.
3. Click the **Select License File** field.
4. Browse to the location of your license file, click the file, and click **Open**.
5. Click **Upload**.
6. Cloudera Manager displays a pop-up describing the features enabled with Cloudera Enterprise Cloudera Enterprise. Click **OK** to proceed. At this point, your installation is upgraded and the Customize Role Assignments page displays.
7. Under **Reports Manager** click **Select a host**. The pageable host selection dialog box displays.

   The following shortcuts for specifying hostname patterns are supported:

   • Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   • IP addresses
   • Rack name

8. When you are satisfied with the assignments, click **Continue**.
9. Choose the database type:

   • Keep the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Record the auto-generated passwords.

Cluster Setup

Database Setup

Configure and test database connections. If using custom databases, create the databases first according to the **Installing and Configuring an External Database** section of the Installation Guide ⬀ .

○ **Use Custom Databases**
◉ **Use Embedded Database**

When using the embedded database, passwords are automatically generated. Please copy them down.

Hive   ✓ Skipped. Cloudera Manager will create this database in a later step.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	hive	hive	t56iwbdk4F

Reports Manager   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	rman	rman	Y6S4IWvfNo

Navigator Audit Server   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	nav	nav	QLR2B0qqO9

Navigator Metadata Server   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	navms	navms	imo07jxOen

Oozie Server   ✓ Skipped. Cloudera Manager will create this database in a later step.

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	oozie_oozie_se	oozie_oozie_se	NTF1KNdpPI

Test Connection

- Select **Use Custom Databases** to specify the external database host and enter the database type, database name, username, and password for the custom database.
- If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged to improve performance, reduce database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See Configuring Oozie Data Purge Settings Using Cloudera Manager.

10. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

The **Cluster Setup Review Changes** screen displays.

11. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

> ⚠ **Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

12. At this point, your installation is upgraded. Click **Continue**.
13. Restart Cloudera Management Services and audited services to pick up configuration changes. The audited services will write audit events to a log file, but the events are not transferred to the Cloudera Navigator Audit Server until

you add and start the Cloudera Navigator Audit Server role as described in <u>Adding Cloudera Navigator Roles</u> on page 55. For information on Cloudera Navigator, see <u>Cloudera Navigator Data Management Overview</u>.

If you want to use the Cloudera Navigator Metadata Server, add its role following the instructions in <u>Adding Cloudera Navigator Roles</u> on page 55.

Renewing a License

1. Download the license file and save it locally.
2. In Cloudera Manager, go to the **Home** page.
3. Select **Administration** > **License**.
4. Click **Upload License**.
5. Browse to the license file you downloaded.
6. Click **Upload**.

Cloudera Manager requires a restart for the new license to take effect.

Sending Usage and Diagnostic Data to Cloudera

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

Cloudera Manager collects anonymous usage information and takes regularly-scheduled snapshots of the state of your cluster and automatically sends them anonymously to Cloudera. This helps Cloudera improve and optimize Cloudera Manager.

If you have a Cloudera Enterprise license, you can also trigger the collection of diagnostic data and send it to Cloudera Support to aid in resolving a problem you may be having.

Configuring a Proxy Server

To configure a proxy server through which usage and diagnostic data is uploaded, follow the instructions in <u>Configuring Network Settings</u> on page 37.

Managing Anonymous Usage Data Collection

Cloudera Manager sends anonymous usage information using Google Analytics to Cloudera. The information helps Cloudera improve Cloudera Manager. By default, anonymous usage data collection is *enabled*.

1. Select **Administration** > **Settings**.
2. Under the **Other** category, set the **Allow Usage Data Collection** property.
3. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Managing Hue Analytics Data Collection

Minimum Required Role: <u>Configurator</u> (also provided by **Cluster Administrator, Full Administrator**)

Hue tracks anonymized pages and application versions to collect information used to compare each application's usage levels. The data collected does not include hostnames or IDs; For example, the data has the format /2.3.0/pig, /2.5.0/beeswax/execute. You can restrict data collection as follows:

1. Go to the Hue service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hue**.
4. Locate the **Enable Usage Data Collection** property or search for it by typing its name in the Search box.
5. Clear the **Enable Usage Data Collection** checkbox.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See <u>Modifying Configuration Properties Using Cloudera Manager</u> on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the Hue service.

Diagnostic Data Collection

To help with solving problems when using Cloudera Manager on your cluster, Cloudera Manager collects diagnostic data on a regular schedule, and automatically sends it to Cloudera. By default Cloudera Manager is configured to collect this data weekly and to send it *automatically*. Cloudera analyzes this data and uses it to improve the software. If Cloudera discovers a serious issue, Cloudera searches this diagnostic data and notifies customers with Cloudera Enterprise licenses who might encounter problems due to the issue. You can schedule the frequency of data collection on a daily, weekly, or monthly schedule, or disable the scheduled collection of data entirely. You can also send a collected data set manually.

Automatically sending diagnostic data requires the Cloudera Manager Server host to have Internet access, and be configured for sending data automatically. If your Cloudera Manager Server does not have Internet access, and you have a Cloudera Enterprise license, you can manually send the diagnostic data as described in Manually Triggering Collection and Transfer of Diagnostic Data to Cloudera on page 46.

Automatically sending diagnostic data might fail sometimes and return an error message of "Could not send data to Cloudera." To work around this issue, you can manually send the data to Cloudera Support.

What Data Does Cloudera Manager Collect?

Cloudera Manager collects and returns a significant amount of information about the health and performance of the cluster. It includes:

- Up to 1000 Cloudera Manager audit events: Configuration changes, add/remove of users, roles, services, and so on.
- One day's worth of Cloudera Manager events: This includes critical errors Cloudera Manager watches for and more.
- Data about the cluster structure which includes a list of all hosts, roles, and services along with the configurations that are set through Cloudera Manager. Where passwords are set in Cloudera Manager, the passwords are not returned.
- Cloudera Manager license and version number.
- Current health information for hosts, service, and roles. Includes results of health tests run by Cloudera Manager.
- Heartbeat information from each host, service, and role. These include status and some information about memory, disk, and processor usage.
- The results of running Host Inspector.
- One day's worth of Cloudera Manager metrics. If you are using Cloudera Express, host metrics are not included.
- A download of the debug pages for Cloudera Manager roles.
- For each host in the cluster, the result of running a number of system-level commands on that host.
- Logs from each role on the cluster, as well as the Cloudera Manager server and agent logs.
- Which parcels are activated for which clusters.
- Whether there's an active trial, and if so, metadata about the trial.
- Metadata about the Cloudera Manager Server, such as its JMX metrics, stack traces, and the database or host it's running with.
- HDFS or Hive replication schedules (including command history) for the deployment.
- Impala query logs.
- Cloudera Data Science Workbench collects aggregate usage data by sending limited tracking events to Google Analytics and Cloudera servers. No customer data or personal information is sent as part of these bundles.

Configuring the Frequency of Diagnostic Data Collection

By default, Cloudera Manager collects diagnostic data on a weekly basis. You can change the frequency to daily, weekly, monthly, or never. If you are a Cloudera Enterprise customer and you set the schedule to **never**, you can still collect and send data to Cloudera on demand. If you are a Cloudera Express customer and you set the schedule to **never**, data is not collected or sent to Cloudera.

1. Select **Administration** > **Settings**.
2. Under the **Support** category, click **Scheduled Diagnostic Data Collection Frequency** and select the frequency.

3. To set the day and time of day that the collection will be performed, click **Scheduled Diagnostic Data Collection Time** and specify the date and time in the pop-up control.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

You can see the current setting of the data collection frequency by viewing **Support** > **Scheduled Diagnostics:** in the main navigation bar.

Specifying the Diagnostic Data Directory

You can configure the directory where collected data is stored.

1. Select **Administration** > **Settings**.
2. Under the **Support** category, set the **Diagnostic Data Bundle Directory** to a directory on the host running Cloudera Manager Server. The directory must exist and be enabled for writing by the user `cloudera-scm`. If this field is left blank, the data is stored in `/tmp`.
3. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Redaction of Sensitive Information from Diagnostic Bundles

By default, Cloudera Manager redacts known sensitive information from inclusion in diagnostic bundles. Cloudera Manager uses a set of standard rules to redact passwords and secrets. You can add additional redaction rules using regular expressions to specify data you want to be redacted from the bundles.

To specify redaction rules for diagnostic bundles:

1. Go to **Administration** > **Settings** and search for the **Redaction Parameters for Diagnostic Bundles** parameter.

   The edit screen for the property displays.

2. To add a new rule, click the ✚ icon. You can add one of the following:

   a. **Credit Card numbers (with separator)**
   b. **Social Security Card numbers (with separator)**
   c. **Email addresses**
   d. **Custom rule** (You must supply values for the **Search** and **Replace** fields, and optionally, the **Trigger** field.)

3. To modify a new rule, click the ➢ icon.
4. Edit the redaction rules as needed. Each rule has a description field where you can enter free text describing the rule and you can modify the following three fields:

   - **Search -** Regular expression to compare against the data. For example, the regular expression `\d{4}[^\w]\d{4}[^\w]\d{4}[^\w]\d{4}` searches for a credit card number pattern. Segments of data that match the regular expression are redacted using the Replace string.
   - **Replace -** String used to redact (obfuscate) data, such as a pattern of Xs to replace digits of a credit card number: `XXXX-XXXX-XXXX-XXXX`.
   - **Trigger -** Optional simple string to be searched before applying the regular expression. If the string is found, the redactor searches for matches using the Search regular expression. Using the Trigger field improves performance: simple string matching is faster than regular expression matching.

5. To delete a redaction rule, click the ▬ icon.
6. Click **Save Changes**.

Collecting and Sending Diagnostic Data to Cloudera

> 🛇 **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See [Managing Licenses](#) on page 37 for more information.

Disabling the Automatic Sending of Diagnostic Data from a Manually Triggered Collection

If you do not want data automatically sent to Cloudera after manually triggering data collection, you can disable this feature. The data you collect will be saved and can be downloaded for sending to Cloudera Support at a later time.

1. Select **Administration** > **Settings**.
2. Under the **Support** category, uncheck the box for **Send Diagnostic Data to Cloudera Automatically**.
3. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Manually Triggering Collection and Transfer of Diagnostic Data to Cloudera

To troubleshoot specific problems, or to re-send an automatic bundle that failed to send, you can manually send diagnostic data to Cloudera:

1. Optionally, change the System Identifier property:

   a. Select **Administration** > **Settings**.
   b. Under the **Other** category, set the System Identifier property and click **Save Changes**.

2. Under the **Support** menu at the top right of the navigation bar, choose **Send Diagnostic Data**. The Send Diagnostic Data form displays.
3. Fill in or change the information here as appropriate:

   - Optionally, you can improve performance by reducing the size of the data bundle that is sent. Click **Restrict log and metrics collection** to expand this section of the form. The three filters, **Host**, **Service**, and **Role Type**, allow you to restrict the data that will be sent. Cloudera Manager will only collect logs and metrics for roles that match all three filters.
   - Select one of the following under **Data Selection**:

     – Select **By Target Size** to manually set the maximum size of the bundle. Cloudera Manager populates the **End Time** based on the setting of the Time Range selector. You should change this to be a few minutes after you observed the problem or condition that you are trying to capture. The time range is based on the timezone of the host where Cloudera Manager Server is running.
     – Select **By Date Range** to manually set the **Start Time** and **End Time** to collect the diagnostic data. Click the **Estimate** button to calculate the size of the bundle based on the start and end times. If the bundle is too large, narrow the selection using the start and end times or by selecting additional filters.

   - If you have a support ticket open with Cloudera Support, include the support ticket number in the field provided.

4. Depending on whether you have disabled automatic sending of data, do one of the following:

   - Click **Collect and Upload Diagnostic Data to Cloudera Support**. A Running Commands window shows you the progress of the data collection steps. When these steps are complete, the collected data is sent to Cloudera.
   - Click **Collect Diagnostic Data only**. A Command Details window shows you the progress of the data collection steps.

     1. In the Command Details window, click **Download Result Data** to download and save a zip file of the information.
     2. Send the data to Cloudera Support by doing one of the following:

        - Send the bundle using a Python script:

          1. Download the phone_home script.
          2. Copy the script and the downloaded data file to a host that has Internet access.
          3. Run the following command on that host:

```
python phone_home.py --file downloaded data file
```

        - Attach the bundle to the SFDC case. Do not rename the bundle as this can cause a delay in processing the bundle.

- Contact [Cloudera Support](#) and arrange to send the data file.

Exporting and Importing Cloudera Manager Configuration

You can use the Cloudera Manager API to programmatically export and import a definition of all the entities in your Cloudera Manager-managed deployment—clusters, service, roles, hosts, users and so on. See the [Cloudera Manager API](#) on page 24 documentation on how to manage deployments using the [/cm/deployment](#) resource.

Backing Up Cloudera Manager

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

The following steps create a complete backup of Cloudera Manager:

Collect Information for Backing Up Cloudera Manager

1. Log in to the Cloudera Manager server host.

```
ssh my_cloudera_manager_server_host
```

2. Collect database information by running the following command:

```
cat /etc/cloudera-scm-server/db.properties
```

For example:

```
...
com.cloudera.cmf.db.type=...
com.cloudera.cmf.db.host=database_hostname:database_port
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=SOME_PASSWORD
```

3. Collect information (host name, port number, database name, user name and password) for the following databases.

- Reports Manager
- Navigator Audit Server
- Navigator Metadata Server
- Activity Monitor

You can find the database information by using the Cloudera Manager Admin Console. Go to **Clusters** > **Cloudera Management Service** > **Configuration** and select the **Database** category. You may need to contact your database administrator to obtain the passwords.

4. Find the host where the Service Monitor, Host Monitor and Event Server roles are running. Go to **Clusters** > **Cloudera Manager Management Service** > **Instances** and note which hosts are running these roles.

5. Identify the location of the Cloudera Navigator Metadata Server storage directory:

  a. Go to **Clusters** > **Cloudera Management Service** > **Instances**.
  b. Click the **Configuration** tab.
  c. Select **Scope** > **Navigator Metadata Server**.
  d. The **Navigator Metadata Server Storage Dir** property stores the location of the directory.

6. Ensure that Navigator Metadata Server Java heap is large enough to complete the upgrade. You can estimate the amount of heap needed from the number of elements and relations stored in the Solr storage directory.

  a. Go to **Clusters** > **Cloudera Management Service** > **Instances**.
  b. In the list of instances, click **Navigator Metadata Server**.
  c. Select **Log Files** > **Role Log File**.
  d. Search the log file for `solr core nav_elements` and note the number of element documents.

    e. Search the log file for `solr core nav_relations` and note the number of relation documents.

    f. Multiply the total number of documents by 200 bytes per document and add to it a baseline of 2 GB:

```
((num_nav_elements + num_nav_relations) * 200 bytes) + 2 GB
```

For example, if you had 68813088 elements and 78813930 relations, the recommended Java heap size is ~30 GB:

```
((68813088 + 78813930) * 200) + 2 GB = 29525403600 bytes = ~29.5 GB + 2 GB = ~ 31.5 GB
```

    g. Set the heap value in the Java Heap Size of Navigator Metadata Server in Bytes property in **Clusters** > **Cloudera Management Service** > **Configuration**.

Back Up Cloudera Manager Agent

> **Note:** Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable `CM_BACKUP_DIR`, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

Backup up the following Cloudera Manager agent files on **all hosts**:

- Create a top level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
mkdir -p $CM_BACKUP_DIR
```

- Back up the Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock
/etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent
/var/lib/cloudera-scm-agent
```

- Back up the existing repository directory.

    RHEL / CentOS

```
sudo cp -rf /etc/yum.repos.d $HOME/yum.repos.d-`date +%F`
```

    SLES

```
sudo cp -rf /etc/zypp/repos.d $HOME/repos.d-`date +%F`
```

    Debian / Ubuntu

```
sudo cp -rf /etc/apt/sources.list.d $HOME/sources.list.d-`date +%F`
```

Back Up the Cloudera Management Service

> **Note:** Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups. You may change these destination paths in the command as needed for your deployment.

1. On the host where the Service Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /var/lib/cloudera-service-monitor /var/lib/cloudera-scm-service-monitor-`date
 +%F`-CM
```

2. On the host where the Host Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /var/lib/cloudera-host-monitor /var/lib/cloudera-scm-host-monitor-`date
+%F`-CM
```

3. On the host where the Event Server role is configured to run, back up the following directory:

```
sudo cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserver-`date
+%F`-CM
```

Back Up Cloudera Navigator Data

1.

> **Important:** Upgrading from Cloudera Manager 5.9 (Navigator 2.8) and earlier can take a significant amount of time, depending on the size of the Navigator Metadata storage directory. When the Cloudera Manager upgrade process completes and Cloudera Navigator services restart, the Solr indexing upgrade automatically begins. No other actions can be performed until Solr indexing completes (a progress message displays during this process). It can take as long as two days to upgrade a storage directory with 60 GB. To help mitigate this extended upgrade step, make sure to clear out all unnecessary metadata using purge, check the size of the storage directory, and consider rerunning purge with tighter conditions to further reduce the size of the storage directory.

2. Make sure a purge task has run recently to clear stale and deleted entities.

- You can see when the last purge tasks were run in the Cloudera Navigator console (From the Cloudera Manager Admin console, go to **Clusters** > **Cloudera Navigator**. Select **Administration** > **Purge Settings**.)
- If a purge hasn't run recently, run it by editing the Purge schedule on the same page.
- Set the purge process options to clear out as much of the backlog of data as you can tolerate for your upgraded system. See [Managing Metadata Storage with Purge](#).

3. Stop the Navigator Metadata Server.

- **a.** Go to **Clusters** > **Cloudera Management Service** > **Instances**.
- **b.** Select **Navigator Metadata Server**.
- **c.** Click **Actions for Selected** > **Stop**.

4. Back up the Cloudera Navigator Solr storage directory.

```
sudo cp -rp /var/lib/cloudera-scm-navigator $HOME/cloudera-scm-navigator-`date +%F`-CM
```

5. If you are using an Oracle database for audit, in SQL*Plus, ensure that the following additional privileges are set:

```
  GRANT EXECUTE ON sys.dbms_crypto TO nav;
  GRANT CREATE VIEW TO nav;
```

where *nav* is the user of the Navigator Audit Server database.

Stop Cloudera Manager Server & Cloudera Management Service

1. Stop the **Cloudera Management Service**.

   a. Log in to the Cloudera Manager Admin Console.
   b. Select **Clusters** > **Cloudera Management Service**.
   c. Select **Actions** > **Stop**.

2. Log in to the Cloudera Manager server host.

```
ssh my_cloudera_manager_server_host
```

3. Stop the **Cloudera Manager Server**.

   RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo systemctl stop cloudera-scm-server
```

   RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

```
sudo service cloudera-scm-server stop
```

Back Up the Cloudera Manager Databases

1. Back up the **Cloudera Manager server database** – Run the following command. (The command displayed below depends on the database you selected in the form at the top of this page. Replace placeholders with the actual values returned from the `db.properties` file):

   MySQL

```
mysqldump --databases database_name --host=database_hostname --port=database_port -u
user_name -p > $HOME/database_name-backup-`date +%F`-CM.sql
```

> **Note:** If the `db.properties` file does not contain a port number, omit the port number parameter from the above command.

   PostgreSQL/Embedded

```
pg_dump -h database_hostname -U user_name -W -p database_port database_name >
$HOME/database_name-backup-`date +%F`-CM.sql
```

   Oracle

   Work with your database administrator to ensure databases are properly backed up.

   For more information about backing up databases, see <u>Backing Up Databases</u> on page 586.

2. Back up **All other databases** - For database information, go to **Clusters** > **Cloudera Management Service** > **Configuration**, select the **Database** category. You may need to contact your database administrator to obtain the passwords.

   These databases can include the following:

   • Reports Manager
   • Navigator Audit Server
   • Navigator Metadata Server
   • Activity Monitor (Only used for MapReduce 1 monitoring).

   Run the following commands to back up the databases. (The command displayed below depends on the database you selected in the form at the top of this page. Replace placeholders with the actual values.):

MySQL

```
mysqldump --databases database_name --host=database_hostname --port=database_port -u
database_username -p > $HOME/database_name-backup-`date +%F`-CM.sql
```

PostgreSQL/Embedded

```
pg_dump -h database_hostname -U database_username -W -p database_port database_name >
$HOME/database_name-backup-`date +%F`-CM.sql
```

Oracle

Work with your database administrator to ensure databases are properly backed up.

Back Up Cloudera Manager Server

> **Note:** Commands are provided below to backup various files and directories used by Cloudera Manager
> Agents. If you have configured custom paths for any of these, substitute those paths in the commands.
> The commands also provide destination paths to store the backups, defined by the environment
> variable CM_BACKUP_DIR, which is used in all the backup commands. You may change these destination
> paths in the command as needed for your deployment.

1. Log in to the Cloudera Manager server host.

```
ssh my_cloudera_manager_server_host
```

2. Create a top-level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
mkdir -p $CM_BACKUP_DIR
```

3. Back up the Cloudera Manager Server directories:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-server
/etc/default/cloudera-scm-server
```

4. Back up the existing repository directory.

RHEL / CentOS

```
sudo cp -rf /etc/yum.repos.d $HOME/yum.repos.d-`date +%F`
```

SLES

```
sudo cp -rf /etc/zypp/repos.d $HOME/repos.d-`date +%F`
```

Debian / Ubuntu

```
sudo cp -rf /etc/apt/sources.list.d $HOME/sources.list.d-`date +%F`
```

Start Cloudera Manager Server & Cloudera Management Service

1.

2. Start the **Cloudera Manager Server**.

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo systemctl start cloudera-scm-server
```

If the Cloudera Manager server starts without errors, no response displays.

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

```
sudo service cloudera-scm-server start
```

You should see the following:

```
Starting cloudera-scm-server: [ OK ]
```

3. Start the **Cloudera Management Service**.

   a. Log in to the Cloudera Manager Admin Console
   b. Select **Clusters** > **Cloudera Management Service**.
   c. Select **Actions** > **Start**.

Other Cloudera Manager Tasks and Settings

From the **Administration** tab you can select options for configuring settings that affect how Cloudera Manager interacts with your clusters.

Settings

The **Settings** page provides a number of categories as follows:

- **Performance** - Set the Cloudera Manager Agent heartbeat interval. See Configuring Agent Heartbeat and Health Status Options on page 34.
- **Advanced** - Enable API debugging and other advanced options.
- **Monitoring** - Set Agent health status parameters. For configuration instructions, see Configuring Cloudera Manager Agents on page 34.
- **Security** - Set TLS encryption settings to enable TLS encryption between the Cloudera Manager Server, Agents, and clients. For configuration instructions, see Configuring TLS Encryption for Cloudera Manager. You can also:
  - Set the realm for Kerberos security and point to a custom keytab retrieval script. For configuration instructions, see Cloudera Security.
  - Specify session timeout and a "Remember Me" option.

- **Ports and Addresses** - Set ports for the Cloudera Manager Admin Console and Server. For configuration instructions, see Configuring Cloudera Manager Server Ports on page 29.
- **Other**
  - Enable Cloudera usage data collection For configuration instructions, see Managing Anonymous Usage Data Collection on page 43.
  - Set a custom header color and banner text for the Admin console.
  - Set an "Information Assurance Policy" statement – this statement will be presented to every user before they are allowed to access the login dialog box. The user must click "I Agree" in order to proceed to the login dialog box.
  - Disable/enable the auto-search for the Events panel at the bottom of a page.

- **Support**
  - Configure diagnostic data collection properties. See Diagnostic Data Collection on page 44.
  - Configure how to access Cloudera Manager help files.

- **External Authentication** - Specify the configuration to use LDAP, Active Directory, or an external program for authentication. See Configuring External Authentication and Authorization for Cloudera Manager for instructions.

- **Parcels** - Configure settings for parcels, including the location of remote repositories that should be made available for download, and other settings such as the frequency with which Cloudera Manager will check for new parcels, limits on the number of downloads or concurrent distribution uploads. See Parcels for more information.
- **Network** - Configure proxy server settings. See Configuring Network Settings on page 37.
- **Custom Service Descriptors** - Configure custom service descriptor properties for Add-on Services on page 195.

Alerts

See Managing Alerts on page 287.

Users

See Cloudera Manager User Accounts.

Kerberos

See Enabling Kerberos Authentication for CDH.

License

See Managing Licenses on page 37.

User Interface Language

You can change the language of the Cloudera Manager Admin Console User Interface through the language preference in your browser. Information on how to do this for the browsers supported by Cloudera Manager is shown under the Administration page. You can also change the language for the information provided with activity and health events, and for alert email messages by selecting **Language**, selecting the language you want from the drop-down list on this page, then clicking **Save Changes**.

Peers

See Designating a Replication Source on page 486.

Cloudera Management Service

The Cloudera Management Service implements various management features as a set of roles:

- Activity Monitor - collects information about activities run by the MapReduce service. This role is not added by default.
- Host Monitor - collects health and metric information about hosts
- Service Monitor - collects health and metric information about services and activity information from the YARN and Impala services
- Event Server - aggregates relevant Hadoop events and makes them available for alerting and searching
- Alert Publisher - generates and delivers alerts for certain types of events
- Reports Manager - generates reports that provide an historical view into disk utilization by user, user group, and directory, processing activities by user and YARN pool, and HBase tables and namespaces. This role is not added in Cloudera Express.

Cloudera Manager manages each role separately, instead of as part of the Cloudera Manager Server, for scalability (for example, on large deployments it's useful to put the monitor roles on their own hosts) and isolation.

In addition, for certain editions of the Cloudera Enterprise license, the Cloudera Management Service provides the Navigator Audit Server and Navigator Metadata Server roles for Cloudera Navigator.

Displaying the Cloudera Management Service Status

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

Starting the Cloudera Management Service

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

1. Do one of the following:

    - 1. Select **Clusters** > **Cloudera Management Service**.
      2. Select **Actions** > **Start**.

    - 1. On the **Home** > **Status** tab, click

      to the right of **Cloudera Management Service** and select **Start**.

2. Click **Start** to confirm. The **Command Details** window shows the progress of starting the roles.
3. When **Command completed with *n/n* successful subcommands** appears, the task is complete. Click **Close**.

Stopping the Cloudera Management Service

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

1. Do one of the following:

    - 1. Select **Clusters** > **Cloudera Management Service**.
      2. Select **Actions** > **Stop**.

    - 1. On the **Home** > **Status** tab, click

      to the right of **Cloudera Management Service** and select **Stop**.

2. Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.
3. When **Command completed with *n/n* successful subcommands** appears, the task is complete. Click **Close**.

Restarting the Cloudera Management Service

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

1. Do one of the following:

    - 1. Select **Clusters** > **Cloudera Management Service**.
      2. Select **Actions** > **Restart**.

    - On the **Home** > **Status** tab, click

      to the right of **Cloudera Management Service** and select **Restart**.

2. Click **Restart** to confirm. The **Command Details** window shows the progress of stopping and then starting the roles.
3. When **Command completed with *n/n* successful subcommands** appears, the task is complete. Click **Close**.

Starting and Stopping Cloudera Management Service Roles

Minimum Required Role: <u>Navigator Administrator</u> (also provided by **Full Administrator**)

1. Do one of the following:

    - Select **Clusters** > **Cloudera Management Service**.
    - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Instances** tab.

3. Check the checkbox next to a role.
4. Do one of the following depending on your user role:

- **Minimum Required Role: Full Administrator**

  Choose and action:

  - Select **Actions for Selected** > **Start** and click **Start** to confirm.
  - Select **Actions for Selected** > **Stop** and click **Stop** to confirm.

- **Minimum Required Role: Navigator Administrator** (also provided by **Full Administrator**)

  1. Click a Cloudera Navigator Audit Server or Cloudera Navigator Metadata Server link.
  2. Choose an action:

     - Select **Actions** > **Start this XXX** and click **Start this XXX** to confirm, where *XXX* is the role name.
     - Select **Actions** > **Stop this XXX** and click **Stop this XXX** to confirm, where *XXX* is the role name.

Configuring Management Service Database Limits

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Each Cloudera Management Service role maintains a database for retaining the data it monitors. These databases (as well as the log files maintained by these services) can grow quite large. For example, the Activity Monitor maintains data at the service level, the activity level (MapReduce jobs and aggregate activities), and at the task attempt level. Limits on these data sets are configured when you create the management services, but you can modify these parameters through the Configuration settings in the Cloudera Manager Admin Console. For example, the Event Server lets you set a total number of events to store, and Activity Monitor gives you "purge" settings (also in hours) for the data it stores.

There are also settings for the logs that these various services create. You can throttle how big the logs are allowed to get and how many previous logs to retain.

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Configuration** tab.
3. Select **Scope** and then one of the following.

   - **Activity Monitor** - the **Purge** or **Expiration** period properties are found in the top-level settings for the role.
   - **Host Monitor** - see Data Storage for Monitoring Data on page 397.
   - **Service Monitor**

4. Select **Category** > **Log Files** to view log file size properties.
5. Edit the appropriate properties.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Click **Save Changes**.

Adding Cloudera Navigator Roles

Minimum Required Role: Navigator Administrator (also provided by **Full Administrator**)

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.

- On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Instances** tab.
3. Click the **Add Role Instances** button. The Customize Role Assignments page displays.
4. Assign the Navigator role to a host.

   a. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

   Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   - IP addresses
   - Rack name

   Click the **View By Host** button for an overview of the role assignment by hostname ranges.

5. When you are finished with the assignments, click **Continue**.
6. Choose the database type:

   - Keep the default setting of **Use Embedded Database** to have Cloudera Manager create and configure required databases. Record the auto-generated passwords.

Cluster Setup

Database Setup

Configure and test database connections. If using custom databases, create the databases first according to the **Installing and Configuring an External Database** section of the
Installation Guide ⬀ .

○ **Use Custom Databases**
◉ **Use Embedded Database**

When using the embedded database, passwords are automatically generated. Please copy them down.

Hive   ✓ Skipped. Cloudera Manager will create this database in a later step.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	hive	hive	t56iwbdk4F

Reports Manager   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	rman	rman	Y6S4IWvfNo

Navigator Audit Server   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	nav	nav	QLR2B0qqO9

Navigator Metadata Server   ✓ Successful

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	navms	navms	imo07jxOen

Oozie Server   ✓ Skipped. Cloudera Manager will create this database in a later step.

Currently assigned to run on **tcdn2-1.ent.cloudera.com**.

Database Host Name:	Database Type:	Database Name :	Username:	Password:
tcdn2-1.ent.cloudera.com:7432	PostgreSQL ⬍	oozie_oozie_se	oozie_oozie_se	NTF1KNdpPl

Test Connection

- Select **Use Custom Databases** to specify the external database host and enter the database type, database name, username, and password for the custom database.
- If you are adding the Oozie service, you can change your Oozie configuration to control when data is purged to improve performance, reduce database disk usage, improve upgrade performance, or to keep the history for a longer period of time. See Configuring Oozie Data Purge Settings Using Cloudera Manager.

7. Click **Test Connection** to confirm that Cloudera Manager can communicate with the database using the information you have supplied. If the test succeeds in all cases, click **Continue**; otherwise, check and correct the information you have provided for the database and then try the test again. (For some servers, if you are using the embedded database, you will see a message saying the database will be created at a later step in the installation process.)

The **Cluster Setup Review Changes** screen displays.

8. Click **Finish.**

Deleting Cloudera Navigator Roles

Minimum Required Role: Navigator Administrator (also provided by **Full Administrator**)

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Instances** tab.
3. Check the checkboxes next to the **Navigator Audit Server** and **Navigator Metadata Server** roles.

4. Do one of the following depending on your role:

- **Minimum Required Role: Full Administrator**

    1. Check the checkboxes next to the **Navigator Audit Server** and **Navigator Metadata Server** roles.
    2. Select **Actions for Selected** > **Stop** and click **Stop** to confirm.

- **Minimum Required Role: Navigator Administrator** (also provided by **Full Administrator**)

    1. Click the **Navigator Audit Server** role link.
    2. Select **Actions** > **Stop this Navigator Audit Server** and click **Stop this Navigator Audit Server** to confirm.
    3. Click the **Navigator Metadata Server** role link.
    4. Select **Actions** > **Stop this Navigator Metadata Server** and click **Stop this Navigator Metadata Server** to confirm.

5. Check the checkboxes next to the **Navigator Audit Server** and **Navigator Metadata Server** roles.
6. Select **Actions for Selected** > **Delete**. Click **Delete** to confirm the deletion.

Extending Cloudera Manager

In addition to the set of software packages and services managed by Cloudera Manager, you can also define and add new types of services using custom service descriptors. When you deploy a custom service descriptor, the implementation is delivered in a Cloudera Manager parcel or other software package. For information on the extension mechanisms provided by Cloudera Manager for creating custom service descriptors and parcels, see Cloudera Manager Extensions.

Cluster Configuration Overview

When Cloudera Manager configures a service, it allocates **roles** that are required for that service to the hosts in your cluster. The role determines which service daemons run on a host.

For example, for an HDFS service instance, Cloudera Manager configures:

- One host to run the NameNode role.
- One host to run as the secondary NameNode role.
- One host to run the Balancer role.
- Remaining hosts as to run DataNode roles.

A **role group** is a set of configuration properties for a role type, as well as a list of role instances associated with that group. Cloudera Manager automatically creates a default role group named **Role Type Default Group** for each role type.

When you run the installation or upgrade wizard, Cloudera Manager configures the default role groups it adds, and adds any other required role groups for a given role type. For example, a DataNode role on the same host as the NameNode might require a different configuration than DataNode roles running on other hosts. Cloudera Manager creates a separate role group for the DataNode role running on the NameNode host and uses the default configuration for DataNode roles running on other hosts.

Cloudera Manager wizards [autoconfigure](#) role group properties based on the resources available on the hosts. For properties that are not dependent on host resources, Cloudera Manager default values typically align with CDH default values for that configuration. Cloudera Manager deviates when the CDH default is not a recommended configuration or when the default values are illegal.

Server and Client Configuration

Administrators are sometimes surprised that modifying `/etc/hadoop/conf` and then restarting HDFS has no effect. That is because service instances started by Cloudera Manager do not read configurations from the default locations. To use HDFS as an example, when not managed by Cloudera Manager, there would usually be one HDFS configuration per host, located at `/etc/hadoop/conf/hdfs-site.xml`. Server-side daemons and clients running on the same host would all use that same configuration.

Cloudera Manager distinguishes between server and client configuration. In the case of HDFS, the file `/etc/hadoop/conf/hdfs-site.xml` contains only configuration relevant to an HDFS client. That is, by default, if you run a program that needs to communicate with Hadoop, it will get the addresses of the NameNode and JobTracker, and other important configurations, from that directory. A similar approach is taken for `/etc/hbase/conf` and `/etc/hive/conf`.

In contrast, the HDFS role instances (for example, NameNode and DataNode) obtain their configurations from a private per-process directory, under `/var/run/cloudera-scm-agent/process/`*unique-process-name*. Giving each process its own private execution and configuration environment allows Cloudera Manager to control each process independently. For example, here are the contents of an example `879-hdfs-NAMENODE` process directory:

```
$ tree -a /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
  /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
      cloudera_manager_Agent_fencer.py
      cloudera_manager_Agent_fencer_secret_key.txt
      cloudera-monitor.properties
      core-site.xml
      dfs_hosts_allow.txt
      dfs_hosts_exclude.txt
      event-filter-rules.json
      hadoop-metrics2.properties
      hdfs.keytab
      hdfs-site.xml
      log4j.properties
      logs
```

```
        stderr.log
        stdout.log
   topology.map
   topology.py
```

Distinguishing between server and client configuration provides several advantages:

- Sensitive information in the server-side configuration, such as the password for the Hive Metastore RDBMS, is not exposed to the clients.
- A service that depends on another service may deploy with customized configuration. For example, to get good HDFS read performance, Impala needs a specialized version of the HDFS client configuration, which may be harmful to a generic client. This is achieved by separating the HDFS configuration for the Impala daemons (stored in the per-process directory mentioned above) from that of the generic client (`/etc/hadoop/conf`).
- Client configuration files are much smaller and more readable. This also avoids confusing non-administrator Hadoop users with irrelevant server-side properties.

Modifying Configuration Properties Using Cloudera Manager

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

When a service is added to Cloudera Manager, either through the installation or upgrade wizard or with the Add Services workflow, Cloudera Manager automatically sets the configuration properties, based on the needs of the service and characteristics of the cluster in which it will run. These configuration properties include both service-wide configuration properties, as well as specific properties for each role type associated with the service, managed through role groups. A *role group* is a set of configuration properties for a role type, as well as a list of role instances associated with that group. Cloudera Manager automatically creates a default role group named *Role Type* **Default Group** for each role type. See Role Groups on page 212.

Changing the Configuration of a Service or Role Instance

1. Go to the service status page. (**Cluster** > *service name*)
2. Click the **Configuration** tab.
3. Locate the property you want to edit. You can type all or part of the property name in the search box, or use the filters on the left side of the screen:

   - The **Status** section limits the displayed properties by their status. Possible statuses include:

     - Error
     - Warning
     - Edited
     - Non-default
     - Has Overrides

   - The **Scope** section of the left hand panel organizes the configuration properties by role types; first those that are **Service-Wide**, followed by various role types within the service. When you select one of these roles, a set of properties whose values are managed by the default role group for the role display. Any additional role groups that apply to the property also appear in this panel and you can modify values for each role group just as you can the default role group.

   - The **Category** section of the left hand panel allows you to limit the displayed properties by category.

4. Edit the property value.

   - To facilitate entering some types of values, you can specify not only the value, but also the units that apply to the value. for example, to enter a setting that specifies bytes per second, you can choose to enter the value in bytes (B), KiBs, MiBs, or GiBs—selected from a drop-down menu that appears when you edit the value.

- If the property allows a list of values, click the ✚ icon to the right of the edit field to add an additional field. An example of this is the HDFS DataNode Data Directory property, which can have a comma-delimited list of directories as its value. To remove an item from such a list, click the ━ icon to the right of the field you want to remove.

Many configuration properties have different values that are configured by multiple role groups. (See Role Groups on page 212).

To edit configuration values for multiple role groups:

1. Go to the property, For example, the configuration panel for the **Heap Dump Directory** property displays the DataNode Default Group (a role group), and a link that says **... and 6 others**.

2. Click the **... and 6 others** link to display all of the role groups:

3. Click the **Show fewer** link to collapse the list of role groups.

If you edit the single value for this property, Cloudera Manager applies the value to all role groups. To edit the values for one or more of these role groups individually, click **Edit Individual Values**. Individual fields display where you can edit the values for each role group. For example:

5. Click **Save Changes** to commit the changes. You can add a note that is included with the change in the Configuration History. This changes the setting for the role group, and applies to all role instances associated with that role group. Depending on the change you made, you may need to restart the service or roles associated with the configuration you just changed. Or, you may need to redeploy your client configuration for the service. You should see a message to that effect at the top of the Configuration page, and services will display an outdated configuration

   (Restart Needed),(Refresh Needed), or outdated client configuration indicator. Click the indicator to display the Stale Configurations on page 77 page.

Searching for Properties

You can use the **Search** box to search for properties by name or label. The search also returns properties whose description matches your search term.

Validation of Configuration Properties

Cloudera Manager validates the values you specify for configuration properties. If you specify a value that is outside the recommended range of values or is invalid, Cloudera Manager displays a warning at the top of the **Configuration** tab and in the text box after you click **Save Changes**. The warning is yellow if the value is outside the recommended range of values and red if the value is invalid.

Overriding Configuration Properties

For role types that allow multiple instances, each role instance inherits its configuration properties from its associated role group. While role groups provide a convenient way to provide alternate configuration properties for selected groups of role instances, there may be situations where you want to make a one-off configuration change—for example when a host has malfunctioned and you want to temporarily reconfigure it. In this case, you can override configuration properties for a specific role instance:

1. Go to the **Status** page for the service whose role you want to change.
2. Click the **Instances** tab.
3. Click the role instance you want to change.
4. Click the **Configuration** tab.
5. Change the configuration values as appropriate.
6. Save your changes.
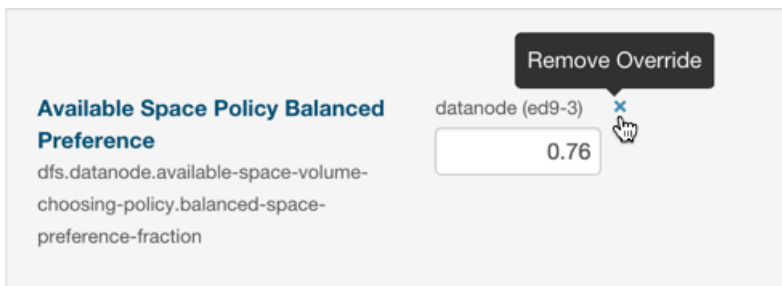
You will most likely need to restart your service or role to have your configuration changes take effect. See Stale Configuration Actions on page 79.

Viewing and Editing Overridden Configuration Properties

To see a list of all role instances that have an override value for a particular configuration setting, go to the Status page for the service and select **Status** > **Has overrides**. A list of configuration properties where values have been overridden displays. The panel for each configuration property displays the values for each role group or instance. You can edit the value of this property for this instance, or, you can click the

   ✖

icon next to an instance name to remove the overridden value.

Resetting Configuration Properties to the Default Value

To reset a property back to its default value, click the

icon. The default value is inserted and the icon turns into an Undo icon

.   )

Explicitly setting a configuration to the same value as its default (inherited value) has the same effect as using the

icon.

Click to revert to default: 50010

DataNode
Transceiver Port
dfs.datanode.address

DataNode Default Group

20002
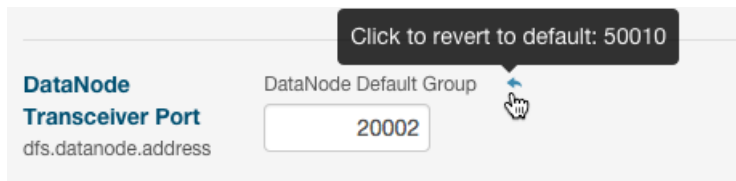
There is no mechanism for resetting to an [autoconfigured](#) value. However, you can use the configuration [history and rollback feature](#) to revert any configuration changes.

Viewing and Editing Host Overrides

You can override the properties of individual hosts in your cluster.

1. Click the **Hosts** tab.
2. Click the **Configuration** tab.
3. Use the Filters or Search box to locate the property that you want to override.
4. Click the **Manage Host Overrides** link.

Cloudera Manager
Agent Process
Directory Free
Space Monitoring
Absolute
Thresholds

Warning: Specify ▲▼   200   MiB ▲▼

Critical: Specify ▲▼   100   MiB ▲▼

Manage Host Overrides

❓ Requires Agent
Restart
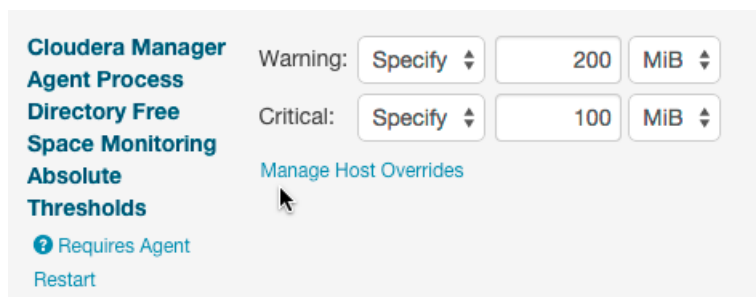
The **Manage Overrides** dialog box displays.

5. Select one or more hosts to override this property.
6. Click **Update**.

A new entry area displays where you can enter the override values. In the example below, servers `ed9-e.ent.cloudera.com` and `ed9-r.cloudera.com` were selected for overrides. Note that the first set of fields displays the value set for all hosts and the two sets of fields that follow allow you to edit the override values for each specified host.

To remove the override, click the

❌

icon next to the hostname.

To apply the same value to all hosts, click **Edit Identical Values**. Click **Edit Individual Values** to apply different values to selected hosts.

7. If the property indicates **Requires Agent Restart**, restart the agent on the affected hosts.

Restarting Services and Instances after Configuration Changes

If you change the configuration properties after you start a service or instance, you may need to restart the service or instance to have the configuration properties become active. If you change configuration properties at the service level that affect a particular role only (such as all DataNodes but not the NameNodes), you can restart only that role; you do not need to restart the entire service. If you changed the configuration for a particular role instance (such as one of four DataNodes), you may need to restart only that instance.

1. Follow the instructions in Restarting a Service on page 198 or Starting, Stopping, and Restarting Role Instances on page 210.
2. If you see a **Finished** status, the service or role instances have restarted.
3. Go to the **Home** > **Status** tab. The service should show a Status of **Started** for all instances and a health status of **Good**.

For more information, see Stale Configurations on page 77.

Suppressing Configuration and Parameter Validation Warnings

You can suppress the warnings that Cloudera Manager issues when a configuration value is outside the recommended range or is invalid. If a warning does not apply to your deployment, you might want to suppress it. Suppressed validation warnings are still retained by Cloudera Manager, and you can unsuppress the warnings at any time. You can suppress each warning when you view it, or you can configure suppression for a specific validation before warnings occur.

Suppressing a Configuration Validation in Cloudera Manager

1. Click the **Suppress...** link to suppress the warning.

A dialog box opens where you can enter a comment about the suppression.

2. Click **Confirm**.

You can also suppress warnings from the **All Configuration Issues** screen:

1. Browse to the **Home** screen.
2. Click **Configurations** > **Configuration Issues**.
3. Locate the validation message in the list and click the **Suppress...** link.

   A dialog box opens where you can enter a comment about the suppression.

4. Click **Confirm**.

The suppressed validation warning is now hidden.

Managing Suppressed Validations

On pages where you have suppressed validations, you see a link that says **Show # Suppressed Warning(s)**. On this screen, you can:

- Click the **Show # Suppressed Warning(s)** link to show the warnings.

  Each suppressed warning displays an icon: .

- Click the **Unsuppress...** link to unsuppress the configuration validation.
- Click the **Hide Suppressed Warnings** link to re-hide the suppressed warnings.

Suppressing Configuration Validations Before They Trigger Warnings

1. Go to the service or host with the configuration validation warnings you want to suppress.
2. Click **Configuration**.
3. In the filters on the left, select **Category** > **Suppressions**.

   A list of suppression properties displays. The names of the properties begin with **Suppress Parameter Validation** or **Suppress Configuration Validator**. You can also use the **Search** function to limit the number of properties that display.

4. Select a suppression property to suppress the validation warning.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Viewing a List of All Suppressed Validations

Do one of the following:

- From the **Home** page or the **Status** page of a cluster, select **Configuration** > **Suppressed Health and Configuration Issues**.
- From the **Status** page of a service, select **Configuration** > **Category** > **Suppressions** and select **Status** > **Non-default**.
- From the **Host** tab, select **Configuration** > **Category** > **Suppressions** and select **Status** > **Non-default**.

Autoconfiguration

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Cloudera Manager provides several interactive wizards to automate common workflows:

- Installation - used to bootstrap a Cloudera Manager deployment
- Add Cluster - used when adding a new cluster
- Add Service - used when adding a new service
- Upgrade - used when upgrading to a new version of CDH
- Static Service Pools - used when configuring static service pools
- Import MapReduce - used when migrating from MapReduce to YARN

Cluster Configuration Overview

In some of these wizards, Cloudera Manager uses a set of rules to automatically configure certain settings to best suit the characteristics of the deployment. For example, the number of hosts in the deployment drives the memory requirements for certain monitoring daemons: the more hosts, the more memory is needed. Additionally, wizards that are tasked with creating new roles will use a similar set of rules to determine an ideal host placement for those roles.

Scope

The following table shows, for each wizard, the scope of entities it affects during autoconfiguration and role-host placement.

Wizard	Autoconfiguration Scope	Role-Host Placement Scope
Installation	New cluster, Cloudera Management Service	New cluster, Cloudera Management Service
Add Cluster	New cluster	New cluster
Add Service	New service	New service
Upgrade	Cloudera Management Service	Cloudera Management Service
Static Service Pools	Existing cluster	N/A
Import MapReduce	Existing YARN service	N/A

Certain autoconfiguration rules are unscoped, that is, they configure settings belonging to entities that aren't necessarily the entities under the wizard's scope. These exceptions are explicitly listed.

Autoconfiguration

Cloudera Manager employs several different rules to drive automatic configuration, with some variation from wizard to wizard. These rules range from the simple to the complex.

Configuration Scope

One of the points of complexity in autoconfiguration is configuration scope. The configuration hierarchy as it applies to services is as follows: configurations may be modified at the service level (affecting every role in the service), role group level (affecting every role instance in the group), or role level (affecting one role instance). A configuration found in a lower level takes precedence over a configuration found in a higher level.

With the exception of the Static Service Pools, and the Import MapReduce wizard, all Cloudera Manager wizards follow a basic pattern:

1. Every role in scope is moved into its own, new, role group.
2. This role group is the receptacle for the role's "idealized" configuration. Much of this configuration is driven by properties of the role's host, which can vary from role to role.
3. Once autoconfiguration is complete, new role groups with common configurations are merged.
4. The end result is a smaller set of role groups, each with an "idealized" configuration for some subset of the roles in scope. A subset can have any number of roles; perhaps all of them, perhaps just one, and so on.

The Static Service Pools and Import MapReduce wizards configure role groups directly and do not perform any merging.

Static Service Pools

Certain rules are only invoked in the context of the Static Service Pools wizard. Additionally, the wizard autoconfigures cgroup settings for certain kinds of roles:

- HDFS DataNodes
- HBase RegionServers
- MapReduce TaskTrackers
- YARN NodeManagers
- Impala Daemons

- Solr Servers
- Spark Standalone Workers
- Accumulo Tablet Servers
- Add-on services

YARN

`yarn.nodemanager.resource.cpu-vcores` - For each NodeManager role group, set to `number of cores, including hyperthreads, on one NodeManager member's host * service percentage chosen in wizard.`

All Services

Cgroup `cpu.shares` - For each role group that supports `cpu.shares`, set to `max(20, (service percentage chosen in wizard) * 20).`

Cgroup `blkio.weight` - For each role group that supports `blkio.weight`, set to `max(100, (service percentage chosen in wizard) * 10).`

Data Directories

Several autoconfiguration rules work with data directories, and there's a common sub-rule used by all such rules to determine, out of all the mountpoints present on a host, which are appropriate for data. The subrule works as follows:

- The initial set of mountpoints for a host includes all those that are disk-backed. Network-backed mountpoints are excluded.
- Mountpoints beginning with `/boot`, `/cdrom`, `/usr`, `/tmp`, `/home`, or `/dev` are excluded.
- Mountpoints beginning with /media are excluded, unless the backing device's name contains `/xvd` somewhere in it.
- Mountpoints beginning with `/var` are excluded, unless they are `/var` or `/var/lib`.
- The largest mount point (in terms of total space, not available space) is determined.
- Other mountpoints with less than 1% total space of the largest are excluded.
- Mountpoints beginning with `/var` or equal to `/` are excluded unless they're the largest mount point.
- Remaining mountpoints are sorted lexicographically and retained for future use.

Memory

The rules used to autoconfigure memory reservations are perhaps the most complicated rules employed by Cloudera Manager. When configuring memory, Cloudera Manager must take into consideration which roles are likely to enjoy more memory, and must not over commit hosts if at all possible. To that end, it needs to consider each host as an entire unit, partitioning its available RAM into segments, one segment for each role. To make matters worse, some roles have more than one memory segment. For example, a Solr server has two memory segments: a JVM heap used for most memory allocation, and a JVM direct memory pool used for HDFS block caching. Here is the overall flow during memory autoconfiguration:

1. The set of participants includes every host under scope as well as every {role, memory segment} pair on those hosts. Some roles are under scope while others are not.
2. For each {role, segment} pair where the role is under scope, a rule is run to determine four different values for that pair:

   - Minimum memory configuration. Cloudera Manager must satisfy this minimum, possibly over-committing the host if necessary.
   - Minimum memory consumption. Like the above, but possibly scaled to account for inherent overhead. For example, JVM memory values are multiplied by 1.3 to arrive at their consumption value.
   - Ideal memory configuration. If RAM permits, Cloudera Manager will provide the pair with all of this memory.
   - Ideal memory consumption. Like the above, but scaled if necessary.

3. For each {role, segment} pair where the role is not under scope, a rule is run to determine that pair's existing memory consumption. Cloudera Manager will not configure this segment but will take it into consideration by setting the pair's "minimum" and "ideal" to the memory consumption value.

4. For each host, the following steps are taken:

   a. 20% of the host's available RAM is subtracted and reserved for the OS.

   b. sum(minimum_consumption) and sum(ideal_consumption) are calculated.

   c. An "availability ratio" is built by comparing the two sums against the host's available RAM.

      a. If RAM < sum(minimum) ratio = 0

      b. If RAM >= sum(ideal) ratio = 1

   d. If the host has more available memory than the total of the ideal memory for all roles assigned to the host, each role is assigned its ideal memory and autoconfiguration is finished.

   e. Cloudera Manager assigns all available host memory by setting each {role, segment} pair to the same consumption value, except in cases where that value is below the minimum memory or above the ideal memory for that pair. In that case, it is set to the minimum memory or the ideal memory as appropriate. This ensures that pairs with low ideal memory requirements are completely satisfied before pairs with higher ideal memory requirements.

5. The {role, segment} pair is set with the value from the previous step. In the Static Service Pools wizard, the role group is set just once (as opposed to each role).

6. Custom post-configuration rules are run.

Customization rules are applied in steps 2, 3 and 7. In step 2, there's a generic rule for most cases, as well as a series of custom rules for certain {role, segment} pairs. Likewise, there's a generic rule to calculate memory consumption in step 3 as well as some custom consumption functions for certain {role, segment} pairs.

Step 2 Generic Rule Excluding Static Service Pools Wizard

For every {role, segment} pair where the segment defines a default value, the pair's minimum is set to the segment's minimum value (or 0 if undefined), and the ideal is set to the segment's default value.

Step 2 Custom Rules Excluding Static Service Pools Wizard

HDFS

For the NameNode and Secondary NameNode JVM heaps, the minimum is 50 MB and the ideal is `max(4 GB, sum_over_all(DataNode mountpoints' available space) / 0.000008)`.

MapReduce

For the JobTracker JVM heap, the minimum is 50 MB and the ideal is `max(1 GB, round((1 GB * 2.3717181092 * ln(number of TaskTrackers in MapReduce service)) – 2.6019933306))`. If the number of TaskTrackers <= 5, the ideal is 1 GB.

For the mapper JVM heaps, the minimum is 1 and the ideal is the number of cores, including hyperthreads, on the TaskTracker host. Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a task's heap).

For the reducer JVM heaps, the minimum is 1 and the ideal is `(number of cores, including hyperthreads, on the TaskTracker host) / 2`. Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a task's heap).

HBase

For the memory total allowed for HBase RegionServer JVM heap, the minimum is 50 MB and the ideal is `min (31 GB ,(total RAM on region server host) * 0.64)`

YARN

For the memory total allowed for containers, the minimum is 1 GB and the ideal is `(total RAM on NodeManager host) * 0.64`.

Hue

With the exception of the Beeswax Server (only in CDH 4), Hue roles do not have memory limits. Therefore, Cloudera Manager treats them as roles that consume a fixed amount of memory by setting their minimum and ideal consumption values, but not their configuration values. The two consumption values are set to 256 MB.

Impala

With the exception of the Impala daemon, Impala roles do not have memory limits. Therefore, Cloudera Manager treats them as roles that consume a fixed amount of memory by setting their minimum/ideal consumption values, but not their configuration values. The two consumption values are set to 150 MB for the Catalog Server and 64 MB for the StateStore.

For the Impala Daemon memory limit, the minimum is 256 MB and the ideal is `(total RAM on daemon host) * 0.64`.

Solr

For the Solr Server JVM heap, the minimum is 50 MB and the ideal is `min(64 GB, (total RAM on Solr Server host) * 0.64) / 2.6`. For the Solr Server JVM direct memory segment, the minimum is 256 MB and the ideal is `min(64 GB, (total RAM on Solr Server host) * 0.64) / 2`.

Cloudera Management Service

- Alert Publisher JVM heap - Treated as if it consumed a fixed amount of memory by setting the minimum/ideal consumption values, but not the configuration values. The two consumption values are set to 256 MB.
- Service and Host Monitor JVM heaps - The minimum is 50 MB and the ideal is either 256 MB (10 or fewer managed hosts), 1 GB (100 or fewer managed hosts), or 2 GB (over 100 managed hosts).
- Event Server, Reports Manager, and Navigator Audit Server JVM heaps - The minimum is 50 MB and the ideal is 1 GB.
- Navigator Metadata Server JVM heap - The minimum is 512 MB and the ideal is 2 GB.
- Service and Host Monitor off-heap memory segments - The minimum is either 768 MB (10 or fewer managed hosts), 2 GB (100 or fewer managed hosts), or 6 GB (over 100 managed hosts). The ideal is always twice the minimum.

Step 2 Generic Rule for Static Service Pools Wizard

For every {role, segment} pair where the segment defines a default value and an autoconfiguration share, the pair's minimum is set to the segment's default value, and the ideal is set to `min((segment soft max (if exists) or segment max (if exists) or 2^63-1), (total RAM on role's host * 0.8 / segment scale factor * service percentage chosen in wizard * segment autoconfiguration share))`.

Autoconfiguration shares are defined as follows:

- HBase RegionServer JVM heap: 1
- HDFS DataNode JVM heap: 1 in CDH 4, 0.2 in CDH 5
- HDFS DataNode maximum locked memory: 0.8 (CDH 5 only)
- Solr Server JVM heap: 0.5
- Solr Server JVM direct memory: 0.5
- Spark Standalone Worker JVM heap: 1
- Accumulo Tablet Server JVM heap: 1
- Add-on services: any

Roles not mentioned here do not define autoconfiguration shares and thus aren't affected by this rule.

Additionally, there's a generic rule to handle `cgroup.memory_limit_in_bytes`, which is unused by Cloudera services but is available for add-on services. Its behavior varies depending on whether the role in question has segments or not.

With Segments

The minimum is the `min(cgroup.memory_limit_in_bytes_min (if exists) or 0, sum_over_all(segment minimum consumption))`, and the ideal is the sum of all segment ideal consumptions.

Without Segments

The minimum is `cgroup.memory_limit_in_bytes_min (if exists)` or 0, and the ideal is `(total RAM on role's host * 0.8 * service percentage chosen in wizard)`.

Step 3 Custom Rules for Static Service Pools Wizard

YARN

For the memory total allowed for containers, the minimum is 1 GB and the ideal is `min(8 GB, (total RAM on NodeManager host) * 0.8 * service percentage chosen in wizard)`.

Impala

For the Impala Daemon memory limit, the minimum is 256 MB and the ideal is `((total RAM on Daemon host) * 0.8 * service percentage chosen in wizard)`.

MapReduce

- Mapper JVM heaps - the minimum is 1 and the ideal is (number of cores, including hyperthreads, on the TaskTracker host * service percentage chosen in wizard). Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a given task's heap).
- Reducer JVM heaps - the minimum is 1 and the ideal is (number of cores, including hyperthreads on the TaskTracker host * service percentage chosen in wizard) / 2. Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a given task's heap).

Step 3 Generic Rule

For every {role, segment} pair, the segment's current value is converted into bytes, and then multiplied by the scale factor (1.0 by default, 1.3 for JVM heaps, and freely defined for Custom Service Descriptor services).

Step 3 Custom Rules

Impala

For the Impala Daemon, the memory consumption is 0 if YARN Service for Resource Management is set. If the memory limit is defined but not -1, its value is used verbatim. If it's defined but -1, the consumption is equal to the total RAM on the Daemon host. If it is undefined, the consumption is (total RAM * 0.8).

MapReduce

See [Step 3 Custom Rules for Static Service Pools Wizard](#) on page 70.

Solr

For the Solr Server JVM direct memory segment, the consumption is equal to the value verbatim provided `solr.hdfs.blockcache.enable` and `solr.hdfs.blockcache.direct.memory.allocation` are both true. Otherwise, the consumption is 0.

Step 7 Custom Rules

HDFS

- NameNode JVM heaps are equalized. For every pair of NameNodes in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.
- JournalNode JVM heaps are equalized. For every pair of JournalNodes in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.
- NameNode and Secondary NameNode JVM heaps are equalized. For every {NameNode, Secondary NameNode} pair in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.

HBase

Master JVM heaps are equalized. For every pair of Masters in an HBase service with different heap sizes, the larger heap size is reset to the smaller one.

Hive

Hive on Spark rules apply only when Hive depends on YARN. The following rules are applied:

- Spark executor cores - Set to 4, 5, or 6. The value that results in the fewest "wasted" cores across the cluster is used, where the number of cores wasted per host is the remainder of `yarn.nodemanager.resource.cpu-vcores / spark.executor.cores`. In case of a tie, use the larger value of Spark executor cores. If no host on the cluster has 4 or more cores, then sets the value to the smallest value of `yarn.nodemanager.resource.cpu-vcores` on the cluster.
- Spark executor memory - 85% of Spark executor memory allocated to `spark.executor.memory` and 15% allocated to `spark.yarn.executor.memoryOverhead`. The total memory is the YARN container memory split evenly between the maximum number of executors that can run on a host. This is `yarn.nodemanager.resource.memory-mb / floor(yarn.nodemanager.resource.cpu-vcores / spark.executor.cores)`. When the memory or vcores vary across hosts in the cluster, choose the smallest calculated value for Spark executor memory.
- Spark driver memory - 90% of Spark driver memory allocated to `spark.driver.memory` and 10% allocated to `spark.yarn.driver.memoryOverhead`. The total memory is based on the lowest value of `yarn.nodemanager.resource.memory-mb` across the cluster.
- Total memory is:
  - 12 GB when `yarn.nodemanager.resource.memory-mb` > 50 GB.
  - 4 GB when `yarn.nodemanager.resource.memory-mb` < 50 GB && >= 12 GB
  - 1 GB when `yarn.nodemanager.resource.memory-mb` < 12 GB
  - 256 MB when `yarn.nodemanager.resource.memory-mb` < 1 GB.

Impala

If an Impala service has YARN Service for Resource Management set, every Impala Daemon memory limit is set to the value of (`yarn.nodemanager.resource.memory-mb` * 1 GB) if there's a YARN NodeManager co-located with the Impala Daemon.

MapReduce

JobTracker JVM heaps are equalized. For every pair of JobTrackers in an MapReduce service with different heap sizes, the larger heap size is reset to the smaller one.

Oozie

Oozie Server JVM heaps are equalized. For every pair of Oozie Servers in an Oozie service with different heap sizes, the larger heap size is reset to the smaller one.

YARN

ResourceManager JVM heaps are equalized. For every pair of ResourceManagers in a YARN service with different heap sizes, the larger heap size is reset to the smaller one.

ZooKeeper

ZooKeeper Server JVM heaps are equalized. For every pair of servers in a ZooKeeper service with different heap sizes, the larger heap size is reset to the smaller one.

General Rules

HBase

- `hbase.replication` - For each HBase service, set to true if there's a Key-Value Store Indexer service in the cluster. *This rule is unscoped; it can fire even if the HBase service is not under scope.*
- `replication.replicationsource.implementation` - For each HBase service, set to `com.ngdata.sep.impl.SepReplicationSource` if there's a Keystore Indexer service in the cluster. *This rule is unscoped; it can fire even if the HBase service is not under scope.*

HDFS

- `dfs.datanode.du.reserved` - For each DataNode, set to `min((total space of DataNode host largest mountpoint) / 10, 10 GB)`.
- `dfs.namenode.name.dir` - For each NameNode, set to the first two mountpoints on the NameNode host with `/dfs/nn` appended.
- `dfs.namenode.checkpoint.dir` - For each Secondary NameNode, set to the first mountpoint on the Secondary NameNode host with `/dfs/snn` appended.
- `dfs.datanode.data.dir` - For each DataNode, set to all the mountpoints on the host with `/dfs/dn` appended.
- `dfs.journalnode.edits.dir` - For each JournalNode, set to the first mountpoint on the JournalNode host with `/dfs/jn` appended.
- `dfs.datanode.failed.volumes.tolerated` - For each DataNode, set to (number of mountpoints on DataNode host) / 2.
- `dfs.namenode.service.handler.count` and `dfs.namenode.handler.count` - For each NameNode, set to `ln(number of DataNodes in this HDFS service) * 20`.
- `dfs.datanode.hdfs-blocks-metadata.enabled` - For each HDFS service, set to true if there's an Impala service in the cluster. *This rule is unscoped; it can fire even if the HDFS service is not under scope.*
- `dfs.client.read.shortcircuit` - For each HDFS service, set to true if there's an Impala service in the cluster. *This rule is unscoped; it can fire even if the HDFS service is not under scope.*
- `dfs.datanode.data.dir.perm` - For each DataNode, set to 755 if there's an Impala service in the cluster and the cluster isn't Kerberized. *This rule is unscoped; it can fire even if the HDFS service is not under scope.*
- `fs.trash.interval` - For each HDFS service, set to 1.

Hue

- **WebHDFS dependency** - For each Hue service, set to either the first HttpFS role in the cluster, or, if there are none, the first NameNode in the cluster.
- **HBase Thrift Server dependency**- For each Hue service in a CDH 4.4 or higher cluster, set to the first HBase Thrift Server in the cluster.

Impala

For each Impala service, set **Enable Audit Collection** and **Enable Lineage Collection** to true if there's a Cloudera Management Service with a Navigator Audit Server and Navigator Metadata Server roles. *This rule is unscoped; it can fire even if the Impala service is not under scope.*

MapReduce

- `mapred.local.dir` - For each JobTracker, set to the first mountpoint on the JobTracker host with `/mapred/jt` appended.
- `mapred.local.dir` - For each TaskTracker, set to all the mountpoints on the host with `/mapred/local` appended.
- `mapred.reduce.tasks` - For each MapReduce service, set to `max(1, sum_over_all(TaskTracker number of reduce tasks (determined via mapred.tasktracker.reduce.tasks.maximum for that TaskTracker, which is configured separately)) / 2)`.
- `mapred.job.tracker.handler.count` - For each JobTracker, set to `max(10, ln(number of TaskTrackers in this MapReduce service) * 20)`.
- `mapred.submit.replication` - If there's an HDFS service in the cluster, for each MapReduce service, set to `max(min(number of DataNodes in the HDFS service, value of HDFS Replication Factor), sqrt(number of DataNodes in the HDFS service))`.
- `mapred.tasktracker.instrumentation` - If there's a management service, for each MapReduce service, set to `org.apache.hadoop.mapred.TaskTrackerCmonInst`. *This rule is unscoped; it can fire even if the MapReduce service is not under scope.*

YARN

- `yarn.nodemanager.local-dirs` - For each NodeManager, set to all the mountpoints on the NodeManager host with `/yarn/nm` appended.
- `yarn.nodemanager.resource.cpu-vcores` - For each NodeManager, set to the number of cores (including hyperthreads) on the NodeManager host.
- `mapred.reduce.tasks` - For each YARN service, set to `max(1,sum_over_all(NodeManager number of cores, including hyperthreads) / 2)`.
- `yarn.resourcemanager.nodemanagers.heartbeat-interval-ms` - For each NodeManager, set to `max(100, 10 * (number of NodeManagers in this YARN service))`.
- `yarn.scheduler.maximum-allocation-vcores` - For each ResourceManager, set to `max_over_all(NodeManager number of vcores (determined via yarn.nodemanager.resource.cpu-vcores for that NodeManager, which is configured separately))`.
- `yarn.scheduler.maximum-allocation-mb` - For each ResourceManager, set to `max_over_all(NodeManager amount of RAM (determined via yarn.nodemanager.resource.memory-mb for that NodeManager, which is configured separately))`.
- `mapreduce.client.submit.file.replication` - If there's an HDFS service in the cluster, for each YARN service, set to `max(min(number of DataNodes in the HDFS service, value of HDFS Replication Factor), sqrt(number of DataNodes in the HDFS service))`.

All Services

If a service dependency is unset, and a service with the desired type exists in the cluster, set the service dependency to the first such target service. Applies to all service dependencies except YARN Service for Resource Management. Applies only to the Installation and Add Cluster wizards.

Role-Host Placement

Cloudera Manager employs the same role-host placement rule regardless of wizard. The set of hosts considered depends on the scope. If the scope is a cluster, all hosts in the cluster are included. If a service, all hosts in the service's cluster are included. If the Cloudera Management Service, all hosts in the deployment are included. The rules are as follows:

1. The hosts are sorted from most to least physical RAM. Ties are broken by sorting on hostname (ascending) followed by host identifier (ascending).
2. The overall number of hosts is used to determine which arrangement to use. These arrangements are hard-coded, each dictating for a given "master" role type, what index (or indexes) into the sorted host list in step 1 to use.

3. Master role types are included based on several factors:

- Is this role type part of the service (or services) under scope?
- Does the service already have the right number of instances of this role type?
- Does the cluster's CDH version support this role type?
- Does the installed Cloudera Manager license allow for this role type to exist?

4. Master roles are placed on each host using the indexes and the sorted host list. If a host already has a given master role, it is skipped.

5. An HDFS DataNode is placed on every host outside of the arrangement described in step 2, provided HDFS is one of the services under scope.

6. Certain "worker" roles are placed on every host where an HDFS DataNode exists, either because it existed there prior to the wizard, or because it was added in the previous step. The supported worker role types are:

- MapReduce TaskTrackers
- YARN NodeManagers
- HBase RegionServers
- Impala Daemons
- Spark Workers

7. Hive gateways are placed on every host, provided a Hive service is under scope and a gateway didn't already exist on a given host.

8. Spark on YARN gateways are placed on every host, provided a Spark on YARN service is under scope and a gateway didn't already exist on a given host.

This rule merely dictates the *default* placement of roles; you are free to modify it before it is applied by the wizard.

Custom Configuration

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Cloudera Manager exposes properties that allow you to insert custom configuration text into XML configuration, property, and text files, or into an environment. The naming convention for these properties is: *XXX* **Advanced Configuration Snippet (Safety Valve)** for *YYY* or *XXX YYY* **Advanced Configuration Snippet (Safety Valve)**, where *XXX* is a service or role and *YYY* is the target.

The values you enter into a configuration snippet must conform to the syntax of the target. For an XML configuration file, the configuration snippet must contain valid XML property definitions. For a properties file, the configuration snippet must contain valid property definitions. Some files simply require a list of host addresses.

The configuration snippet mechanism is intended for use in cases where there is configuration setting that is not exposed as a configuration property in Cloudera Manager. Configuration snippets generally override normal configuration. Contact Cloudera Support if you are required to use a configuration snippet that is not explicitly documented.

Service-wide configuration snippets apply to all roles in the service; a configuration snippet for a role group applies to all instances of the role associated with that role group.

Server and client configurations have separate configuration snippets. In general after changing a server configuration snippet you must restart the server, and after changing a client configuration snippet you must redeploy the client configuration. Sometimes you can refresh instead of restart. In some cases however, you must restart a dependent server after changing a client configuration. For example, changing a MapReduce client configuration marks the dependent Hive server as stale, which must be restarted. The Admin Console displays an indicator when a server must be restarted. In addition, the All Configuration Issues tab on the Home page indicates the actions you must perform to resolve stale configurations.

Configuration Snippet Types and Syntax

Configuration

Set configuration properties in various configuration files; the property name indicates into which configuration file the configuration will be placed. Configuration files have the extension `.xml` or `.conf`.

For example, there are several configuration snippets for the Hive service. One Hive configuration snippet property is called the **HiveServer2 Advanced Configuration Snippet for hive-site.xml**; configurations you enter here are inserted verbatim into the `hive-site.xml` file associated with the HiveServer2 role group.

To see a list of configuration snippets that apply to a specific configuration file, enter the configuration file name in the Search field in the top navigation bar. For example, searching for `mapred-site.xml` shows the configuration snippets that have `mapred-site.xml` in their name.

Some configuration snippet descriptions include the phrase *for this role only*. These configurations are stored in memory, and only inserted to the configuration when running an application from Cloudera Manager. Otherwise, the configuration changes are added to the configuration file on disk, and are used when running the application both from Cloudera Manager and from the command line.

Syntax:

```
<property>
  <name>property_name</name>
  <value>property_value</value>
</property>
```

For example, to specify a MySQL connector library, put this property definition in that configuration snippet:

```
<property>
  <name>hive.aux.jars.path</name>
  <value>file:///usr/share/java/mysql-connector-java.jar</value>
</property>
```

Environment

Specify key-value pairs for a service, role, or client that are inserted into the respective environment.

One example of using an environment configuration snippet is to add a JAR to a classpath. Place JARs in a custom location such as `/opt/myjars` and extend the classpath using the appropriate service environment configuration snippet. The value of a JAR property must conform to the syntax supported by its environment. See Setting the class path.

Do not place JARs inside locations such as `/opt/cloudera` or `/usr/lib/{hadoop*,hbase*,hive*}` that are managed by Cloudera because they are overwritten at upgrades.

Syntax:

```
key=value
```

For example, to add JDBC connectors to a Hive gateway classpath, add

```
AUX_CLASSPATH=/usr/share/java/mysql-connector-java.jar:\
/usr/share/java/oracle-connector-java.jar
```

or

```
AUX_CLASSPATH=/usr/share/java/*
```

to **Gateway Client Advanced Configuration Snippet for hive-env.sh**.

Logging

Set log4j properties in a `log4j.properties` file.

Syntax:

```
key1=value1
key2=value2
```

For example:

```
log4j.rootCategory=INFO, console max.log.file.size=200MB
max.log.file.backup.index=10
```

Metrics

Set properties to configure Hadoop metrics in a `hadoop-metrics.properties` or `hadoop-metrics2.properties` file.

Syntax:

```
key1=value1
key2=value2
```

For example:

```
*.sink.foo.class=org.apache.hadoop.metrics2.sink.FileSink
namenode.sink.foo.filename=/tmp/namenode-metrics.out
secondarynamenode.sink.foo.filename=/tmp/secondarynamenode-metrics.out
```

Whitelists and blacklists

Specify a list of host addresses that are allowed or disallowed from accessing a service.

Syntax:

```
host1.domain1 host2.domain2
```

Setting an Advanced Configuration Snippet

1. Click a service.
2. Click the **Configuration** tab.
3. In the Search box, type `Advanced Configuration Snippet`.
4. Choose a property that contains the string **Advanced Configuration Snippet (Safety Valve)**.
5. Specify the snippet properties. If the snippet is an XML file, you have the option to use a snippet editor (the default) or an XML text field:

   - Snippet editor

     1. Click ✚ to add a property. Enter the property name, value, and optional description. To indicate that the property value cannot be overridden by another , select the **Final** checkbox.

   - XML text field - Enter the property name, value, and optional description in as XML elements.

   ```
   <property>
     <name>name</name>
     <value>property_value</value>
     <final>final_value</final>
   </property>
   ```

   To indicate that the property value cannot be overridden, specify `<final>true</final>`.

   To switch between the editor and text field, click the **View Editor** and **View XML** links at the top right of the snippet row.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the service or role or redeploy client configurations as indicated.

Setting Advanced Configuration Snippets for a Cluster or Clusters

1. Do one of the following

   - **specific cluster**

1. On the **Home** > **Status** tab, click a cluster name.
2. Select **Configuration** > **Advanced Configuration Snippets**.

- **all clusters**

    1. Select **Configuration** > **Advanced Configuration Snippets**.

2. Specify the snippet properties. If the snippet is an XML file, you have the option to use a snippet editor (the default) or an XML text field:

   - Snippet editor

Name	Name	🗑
Value	Value	
Description	Description	
	☐ Final	
➕		

       1. Click ➕ to add a property. Enter the property name, value, and optional description. To indicate that the property value cannot be overridden by another , select the **Final** checkbox.

   - XML text field - Enter the property name, value, and optional description in as XML elements.

```
<property>
  <name>name</name>
  <value>property_value</value>
  <final>final_value</final>
</property>
```

   To indicate that the property value cannot be overridden, specify `<final>true</final>`.

   To switch between the editor and text field, click the **View Editor** and **View XML** links at the top right of the snippet row.

3. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
4. Restart the service or role or redeploy client configurations as indicated.

Stale Configurations

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

The Stale Configurations page provides differential views of changes made in a cluster. For any configuration change, the page contains entries of all affected attributes. For example, the following File entry shows the change to the file `hdfs-site.xml` when you update the property controlling how much disk space is reserved for non-HDFS use on each DataNode:

```
File: hdfs-site.xml                                                                hdfs (3)  Show
...  ...  @@ -91,9 +91,9 @@
91   91       <value>4096</value>
92   92     </property>
93   93     <property>
94   94       <name>dfs.datanode.du.reserved</name>
95    -       <value>5077964390</value>
     95 +       <value>2147483648</value>
96   96     </property>
97   97     <property>
98   98       <name>dfs.datanode.failed.volumes.tolerated</name>
99   99       <value>0</value>
```

To display the entities affected by a change, click the **Show** button at the right of the entry. The following dialog box shows that three DataNodes were affected by the disk space change:

Viewing Stale Configurations

To view stale configurations, click the ⏻ , ↻ , or ⎗ indicator next to a service on the [Cloudera Manager Admin Console Home Page](#) on page 18 or on a service status page.

Attribute Categories

The categories of attributes include:

- **Environment** - represents environment variables set for the role. For example, the following entry shows the change to the environment that occurs when you update the heap memory configuration of the SecondaryNameNode.

- **Files** - represents configuration files used by the role.
- **Process User & Group** - represents the user and group for the role. Every role type has a configuration to specify the user/group for the process. If you change a value for a user or group on any service's configuration page it will appear in the Stale Configurations page.
- **System Resources** - represents system resources allocated for the role, including ports, directories, and cgroup limits. For example, a change to the port of role instance will appear in the System Resources category.
- **Client Configs Metadata** - represents client configurations.

Filtering Stale Configurations

You filter the entries on the Stale Configurations page by selecting from one of the drop-down lists:

- **Attribute** - you can filter by an attribute category such as All Files or by a specific file such as `topology.map` or `yarn-site.xml`.
- **Service**
- **Role**

After you make a selection, both the page and the drop-down show only entries that match that selection.

To reset the view, click **Remove Filter** or select **All** *XXX*, where *XXX* is Files, Services, or Roles, from the drop-down. For example, to see all the files, select **All Files**.

Stale Configuration Actions

The Stale Configurations page displays action buttons. The action depends on what is required to bring the entire cluster's configuration up to date. If you go to the page by clicking a ⟳ (Refresh Needed) indicator, the action button will say **Restart Stale Services** if *one* of the roles listed on the page need to be restarted.

- **Refresh Stale Services** - Refreshes stale services.
- **Restart Stale Services** - Restarts stale services.
- **Restart Cloudera Management Service** - Runs the restart Cloudera Management Service action.
- **Deploy Client Configuration** - Runs the cluster deploy client configurations action.

Client Configuration Files

> ✎ **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

To allow clients to use the HBase, HDFS, Hive, MapReduce, and YARN services, Cloudera Manager creates zip archives of the configuration files containing the service properties. The zip archive is referred to as a ***client configuration file***. Each archive contains the set of configuration files needed to access the service: for example, the MapReduce client configuration file contains copies of `core-site.xml`, `hadoop-env.sh`, `hdfs-site.xml`, `log4j.properties`, and `mapred-site.xml`.

Client configuration files are generated automatically by Cloudera Manager based on the services and roles you have installed and Cloudera Manager deploys these configurations automatically when you install your cluster, add a service on a host, or add a gateway role on a host. Specifically, for each host that has a service role instance installed, and for each host that is configured as a gateway role for that service, the deploy function downloads the configuration zip file, unzips it into the appropriate configuration directory, and uses the Linux alternatives mechanism to set a given, configurable priority level. If you are installing on a system that happens to have pre-existing alternatives, then it is possible another alternative may have higher priority and will continue to be used. The alternatives priority of the Cloudera Manager client configuration is configurable under the **Gateway** scope of the **Configuration** tab for the appropriate service.

You can also manually distribute client configuration files to the clients of a service.

The main circumstance that may require a redeployment of the client configuration files is when you have modified a configuration. In this case you will typically see a message instructing you to redeploy your client configurations. The affected service(s) will also display a 📤 icon. Click the indicator to display the Stale Configurations on page 77 page.

How Client Configurations are Deployed

Client configuration files are deployed on any host that is a client for a service—that is, that has a role for the service on that host. This includes roles such as DataNodes, TaskTrackers, RegionServers and so on as well as gateway roles for the service.

If roles for multiple services are running on the same host (for example, a DataNode role and a TaskTracker role on the same host) then the client configurations for both roles are deployed on that host, with the alternatives priority determining which configuration takes precedence.

For example, suppose we have six hosts running roles as follows: host H1: HDFS-NameNode; host H2: MR-JobTracker; host H3: HBase-Master; host H4: MR-TaskTracker, HDFS-DataNode, HBase-RegionServer; host H5: MR-Gateway; host H6: HBase-Gateway. Client configuration files will be deployed on these hosts as follows: host H1: hdfs-clientconfig (only); host H2: mapreduce-clientconfig, host H3: hbase-clientconfig; host H4: hdfs-clientconfig, mapreduce-clientconfig, hbase-clientconfig; host H5: mapreduce-clientconfig; host H6: hbase-clientconfig

If the HDFS NameNode and MapReduce JobTracker were on the same host, then that host would have both hdfs-clientconfig and mapreduce-clientconfig installed.

Downloading Client Configuration Files

1. Follow the appropriate procedure according to your starting point:

Page	Procedure
Home	1. On the **Home** > **Status** tab, click ▼ to the right of the cluster name and select **View Client Configuration URLs**. A pop-up window with links to the configuration files for the services you have installed displays. 2. Click a link or save the link URL and download the file using `wget` or `curl`.
Service	1. Go to a service whose client configuration you want to download. 2. Select **Actions** > **Download Client Configuration**.

Manually Redeploying Client Configuration Files

Although Cloudera Manager will deploy client configuration files automatically in many cases, if you have modified the configurations for a service, you may need to redeploy those configuration files.

If your client configurations were deployed automatically, the command described in this section will attempt to redeploy them as appropriate.

> **Note:** If you are deploying client configurations on a host that has multiple services installed, some of the same configuration files, though with different configurations, will be installed in the `conf` directories for each service. Cloudera Manager uses the `priority` parameter in the `alternatives --install` command to ensure that the correct configuration directory is made active based on the combination of services on that host. The priority order is YARN > MapReduce > HDFS. The priority can be configured under the **Gateway** sections of the **Configuration** tab for the appropriate service.

1. On the **Home** > **Status** tab, click

▼

to the right of the cluster name and select **Deploy Client Configuration**.
2. Click **Deploy Client Configuration**.

Viewing and Reverting Configuration Changes

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See [Managing Licenses](#) on page 37 for more information.

Whenever you change and save a set of configuration settings for a service or role instance or a host, Cloudera Manager saves a revision of the previous settings and the name of the user who made the changes. You can then view past revisions of the configuration settings, and, if desired, roll back the settings to a previous state.

Viewing Configuration Changes

1. For a service, role, or host, click the **Configuration** tab.
2. Click the **History and Rollback** button. The most recent revision, currently in effect, is shown under **Current Revision**. Prior revisions are shown under **Past Revisions**.

- By default, or if you click **Show All**, a list of all revisions is shown. If you are viewing a service or role instance, all service/role group related revisions are shown. If you are viewing a host or all hosts, all host/all hosts related revisions are shown.
- To list only the configuration revisions that were done in a particular time period, use the Time Range Selector to select a time range. Then, click **Show within the Selected Time Range**.

3. Click the **Details...** link. The Revision Details dialog box displays.

Revision Details Dialog

For a service or role instance, shows the following:

- A brief message describing the context of the changes
- The date/time stamp of the change
- The user who performed the change
- The names of any role groups created
- The names of any role groups deleted

For a host instance, shows just a message, date and time stamp, and the user.

The dialog box contains two tabs:

- **Configuration Values** - displays configuration value changes, where changes are organized under the role group to which they were applied. (For example, if you changed a Service-Wide property, it will affect all role groups for that service). For each modified property, the Value column shows the new value of the property and the previous value.
- **Group Membership** - displays changes to the changed the group membership of a role instance (moved the instance from one group to another). This tab is only shown for service and role configurations.

Reverting Configuration Changes

1. Select the current or past revision to which to roll back.
2. Click the **Details...** link. The Revision Details dialog box displays.
3. Click the **Configuration Values** tab.
4. Click the **Revert Configuration Changes** button. The revert action occurs immediately. You may need to restart the service or the affected roles for the change to take effect.

> **❗ Important:** This feature can only be used to revert changes to configuration values. You cannot use this feature to:
>
> - Revert NameNode high availability. You must perform this action by explicitly disabling high availability.
> - Disable Kerberos security.
> - Revert role group actions (creating, deleting, or moving membership among groups). You must perform these actions explicitly in the Role Groups on page 212 feature.

Exporting and Importing Cloudera Manager Configuration

You can use the Cloudera Manager API to programmatically export and import a definition of all the entities in your Cloudera Manager-managed deployment—clusters, service, roles, hosts, users and so on. See the Cloudera Manager API on page 24 documentation on how to manage deployments using the /cm/deployment resource.

Cloudera Manager Configuration Properties Reference

Refer to the links below for a list of available CDH configuration properties for each version of CDH when managed by one of the following versions of Cloudera Manager:

- [Cloudera Manager 6.0.0 CDH Properties](#)
- [Cloudera Manager 5.14.x CDH Properties](#)
- [Cloudera Manager 5.13.x CDH Properties](#)
- [Cloudera Manager 5.12.x CDH Properties](#)
- [Cloudera Manager 5.11.x CDH Properties](#)
- [Cloudera Manager 5.10.x CDH Properties](#)
- [Cloudera Manager 5.9.x CDH Properties](#)
- [Cloudera Manager 5.8.x CDH Properties](#)
- [Cloudera Manager 5.7.x CDH Properties](#)
- [Cloudera Manager 5.6.x CDH Properties](#)
- [Cloudera Manager 5.5.x CDH Properties](#)
- [Cloudera Manager 5.4.x CDH Properties](#)
- [Cloudera Manager 5.3.x CDH Properties](#)
- [Cloudera Manager 5.2.x CDH Properties](#)
- [Cloudera Manager 5.1.x CDH Properties](#)
- [Cloudera Manager 5.0.x CDH Properties](#)

For information on managing configuration settings, see [Modifying Configuration Properties Using Cloudera Manager](#) on page 60 and [Viewing and Reverting Configuration Changes](#) on page 80.

Managing Clusters

Cloudera Manager can manage multiple clusters, however each cluster can only be associated with a single Cloudera Manager Server or Cloudera Manager HA pair. Once you have successfully installed your first cluster, you can add additional clusters, running the same or a different version of CDH. You can then manage each cluster and its services independently.

On the **Home** > **Status** tab you can access many cluster-wide actions by selecting

[▾]

to the right of the cluster name: add a service, start, stop, restart, deploy client configurations, enable Kerberos, and perform cluster refresh, rename, upgrade, and maintenance mode actions.

Adding and Deleting Clusters

Minimum Required Role: Full Administrator

> **Important:** As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.
>
> This can affect new installations, upgrades, adding new hosts to a cluster, adding a cluster and adding new services to a cluster.
>
> For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

Cloudera Manager can manage multiple clusters. Furthermore, the clusters do not need to run the same major version of CDH.

Adding a Cluster

Action	Procedure
New Hosts	1. On the **Home** > **Status** tab, click [☑ ▾] and select **Add Cluster**. This begins the Installation Wizard, just as if you were installing a cluster for the first time. (See Step 7: Set Up a Cluster Using the Wizard in the installation guide for detailed instructions.) 2. To find new hosts, not currently managed by Cloudera Manager, where you want to install CDH, enter the hostnames or IP addresses, and click **Search**. Cloudera Manager lists the hosts you can use to configure a new cluster. Managed hosts that already have services installed will not be selectable. 3. Click **Continue** to install the new cluster. At this point the installation continues through the wizard the same as it did when you installed your first cluster. You will be asked to select the version of CDH to install, which services you want and so on, just as previously. 4. Restart the Reports Manager role.
Managed Hosts	You may have hosts that are already "managed" but are not part of a cluster. You can have managed hosts that are not part of a cluster when you have added hosts to Cloudera Manager either through the Add Host wizard, or by manually installing the Cloudera Manager agent onto

Action	Procedure
	hosts where you have not install any other services. This will also be the case if you remove all services from a host so that it no longer is part of a cluster. 1. On the **Home** > **Status** tab, click ⬚▾ and select **Add Cluster**. This begins the Installation Wizard, just as if you were installing a cluster for the first time. (See [Step 7: Set Up a Cluster Using the Wizard](#) in the installation guide for detailed instructions.) 2. To see the list of the currently managed hosts, click the **Currently Managed Hosts** tab. This tab does not appear if you have no currently managed hosts that are not part of a cluster. 3. To perform the installation, click **Continue**. Instead of searching for hosts, this will attempt to install onto any hosts managed by Cloudera Manager that are not already part of a cluster. It will proceed with the installation wizard as for a new cluster installation. 4. Restart the Reports Manager role.

Deleting a Cluster

1. [Stop](#) the cluster.
2. On the **Home** > **Status** tab, click

   ▾

   to the right of the cluster name and select **Delete**.

Starting, Stopping, Refreshing, and Restarting a Cluster

Minimum Required Role: [Operator](#) (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

Complete the steps below to start, stop, refresh, and restart a cluster.

You can also view the following video, which shows you how to stop, start, and restart a cluster in Cloudera Manager:

[Stopping, Starting, and Restarting a Cluster in Cloudera Manager](#)

Starting a Cluster

1. On the **Home** > **Status** tab, click

   ▾

   to the right of the cluster name and select **Start**.
2. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.

   When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

> **Note:** The cluster-level Start action starts only CDH and other product services (Impala, Cloudera Search). It does not start the Cloudera Management Service. You must [start the Cloudera Management Service](#) separately if it is not already running.

Stopping a Cluster

1. On the **Home** > **Status** tab, click

   [▼]

   to the right of the cluster name and select **Stop**.
2. Click **Stop** in the confirmation screen. The **Command Details** window shows the progress of stopping services.

   When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.

   > **Note:** The cluster-level Stop action does not stop the Cloudera Management Service. You must stop the Cloudera Management Service separately.

Refreshing a Cluster

Runs a cluster refresh action to bring the configuration up to date without restarting all services. For example, certain masters (for example NameNode and ResourceManager) have some configuration files (for example, `fair-scheduler.xml`, `mapred_hosts_allow.txt`, `topology.map`) that can be refreshed. If anything changes in those files then a refresh can be used to update them in the master. Here is a summary of the operations performed in a refresh action:

✓ **Refresh Cluster**   Cluster 1   Finished   Mar 19, 2014 11:31:55 AM PDT   Mar 19, 2014 11:32:09 AM PDT

Successfully refreshed roles in the cluster.

Command Progress

Completed 4 of 4 steps.

✓ Run 1 steps in parallel
  Successfully refreshed datanode allow/exclude lists.
  Details ↗

✓ Run 1 steps in parallel
  Successfully refreshed ResourceManager.
  Details ↗

✓ Run 3 steps in parallel
  Successfully refreshed NodeManager.
  Details ↗

✓ Run 3 steps in parallel
  Refreshed Impala Daemon's Pools configuration and ACLs successfully.
  Details ↗

To refresh a cluster, in the **Home** > **Status** tab, click

[▼]

to the right of the cluster name and select **Refresh Cluster**.

Restarting a Cluster

1. On the **Home** > **Status** tab, click

   [▼]

   to the right of the cluster name and select **Restart**.
2. Click **Restart** that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose **Rolling Restart** instead to minimize cluster downtime. The **Command Details** window shows the progress of stopping services.

When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

Pausing a Cluster in AWS

If all data for a cluster is stored on EBS volumes, you can pause the cluster and stop your AWS EC2 instances during periods when the cluster will not be used. The cluster will not be available while paused and can't be used to ingest or process data, but you won't be billed by Amazon for the stopped EC2 instances. Provisioned EBS storage volumes will continue to accrue charges.

> ❗ **Important:** Pausing a cluster requires using EBS volumes for all storage, both on management and worker nodes. Data stored on ephemeral disks will be lost after EC2 instances are stopped.

Shutting Down and Starting Up the Cluster

In the shutdown and startup procedures below, some steps are performed in the AWS console and some are performed in Cloudera Manager:

- For AWS actions, use one of the following interfaces:

  - AWS console
  - AWS CLI
  - AWS API

- For cluster actions, use one of the following interfaces:

  - The Cloudera Manager web UI
  - The Cloudera API **start** and **stop** commands

Shutdown procedure

To pause the cluster, take the following steps:

1. Navigate to the Cloudera Manager web UI.
2. Stop the cluster.

    a. On the **Home** > **Status** tab, click

    ⏷

    to the right of the cluster name and select **Stop**.

    b. Click **Stop** in the confirmation screen. The **Command Details** window shows the progress of stopping services.

    When **All services successfully stopped** appears, the task is complete and you can close the **Command Details** window.

3. Stop the Cloudera Management Service.

    a. On the **Home** > **Status** tab, click

    ⏷

    to the right of the service name and select **Stop**.

    b. Click **Stop** in the next screen to confirm. When you see a **Finished** status, the service has stopped.

4. In AWS, stop all cluster EC2 instances, including the Cloudera Manager host .

Startup procedure

To restart the cluster after a pause, the steps are reversed:

1. In AWS, start all cluster EC2 instances.
2. Navigate to the Cloudera Manager UI.
3. Start the Cloudera Management Service.

   a. On the **Home** > **Status** tab, click

   ▾

   to the right of the service name and select **Start**.
   b. Click **Start** in the next screen to confirm. When you see a **Finished** status, the service has started.

4. Start the cluster.

   a. On the **Home** > **Status** tab, click

   ▾

   to the right of the cluster name and select **Start**.
   b. Click **Start** that appears in the next screen to confirm. The **Command Details** window shows the progress of starting services.

   When **All services successfully started** appears, the task is complete and you can close the **Command Details** window.

More information

For more information about stopping the Cloudera Management Service, see Stopping the Cloudera Management Service in the Cloudera Enterprise documentation.

For more information about restarting the Cloudera Management Service, see Restarting the Cloudera Management Service in the Cloudera Enterprise documentation.

For more information about starting and stopping a cluster in Cloudera Manager, see Starting, Stopping, Refreshing, and Restarting a Cluster on page 84 in the Cloudera Enterprise documentation.

For more information about stopping and starting EC2 instances, see Stop and Start Your Instance in the AWS documentation.

Considerations after Restart

Since the cluster was completely stopped before stopping the EC2 instances, the cluster should be healthy upon restart and ready for use. You should be aware of the following about the restarted cluster:

- After starting the EC2 instances, Cloudera Manager and its agents will be running but the cluster will be stopped. There will be gaps in Cloudera Manager's time-based metrics and charts.
- EC2 instances retain their internal IP address and hostname for their lifetime, so no reconfiguration of CDH is required after restart. The public IP and DNS hostnames, however, will be different. Elastic IPs can be configured to remain associated with a stopped instance at additional cost, but it isn't necessary to maintain proper cluster operation.

Renaming a Cluster

Minimum Required Role: Full Administrator

1. On the **Home** > **Status** tab, click

   ▾

   to the right of the cluster name and select **Rename Cluster**.
2. Type the new cluster name and click **Rename Cluster**.

Cluster-Wide Configuration

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

To make configuration changes that apply to an entire cluster, do one of the following to open the configuration page:

- **all clusters**

  1. Select **Configuration** and then select one of the following classes of properties:

     - Advanced Configuration Snippets
     - Databases
     - Disk Space Thresholds
     - Local Data Directories
     - Local Data Files
     - Log Directories
     - Navigator Settings
     - Non-default Values - properties whose value differs from the default value
     - Non-uniform Values - properties whose values are not uniform across the cluster or clusters
     - Port Configurations
     - Service Dependencies

     You can also select **Configuration Issues** to view a list of configuration issues for all clusters.

- **specific cluster**

  1. On the **Home** page, click a cluster name.
  2. Select **Configuration** and then select one of the classes of properties listed above.

You can also apply the following filters to limit the displayed properties:

- Enter a search term in the **Search** box to search for properties by name or description.
- Expand the **Status** filter to select options that limit the displayed properties to those with errors or warnings, properties that have been edited, properties with non-default values, or properties with overrides. Select **All** to remove any filtering by Status.
- Expand the **Scope** filter to display a list of service types. Expand a service type heading to filter on **Service-Wide** configurations for a specific service instance or select one of the default role groups listed under each service type. Select **All** to remove any filtering by Scope.
- Expand the **Category** filter to filter using a sub-grouping of properties. Select **All** to remove any filtering by Category.

Managing Services

The following sections cover the configuration and management of individual CDH and other services that have specific and unique requirements or options.

Managing the HBase Service

For information about how to perform HBase tasks such as starting and stopping HBase, writing data to HBase, and importing data to HBase, see the Managing HBase section of the Apache HBase Guide.

Managing HDFS

The section contains configuration tasks for the HDFS service. For information on configuring HDFS for high availability, see HDFS High Availability on page 421.

Data Durability
Overview

> ⚠ **Warning:** HDFS Erasure Coding is an experimental feature that is not supported.

CDH provides two options for data durability, how resilient data is to loss, when data is stored in HDFS. You can use 3x replication, which HDFS was originally built on, or Erasure Coding (EC). With 3x replication, HDFS creates two copies of data, resulting in three total instances of data. These copies are stored on separate DataNodes to guard against data loss when a node is unreachable. When the data stored on a node is lost or inaccessible, it is replicated from one of the other nodes to a new node so that there are always three copies. The number of replications is configurable, but the default is three. Cloudera recommends keeping the replication factor to at least three when you have three or more DataNodes. A lower replication factor can lead to data loss.

EC is an alternative to the 3x replication scheme. When an HDFS cluster uses EC, no additional copies of the data are generated. Instead, data is striped into blocks and encoded to generate parity blocks. If there is data missing or corrupt, HDFS uses the remaining data and parity blocks to reconstruct the missing pieces in the background. This process provides a similar level of data durability to replication but at a lower storage cost. With both data protection schemes, replication and EC, recovery happens in the background and requires no direct input from a user.

EC can be the only data protection policy in effect or it can be used in conjunction with 3x data replication in a sort of hybrid deployment. This decision should be based on the temperature of the data (how often the data is accessed) that is stored in HDFS. Additionally, EC is applied when data is written. This means that to use EC, you must replicate existing data to directories with EC set as the policy or write new data to directories with EC set as the policy. For 3x replication, there are no additional steps required.

Understanding Erasure Coding Policies

The EC policy determines how data is encoded and decoded. An EC policy is made up of the following parts: codec-number of data blocks-number of parity blocks-cell size.

- **Codec**: The erasure codec that the policy uses. It can be XOR or Reed-Solomon (RS)
- **Number of Data Blocks**: The number of data blocks per stripe. The higher this number, the more nodes that need to be read when accessing data because HDFS attempts to distribute the blocks evenly across DataNodes.
- **Number of Parity Blocks**: The number of parity blocks per stripe. Even if a file does not use up all the data blocks available to it, the number of parity blocks will always be the total number listed in the policy.
- **Cell Size**: The size of one basic unit of striped data.

For example, a RS-6-3-1024k policy has the following attributes:

- **Codec**: Reed-Solomon
- **Number of Data Blocks**: 6
- **Number of Parity Blocks**: 3
- **Cell Size**: 1024k

The sum of the number of data blocks and parity blocks is the data stripe width. When you make hardware plans for your cluster, the number of racks should at least equal the stripe width in order for the data to be resistant to rack failures. Ideally, the number of racks exceeds the data stripe width to account for downtime and outages. If there are fewer racks than the data stripe width, HDFS spreads data blocks across multiple nodes to maintain fault tolerance at the node level. When distributing blocks to racks, HDFS attempts to distribute the blocks evenly across all racks. Because of this behavior, Cloudera recommends setting up each rack with a similar number of DataNodes. Otherwise, racks with fewer DataNodes may be filled up faster than racks with more DataNodes.

To achieve node-level fault tolerance, the number of nodes needs to equal the data stripe width. For example, in order for a RS-6-3-1024k policy to be node failure tolerant, you need at least 9 nodes. For rack-level fault tolerance, spread the 9 nodes evenly between across three racks. The data and parity blocks, when distributed evenly, lead to the following placement on the racks:

- Rack 1: Three blocks
- Rack 2: Three blocks

- Rack 3: Three blocks

Cloudera recommends at least nine racks though, which leads to one data or parity block on each rack.

A policy with a wide data-stripe width like RS-6-3-1024k comes with a tradeoff though. Data must be read from 6 blocks, increasing the read time. Therefore, the larger the cluster and colder the data, the more appropriate it is to use EC policies with large data stripe widths.

Comparing Replication and Erasure Coding

Consider the following factors when you examine which data protection scheme to use:

Data Temperature

Data temperature refers to how often data is accessed. EC works best with cold data that is accessed and modified infrequently. Replication is more suitable for hot data, data that is accessed and modified frequently.

I/O Cost

EC has higher I/O costs than replication for the following reasons:

- EC spreads data across nodes and racks, which means reading and writing data comes at a higher cost.
- A parity block is generated when data is written, thus impacting write speed.
- If data is missing or corrupt, a DataNode reads the remaining data and parity blocks in order to reconstruct the data. This process requires CPU and network resources.

Cloudera recommends at least a 10GB network connection if you want to use EC.

Storage Cost

EC has a lower storage overhead than replication because multiple copies of data are not maintained. Instead, a number of parity blocks are generated based on the EC policy. For the same amount of data, EC will store fewer blocks than 3x replication in most cases. For example with a RS (10,4) EC, HDFS stores four parity blocks for each set of 10 data blocks. With replication, HDFS stores 12 replica blocks for every six data blocks, the original block and three replicas. The case where 3x replication requires fewer blocks is when dat is stored in small files.

File Size

Erasure coding works best with larger files. The total number of blocks is determined by data blocks + parity blocks, which is the data stripe width discussed earlier.

With RS (6,3), each block group can hold up to (128MB * 6) = 768 MB of data. 128MB is the default block size . Inside each block group, there will be 9 total blocks, 6 data blocks, each holding up to 128MB, and 3 parity blocks. For the best utilization of blocks, the erasure coded file size should be close to 768 MB. For a chunk of data less than the block size, HDFS uses one data block, but the full number of parity blocks are still required. This leads to a situation where erasure coded files will generate more blocks than 3x replication because of the parity blocks required.

Enabling Erasure Coding
Before You Begin

> ⚠️ **Warning:** HDFS Erasure Coding is an experimental feature that is not supported.

Before you enable Erasure Coding (EC), perform the following tasks:

- Verify that the clusters run CDH 6.0 or higher.
- Determine which EC policy you want to use.
- Determine if you want to use EC for existing data or new data

Enabling Erasure Coding

Enable EC using the Cloudera Manager Admin Console:

1. Select **Clusters** and choose the HDFS cluster you want to enable EC for.
2. Navigate to the **Configuration** tab and select the **Erasure Coding** category.

3. Configure the EC properties:

- **DataNode Striped Read Timeout**: DataNode reconstruction striped read timeout in milliseconds.
- **DataNode Striped Read Threads**: Number of threads used by the DataNode to read striped blocks during background reconstruction work.
- **Erasure Coding Reconstruction Weight**: Relative weight of resources used by EC background recovery tasks, which require reading multiple blocks, 6 in the case of RS-6-3-1024k, compared to replicated block recovery, which only requires reading a single replica. Higher values result in fewer reconstruction tasks being able to run concurrently. Blocks required to be read to complete recovery are multiplied by this weight to determine the total weight of the recovery task. These units of weight count against the limit set in the `dfs.namenode.replication.max-streams` property.
- **Default Policy when Setting Erasure Coding**: The erasure coding policy used when enabling erasure coding for a directory without specifying a policy.
- **Erasure Coding Enabled**: Allows erasure coding policies to be enabled and set for directories. Note that erasure coding is currently not supported and is experimental only.

4. Optionally, you can view the supported EC policies with the following command:

```
hdfs ec -listPolicies
```

5. Enable a supported EC policy from step 4:

```
hdfs ec -enablePolicy <policy>
```

6. Set the EC policy for a directory with the following command:

```
hdfs ec -setPolicy -path <directory> [-policy <policyName>]
```

- `path`. Required. Specify the HDFS directory you want to apply the EC policy to.
- `policy`. Optional. The EC policy you want to use for the directory you specified. If you do not provide this parameter, the EC policy you specified in step 3 for the **Default Policy when Setting Erasure Coding** setting is used.

This command applies the EC policy to data written to the directory after the command is run. It does not apply EC policies to existing data. See Using Erasure Coding for Existing Data on page 91 for information about how to use EC with existing data.

Using Erasure Coding for Existing Data

To use EC with existing data, that data must be copied into a directory that has EC enabled. Use the `distcp` tool or Cloudera Manager's Backup and Disaster Recovery (BDR).

Using Erasure Coding for New Data

To use EC with new data, set the destination for the data to a directory with EC enabled. No action beyond that is required. When data is written to the directory, it will be erasure coded based on the policy you set.

NameNodes

NameNodes maintain the namespace tree for HDFS and a mapping of file blocks to DataNodes where the data is stored. A simple HDFS cluster can have only one primary NameNode, supported by a secondary NameNode that periodically compresses the NameNode edits log file that contains a list of HDFS metadata modifications. This reduces the amount of disk space consumed by the log file on the NameNode, which also reduces the restart time for the primary NameNode. A high availability cluster contains two NameNodes: active and standby.

Formatting the NameNode and Creating the /tmp Directory

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

When you add an HDFS service, the wizard automatically formats the NameNode and creates the `/tmp` directory on HDFS. If you quit the wizard or it does not finish, you can format the NameNode and create the `/tmp` directory outside the wizard by doing these steps:

1. Stop the HDFS service if it is running. See <u>Starting, Stopping, and Restarting Services</u> on page 197.
2. Click the **Instances** tab.
3. Click the NameNode role instance.
4. Select **Actions** > **Format**.
5. Start the HDFS service.
6. Select **Actions** > **Create /tmp Directory**.

Backing Up and Restoring HDFS Metadata

Backing Up HDFS Metadata Using Cloudera Manager

HDFS metadata backups can be used to restore a NameNode when both NameNode roles have failed. In addition, Cloudera recommends backing up HDFS metadata before a major upgrade.

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

This backup method requires you to shut down the cluster.

1. Note the active NameNode.
2. Stop the cluster. It is particularly important that the NameNode role process is not running so that you can make a consistent backup.
3. Go to the HDFS service.
4. Click the **Configuration** tab.
5. In the Search field, search for "NameNode Data Directories" and note the value.
6. On the active NameNode host, back up the directory listed in the NameNode Data Directories property. If more than one is listed, make a backup of one directory, because each directory is a complete copy. For example, if the NameNode data directory is `/data/dfs/nn`, do the following as root:

```
# cd /data/dfs/nn
# tar -cvf /root/nn_backup_data.tar .
```

You should see output like this:

```
/dfs/nn/current

./
./VERSION
./edits_0000000000000000001-0000000000000008777
./edits_0000000000000008778-0000000000000009337
./edits_0000000000000009338-0000000000000009897
./edits_0000000000000009898-0000000000000010463
./edits_0000000000000010464-0000000000000011023
<snip>
./edits_0000000000000063396-0000000000000063958
./edits_0000000000000063959-0000000000000064522
./edits_0000000000000064523-0000000000000065091
./edits_0000000000000065092-0000000000000065648
./edits_inprogress_0000000000000065649
./fsimage_0000000000000065091
./fsimage_0000000000000065091.md5
./fsimage_0000000000000065648
./fsimage_0000000000000065648.md5
./seen_txid
```

If a file with the extension *lock* exists in the NameNode data directory, the NameNode most likely is still running. Repeat the steps, beginning with shutting down the NameNode role.

Restoring HDFS Metadata From a Backup Using Cloudera Manager

The following process assumes a scenario where both NameNode hosts have failed and you must restore from a backup.

1. Remove the NameNode, JournalNode, and Failover Controller roles from the HDFS service.
2. Add the host on which the NameNode role will run.
3. Create the NameNode data directory, ensuring that the permissions, ownership, and group are set correctly.
4. Copy the backed up files to the NameNode data directory.
5. Add the NameNode role to the host.
6. Add the Secondary NameNode role to another host.
7. Enable high availability. If not all roles are started after the wizard completes, restart the HDFS service. Upon startup, the NameNode reads the fsimage file and loads it into memory. If the JournalNodes are up and running and there are edit files present, any edits newer than the fsimage are applied.

Moving NameNode Roles

This section describes two procedures for moving NameNode roles. Both procedures require cluster downtime. If highly availability is enabled for the NameNode, you can use a Cloudera Manager wizard to automate the migration process. Otherwise you must manually delete and add the NameNode role to a new host.

After moving a NameNode, if you have a Hive or Impala service, perform the steps in NameNode Post-Migration Steps on page 95.

Moving Highly Available NameNode, Failover Controller, and JournalNode Roles Using the Migrate Roles Wizard

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

The Migrate Roles wizard allows you to move roles of a highly available HDFS service from one host to another. You can use it to move NameNode, JournalNode, and Failover Controller roles.

Requirements and Limitations

- Nameservice federation (multiple namespaces) is *not supported*.
- This procedure requires cluster downtime. The services discussed in this list must be running for the migration to complete.
- The configuration of HDFS and services that depend on it must be valid.
- The source and destination hosts must be commissioned and healthy.
- The NameNode must be highly available using quorum-based storage.
- HDFS automatic failover must be enabled, and the cluster must have a running ZooKeeper service.
- If a Hue service is present in the cluster, its HDFS Web Interface Role property must refer to an HttpFS role, not to a NameNode role.
- A majority of configured JournalNode roles must be running.
- The Failover Controller role that is not located on the source host must be running.

Before You Begin
Do the following before you run the wizard:

- On hosts running active and standby NameNodes, back up the data directories.
- On hosts running JournalNodes, back up the JournalNode edits directory.
- If the source host is not functioning properly, or is not reliably reachable, decommission the host.
- If CDH and HDFS metadata was recently upgraded, and the metadata upgrade was not finalized, finalize the metadata upgrade.

Running the Migrate Roles Wizard

1. If the host to which you want to move the NameNode is not in the cluster, follow the instructions in Adding a Host to the Cluster on page 174 to add the host.
2. Go to the HDFS service.
3. Click the **Instances** tab.
4. Click the **Migrate Roles** button.

5. Click the **Source Host** text field and specify the host running the roles to migrate. In the Search field optionally enter hostnames to filter the list of hosts and click **Search**.

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   - IP addresses
   - Rack name

   Select the checkboxes next to the desired host. The list of available roles to migrate displays. Clear any roles you do not want to migrate. When migrating a NameNode, the co-located Failover Controller must be migrated as well.

6. Click the **Destination Host** text field and specify the host to which the roles will be migrated. On destination hosts, indicate whether to delete data in the NameNode data directories and JournalNode edits directory. If you choose not to delete data and such role data exists, the Migrate Roles command will not complete successfully.

7. Acknowledge that the migration process incurs service downtime by selecting the **Yes, I am ready to restart the cluster now** checkbox.

8. Click **Continue**. The Command Progress screen displays listing each step in the migration process.

9. When the migration completes, click **Finish**.

Moving a NameNode to a Different Host Using Cloudera Manager

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

> **Note:** This procedure requires cluster downtime.

1. If the host to which you want to move the NameNode is not in the cluster, follow the instructions in [Adding a Host to the Cluster](#) on page 174 to add the host.
2. [Stop all cluster services](#).
3. Make a backup of the `dfs.name.dir` directories on the existing NameNode host. Make sure you back up the `fsimage` and `edits` files. They should be the same across all of the directories specified by the `dfs.name.dir` property.
4. Copy the files you backed up from `dfs.name.dir` directories on the old NameNode host to the host where you want to run the NameNode.
5. Go to the HDFS service.
6. Click the **Instances** tab.
7. Select the checkbox next to the NameNode role instance and then click the **Delete** button. Click **Delete** again to confirm.
8. In the **Review configuration changes** page that appears, click **Skip**.
9. Click **Add Role Instances** to add a NameNode role instance.
10. Select the host where you want to run the NameNode and then click **Continue**.
11. Specify the location of the `dfs.name.dir` directories where you copied the data on the new host, and then click **Accept Changes**.
12. [Start cluster services](#). After the HDFS service has started, Cloudera Manager distributes the new configuration files to the DataNodes, which will be configured with the IP address of the new NameNode host.

NameNode Post-Migration Steps

After moving a NameNode, if you have a Hive or Impala service, perform the following steps:

1. Go to the Hive service.
2. Stop the Hive service.
3. Select **Actions** > **Update Hive Metastore NameNodes**.
4. If you have an Impala service, restart the Impala service or run an `INVALIDATE METADATA` query.

Sizing NameNode Heap Memory

Each workload has a unique byte-distribution profile. Some workloads can use the default JVM settings for heap memory and garbage collection, but others require tuning. This topic provides guidance on sizing your NameNode JVM if the dynamic heap settings cause a bottleneck.

All Hadoop processes run on a Java Virtual Machine (JVM). The number of JVMs depend on your deployment mode:

- **Local** (or standalone) mode - There are no daemons and everything runs on a single JVM.
- **Pseudo-distributed** mode - Each daemon (such as the NameNode daemon) runs on its own JVM on a single host.
- **Distributed** mode - Each daemon runs on its own JVM across a cluster of hosts.

The legacy NameNode configuration is one active (and primary) NameNode for the entire namespace and one Secondary NameNode for checkpoints (but not failover). The recommended high-availability configuration replaces the Secondary NameNode with a Standby NameNode that prevents a single point of failure. Each NameNode uses its own JVM.

Environment Variables

HADOOP_HEAPSIZE sets the JVM heap size for *all* Hadoop project servers such as HDFS, YARN, and MapReduce. *HADOOP_HEAPSIZE* is an integer passed to the JVM as the maximum memory (Xmx) argument. For example:

```
HADOOP_HEAPSIZE=1024
```

HADOOP_NAMENODE_OPTS is specific to the NameNode and sets all JVM flags, which must be specified. *HADOOP_NAMENODE_OPTS* overrides the *HADOOP_HEAPSIZE* Xmx value for the NameNode. For example:

```
HADOOP_NAMENODE_OPTS=-Xms1024m -Xmx1024m -XX:+UseParNewGC -XX:+UseConcMarkSweepGC
-XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled
-XX:+PrintTenuringDistribution -XX:OnOutOfMemoryError={{AGENT_COMMON_DIR}}/killparent.sh
```

Both *HADOOP_NAMENODE_OPTS* and *HADOOP_HEAPSIZE* are stored in `/etc/hadoop/conf/hadoop-env.sh`.

Monitoring Heap Memory Usage

You can monitor your heap memory usage several ways:

- **Cloudera Manager:** Look at the NameNode chart for heap memory usage. If you need to build the chart from scratch, run:

```
select jvm_max_memory_mb, jvm_heap_used_mb where roleType="NameNode"
```

- **NameNode Web UI:** Scroll down to the Summary and look for "Heap Memory used."
- **Command line:** Generate a heap dump.

Files and Blocks

In HDFS, data and metadata are decoupled. Data files are split into block files that are stored, and replicated, on DataNodes across the cluster. The filesystem namespace tree and associated metadata are stored on the NameNode.

Namespace objects are file inodes and blocks that point to block files on the DataNodes. These namespace objects are stored as a file system image (fsimage) in the NameNode's memory and also persist locally. Updates to the metadata are written to an edit log. When the NameNode starts, or when a checkpoint is taken, the edits are applied, the log is cleared, and a new fsimage is created.

> **Important:** The NameNode keeps the entire namespace image in memory. The Secondary NameNode, on its own JVM, does the same when creating an image checkpoint.

On average, each file consumes 1.5 blocks of storage. That is, the average file is split into two block files—one that consumes the entire allocated block size and a second that consumes half of that. On the NameNode, this same average file requires three namespace objects—one file inode and two blocks.

Disk Space versus Namespace

The CDH default block size (*dfs.blocksize*) is set to 128 MB. Each namespace object on the NameNode consumes approximately 150 bytes.

On DataNodes, data files are measured by disk space consumed—the actual data length—and not necessarily the full block size. For example, a file that is 192 MB consumes 192 MB of disk space and *not* some integral multiple of the block size. Using the default block size of 128 MB, a file of 192 MB is split into two block files, one 128 MB file and one 64 MB file. On the NameNode, namespace objects are measured by the number of files and blocks. The same 192 MB file is represented by three namespace objects (1 file inode + 2 blocks) and consumes approximately 450 bytes of memory.

Large files split into fewer blocks generally consume less memory than small files that generate many blocks. One data file of 128 MB is represented by two namespace objects on the NameNode (1 file inode + 1 block) and consumes approximately 300 bytes of memory. By contrast, 128 files of 1 MB each are represented by 256 namespace objects (128 file inodes + 128 blocks) and consume approximately 38,400 bytes. The optimal split size, then, is some integral multiple of the block size, for memory management as well as data locality optimization.

By default, Cloudera Manager allocates a maximum heap space of 1 GB for every million blocks (but never less than 1 GB). How much memory you actually need depends on your workload, especially on the number of files, directories, and blocks generated in each namespace. If all of your files are split at the block size, you could allocate 1 GB for every million **files**. But given the historical average of 1.5 blocks per file (2 block objects), a more conservative estimate is 1 GB of memory for every million **blocks**.

> **Important:** Cloudera recommends 1 GB of NameNode heap space per million blocks to account for the namespace objects, necessary bookkeeping data structures, and the remote procedure call (RPC) workload. In practice, your heap requirements will likely be less than this conservative estimate.

Replication

The default block replication factor (*dfs.replication*) is three. **Replication affects disk space but not memory consumption.** Replication changes the amount of storage required for each block but not the number of blocks. If one block file on a DataNode, represented by one block on the NameNode, is replicated three times, the number of block files is tripled but not the number of blocks that represent them.

With replication off, one file of 192 MB consumes 192 MB of disk space and approximately 450 bytes of memory. If you have one million of these files, or 192 TB of data, you need 192 TB of disk space and, *without considering the RPC workload*, 450 MB of memory: (1 million inodes + 2 million blocks) * 150 bytes. With default replication on, you need 576 TB of disk space: (192 TB * 3) but the memory usage stay the same, 450 MB. When you account for bookkeeping and RPCs, and follow the recommendation of 1 GB of heap memory for every million blocks, a much safer estimate for this scenario is 2 GB of memory (with or without replication).

Examples

Example 1: Estimating NameNode Heap Memory Used

Alice, Bob, and Carl each have 1 GB (1024 MB) of data on disk, but sliced into differently sized files. Alice and Bob have files that are some integral of the block size and require the least memory. Carl does not and fills the heap with unnecessary namespace objects.

Alice: 1 x 1024 MB file

- 1 file inode

- 8 blocks (1024 MB / 128 MB)

Total = 9 objects * 150 bytes = **1,350 bytes** of heap memory

Bob: 8 x 128 MB files

- 8 file inodes
- 8 blocks

Total = 16 objects * 150 bytes = **2,400 bytes** of heap memory

Carl: 1,024 x 1 MB files

- 1,024 file inodes
- 1,024 blocks

Total = 2,048 objects * 150 bytes = **307,200 bytes** of heap memory

Example 2: Estimating NameNode Heap Memory Needed

In this example, memory is estimated by considering the capacity of a cluster. Values are rounded. Both clusters physically store 4800 TB, or approximately 36 million *block files* (at the default block size). Replication determines how many namespace blocks represent these block files.

Cluster A: 200 hosts of 24 TB each = 4800 TB.

- Blocksize=128 MB, **Replication=1**
- Cluster capacity in MB: 200 * 24,000,000 MB = 4,800,000,000 MB (4800 TB)
- Disk space needed per block: 128 MB per block * 1 = **128 MB** storage per block
- Cluster capacity in blocks: 4,800,000,000 MB / 128 MB = **36,000,000 blocks**

At capacity, with the recommended allocation of 1 GB of memory per million blocks, Cluster A needs 36 GB of maximum heap space.

Cluster B: 200 hosts of 24 TB each = 4800 TB.

- Blocksize=128 MB, **Replication=3**
- Cluster capacity in MB: 200 * 24,000,000 MB = 4,800,000,000 MB (4800 TB)
- Disk space needed per block: 128 MB per block * 3 = **384 MB** storage per block
- Cluster capacity in blocks: 4,800,000,000 MB / 384 MB = **12,000,000 blocks**

At capacity, with the recommended allocation of 1 GB of memory per million blocks, Cluster B needs 12 GB of maximum heap space.

Both Cluster A and Cluster B store the same number of *block files*. In Cluster A, however, each block file is unique and represented by one block on the NameNode; in Cluster B, only one-third are unique and two-thirds are replicas.

Backing Up and Restoring NameNode Metadata

This topic describes the steps for backing up and restoring NameNode metadata.

Backing Up NameNode Metadata

This section describes how to back up NameNode metadata.

1. Make a single backup of the VERSION file. This does not need to be backed up regularly as it does not change, but it is important since it contains the clusterID, along with other details.
2. Use the following command to back up the NameNode metadata. It automatically determines the active NameNode, retrieves the current fsimage, and places it in the defined backup_dir.

```
$ hdfs dfsadmin -fetchImage backup_dir
```

On startup, the NameNode process reads the `fsimage` file and commits it to memory. If the JournalNodes are up and running, and there are edit files present, any edits newer than the `fsimage` are also applied. If the JournalNodes are unavailable, it is possible to lose any data transferred in the interim.

Restoring NameNode Metadata

This section describes how to restore NameNode metadata. If both the NameNode and the secondary NameNode were to suddenly go offline, you can restore the NameNode by doing the following:

1. Add a new host to your Hadoop cluster.
2. Add the NameNode role to the host. Make sure it has the same hostname as the original NameNode.
3. Create a directory path for the NameNode `name.dir` (for example, `/dfs/nn/current`), ensuring that the permissions are set correctly.
4. Copy the `VERSION` and latest `fsimage` file to the `/dfs/nn/current` directory.
5. Run the following command to create the md5 file for the fsimage.

```
$ md5sum fsimage > fsimage.md5
```

6. Start the NameNode process.

DataNodes

DataNodes store data in a Hadoop cluster and is the name of the daemon that manages the data. File data is replicated on multiple DataNodes for reliability and so that localized computation can be executed near the data. Within a cluster, DataNodes should be uniform. If they are not uniform, issues can occur. For example, DataNodes with less memory fill up more quickly than DataNodes with more memory, which can result in job failures.

> **Important:** The default replication factor for HDFS is three. That is, three copies of data are maintained at all times. Cloudera recommends that you do not configure a lower replication factor when you have at least three DataNodes. A lower replication factor may lead to data loss.

How NameNode Manages Blocks on a Failed DataNode

A DataNode is considered dead after a set period without any heartbeats (10.5 minutes by default). When this happens, the NameNode performs the following actions to maintain the configured replication factor (3x replication by default):

1. The NameNode determines which blocks were on the failed DataNode.
2. The NameNode locates other DataNodes with copies of these blocks.
3. The DataNodes with block copies are instructed to copy those blocks to other DataNodes to maintain the configured replication factor.

Keep the following in mind when working with dead DataNodes:

- If the DataNode failed due to a disk failure, follow the procedure in Replacing a Disk on a DataNode Host on page 98 or Performing Disk Hot Swap for DataNodes on page 101 to bring a repaired DataNode back online. If a DataNode failed to heartbeat for other reasons, they need to be recommissioned to be added back to the cluster. For more information, see Recommissioning Hosts on page 183
- If a DataNode rejoins the cluster, there is a possibility for surplus replicas of blocks that were on that DataNode. The NameNode will randomly remove excess replicas adhering to Rack-Awareness policies.

Replacing a Disk on a DataNode Host

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

For CDH 5.3 and higher, see Performing Disk Hot Swap for DataNodes on page 101.

If one of your DataNode hosts experiences a disk failure, follow this process to replace the disk:

1. Stop managed services.
2. Decommission the DataNode role instance.
3. Replace the failed disk.

4. Recommission the DataNode role instance.
5. Run the HDFS `fsck` utility to validate the health of HDFS. The utility normally reports over-replicated blocks immediately after a DataNode is reintroduced to the cluster, which is automatically corrected over time.
6. Start managed services.

Removing a DataNode

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

1. The number of DataNodes in your cluster must be greater than or equal to the replication factor you have configured for HDFS. (This value is typically 3.) In order to satisfy this requirement, add the DataNode roles on other hosts as required and start the role instances *before removing any DataNodes* .
2. Ensure the DataNode that is to be removed is running
3. Decommission the DataNode role. When asked to select the role instance to decommission, select the DataNode role instance.
4. The decommissioning process moves the data blocks to the other available DataNodes.

> **⊘ Important:** There must be at least as many DataNodes running as the replication factor or the decommissioning process will not complete.

5. Once decommissioning is completed, stop the DataNode role. When asked to select the role instance to stop, select the DataNode role instance.
6. Verify that the integrity of the HDFS service:

   a. Run the following command to identify any problems in the HDFS file system:

```
hdfs fsck /
```

   b. Fix any errors reported by the `fsck` command. If required, create a Cloudera support case.

7. After all errors are resolved:

   a. Remove the DataNode role.
   b. Manually remove the DataNode data directories. You can determine the location of these directories by examining the **DataNode Data Directory** property in the HDFS configuration. In Cloudera Manager, go to the HDFS service, select the **Configuration** tab and search for the property.

Configuring Storage Directories for DataNodes

Adding and Removing Storage Directories Using Cloudera Manager

Adding Storage Directories

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **DataNode**.
4. Add the new storage directory to the **DataNode Data Directory** property. To specify the storage type for HDFS heterogeneous storage, add the storage type, surrounded by brackets, at the front of the path. For example: `[SSD]/data/example_dir/`

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart the DataNode.

> **Important:** You must restart the DataNodes for heterogeneous storage configuration changes to take effect.

Removing Storage Directories

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. Stop the cluster.
2. Go to the HDFS service.
3. Click the **Configuration** tab.
4. Select **Scope** > **DataNode**.
5. Remove the current directories and add new ones to the **DataNode Data Directory** property.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Copy the contents under the old directory to the new directory.
8. Start the cluster.

Configuring Storage Balancing for DataNodes

You can configure HDFS to distribute writes on each DataNode in a manner that balances out available storage among that DataNode's disk volumes.

By default a DataNode writes new block replicas to disk volumes solely on a round-robin basis. You can configure a volume-choosing policy that causes the DataNode to take into account how much space is available on each volume when deciding where to place a new replica.

You can configure

- how much DataNode volumes are allowed to differ in terms of bytes of free disk space before they are considered imbalanced, *and*
- what percentage of new block allocations will be sent to volumes with more available disk space than others.

Configuring Storage Balancing for DataNodes Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **DataNode**.
4. Select **Category** > **Advanced**.
5. Configure the following properties (you can use the Search box to locate the properties):

Property	Value	Description
`dfs.datanode.fsdataset.volume.choosing.policy`	`org.apache.hadoop.hdfs.server.datanode.fsdataset.AvailableSpaceVolumeChoosingPolicy`	Enables storage balancing among the DataNode's volumes.
`dfs.datanode.available-space-volume-choosing-policy.balanced-space-threshold`	10737418240 (default)	The amount by which volumes are allowed to differ from each other in terms of bytes of free disk space before they are considered imbalanced. The default is 10737418240 (10 GB). If the free space on each volume is within this range of the other volumes, the volumes will be

Property	Value	Description
		considered balanced and block assignments will be done on a pure round-robin basis.
`dfs.datanode. available-space- volume-choosing- policy.balanced- space-preference- fraction`	0.75 (default)	What proportion of new block allocations will be sent to volumes with more available disk space than others. The allowable range is 0.0-1.0, but set it in the range 0.5 - 1.0 (that is, 50-100%), since there should be no reason to prefer that volumes with less available disk space receive more block allocations.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the role.

Performing Disk Hot Swap for DataNodes

This section describes how to replace HDFS disks without shutting down a DataNode. This is referred to as ***hot swap***.

> ⚠️ **Warning: Requirements and Limitations**
>
> - Hot swap can only add disks with empty data directories.
> - Removing a disk does not move the data off the disk, which could potentially result in data loss.
> - Do not perform hot swap on multiple hosts at the same time.

Performing Disk Hot Swap for DataNodes Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. Configure data directories to remove the disk you are swapping out:

   a. Go to the HDFS service.
   b. Click the **Instances** tab.
   c. In the **Role Type** column, click on the affected DataNode.
   d. Click the **Configuration** tab.
   e. Select **Scope** > **DataNode**.
   f. Select **Category** > **Main**.
   g. Change the value of the **DataNode Data Directory** property to remove the directories that are mount points for the disk you are removing.

   > ⚠️ **Warning:** Change the value of this property only for the specific DataNode instance where you are planning to hot swap the disk. *Do not* edit the role group value for this property. Doing so will cause data loss.

2. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
3. Refresh the affected DataNode. Select **Actions** > **Refresh DataNode configuration**.
4. Remove the old disk and add the replacement disk.
5. Change the value of the **DataNode Data Directory** property to add back the directories that are mount points for the disk you added.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Refresh the affected DataNode. Select **Actions** > **Refresh DataNode configuration**.
8. Run the `hdfs fsck /` command to validate the health of HDFS.

JournalNodes

High-availability clusters use JournalNodes to synchronize active and standby NameNodes. The active NameNode writes to each JournalNode with changes, or "edits," to HDFS namespace metadata. During failover, the standby NameNode applies all edits from the JournalNodes before promoting itself to the active state.

Moving the JournalNode Edits Directory
Moving the JournalNode Edits Directory for an Role Instance Using Cloudera Manager

To change the location of the edits directory for one JournalNode instance:

1. Reconfigure the **JournalNode Edits Directory**.

   a. Go to the **HDFS** service in Cloudera Manager.
   b. Click **JournalNode** under **Status Summary**.
   c. Click the **JournalNode** link for the instance you are changing.
   d. Click the **Configuration** tab.
   e. Set `dfs.journalnode.edits.dir` to the path of the new `jn` directory.
   f. Click Save Changes.

2. Move the location of the JournalNode (`jn`) directory at the command line:

   a. Connect to host of the JournalNode.
   b. Copy the JournalNode (`jn`) directory to its new location with the `-a` option to preserve permissions:

```
cp -a /<old_path_to_jn_dir>/jn /<new_path_to_jn_dir>/jn
```

   c. Rename the old `jn` directory to avoid confusion:

```
mv /<old_path_to_jn_dir>/jn /<old_path_to_jn_dir>/jn_to_delete
```

3. Redeploy the HDFS client configuration:

   a. Go to the **HDFS** service.
   b. Select **Actions** > **Deploy Client Configuration**.

4. Perform a [Rolling Restart](#) on page 198 for HDFS by selecting **Actions** > **Rolling Restart**. Use the default settings.
5. From the command line, delete the old `jn_to_delete` directory.

Moving the JournalNode Edits Directory for a Role Group Using Cloudera Manager

To change the location of the edits directory for each JournalNode in the JournalNode Default Group:

1. Stop all services on the cluster in Cloudera Manager:

   a. Go to the **Cluster**.
   b. Select **Actions** > **Stop**.

2. Find the list of JournalNode hosts:

   a. Go to the **HDFS** service.
   b. Click **JournalNode** under **Status Summary**.

3. Move the location of each JournalNode (`jn`) directory at the command line:

   a. Connect to each host with a JournalNode.
   b. Per host, copy the JournalNode (`jn`) directory to its new location with the `-a` option to preserve permissions:

```
cp -a /<old_path_to_jn_dir>/jn /<new_path_to_jn_dir>/jn
```

    c. Per host, rename the old `jn` directory to avoid confusion:

```
mv /<old_path_to_jn_dir>/jn /<old_path_to_jn_dir>/jn_to_delete
```

4. Reconfigure the **JournalNode Default Group**:

    a. Go to the **HDFS** service.

    b. Click the **Configuration** tab.

    c. Click **JournalNode** under **Scope**.

    d. Set `dfs.journalnode.edits.dir` to the path of the new `jn` directory for all JournalNodes in the group.

    e. Click **Save Changes**.

5. Redeploy the client configuration for the cluster:

    a. Go to the **Cluster**.

    b. Select **Actions** > **Deploy Client Configuration**.

6. Start all services on the cluster by selecting **Actions** > **Start**.

7. Delete the old `jn_to_delete` directories from the command line.

Moving JournalNodes Across Hosts

To move JournalNodes to a new host, see Moving Highly Available NameNode, Failover Controller, and JournalNode Roles Using the Migrate Roles Wizard on page 93.

Configuring Short-Circuit Reads

So-called "short-circuit" reads bypass the DataNode, allowing a client to read the file directly, as long as the client is co-located with the data. Short-circuit reads provide a substantial performance boost to many applications and help improve HBase random read profile and Impala performance.

Short-circuit reads require `libhadoop.so` (the Hadoop Native Library) to be accessible to both the server and the client. You must install from an `.rpm`, `.deb`, or parcel to use short-circuit local reads.

Configuring Short-Circuit Reads Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

> **Note:** Short-circuit reads are enabled by default in Cloudera Manager.

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Gateway or HDFS (Service-Wide)**.
4. Select **Category** > **Performance**.
5. Locate the **Enable HDFS Short Circuit Read** property or search for it by typing its name in the Search box. Check the box to enable it.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Configuring HDFS Trash

The Hadoop trash feature helps prevent accidental deletion of files and directories. When you delete a file in HDFS, the file is not immediately expelled from HDFS. Deleted files are first moved to the `/user/<username>/.Trash/Current` directory, with their original filesystem path being preserved. After a user-configurable period of time (`fs.trash.interval`), a process known as trash checkpointing renames the `Current` directory to the current timestamp, that is, `/user/<username>/.Trash/<timestamp>`. The checkpointing process

also checks the rest of the `.Trash` directory for any existing timestamp directories and removes them from HDFS permanently. You can restore files and directories in the trash simply by moving them to a location outside the `.Trash` directory.

> **Important:**
>
> - The trash feature is enabled by default. Cloudera recommends that you enable it on all production clusters.
> - The trash feature works by default only for files and directories deleted using the Hadoop shell. Files or directories deleted programmatically using other interfaces (WebHDFS or the Java APIs, for example) are not moved to trash, even if trash is enabled, unless the program has implemented a call to the trash functionality.
>
>   Users can bypass trash when deleting files using the shell by specifying the `-skipTrash` option to the `hadoop fs -rm -r` command. This can be useful when it is necessary to delete files that are too large for the user's quota.

Trash Behavior with HDFS Transparent Encryption Enabled

Starting with CDH 5.7.1, you can delete files or directories that are part of an HDFS encryption zone. As is evident from the procedure described above, moving and renaming files or directories is an important part of trash handling in HDFS. However, currently HDFS transparent encryption only supports renames *within* an encryption zone. To accommodate this, HDFS creates a local `.Trash` directory every time a new encryption zone is created. For example, when you create an encryption zone, `/enc_zone`, HDFS will also create the `/enc_zone/.Trash/` sub-directory. Files deleted from `enc_zone` are moved to `/enc_zone/.Trash/<username>/Current/`. After the checkpoint, the `Current` directory is renamed to the current timestamp, `/enc_zone/.Trash/<username>/<timestamp>`.

If you delete the entire encryption zone, it will be moved to the `.Trash` directory under the user's home directory, `/users/<username>/.Trash/Current/enc_zone`. Trash checkpointing will occur only after the entire zone has been moved to `/users/<username>/.Trash`. However, if the user's home directory is already part of an encryption zone, then attempting to delete an encryption zone will fail because you cannot move or rename directories across encryption zones.

If you have upgraded your cluster to CDH 5.7.1 (or higher), and you have an encryption zone that was created before the upgrade, create the `.Trash` directory using the `-provisionTrash` option as follows:

```
$ hdfs crypto -provisionTrash -path /enc_zone
```

In **CDH 5.7.0**, HDFS does not automatically create the `.Trash` directory when an encryption zone is created. However, you can use the following commands to manually create the `.Trash` directory within an encryption zone. Make sure you run the commands as an admin user.

```
$ hdfs dfs -mkdir /enc_zone/.Trash
$ hdfs dfs -chmod 1777 /enc_zone/.Trash
```

Configuring HDFS Trash Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Enabling and Disabling Trash

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Gateway**.
4. Select or clear the **Use Trash** checkbox.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart the cluster and deploy the cluster client configuration.

Setting the Trash Interval

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **NameNode**.
4. Specify the **Filesystem Trash Interval** property, which controls the number of minutes after which a trash checkpoint directory is deleted and the number of minutes between trash checkpoints. For example, to enable trash so that deleted files are deleted after 24 hours, set the value of the **Filesystem Trash Interval** property to 1440.

   > **Note:** The trash interval is measured from the point at which the files are moved to trash, not from the last time the files were modified.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart all NameNodes.

HDFS Balancers

HDFS data might not always be distributed uniformly across DataNodes. One common reason is addition of new DataNodes to an existing cluster. HDFS provides a balancer utility that analyzes block placement and balances data across the DataNodes. The balancer moves blocks until the cluster is deemed to be balanced, which means that the utilization of every DataNode (ratio of used space on the node to total capacity of the node) differs from the utilization of the cluster (ratio of used space on the cluster to total capacity of the cluster) by no more than a given threshold percentage. The balancer does not balance between individual volumes on a single DataNode.

Configuring and Running the HDFS Balancer Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

In Cloudera Manager, the HDFS balancer utility is implemented by the Balancer role. The Balancer role usually shows a health of **None** on the HDFS Instances tab because it does not run continuously.

The Balancer role is normally added (by default) when the HDFS service is installed. If it has not been added, you must add a Balancer role to rebalance HDFS and to see the **Rebalance** action.

Configuring the Balancer Threshold

The Balancer has a default threshold of 10%, which ensures that disk usage on each DataNode differs from the overall usage in the cluster by no more than 10%. For example, if overall usage across all the DataNodes in the cluster is 40% of the cluster's total disk-storage capacity, the script ensures that DataNode disk usage is between 30% and 50% of the DataNode disk-storage capacity. To change the threshold:

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Balancer**.
4. Select **Category** > **Main**.
5. Set the **Rebalancing Threshold** property.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Configuring Concurrent Moves

The property `dfs.datanode.balance.max.concurrent.moves` sets the maximum number of threads used by the DataNode balancer for pending moves. It is a throttling mechanism to prevent the balancer from taking too many resources from the DataNode and interfering with normal cluster operations. Increasing the value allows the balancing process to complete more quickly, decreasing the value allows rebalancing to complete more slowly, but is less likely to compete for resources with other tasks on the DataNode. To use this property, you need to set the value on both the DataNode and the Balancer.

- To configure the Datanode:
  - Go to the HDFS service.
  - Click the **Configuration** tab.
  - Search for **DataNode Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml**.
  - Add the following code to the configuration field, for example, setting the value to 50.

```
<property>
  <name>dfs.datanode.balance.max.concurrent.moves</name>
  <value>50</value>
</property>
```

  - Restart the DataNode.

- To configure the Balancer:
  1. Go to the HDFS service.
  2. Click the **Configuration** tab.
  3. Search for **Balancer Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml**.
  4. Add the following code to the configuration field, for example, setting the value to 50.

```
<property>
  <name>dfs.datanode.balance.max.concurrent.moves</name>
  <value>50</value>
</property>
```

Running the Balancer

1. Go to the HDFS service.
2. Ensure the service has a Balancer role.
3. Select **Actions** > **Rebalance**.
4. Click **Rebalance** to confirm. If you see a **Finished** status, the Balancer ran successfully.

Configuring Block Size

You can configure the Block Metadata Batch Size (`dfs.balancer.getBlocks.size`) and **Minimum Block Size** (`dfs.balancer.getBlocks.min-block-size`) for HDFS. The **Block Metadata Batch Size** property configures the amount of block metadata that gets retrieved. The **Minimum Block Size** property configures the smallest block to consider for moving.

Tuning these properties can improve performance during balancing:

1. In the Cloudera Manager Admin Console, select **Clusters** > **<HDFS cluster>**.
2. On the **Configuration** tab, search for the following properties:
   - **Block Metadata Batch Size** (`dfs.balancer.getBlocks.size`)
   - **Minimum Block Size** (`dfs.balancer.getBlocks.min-block-size`)

Enabling WebHDFS

Minimum Required Role: [Configurator] (also provided by **Cluster Administrator, Full Administrator**)

To enable WebHDFS, proceed as follows:

1. Select the HDFS service.
2. Click the **Configuration** tab.
3. Select **Scope** > **HDFS-1 (Service Wide)**
4. Select the **Enable WebHDFS** property.
5. Click the **Save Changes** button.
6. Restart the HDFS service.

WebHDFS uses the following prefix and URI format: `webhdfs://<HOST>:<HTTP_PORT>/<PATH>`

Secure WebHDFS uses the following prefix and URI format: `swebhdfs://<HOST>:<HTTP_PORT>/<PATH>`

You can find a full explanation of the WebHDFS API in the [WebHDFS API documentation](#).

Adding HttpFS

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

Apache Hadoop HttpFS is a service that provides HTTP access to HDFS.

HttpFS has a REST HTTP API supporting all HDFS filesystem operations (both read and write).

Common HttpFS use cases are:

- Read and write data in HDFS using HTTP utilities (such as `curl` or `wget`) and HTTP libraries from languages other than Java (such as Perl).
- Transfer data between HDFS clusters running different versions of Hadoop (overcoming RPC versioning issues), for example using Hadoop DistCp.
- Accessing WebHDFS using the Namenode WebUI port (default port 50070). Access to all data hosts in the cluster is required, because WebHDFS redirects clients to the datanode port (default 50075). If the cluster is behind a firewall, and you use WebHDFS to read and write data to HDFS, then Cloudera recommends you use the HttpFS server. The HttpFS server acts as a gateway. It is the only system that is allowed to send and receive data through the firewall.

HttpFS supports Hadoop pseudo-authentication, HTTP SPNEGO Kerberos, and additional authentication mechanisms using a plugin API. HttpFS also supports Hadoop proxy user functionality.

The `webhdfs` client file system implementation can access HttpFS using the Hadoop filesystem command (`hadoop fs`), by using Hadoop DistCp, and from Java applications using the Hadoop file system Java API.

The HttpFS HTTP REST API is interoperable with the WebHDFS REST HTTP API.

For more information about HttpFS, see [Hadoop HDFS over HTTP](#).

The HttpFS role is required for Hue when you enable [HDFS high availability](#).

Adding the HttpFS Role

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click **Add Role Instances**.
4. Click the text box below the **HttpFS** field. The Select Hosts dialog box displays.
5. Select the host on which to run the role and click **OK**.
6. Click **Continue**.
7. Check the checkbox next to the **HttpFS** role and select **Actions for Selected** > **Start**.

Using Load Balancer with HttpFS

Configure the HttpFS Service to work with the load balancer you configured for the service:

1. In the **Cloudera Manager Admin Console**, navigate to **Cluster** > **<HDFS service>**.

2. On the **Configuration** tab, search for the following property:

```
HttpFS Load Balancer
```

3. Enter the hostname and port for the load balancer in the following format:

```
<hostname>:<port>
```

4. Save the changes.

> **Note:**
>
> When you set this property, Cloudera Manager regenerates the keytabs for HttpFS roles. The principal in these keytabs contains the load balancer hostname.
>
> If there is a Hue service that depends on this HDFS service, the Hue service has the option to use the load balancer as its HDFS Web Interface Role.

Adding and Configuring an NFS Gateway

The NFSv3 gateway allows a client to mount HDFS as part of the client's local file system. The gateway machine can be any host in the cluster, including the NameNode, a DataNode, or any HDFS client. The client can be any NFSv3-client-compatible machine.

> **Important:**
>
> HDFS does not currently provide ACL support for an NFS gateway.

After mounting HDFS to his or her local filesystem, a user can:

- Browse the HDFS file system as though it were part of the local file system
- Upload and download files from the HDFS file system to and from the local file system.
- Stream data directly to HDFS through the mount point.

File append is supported, but random write is not.

Adding and Configuring an NFS Gateway Using Cloudera Manager

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

The NFS Gateway role implements an NFSv3 gateway. It is an optional role for the HDFS service.

Requirements and Limitations

- The nfs-utils OS package is required for a client to mount the NFS export and to run commands such as showmount from the NFS Gateway.
- If any NFS server is already running on the NFS Gateway host, it must be stopped before the NFS Gateway role is started.
- There are two configuration options related to NFS Gateway role: **Temporary Dump Directory** and **Allowed Hosts and Privileges**. The **Temporary Dump Directory** is automatically created by the NFS Gateway role and should be configured before starting the role.
- The **Access Time Precision** property in the HDFS service must be enabled.

Adding and Configuring the NFS Gateway Role

1. Go to the HDFS service.

2. Click the **Instances** tab.

3. Click **Add Role Instances**.

4. Click the text box below the **NFS Gateway** field. The Select Hosts dialog box displays.
5. Select the host on which to run the role and click **OK**.
6. Click **Continue**.
7. Click the **NFS Gateway** role.
8. Click the **Configuration** tab.
9. Select **Scope** > **NFS Gateway**.
10. Select **Category** > **Main**.
11. Ensure that the requirements on the directory set in the **Temporary Dump Directory** property are met.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

12. Optionally edit **Allowed Hosts and Privileges**.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

13. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
14. Click the **Instances** tab.
15. Check the checkbox next to the **NFS Gateway** role and select **Actions for Selected** > **Start**.

Setting HDFS Quotas

You can set quotas in HDFS for:

- The number of file and directory names used
- The amount of space used by given directories

Points to note:

- The quotas for names and the quotas for space are independent of each other.
- File and directory creation fails if the creation would cause the quota to be exceeded.
- The Reports Manager must index a file or directory before you can set a quota for it.
- Allocation fails if the quota would prevent a full block from being written; keep this in mind if you are using a large block size.
- If you are using replication, remember that each replica of a block counts against the quota.

About file count limits

- The file count quota is a limit on the number of file and directory names in the directory configured.
- A directory counts against its own quota, so a quota of 1 forces the directory to remain empty.
- File counts are based on the intended replication factor for the files; changing the replication factor for a file will credit or debit quotas.

About disk space limits

- The space quota is a hard limit on the number of bytes used by files in the tree rooted at the directory being configured.
- Each replica of a block counts against the quota.
- The disk space quota calculation takes replication into account, so it uses the replicated size of each file, not the user-facing size.
- The disk space quota calculation includes open files (files presently being written), as well as files already written.
- Block allocations for files being written will fail if the quota would not allow a full block to be written.

Setting HDFS Quotas Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. From the HDFS service page, select the **File Browser** tab.
2. Browse the file system to find the directory for which you want to set quotas.

3. Click the directory name so that it appears in the gray panel above the listing of its contents and in the detail section to the right of the File Browser table.

4. Click the **Edit Quota** button for the directory. A **Manage Quota** pop-up displays, where you can set file count or disk space limits for the directory you have selected.

5. When you have set the limits you want, click **OK**.

For More Information

For more information, see the HDFS Quotas Guide.

Configuring Mountable HDFS

CDH includes a FUSE (Filesystem in Userspace) interface into HDFS. The `hadoop-hdfs-fuse` package enables you to use your HDFS cluster as if it were a traditional filesystem on Linux. Proceed as follows.

> **Note:** FUSE does not currently support file append operations.

> **Important:** Cloudera *does not* recommend running FUSE in production environments.

Before you start: You must have a working HDFS cluster and know the hostname and port that your NameNode exposes. If you use parcels to install CDH, you do not need to install the FUSE packages.

To install hadoop-hdfs-fuses On Red Hat-compatible systems:

```
$ sudo yum install hadoop-hdfs-fuse
```

To install hadoop-hdfs-fuse on Ubuntu systems:

```
$ sudo apt-get install hadoop-hdfs-fuse
```

To install hadoop-hdfs-fuse on SLES systems:

```
$ sudo zypper install hadoop-hdfs-fuse
```

You now have everything you need to begin mounting HDFS on Linux.

To set up and test your mount point in a non-HA installation:

```
$ mkdir -p <mount_point>
$ hadoop-fuse-dfs dfs://<name_node_hostname>:<namenode_port> <mount_point>
```

where `namenode_port` is the NameNode's RPC port, `dfs.namenode.servicerpc-address`.

To set up and test your mount point in an HA installation:

```
$ mkdir -p <mount_point>
$ hadoop-fuse-dfs dfs://<nameservice_id> <mount_point>
```

where `nameservice_id` is the value of `fs.defaultFS`. In this case the port defined for `dfs.namenode.rpc-address.[nameservice ID].[name node ID]` is used automatically. See Enabling HDFS HA on page 424 for more information about these properties.

You can now run operations as if they are on your mount point. Press **Ctrl+C** to end the `fuse-dfs` program, and `umount` the partition if it is still mounted.

> **Note:**
>
> To find its configuration directory, `hadoop-fuse-dfs` uses the `HADOOP_CONF_DIR` configured at the time the `mount` command is invoked.

To clean up your test:

```
$ umount <mount_point>
```

You can now add a permanent HDFS mount which persists through reboots.

To add a system mount:

1. Open `/etc/fstab` and add lines to the bottom similar to these:

```
hadoop-fuse-dfs#dfs://<name_node_hostname>:<namenode_port> <mount_point> fuse
allow_other,usetrash,rw 2 0
```

For example:

```
hadoop-fuse-dfs#dfs://localhost:8020 /mnt/hdfs fuse allow_other,usetrash,rw 2 0
```

> **Note:**
>
> In an HA deployment, use the HDFS nameservice instead of the NameNode URI; that is, use the value of `dfs.nameservices` in `hdfs-site.xml`.

2. Test to make sure everything is working properly:

```
$ mount <mount_point>
```

Your system is now configured to allow you to use the `ls` command and use that mount point as if it were a normal system disk.

For more information, see the help for `hadoop-fuse-dfs`:

```
$ hadoop-fuse-dfs --help
```

Optimizing Mountable HDFS

- Cloudera recommends that you use the `-obig_writes` option on kernels later than 2.6.26. This option allows for better performance of writes.
- By default, the CDH package installation creates the `/etc/default/hadoop-fuse` file with a maximum heap size of 128 MB. You might need to change the JVM minimum and maximum heap size for better performance. For example:

```
export LIBHDFS_OPTS="-Xms64m -Xmx256m"
```

Be careful not to set the minimum to a higher value than the maximum.

Configuring Centralized Cache Management in HDFS

Centralized cache management in HDFS is an explicit caching mechanism that allows users to specify paths to be cached by HDFS. The NameNode communicates with DataNodes and instructs them to cache specific blocks in off-heap caches.

Centralized and explicit caching has several advantages:

- Frequently used data is pinned in memory. This is important when the size of the working set exceeds the size of main memory, which is common for many HDFS workloads.
- Cluster memory is optimized because you can pin `m` of `n` block replicas, saving `n-m` memory. Before centralized pinning, repeated reads of a block caused all `n` replicas to be pulled into each DataNode buffer cache.
- Tasks are co-located with cached block replicas, improving read performance. Because the NameNode manages DataNode caches, applications can query the set of cached block locations when making task placement decisions.
- Clients can use the zero-copy read API, and incur almost no overhead, because each DataNode does a checksum verification of cached data only once.

Use Cases

Centralized cache management is best used for files that are accessed repeatedly. For example, a fact table in Hive that is often used in `JOIN` clauses is a good candidate for caching. Caching the input of an annual reporting query is probably less useful, as the historical data might be read only once.

Centralized cache management is also useful for mixed workloads with performance service-level agreements (SLAs). Caching the working set of a high-priority workload insures that it does not contend for disk I/O with a low-priority workload.

Architecture
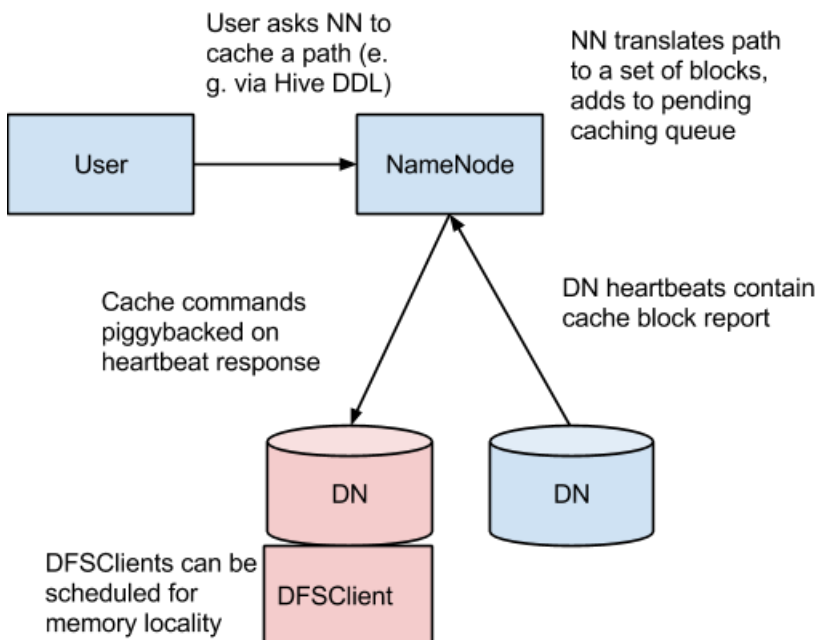
In this architecture, the NameNode is responsible for coordinating all the DataNode off-heap caches in the cluster. The NameNode periodically receives a "cache report" from each DataNode which describes all the blocks cached on a given DataNode. The NameNode manages DataNode caches by piggybacking cache and uncache commands on the DataNode heartbeat.

The NameNode queries its set of cache directives to determine which paths should be cached. Cache directives are persistently stored in the fsimage and edit log, and can be added, removed, and modified using Java and command-line APIs. The NameNode also stores a set of cache pools, which are administrative entities used to group cache directives together for resource management and enforcing permissions.

The NameNode periodically rescans the namespace and active cache directories to determine which blocks need to be cached or uncached and assigns caching to DataNodes. Rescans can also be triggered by user actions such as adding or removing a cache directive or removing a cache pool.

Currently, blocks that are under construction, corrupt, or otherwise incomplete are not cached. If a cache directive covers a symlink, the symlink target is not cached. Caching is currently done on a per-file basis (and not at the block-level).

Concepts
Cache Directive

A **cache directive** defines a path that should be cached. Paths can be either directories or files. Directories are cached non-recursively, meaning only files in the first-level listing of the directory are cached.

Directives have parameters, such as the cache replication factor and expiration time. Replication factor specifies the number of block replicas to cache. If multiple cache directives refer to the same file, the maximum cache replication factor is applied. Expiration time is specified on the command line as a `time-to-live` (TTL), a relative expiration time in the future. After a cache directive expires, it is no longer considered by the NameNode when making caching decisions.

Cache Pool

A **cache pool** is an administrative entity used to manage groups of cache directives. Cache pools have UNIX-like permissions that restrict which users and groups have access to the pool. Write permissions allow users to add and remove cache directives to the pool. Read permissions allow users to list the cache directives in a pool, as well as additional metadata. Execute permissions are not used.

Cache pools are also used for resource management. Pools can enforce a maximum `limit` that restricts the aggregate number of bytes that can be cached by directives in the pool. Normally, the sum of the pool limits roughly equals the amount of aggregate memory reserved for HDFS caching on the cluster. Cache pools also track a number of statistics to help cluster users determine what is and should be cached.

Pools also enforce a maximum time-to-live. This restricts the maximum expiration time of directives being added to the pool.

cacheadmin Command-Line Interface

On the command-line, administrators and users can interact with cache pools and directives using the `hdfs cacheadmin` subcommand. Cache directives are identified by a unique, non-repeating 64-bit integer ID. IDs are not reused even if a cache directive is later removed. Cache pools are identified by a unique string name.

Cache Directive Commands

addDirective

Description: Add a new cache directive.

Usage: `hdfs cacheadmin -addDirective -path <path> -pool <pool-name> [-force] [-replication <replication>] [-ttl <time-to-live>]`

Where, `path`: A path to cache. The path can be a directory or a file.

`pool-name`: The pool to which the directive will be added. You must have write permission on the cache pool to add new directives.

`force`: Skips checking of cache pool resource limits.

`replication`: The cache replication factor to use. Defaults to 1.

`time-to-live`: Time period for which the directive is valid. Can be specified in seconds, minutes, hours, and days, for example: 30m, 4h, 2d. The value `never` indicates a directive that never expires. If unspecified, the directive never expires.

removeDirective

Description: Remove a cache directive.

Usage: `hdfs cacheadmin -removeDirective <id>`

Where, `id`: The id of the cache directive to remove. You must have write permission on the pool of the directive to remove it. To see a list of PathBasedCache directive IDs, use the -listDirectives command.

removeDirectives

Description: Remove every cache directive with the specified path.

Usage: `hdfs cacheadmin -removeDirectives <path>`

Where, `path`: The path of the cache directives to remove. You must have write permission on the pool of the directive to remove it.

listDirectives

Description: List PathBasedCache directives.

Usage: `hdfs cacheadmin -listDirectives [-stats] [-path <path>] [-pool <pool>]`

Where, `path`: List only PathBasedCache directives with this path. Note that if there is a PathBasedCache directive for `path` in a cache pool that we do not have read access for, it will not be listed.

`pool`: List only path cache directives in that pool.

`stats`: List path-based cache directive statistics.

Cache Pool Commands

addPool

Description: Add a new cache pool.

Usage: `hdfs cacheadmin -addPool <name> [-owner <owner>] [-group <group>] [-mode <mode>]
[-limit <limit>] [-maxTtl <maxTtl>]`

Where, `name`: Name of the new pool.

`owner`: Username of the owner of the pool. Defaults to the current user.

`group`: Group of the pool. Defaults to the primary group name of the current user.

`mode`: UNIX-style permissions for the pool. Permissions are specified in octal, for example: 0755. By default, this is set to 0755.

`limit`: The maximum number of bytes that can be cached by directives in this pool, in aggregate. By default, no limit is set.

`maxTtl`: The maximum allowed time-to-live for directives being added to the pool. This can be specified in seconds, minutes, hours, and days, for example: 120s, 30m, 4h, 2d. By default, no maximum is set. A value of `never` specifies that there is no limit.

modifyPool

Description: Modify the metadata of an existing cache pool.

Usage: `hdfs cacheadmin -modifyPool <name> [-owner <owner>] [-group <group>] [-mode <mode>]
[-limit <limit>] [-maxTtl <maxTtl>]`

Where, `name`: Name of the pool to modify.

`owner`: Username of the owner of the pool.

`group`: Groupname of the group of the pool.

`mode`: Unix-style permissions of the pool in octal.

`limit`: Maximum number of bytes that can be cached by this pool.

`maxTtl`: The maximum allowed time-to-live for directives being added to the pool.

removePool

Description: Remove a cache pool. This also uncaches paths associated with the pool.

Usage: `hdfs cacheadmin -removePool <name>`

Where, `name`: Name of the cache pool to remove.

listPools

Description: Display information about one or more cache pools, for example: name, owner, group, permissions, and so on.

Usage: `hdfs cacheadmin -listPools [-stats] [<name>]`

Where, `name`: If specified, list only the named cache pool.

`stats`: Display additional cache pool statistics.

help

Description: Get detailed help about a command.

Usage: `hdfs cacheadmin -help <command-name>`

Where, `command-name`: The command for which to get detailed help. If no command is specified, print detailed help for all commands.

Configuration
Native Libraries

To lock block files into memory, the DataNode relies on native JNI code found in `libhadoop.so`. Be sure to enable JNI if you are using HDFS centralized cache management.

Configuration Properties

Required

Be sure to configure the following in `/etc/default/hadoop/conf/hdfs-default.xml`:

- `dfs.datanode.max.locked.memory`: The maximum amount of memory a DataNode uses for caching (in bytes). The "locked-in-memory size" ulimit (`ulimit -l`) of the DataNode user also needs to be increased to match this parameter (see OS Limits). When setting this value, remember that you need space in memory for other things as well, such as the DataNode and application JVM heaps and the operating system page cache.

Optional

The following properties are not required, but may be specified for tuning:

- `dfs.namenode.path.based.cache.refresh.interval.ms`: The NameNode uses this as the amount of milliseconds between subsequent path cache rescans. This calculates the blocks to cache and each DataNode containing a replica of the block that should cache it. By default, this parameter is set to 300000, which is five minutes.
- `dfs.datanode.fsdatasetcache.max.threads.per.volume`: The DataNode uses this as the maximum number of threads per volume to use for caching new data. By default, this parameter is set to 4.
- `dfs.cachereport.intervalMsec`: The DataNode uses this as the amount of milliseconds between sending a full report of its cache state to the NameNode. By default, this parameter is set to 10000, which is 10 seconds.
- `dfs.namenode.path.based.cache.block.map.allocation.percent`: The percentage of the Java heap which we will allocate to the cached blocks map. The cached blocks map is a hash map which uses chained hashing. Smaller maps may be accessed more slowly if the number of cached blocks is large; larger maps will consume more memory. By default, this parameter is set to 0.25 percent.

OS Limits

If you get the error, `Cannot start datanode because the configured max locked memory size... is more than the datanode's available RLIMIT_MEMLOCK ulimit`, the operating system is imposing a lower limit on the amount of memory that you can lock than what you have configured. To fix this, adjust the DataNode `ulimit -l` value. Usually, this value is configured in `/etc/security/limits.conf`; but varies depending on your operating system and distribution.

You have correctly configured this value when you can run `ulimit -l` from the shell and get back either a higher value than what you have configured with `dfs.datanode.max.locked.memory`, or the string `unlimited`, indicating that there is no limit. It is typical for `ulimit -l` to output the memory lock limit in KB, but `dfs.datanode.max.locked.memory` must be specified in bytes.

Configuring Proxy Users to Access HDFS

Hadoop allows you to configure proxy users to submit jobs or access HDFS on behalf of other users; this is called *impersonation*. When you enable impersonation, any jobs submitted using a proxy are executed with the impersonated user's existing privilege levels rather than those of a superuser (such as `hdfs`). Because all proxy users are configured in one location, `core-site.xml`, Hadoop administrators to implement centralized access control.

To configure proxy users, set the `hadoop.proxyuser.<proxy_user>.hosts`, `hadoop.proxyuser.<proxy_group>.groups` and `hadoop.proxyuser.<proxy_user>.users` in `core-site.xml` properties.

For example, to allow user `alice` to impersonate a user belonging to `group_a` and `group_b`, set `hadoop.proxyuser.<proxy_group>.groups` as follows:

```
<property>
  <name>hadoop.proxyuser.alice.groups</name>
  <value>group_a,group_b</value>
</property>
```

To limit the hosts from which impersonated connections are allowed, use `hadoop.proxyuser.<proxy_user>.hosts`. For example, to allow user `alice` impersonated connections only from `host_a` and `host_b`:

```
<property>
    <name>hadoop.proxyuser.alice.hosts</name>
    <value>host_a,host_b</value>
</property>
```

If the configuration properties described are not present, impersonation is not allowed and connections will fail.

For looser restrictions, use a wildcard (`*`) to allow impersonation from any host and of any user. For example, to allow user `bob` to impersonate any user belonging to any group, and from any host, set the properties as follows:

```
<property>
  <name>hadoop.proxyuser.bob.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.bob.groups</name>
  <value>*</value>
</property>
```

The `hadoop.proxyuser.<proxy_user>.hosts` property also accepts comma-separated lists of IP addresses, IP address ranges in CIDR format, or host names. For example, to allow user `kate` access from hosts in the range `10.222.0.0-15` and `10.113.221.221`, to impersonate `user_a` and `user_b`, set the proxy user properties as follows:

```
<property>
    <name>hadoop.proxyuser.super.hosts</name>
    <value>10.222.0.0/16,10.113.221.221</value>
</property>
<property>
    <name>hadoop.proxyuser.super.users</name>
    <value>user1,user2</value>
</property>
```

Proxy Users for Kerberos-Enabled Clusters

For secure clusters, the proxy users must have Kerberos credentials to impersonate another user.

Proxy users cannot use delegation tokens. If a user is allowed to add its own delegation token to the proxy user UGI, it also allows the proxy user to connect to the service with the privileges of the original user.

If a superuser wants to give a delegation token to a proxy-user UGI, for example, `alice`, the superuser must first impersonate `alice`, get a delegation token for `alice`, and add it to the UGI for the newly created proxy UGI. This way, the delegation token has its owner set to `alice`.

Using CDH with Isilon Storage

> ⚠️ **Warning:** CDH 6 is not currently supported on Dell EMC Isilon.

Dell EMC Isilon is a storage service with a distributed filesystem that can used in place of HDFS to provide storage for CDH services.

> 📝 **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

> 📝 **Note:** This documentation covers only the Cloudera Manager portion of using EMC Isilon storage with CDH. For information about tasks performed on Isilon OneFS, see the Dell EMC Community's [Isilon Info Hub](#).

Supported Versions

For Cloudera and Isilon compatibility information, see the product compatibility matrix for [Product Compatibility for Dell EMC Isilon](#).

Differences Between Isilon HDFS and CDH HDFS

The following features of HDFS are not implemented with Isilon OneFS:

- HDFS caching
- HDFS encryption
- HDFS ACLs

Installing Cloudera Manager and CDH with Isilon

For instructions on configuring Isilon and installing Cloudera Manager and CDH with Isilon, see the following EMC documentation:

- [EMC Isilon OneFS with Hadoop and Cloudera for Kerberos Installation Guide (PD](#)
- [EMC Isilon OneFS with Cloudera Hadoop Installation Guide (PDF)](#)

Upgrading a Cluster with Isilon

To upgrade CDH and Cloudera Manager in a cluster that uses Isilon:

1. If required, upgrade OneFS to a version compatible with the version of CDH to which you are upgrading. See the product compatibility matrix for [Product Compatibility for Dell EMC Isilon](#). For OneFS upgrade instructions, see the EMC Isilon documentation.
2. (Optional) Upgrade Cloudera Manager. See [Upgrading Cloudera Manager](#).
3. Upgrade CDH. See [Upgrading CDH](#).

Configuring Replication with Kerberos and Isilon

If you plan to use replication between clusters that use Isilon storage and that also have enabled Kerberos, do the following:

1. Create a custom Kerberos Keytab and Kerberos principal that the replication jobs use to authenticate to storage and other CDH services. See [Authentication](#).
2. In Cloudera Manager, select **Administration** > **Settings**.
3. Search for and enter values for the following properties:

   - **Custom Kerberos Keytab Location** – Enter the location of the Custom Kerberos Keytab.
   - **Custom Kerberos Principal Name** – Enter the principal name to use for replication between secure clusters.

4. When you create a replication schedule, enter the **Custom Kerberos Principal Name** in the **Run As Username** field. See [Configuring Replication of HDFS Data](#) on page 489 and [Configuring Replication of Hive/Impala Data](#) on page 500.
5. Ensure that both the source and destination clusters have the same set of users and groups. When you set ownership of files (or when maintaining ownership), if a user or group does not exist, the `chown` command fails on Isilon. See [Performance and Scalability Limitations](#) on page 488
6. Cloudera recommends that you do not select the **Replicate Impala Metadata** option for Hive/Impala replication schedules. If you need to use this feature, create a custom principal of the form `hdfs/`*`hostname@realm`* or `impala/`*`hostname@realm.`*
7. Add the following property and value to the **HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** and **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** properties:

```
hadoop.security.token.service.use_ip = false
```

If the replication MapReduce job fails with the an error similar to the following:

```
java.io.IOException: Failed on local exception: java.io.IOException:
  org.apache.hadoop.security.AccessControlException:
  Client cannot authenticate via:[TOKEN, KERBEROS];
  Host Details : local host is: "foo.mycompany.com/172.1.2.3";
  destination host is: "myisilon-1.mycompany.com":8020;
```

Set the Isilon cluster-wide time-to-live setting to a higher value on the *destination* cluster for the replication: Note that higher values may affect load balancing in the Isilon cluster by causing workloads to be less distributed. A value of 60 is a good starting point. For example:

```
isi networks modify pool subnet4:nn4 --ttl=60
```

You can view the settings for a subnet with a command similar to the following:

```
isi networks list pools --subnet subnet3 -v
```

Configuring Heterogeneous Storage in HDFS

CDH supports a variety of storage types in the [Hadoop Distributed File System (HDFS)](#). Earlier releases of CDH used a single (or homogeneous) storage model. Now you can choose which storage type to assign to each DataNode Data Directory. Specifying a storage type allows you to optimize your data usage and lower your costs, based on your data usage frequency. This topic describes these storage types and how to configure CDH to use them.

Overview

Each DataNode in a cluster is configured with a set of data directories. You can configure each data directory with a storage type. The storage policy dictates which storage types to use when storing the file or directory.

Some reasons to consider using different types of storage are:

- You have datasets with temporal locality (for example, time-series data). The latest data can be loaded initially into SSD for improved performance, then migrated out to disk as it ages.

- You need to move cold data to denser archival storage because the data will rarely be accessed and archival storage is much cheaper. This could be done with simple age-out policies: for example, moving data older than six months to archival storage.

Storage Types

The storage type identifies the underlying storage media. HDFS supports the following storage types:

- ARCHIVE - Archival storage is for very dense storage and is useful for rarely accessed data. This storage type is typically cheaper per TB than normal hard disks.
- DISK - Hard disk drives are relatively inexpensive and provide sequential I/O performance. This is the default storage type.
- SSD - Solid state drives are useful for storing hot data and I/O-intensive applications.
- RAM_DISK - This special in-memory storage type is used to accelerate low-durability, single-replica writes.

When you add the DataNode Data Directory, you can specify which type of storage it uses, by prefixing the path with the storage type, in brackets. If you do not specify a storage type, it is assumed to be `DISK`. See Adding Storage Directories on page 99.

Storage Policies

A storage policy contains information that describes the type of storage to use. This policy also defines the fallback storage type if the primary type is out of space or out of quota. If a target storage type is not available, HDFS attempts to place replicas on the default storage type.

Each storage policy consists of a policy ID, a policy name, a list of storage types, a list of fallback storage types for file creation, and a list of fallback storage types for replication.

HDFS has six preconfigured storage policies.

- Hot - All replicas are stored on DISK.
- Cold - All replicas are stored ARCHIVE.
- Warm - One replica is stored on DISK and the others are stored on ARCHIVE.
- All_SSD - All replicas are stored on SSD.
- One_SSD - One replica is stored on SSD and the others are stored on DISK.
- Lazy_Persist - The replica is written to RAM_DISK and then lazily persisted to DISK.

> **Note:** You cannot create your own storage policy. You must use one of the six pre-configured policies. HDFS clients such as HBase may support different storage policies.

Setting Up SSD Storage Using Cloudera Manager

1. Set up your cluster normally, but customize your DataNodes with the [ssd] prefix for data directories. Adding [ssd] can also be done after initial setup (which requires an extra HDFS restart).
2. Stop HBase.
3. Using the HDFS client, move `/hbase` to `/hbase_backup`.
4. Re-create `/hbase` using the Cloudera Manager command in the HBase service (this ensures that proper permissions are used).
5. Using the HDFS client, set the storage policy for `/hbase` to be SSD only.
6. Use the DistCp to copy `/hbase_backup` to `/hbase`.

   ```
   hadoop distcp /hbase_backup /hbase
   ```

7. Start HBase.

Setting a Storage Policy for HDFS

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

To set a storage policy on a DataNode Data Directory using Cloudera Manager, perform the following tasks:

1. Check the **HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** to be sure that `dfs.storage.policy.enabled` has not been changed from its default value of `true`.

2. Specify the storage types for each DataNode Data Directory that is not a standard disk, by adding the storage type in brackets at the beginning of the directory path. For example:

```
[SSD]/dfs/dn1
[DISK]/dfs/dn2
[ARCHIVE]/dfs/dn3
```

3. Open a terminal session on any HDFS host. Run the following hdfs command for each path on which you want to set a storage policy:

```
$ hdfs storagepolicies -setStoragePolicy -path <path> -policy <policy>
path_to_file_or_directory -policy policy_name
```

4. To move the data to the appropriate storage based on the current storage policy, use the `mover` utility, from any HDFS host. Use `mover -h` to get a list of available options. To migrate all data at once (this may take a long time), you can set the path to `/`.

```
$ hdfs mover -p <path>
```

> **Note:** Quotas are enforced at the time you set the storage policy or when writing the file, not when quotas are changed. The Mover tool does not recognize quota violations. It only verifies that a file is stored on the storage types specified in its policy. For more information about quotas, see [Setting HDFS Quotas](#) on page 109.

Managing Storage Policies

- To get the storage policy for a specific file or directory on a DataNode, use the following command, which is available using the command line on a any HDFS host.

```
$ hdfs storagepolicies -getStoragePolicy -path <path>
```

- To list all policies on a DataNode, enter the following command:

```
$ hdfs storagepolicies -listPolicies
```

- To reset a storage policy, follow the steps used in [Setting a Storage Policy for HDFS](#) on page 119.

Migrating Existing Data

To move the data to the appropriate storage based on the current storage policy, use the `mover` utility, from any HDFS host. Use `mover -h` to get a list of available options. To migrate all data at once (this may take a long time), you can set the path to `/`.

```
$ hdfs mover -p <path>
```

> **Note:** Quotas are enforced at the time you set the storage policy or when writing the file, not when quotas are changed. The Mover tool does not recognize quota violations. It only verifies that a file is stored on the storage types specified in its policy. For more information about quotas, see [Setting HDFS Quotas](#) on page 109.

Managing Apache Hive in CDH

Cloudera recommends using Cloudera Manager to manage Hive services, which are called managed deployments. If yours is not a managed deployment, configure HiveServer2 Web UI to manage Hive services. Also see [Managing Hive](#) in the Hive Guide for more information about managing the Hive service in CDH. For information about configuring

Hive, see Configuring Apache Hive in CDH, which includes information about selecting JDBC drivers and ODBC drivers to use with Hive.

Using Cloudera Manager to Manage Hive

Cloudera Manager uses the Hive metastore, HiveServer2, and the WebHCat roles to manage the Hive service across your cluster. Using Cloudera Manager, you can configure the Hive metastore, the execution engine (either MapReduce or Spark), and manage HiveServer2.

See Managing Hive Using Cloudera Manager

Using HiveServer2 Web UI to Manage Hive

The HiveServer2 web UI provides access to Hive configuration settings, local logs, metrics, and information about active sessions and queries. The HiveServer2 web UI is enabled in newly created clusters running CDH 5.7 and higher, and those using Kerberos are configured for SPNEGO. Clusters upgraded from a previous CDH version must be configured to enable the web UI; see HiveServer2 Web UI Configuration.

Managing Hue

Hue is a set of web UIs that enable you to interact with a CDH cluster. This section describes tasks for managing Hue.

Adding a Hue Service and Role Instance
Adding the Hue Service

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

After initial installation, you can use the **Add a Service** wizard in Cloudera Manager to add and configure a new Hue service instance.

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the cluster name.
2. Select **Add a Service**.
3. Select the **Hue** service and click **Continue**.
4. Select the row with the Hue dependencies required for your cluster.
5. Click **Continue** to accept the default role assignments; or click the gray field below each role to open the hosts dialog, customize assignments, and click **OK** to save.

   If a drop down menu displays (indicating that all hosts apply), select **All Hosts**, or else click **Custom** to display the hosts dialog. Click **OK** to accept custom assignments.

   The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

6. Select **Use Custom Databases** for production clusters and input values for database *hostname*, *type*, *name*, *username*, and *password*.
7. Click **Test Connection**, and when green, click **Continue**. Cloudera Manager starts the Hue service.
8. Click **Continue** and **Finish**.

9. If your cluster uses Kerberos, Cloudera Manager *automatically* adds a **Hue Kerberos Ticket Renewer** role to each host where you assigned the Hue Server role instance. See Enable Hue to Use Kerberos for Authentication.

Adding a Hue Role Instance

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Roles are functions that comprise a service and role instances must be assigned to one or more hosts. You can easily assign roles to hosts in Cloudera Manager.

1. Go to the **Hue** service.
2. Click the **Instances** tab.
3. Click the **Add Role Instances** button.
4. Click **Continue** to accept the default role assignments; or click the gray field below each role to open the hosts dialog, customize assignments, and click **OK** to save.

   If a drop down menu displays (indicating that all hosts apply), select **All Hosts**, or else click **Custom** to display the hosts dialog. Click **OK** to accept custom assignments.

   The wizard evaluates host hardware configurations to determine the best hosts for each role. All worker roles are automatically assigned to the same set of hosts as the HDFS DataNode. You can reassign if necessary. Specify hostnames by IP address, rack name, or by range:

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

5. If your cluster uses Kerberos, you must *manually* add the **Hue Kerberos Ticket Renewer** role to each host where you assigned the Hue Server role instance. Cloudera Manager throws a validation error if the new Hue Server role does not have a colocated KT Renewer role. See Enable Hue to Use Kerberos for Authentication.
6. Click **Continue**.

Managing Hue Analytics Data Collection

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Hue tracks anonymized pages and application versions to collect information used to compare each application's usage levels. The data collected does not include hostnames or IDs; For example, the data has the format /2.3.0/pig, /2.5.0/beeswax/execute. You can restrict data collection as follows:

1. Go to the Hue service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hue**.
4. Locate the **Enable Usage Data Collection** property or search for it by typing its name in the Search box.
5. Clear the **Enable Usage Data Collection** checkbox.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the Hue service.

Enabling Hue Applications Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Most Hue applications are configured by default, based on the services you have installed. Cloudera Manager selects the service instance that Hue depends on. If you have more than one service, you may want to verify or change the

service dependency for Hue. Also, if you add a service such as Sqoop or Oozie after you have set up Hue, you need to set the dependency because it is not done automatically. To add a dependency:

1. Go to the **Hue** service and click the **Configuration** tab.
2. Filter by **Scope** > **Hue (Service-Wide)** and **Category** > **Main**.
3. Select each *service name* **Service** property to set the dependency. Select **none** to remove the dependency.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
5. Restart the Hue service.

Enabling the HBase Browser Application with doAs Impersonation

Minimum Required Role: [Full Administrator](#)

The Hue HBase application communicates through the proxy, HBase Thrift Server, which forwards commands to HBase. Because Hue stands between the Thrift server and the user, all HBase operations appear to come from the Hue user and not the actual user who is logged on. In a Keberos cluster, you can enable impersonation so that operations appear to come from the actual user.

1. Logon to Cloudera Manager.
2. [Add the HBase Thrift Server role](#):

   a. Go to the **HBase** service and click the **Instances** tab.
   b. Click the button, **Add Role Instances**.
   c. Click **Select hosts** under HBase Thrift Server.
   d. Click anywhere in host row to add the purple icon, "HBTS," under Existing Roles.
   e. Click **OK** and **Continue**.
   f. Check the box by your new HBase Thrift Server and select **Actions for Selected** > **Start**.

3. If you have a [Kerberos cluster](#), enable impersonation. If your cluster does not have Kerberos or TLS enabled, skip to step 6..

   > **Note:** Enabling impersonation requires that you grant Hbase permissions to each individual user. Otherwise, grant all HBase permissions to the Hue user.

   a. Click the HBase **Configuration** tab.
   b. Filter by **Scope** > **Service-Wide** and **Category** > **Security**.
   c. Set the property, **HBase Thrift Authentication** (hbase.thrift.security.qop), to one of the following values:

      - `auth-conf`: authentication, integrity and confidentiality checking
      - `auth-int`: authentication and integrity checking
      - `auth`: authentication only

   d. Filter by **Scope** > **Service-Wide** and **Category** > **Main**.
   e. Check the **Service-Wide** box for **Enable HBase Thrift Http Server** (hbase.regionserver.thrift.http) and **Enable HBase Thrift Proxy Users** (hbase.thrift.support.proxyuser).
   f. Click **Save Changes**.

4. If you have a [Kerberos cluster](#) with doAs and force principal names to lower case, be sure to exclude the HTTP principal:

   a. Go to the **HDFS** service.
   b. Filter by **Scope** > **HDFS (Service-Wide)** and **Category** > **Security**.
   c. Search on **Additional Rules to Map Kerberos Principals to Short Names** (auth_to_local) and add two HTTP rules above your existing rules:

```
# Exclude HTTP
RULE:[1:$1@$0](HTTP@\QEXAMPLE.COM\E$)s/@\Q.EXAMPLE.COM\E$//
RULE:[2:$1@$0](HTTP@\QEXAMPLE.COM\E$)s/@\Q.EXAMPLE.COM\E$//
```

```
# Force to Lower Case
RULE:[1:$1@$0](.*@\QEXAMPLE.COM\E$)s/@\Q.EXAMPLE.COM\E$///L
RULE:[2:$1@$0](.*@\QEXAMPLE.COM\E$)s/@\Q.EXAMPLE.COM\E$///L
```

    d. Click **Save Changes**.

    e. Select **Actions** > **Deploy Client Configuration**.

    f. Select **Cluster** > **Actions** > **Rolling Restart**, check the boxes for HDFS, HBase, and Hue and click **Rolling Restart**.

  5. Enable TLS/SSL for the HBase Thrift Server:

    a. Filter by **Scope** > **HBase Thrift Server** and **Category** > **Security**.

    b. Set the TLS/SSL properties according to your cluster configuration:

Property	Description
Enable TLS/SSL for HBase Thrift Server over HTTP	Encrypt communication between clients and HBase Thrift Server over HTTP using Transport Layer Security (TLS).
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore File Location	Path to the TLS/SSL keystore file (in JKS format) with the TLS/SSL server certificate and private key. Used when HBase Thrift Server over HTTP acts as a TLS/SSL server.
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore File Password	Password for the HBase Thrift Server JKS keystore file.
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore Key Password	Password that protects the private key contained in the JKS keystore used when HBase Thrift Server over HTTP acts as a TLS/SSL server.

    c. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

    d. Restart the **HBase** service.

  6. Configure Hue to point to the Thrift Server and to a valid HBase configuration directory:

    a. Go to the **Hue** service and click the **Configuration** tab.

    b. Filter by **Scope** > **All** and **Category** > **Main**.

    c. Set the property, **HBase Service**, to the service for which you enabled the Thrift Server role (if you have more than one HBase service instance).

    d. Set the property, **HBase Thrift Server**, to the Thrift Server role for Hue to use.

    e. Filter by **Category** > **Advanced**.

    f. Edit the property, **Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini**, by adding a valid HBase configuration directory as follows:

```
[hbase]
hbase_conf_dir={{HBASE_CONF_DIR}}
```

    g. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Managing Impala

This section explains how to configure Impala to accept connections from applications that use popular programming APIs:

- Post-Installation Configuration for Impala on page 129
- Configuring Impala to Work with ODBC on page 131
- Configuring Impala to Work with JDBC on page 133

This type of configuration is especially useful when using Impala in combination with Business Intelligence tools, which use these standard interfaces to query different kinds of database and Big Data systems.

You can also configure these other aspects of Impala:

- Impala Security Overview
- Modifying Impala Startup Options on page 127

The Impala Service

The Impala Service is the Cloudera Manager representation of the three daemons that make up the Impala interactive SQL engine. Through the Impala Service page, you can monitor, start and stop, and configure all the related daemons from a central page.

For general information about Impala and how to use it, especially for writing Impala SQL queries, see Apache Impala - Interactive SQL.

For information on features that support Impala resource management see Admission Control and Query Queuing on page 389.

Installing Impala and Creating the Service

You can install Impala through the Cloudera Manager installation wizard, using either parcels or packages, and have the service created and started as part of the Installation wizard. See Cloudera Installation Guide.

If you elect not to include the Impala service using the Installation wizard, you can use the **Add Service** wizard to perform the installation. The wizard will automatically configure and start the dependent services and the Impala service. See Adding a Service on page 193 for instructions.

Configuring the Impala Service

There are several types of configuration settings you may need to apply, depending on your situation.

Configuring Table Statistics

Configuring table statistics is highly recommended when using Impala. It allows Impala to make optimizations that can result in significant (over 10x) performance improvement for some joins. If these are not available, Impala will still function, but at lower performance.

The Impala implementation to compute table statistics is available in CDH 5.0.0 or higher and in Impala version 1.2.2 or higher. The Impala implementation of `COMPUTE STATS` requires no setup steps and is preferred over the Hive implementation. See Overview of Table Statistics. If you are running an older version of Impala, follow the procedure in Accessing Apache Hive Table Statistics in CDH.

Using a Load Balancer with Impala

To configure a load balancer:

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Impala Daemon**
4. Select **Category** > **All**
5. Enter the hostname and port number of the load balancer in the **Impala Daemons Load Balancer** property in the format *hostname:port number*.

> **Note:**
>
> When you set this property, Cloudera Manager regenerates the keytabs for Impala Daemon roles. The principal in these keytabs contains the load balancer hostname.
>
> If there is a Hue service that depends on this Impala service, it also uses the load balancer to communicate with Impala.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Impala Web Servers

Enabling and Disabling Access to Impala Web Servers

Each of the Impala-related daemons includes a built-in web server that lets an administrator diagnose issues with each daemon on a particular host, or perform other administrative actions such as cancelling a running query. By default, these web servers are enabled. You might turn them off in a high-security configuration where it is not appropriate for users to have access to this kind of monitoring information through a web interface. (To leave the web servers enabled but control who can access their web pages, consult the *Configuring Secure Access for Impala Web Servers* later in this section.)

- **Impala Daemon**

  1. Go to the Impala service.
  2. Click the **Configuration** tab.
  3. Select **Scope** > **Impala Daemon**
  4. Select **Category** > **Ports and Addresses**.
  5. Select or clear **Enable Impala Daemon Web Server**.
  6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
  7. Restart the Impala service.

- **Impala StateStore**

  1. Go to the Impala service.
  2. Click the **Configuration** tab.
  3. Select **Scope** > **Impala StateStore**.
  4. Select **Category** > **All**
  5. Select or clear **Enable StateStore Web Server**.
  6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
  7. Restart the Impala service.

- **Impala Catalog Server**

  1. Go to the Impala service.
  2. Click the **Configuration** tab.
  3. Select **Scope** > **Impala Catalog Server**.
  4. Select **Category** > **All**
  5. Check or uncheck **Enable Catalog Server Web Server**.
  6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
  7. Restart the Impala service.

Opening Impala Web Server UIs

- **Impala StateStore**

  1. Go to the Impala service.
  2. Select **Web UI** > **Impala StateStore Web UI**.

- **Impala Daemon**

  1. Go to the Impala service.
  2. Click the **Instances** tab.
  3. Click an **Impala Daemon** instance.
  4. Click **Impala Daemon Web UI**.

- **Impala Catalog Server**

  1. Go to the Impala service.

   2. Select **Web UI** > **Impala Catalog Web UI**.

- **Impala Llama ApplicationMaster**

   1. Go to the Impala service.
   2. Click the **Instances** tab.
   3. Click a **Impala Llama ApplicationMaster** instance.
   4. Click **Llama Web UI**.

Configuring Secure Access for Impala Web Servers

Cloudera Manager supports two methods of authentication for secure access to the Impala Catalog Server, Daemon, and StateStoreweb servers: password-based authentication and TLS/SSL certificate authentication. Both of these can be configured through properties of the Impala Catalog Server, Daemon, and StateStore. Authentication for the three types of daemons can be configured independently.

Configuring Password Authentication

   1. Go to the Impala service.
   2. Click the **Configuration** tab.
   3. Search for "password" using the Search box within the Configuration page. This should display the password-related properties (Username and Password properties) for the Impala Catalog Server, Daemon, and StateStore. If there are multiple role groups configured for Impala Daemon instances, the search should display all of them.
   4. Enter a username and password into these fields.
   5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
   6. Restart the Impala service.

Now when you access the Web UI for the Impala Catalog Server, Daemon, and StateStore, you are asked to log in before access is granted.

Configuring TLS/SSL Certificate Authentication

   1. Create or obtain an TLS/SSL certificate.
   2. Place the certificate, in `.pem` format, on the hosts where the Impala Catalog Server and StateStore are running, and on each host where an Impala Daemon is running. It can be placed in any location (path) you choose. If all the Impala Daemons are members of the same role group, then the `.pem` file must have the same path on every host.
   3. Go to the Impala service page.
   4. Click the **Configuration** tab.
   5. Search for "certificate" using the Search box within the Configuration page. This should display the certificate file location properties for the Impala Catalog Server, Daemon, and StateStore. If there are multiple role groups configured for Impala Daemon instances, the search should display all of them.
   6. In the property fields, enter the full path name to the certificate file.
   7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
   8. Restart the Impala service.

> **Important:** If Cloudera Manager cannot find the `.pem` file on the host for a specific role instance, that role will fail to start.

When you access the Web UI for the Impala Catalog Server, Daemon, and StateStore, `https` will be used.

Modifying Impala Startup Options

The configuration options for the Impala-related daemons let you choose which hosts and ports to use for the services that run on a single host, specify directories for logging, control resource usage and security, and specify other aspects of the Impala software.

Configuring Impala Startup Options through Cloudera Manager

If you manage your cluster through Cloudera Manager, configure the settings for all the Impala-related daemons by navigating to this page: **Clusters** > **Impala** > **Configuration** > **View and Edit**. See the Cloudera Manager documentation for [instructions about how to configure Impala through Cloudera Manager](#).

If the Cloudera Manager interface does not yet have a form field for a newly added option, or if you need to use special options for debugging and troubleshooting, the **Advanced** option page for each daemon includes one or more fields where you can enter option names directly. In Cloudera Manager 4, these fields are labelled **Safety Valve**; in Cloudera Manager 5, they are called **Advanced Configuration Snippet**. There is also a free-form field for query options, on the top-level **Impala Daemon** options page.

Impala Startup Options

Some common settings to change include:

- Memory limits. You can limit the amount of memory available to Impala.

  You can specify the memory limit using absolute notation such as `500m` or `2G`, or as a percentage of physical memory such as `60%`.

  > **Note:** Queries that exceed the specified memory limit are aborted. Percentage limits are based on the physical memory of the machine and do not consider cgroups.

- Core dump enablement. Enable the **Enable Core Dump** setting for the Impala service.

  > **Note:**
  >
  > - The location of core dump files may vary according to your operating system configuration.
  >
  > - Other security settings may prevent Impala from writing core dumps even when this option is enabled.
  >
  > - The default location for core dumps is on a temporary filesystem, which can lead to out-of-space issues if the core dumps are large, frequent, or not removed promptly. To specify an alternative location for the core dumps, filter the Impala configuration settings to find the `core_dump_dir` option. This option lets you specify a different directory for core dumps for each of the Impala-related daemons.

- Authorization using the open source Sentry plugin. See [Enabling Sentry for Impala in Cloudera Manager](#) for details.

- Auditing for successful or blocked Impala queries, another aspect of security. Specify the `-audit_event_log_dir=directory_path` option and optionally the `-max_audit_event_log_file_size=number_of_queries` and `-abort_on_failed_audit_event` options as part of the `IMPALA_SERVER_ARGS` settings, for each Impala node, to enable and customize auditing. See [Auditing Impala Operations](#) for details.

- Password protection for the Impala web UI, which listens on port 25000 by default. This feature involves adding some or all of the `--webserver_password_file`, `--webserver_authentication_domain`, and `--webserver_certificate_file` options to the `IMPALA_SERVER_ARGS` and `IMPALA_STATE_STORE_ARGS` settings. See [Security Guidelines for Impala](#) for details.

- Another setting you might add to `IMPALA_SERVER_ARGS` is a comma-separated list of query options and values:

```
-default_query_options='option=value,option=value,...'
```

  These options control the behavior of queries performed by this `impalad` instance. The option values you specify here override the default values for [Impala query options](#), as shown by the `SET` statement in `impala-shell`.

- During troubleshooting, might direct you to change other values, particularly for `IMPALA_SERVER_ARGS`, to work around issues or gather debugging information.

> **Note:**
>
> These startup options for the `impalad` daemon are different from the command-line options for the `impala-shell` command. For the `impala-shell` options, see impala-shell Configuration Options.

Checking the Values of Impala Configuration Options

You can check the current runtime value of all these settings through the Impala web interface, available by default at `http://impala_hostname:25000/varz` for the `impalad` daemon, `http://impala_hostname:25010/varz` for the `statestored` daemon, or `http://impala_hostname:25020/varz` for the `catalogd` daemon. In the Cloudera Manager interface, you can see the link to the appropriate ***service_name* Web UI** page when you look at the status page for a specific daemon on a specific host.

Startup Options for catalogd Daemon

The `catalogd` daemon implements the Impala catalog service, which broadcasts metadata changes to all the Impala nodes when Impala creates a table, inserts data, or performs other kinds of DDL and DML operations.

Use `--load_catalog_in_background` option to control when the metadata of a table is loaded.

- If set to `false`, the metadata of a table is loaded when it is referenced for the first time. This means that the first run of a particular query can be slower than subsequent runs. Starting in Impala 2.2, the default for `load_catalog_in_background` is `false`.
- If set to `true`, the catalog service attempts to load metadata for a table even if no query needed that metadata. So metadata will possibly be already loaded when the first query that would need it is run. However, for the following reasons, we recommend not to set the option to `true`.

  - Background load can interfere with query-specific metadata loading. This can happen on startup or after invalidating metadata, with a duration depending on the amount of metadata, and can lead to a seemingly random long running queries that are difficult to diagnose.
  - Impala may load metadata for tables that are possibly never used, potentially increasing catalog size and consequently memory usage for both catalog service and Impala Daemon.

Post-Installation Configuration for Impala

This section describes the mandatory and recommended configuration settings for Impala. If Impala is installed using Cloudera Manager, some of these configurations are completed automatically; you must still configure short-circuit reads manually. If you installed Impala without Cloudera Manager, or if you want to customize your environment, consider making the changes described in this topic.

In some cases, depending on the level of Impala, CDH, and Cloudera Manager, you might need to add particular component configuration details in one of the free-form fields on the Impala configuration pages within Cloudera Manager. In Cloudera Manager 4, these fields are labelled **Safety Valve**; in Cloudera Manager 5, they are called **Advanced Configuration Snippet**.

- You must enable short-circuit reads, whether or not Impala was installed through Cloudera Manager. This setting goes in the Impala configuration settings, not the Hadoop-wide settings.
- If you installed Impala in an environment that is not managed by Cloudera Manager, you must enable block location tracking, and you can optionally enable native checksumming for optimal performance.
- If you deployed Impala using Cloudera Manager see Testing Impala Performance to confirm proper configuration.

Mandatory: Short-Circuit Reads

Enabling short-circuit reads allows Impala to read local data directly from the file system. This removes the need to communicate through the DataNodes, improving performance. This setting also minimizes the number of additional

copies of data. Short-circuit reads requires `libhadoop.so` (the Hadoop Native Library) to be accessible to both the server and the client. You must install it from an `.rpm`, `.deb`, or parcel to use short-circuit local reads.

> **Note:** If you use Cloudera Manager, you can enable short-circuit reads through a checkbox in the user interface and that setting takes effect for Impala as well.

To configure DataNodes for short-circuit reads:

1. Copy the client `core-site.xml` and `hdfs-site.xml` configuration files from the Hadoop configuration directory to the Impala configuration directory. The default Impala configuration location is `/etc/impala/conf`.

2. On all Impala nodes, configure the following properties in Impala's copy of `hdfs-site.xml` as shown:

```
<property>
    <name>dfs.client.read.shortcircuit</name>
    <value>true</value>
</property>

<property>
    <name>dfs.domain.socket.path</name>
    <value>/var/run/hdfs-sockets/dn</value>
</property>

<property>
    <name>dfs.client.file-block-storage-locations.timeout.millis</name>
    <value>10000</value>
</property>
```

3. If `/var/run/hadoop-hdfs/` is group-writable, make sure its group is `root`.

> **Note:** If you are also going to enable block location tracking, you can skip copying configuration files and restarting DataNodes and go straight to Optional: Block Location Tracking. Configuring short-circuit reads and block location tracking require the same process of copying files and restarting services, so you can complete that process once when you have completed all configuration changes. Whether you copy files and restart services now or during configuring block location tracking, short-circuit reads are not enabled until you complete those final steps.

4. After applying these changes, restart all DataNodes.

Mandatory: Block Location Tracking

Enabling block location metadata allows Impala to know which disk data blocks are located on, allowing better utilization of the underlying disks. Impala will not start unless this setting is enabled.

To enable block location tracking:

1. For each DataNode, adding the following to the `hdfs-site.xml` file:

```
<property>
  <name>dfs.datanode.hdfs-blocks-metadata.enabled</name>
  <value>true</value>
</property>
```

2. Copy the client `core-site.xml` and `hdfs-site.xml` configuration files from the Hadoop configuration directory to the Impala configuration directory. The default Impala configuration location is `/etc/impala/conf`.

3. After applying these changes, restart all DataNodes.

Optional: Native Checksumming

Enabling native checksumming causes Impala to use an optimized native library for computing checksums, if that library is available.

To enable native checksumming:

If you installed CDH from packages, the native checksumming library is installed and setup correctly. In such a case, no additional steps are required. Conversely, if you installed by other means, native checksumming may not be available due to missing shared objects. Finding the message `"Unable to load native-hadoop library for your platform... using builtin-java classes where applicable"` in the Impala logs indicates native checksumming may be unavailable. To enable native checksumming, you must build and install `libhadoop.so` (the Hadoop Native Library).

Configuring Impala to Work with ODBC

Third-party products can be designed to integrate with Impala using ODBC. For the best experience, ensure any third-party product you intend to use is supported. Verifying support includes checking that the versions of Impala, ODBC, the operating system, and the third-party product have all been approved for use together. Before configuring your systems to use ODBC, download a connector. You may need to sign in and accept license agreements before accessing the pages required for downloading ODBC connectors.

Downloading the ODBC Driver

> **Important:** As of late 2015, most business intelligence applications are certified with the 2.x ODBC drivers. Although the instructions on this page cover both the 2.x and 1.x drivers, expect to use the 2.x drivers exclusively for most ODBC applications connecting to Impala. CDH 6.0 has been tested with the Impala ODBC driver version 2.5.42, and Cloudera recommends that you use this version when you start using CDH 6.0.

See the database drivers section on the [Cloudera downloads web page](#) to download and install the driver.

Configuring the ODBC Port

Versions 2.5 and 2.0 of the Cloudera ODBC Connector, currently certified for some but not all BI applications, use the HiveServer2 protocol, corresponding to Impala port 21050. Impala supports Kerberos authentication with all the supported versions of the driver, and requires ODBC 2.05.13 for Impala or higher for LDAP username/password authentication.

Version 1.x of the Cloudera ODBC Connector uses the original HiveServer1 protocol, corresponding to Impala port 21000.

Example of Setting Up an ODBC Application for Impala

To illustrate the outline of the setup process, here is a transcript of a session to set up all required drivers and a business intelligence application that uses the ODBC driver, under Mac OS X. Each `.dmg` file runs a GUI-based installer, first for the [underlying IODBC driver](#) needed for non-Windows systems, then for the Cloudera ODBC Connector, and finally for the BI tool itself.

```
$ ls -1
Cloudera-ODBC-Driver-for-Impala-Install-Guide.pdf
BI_Tool_Installer.dmg
iodbc-sdk-3.52.7-macosx-10.5.dmg
ClouderaImpalaODBC.dmg
$ open iodbc-sdk-3.52.7-macosx-10.dmg
Install the IODBC driver using its installer
$ open ClouderaImpalaODBC.dmg
Install the Cloudera ODBC Connector using its installer
$ installer_dir=$(pwd)
$ cd /opt/cloudera/impalaodbc
$ ls -1
Cloudera ODBC Driver for Impala Install Guide.pdf
Readme.txt
Setup
lib
ErrorMessages
Release Notes.txt
Tools
$ cd Setup
```

```
$ ls
odbc.ini     odbcinst.ini
$ cp odbc.ini ~/.odbc.ini
$ vi ~/.odbc.ini
$ cat ~/.odbc.ini
[ODBC]
# Specify any global ODBC configuration here such as ODBC tracing.

[ODBC Data Sources]
Sample Cloudera Impala DSN=Cloudera ODBC Driver for Impala

[Sample Cloudera Impala DSN]

# Description: DSN Description.
# This key is not necessary and is only to give a description of the data source.
Description=Cloudera ODBC Driver for Impala DSN

# Driver: The location where the ODBC driver is installed to.
Driver=/opt/cloudera/impalaodbc/lib/universal/libclouderaimpalaodbc.dylib

# The DriverUnicodeEncoding setting is only used for SimbaDM
# When set to 1, SimbaDM runs in UTF-16 mode.
# When set to 2, SimbaDM runs in UTF-8 mode.
#DriverUnicodeEncoding=2

# Values for HOST, PORT, KrbFQDN, and KrbServiceName should be set here.
# They can also be specified on the connection string.
HOST=hostname.sample.example.com
PORT=21050
Schema=default

# The authentication mechanism.
# 0 - No authentication (NOSASL)
# 1 - Kerberos authentication (SASL)
# 2 - Username authentication (SASL)
# 3 - Username/password authentication (SASL)
# 4 - Username/password authentication with SSL (SASL)
# 5 - No authentication with SSL (NOSASL)
# 6 - Username/password authentication (NOSASL)
AuthMech=0

# Kerberos related settings.
KrbFQDN=
KrbRealm=
KrbServiceName=

# Username/password authentication with SSL settings.
UID=
PWD
CAIssuedCertNamesMismatch=1
TrustedCerts=/opt/cloudera/impalaodbc/lib/universal/cacerts.pem

# Specify the proxy user ID to use.
#DelegationUID=

# General settings
TSaslTransportBufSize=1000
RowsFetchedPerBlock=10000
SocketTimeout=0
StringColumnLength=32767
UseNativeQuery=0
$ pwd
/opt/cloudera/impalaodbc/Setup
$ cd $installer_dir
$ open BI_Tool_Installer.dmg
Install the BI tool using its installer
$ ls /Applications | grep BI_Tool
BI_Tool.app
$ open -a BI_Tool.app
In the BI tool, connect to a data source using port 21050
```

Notes about JDBC and ODBC Interaction with Impala SQL Features

Most Impala SQL features work equivalently through the `impala-shell` interpreter of the JDBC or ODBC APIs. The following are some exceptions to keep in mind when switching between the interactive shell and applications using the APIs:

> **Note:** If your JDBC or ODBC application connects to Impala through a load balancer such as `haproxy`, be cautious about reusing the connections. If the load balancer has set up connection timeout values, either check the connection frequently so that it never sits idle longer than the load balancer timeout value, or check the connection validity before using it and create a new one if the connection has been closed.

Configuring Impala to Work with JDBC

Impala supports the standard JDBC interface, allowing access from commercial Business Intelligence tools and custom software written in Java or other programming languages. The JDBC driver allows you to access Impala from a Java program that you write, or a Business Intelligence or similar tool that uses JDBC to communicate with various database products.

Setting up a JDBC connection to Impala involves the following steps:

- Verifying the communication port where the Impala daemons in your cluster are listening for incoming JDBC requests.
- Installing the JDBC driver on every system that runs the JDBC-enabled application.
- Specifying a connection string for the JDBC application to access one of the servers running the `impalad` daemon, with the appropriate security settings.

Configuring the JDBC Port

The default port used by JDBC 2.0 and later (as well as ODBC 2.x) is 21050. Impala server accepts JDBC connections through this same port 21050 by default. Make sure this port is available for communication with other hosts on your network, for example, that it is not blocked by firewall software. If your JDBC client software connects to a different port, specify that alternative port number with the `--hs2_port` option when starting `impalad`. See Impala Startup Options on page 128 for details about Impala startup options. See Ports Used by Impala for information about all ports used for communication between Impala and clients or between Impala components.

Choosing the JDBC Driver

In Impala 2.0 and later, you have the choice between the Cloudera JDBC Connector and the Hive 0.13 JDBC driver. Cloudera recommends using the Cloudera JDBC Connector where practical.

If you are already using JDBC applications with an earlier Impala release, you must update your JDBC driver to one of these choices, because the Hive 0.12 driver that was formerly the only choice is not compatible with Impala 2.0 and later.

Both the Cloudera JDBC 2.5 Connector and the Hive JDBC driver provide a substantial speed increase for JDBC applications with Impala 2.0 and higher, for queries that return large result sets.

> **Important:** CDH 6.0 has been tested using the Impala JDBC driver versions 2.5.45 and 2.6.2. Cloudera recommends that you use one of these two versions when you start using Impala in CDH 6.0.

Enabling Impala JDBC Support on Client Systems

Using the Cloudera JDBC Connector (recommended)

Download and install the Cloudera JDBC connector on any Linux, Windows, or Mac system where you intend to run JDBC-enabled applications. From the Cloudera Downloads page, navigate to the **Database Drivers** section of the page and choose the appropriate protocol (JDBC or ODBC) and target product (Impala or Hive). The ease of downloading and installing on a wide variety of systems makes this connector a convenient choice for organizations with heterogeneous environments. This is the download pagefor the **Impala JDBC Connector**.

Using the Hive JDBC Driver

You install the Hive JDBC driver (`hive-jdbc` package) through the Linux package manager, on hosts within the cluster. The driver consists of several Java JAR files. The same driver can be used by Impala and Hive.

To get the JAR files, install the Hive JDBC driver on each host in the cluster that will run JDBC applications. Follow the instructions for [Installing Cloudera JDBC and ODBC Drivers on Clients in CDH](#).

> **Note:** The latest JDBC driver, corresponding to Hive 0.13, provides substantial performance improvements for Impala queries that return large result sets. Impala 2.0 and later are compatible with the Hive 0.13 driver. If you already have an older JDBC driver installed, and are running Impala 2.0 or higher, consider upgrading to the latest Hive JDBC driver for best performance with JDBC applications.

If you are using JDBC-enabled applications on hosts outside the CDH cluster, you cannot use the CDH install procedure on the non-CDH hosts. Install the JDBC driver on at least one CDH host using the preceding procedure. Then download the JAR files to each client machine that will use JDBC with Impala:

```
commons-logging-X.X.X.jar
  hadoop-common.jar
  hive-common-X.XX.X-cdhX.X.X.jar
  hive-jdbc-X.XX.X-cdhX.X.X.jar
  hive-metastore-X.XX.X-cdhX.X.X.jar
  hive-service-X.XX.X-cdhX.X.X.jar
  httpclient-X.X.X.jar
  httpcore-X.X.X.jar
  libfb303-X.X.X.jar
  libthrift-X.X.X.jar
  log4j-X.X.XX.jar
  slf4j-api-X.X.X.jar
  slf4j-logXjXX-X.X.X.jar
```

To enable JDBC support for Impala on the system where you run the JDBC application:

1. Download the JAR files listed above to each client machine.

   > **Note:** For Maven users, see [this sample github page](#) for an example of the dependencies you could add to a `pom` file instead of downloading the individual JARs.

2. Store the JAR files in a location of your choosing, ideally a directory already referenced in your `CLASSPATH` setting. For example:

   - On Linux, you might use a location such as `/opt/jars/`.
   - On Windows, you might use a subdirectory underneath `C:\Program Files`.

3. To successfully load the Impala JDBC driver, client programs must be able to locate the associated JAR files. This often means setting the `CLASSPATH` for the client process to include the JARs. Consult the documentation for your JDBC client for more details on how to install new JDBC drivers, but some examples of how to set `CLASSPATH` variables include:

   - On Linux, if you extracted the JARs to `/opt/jars/`, you might issue the following command to prepend the JAR files path to an existing classpath:

```
export CLASSPATH=/opt/jars/*.jar:$CLASSPATH
```

   - On Windows, use the **System Properties** control panel item to modify the **Environment Variables** for your system. Modify the environment variables to include the path to which you extracted the files.

> **Note:** If the existing `CLASSPATH` on your client machine refers to some older version of the Hive JARs, ensure that the new JARs are the first ones listed. Either put the new JAR files earlier in the listings, or delete the other references to Hive JAR files.

Establishing JDBC Connections

The JDBC driver class depends on which driver you select.

> **Note:** If your JDBC or ODBC application connects to Impala through a load balancer such as `haproxy`, be cautious about reusing the connections. If the load balancer has set up connection timeout values, either check the connection frequently so that it never sits idle longer than the load balancer timeout value, or check the connection validity before using it and create a new one if the connection has been closed.

Using the Cloudera JDBC Connector (recommended)

Depending on the level of the JDBC API your application is targeting, you can use the following fully-qualified class names (FQCNs):

- `com.cloudera.impala.jdbc41.Driver`
- `com.cloudera.impala.jdbc41.DataSource`

- `com.cloudera.impala.jdbc4.Driver`
- `com.cloudera.impala.jdbc4.DataSource`

- `com.cloudera.impala.jdbc3.Driver`
- `com.cloudera.impala.jdbc3.DataSource`

The connection string has the following format:

```
jdbc:impala://Host:Port[/Schema];Property1=Value;Property2=Value;...
```

The `port` value is typically 21050 for Impala.

For full details about the classes and the connection string (especially the property values available for the connection string), download the appropriate driver documentation for your platform from the Impala JDBC Connector download page.

Using the Hive JDBC Driver

For example, with the Hive JDBC driver, the class name is `org.apache.hive.jdbc.HiveDriver`. Once you have configured Impala to work with JDBC, you can establish connections between the two. To do so for a cluster that does not use Kerberos authentication, use a connection string of the form `jdbc:hive2://host:port/;auth=noSasl`. For example, you might use:

```
jdbc:hive2://myhost.example.com:21050/;auth=noSasl
```

To connect to an instance of Impala that requires Kerberos authentication, use a connection string of the form `jdbc:hive2://host:port/;principal=principal_name`. The principal must be the same user principal you used when starting Impala. For example, you might use:

```
jdbc:hive2://myhost.example.com:21050/;principal=impala/myhost.example.com@H2.EXAMPLE.COM
```

To connect to an instance of Impala that requires LDAP authentication, use a connection string of the form
`jdbc:hive2://`*`host`*`:`*`port`*`/`*`db_name`*`;user=`*`ldap_userid`*`;password=`*`ldap_password`*. For example, you might
use:

```
jdbc:hive2://myhost.example.com:21050/test_db;user=fred;password=xyz123
```

> **Note:**
>
> Prior to CDH 5.7 / Impala 2.5, the Hive JDBC driver did not support connections that use both Kerberos
> authentication and SSL encryption. If your cluster is running an older release that has this restriction,
> to use both of these security features with Impala through a JDBC application, use the Cloudera JDBC
> Connector as the JDBC driver.

Notes about JDBC and ODBC Interaction with Impala SQL Features

Most Impala SQL features work equivalently through the `impala-shell` interpreter of the JDBC or ODBC APIs. The
following are some exceptions to keep in mind when switching between the interactive shell and applications using
the APIs:

- **Complex type considerations:**

    - Queries involving the complex types (`ARRAY`, `STRUCT`, and `MAP`) require notation that might not be available
      in all levels of JDBC and ODBC drivers. If you have trouble querying such a table due to the driver level or
      inability to edit the queries used by the application, you can create a view that exposes a "flattened" version
      of the complex columns and point the application at the view. See Complex Types (or higher only) for details.

    - The complex types available in and higher are supported by the JDBC `getColumns()` API. Both `MAP` and
      `ARRAY` are reported as the JDBC SQL Type `ARRAY`, because this is the closest matching Java SQL type. This
      behavior is consistent with Hive. `STRUCT` types are reported as the JDBC SQL Type `STRUCT`.

      To be consistent with Hive's behavior, the TYPE_NAME field is populated with the primitive type name for
      scalar types, and with the full `toSql()` for complex types. The resulting type names are somewhat inconsistent,
      because nested types are printed differently than top-level types. For example, the following list shows how
      `toSQL()` for Impala types are translated to `TYPE_NAME` values:

```
DECIMAL(10,10)        becomes   DECIMAL
CHAR(10)              becomes   CHAR
VARCHAR(10)           becomes   VARCHAR
ARRAY<DECIMAL(10,10)> becomes   ARRAY<DECIMAL(10,10)>
ARRAY<CHAR(10)>       becomes   ARRAY<CHAR(10)>
ARRAY<VARCHAR(10)>    becomes   ARRAY<VARCHAR(10)>
```

Kudu Considerations for DML Statements

Currently, Impala `INSERT`, `UPDATE`, or other DML statements issued through the JDBC interface against a Kudu table
do not return JDBC error codes for conditions such as duplicate primary key columns. Therefore, for applications that
issue a high volume of DML statements, prefer to use the Kudu Java API directly rather than a JDBC application.

Managing Key-Value Store Indexer

The Key-Value Store Indexer service uses the Lily HBase Indexer Service to index the stream of records being added to
HBase tables. Indexing allows you to query data stored in HBase with the Solr service.

The Key-Value Store Indexer service is installed in the same parcel or package along with the CDH or Solr service.

Adding the Key-Value Store Indexer Service

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the cluster name and select **Add a Service**. A list of service types display. You can add one type of service at a time.

2. Select the **Key-Value Store Indexer** service and click **Continue**.

3. Select the services on which the new service should depend. All services must depend on the *same* ZooKeeper service. Click **Continue**.

4. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

   Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

   The following shortcuts for specifying hostname patterns are supported:

   - Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   - IP addresses
   - Rack name

   Click the **View By Host** button for an overview of the role assignment by hostname ranges.

5. Click **Continue**.

6. Review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. If you chose to add the Sqoop service, indicate whether to use the default Derby database or the embedded PostgreSQL database. If the latter, type the database name, host, and user credentials that you specified when you created the database.

   > ⚠ **Warning:** Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which will result in reports of missing blocks.

7. Click **Continue**.

8. Click **Finish**.

Enabling Morphlines with Search and HBase Indexing

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Cloudera Morphlines is an open source framework that reduces the time and skills necessary to build or change Search indexing applications. A morphline is a rich configuration file that simplifies defining an ETL transformation chain.

1. Go to the Indexer service.
2. Click the **Configuration** tab.
3. Select **Scope** > **All**.
4. Select **Category** > **Morphlines**.
5. Create the necessary configuration files, and modify the content in the following properties:

- **Morphlines File** — Text that goes into the `morphlines.conf` used by HBase indexers. You should use `$ZK_HOST` in this file instead of specifying a ZooKeeper quorum. Cloudera Manager automatically replaces the `$ZK_HOST` variable with the correct value during the Solr configuration deployment.
- **Custom MIME-types File** — Text that goes verbatim into the `custom-mimetypes.xml` file used by HBase Indexers with the `detectMimeTypes` command. See the [Cloudera Morphlines Reference Guide](#) for details on this command.
- **Grok Dictionary File** — Text that goes verbatim into the `grok-dictionary.conf` file used by HBase Indexers with the `grok` command. See the [Cloudera Morphlines Reference Guide](#) for details of this command.

See [Extracting, Transforming, and Loading Data With Cloudera Morphlines](#) for information about using morphlines with Search and HBase.

Managing Kudu

This topic describes the tasks you can perform to manage the Kudu service using Cloudera Manager. You can use Cloudera Manager to upgrade the Kudu service, start and stop the Kudu service, monitor operations, and configure the Kudu master and tablet servers, among other tasks. Depending on your deployment, there are several different configuration settings you may need to modify.

For detailed information about Apache Kudu, view the [Apache Kudu Guide](#).

Installing and Upgrading the Kudu Service

You can install Kudu through the Cloudera Manager installation wizard, using either parcels or packages. For instructions, see [Cloudera Installation Guide](#).

Enabling Core Dump for the Kudu Service

If Kudu crashes, you can use Cloudera Manager to generate a core dump to get more information about the crash.

1. Go to the Kudu service.
2. Click the **Configuration** tab.
3. Search for `core dump`.
4. Check the checkbox for the **Enable Core Dump** property.
5. (Optional) Unless otherwise configured, the dump file is generated in the default core dump directory, `/var/log/kudu`, for both the Kudu master and the tablet servers.

   - To configure a different dump directory for the Kudu master, modify the value of the **Kudu Master Core Dump Directory** property.
   - To configure a different dump directory for the Kudu tablet servers, modify the value of the **Kudu Tablet Server Core Dump Directory** property.

6. Click **Save Changes**.

Verifying the Impala Dependency on Kudu

In a Cloudera Manager deployment, once the Kudu service is installed, Impala will automatically identify the Kudu Master. However, if your Impala queries don't work as expected, use the following steps to make sure that the Impala service is set to be dependent on Kudu.

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Search for `kudu`.
4. Make sure the **Kudu Service** property is set to the right Kudu service.
5. Click **Save Changes**.

Using the Charts Library with the Kudu Service

By default, the **Status** tab for the Kudu service displays a dashboard containing a limited set of charts. For details on the terminology used in these charts, and instructions on how to query for time-series data, display chart details, and edit charts, see Charting Time-Series Data on page 303.

The Kudu service's Charts Library tab also displays a dashboard containing a much larger set of charts, organized by categories such as process charts, host charts, CPU charts, and so on, depending on the entity (service, role, or host) that you are viewing. You can use these charts to keep track of disk space usage, the rate at which data is being inserted/modified in Kudu across all tables, or any critical cluster events. You can also use them to keep track of individual tables. For example, to find out how much space a Kudu table is using on disk:

1. Go to the Kudu service and navigate to the **Charts Library** tab.
2. On the left-hand side menu, click **Tables** to display the list of tables currently stored in Kudu.
3. Click on a table name to view the default dashboard for that table. The **Total Tablet Size On Disk Across Kudu Replicas** chart displays the total size of the table on disk using a time-series chart.

   Hovering with your mouse over the line on the chart opens a small pop-up window that displays information about that data point. Click the data stream within the chart to display a larger pop-up window that includes additional information for the table at the point in time where the mouse was clicked.

Managing Oozie

This section describes tasks used for managing Oozie.

Managing Solr

You can install the Solr service through the Cloudera Manager installation wizard, using either parcels or packages. See Cloudera Installation Guide.

You can elect to have the service created and started as part of the Installation wizard. If you elect not to create the service using the Installation wizard, you can use the **Add Service** wizard to perform the installation. The wizard will automatically configure and start the dependent services and the Solr service. See Adding a Service on page 193 for instructions.

For further information on the Solr service, see Search Guide.

The following sections describe how to configure other CDH components to work with the Solr service.

Configuring the Flume Morphline Solr Sink for Use with the Solr Service

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

To use a Flume Morphline Solr sink, the Flume service must be running on your cluster. See the Flume Near Real-Time Indexing Reference for information about the Flume Morphline Solr Sink and Configuring Apache Flume.

1. Go to the Flume service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Agent**
4. Select **Category** > **Flume-NG Solr Sink**.
5. Edit the following settings, which are templates that you must modify for your deployment:

   - **Morphlines File** (`morphlines.conf`) - Configures Morphlines for Flume agents. You must use `$ZK_HOST` in this field instead of specifying a ZooKeeper quorum. Cloudera Manager automatically replaces the `$ZK_HOST` variable with the correct value during the Flume configuration deployment.
   - **Custom MIME-types File** (`custom-mimetypes.xml`) - Configuration for the `detectMimeTypes` command. See the Cloudera Morphlines Reference Guide for details on this command.
   - **Grok Dictionary File** (`grok-dictionary.conf`) - Configuration for the `grok` command. See the Cloudera Morphlines Reference Guide for details on this command.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

Once configuration is complete, Cloudera Manager automatically deploys the required files to the Flume agent process directory when it starts the Flume agent. Therefore, you can reference the files in the Flume agent configuration using their relative path names. For example, you can use the name `morphlines.conf` to refer to the location of the Morphlines configuration file.

Using a Load Balancer with Solr

To configure a load balancer:

1. Go to the Solr service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Solr**
4. Select **Category** > **All**
5. Enter the hostname and port number of the load balancer in the **Solr Load Balancer** property in the format *hostname:port number*.

   > **Note:**
   >
   > When you set this property, Cloudera Manager regenerates the keytabs for Solr roles. The principal in these keytabs contains the load balancer hostname.
   >
   > If there is a Hue service that depends on this Solr service, it also uses the load balancer to communicate with Solr.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Migrating Solr Replicas

When you replace a host, migrating replicas on that host to the new host, instead of depending on failure recovery, can help ensure optimal performance.

Where possible, the Solr service routes requests to the proper host. Both `ADDREPLICA` and `DELETEREPLICA` Collections API calls can be sent to any host in the cluster. For more information on the Collections API, see the Collections API section of Apache Solr Reference Guide 4.10 (PDF).

- For adding replicas, the `node` parameter ensures the new replica is created on the intended host. If no host is specified, Solr selects a host with relatively fewer replicas.
- For deleting replicas, the request is routed to the host that hosts the replica to be deleted.

Adding replicas can be resource intensive. For best results, add replicas when the system is not under heavy load. For example, do not add replicas when heavy indexing is occurring or when MapReduceIndexerTool jobs are running.

Cloudera recommends using API calls to create and unload cores. Do not use the Cloudera Manager Admin Console or the Solr Admin UI for these tasks.

This procedure uses the following names:

- Host names:

  - Origin: `solr01.example.com`.
  - Destination: `solr02.example.com`.

- Collection name: `email`
- Replicas:

  - The original replica `email_shard1_replica1`, which is on `solr01.example.com`.
  - The new replica `email_shard1_replica2`, which will be on `solr02.example.com`.

To migrate a replica to a new host:

1. (Optional) If you want to add a replica to a particular node, review the contents of the `live_nodes` directory on ZooKeeper to find all nodes available to host replicas. Open the Solr Administration User interface, click **Cloud**, click **Tree**, and expand **live_nodes**. The Solr Administration User Interface, including **live_nodes**, might appear as follows:
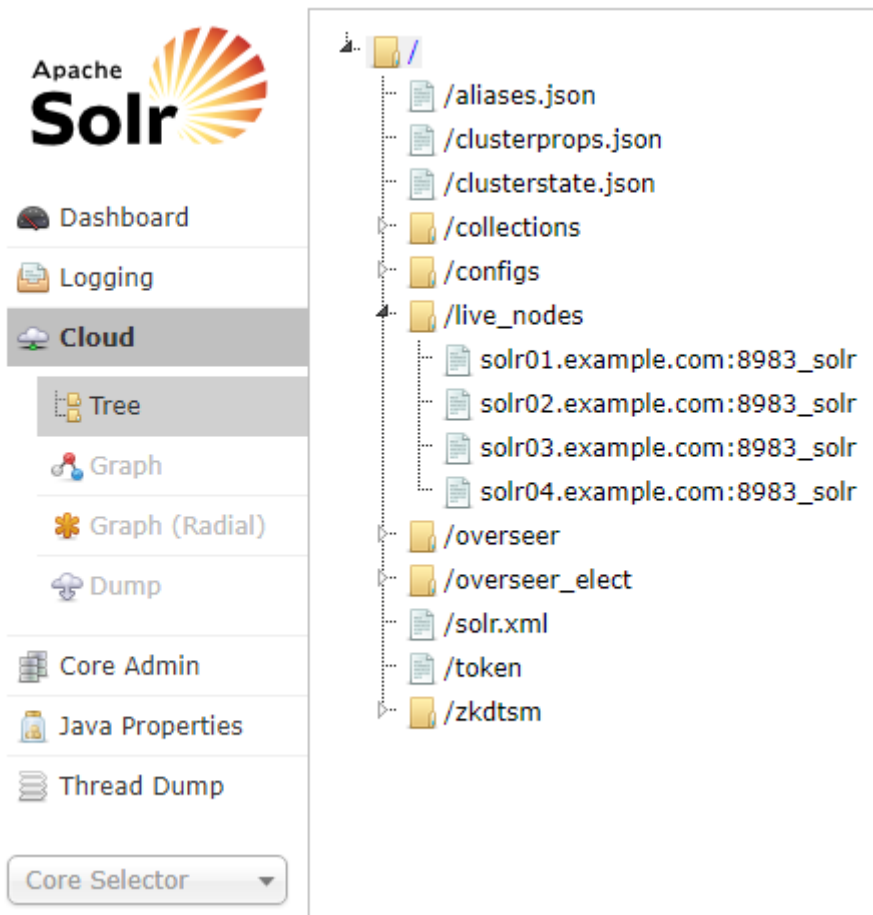
> **Note:** Information about Solr nodes can also be found in `clusterstate.json`, but that file only lists nodes currently hosting replicas. Nodes running Solr but not currently hosting replicas are not listed in `clusterstate.json`.

2. Add the new replica on `solr02.example.com` using the `ADDREPLICA` API call.

```
http://solr01.example.com:8983/solr/admin/collections?action=ADDREPLICA&collection=email&shard=shard1&node=solr02.example.com:8983_solr
```

3. Verify that the replica creation succeeds and moves from recovery state to **ACTIVE**. You can check the replica status in the Cloud view, which can be found at a URL similar to:
    `http://solr02.example.com:8983/solr/#/~cloud`.

> **Note:** Do not delete the original replica until the new one is in the **ACTIVE** state. When the newly added replica is listed as **ACTIVE**, the index has been fully replicated to the newly added replica. The total time to replicate an index varies according to factors such as network bandwidth and the size of the index. Replication times on the scale of hours are not uncommon and do not necessarily indicate a problem.
>
> You can use the `details` command to get an XML document that contains information about replication progress. Use `curl` or a browser to access a URI similar to:
>
> ```
> http://solr02.example.com:8983/solr/email_shard1_replica2/replication?command=details
> ```
>
> Accessing this URI returns an XML document that contains content about replication progress. A snippet of the XML content might appear as follows:
>
> ```
> ...
> <str name="numFilesDownloaded">126</str>
> <str name="replication StartTime">Tue Jan 21 14:34:43 PST 2014</str>
> <str name="timeElapsed">457s</str>
> <str name="currentFile">4xt_Lucene41_0.pos</str>
> <str name="currentFileSize">975.17 MB</str>
> <str name="currentFileSizeDownloaded">545 MB</str>
> <str name="currentFileSizePercent">55.0</str>
> <str name="bytesDownloaded">8.16 GB</str>
> <str name="totalPercent">73.0</str>
> <str name="timeRemaining">166s</str>
> <str name="downloadSpeed">18.29 MB</str>
> ...
> ```

4. Use the `CLUSTERSTATUS` API call to retrieve information about the cluster, including current cluster status:

```
http://solr01.example.com:8983/solr/admin/collections?action=clusterstatus&wt=json&indent=true
```

Review the returned information to find the correct replica to remove. An example of the JSON file might appear as follows:
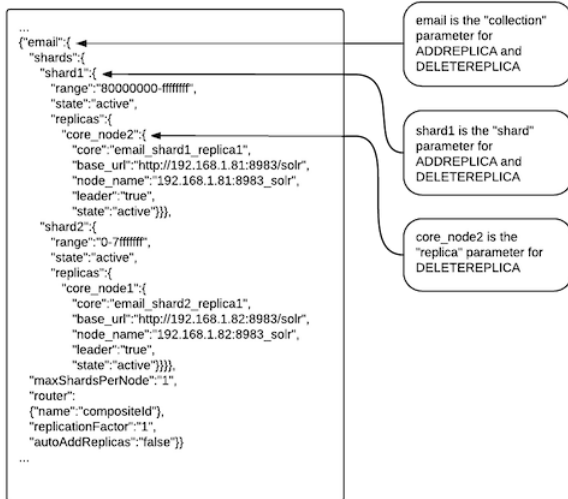
5. Delete the old replica on `solr01.example.com` server using the `DELETEREPLICA` API call:

```
http://solr01.example.com:8983/solr/admin/collections?action=DELETEREPLICA&collection=email&shard=shard1&replica=core_node2
```

The `DELETEREPLICA` call removes the `datadir`.

Managing Spark

Apache Spark is a general framework for distributed computing that offers high performance for both batch and interactive processing.

To run applications distributed across a cluster, Spark requires a cluster manager. In CDH 6, Cloudera supports only the YARN cluster manager. When run on YARN, Spark application processes are managed by the YARN ResourceManager and NodeManager roles. Spark Standalone is no longer supported.

In CDH 6, Cloudera only supports running Spark applications on a YARN cluster manager. The Spark Standalone cluster manager is not supported.

Related Information

- Spark Guide
- Monitoring Spark Applications on page 279
- Tuning Apache Spark Applications on page 351
- Spark Authentication
- Cloudera Spark forum
- Apache Spark documentation

This section describes how to manage Spark services.

Managing Spark Using Cloudera Manager

You can install, add, and start Spark through the Cloudera Manager Installation wizard using parcels. For more information, see Cloudera Installation Guide.

If you do not add the Spark service using the Installation wizard, you can use the **Add Service** wizard to create the service. The wizard automatically configures dependent services and the Spark service. For instructions, see Adding a Service on page 193.

For information on Spark applications, see Spark Application Overview.

How Spark Configurations are Propagated to Spark Clients

Because the Spark service does not have worker roles, another mechanism is needed to enable the propagation of client configurations to the other hosts in your cluster. In Cloudera Manager gateway roles fulfill this function. Whether you add a Spark service at installation time or at a later time, ensure that you assign the gateway roles to hosts in the cluster. If you do not have gateway roles, client configurations are not deployed.

Managing the Spark History Server

The Spark History Server displays information about the history of completed Spark applications. For further information, see Monitoring Spark Applications on page 279.

For instructions for configuring the Spark History Server to use Kerberos, see Spark Authentication.

Adding the Spark History Server Using Cloudera Manager

By default, the Spark (Standalone) service does not include a History Server. To configure applications to store history, on Spark clients, set `spark.eventLog.enabled` to true before starting the application.

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

To add the History Server:

1. Go to the Spark service.
2. Click the **Instances** tab.
3. Click the **Add Role Instances** button.
4. Select a host in the column under **History Server**, and then click **OK**.
5. Click **Continue**.
6. Check the checkbox next to the History Server role.

7. Select **Actions for Selected** > **Start** and click **Start**.
8. Click **Close** when the action completes.

Managing the Sqoop 1 Client

To add Sqoop 1 to your cluster, add the **Sqoop 1 Client** service and a Sqoop 1 gateway and deploy the client configuration:

Adding the Sqoop 1 Client

Minimum Required Role: Full Administrator

The Sqoop 1 client packages are installed by the Installation wizard. However, the client configuration is not deployed. To create a Sqoop 1 gateway and deploy the client configuration:

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the cluster name and select **Add a Service**. A list of service types display. You can add one type of service at a time.
2. Select the **Sqoop 1 Client** service and click **Continue**.
3. Select the services on which the new service should depend. All services must depend on the *same* ZooKeeper service. Click **Continue**.
4. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

   Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

   The following shortcuts for specifying hostname patterns are supported:

   • Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

   • IP addresses
   • Rack name

   Click the **View By Host** button for an overview of the role assignment by hostname ranges.

5. Click **Continue**. The client configuration deployment command runs.
6. Click **Continue** and click **Finish**.

Installing the JDBC Drivers for Sqoop 1

Sqoop 1 does not ship with third party JDBC drivers. You must download them separately and save them to the `/var/lib/sqoop/` directory on the server. Ensure that you do not save JARs in the CDH parcel directory `/opt/cloudera/parcels/CDH`, because this directory is overwritten when you upgrade CDH.

The following sections show how to install the most common JDBC drivers.

> **Note:**
> - The JDBC drivers need to be installed only on the machine where Sqoop runs; you do not need to install them on all hosts in your Hadoop cluster.
> - Kerberos authentication is not supported by the Sqoop Connector for Teradata.
> - Use the JDBC driver jar that your database server and java version support.

Before you begin:

Make sure the `/var/lib/sqoop` directory exists and has the correct ownership and permissions:

```
mkdir -p /var/lib/sqoop
chown sqoop:sqoop /var/lib/sqoop
chmod 755 /var/lib/sqoop
```

This sets permissions to `drwxr-xr-x`.

For JDBC drivers for Hive, Impala, Teradata, or Netezza, see the [Connectors documentation.](#)

CDP does not support the Sqoop exports using the Hadoop jar command (the Java API). The connector documentation from Teradata includes instructions that include the use of this API. CDP users have reportedly mistaken the unsupported API commands, such as `-forcestage`, for the supported Sqoop commands, such as `--staging-force`. Cloudera supports the use of Sqoop only with commands documented in [Using the Cloudera Connector Powered by Teradata](#). Cloudera does not support using Sqoop with Hadoop jar commands, such as those described in the Teradata Connector for Hadoop Tutorial.

Installing the MySQL JDBC Driver

Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html. You will need to sign up for an account if you do not already have one, and log in, before you can download it. Then copy it to the `/var/lib/sqoop/` directory. For example:

```
$ sudo cp mysql-connector-java-version/mysql-connector-java-version-bin.jar
/var/lib/sqoop/
```

> **Note:**
> At the time of publication, *version* was `5.1.31`, but the version may have changed by the time you read this.
>
> > **Important:**
> > Make sure you have at least version 5.1.31. Some systems ship with an earlier version that may not work correctly with Sqoop.

Installing the Oracle JDBC Driver

You can download the JDBC Driver from the Oracle website, for example http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html. You must accept the license agreement before you can download the driver. Download the `ojdbc6.jar` file and copy it to the `/var/lib/sqoop/` directory:

```
$ sudo cp ojdbc6.jar /var/lib/sqoop/
```

Installing the Microsoft SQL Server JDBC Driver

Download the Microsoft SQL Server JDBC driver from
http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774 and copy it to the
/var/lib/sqoop/ directory. For example:

```
$ curl -L
'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz'
 | tar xz
$ sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop/
```

Installing the PostgreSQL JDBC Driver

Download the PostgreSQL JDBC driver from http://jdbc.postgresql.org/download.html and copy it to the
/var/lib/sqoop/ directory. For example:

```
$ curl -L 'http://jdbc.postgresql.org/download/postgresql-9.2-1002.jdbc4.jar' -o
postgresql-9.2-1002.jdbc4.jar
$ sudo cp postgresql-9.2-1002.jdbc4.jar /var/lib/sqoop/
```

Syntax for Configuring JDBC Connection Strings

These are the JDBC connection strings for supported databases.

MySql Connection String

Syntax:

```
jdbc:mysql://<HOST>:<PORT>/<DATABASE_NAME>
```

Example:

```
jdbc:mysql://my_mysql_server_hostname:3306/my_database_name
```

Oracle Connection String

Syntax:

```
jdbc:oracle:thin:@<HOST>:<PORT>:<DATABASE_NAME>
```

Example:

```
jdbc:oracle:thin:@my_oracle_server_hostname:1521:my_database_name
```

PostgreSQL Connection String

Syntax:

jdbc:postgresql://<HOST>:<PORT>/<DATABASE_NAME>

Example:

```
jdbc:postgresql://my_postgres_server_hostname:5432/my_database_name
```

Netezza Connection String

Syntax:

jdbc:netezza://<HOST>:<PORT>/<DATABASE_NAME>

Example:

```
jdbc:netezza://my_netezza_server_hostname:5480/my_database_name
```

Teradata Connection String

> **Note:** Kerberos authentication is not supported by the Sqoop Connector for Teradata.

CDP does not support the Sqoop exports using the Hadoop jar command (the Java API). The connector documentation from Teradata includes instructions that include the use of this API. CDP users have reportedly mistaken the unsupported API commands, such as -forcestage, for the supported Sqoop commands, such as –-staging-force. Cloudera supports the use of Sqoop only with commands documented in Using the Cloudera Connector Powered by Teradata. Cloudera does not support using Sqoop with Hadoop JAR commands, such as those described in the Teradata Connector for Hadoop Tutorial.

Syntax:

```
jdbc:teradata://<HOST>/DBS_PORT=1025/DATABASE=<DATABASE_NAME>
```

Example:

```
jdbc:teradata://my_teradata_server_hostname/DBS_PORT=1025/DATABASE=my_database_name
```

Setting HADOOP_MAPRED_HOME for Sqoop 1

- For each user who will be submitting MapReduce jobs using MapReduce v2 (YARN), or running Pig, Hive, or Sqoop 1 in a YARN installation, make sure that the HADOOP_MAPRED_HOME environment variable is set correctly, as follows:

```
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
```

Managing YARN (MRv2) and MapReduce (MRv1)

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

CDH supports two versions of the MapReduce computation framework: MRv1 and MRv2, which are implemented by the MapReduce (MRv1) and YARN (MRv2) services. YARN is backwards-compatible with MapReduce. (All jobs that run against MapReduce also run in a YARN cluster).

The MapReduce v2 (MRv2) or YARN architecture splits the two primary responsibilities of the JobTracker — resource management and job scheduling/monitoring — into separate daemons: a global ResourceManager and per-application ApplicationMasters. With YARN, the ResourceManager and per-host NodeManagers form the data-computation framework. The ResourceManager service effectively replaces the functions of the JobTracker, and NodeManagers run on worker hosts instead of TaskTracker daemons. The per-application ApplicationMaster is, in effect, a framework-specific library and negotiates resources from the ResourceManager and works with the NodeManagers to run and monitor the tasks. For details of this architecture, see Apache Hadoop NextGen MapReduce (YARN).

- The Cloudera Manager Admin Console has different methods for displaying MapReduce and YARN job history. See Monitoring MapReduce Jobs on page 249 and Monitoring YARN Applications on page 266.
- For information on configuring the MapReduce and YARN services for high availability, see MapReduce (MRv1) and YARN (MRv2) High Availability on page 431.
- For information on configuring MapReduce and YARN resource management features, see Resource Management on page 367.

Defaults and Recommendations

- In a Cloudera Manager deployment of a CDH cluster, the YARN service is the default MapReduce computation framework.In CDH 5, the MapReduce 1 service has been deprecated. However, the MapReduce service is fully supported for backward compatibility through the CDH 5 lifecycle.
- For production uses, Cloudera recommends that *only one* MapReduce framework should be running at any given time. If development needs or other use case requires switching between MapReduce and YARN, both services can be configured at the same time, but only one should be running (to fully optimize the hardware resources available).

Migrating from MapReduce to YARN

Cloudera Manager provides a wizard described in [Importing MapReduce Configurations to YARN](#) on page 152 to easily migrate MapReduce configurations to YARN. The wizard performs all the steps ([Switching Between MapReduce and YARN Services](#) on page 148, [Updating Services Dependent on MapReduce](#) on page 149, and [Configuring Alternatives Priority for Services Dependent on MapReduce](#) on page 149) on this page.

The Activity Monitor role collects information about activities run by the MapReduce service. If MapReduce is not being used and the reporting data is no longer required, then the Activity Monitor role and database can be removed:

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Instances** tab.
3. Select checkbox for Activity Monitor, select **Actions for Selected** > **Stop**, and click **Stop** to confirm.
4. Select checkbox for Activity Monitor, select **Actions for Selected** > **Delete**, and click **Delete** to confirm.
5. Manage the Activity Monitor database. The example below is for a MySQL backend database:

   a. Verify the Activity Monitor database:

```
mysql> show databases;
+-------------------+
| Database          |
+-------------------+
| amon              |
+-------------------+
```

   b. Back up the database:

```
$ mysqldump -uroot -pcloudera amon > /safe_backup_directory/amon.sql
```

   Drop the database:

```
mysql> drop database amon;
```

Once you have migrated to YARN and deleted the MapReduce service, you can remove local data from each TaskTracker host. The `mapred.local.dir` parameter is a directory on the local filesystem of each TaskTracker that contains temporary data for MapReduce. Once the service is stopped, you can remove this directory to free disk space on each host.

For detailed information on migrating from MapReduce to YARN, see [Migrating from MapReduce 1 (MRv1) to MapReduce 2 (MRv2)](#).

Switching Between MapReduce and YARN Services

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

MapReduce and YARN use separate sets of configuration files. No files are removed or altered when you change to a different framework. To change from YARN to MapReduce (or vice versa):

1. (Optional) Configure the new MapReduce or YARN service.
2. Update dependent services to use the chosen framework.
3. Configure the alternatives priority.
4. Redeploy the Oozie ShareLib.
5. Redeploy the client configuration.
6. Start the framework service to switch to.
7. (Optional) Stop the unused framework service to free up the resources it uses.

Updating Services Dependent on MapReduce

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

When you change the MapReduce framework, the dependent services that must be updated to use the new framework are:

- Hive
- Sqoop
- Oozie

To update a service:

1. Go to the service.
2. Click the **Configuration** tab.
3. Select **Scope** > *service name***(Service Wide)**.
4. Select **Scope** > **All**.
5. Locate the **MapReduce Service** property and select the YARN or MapReduce service.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Select **Actions** > **Restart**.

The Hue service is automatically reconfigured to use the same framework as Oozie and Hive. This cannot be changed. To update the Hue service:

1. Go to the Hue service.
2. Select **Actions** > **Restart**.

Configuring Alternatives Priority for Services Dependent on MapReduce

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

The alternatives priority property determines which service—MapReduce or YARN—is used by clients to run MapReduce jobs. The service with a higher value of the property is used. The MapReduce service alternatives priority is set to 91 and the YARN service is set to 92.

To configure the alternatives priority:

1. Go to the MapReduce or YARN service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Gateway Default Group**.
4. Select **Category** > **All**.
5. Type `Alternatives` in **Search** box.
6. In the **Alternatives Priority** property, set the priority value.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
8. Redeploy the client configuration.

Configuring MapReduce To Read/Write With Amazon Web Services

These are the steps required to configure MapReduce to read and write with AWS.

1. Save your AWS access key in a `.jceks` file in HDFS.

```
hadoop credential create fs.s3a.access.key -provider \
    jceks://hdfs/<hdfs directory>/<file name>.jceks -value <AWS access key id>
```

2. Put the AWS secret in the same `.jceks` file created in previous step.

```
hadoop credential create fs.s3a.secret.key -provider \
    jceks://hdfs/<hdfs directory>/<file name>.jceks -value <AWS secret access key>
```

3. Set your `hadoop.security.credential.provider.path` to the path of the `.jceks` file in the job configuration so that the MapReduce framework loads AWS credentials from the `.jceks` file in HDFS. The following example shows a Teragen MapReduce job that writes to an S3 bucket.

```
hadoop jar <path to the Hadoop MapReduce example jar file> teragen \
    -Dhadoop.security.credential.provider.path= \
    jceks://hdfs/<hdfs directory>/<file name>.jceks \
    100 s3a://<bucket name>/teragen1
```

You can specify the variables *<hdfs directory>, <file name>, <AWS access key id>*, and *<AWS secret access key>*. *<hdfs directory>* is the HDFS directory where you store the `.jceks` file. *<file name>* is the name of the `.jceks` file in HDFS.

To configure Oozie to submit S3 MapReduce jobs, see Configuring Oozie to Enable MapReduce Jobs To Read/Write from Amazon S3.

Managing YARN

Note: This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

For an overview of computation frameworks, insight into their usage and restrictions, and examples of common tasks they perform, see Managing YARN (MRv2) and MapReduce (MRv1) on page 147.

Adding the YARN Service

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the cluster name and select **Add a Service**. A list of service types display. You can add one type of service at a time.
2. Select **YARN (MR2 Included)** and click **Continue**.
3. Select the services on which the new service should depend. All services must depend on the *same* ZooKeeper service. Click **Continue**.
4. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

   Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

   The following shortcuts for specifying hostname patterns are supported:

   • Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com

Range Definition	Matching Hosts
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

Configuring Memory Settings for YARN and MRv2

The memory configuration for YARN and MRv2 memory is important to get the best performance from your cluster. Several different settings are involved. The table below shows the default settings, as well as the settings that Cloudera recommends, for each configuration option. See Managing YARN (MRv2) and MapReduce (MRv1) on page 147 for more configuration specifics; and, for detailed tuning advice with sample configurations, see Tuning YARN on page 358.

Table 1: YARN and MRv2 Memory Configuration

Cloudera Manager Property Name	CDH Property Name	Default Configuration	Cloudera Tuning Guidelines
Container Memory Minimum	`yarn.scheduler.minimum-allocation-mb`	1 GB	0
Container Memory Maximum	`yarn.scheduler.maximum-allocation-mb`	64 GB	amount of memory on largest host
Container Memory Increment	`yarn.scheduler.increment-allocation-mb`	512 MB	Use a fairly large value, such as 128 MB
Container Memory	`yarn.nodemanager.resource.memory-mb`	8 GB	8 GB
Map Task Memory	`mapreduce.map.memory.mb`	1 GB	1 GB
Reduce Task Memory	`mapreduce.reduce.memory.mb`	1 GB	1 GB
Map Task Java Opts Base	`mapreduce.map.java.opts`	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m
Reduce Task Java Opts Base	`mapreduce.reduce.java.opts`	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m
ApplicationMaster Memory	`yarn.app.mapreduce.am.resource.mb`	1 GB	1 GB
ApplicationMaster Java Opts Base	`yarn.app.mapreduce.am.command-opts`	-Djava.net.preferIPv4Stack=true	-Djava.net.preferIPv4Stack=true -Xmx768m

Configuring Directories

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Creating the Job History Directory

When adding the YARN service, the **Add Service** wizard automatically creates a job history directory. If you quit the **Add Service** wizard or it does not finish, you can create the directory outside the wizard:

1. Go to the YARN service.

2. Select **Actions** > **Create Job History Dir**.
3. Click **Create Job History Dir** again to confirm.

Creating the NodeManager Remote Application Log Directory

When adding the YARN service, the **Add Service** wizard automatically creates a remote application log directory. If you quit the **Add Service** wizard or it does not finish, you can create the directory outside the wizard:

1. Go to the YARN service.
2. Select **Actions** > **Create NodeManager Remote Application Log Directory**.
3. Click **Create NodeManager Remote Application Log Directory** again to confirm.

Importing MapReduce Configurations to YARN

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

> ⚠ **Warning:**
>
> In addition to importing configuration settings, the import process:
>
> - Configures services to use YARN as the MapReduce computation framework instead of MapReduce.
> - Overwrites existing YARN configuration and role assignments.

You can import MapReduce configurations to YARN as part of the upgrade wizard. If you do not import configurations during upgrade, you can manually import the configurations at a later time:

1. Go to the YARN service page.
2. Stop the YARN service.
3. Select **Actions** > **Import MapReduce Configuration**. The import wizard presents a warning letting you know that it will import your configuration, restart the YARN service and its dependent services, and update the client configuration.
4. Click **Continue** to proceed. The next page indicates some additional configuration required by YARN.
5. Verify or modify the configurations and click **Continue**. The Switch Cluster to MR2 step proceeds.
6. When all steps have been completed, click **Finish**.
7. (Optional) Remove the MapReduce service.

   a. Click the Cloudera Manager logo to return to the **Home** page.
   b. In the MapReduce row, right-click

   [▼]

   and select **Delete**. Click **Delete** to confirm.

8. Recompile JARs used in MapReduce applications. For further information, see <u>For MapReduce Programmers: Writing and Running Jobs</u>.

Configuring the YARN Scheduler

Minimum Required Role: <u>Configurator</u> (also provided by **Cluster Administrator, Full Administrator**)

The YARN service is configured by default to use the Fair Scheduler. You can change the scheduler type to FIFO or Capacity Scheduler. You can also modify the Fair Scheduler and Capacity Scheduler configuration. For further information on schedulers, see <u>YARN (MRv2) and MapReduce (MRv1) Schedulers</u> on page 384.

Configuring the Scheduler Type

1. Go to the YARN service.
2. Click the **Configuration** tab.
3. Select **Scope** > **ResourceManager**.
4. Select **Category** > **Main**.
5. Select a scheduler class.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the YARN service.

Modifying the Scheduler Configuration

1. Go to the YARN service.
2. Click the **Configuration** tab.
3. Click the **ResourceManager Default Group** category.
4. Select **Scope** > **ResourceManager**.
5. Type `Scheduler` in the Search box.
6. Locate a property and modify the configuration.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
8. Restart the YARN service.

Dynamic Resource Management

In addition to the static resource management available to all services, the YARN service also supports dynamic management of its static allocation. See Dynamic Resource Pools on page 372.

Configuring YARN for Long-running Applications

On a secure cluster, long-running applications such as Spark Streaming jobs will need additional configuration since the default settings only allow the `hdfs` user's delegation tokens a maximum lifetime of 7 days, which is not always sufficient.

Task Process Exit Codes

All YARN tasks on the NodeManager are run in a JVM. When a task runs successfully, the exit code is 0. Exit codes of 0 are not logged, as they are the expected result. Any non-zero exit code is logged as an error. The non-zero exit code is reported by the NodeManager as an error in the child process. The NodeManager itself is not affected by the error.

The task JVM might exit with a non-zero code for multiple reasons, though there is no exhaustive list. Exit codes can be split into two categories:

* Set by the JVM based on the OS signal received by the JVM
* Directly set in the code

Signal-Related Exit Codes

When the OS sends a signal to the JVM, the JVM handles the signal, which could cause the JVM to exit. Not all signals cause the JVM to exit. Exit codes for OS signals have a value between 128 and 160. Logs show non-zero status codes without further explanation.

Two exit values that typically do not require investigation are 137 and 143. These values are logged when the JVM is killed by the NodeManager or the OS. The NodeManager might kill a JVM due to task preemption (if that is configured) or a speculative run. The OS might kill the JVM when the JVM exceeds system limits like CPU time. You should investigate these codes if they appear frequently, as they might indicate a misconfiguration or a structural problem with regard to resources.

Exit code 154 is used in `RecoveredContainerLaunch#call` to indicate containers that were lost between NodeManager restarts without an exit code being recorded. This is usually a bug, and requires investigation.

Other Exit Codes

The JVM might exit if there is an unrecoverable error while running a task. The exit code and the message logged should provide more detail. A Java stack trace might also be logged as part of the exit. These exits should be investigated further to discover a root cause.

In the case of a streaming MapReduce job, the exit code of the JVM is the same as the mapper or reducer in use. The mapper or reducer can be a shell script or Python script. This means that the underlying script dictates the exit code: in streaming jobs, you should take this into account during your investigation.

Managing YARN ACLs

An Access Control List (ACL) is a list of specific permissions or controls that allow individual users or groups to perform specific actions upon specific objects, as well as defining what operations are allowed on a given object. YARN ACLs do not deny access; rather, they identify a user, list of users, group, or list of groups who can access a particular object.

Like [HDFS ACLs](), YARN ACLs provide a way to set different permissions for specific named users or named groups. ACLs enhance the traditional permissions model by defining access control for arbitrary combinations of users and groups instead of a single owner/user or a single group.

YARN ACL Rules and Syntax

This section describes the rules governing YARN ACLs and includes syntax examples.

YARN ACL Rules

All YARN ACLs must adhere to the following rules:

- **Special Values:**

  - The wildcard character (*) indicates that everyone has access.

    > **Note:** You cannot use the wildcard (*) character along with a list of users and/or groups in the same ACL. If you use the wildcard character it must be the *only* item in the ACL.

  - A single space entry indicates that no one has access.

- If there are no spaces in an ACL, then all entries (the listed users and/or groups) are considered authorized users.
- Group names in YARN Resource Manager ACLs are case sensitive. So, if you specify an uppercase group name in the ACL, it will not match the group name resolved from the Active Directory because Active Directory group names are resolved in lowercase.
- If an ACL starts with a single space, then it must consist of groups only.
- All entries after the occurrence of a second single space in an ACL are ignored.
- There are no ACLs that deny access to a user or group. However, if you wish to block access to an operation entirely, enter a value for a non-existent user or group (for example, `'NOUSERS NOGROUPS'`), or simply enter a single space. By doing so, you ensure that no user or group maps to a particular operation by default.
- If you wish to deny only a certain set of users and/or groups, specify every single user and/or group that requires access. Users and/or groups that are not included are "implicitly" denied access.

YARN ACL Syntax

Following are examples of YARN ACL syntax:

> **Note:** In all cases where a single space is required, you will see: `<single space>`.

- Users only

```
user1,user2,userN
```

Use a comma-separated list of user names. Do not place spaces after the commas separating the users in the list.

- Groups only

```
<single space>HR,marketing,support
```

You *must* begin group-only ACLs with a single space. Group-only ACLs use the same syntax as users, except each entry is a group name rather than user name.

- Users and Groups

```
fred,alice,haley<single space>datascience,marketing,support
```

A comma-separated list of user names, followed by a single space, followed by a comma-separated list of group names. This sample ACL authorizes access to users "fred", "alice", and "haley", and to those users in the groups "datascience", "marketing", and "support".

Examples

The following ACL entry authorizes access only to the members of "my_group":

```
<single space>my_group
```

The following ACL entry authorizes access to anyone:

```
*
```

The following ACL authorizes access to the users "john", "jane", and the group "HR":

```
john,jane<single space>HR
```

In this example, six groups ("group_1" through "group_6") are defined in the system. The following ACL authorizes access to a subset of the defined groups, allowing access to all members of groups 1 through 5 (and implicitly denies access to members of the group "group_6"):

```
<single space>group_1,group_2,group_3,group_4,group_5
```

Activating YARN ACLs

> **Important:** See YARN Admin ACL on page 156 before activating YARN ACLs, because you must configure the YARN Admin ACL first, before activation.

In a default Cloudera Manager managed YARN deployment, ACL checks are turned on but do not provide any security, which means that any user can execute administrative commands or submit an application to any YARN queue. To provide security the ACL must be changed from its default value, the wildcard character (*).

In non-Cloudera Manager managed clusters, the default YARN ACL setting is `false`, and ACLs are turned off and provide security out-of-the-box.

Activate YARN ACLs via the `yarn.acl.enable` property (values are either `true` or `false`):

```
<property>
    <name>yarn.acl.enable</name>
    <value>true</value>
</property>
```

YARN ACLs are independent of HDFS or protocol ACLs, which secure communications between clients and servers at a low level.

YARN ACL Types

This section describes the types of YARN ACLs available for use:

- [YARN Admin ACL](#) on page 156

  (`yarn.admin.acl`)

- [Queue ACL](#) on page 156

  (`aclSubmitApps and aclAdministerApps`)

- [Application ACL](#)

  (`mapreduce.job.acl-view-job` and `mapreduce.job.acl-modify-job`)

YARN Admin ACL

Use the YARN Admin ACL to allow users to run YARN administrator sub-commands, which are executed via the `yarn rmadmin <command>`.

> **!** **Important:** The YARN Admin ACL is triggered and applied *only* when you run YARN sub-commands via `yarn rmadmin <cmd>`. If you run other YARN commands via the YARN command line (for example, starting the Resource or Node Manager), it does not trigger the YARN Admin ACL check or provide the same level of security.

The default YARN Admin ACL is set to the wildcard character (*), meaning all users and groups have YARN Administrator access and privileges. So after YARN ACL enforcement is enabled, (via the `yarn.acl.enable` property) every user has YARN ACL Administrator access. Unless you wish for all users to have YARN Admin ACL access, edit the `yarn.admin.acl` setting upon initial YARN configuration, and before enabling YARN ACLs.

A typical YARN Admin ACL looks like the following, where the system's Hadoop administrator and multiple groups are granted access:

```
hadoopadmin<space>yarnadmgroup,hadoopadmgroup
```

Queue ACL

Use Queue ACLs to identify and control which users and/or groups can take actions on particular queues. Configure Queue ACLs using the [aclSubmitApps](#) and [aclAdministerApps](#) properties, which are set per queue. Queue ACLs are scheduler dependent, and the implementation and enforcement differ per scheduler type.

> **✎** **Note:** Cloudera only supports the [Fair Scheduler](#) in CDH. Cloudera does not support Scheduler Reservations (including `aclAdministerReservations`, `aclListReservations`, and `aclSubmitReservations`) and their related ACLs. For details, see [YARN Unsupported Features](#).

Unlike the YARN Admin ACL, Queue ACLs are not enabled and enforced by default. Instead, you must explicitly enable Queue ACLs. Queue ACLs are defined, per queue, in the Fair Scheduler configuration. By default, neither of the Queue ACL property types is set on any queue, and access is allowed or open to any user.

The users and groups defined in the `yarn.admin.acl` are considered to be part of the Queue ACL, `aclAdministerApps`. So any user or group that is defined in the `yarn.admin.acl` can submit to any queue and kill any running application in the system.

The aclSubmitApps Property

Use the Queue ACL `aclSubmitApps` property type to enable users and groups to submit or add an application to the queue upon which the property is set. To move an application from one queue to another queue, you must have Submit permissions for both the queue in which the application is running, and the queue into which you are moving the application. You must be an administrator to set Admin ACLs; contact your system administrator to request Submit permission on this queue.

The aclAdministerApps Property

Use the Queue ACL `aclAdministerApps` property type to enable all actions defined in the aclSubmitApps property, plus any administrative actions that have been defined (the only administrative action currently defined and supported in this context is killing an application).

> **Important:** The users and groups defined in the `yarn.admin.acl` are considered to be part of the Queue ACL, `aclAdministerApps`. So any user or group that is defined in the `yarn.admin.acl` can submit to any queue and kill any running application in the system.

Following is an example of a Queue ACL with both types defined. Note that the single space in `aclAdministerApps` indicates a group-only rule:

```
<queue name="Marketing">
  <aclSubmitApps>john,jane</aclSubmitApps>
  <aclAdministerApps><single space>others</aclAdministerApps>
</queue>
```

Queue ACL Evaluation

The better you understand how Queue ACLs are evaluated, the more prepared you are to define and configure them. First, you should have a basic understanding of how Fair Scheduler queues work.

CDH Fair Scheduler supports hierarchical queues, all of which descend from a root queue, which is automatically created and defined within the system when the Scheduler starts.

Available resources are distributed among the children ("leaf" queues) of the root queue in a typical fair scheduling fashion. Then, the children distribute their assigned resources to their children in the same fashion.

As mentioned earlier, applications are scheduled on leaf queues only. You specify queues as children of other queues by placing them as sub-elements of their parents in the Fair Scheduler allocation file (`fair-scheduler.xml`). The default Queue ACL setting for all parent and leaf queues is " " (a single space), which means that by default, no one can access any of these queues.

Queue ACL inheritance is enforced by assessing the ACLs defined in the queue hierarchy in a bottom-up order to the root queue. So within this hierarchy, access evaluations start at the level of the bottom-most leaf queue. If the ACL does not provide access, then the parent Queue ACL is checked. These evaluations continue upward until the root queue is checked.

Queue ACLs do not interact directly with the placement policy rules (the rules that determine the pools to which applications and queries are assigned) and are not part of the placement policy rules, which are executed before the ACLs are checked. The policy rules return a final result in the form of a queue name. The queue is then evaluated for access, as described earlier. The Queue ACL allows or denies access to this final queue, which means that an application can be rejected even if the placement policy returns back a queue.

> **Important:**
>
> In all YARN systems, the default setting for the root queue is reversed compared to all other queues–the root queue has a default setting of "*", which means everyone has access:
>
> | Queue ACL Property Type | Default Values | |
> |---|---|---|
> | | **Root Queue** | **All Other Queues** |
> | `aclSubmitApps` | * | " " (single space) |
> | `aclAdministerApps` | * | " " (single space) |
>
> So even when the Queue ACLs are turned on by default, everyone has access because the root queue ACL is inherited by all the leaf queues.

Best practice: A best practice for securing an environment is to set the root queue `aclSubmitApps` ACL to `<single space>`, and specify a limited set of users and groups in `aclAdministerApps`. Set the ACLs for all other queues to provide submit or administrative access as appropriate.

The order in which the two types of Queue ACLs are evaluated is always:

1. `aclSubmitApps`
2. `aclAdministerApps`

The following diagram shows the evaluation flow for Queue ACLs:
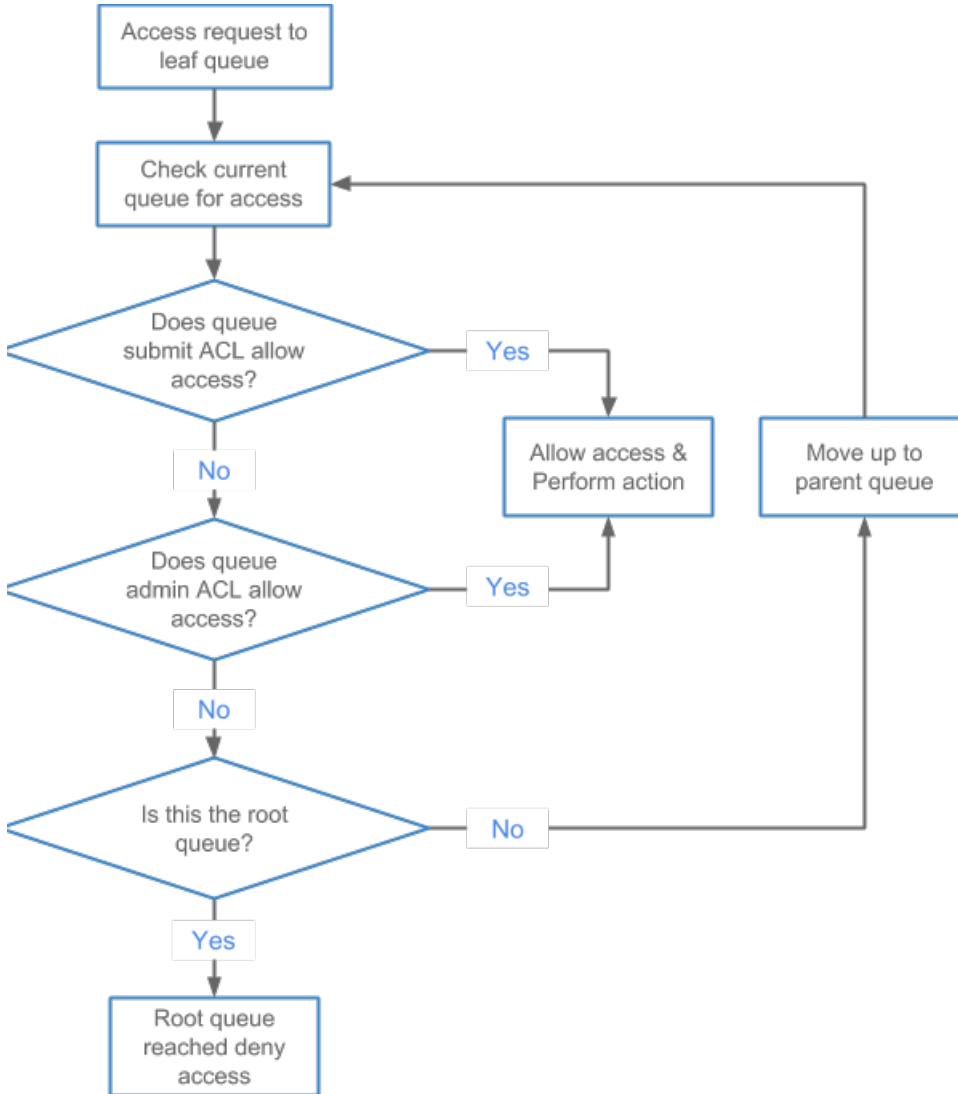
Figure 2: Queue ACL Evaluation Flow

Application ACLs

Use Application ACLs to provide a user and/or group–neither of whom is the owner–access to an application. The most common use case for Application ACLs occurs when you have a team of users collaborating on or managing a set of applications, and you need to provide read access to logs and job statistics, or access to allow for the modification of a job ([killing the job](#)) and/or application. Application ACLs are set per application and are managed by the application owner.

Users who start an application (the owners) always have access to the application they start, which includes the application logs, job statistics, and ACLs. No other user can remove or change owner access. By default, no other users

have access to the application data because the Application ACL defaults to " " (single space), which means no one has access.

MapReduce

Create and use the following MapReduce Application ACLs to [view YARN logs](#):

- `mapreduce.job.acl-view-job`

  Provides read access to the MapReduce history and the YARN logs.

- `mapreduce.job.acl-modify-job`

  Provides the same access as `mapreduce.job.acl-view-job`, and also allows the user to modify a running job.

  > **Note:** Job modification is currently limited to killing the job. No other YARN system modifications are supported.

During a search or other activities, you may come across the following two legacy settings from MapReduce; they are not supported by YARN. Do *not* use them:

- `mapreduce.cluster.acls.enabled`
- `mapreduce.cluster.administrators`

Spark

Spark ACLs follow a slightly different format, using a separate property for users and groups. Both user and group lists use a comma-separated list of entries. The wildcard character "*" allows access to anyone, and the single space " " allows access to no one. Enable Spark ACLs using the property `spark.acls.enable`, which is set to `false` by default (not enabled) and must be changed to `true` to enforce ACLs at the Spark level.

Create and use the following Application ACLs for the Spark application:

- Set `spark.acls.enable` to `true` (default is `false`).
- Set `spark.admin.acls` and `spark.admin.acls.groups` for administrative access to all Spark applications.
- Set `spark.ui.view.acls` and `spark.ui.view.acls.groups` for view access to the specific Spark application.
- Set `spark.modify.acls` and `spark.modify.acls.groups` for administrative access to the specific Spark application.

Refer to [Spark Security](#) and [Spark Configuration Security](#) for additional details.

Viewing Application Logs

The MapReduce Application ACL `mapreduce.job.acl-view-job` determines whether or not you can view an application log, and access is evaluated via the following ACLs:

- YARN Admin and Queue ACLs
- Application ACLs

After an application is in the "finished" state, logs are aggregated, depending on your cluster setup. You can access the aggregated logs via the MapReduce History server web interface. Aggregated logs are stored on shared cluster storage, which in most cases is HDFS. You can also share log aggregation via storage options like S3 or Azure by modifying the `yarn.nodemanager.remote-app-log-dir` setting in Cloudera Manager to point to either S3 or Azure, which should already be configured.

The shared storage on which the logs are aggregated helps to prevent access to the log files via file level permissions. Permissions on the log files are also set at the file system level, and are enforced by the file system: the file system can block any user from accessing the file, which means that the user cannot open/read the file to check the ACLs that are contained within.

In the cluster storage use case of HDFS, you can only access logs that are aggregated via the:

- Application owner

- Group defined for the MapReduce History server

When an application runs, generates logs, and then places the logs into HDFS, a path/structure is generated (for example: `/tmp/logs/john/logs/application_1536220066338_0001`). So access for the application owner "john" might be set to `700`, which means `read`, `write`, `execute`; no one else can view files underneath this directory. If you don't have HDFS access, you will be denied access. Command line users identified in `mapreduce.job.acl-view-job` are also denied access at the file level. In such a use case, the Application ACLs stored inside the aggregated logs will never be evaluated because the Application ACLs do not have file access.

For clusters that do not have log aggregation, logs for running applications are kept on the node where the container runs. You can access these logs via the Resource Manager and Node Manager web interface, which performs the ACL checks.

Killing an Application

The Application ACL `mapreduce.job.acl-modify-job` determines whether or not a user can modify a job, but in the context of YARN, this only allows the user to kill an application. The kill action is application agnostic and part of the YARN framework. Other application types, like MapReduce or Spark, implement their own kill action independent of the YARN framework. MapReduce provides the kill actions via the `mapred` command.

For YARN, the following three groups of users are allowed to kill a running application:

- The application owner
- A cluster administrator defined in `yarn.admin.acl`
- A queue administrator defined in `aclAdministerApps` for the queue in which the application is running

Note that for the queue administrators, ACL inheritance applies, as described earlier.

Application ACL Evaluation

The better you understand how YARN ACLs are evaluated, the more prepared you will be to define and configure the various YARN ACLs available to you. For example, if you enable user access in Administrator ACLs, then you must be aware that user may have access to/see sensitive data, and should plan accordingly. So if you are the administrator for an entire cluster, you also have access to the logs for running applications, which means you can view sensitive information in those logs associated with running the application.

Best Practice: A best practice for securing an environment is to set the YARN Admin ACL to include a limited set of users and or groups.

The following diagram shows the evaluation flow for Application ACLs:
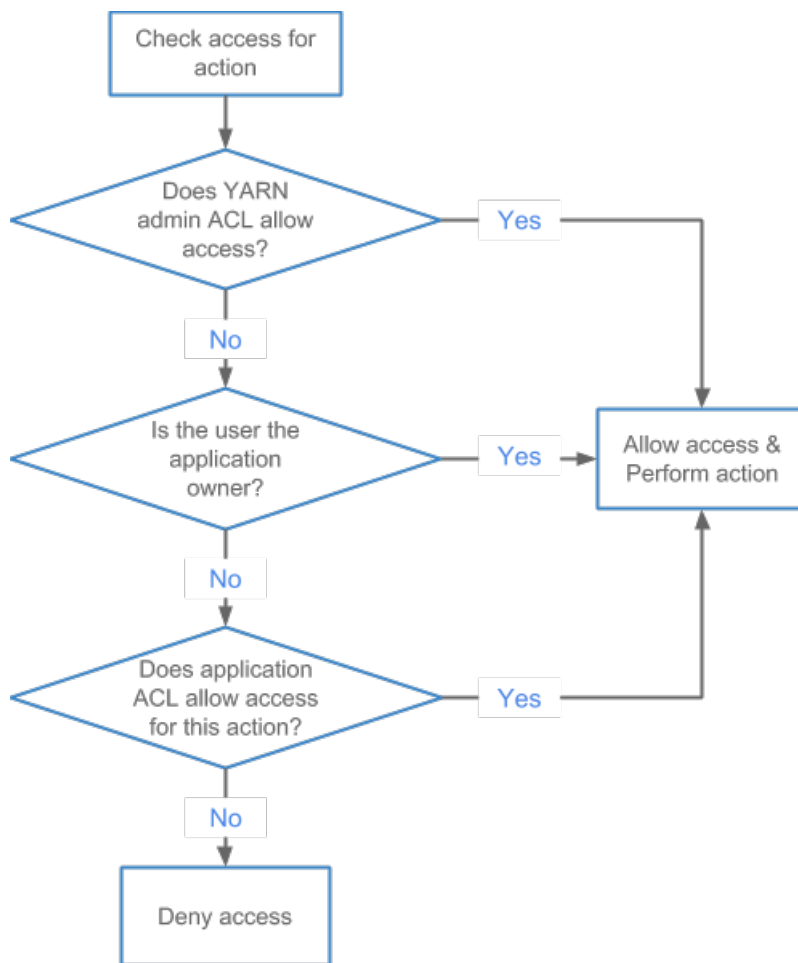
Figure 3: Application ACL Evaluation Flow

The following diagram shows a sample queue structure, starting with leaf queues on the bottom, up to root queue at the top; use it to follow the examples of killing an application and viewing a log:
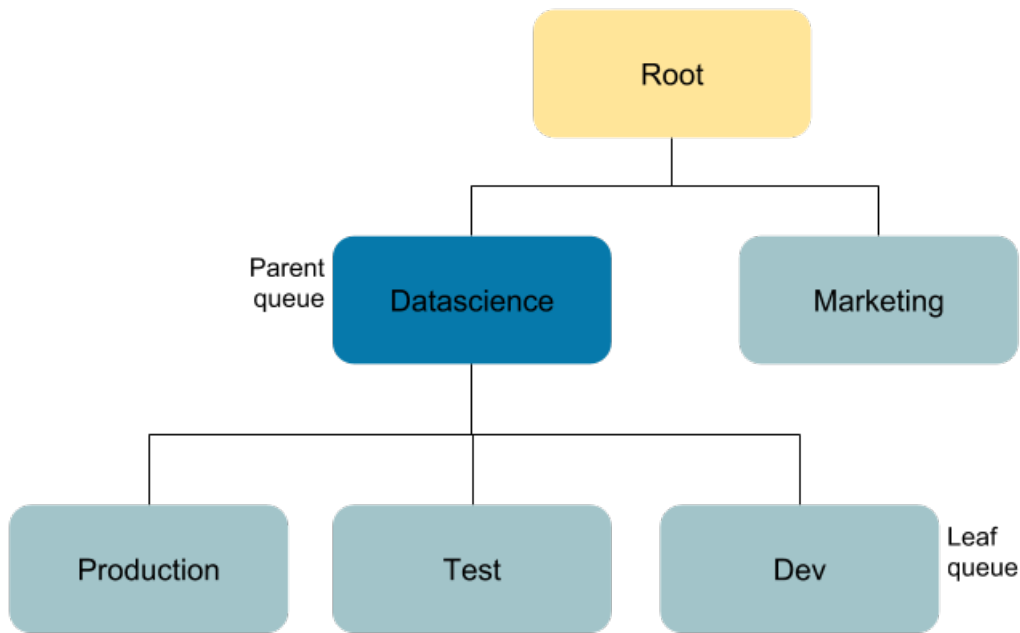
Figure 4: Queue Structure

Example: Killing an Application in the Queue "Production"

For this Application ACL evaluation flow example, assume the following for `application_1536220066338_0001` running in the queue "Production":

- Application owner: John
- "Datascience" queue administrator: Jane
- YARN cluster administrator: Bob

In this use case, John attempts to kill the application (see), which is allowed because he is the application owner.

Working as the queue administrator, Jane attempts to kill a job in the queue "Production", which she can do as the queue administrator of the parent queue.

Bob is the YARN cluster administrator and he is also listed as a user in the Admin ACL. He attempts to kill the job for which he is not the owner, but because he is the YARN cluster administrator, he can kill the job.

Example: Moving the Application and Viewing the Log in the Queue "Test"

For this Application ACL evaluation flow example, assume the following for `application_1536220066338_0002` running in the queue "Test":

- Application owner: John
- "Marketing" and "Dev" queue administrator: Jane
- Jane has log view rights via the `mapreduce.job.acl-view-job` ACL
- YARN cluster administrator: Bob

In this use case, John attempts to view the logs for his job, which is allowed because he is the application owner.

Jane attempts to access `application_1536220066338_0002` in the queue "Test" to move the application to the "Marketing" queue. She is denied access to the "Test" queue via the queue ACLs—so she cannot submit to or administer the queue "Test". She is also unable to kill a job running in queue "Test". She then attempts to access the logs for `application_1536220066338_0002` and is allowed access via the `mapreduce.job.acl-view-job` ACL.

Bob attempts to access `application_1536220066338_0002` in the queue "Test" to move the application to the "Marketing" queue. As the YARN cluster administrator, he has access to all queues and can move the application.

> **Note:** Permissions on the log files are also set at the filesystem level and are enforced by the filesystem: the filesystem can block you from accessing the file, which means that you can not open/read the file to check the ACLs that are contained in the file.

Configuring and Enabling YARN ACLs

To configure YARN ACLs, refer to Configuring ACLs on page 378.

To enable YARN ACLs, refer to Enabling ACLs on page 378.

Managing MapReduce

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

For an overview of computation frameworks, insight into their usage and restrictions, and examples of common tasks they perform, see Managing YARN (MRv2) and MapReduce (MRv1) on page 147.

Configuring the MapReduce Scheduler

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

The MapReduce service is configured by default to use the FairScheduler. You can change the scheduler type to FIFO or Capacity Scheduler. You can also modify the Fair Scheduler and Capacity Scheduler configuration. For further information on schedulers, see YARN (MRv2) and MapReduce (MRv1) Schedulers on page 384.

Configuring the Task Scheduler Type

1. Go to the MapReduce service.
2. Click the **Configuration** tab.
3. Select **Scope** > **JobTracker**.
4. Select **Category** > **Classes** .
5. In the **Task Scheduler** property, select a scheduler.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the JobTracker to apply the new configuration:

   a. Click the **Instances** tab.
   b. Click the **JobTracker** role.
   c. Select **Actions for Selected** > **Restart**.

Modifying the Scheduler Configuration

1. Go to the MapReduce service.
2. Click the **Configuration** tab.
3. Select **Scope** > **JobTracker**.
4. Select **Category** > **Jobs**.
5. Modify the configuration properties.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the JobTracker to apply the new configuration:

    a. Click the **Instances** tab.

    b. Click the **JobTracker** role.

    c. Select **Actions for Selected** > **Restart**.

Configuring the MapReduce Service to Save Job History

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Normally job history is saved on the host on which the JobTracker is running. You can configure JobTracker to write information about every job that completes to a specified HDFS location. By default, the information is retained for 7 days.

Enabling Map Reduce Job History To Be Saved to HDFS

1. Create a folder in HDFS to contain the history information. When creating the folder, set the owner and group to `mapred:hadoop` with permission setting 775.
2. Go to the MapReduce service.
3. Click the **Configuration** tab.
4. Select **Scope** > **JobTracker**.
5. Select **Category** > **Paths**.
6. Set the **Completed Job History Location** property to the location that you created in step 1.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

7. Click **Save Changes**.
8. Restart the MapReduce service.

Setting the Job History Retention Duration

1. Select the **JobTracker Default Group** category.
2. Set the **Job History Files Maximum Age** property (`mapreduce.jobhistory.max-age-ms`) to the length of time (in milliseconds, seconds, minutes, or hours) that you want job history files to be kept.
3. Restart the MapReduce service.

The Job History Files Cleaner runs at regular intervals to check for job history files that are ready to be deleted. By default, the interval is 24 hours. To change the frequency with which the Job History Files Cleaner runs:

1. Select the **JobTracker Default Group** category.
2. Set the **Job History Files Cleaner Interval** property (`mapreduce.jobhistory.cleaner.interval`) to the desired frequency (in milliseconds, seconds, minutes, or hours).
3. Restart the MapReduce service.

Configuring Client Overrides

A configuration property qualified with **(Client Override)** is a server-side setting that ignores any value a client tries to set for that property. It performs the same role as its unqualified counterpart, and applies the configuration to the service with the setting `<final>true</final>`.

For example, if you set the Map task heap property to 1 GB in the job configuration code, but the service's heap property qualified with (Client Override) is set to 500 MB, then 500 MB is applied.

Managing ZooKeeper

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

This topic describes how to add, remove, and replace ZooKeeper roles.

Using Multiple ZooKeeper Services

Cloudera Manager requires dependent services within CDH to use the same ZooKeeper service. If you configure dependent CDH services to use different ZooKeeper services, Cloudera Manager reports the following error:

```
com.cloudera.cmf.command.CmdExecException:java.lang.RuntimeException:
java.lang.IllegalStateException: Assumption violated:
getAllDependencies returned multiple distinct services of the same type
at SeqFlowCmd.java line 120
in com.cloudera.cmf.command.flow.SeqFlowCmd run()
```

CDH services that are not dependent can use different ZooKeeper services. For example, Kafka does not depend on any services other than ZooKeeper. You might have one ZooKeeper service for Kafka, and one ZooKeeper service for the rest of your CDH services.

Adding a ZooKeeper Service Using Cloudera Manager

Minimum Required Role: Full Administrator

When adding the ZooKeeper service, the **Add Service** wizard automatically initializes the data directories.

When you add Zookeeper servers to an existing ensemble, a rolling restart of all zookeeper is required in order to allow all zookeeper servers to have the same configurations

If you quit the **Add Service** wizard or it does not finish successfully, you can initialize the directories outside the wizard by following these steps:

1. Go to the ZooKeeper service.
2. Select **Actions** > **Initialize**.
3. Click **Initialize** again to confirm.

> **Note:** If the data directories are not initialized, the ZooKeeper servers cannot be started.

In a production environment, you should deploy ZooKeeper as an ensemble with an odd number of servers. As long as a majority of the servers in the ensemble are available, the ZooKeeper service will be available. The minimum recommended ensemble size is three ZooKeeper servers, and Cloudera recommends that each server run on a separate machine. In addition, the ZooKeeper server process should have its own dedicated disk storage if possible.

Replacing a Zookeeper Disk Using Cloudera Manager

Minimum Required Role: Full Administrator

1. In Cloudera Manager, update the **Data Directory** and **Transaction Log Directory** settings.
2. Stop a single ZooKeeper role.
3. Move the contents to the new disk location (modify mounts as needed). Make sure the permissions and ownership are correct.
4. Start the ZooKeeper role.
5. Repeat steps 2-4 for any remaining ZooKeeper roles.

Replacing a ZooKeeper Role Using Cloudera Manager

Minimum Required Role: Full Administrator

1. Go to **ZooKeeper Instances**.
2. Stop the ZooKeeper role on the old host.
3. Confirm the ZooKeeper service has elected one of the remaining hosts as a leader on the ZooKeeper Status page. See Confirming the Election Status of a ZooKeeper Service on page 167.
4. On the **ZooKeeper Instances** page, remove the ZooKeeper role from old host.
5. Add a new ZooKeeper role on the new host.
6. Start the new ZooKeeper role.

7. Confirm on the **ZooKeeper Status** page that there is a leader and all other hosts are followers.

8. Restart the other ZooKeeper roles.

9. Restart any dependent services such as HBase, HDFS Failover Controllers with HDFS High Availability, YARN or Mapreduce v1 with High Availability, or other service.

10. Perform a failover to make one HDFS NameNode active. See Manually Failing Over to the Standby NameNode on page 429.

Replacing a ZooKeeper Role Using Cloudera Manager with Zookeeper Service Downtime

Minimum Required Role: **Full Administrator**

1. Go to **ZooKeeper Instances**.
2. Stop the ZooKeeper role on the old host.
3. Remove the ZooKeeper role from old host on the **ZooKeeper Instances** page.
4. Add a new ZooKeeper role on the new host.
5. Restart the old ZooKeeper servers that have outdated configuration.
6. Confirm the ZooKeeper service has elected one of the restarted hosts as a leader on the **ZooKeeper Status** page. See Confirming the Election Status of a ZooKeeper Service.
7. Restart the newly added Zookeeper server.
8. Restart/rolling restart any dependent services such as HBase, HDFS, YARN, Hive, or other services that are marked to have stale configuration.

Replacing a ZooKeeper Role Using Cloudera Manager without Zookeeper Service Downtime

Minimum Required Role: **Full Administrator**

> **Note:** This process is valid only if the SASL authentication is not turned on between the Zookeeper servers. You can check this in **Cloudera Manager** > **Zookeeper** > **Configuration** > **Enable Server to Server SASL Authentication**.

1. Go to **ZooKeeper Instances**.
2. Stop the ZooKeeper role on the old host.
3. Confirm the ZooKeeper service has elected one of the remaining hosts as a leader on the **ZooKeeper Status** page. See Confirming the Election Status of a ZooKeeper Service.
4. On the **ZooKeeper Instances** page, remove the ZooKeeper role from the old host.
5. Add a new ZooKeeper role on the new host.
6. Change the individual configuration of the newly added Zookeeper role to have the highest ZooKeeper Server ID set in the cluster.
7. Go to **Zookeeper** > **Instances** and click the newly added **Server** instance.
8. In the individual **Server** page, select **Start this Server** from the **Actions** dropdown menu to start the new ZooKeeper role.

> **Note:** If you try it from elsewhere, you may see an error message.

9. On the **ZooKeeper Status** page, confirm that there is a leader and all other hosts are followers.
10. Restart the ZooKeeper server that has an outdated configuration and is a follower.
11. Restart the leader Zookeeper server that has an outdated configuration.
12. Confirm that a leader has been elected after the restart, and the whole Zookeeper service is in green state.
13. Restart/rolling restart any dependent services such as HBase, HDFS, YARN, Hive, or other services that are marked to have stale configuration.

Adding or Deleting a ZooKeeper Role on an Unmanaged Cluster

Minimum Required Role: **Full Administrator**

For information on administering ZooKeeper from the command line, see the ZooKeeper Getting Started Guide.

Replacing a ZooKeeper Role on an Unmanaged Cluster

Minimum Required Role: Full Administrator

These instructions assume you are using ZooKeeper from the command line. For more information, see the ZooKeeper Getting Started Guide.

1. Stop the ZooKeeper role on the old host.
2. Confirm the ZooKeeper Quorum has elected a leader. See Confirming the Election Status of a ZooKeeper Service on page 167.
3. Add a new ZooKeeper role on the new server.
4. Identify the `dataDir` location from the `zoo.cfg` file. This defaults to `/var/lib/zookeeper`.
5. Identify the ID number for the ZooKeeper Server from the `myid` file in the configuration: `cat`
   `/var/lib/zookeeper/myid`
6. On all the ZooKeeper hosts, edit the `zoo.cfg` file so the server ID references the new server hostname. For example:

   ```
   server.1=zk1.example.org:3181:4181
   server.2=zk2.example.org:3181:4181
   server.4=zk4.example.org:3181:4181
   ```

7. Restart the ZooKeeper hosts.
8. Confirm the ZooKeeper Quorum has elected a leader and the other hosts are followers. See Confirming the Election Status of a ZooKeeper Service on page 167.
9. Restart any dependent services such as HBase, HDFS Failover Controllers with HDFS High Availability, or YARN or Mapreduce v1 with High Availability.
10. Perform a failover to make one HDFS NameNode active. See Manually Failing Over to the Standby NameNode on page 429.

Confirming the Election Status of a ZooKeeper Service

Determining the election status of a ZooKeeper host requires that you have installed telnet or nc (netcat), running from a host with network access to the ZooKeeper host. The default ZooKeeper client port is 2181. Run the following command against each ZooKeeper host:

echo "stat" | nc server.example.org 2181 | grep Mode

For example, a follower host would return the message:

```
Mode: follower
```

You can use telnet, if you prefer.

```
$ telnet server.example.org 2181
```

Sample output would be similar to the following.

```
Trying 10.1.2.154...
Connected to server.example.org.
Escape character is '^]'.
stat
Zookeeper version: 3.4.5-cdh5.4.4--1, built on 07/06/2015 23:54 GMT
...

Latency min/avg/max: 0/1/40
Received: 631
Sent: 677
Connections: 7
Outstanding: 0
Zxid: 0x30000011a
Mode: follower                  <----
```

```
Node count: 40
Connection closed by foreign host.
```

Configuring Services to Use the GPL Extras Parcel

After you install the GPL Extras parcel, reconfigure and restart services that need to use LZO functionality. Any service that does not require the use of LZO need not be configured.

HDFS and MapReduce

1. Go to the HDFS service.
2. Click the **Configuration** tab.
3. Search for the `io.compression.codecs` property.
4. In the **Compression Codecs** property, click in the field, then click the **+** sign to open a new value field.
5. Add the following two codecs:

   - `com.hadoop.compression.lzo.LzoCodec`
   - `com.hadoop.compression.lzo.LzopCodec`

6. Save your configuration changes.
7. Restart HDFS.
8. Redeploy the HDFS client configuration.

Oozie

1. Go to `/var/lib/oozie` on each Oozie server and even if the LZO JAR is present, symlink the Hadoop LZO JAR:

   - **CDH 5** - `/opt/cloudera/parcels/GPLEXTRAS/lib/hadoop/lib/hadoop-lzo.jar`
   - **CDH 4** - `/opt/cloudera/parcels/HADOOP_LZO/lib/hadoop/lib/hadoop-lzo.jar`

2. Restart Oozie.

HBase
Restart HBase.

Impala
Restart Impala.

Hive
Restart the Hive server.

Sqoop 1

1. Add the following entries to the Sqoop 1 Client Client Advanced Configuration Snippet (Safety Valve)

   - `HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/cloudera/parcels/GPLEXTRAS/lib/hadoop/lib/`
   - `JAVA_LIBRARY_PATH=$JAVA_LIBRARY_PATH:/opt/cloudera/parcels/GPLEXTRAS/lib/hadoop/lib/native`

2. Re-deploy the client configuration.

Managing Hosts

Cloudera Manager provides a number of features that let you configure and manage the hosts in your clusters.

The Hosts screen has the following tabs:

The Status Tab

Viewing All Hosts

To display summary information about all the hosts managed by Cloudera Manager, click **Hosts** in the main navigation bar. The All Hosts page displays with a list of all the hosts managed by Cloudera Manager.

The list of hosts shows the overall status of the Cloudera Manager-managed hosts in your cluster.

- The information provided varies depending on which columns are selected. To change the columns, click the **Columns:** *n* **Selected** drop-down and select the checkboxes next to the columns to display.
- Click ❯ to the left of the number of roles to list all the role instances running on that host.
- Filter the hosts list by entering search terms (hostname, IP address, or role) in the search box separated by commas or spaces. Use quotes for exact matches (for example, strings that contain spaces, such as a role name) and brackets to search for ranges. Hosts that match any of the search terms are displayed. For example:

```
hostname[1-3], hostname8 hostname9, "hostname.example.com"
hostname.example.com "HDFS DataNode"
```

- You can also search for hosts by selecting a value from the facets in the **Filters** section at the left of the page.
- If the Configuring Agent Heartbeat and Health Status Options on page 34 are configured as follows:

  - Send Agent heartbeat every *x*
  - Set health status to Concerning if the Agent heartbeats fail *y*
  - Set health status to Bad if the Agent heartbeats fail *z*

  The value *v* for a host's Last Heartbeat facet is computed as follows:

  - $v < x * y$ = Good
  - $v >= x * y$ and $<= x * z$ = Concerning
  - $v >= x * z$ = Bad

Viewing the Hosts in a Cluster

Do one of the following:

- Select **Clusters** > *Cluster name* > **Hosts**.
- In the Home screen, click ▤ Hosts in a full form cluster table.

The All Hosts page displays with a list of the hosts filtered by the cluster name.

Viewing Individual Hosts

You can view detailed information about an individual host—resources (CPU/memory/storage) used and available, which processes it is running, details about the host agent, and much more—by clicking a host link on the All Hosts page. See Viewing Host Details on page 170.

The Configuration Tab

The **Configuration** tab lets you set properties related to parcels and to resource management, and also monitoring properties for the hosts under management. The configuration settings you make here will affect all your managed hosts. You can also configure properties for individual hosts from the Host Details page (see Viewing Host Details on page 170) which will override the global properties set here).

To edit the **Default** configuration properties for hosts:

    1. Click the **Configuration** tab.

For more information on making configuration changes, see Modifying Configuration Properties Using Cloudera Manager on page 60.

The Roles and Disks Overview Tabs

Role Assignments

You can view the assignment of roles to hosts as follows:

    1. Click the **Roles** tab.
    2. Click a cluster name or **All Clusters**.

Disks Overview

Click the **Disks Overview** tab to display an overview of the status of all disks in the deployment. The statistics exposed match or build on those in `iostat`, and are shown in a series of histograms that by default cover every physical disk in the system.

Adjust the endpoints of the time line to see the statistics for different time periods. Specify a filter in the box to limit the displayed data. For example, to see the disks for a single rack rack1, set the filter to: `logicalPartition = false and rackId = "rack1"` and click **Filter**. Click a histogram to drill down and identify outliers. Mouse over the graph and click ⤢ to display additional information about the chart.

The Templates Tab

The **Templates** tab lets you create and manage host templates, which provide a way to specify a set of role configurations that should be applied to a host. This greatly simplifies the process of adding new hosts, because it lets you specify the configuration for multiple roles on a host in a single step, and then (optionally) start all those roles.

The Parcels Tab

In the **Parcels** tab you can download, distribute, and activate available parcels to your cluster. You can use parcels to add new products to your cluster, or to upgrade products you already have installed.

Viewing Host Details

You can view detailed information about each host, including:

- Name, IP address, rack ID
- Health status of the host and last time the Cloudera Manager Agent sent a heartbeat to the Cloudera Manager Server
- Number of cores
- System load averages for the past 1, 5, and 15 minutes
- Memory usage

- File system disks, their mount points, and usage
- Health test results for the host
- Charts showing a variety of metrics and health test results over time.
- Role instances running on the host and their health
- CPU, memory, and disk resources used for each role instance

To view detailed host information:

1. Click the **Hosts** tab.
2. Click the name of one of the hosts. The Status page is displayed for the host you selected.
3. Click tabs to access specific categories of information. Each tab provides various categories of information about the host, its services, components, and configuration.

From the status page you can view details about several categories of information.

Status

The Status page is displayed when a host is initially selected and provides summary information about the status of the selected host. Use this page to gain a general understanding of work being done by the system, the configuration, and health status.

If this host has been decommissioned or is in maintenance mode, you will see the following icon(s) (⬛, ⚙) in the top bar of the page next to the status message.

Details

This panel provides basic system configuration such as the host's IP address, rack, health status summary, and disk and CPU resources. This information summarizes much of the detailed information provided in other panes on this tab. To view details about the Host agent, click the Host Agent link in the Details section.

Health Tests

Cloudera Manager monitors a variety of metrics that are used to indicate whether a host is functioning as expected. The Health Tests panel shows health test results in an expandable/collapsible list, typically with the specific metrics that the test returned. (You can Expand All or Collapse All from the links at the upper right of the Health Tests panel).

- The color of the text (and the background color of the field) for a health test result indicates the status of the results. The tests are sorted by their health status – Good, Concerning, Bad, or Disabled. The list of entries for good and disabled health tests are collapsed by default; however, Bad or Concerning results are shown expanded.
- The text of a health test also acts as a link to further information about the test. Clicking the text will pop up a window with further information, such as the meaning of the test and its possible results, suggestions for actions you can take or how to make configuration changes related to the test. The help text for a health test also provides a link to the relevant monitoring configuration section for the service. See Configuring Monitoring Settings on page 223 for more information.

Health History

The Health History provides a record of state transitions of the health tests for the host.

- Click the arrow symbol at the left to view the description of the health test state change.
- Click the **View** link to open a new page that shows the state of the host at the time of the transition. In this view some of the status settings are greyed out, as they reflect a time in the past, not the current status.

File Systems

The File systems panel provides information about disks, their mount points and usage. Use this information to determine if additional disk space is required.

Roles

Use the Roles panel to see the role instances running on the selected host, as well as each instance's status and health. Hosts are configured with one or more role instances, each of which corresponds to a service. The role indicates which daemon runs on the host. Some examples of roles include the NameNode, Secondary NameNode, Balancer, JobTrackers, DataNodes, RegionServers and so on. Typically a host will run multiple roles in support of the various services running in the cluster.

Clicking the role name takes you to the role instance's status page.

You can delete a role from the host from the Instances tab of the Service page for the parent service of the role. You can add a role to a host in the same way. See Role Instances on page 209.

Charts

Charts are shown for each host instance in your cluster.

See Viewing Charts for Cluster, Service, Role, and Host Instances on page 221 for detailed information on the charts that are presented, and the ability to search and display metrics of your choice.

Processes

The Processes page provides information about each of the processes that are currently running on this host. Use this page to access management web UIs, check process status, and access log information.

> **Note:** The Processes page may display exited startup processes. Such processes are cleaned up within a day.

The Processes tab includes a variety of categories of information.

- **Service** - The name of the service. Clicking the service name takes you to the service status page. Using the triangle to the right of the service name, you can directly access the tabs on the role page (such as the Instances, Commands, Configuration, Audits, or Charts Library tabs).
- **Instance** - The role instance on this host that is associated with the service. Clicking the role name takes you to the role instance's status page. Using the triangle to the right of the role name, you can directly access the tabs on the role page (such as the Processes, Commands, Configuration, Audits, or Charts Library tabs) as well as the status page for the parent service of the role.
- **Name** - The process name.
- **Links** - Links to management interfaces for this role instance on this system. These is not available in all cases.
- **Status** - The current status for the process. Statuses include stopped, starting, running, and paused.
- **PID** - The unique process identifier.
- **Uptime** - The length of time this process has been running.
- **Full log file** - A link to the full log (a file external to Cloudera Manager) for this host log entries for this host.
- **Stderr** - A link to the stderr log (a file external to Cloudera Manager) for this host.
- **Stdout** - A link to the stdout log (a file external to Cloudera Manager) for this host.

Resources

The Resources page provides information about the resources (CPU, memory, disk, and ports) used by every service and role instance running on the selected host.

Each entry on this page lists:

- The service name
- The name of the particular instance of this service
- A brief description of the resource
- The amount of the resource being consumed or the settings for the resource

The resource information provided depends on the type of resource:

- **CPU** - An approximate percentage of the CPU resource consumed.
- **Memory** - The number of bytes consumed.
- **Disk** - The disk location where this service stores information.
- **Ports** - The port number being used by the service to establish network connections.

Commands

The Commands page shows you running or recent commands for the host you are viewing. See [Viewing Running and Recent Commands](#) on page 241 for more information.

Configuration

Minimum Required Role: [Full Administrator](#)

The Configuration page for a host lets you set properties for the selected host. You can set properties in the following categories:

- **Advanced** - Advanced configuration properties. These include the Java Home Directory, which explicitly sets the value of `JAVA_HOME` for all processes. This overrides the auto-detection logic that is normally used.
- **Monitoring** - Monitoring properties for this host. The monitoring settings you make on this page will override the global host monitoring settings you make on the Configuration tab of the Hosts page. You can configure monitoring properties for:
  - health check thresholds
  - the amount of free space on the filesystem containing the Cloudera Manager Agent's log and process directories
  - a variety of conditions related to memory usage and other properties
  - alerts for health check events

  For some monitoring properties, you can set thresholds as either a percentage or an absolute value (in bytes).

- **Other** - Other configuration properties.
- **Parcels** - Configuration properties related to parcels. Includes the **Parcel Director** property, the directory that parcels will be installed into on this host. If the `parcel_dir` variable is set in the Agent's `config.ini` file, it will override this value.
- **Resource Management** - Enables resource management using control groups (cgroups).

For more information, see the description for each or property or see [Modifying Configuration Properties Using Cloudera Manager](#) on page 60.

Components

The Components page lists every component installed on this host. This may include components that have been installed but have not been added as a service (such as YARN, Flume, or Impala).

This includes the following information:

- **Component** - The name of the component.
- **Version** - The version of CDH from which each component came.
- **Component Version** - The detailed version number for each component.

Audits

The Audits page lets you filter for audit events related to this host. See [Lifecycle and Security Auditing](#) on page 301 for more information.

Charts Library

The Charts Library page for a host instance provides charts for all metrics kept for that host instance, organized by category. Each category is collapsible/expandable. See [Viewing Charts for Cluster, Service, Role, and Host Instances](#) on page 221 for more information.

Using the Host Inspector

Minimum Required Role: [Full Administrator](#)

You can use the host inspector to gather information about hosts that Cloudera Manager is currently managing. You can review this information to better understand system status and troubleshoot any existing issues. For example, you might use this information to investigate potential DNS misconfiguration.

The inspector runs tests to gather information for functional areas including:

- Networking
- System time
- User and group configuration
- HDFS settings
- Component versions

Common cases in which this information is useful include:

- Installing components
- Upgrading components
- Adding hosts to a cluster
- Removing hosts from a cluster

Running the Host Inspector

1. Click the **Hosts** tab and select **All Hosts**.
2. Click the **Inspect All Hosts** button. Cloudera Manager begins several tasks to inspect the managed hosts.
3. After the inspection completes, click **Download Result Data** or **Show Inspector Results** to review the results.

The results of the inspection displays a list of all the validations and their results, and a summary of all the components installed on your managed hosts.

If the validation process finds problems, the **Validations** section will indicate the problem. In some cases the message may indicate actions you can take to resolve the problem. If an issue exists on multiple hosts, you may be able to view the list of occurrences by clicking a small triangle that appears at the end of the message.

The **Version Summary** section shows all the components that are available from Cloudera, their versions (if known) and the CDH distribution to which they belong.

Viewing Past Host Inspector Results

You can view the results of a past host inspection by looking for the Host Inspector command using the **Recent Commands** feature.

1. Click the Running Commands indicator ( ) just to the left of the Search box at the right hand side of the navigation bar.
2. Click the **Recent Commands** button.
3. If the command is too far in the past, you can use the Time Range Selector to move the time range back to cover the time period you want.
4. When you find the Host Inspector command, click its name to display its subcommands.
5. Click the **Show Inspector Results** button to view the report.

See [Viewing Running and Recent Commands](#) on page 241 for more information about viewing past command activity.

Adding a Host to the Cluster

Minimum Required Role: [Full Administrator](#)

> **Important:** As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.
>
> This can affect new installations, upgrades, adding new hosts to a cluster, adding a cluster and adding new services to a cluster.
>
> For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

You can add one or more hosts to your cluster using the Add Hosts wizard, which installs the Oracle JDK, CDH, and Cloudera Manager Agent software. After the software is installed and the Cloudera Manager Agent is started, the Agent connects to the Cloudera Manager Server and you can use the Cloudera Manager Admin Console to manage and monitor CDH on the new host.

The Add Hosts wizard does not create roles on the new host; once you have successfully added the host(s) you can either add roles, one service at a time, or apply a host template, which can define role configurations for multiple roles.

> **Important:**
>
> - Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both *host01.example.com* and *host01.standby.example.com* managed by the same Cloudera Manager Server.
> - All hosts in a single cluster must be running the same version of CDH.
> - When you add a new host, you must install the same version of CDH to enable the new host to work with the other hosts in the cluster. The installation wizard lets you select the version of CDH to install, and you can choose a custom repository to ensure that the version you install matches the version on the other hosts.
> - If you are managing multiple clusters, select the version of CDH that matches the version in use on the cluster where you plan to add the new host.
> - When you add a new host, the following occurs:
>
>   - YARN `topology.map` is updated to include the new host
>   - Any service that includes `topology.map` in its configuration—Flume, Hive, Hue, Oozie, Solr, Spark, Sqoop 2, YARN—is marked stale
>
>   At a convenient point after adding the host you should restart the stale services to pick up the new configuration.

Use one of the following methods to add a new host:

Using the Add Hosts Wizard to Add Hosts

You can use the Add Hosts wizard to install CDH, Impala, and the Cloudera Manager Agent on a host.

Disable TLS Encryption or Authentication

If you have enabled TLS encryption or authentication for the Cloudera Manager Agents, you must disable both of them before starting the Add Hosts wizard. Otherwise, skip to the next step.

If you perform this step, then skip step 2 (Alternate Method of Installing Cloudera Manager Agent without Disabling TLS). If you skip step 1 and perform step 2, then continue to step 3 (Using the Add Hosts Wizard to Add Hosts).

> **Important:** This step temporarily puts the existing cluster hosts in an unmanageable state; they are still configured to use TLS and so cannot communicate with the Cloudera Manager Server. Roles on these hosts continue to operate normally, but Cloudera Manager is unable to detect errors and issues in the cluster and reports all hosts as being in bad health. To work around this issue, you can manually install the Cloudera Manager Agent on the new host. See Alternate Method of Installing Cloudera Manager Agent without Disabling TLS on page 176.

1. From the **Administration** tab, select **Settings**.
2. Select the **Security** category.
3. Disable TLS by clearing the following options: **Use TLS Encryption for Agents**, and **Use TLS Authentication of Agents to Server**.
4. Click **Save Changes** to save the settings.
5. Log in to the Cloudera Manager Server host.
6. Restart the Cloudera Manager Server with the following command:

```
sudo service cloudera-scm-server restart
```

The changes take effect after the restart.

Alternate Method of Installing Cloudera Manager Agent without Disabling TLS

If you have TLS encryption or authentication enabled in your cluster, you must either disable TLS during the installation, or install the Cloudera Manager Agent manually using the following procedure:

1. Copy the repository configuration file from an existing host in the cluster to the new host. For example:

OS	Command
RHEL	`$ sudo scp mynode.example.com:/etc/yum.repos.d/ cloudera-manager.repo /etc/yum.repos.d/ cloudera-manager.repo`
SLES	`$ sudo scp mynode.example.com:/etc/zypp/zypper.conf/ cloudera-cm.repo /etc/zypp/zypper.conf/cloudera-cm.repo`
Ubuntu or Debian	`$ sudo scp mynode.example.com:/etc/apt/sources.list.d/ cloudera.list /etc/apt/sources.list.d/cloudera.list`

2. Remove cached package lists and other transient data by running the following command:

OS	Command
RHEL	`$ sudo yum clean all`
SLES	`$ sudo zypper clean --all`
Ubuntu or Debian	`$ sudo apt-get clean`

3. Install the Oracle JDK package from the Cloudera Manager repository. Install the same version as is used on other cluster hosts. Both JDK 1.7 and 1.8 are supported:

Table 2: JDK 1.7

OS	Command
RHEL	```$ sudo yum install oracle-j2sdk1.7```
SLES	```$ sudo zypper install oracle-j2sdk1.7```
Ubuntu or Debian	```$ sudo apt-get install oracle-j2sdk1.7```

Table 3: JDK 1.8

OS	Command
RHEL	```$ sudo yum install jdk1.8.0_144-cloudera```
SLES	```$ sudo zypper install jdk1.8.0_144-cloudera```
Ubuntu or Debian	```$ sudo apt-get install jdk1.8.0_144-cloudera```

> **Note:** If you need to install the JCE unlimited strength encryption policy files, these files are not included in the JDK package and Cloudera Manager does not install them. Copy the files from an existing host to the new host. For example:
>
> ```
> # scp
> mynode.example.com:/usr/java/jdk1.7.0_67-cloudera/jre/lib/security/*policy.jar
>   /usr/java/jdk1.7.0_67-cloudera/jre/lib/security/
> ```

4. Set up the TLS certificates using the same procedure that was used to set them up on other cluster hosts. See Configuring TLS Encryption for Cloudera Manager. If you have set up a custom truststore (For example, `/usr/java/jdk1.7.0_67-cloudera/jre/lib/security/jssecacerts`, copy that file from an existing host to the same location on the new host.

5. Install the Cloudera Manager Agent:

OS	Command
RHEL	```$ sudo yum install cloudera-manager-agent```
SLES	```$ sudo zypper install cloudera-manager-agent```
Ubuntu or Debian	```$ sudo apt-get install cloudera-manager-agent```

6. Copy the Cloudera Manager Agent configuration file from an existing cluster host that is already configured for TLS to the same location on the new host. For example:

```
$ sudo scp mynode.example.com:/etc/cloudera-scm-agent/config.ini
/etc/cloudera-scm-agent/config.ini
```

7. Create and secure the file containing the password used to protect the private key of the Agent:

a. Use a text editor to create a file called `agentkey.pw` that contains the password. Save the file in the `/etc/cloudera-scm-agent` directory.

b. Change ownership of the file to `root`:

```
$ sudo chown root:root /etc/cloudera-scm-agent/agentkey.pw
```

c. Change the permissions of the file:

```
$ sudo chmod 440 /etc/cloudera-scm-agent/agentkey.pw
```

8. Start the Agent on the new host:

```
$ sudo service cloudera-scm-agent start
```

9. Log in to Cloudera Manager and go to **Hosts** > **All Hosts** page and verify that the new host is recognized by Cloudera Manager.

Using the Add Hosts Wizard

1. Click the **Hosts** tab.
2. Click the **Add New Hosts** button.
3. Follow the instructions in the wizard to install the Oracle JDK and Cloudera Manager Agent packages and start the Agent.
4. In the **Specify hosts for your CDH Cluster installation** page, you can search for new hosts to add under the **New Hosts** tab. However, if you have hosts that are already known to Cloudera Manager but have no roles assigned, (for example, a host that was previously in your cluster but was then removed) these will appear under the **Currently Managed Hosts** tab.
5. You will have an opportunity to add (and start) role instances to your newly-added hosts using a host template.

    a. You can select an existing host template, or create a new one.
    b. To create a new host template, click the **+ Create...** button. This will open the **Create New Host Template** pop-up. See Host Templates on page 180 for details on how you select the role groups that define the roles that should run on a host. When you have created the template, it will appear in the list of host templates from which you can choose.
    c. Select the host template you want to use.
    d. By default Cloudera Manager will automatically start the roles specified in the host template on your newly added hosts. To prevent this, uncheck the option to start the newly-created roles.

6. When the wizard is finished, you can verify the Agent is connecting properly with the Cloudera Manager Server by clicking the **Hosts** tab and checking the health status for the new host. If the Health Status is **Good** and the value for the Last Heartbeat is recent, then the Agent is connecting properly with the Cloudera Manager Server.

If you did not specify a host template during the Add Hosts wizard, then no roles will be present on your new hosts until you add them. You can do this by adding individual roles under the **Instances** tab for a specific service, or by using a host template. See Role Instances on page 209 for information about adding roles for a specific service. See Host Templates on page 180 to create a host template that specifies a set of roles (from different services) that should run on a host.

Enable TLS Encryption or Authentication

If you previously enabled TLS security on your cluster, you must re-enable the TLS options on the **Administration** page and also configure TLS on each new host after using the Add Hosts wizard. Otherwise, you can ignore this step. For instructions, see Configuring TLS Encryption for Cloudera Manager.

Enable TLS/SSL for CDH Components

If you have previously enabled TLS/SSL on your cluster, and you plan to start these roles on this new host, make sure you install a new host certificate to be configured from the same path and naming convention as the rest of your hosts. Since the new host and the roles configured on it are inheriting their configuration from the previous host, ensure that the keystore or truststore passwords and locations are the same on the new host. For instructions on configuring TLS/SSL, see Configuring TLS/SSL Encryption for CDH Services.

Enable Kerberos

If you have previously enabled Kerberos on your cluster:

1. Install the packages required to `kinit` on the new host (see the list in Before you Begin Using the Wizard).
2. If you have set up Cloudera Manager to manage `krb5.conf`, it will automatically deploy the file on the new host. Note that Cloudera Manager will deploy `krb5.conf` only if you use the Kerberos wizard. If you have used the API, you will need to manually perform the commands that the wizard calls.

   If Cloudera Manager does not manage `krb5.conf`, you must manually update the file at `/etc/krb5.conf`.

Adding a Host by Installing the Packages Using Your Own Method

If you used a different mechanism to install the Oracle JDK, CDH, Cloudera Manager Agent packages, you can use that same mechanism to install the Oracle JDK, CDH, Cloudera Manager Agent packages and then start the Cloudera Manager Agent.

1. Install the Oracle JDK, CDH, and Cloudera Manager Agent packages using your own method. For instructions on installing these packages, see Installing Cloudera Manager, CDH, and Managed Services.
2. After installation is complete, start the Cloudera Manager Agent. For instructions, see Starting, Stopping, and Restarting Cloudera Manager Agents on page 32.
3. After the Agent is started, you can verify the Agent is connecting properly with the Cloudera Manager Server by clicking the **Hosts** tab and checking the health status for the new host. If the Health Status is **Good** and the value for the Last Heartbeat is recent, then the Agent is connecting properly with the Cloudera Manager Server.
4. If you have enabled TLS security on your cluster, you must enable and configure TLS on each new host. Otherwise, ignore this step.

   a. Enable and configure TLS on each new host by specifying `1` for the `use_tls` property in the `/etc/cloudera-scm-agent/config.ini` configuration file.
   b. Configure TLS security on the new hosts by following the instructions in Configuring TLS Encryption for Cloudera Manager.

5. If you have previously enabled TLS/SSL on your cluster, and you plan to start these roles on this new host, make sure you install a new host certificate to be configured from the same path and naming convention as the rest of your hosts. Since the new host and the roles configured on it are inheriting their configuration from the previous host, ensure that the keystore or truststore passwords and locations are the same on the new host. For instructions on configuring TLS/SSL, see Configuring TLS/SSL Encryption for CDH Services.

6. If you have previously enabled Kerberos on your cluster:

   1. Install the packages required to `kinit` on the new host (see the list in Before you Begin Using the Wizard).
   2. If you have set up Cloudera Manager to manage `krb5.conf`, it will automatically deploy the file on the new host. Note that Cloudera Manager will deploy `krb5.conf` only if you use the Kerberos wizard. If you have used the API, you will need to manually perform the commands that the wizard calls.

      If Cloudera Manager does not manage `krb5.conf`, you must manually update the file at `/etc/krb5.conf`.

Specifying Racks for Hosts

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

To get maximum performance, it is important to configure CDH so that it knows the topology of your network. Network locations such as hosts and racks are represented in a tree, which reflects the network "distance" between locations. HDFS will use the network location to be able to place block replicas more intelligently to trade off performance and resilience. When placing jobs on hosts, CDH will prefer within-rack transfers (where there is more bandwidth available) to off-rack transfers; the MapReduce and YARN schedulers use network location to determine where the closest replica is as input to a map task. These computations are performed with the assistance of rack awareness scripts.

Cloudera Manager includes internal rack awareness scripts, but you must specify the racks where the hosts in your cluster are located. If your cluster contains more than 10 hosts, Cloudera recommends that you specify the rack for each host. HDFS, MapReduce, and YARN will automatically use the racks you specify.

Cloudera Manager supports nested rack specifications. For example, you could specify the rack `/rack3`, or `/group5/rack3` to indicate the third rack in the fifth group. All hosts in a cluster must have the *same number* of path components in their rack specifications.

To specify racks for hosts:

1. Click the **Hosts** tab.
2. Check the checkboxes next to the host(s) for a particular rack, such as all hosts for `/rack123`.
3. Click **Actions for Selected (*n*)** > **Assign Rack**, where *n* is the number of selected hosts.
4. Enter a rack name or ID that starts with a slash `/`, such as `/rack123` or `/aisle1/rack123`, and then click **Confirm**.
5. Optionally [restart affected services](). Rack assignments are not automatically updated for running services.

Host Templates

Minimum Required Role: [Full Administrator]()

Host templates let you designate a set of role groups that can be applied in a single operation to a host or a set of hosts. This significantly simplifies the process of configuring new hosts when you need to expand your cluster.

> **Important:** A host template can only be applied on a host with a version of CDH that matches the CDH version running on the cluster to which the host template belongs.

You can create and manage host templates under the Templates tab from the Hosts page.

1. Click the **Hosts** tab on the main Cloudera Manager navigation bar.
2. Click the **Templates** tab on the Hosts page.

Templates are not required; Cloudera Manager assigns roles and role groups to the hosts of your cluster when you perform the initial cluster installation. However, if you want to add new hosts to your cluster, a host template can make this much easier.

If there are existing host templates, they are listed on the page, along with links to each role group included in the template.

If you are managing multiple clusters, you must create separate host templates for each cluster, as the templates specify role configurations specific to the roles in a single cluster. Existing host templates are listed under the cluster to which they apply.

- You can click a role group name to be taken to the Edit configuration page for that role group, where you can modify the role group settings.
- From the **Actions** menu associated with the template you can edit the template, clone it, or delete it.

Creating a Host Template

1. From the **Templates** tab, click **Click here**
2. In the **Create New Host Template** pop-up window that appears:
   - Type a name for the template.

- For each role, select the appropriate role group. There may be multiple role groups for a given role type — you want to select the one with the configuration that meets your needs.

3. Click **Create** to create the host template.

Editing a Host Template

1. From the **Hosts** tab, click the **Templates** tab.
2. Pull down the **Actions** menu for the template you want to modify, and click **Edit**. This put you into the **Edit Host Template** pop-up window. This works exactly like the **Create New Host Template** window — you can modify they template name or any of the role group selections.
3. Click **OK** when you have finished.

Applying a Host Template to a Host

You can use a host template to apply configurations for multiple roles in a single operation.

You can apply a template to a host that has no roles on it, or that has roles from the same services as those included in the host template. New roles specified in the template that do not already exist on the host will be added. A role on the host that is already a member of the role group specified in the template will be left unchanged. If a role on the host matches a role in the template, but is a member of a different role group, it will be moved to the role group specified by the template.

For example, suppose you have two role groups for a DataNode (DataNode Default Group and DataNode (1)). The host has a DataNode role that belongs to DataNode Default Group. If you apply a host template that specifies the DataNode (1) group, the role on the host will be moved from DataNode Default Group to DataNode (1).

However, if you have two instances of a service, such as MapReduce (for example, *mr1* and *mr2*) and the host has a TaskTracker role from service *mr2*, you cannot apply a TaskTracker role from service *mr1*.

A host may have no roles on it if you have just added the host to your cluster, or if you decommissioned a managed host and removed its existing roles.

Also, the host must have the same version of CDH installed as is running on the cluster whose host templates you are applying.

If a host belongs to a different cluster than the one for which you created the host template, you can apply the host template if the "foreign" host either has no roles on it, or has only management roles on it. When you apply the host template, the host will then become a member of the cluster whose host template you applied. The following instructions assume you have already created the appropriate host template.

1. Go to the **Hosts** page, **Status** tab.
2. Select the host(s) to which you want to apply your host template.
3. From the **Actions for Selected** menu, select **Apply Host Template**.
4. In the pop-up window that appears, select the host template you want to apply.
5. Optionally you can have Cloudera Manager start the roles created per the host template – check the box to enable this.
6. Click **Confirm** to initiate the action.

Performing Maintenance on a Cluster Host

You can perform minor maintenance on cluster hosts by using Cloudera Manager to manage the host decommission and recommission process. In this process, you can specify whether to suppress alerts from the decommissioned host and, for hosts running the DataNode role, you can specify whether or not to replicate under-replicated data blocks to other DataNodes to maintain the cluster's replication factor. This feature is useful when performing minor maintenance on cluster hosts, such as adding memory or changing network cards or cables where the maintenance window is expected to be short and the extra cluster resources consumed by replicating missing blocks is undesirable.

You can also place hosts into [Maintenance Mode](#), which suppresses unneeded alerts during a maintenance window but does not decommission the hosts.

To perform host maintenance on cluster hosts:

1. [Decommission the hosts](#).
2. Perform the necessary maintenance on the hosts.
3. [Recommission the hosts](#).

Decommissioning Hosts

Minimum Required Role: [Limited Operator](#) (also provided by **Operator**, **Configurator**, **Cluster Administrator**, or **Full Administrator**)

Note that the **Limited Operator** and **Operator** roles do not allow you to suppress or enable alerts.

> **Note:** Hosts with DataNodes and DataNode roles themselves can only be decommissioned if the resulting action leaves enough DataNodes commissioned to maintain the configured HDFS replication factor (by default 3). If you attempt to decommission a DataNode or a host with a DataNode in such situations, the decommission process will not complete and must be aborted.

Cloudera Manager manages the host decommission and recommission process and allows you the option to specify whether to replicate the data to other DataNodes, and whether or not to suppress alerts.

Decommissioning a host decommissions and stops all roles on the host without requiring you to individually decommission the roles on each service. Decommissioning applies to only to HDFS DataNode, MapReduce TaskTracker, YARN NodeManager, and HBase RegionServer roles. If the host has other roles running on it, those roles are stopped.

To decommission one or more hosts:

1. If the host has a DataNode, and you are planning to replicate data to other hosts (for longer term maintenance operations or to permanently decommission or repurpose the host), perform the steps in [Tuning HDFS Prior to Decommissioning DataNodes](#) on page 184.
2. In Cloudera Manager, select the cluster where you want to decommission hosts.
3. Click **Hosts** > **All Hosts**.
4. Select the hosts that you want to decommission.
5. Select **Actions for Selected** > **Begin Maintenance (Suppress Alerts/Decommission**.

   (If you are logged in as a user with the **Limited Operator** or **Operator** role, the menu item is labeled **Decommission Host(s)** and you will not see the option to suppress alerts.)

   The **Begin Maintenance (Suppress Alerts/Decommission)** dialog box opens. The role instances running on the hosts display at the top.

6. To decommission the hosts and suppress alerts, select **Decommission Host(s)**. When you select this option for hosts running a DataNode role, choose one of the following (if the host is not running a DataNode role, you will only see the Decommission Host(s) option:):

   - **Decommission DataNodes**

     This option re-replicates data to other DataNodes in the cluster according to the configured replication factor. Depending on the amount of data and other factors, this can take a significant amount of time and uses a great deal of network bandwidth. This option is appropriate when replacing disks, repurposing hosts for non-HDFS use, or permanently retiring hardware.

   - **Take DataNode Offline**

     This option does not re-replicate HDFS data to other DataNodes until the amount of time you specify has passed, making it less disruptive to active workloads. After this time has passed, the DataNode is automatically recommissioned, but the DataNode role is not started. This option is appropriate for short-term maintenance tasks such not involving disks, such as rebooting, CPU/RAM upgrades, or switching network cables.

7. Click **Begin Maintenance**.

   The **Host Decommission Command** dialog box opens and displays the progress of the command.

   > **Note:**
   >
   > - You cannot start roles on a decommissioned host.
   > - When a DataNode is decommissioned, although HDFS data is replicated to other DataNodes, local files containing the original data blocks are not automatically removed from the storage directories on the host. If you want to permanently remove these files from the host to reclaim disk space, you must do so manually.

Recommissioning Hosts

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

Only hosts that are decommissioned using Cloudera Manager can be recommissioned.

1. In Cloudera Manager, select the cluster where you want to recommission hosts.
2. Click **Hosts** > **All Hosts**.
3. Select the hosts that you want to recommission.
4. Select **Actions for Selected** > **End Maintenance (Suppress Alerts/Decommission**.

   The **End Maintenance (Suppress Alerts/Decommission** dialog box opens. The role instances running on the hosts display at the top.

5. To recommission the hosts, select **Recommission Host(s)**.
6. Choose one of the following:

   - **Bring hosts online and start all roles**

     All decommissioned roles will be recommissioned and started. HDFS DataNodes will be started first and brought online before decommissioning to avoid excess replication.

   - **Bring hosts online**

     All decommissioned roles will be recommissioned but remain stopped. You can restart the roles later.

7. Click **End Maintenance**.

   The **Recommission Hosts and Start Roles Command** dialog box opens and displays the progress of recommissioning the hosts and restarting the roles.

Stopping All the Roles on a Host

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

1. Click the **Hosts** tab.
2. Select one or more hosts on which to stop all roles.
3. Select **Actions for Selected** > **Stop Roles on Hosts**.

Starting All the Roles on a Host

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

1. Click the **Hosts** tab.
2. Select one or more hosts on which to start all roles.
3. Select **Actions for Selected** > **Start Roles on Hosts**.

Tuning and Troubleshooting Host Decommissioning

Decommissioning a host decommissions and stops all roles on the host without requiring you to individually decommission the roles on each service. The decommissioning process can take a long time and uses a great deal of cluster resources, including network bandwidth. You can tune the decommissioning process to improve performance and mitigate the performance impact on the cluster.

You can use the Decommission and Recommission features to perform minor maintenance on cluster hosts using Cloudera Manager to manage the process. See Performing Maintenance on a Cluster Host on page 181.

Tuning HDFS Prior to Decommissioning DataNodes

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

When a DataNode is decommissioned, the NameNode ensures that every block from the DataNode will still be available across the cluster as dictated by the replication factor. This procedure involves copying blocks from the DataNode in small batches. If a DataNode has thousands of blocks, decommissioning can take several hours. Before decommissioning hosts with DataNodes, you should first tune HDFS:

1. Run the following command to identify any problems in the HDFS file system:

```
hdfs fsck / -list-corruptfileblocks -openforwrite -files -blocks -locations 2>&1 >
/tmp/hdfs-fsck.txt
```

2. Fix any issues reported by the `fsck` command. If the command output lists corrupted files, use the `fsck` command to move them to the `lost+found` directory or delete them:

```
hdfs fsck file_name -move
```

    or

```
hdfs fsck file_name -delete
```

3. Raise the heap size of the DataNodes. DataNodes should be configured with at least 4 GB heap size to allow for the increase in iterations and max streams.

   a. Go to the HDFS service page.
   b. Click the **Configuration** tab.
   c. Select **Scope** > **DataNode**.
   d. Select **Category** > **Resource Management**.
   e. Set the **Java Heap Size of DataNode in Bytes** property as recommended.

      To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

   f. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

4. Increase the replication work multiplier per iteration to a larger number (the default is 2, however 10 is recommended):

   a. Select **Scope** > **NameNode**.
   b. Expand the **Category** > **Advanced** category.
   c. Configure the **Replication Work Multiplier Per Iteration** property to a value such as 10.

      To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

   d. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

5. Increase the replication maximum threads and maximum replication thread hard limits:

   a. Select **Scope** > **NameNode**.

 b. Expand the **Category** > **Advanced** category.

 c. Configure the **Maximum number of replication threads on a DataNode** and **Hard limit on the number of replication threads on a DataNode** properties to 50 and 100 respectively. You can decrease the number of threads (or use the default values) to minimize the impact of decommissioning on the cluster, but the trade off is that decommissioning will take longer.

 To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.

 d. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

 6. Restart the HDFS service.

For additional tuning recommendations, see Performance Considerations on page 185.

Tuning HBase Prior to Decommissioning DataNodes

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

To increase the speed of a rolling restart of the HBase service, set the **Region Mover Threads** property to a higher value. This increases the number of regions that can be moved in parallel, but places additional strain on the HMaster. In most cases, **Region Mover Threads** should be set to 5 or lower.

Performance Considerations

Decommissioning a DataNode does not happen instantly because the process requires replication of a potentially large number of blocks. During decommissioning, the performance of your cluster may be impacted. This section describes the decommissioning process and suggests solutions for several common performance issues.

Decommissioning occurs in two steps:

 1. The **Commission State** of the DataNode is marked as **Decommissioning** and the data is replicated from this node to other available nodes. Until all blocks are replicated, the node remains in a **Decommissioning** state. You can view this state from the NameNode Web UI. (Go to the HDFS service and select **Web UI** > **NameNode Web UI**.)

 2. When all data blocks are replicated to other nodes, the node is marked as **Decommissioned**.

Decommissioning can impact performance in the following ways:

- There must be enough disk space on the other active DataNodes for the data to be replicated. After decommissioning, the remaining active DataNodes have more blocks and therefore decommissioning these DataNodes in the future may take more time.
- There will be increased network traffic and disk I/O while the data blocks are replicated.
- Data balance and data locality can be affected, which can lead to a decrease in performance of any running or submitted jobs.
- Decommissioning a large numbers of DataNodes at the same time can decrease performance.
- If you are decommissioning a minority of the DataNodes, the speed of data reads from these nodes limits the performance of decommissioning because decommissioning maxes out network bandwidth when reading data blocks from the DataNode and spreads the bandwidth used to replicate the blocks among other DataNodes in the cluster. To avoid performance impacts in the cluster, Cloudera recommends that you only decommission a minority of the DataNodes at the same time.
- You can decrease the number of replication threads to decrease the performance impact of the replications, but this will cause the decommissioning process to take longer to complete. See Tuning HDFS Prior to Decommissioning DataNodes on page 184.

Cloudera recommends that you add DataNodes and decommission DataNodes in parallel, in smaller groups. For example, if the replication factor is 3, then you should add two DataNodes and decommission two DataNodes at the same time.

Troubleshooting Performance of Decommissioning

The following conditions can also impact performance when decommissioning DataNodes:

- Open Files on page 186

-

Open Files

Write operations on the DataNode do not involve the NameNode. If there are blocks associated with open files located on a DataNode, they are not relocated until the file is closed. This commonly occurs with:

- Clusters using HBase
- Open Flume files
- Long running tasks

To find open files, run the following command:

```
hdfs dfsadmin -listOpenFiles -blockingDecommission
```

The command returns output similar to the following example:

```
Client Host          Client Name          Open File Path
172.26.12.77         DFSClient_NONMAPREDUCE_-698274460_1
/hbase/oldWALs/dn3.cloudera.com%2C22101%2C1540973344249.dn3.cloudera.com%2C22101%2C1540973344249.regiongroup-0.154099857098
```

After you find the open files, perform the appropriate action to restart process to close the file. For example, `major compaction` closes all files in a region for HBase.

Alternatively, you may evict writers to those decommissioning DataNodes with the following command:

```
hdfs dfsadmin -evictWriters <datanode_host:ipc_port>
```

For example:

```
hdfs dfsadmin -evictWriters datanode1:20001
```

A block cannot be relocated because there are not enough DataNodes to satisfy the block placement policy.

For example, for a 10 node cluster, if the `mapred.submit.replication` is set to the default of 10 while attempting to decommission one DataNode, there will be difficulties relocating blocks that are associated with map/reduce jobs.   This condition will lead to errors in the NameNode logs similar to the following:

```
org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyDefault: Not able to
 place enough replicas, still in need of 3 to reach 3
```

Use the following steps to find the number of files where the block replication policy is equal to or above your current cluster size:

1. Provide a listing of open files, their blocks, the locations of those blocks by running the following command:

```
hadoop fsck / -files -blocks -locations -openforwrite 2>&1 > openfiles.out
```

2. Run the following command to return a list of how many files have a given replication factor:

```
grep repl= openfiles.out | awk '{print $NF}' | sort | uniq -c
```

For example, when the replication factor is 10 , and decommissioning one:

```
egrep -B4 "repl=10" openfiles.out | grep -v '<dir>' | awk '/^\//{print $1}'
```

3. Examine the paths, and decide whether to reduce the replication factor of the files, or remove them from the cluster.

Maintenance Mode

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

Maintenance mode allows you to suppress alerts for a host, service, role, or an entire cluster. This can be useful when you need to take actions in your cluster (make configuration changes and restart various elements) and do not want to see the alerts that will be generated due to those actions.

Putting an entity into maintenance mode does not prevent events from being logged; it only suppresses the alerts that those events would otherwise generate. You can see a history of all the events that were recorded for entities during the period that those entities were in maintenance mode.

Explicit and Effective Maintenance Mode

When you enter maintenance mode on an entity (cluster, service, or host) that has subordinate entities (for example, the roles for a service) the subordinate entities are also put into maintenance mode. These are considered to be in *effective maintenance mode*, as they have inherited the setting from the higher-level entity.

For example:

- If you set the HBase service into maintenance mode, then its roles (HBase Master and all RegionServers) are put into effective maintenance mode.
- If you set a host into maintenance mode, then any roles running on that host are put into effective maintenance mode.

Entities that have been explicitly put into maintenance mode show the icon . Entities that have entered effective maintenance mode as a result of inheritance from a higher-level entity show the icon .

When an entity (role, host or service) is in effective maintenance mode, it can only be removed from maintenance mode when the higher-level entity exits maintenance mode. For example, if you put a service into maintenance mode, the roles associated with that service are entered into effective maintenance mode, and remain in effective maintenance mode until the service exits maintenance mode. You cannot remove them from maintenance mode individually.

Alternatively, an entity that is in effective maintenance mode can be put into explicit maintenance mode. In this case, the entity remains in maintenance mode even when the higher-level entity exits maintenance mode. For example, suppose you put a host into maintenance mode, (which puts all the roles on that host into effective maintenance mode). You then select one of the roles on that host and put it explicitly into maintenance mode. When you have the host exit maintenance mode, that one role remains in maintenance mode. You need to select it individually and specifically have it exit maintenance mode.

Viewing Maintenance Mode Status

You can view the status of Maintenance Mode in your cluster by clicking

to the right of the cluster name and selecting **View Maintenance Mode Status**.

Entering Maintenance Mode

You can enable maintenance mode for a cluster, service, role, or host.

Putting a Cluster into Maintenance Mode

1.

   to the right of the cluster name and select **Enter Maintenance Mode**.
2. Confirm that you want to do this.

The cluster is put into explicit maintenance mode, as indicated by the icon. All services and roles in the cluster are entered into effective maintenance mode, as indicated by the icon.

Putting a Service into Maintenance Mode

1. Go to the service page in Cloudera Manager.

2. Click **Actions** > **Enter Maintenance Mode**.

3. Confirm that you want to do this.

The service is put into explicit maintenance mode, as indicated by the icon. All roles for the service are entered into effective maintenance mode, as indicated by the icon.

Putting Roles into Maintenance Mode

1. Go to the service page that includes the role.

2. Go to the **Instances** tab.

3. Select the role(s) you want to put into maintenance mode.

4. From the **Actions for Selected** menu, select **Enter Maintenance Mode**.

5. Confirm that you want to do this.

The roles will be put in explicit maintenance mode. If the roles were already in effective maintenance mode (because its service or host was put into maintenance mode) the roles will now be in explicit maintenance mode. This means that they will not exit maintenance mode automatically if their host or service exits maintenance mode; they must be explicitly removed from maintenance mode.

Putting Hosts into Maintenance Mode

1. In Cloudera Manager, select the cluster where you want to decommission hosts.

2. Click **Hosts** > **All Hosts**.

3. Select the hosts that you want to put into Maintenance Mode.

4. Select **Actions for Selected** > **Begin Maintenance (Suppress Alerts/Decommission**.

The **Begin Maintenance (Suppress Alerts/Decommission)** dialog box opens. The role instances running on the hosts display at the top. You can also use this dialog box to decommission the host. See Tuning and Troubleshooting Host Decommissioning on page 184.

5. Deselect the **Decommission Host(s)** option to put the host into Maintenance Mode. In this mode, alerts from the hosts are suppressed until the host exits Maintenance Mode. The events, however, are still logged. Hosts that are currently in Maintenance Mode display the icon.

6. Click **Begin Maintenance**.

The **Host Decommission Command** dialog box opens and displays the progress of the command.

Exiting Maintenance Mode

When you exit maintenance mode, the maintenance mode icons are removed and alert notification resumes.

Exiting a Cluster from Maintenance Mode

1.

to the right of the cluster name and select **Exit Maintenance Mode**.

2. Confirm that you want to do this.

Exiting a Service from Maintenance Mode

1.

to the right of the service name and select **Exit Maintenance Mode**.

2. Confirm that you want to do this.

Exiting Roles from Maintenance Mode

1. Go to the services page that includes the role.

2. Go to the **Instances** tab.
3. Select the role(s) you want to exit from maintenance mode.
4. From the **Actions for Selected** menu, select **Exit Maintenance Mode**.
5. Confirm that you want to do this.

Taking Hosts out of Maintenance Mode

1. In Cloudera Manager, to go the cluster with the hosts you want to take out of Maintenance Mode.
2. Click **Hosts** > **All Hosts**.
3. Select the hosts that are ready to exit Maintenance Mode.
4. Select **Actions for Selected** > **End Maintenance (Suppress Alerts/Decommission**.

   The **End Maintenance (Suppress Alerts/Decommission)** dialog box opens. The role instances running on the hosts display at the top.

5. Deselect the **Recommission Host(s)** option to take the host out of Maintenance Mode and re-enable alerts from

   the hosts. Hosts that are currently in Maintenance Mode display the ⚙ icon on the **All Hosts** page.
6. Click **End Maintenance**.

Changing Hostnames

Minimum Required Role: Full Administrator

> ❗ **Important:**
> - The process described here requires Cloudera Manager and cluster downtime.
> - If any user created scripts reference specific hostnames those must also be updated.
> - Due to the length and complexity of the following procedure, changing cluster hostnames is not recommended by Cloudera.

After you have installed Cloudera Manager and created a cluster, you may need to update the names of the hosts running the Cloudera Manager Server or cluster services. To update a deployment with new hostnames, follow these steps:

1. Verify if TLS/SSL certificates have been issued for any of the services and make sure to create new TLS/SSL certificates in advance for services protected by TLS/SSL. See Encryption Mechanisms Overview.
2. Export the Cloudera Manager configuration using one of the following methods:

   - Open a browser and go to this URL `http://cm_hostname:7180/api/api_version/cm/deployment`. Save the displayed configuration.
   - From terminal type:

     ```
     $ curl -u admin:admin http://cm_hostname:7180/api/api_version/cm/deployment >
     cme-cm-export.json
     ```

     If Cloudera Manager SSL is in use, specify the `-k` switch:

     ```
     $ curl -k -u admin:admin http://cm_hostname:7180/api/api_version/cm/deployment >
     cme-cm-export.json
     ```

   where *cm_hostname* is the name of the Cloudera Manager host and *api_version* is the correct version of the API for the version of Cloudera Manager you are using. For example,
   `http://tcdn5-1.ent.cloudera.com:7180/api/v30/cm/deployment`.
3. Stop all services on the cluster.
4. Stop the Cloudera Management Service.
5. Stop the Cloudera Manager Server.

6. Stop the Cloudera Manager Agents on the hosts that will be having the hostname changed.
7. Back up the Cloudera Manager Server database using `mysqldump`, `pg_dump`, or another preferred backup utility. Store the backup in a safe location.
8. Update names and principals:

   a. Update the target hosts using standard per-OS/name service methods (`/etc/hosts`, `dns`, `/etc/sysconfig/network`, `hostname`, and so on). Ensure that you remove the old hostname.

   b. If you are changing the hostname of the host running Cloudera Manager Server do the following:

      a. Change the hostname per step 8.a.
      b. Update the Cloudera Manager hostname in `/etc/cloudera-scm-agent/config.ini` on all Agents.

   c. If the cluster is configured for Kerberos security, do the following:

      a. Remove the old hostname cluster principals.

         • If you are using an MIT KDC, remove old hostname cluster service principals from the KDC database using one of the following:

            – Use the `delprinc` command within `kadmin.local` interactive shell.

               OR

            – From the command line:

```
kadmin.local -q "listprincs" | grep -E
"(HTTP|hbase|hdfs|hive|httpfs|hue|impala|mapred|solr|oozie|yarn|zookeeper)[^/]*/[^/]*@"
 > cluster-princ.txt
```

            Open `cluster-princ.txt` and remove any noncluster service principal entries. Make sure that the default `krbtgt` and other principals you created, or that were created by Kerberos by default, are not removed by running the following: `for i in `cat cluster-princ.txt`; do yes yes | kadmin.local -q "delprinc $i"; done.`

         • For an Active Directory KDC, an AD administrator must manually delete the principals for the old hostname from Active Directory.

      b. Start the Cloudera Manager database and Cloudera Manager Server.
      c. Start the Cloudera Manager Agents on the newly renamed hosts. The Agents should show a current heartbeat in Cloudera Manager.
      d. Within the Cloudera Manager Admin Console click the **Hosts** tab.
      e. Select the checkbox next to the host with the new name.
      f. Select **Actions** > **Regenerate Keytab**.

9. If one of the hosts that was renamed has a NameNode configured with high availability and automatic failover enabled, reconfigure the ZooKeeper Failover Controller znodes to reflect the new hostname.

   a. Start ZooKeeper Servers.

   > ⚠ **Warning:** All other services, and most importantly HDFS, and the ZooKeeper Failover Controller (FC) role within the HDFS, should not be running.

   b. On one of the hosts that has a ZooKeeper Server role, run `zookeeper-client`.

      a. If the cluster is configured for Kerberos security, configure ZooKeeper authorization as follows:

         a. Go to the HDFS service.
         b. Click the **Instances** tab.
         c. Click the **Failover Controller** role.
         d. Click the **Process** tab.

    e. In the Configuration Files column of the `hdfs/hdfs.sh ["zkfc"]` program, expand **Show**.

    f. Inspect `core-site.xml` in the displayed list of files and determine the value of the `ha.zookeeper.auth` property, which will be something like: `digest:hdfs-fcs:TEbW2bgoODa96rO3ZTn7ND5fSOGx0h`. The part after `digest:hdfs-fcs:` is the password (in the example it is `TEbW2bgoODa96rO3ZTn7ND5fSOGx0h`)

    g. Run the `addauth` command with the password:

```
addauth digest hdfs-fcs:TEbW2bgoODa96rO3ZTn7ND5fSOGx0h
```

    b. Verify that the HA znode exists: `ls /hadoop-ha`.

    c. Delete the HDFS znode: `rmr /hadoop-ha/nameservice1`.

    d. If you *are not* running JobTracker in a high availability configuration, delete the HA znode: `rmr /hadoop-ha`.

  c. In the Cloudera Manager Admin Console, go to the HDFS service.

  d. Click the **Instances** tab.

  e. Select **Actions** > **Initialize High Availability State in ZooKeeper...**.

10. Update the Hive metastore:

  a. Back up the Hive metastore database.

  b. In the Cloudera Manager Admin Console, go to the Hive service.

  c. Select **Actions** > **Update Hive Metastore NameNodes** and confirm the command.

11. Update the **Database Hostname** property for each of the cluster roles for which a database is located on the host being renamed. This is required for both Cloudera Management Service roles (Reports Manager, Activity Monitor, Navigator Audit and Metadata Server) and for cluster services such as Hue, Hive, and so on.

12. Start all cluster services.

13. Start the Cloudera Management Service.

14. Deploy client configurations.

Deleting Hosts

Minimum Required Role: [Full Administrator](#)

You can remove a host from a cluster in two ways:

- Delete the host entirely from Cloudera Manager.
- Remove a host from a cluster, but leave it available to other clusters managed by Cloudera Manager.

Both methods [decommission the hosts](#), delete roles, and remove managed service software, but preserve data directories.

Deleting a Host from Cloudera Manager

1. In the Cloudera Manager Admin Console, click the **Hosts** tab.
2. Select the hosts to delete.
3. Select **Actions for Selected** > **Begin Maintenance**.
4. Stop the Agent on the host. For instructions, see [Starting, Stopping, and Restarting Cloudera Manager Agents](#) on page 32.
5. In the Cloudera Manager Admin Console, click the **Hosts** tab.
6. Reselect the hosts you selected in [step 2](#).
7. Select **Actions for Selected** > **Remove from Cloudera Manager**.

Removing a Host From a Cluster

This procedure leaves the host managed by Cloudera Manager and preserves the Cloudera Management Service roles (such as the Events Server, Activity Monitor, and so on).

1. In the Cloudera Manager Admin Console, click the **Hosts** tab.
2. Select the hosts to delete.
3. Select **Actions for Selected** > **Remove From Cluster**. The Remove Hosts From Cluster dialog box displays.
4. Leave the selections to decommission roles and skip removing the Cloudera Management Service roles. Click **Confirm** to proceed with removing the selected hosts.

Moving a Host Between Clusters

Minimum Required Role: Full Administrator

Moving a host between clusters can be accomplished by:

1. Decommissioning the host (see Decommissioning Role Instances on page 210).
2. Removing all roles from the host (except for the Cloudera Manager management roles). See Deleting Role Instances on page 211.
3. Deleting the host from the cluster (see Deleting Hosts on page 191), specifically the section on removing a host from a cluster but leaving it available to Cloudera Manager.
4. Adding the host to the new cluster (see Adding a Host to the Cluster on page 174).
5. Adding roles to the host (optionally using one of the host templates associated with the new cluster). See Adding a Role Instance on page 209 and Host Templates on page 180.

Managing Services

Cloudera Manager service configuration features let you manage the deployment and configuration of CDH and managed services. You can add new services and roles if needed, gracefully start, stop and restart services or roles, and decommission and delete roles or services if necessary. Further, you can modify the configuration properties for services or for individual role instances. If you have a Cloudera Enterprise license, you can view past configuration changes and roll back to a previous revision. You can also generate client configuration files, enabling you to easily distribute them to the users of a service.

The topics in this chapter describe how to configure and use the services on your cluster. Some services have unique configuration requirements or provide unique features: those are covered in .

Adding a Service

Minimum Required Role: Full Administrator

> ⓘ **Important:** As of February 1, 2021, all downloads of CDH and Cloudera Manager require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository contents described below. You may need to upgrade Cloudera Manager to a newer version that uses the modified URLs.
>
> This can affect new installations, upgrades, adding new hosts to a cluster, adding a cluster and adding new services to a cluster.
>
> For more information, see Updating an existing CDH/Cloudera Manager deployment to access downloads with authentication.

After initial installation, you can use the **Add a Service** wizard to add and configure new service instances. For example, you may want to add a service such as Oozie that you did not select in the wizard during the initial installation.

The binaries for the following services are not packaged in CDH and must be installed individually before being adding the service:

Service	Installation Documentation
Accumulo	Apache Accumulo Documentation
Key Trustee KMS	Installing Key Trustee KMS

If you do not add the binaries before adding the service, the service will fail to start.

To add a service:

1. On the **Home** > **Status** tab, click

   ▾

   to the right of the cluster name and select **Add a Service**. A list of service types display. You can add one type of service at a time.
2. Select a service and click **Continue**. If you are missing required binaries, a pop-up displays asking if you want to continue with adding the service.
3. Select the services on which the new service should depend. All services must depend on the *same* ZooKeeper service. Click **Continue**.
4. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

5. Review and modify configuration settings, such as data directory paths and heap sizes and click **Continue**. The service is started.
6. Click **Continue** then click **Finish**. You are returned to the Home page.
7. Verify the new service is started properly by checking the health status for the new service. If the Health Status is **Good**, then the service started properly.

Comparing Configurations for a Service Between Clusters

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

To compare the configuration settings for a particular service between two different clusters in a Cloudera Manager deployment, perform the following steps:

1. On the **Home** > **Status** tab, click the name of the service you want to compare, or click the **Clusters** menu and select the name of the service.
2. Click the **Configuration** tab.
3. Click the drop-down menu above the Filters pane, and select from one of the options that begins **Diff with...**:

   - *service* on *cluster* - For example, HBASE-1 on Cluster 1. This is the default display setting. All properties are displayed for the selected instance of the service.
   - *service* on all clusters - For example, HBase on all clusters. All properties are displayed for all instances of the service.
   - **Diff with** *service* on *cluster* - For example, Diff with HBase on Cluster 2. Properties are displayed only if the values for the instance of the service whose page you are on differ from the values for the instance selected in the drop-down menu.
   - **Diff with** *service* on all clusters - For example, Diff with HBase on all clusters. Properties are displayed if the values for the instance of the service whose page you are on differ from the values for one or more other instances in the Cloudera Manager deployment.

The service's properties will be displayed showing the values for each property for the selected clusters. The filters on the left side can be used to limit the properties displayed.

You can also view property configuration values that differ between clusters across a deployment by selecting **Non-uniform Values** on the **Configuration** tab of the Cloudera Manager **Home** > **Status** tab. For more information, see Cluster-Wide Configuration on page 88

Add-on Services

Minimum Required Role: Full Administrator

Cloudera Manager supports adding new types of services (referred to as an **add-on service**) to Cloudera Manager, allowing such services to leverage Cloudera Manager distribution, configuration, monitoring, resource management, and life-cycle management features. An add-on service can be provided by Cloudera or an independent software vendor (ISV). If you have multiple clusters managed by Cloudera Manager, an add-on service can be deployed on any of the clusters.

> **Note:** If the add-on service is already installed and running on hosts that are not currently being managed by Cloudera Manager, you must first add the hosts to a cluster that's under management. See Adding a Host to the Cluster on page 174 for details.

Custom Service Descriptor Files

Integrating an add-on service requires a Custom Service Descriptor (CSD) file. A CSD file contains all the configuration needed to describe and manage a new service. A CSD is provided in the form of a JAR file.

Depending on the service, the CSD and associated software may be provided by Cloudera or by an ISV. The integration process assumes that the add-on service software (parcel or package) has been installed and is present on the cluster. The recommended method is for the ISV to provide the software as a parcel, but the actual mechanism for installing the software is up to the ISV. The instructions in Installing an Add-on Service on page 195 assume that you have obtained the CSD file from the Cloudera repository or from an ISV. It also assumes you have obtained the service software, ideally as a parcel, and have or will install it on your cluster either prior to installing the CSD or as part of the CSD installation process.

Configuring the Location of Custom Service Descriptor Files

The default location for CSD files is `/opt/cloudera/csd`. You can change the location in the Cloudera Manager Admin Console as follows:

1. Select **Administration** > **Settings**.
2. Click the **Custom Service Descriptors** category.
3. Edit the **Local Descriptor Repository Path** property.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
5. Restart Cloudera Manager Server:

   RHEL 7 compatible, SLES, Ubuntu:

```
sudo systemctl restart cloudera-scm-server
```

   RHEL 6 compatible:

```
sudo service cloudera-scm-server restart
```

Installing an Add-on Service

An ISV may provide its software in the form of a parcel, or they may have a different way of installing their software. If their software is not available as a parcel, then you must install their software *before* adding the CSD file. Follow the instructions from the ISV for installing the software. If the ISV has provided their software as a parcel, they may also have included the location of their parcel repository in the CSD they have provided. In that case, install the CSD first and then install the parcel.

Installing the Custom Service Descriptor File

1. Acquire the CSD file from Cloudera or an ISV.

2. Log on to the Cloudera Manager Server host, and place the CSD file under the [location configured](#) for CSD files.
3. Set the file ownership to `cloudera-scm:cloudera-scm` with permission 644.
4. Restart the Cloudera Manager Server:

```
service cloudera-scm-server restart
```

5. Log into the Cloudera Manager Admin Console and restart the Cloudera Management Service.

   a. Do one of the following:

   - 1. Select **Clusters** > **Cloudera Management Service**.
     2. Select **Actions** > **Restart**.

   - On the **Home** > **Status** tab, click

     ▼

     to the right of **Cloudera Management Service** and select **Restart**.

   b. Click **Restart** to confirm. The **Command Details** window shows the progress of stopping and then starting the roles.

   c. When **Command completed with *n/n* successful subcommands** appears, the task is complete. Click **Close**.

Installing the Parcel

> **Note:** It is not required that the Cloudera Manager server host be part of a managed cluster and have an agent installed. Although you initially copy the CSD file to the Cloudera Manager server, the Parcel for the add-on service will not be installed on the Cloudera Manager Server host unless the host is managed by Cloudera Manager.

If you have already installed the external software onto your cluster, you can skip these steps and proceed to [Adding an Add-on Service](#) on page 197.

1.
   Click 🎁 in the main navigation bar. If the vendor has included the location of the repository in the CSD, the parcel should already be present and ready for downloading. If the parcel is available, skip to [step 7](#).

2. Use one of the following methods to open the parcel settings page:

   - **Navigation bar**

     1. Click the parcel icon in the top navigation bar or click **Hosts** and click the **Parcels** tab.
     2. Click the **Configuration** button.

   - **Menu**

     1. Select **Administration** > **Settings**.
     2. Select **Category** > **Parcels** .

3. In the **Remote Parcel Repository URLs** list, click the addition symbol to open an additional row.
4. Enter the path to the repository.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6.
   Click 🎁 . The external parcel should appear in the set of parcels available for download.
7. Download, distribute, and activate the parcel. See [Managing Parcels](#).

Adding an Add-on Service

Add the service following the procedure in <u>Adding a Service</u> on page 193.

Uninstalling an Add-on Service

1. Stop all instances of the service.
2. Delete the service from all clusters. If there are other services that depend on the service you are trying to delete, you must delete those services first.
3. Log on to the Cloudera Manager Server host and remove the CSD file.
4. Restart the Cloudera Manager Server:

```
service cloudera-scm-server restart
```

5. After the server has restarted, log into the Cloudera Manager Admin Console and restart the Cloudera Management Service.
6. Optionally remove the parcel.

Starting, Stopping, and Restarting Services

Minimum Required Role: <u>Operator</u> (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

Starting and Stopping Services

It's important to start and stop services that have dependencies in the correct order. For example, because MapReduce and YARN have a dependency on HDFS, you must start HDFS before starting MapReduce or YARN. The Cloudera Management Service and Hue are the only two services on which no other services depend; although you can start and stop them at anytime, their preferred order is shown in the following procedures.

The Cloudera Manager cluster actions start and stop services in the correct order. To start or stop all services in a cluster, follow the instructions in <u>Starting, Stopping, Refreshing, and Restarting a Cluster</u> on page 84.

Starting a Service on All Hosts

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the service name and select **Start**.
2. Click **Start** in the next screen to confirm. When you see a **Finished** status, the service has started.

The order in which to start services is:

1. Cloudera Management Service
2. ZooKeeper
3. HDFS
4. Solr
5. Flume
6. HBase
7. Key-Value Store Indexer
8. MapReduce or YARN
9. Hive
10. Impala
11. Oozie
12. Sqoop
13. Hue

> **Note:** If you are unable to start the HDFS service, it's possible that one of the roles instances, such as a DataNode, was running on a host that is no longer connected to the Cloudera Manager Server host, perhaps because of a hardware or network failure. If this is the case, the Cloudera Manager Server will be unable to connect to the Cloudera Manager Agent on that disconnected host to start the role instance, which will prevent the HDFS service from starting. To work around this, you can stop all services, abort the pending command to start the role instance on the disconnected host, and then restart all services again without that role instance. For information about aborting a pending command, see Aborting a Pending Command on page 201.

Stopping a Service on All Hosts

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the service name and select **Stop**.
2. Click **Stop** in the next screen to confirm. When you see a **Finished** status, the service has stopped.

The order in which to stop services is:

1. Hue
2. Sqoop
3. Oozie
4. Impala
5. Hive
6. MapReduce or YARN
7. Key-Value Store Indexer
8. HBase
9. Flume
10. Solr
11. HDFS
12. ZooKeeper
13. Cloudera Management Service

Restarting a Service

It is sometimes necessary to restart a service, which is essentially a combination of stopping a service and then starting it again. For example, if you change the hostname or port where the Cloudera Manager is running, or you enable TLS security, you must restart the Cloudera Management Service to update the URL to the Server.

1. On the **Home** > **Status** tab, click

   ▼

   to the right of the service name and select **Restart**.
2. Click **Start** on the next screen to confirm. When you see a **Finished** status, the service has restarted.

To restart all services, use the restart cluster action.

Rolling Restart

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See Managing Licenses on page 37 for more information.

Rolling restart allows you to conditionally restart the role instances of the following services to update software or use a new configuration:

- Flume
- HBase
- HDFS
- Kafka
- Key Trustee KMS
- Key Trustee Server
- MapReduce
- Oozie
- YARN
- ZooKeeper

If the service is not running, rolling restart is not available for that service. You can specify a rolling restart of each service individually.

If you have HDFS high availability enabled, you can also perform a cluster-level rolling restart. At the cluster level, the rolling restart of worker hosts is performed on a host-by-host basis, rather than per service, to avoid all roles for a service potentially being unavailable at the same time. During a cluster restart, to avoid having your NameNode (and thus the cluster) be unavailable during the restart, Cloudera Manager forces a failover to the standby NameNode.

MapReduce (MRv1) JobTracker High Availability on page 435 and YARN (MRv2) ResourceManager High Availability on page 431 is *not* required for a cluster-level rolling restart. However, if you have JobTracker or ResourceManager high availability enabled, Cloudera Manager will force a failover to the standby JobTracker or ResourceManager.

Performing a Service or Role Rolling Restart

You can initiate a rolling restart from either the Status page for one of the eligible services, or from the service's Instances page, where you can select individual roles to be restarted.

1. Go to the service you want to restart.
2. Do one of the following:

   - **service** - Select **Actions** > **Rolling Restart**.
   - **role** -

     1. Click the **Instances** tab.
     2. Select the roles to restart.
     3. Select **Actions for Selected** > **Rolling Restart**.

3. In the pop-up dialog box, select the options you want:

   - Restart only roles whose configurations are stale
   - Restart only roles that are running outdated software versions
   - Which role types to restart

4. If you select an HDFS, HBase, MapReduce, or YARN service, you can have their worker roles restarted in batches. You can configure:

   - How many roles should be included in a batch - Cloudera Manager restarts the worker roles rack-by-rack in alphabetical order, and within each rack, hosts are restarted in alphabetical order. If you are using the default replication factor of 3, Hadoop tries to keep the replicas on at least 2 different racks. So if you have multiple racks, you can use a higher batch size than the default 1. But you should be aware that using too high batch size also means that fewer worker roles are active at any time during the upgrade, so it can cause temporary

performance degradation. If you are using a single rack only, you should only restart *one worker node at a time* to ensure data availability during upgrade.

- How long should Cloudera Manager wait before starting the next batch.
- The number of *batch* failures that will cause the entire rolling restart to fail (this is an advanced feature). For example if you have a very large cluster you can use this option to allow failures because if you know that your cluster will be functional even if some worker roles are down.

> **Note:**
>
> - **HDFS** - If you do not have HDFS high availability configured, a warning appears reminding you that the service will become unavailable during the restart while the NameNode is restarted. Services that depend on that HDFS service will also be disrupted. Cloudera recommends that you restart the DataNodes one at a time—one host per batch, which is the default.
> - **HBase**
>   - Administration operations such as any of the following should not be performed during the rolling restart, to avoid leaving the cluster in an inconsistent state:
>   - Split
>   - Create, disable, enable, or drop table
>   - Metadata changes
>   - Create, clone, or restore a snapshot. Snapshots rely on the RegionServers being up; otherwise the snapshot will fail.
>   - To increase the speed of a rolling restart of the HBase service, set the **Region Mover Threads** property to a higher value. This increases the number of regions that can be moved in parallel, but places additional strain on the HMaster. In most cases, **Region Mover Threads** should be set to 5 or lower.
>   - Another option to increase the speed of a rolling restart of the HBase service is to set the **Skip Region Reload During Rolling Restart** property to `true`. This setting can cause regions to be moved around multiple times, which can degrade HBase client performance.
> - **MapReduce** - If you restart the JobTracker, all current jobs will fail.
> - **YARN** - If you restart ResourceManager and ResourceManager HA is enabled, current jobs continue running: they do not restart or fail. ResourceManager HA is supported for CDH 5.2 and higher.
> - **ZooKeeper** and **Flume** - For both ZooKeeper and Flume, the option to restart roles in batches is not available. They are always restarted one by one.

5. Click **Confirm** to start the rolling restart.

Performing a Cluster-Level Rolling Restart

You can perform a cluster-level rolling restart on demand from the Cloudera Manager Admin Console. A cluster-level rolling restart is also performed as the last step in a rolling upgrade when the cluster is configured with HDFS high availability enabled.

1. If you have not already done so, enable high availability. See HDFS High Availability on page 421 for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you want. Automatic failover does not affect the rolling restart operation.
2. For the cluster you want to restart select **Actions** > **Rolling Restart**.
3. In the pop-up dialog box, select the services you want to restart. Please review the caveats in the preceding section for the services you elect to have restarted. The services that do not support rolling restart will simply be restarted, and will be unavailable during their restart.

4. If you select an HDFS, HBase, or MapReduce service, you can have their worker roles restarted in batches. You can configure:

   - How many roles should be included in a batch - Cloudera Manager restarts the worker roles rack-by-rack in alphabetical order, and within each rack, hosts are restarted in alphabetical order. If you are using the default replication factor of 3, Hadoop tries to keep the replicas on at least 2 different racks. So if you have multiple racks, you can use a higher batch size than the default 1. But you should be aware that using too high batch size also means that fewer worker roles are active at any time during the upgrade, so it can cause temporary performance degradation. If you are using a single rack only, you should only restart *one worker node at a time* to ensure data availability during upgrade.
   - How long should Cloudera Manager wait before starting the next batch.
   - The number of *batch* failures that will cause the entire rolling restart to fail (this is an advanced feature). For example if you have a very large cluster you can use this option to allow failures because if you know that your cluster will be functional even if some worker roles are down.

5. Click **Restart** to start the rolling restart. While the restart is in progress, the Command Details page shows the steps for stopping and restarting the services.

Aborting a Pending Command

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

Commands will time out if they are unable to complete after a period of time.

If necessary, you can abort a pending command. For example, this may become necessary because of a hardware or network failure where a host running a role instance becomes disconnected from the Cloudera Manager Server host. In this case, the Cloudera Manager Server will be unable to connect to the Cloudera Manager Agent on that disconnected host to start or stop the role instance which will prevent the corresponding service from starting or stopping. To work around this, you can abort the command to start or stop the role instance on the disconnected host, and then you can start or stop the service again.

To abort any pending command:

You can click the indicator (   ) with the blue badge, which shows the number of commands that are currently running in your cluster (if any). This indicator is positioned just to the left of the **Support** link at the right hand side of the navigation bar. Unlike the Commands tab for a role or service, this indicator includes all commands running for all services or roles in the cluster. In the Running Commands window, click **Abort** to abort the pending command. For more information, see Viewing Running and Recent Commands on page 241.

To abort a pending command for a service or role:

1. Go to the **Service > Instances** tab for the service where the role instance you want to stop is located. For example, go to the **HDFS Service > Instances** tab if you want to abort a pending command for a DataNode.
2. In the list of instances, click the link for role instance where the command is running (for example, the instance that is located on the disconnected host).
3. Go to the **Commands** tab.
4. Find the command in the list of **Running Commands** and click **Abort Command** to abort the running command.

Deleting Services

Minimum Required Role: Full Administrator

1. Stop the service. For information on starting and stopping services, see Starting, Stopping, and Restarting Services on page 197.

2. On the **Home** > **Status** tab, click

▾

to the right of the service name and select **Delete**.

3. Click **Delete** to confirm the deletion. Deleting a service does *not* clean up the associated client configurations that have been deployed in the cluster or the user data stored in the cluster. For a given "alternatives path" (for example `/etc/hadoop/conf`) if there exist both "live" client configurations (ones that would be pushed out with deploy client configurations for active services) and ones that have been "orphaned" client configurations (the service they correspond to has been deleted), the orphaned ones will be removed from the alternatives database. In other words, to trigger cleanup of client configurations associated with a deleted service you must create a service to replace it. To remove user data, see Remove User Data.

Renaming a Service

Minimum Required Role: Full Administrator

A service is given a name upon installation, and that name is used as an identifier internally. However, Cloudera Manager allows you to provide a display name for a service, and that name will appear in the Cloudera Manager Admin Console instead of the original (internal) name.

1. On the **Home** > **Status** tab, click

▾

to the right of the service name and select **Rename**.
2. Type the new name.
3. Click **Rename**.

The original service name will still be used internally, and may appear or be required in certain circumstances, such as in log messages or in the API.

The rename action is recorded as an Audit event.

When looking at Audit or Event search results for the renamed service, it is possible that these search results might contain either only the original (internal) name, or both the display name and the original name.

Configuring Maximum File Descriptors

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

You can set the maximum file descriptor parameter for all daemon roles. When not specified, the role uses whatever value it inherits from supervisor. When specified, configures soft and hard limits to the configured value.

1. Go to a service.
2. Click the **Configuration** tab.
3. In the Search box, type **rlimit_fds**.
4. Set the **Maximum Process File Descriptors** property for one or more roles.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart the affected role instances.

Exposing Hadoop Metrics to Graphite

Core Hadoop services and HBase support the writing of their metrics to Graphite, a real-time graphing system.

HDFS, YARN, and HBase support the Metrics2 framework; MapReduce1 and HBase support the Metrics framework. See the Cloudera blog post, What is Hadoop Metrics2?

Configure Hadoop Metrics for Graphite Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

1. Go to the **Home** page by clicking the Cloudera Manager logo.
2. Click **Configuration** > **Advanced Configuration Snippets**.
3. Search on the term *Metrics*.
4. To configure HDFS, YARN, or HBase, use **Hadoop Metrics2 Advanced Configuration Snippet (Safety Valve)**. For MapReduce1 (or HBase), use **Hadoop Metrics Advanced Configuration Snippet (Safety Valve)**.
5. Click **Edit Individual Values** to see the supported daemons and their default groups.
6. Configure each default group with a metrics class, sampling period, and Graphite server. See the tables below.
7. To add optional parameters for socket connection retry, modify this example as necessary:

```
*.sink.graphite.retry_socket_interval=60000 #in milliseconds
*.sink.graphite.socket_connection_retries=10 #Set it to 0 if you do not want it to be
retried
```

8. Click **Save Changes**.
9. **Restart** the cluster or service depending on the scope of your changes.

Graphite Configuration Settings Per Daemon

Table 4: Hadoop Metrics2 Graphite Configuration

Service	Daemon Default Group	Graphite Configuration Settings

HBase	Master and RegionServer	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`hbase.sink.graphite.server_host=<hostname>`
		`hbase.sink.graphite.server_port=<port>`
		`hbase.sink.graphite.metrics_prefix=<prefix>`

HDFS	DataNode	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`datanode.sink.graphite.server_host=<hostname>`
		`datanode.sink.graphite.server_port=<port>`
		`datanode.sink.graphite.metrics_prefix=<prefix>`

	NameNode	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`namenode.sink.graphite.server_host=<hostname>`
		`namenode.sink.graphite.server_port=<port>`
		`namenode.sink.graphite.metrics_prefix=<prefix>`

	SecondaryNameNode	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`secondarynamenode.sink.graphite.server_host=<hostname>`
		`secondarynamenode.sink.graphite.server_port=<port>`
		`secondarynamenode.sink.graphite.metrics_prefix=<prefix>`

YARN	NodeManager	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`nodemanager.sink.graphite.server_host=<hostname>`
		`nodemanager.sink.graphite.server_port=<port>`
		`nodemanager.sink.graphite.metrics_prefix=<prefix>`

	ResourceManager	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`resourcemanager.sink.graphite.server_host=<hostname>`
		`resourcemanager.sink.graphite.server_port=<port>`
		`resourcemanager.sink.graphite.metrics_prefix=<prefix>`

	JobHistory Server	`*.sink.graphite.class=org.apache.hadoop.metrics2.sink.GraphiteSink`
		`*.period=10`
		`jobhistoryserver.sink.graphite.server_host=<hostname>`
		`jobhistoryserver.sink.graphite.server_port=<port>`
		`jobhistoryserver.sink.graphite.metrics_prefix=<prefix>`

Note: To use metrics, set values for each *context*. For example, for MapReduce1, add values for both the **JobTracker Default Group** and the **TaskTracker Default Group**.

Table 5: Hadoop Metrics Graphite Configuration

Service	Daemon Default Group	Graphite Configuration Settings
HBase	Master	`hbase.class=org.apache.hadoop.metrics.graphite.GraphiteContext`
	RegionServer	

Service	Daemon Default Group	Graphite Configuration Settings

		`hbase.period=10`
		`hbase.servers=<graphite`
		`  hostname>:<port>`
		
		jvmclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`jvm.period=10`
		`jvm.servers=<graphite`
		`hostname>:<port>`
		
		rpcclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`rpc.period=10`
		`rpc.servers=<graphite`
		`hostname>:<port>`

MapReduce1	JobTracker	dfsclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`dfs.period=10`
		`dfs.servers=<graphite`
		`hostname>:<port>`
		
		mapredclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`mapred.period=10`
		`mapred.servers=<graphite`
		`  hostname>:<port>`
		
		jvmclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`jvm.period=10`
		`jvm.servers=<graphite`
		`hostname>:<port>`
		
		rpcclass=org.apache.hadoop.metrics.graphite.GraphiteContext
		
		`rpc.period=10`
		`rpc.servers=<graphite`
		`hostname>:<port>`

	TaskTracker	

Exposing Hadoop Metrics to Ganglia

Core Hadoop services and HBase support the writing of their metrics to Ganglia, a data representation and visualization tool.

HDFS, YARN, and HBase support the Metrics2 framework; MapReduce1 and HBase support the Metrics framework. See the Cloudera blog post, What is Hadoop Metrics2?

Configure Hadoop Metrics for Ganglia Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

1. Go to the **Home** page by clicking the Cloudera Manager logo.
2. Click **Configuration** > **Advanced Configuration Snippets**.
3. Search on the term *Metrics*.
4. To configure HDFS, YARN, or HBase, use **Hadoop Metrics2 Advanced Configuration Snippet (Safety Valve)**. For MapReduce1 (or HBase), use **Hadoop Metrics Advanced Configuration Snippet (Safety Valve)**.
5. Click **Edit Individual Values** to see the supported daemons and their default groups.
6. Configure each default group with a metrics class, sampling period, and Ganglia server. See the tables below.

7. To add optional parameters for socket connection retry, modify this example as necessary:

```
*.sink.ganglia.retry_socket_interval=60000 #in milliseconds
*.sink.ganglia.socket_connection_retries=10 #Set it to 0 if you do not want it to be
retried
```

8. To define a filter, which is recommended for preventing YARN metrics from overwhelming the Ganglia server, do so *on the sink side*. For example:

```
*.source.filter.class=org.apache.hadoop.metrics2.filter.GlobFilter
*.record.filter.class=${*.source.filter.class}
*.metric.filter.class=${*.source.filter.class}
nodemanager.sink.ganglia.record.filter.exclude=ContainerResource*
```

9. Click **Save Changes**.

10. **Restart** the Cluster or Service depending on the scope of your changes.

Ganglia Configuration Settings Per Daemon

Table 6: Hadoop Metrics2 Ganglia Configuration

Service	Daemon Default Group	Ganglia Configuration Settings

HBase	Master and RegionServer	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```hbase.sink.ganglia.servers=<hostname>:<port>```

HDFS	DataNode	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```datanode.sink.ganglia.servers=<hostname>:<port>```

	NameNode	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```namenode.sink.ganglia.servers=<hostname>:<port>```

	SecondaryNameNode	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```secondarynamenode.sink.ganglia.servers=<hostname>:<port>```

YARN	NodeManager	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```nodemanager.sink.ganglia.servers=<hostname>:<port>```

	ResourceManager	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```resourcemanager.sink.ganglia.servers=<hostname>:<port>```

	JobHistory Server	```*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31```
		```*.period=10```
		```jobhistoryserver.sink.ganglia.servers=<hostname>:<port>```

> **Note:** To use metrics, set values for each *context*. For example, for MapReduce1, add values for both the **JobTracker Default Group** and the **TaskTracker Default Group**.

Table 7: Hadoop Metrics Ganglia Configuration

Service	Daemon Default Group	Ganglia Configuration Settings

HBase	Master	```hbase.class=org.apache.hadoop.metrics.ganglia.GangliaContext31```
		```hbase.period=10```
		```hbase.servers=<hostname>:<port>```

	RegionServer	```jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext31```
		```jvm.period=10```
		```jvm.servers=<hostname>:<port>```

Service	Daemon Default Group	Ganglia Configuration Settings

		rpc.class=org.apache.hadoop.metrics.ganglia.GangliaContext31
		
		 `rpc.period=10`
		 `rpc.servers=<hostname>:<port>`

MapReduce1	JobTracker	dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext
		
		 `dfs.period=10`
		 `dfs.servers=<hostname>:<port>`
		
		 mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext
		
		 `mapred.period=10`
		 `mapred.servers=<hostname>:<port>`
		
		 jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext
		
		 `jvm.period=10`
		 `jvm.servers=<hostname>:<port>`
		
		 rpc.class=org.apache.hadoop.metrics.ganglia.GangliaContext
		
		 `rpc.period=10`
		 `rpc.servers=<hostname>:<port>`

	TaskTracker	

Managing Roles

When Cloudera Manager configures a service, it configures hosts in your cluster with one or more functions (called roles in Cloudera Manager) that are required for that service. The role determines which Hadoop daemons run on a given host. For example, when Cloudera Manager configures an HDFS service instance it configures one host to run the NameNode role, another host to run as the Secondary NameNode role, another host to run the Balancer role, and some or all of the remaining hosts to run DataNode roles.

Configuration settings are organized in role groups. A **role group** includes a set of configuration properties for a specific group, as well as a list of role instances associated with that role group. Cloudera Manager automatically creates default role groups.

For role types that allow multiple instances on multiple hosts, such as DataNodes, TaskTrackers, RegionServers (and many others), you can create multiple role groups to allow one set of role instances to use different configuration settings than another set of instances of the same role type. In fact, upon initial cluster setup, if you are installing on identical hosts with limited memory, Cloudera Manager will (typically) automatically create two role groups for each worker role — one group for the role instances on hosts with only other worker roles, and a separate group for the instance running on the host that is also hosting master roles.

The HDFS service is an example of this: Cloudera Manager typically creates one role group (DataNode Default Group) for the DataNode role instances running on the worker hosts, and another group (HDFS-1-DATANODE-1) for the DataNode instance running on the host that is also running the master roles such as the NameNode, JobTracker, HBase Master and so on. Typically the configurations for those two classes of hosts will differ in terms of settings such as memory for JVMs.

Cloudera Manager configuration screens offer two layout options: classic and new. The new layout is the default; however, on each configuration page you can easily switch between layouts using the **Switch to *XXX* layout** link at the top right of the page. For more information, see [Cluster Configuration Overview](#) on page 59.

Gateway Roles

A **gateway** is a special type of role whose sole purpose is to designate a host that should receive a client configuration for a specific service, when the host does not have any roles running on it. Gateway roles enable Cloudera Manager to install and manage client configurations on that host. There is no process associated with a gateway role, and its status will always be Stopped. You can configure gateway roles for HBase, HDFS, Hive, Kafka, MapReduce, Solr, Spark, Sqoop 1 Client, and YARN.

Role Instances

Adding a Role Instance

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

After creating services, you can add role instances to the services. For example, after initial installation in which you created the HDFS service, you can add a DataNode role instance to a host where one was not previously running. Upon upgrading a cluster to a new version of CDH you might want to create a role instance for a role added in the new version.

1. Go to the service for which you want to add a role instance. For example, to add a DataNode role instance, go to the HDFS service.
2. Click the **Instances** tab.
3. Click the **Add Role Instances** button.
4. Customize the assignment of role instances to hosts. The wizard evaluates the hardware configurations of the hosts to determine the best hosts for each role. The wizard assigns all worker roles to the same set of hosts to which the HDFS DataNode role is assigned. You can reassign role instances.

Click a field below a role to display a dialog box containing a list of hosts. If you click a field containing multiple hosts, you can also select **All Hosts** to assign the role to all hosts, or **Custom** to display the hosts dialog box.

The following shortcuts for specifying hostname patterns are supported:

- Range of hostnames (without the domain portion)

Range Definition	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].company.com	host1.company.com, host2.company.com, host3.company.com
host[07-10].company.com	host07.company.com, host08.company.com, host09.company.com, host10.company.com

- IP addresses
- Rack name

Click the **View By Host** button for an overview of the role assignment by hostname ranges.

5. Click **Continue**.
6. In the Review Changes page, review the configuration changes to be applied. Confirm the settings entered for file system paths. The file paths required vary based on the services to be installed. For example, you might confirm the NameNode Data Directory and the DataNode Data Directory for HDFS. Click **Continue**. The wizard finishes by performing any actions necessary to prepare the cluster for the new role instances. For example, new DataNodes are added to the NameNode `dfs_hosts_allow.txt` file. The new role instance is configured with the default role group for its role type, even if there are multiple role groups for the role type. If you want to use a different role group, follow the instructions in Managing Role Groups on page 213 for moving role instances to a different role group. The new role instances are not started automatically.

Starting, Stopping, and Restarting Role Instances

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

If the host for the role instance is currently decommissioned, you will not be able to start the role until the host has been recommissioned.

> **Important:** Use Cloudera Manager to stop the Node Manager service. If it is stopped manually, it can cause jobs to fail.

1. Go to the service that contains the role instances to start, stop, or restart.
2. Click the **Instances** tab.
3. Check the checkboxes next to the role instances to start, stop, or restart (such as a DataNode instance).
4. Select **Actions for Selected** > **Start**, **Stop**, or **Restart**, and then click **Start**, **Stop**, or **Restart** again to start the process. When you see a **Finished** status, the process has finished.

Also see Rolling Restart on page 198.

Decommissioning Role Instances

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

You can remove a role instance such as a DataNode from a cluster while the cluster is running by decommissioning the role instance. When you decommission a role instance, Cloudera Manager performs a procedure so that you can safely retire a host without losing data. Role decommissioning applies to HDFS DataNode, MapReduce TaskTracker, YARN NodeManager, and HBase RegionServer roles.

Hosts with DataNodes and DataNode roles themselves can only be decommissioned if the resulting action leaves enough DataNodes commissioned to maintain the configured HDFS replication factor (by default 3). If you attempt to

decommission a DataNode or a host with a DataNode in such situations, the decommission process will not complete and must be aborted.

A role will be decommissioned if its host is decommissioned. See Tuning and Troubleshooting Host Decommissioning on page 184 for more details.

To remove a DataNode from the cluster, you decommission the DataNode role as described here and then perform a few additional steps to remove the role. See Removing a DataNode on page 99.

To decommission role instances:

1. If you are decommissioning DataNodes, perform the steps in Tuning HDFS Prior to Decommissioning DataNodes on page 184.
2. Click the service instance that contains the role instance you want to decommission.
3. Click the **Instances** tab.
4. Check the checkboxes next to the role instances to decommission.
5. Select **Actions for Selected** > **Decommission**, and then click **Decommission** again to start the process. A Decommission Command pop-up displays that shows each step or decommission command as it is run. In the Details area, click ▸ to see the subcommands that are run. Depending on the role, the steps may include adding the host to an "exclusions list" and refreshing the NameNode, JobTracker, or NodeManager; stopping the Balancer (if it is running); and moving data blocks or regions. Roles that do not have specific decommission actions are stopped.

   You can abort the decommission process by clicking the **Abort** button, but you must recommission and restart the role.

   The Commission State facet in the Filters list displays 🗄 Decommissioning while decommissioning is in progress, and 🗄 Decommissioned when the decommissioning process has finished. When the process is complete, a ✔ is added in front of Decommission Command.

Recommissioning Role Instances

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

1. Click the service that contains the role instance you want to recommission.
2. Click the **Instances** tab.
3. Check the checkboxes next to the decommissioned role instances to recommission.
4. Select **Actions for Selected** > **Recommission**, and then click **Recommission** to start the process. A Recommission Command pop-up displays that shows each step or recommission command as it is run. When the process is complete, a ✔ is added in front of Recommission Command.
5. Restart the role instance.

Deleting Role Instances

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. Click the service instance that contains the role instance you want to delete. For example, if you want to delete a DataNode role instance, click an HDFS service instance.
2. Click the **Instances** tab.
3. Check the checkboxes next to the role instances you want to delete.
4. If the role instance is running, select **Actions for Selected** > **Stop** and click **Stop** to confirm the action.
5. Select **Actions for Selected** > **Delete**. Click **Delete** to confirm the deletion.

> **Note:** Deleting a role instance does not clean up the associated client configurations that have been deployed in the cluster.

Configuring Roles to Use a Custom Garbage Collection Parameter

Every Java-based role in Cloudera Manager has a configuration setting called **Java Configuration Options for** *role* where you can enter command line options. Commonly, garbage collection flags or extra debugging flags would be passed here. To find the appropriate configuration setting, select the service you want to modify in the Cloudera Manager Admin Console, then use the Search box to search for Java Configuration Options.

You can add configuration options for all instances of a given role by making this configuration change at the service level. For example, to modify the setting for all DataNodes, select the HDFS service, then modify the **Java Configuration Options for DataNode** setting.

To modify a configuration option for a given instance of a role, select the service, then select the particular role instance (for example, a specific DataNode). The configuration settings you modify will apply to the selected role instance only.

For detailed instructions see <u>Modifying Configuration Properties Using Cloudera Manager</u> on page 60.

Role Groups

Minimum Required Role: <u>Configurator</u> (also provided by **Cluster Administrator, Full Administrator**)

A *role group* is a set of configuration properties for a role type, as well as a list of role instances associated with that group. Cloudera Manager automatically creates a default role group named *Role Type* **Default Group** for each role type.Each role instance can be associated with only a single role group.

Role groups provide two types of properties: those that affect the configuration of the service itself and those that affect monitoring of the service, if applicable (the **Monitoring** subcategory). (Not all services have monitoring properties). For more information about monitoring properties see <u>Configuring Monitoring Settings</u> on page 223.

When you run the installation or upgrade wizard, Cloudera Manager configures the default role groups it adds, and adds any other required role groups for a given role type. For example, a DataNode role on the same host as the NameNode might require a different configuration than DataNode roles running on other hosts. Cloudera Manager creates a separate role group for the DataNode role running on the NameNode host and uses the default configuration for DataNode roles running on other hosts.

You can modify the settings of the default role group, or you can create new role groups and associate role instances to whichever role group is most appropriate. This simplifies the management of role configurations when one group of role instances may require different settings than another group of instances of the same role type—for example, due to differences in the hardware the roles run on. You modify the configuration for any of the service's role groups through the Configuration tab for the service. You can also <u>override</u> the settings inherited from a role group for a role instance.

If there are multiple role groups for a role type, you can move role instances from one group to another. When you move a role instance to a different group, it inherits the configuration settings for its new group.

Creating a Role Group

1. Go to a service status page.
2. Click the **Instances** or **Configuration** tab.
3. Click **Role Groups**.
4. Click **Create new group...**.
5. Provide a name for the group.
6. Select the role type for the group. You can select role types that allow multiple instances and that exist for the service you have selected.
7. In the **Copy From** field, select the source of the basic configuration information for the role group:

   - An existing role group of the appropriate type.
   - **None...**. The role group is set up with generic default values that are *not* the same as the values Cloudera Manager sets in the default role group, as Cloudera Manager specifically sets the appropriate configuration properties for the services and roles it installs. After you create the group you must <u>edit the configuration</u> to

set missing properties (for example the TaskTracker Local Data Directory List property, which is not populated if you select None) and clear other validation warnings and errors.

Managing Role Groups

You can rename or delete existing role groups, and move roles of the same role type from one group to another.

1. Go to a service status page.
2. Click the **Instances** or **Configuration** tab.
3. Click **Role Groups**.
4. Click the group you want to manage. Role instances assigned to the role group are listed.
5. Perform the appropriate procedure for the action:

Action	Procedure
Rename	1. Click the role group name, click ⬇ next to the name on the right and click **Rename**. 2. Specify the new name and click **Rename**.
Delete	You cannot delete any of the default groups. The group must first be empty; if you want to delete a group you've created, you must move any role instances to a different role group. 1. Click the role group name. 2. Click ⬇ next to the role group name on the right, select **Delete**, and confirm by clicking **Delete**. Deleting a role group removes it from host templates.
Move	1. Select the role instance(s) to move. 2. Select **Actions for Selected** > **Move To Different Role Group...**. 3. In the pop-up that appears, select the target role group and click **Move**.

Monitoring and Diagnostics

This section is for system administrators who want to use Cloudera Manager to monitor and diagnose their CDH installation. You can use the Cloudera Manager Admin Console to monitor cluster health, metrics, and usage, view processing activities, and view events, logs, and reports to troubleshoot problems and monitor compliance.

Introduction to Cloudera Manager Monitoring

Cloudera Manager provides many features for monitoring the health and performance of the components of your clusters (hosts, service daemons) as well as the performance and resource demands of the jobs running on your clusters. This guide has information on the following monitoring features:

- **Monitoring Services on page 231** - describes how to view the results of health tests at both the service and role instance level. Various types of metrics are displayed in charts that help with problem diagnosis. Health tests include advice about actions you can take if the health of a component becomes concerning or bad. You can also view the history of actions performed on a service or role, and can view an audit log of configuration changes.
- **Monitoring Hosts on page 243** - describes how to view information pertaining to all the hosts on your cluster: which hosts are up or down, current resident and virtual memory consumption for a host, what role instances are running on a host, which hosts are assigned to different racks, and so on. You can look at a summary view for all hosts in your cluster or drill down for extensive details about an individual host, including charts that provide a visual overview of key metrics on your host.
- **Monitoring Activities on page 248** - describes how to view the activities running on the cluster, both at the current time and through dashboards that show historical activity, and provides many statistics, both in tabular displays and charts, about the resources used by individual jobs. You can compare the performance of similar jobs and view the performance of individual task attempts across a job to help diagnose behavior or performance problems.
- **Events on page 285** - describes how to view events and make them available for alerting and for searching, giving you a view into the history of all relevant events that occur cluster-wide. You can filter events by time range, service, host, keyword, and so on.
- **Alerts on page 287** - describes how to configure Cloudera Manager to generate alerts from certain events. You can configure thresholds for certain types of events, enable and disable them, and configure alert notifications by email or using SNMP trap for critical events. You can also suppress alerts temporarily for individual roles, services, hosts, or even the entire cluster to allow system maintenance/troubleshooting without generating excessive alert traffic.
- **Lifecycle and Security Auditing on page 301** - describes how to view service, role, and host lifecycle events such as creating a role or service, making configuration revisions for a role or service, decommissioning and recommissioning hosts, and running commands recorded by Cloudera Manager management services. You can filter audit event entries by time range, service, host, keyword, and so on.
- **Charting Time-Series Data on page 303** - describes how to search metric data, create charts of the data, group (facet) the data, and save those charts to user-defined dashboards.
- **Logs on page 327 -** describes how to access logs in a variety of ways that take into account the current context you are viewing. For example, when monitoring a service, you can easily click a single link to view the log entries related to that specific service, through the same user interface. When viewing information about a user's activity, you can easily view the relevant log entries that occurred on the hosts used by the job while the job was running.
- **Reports on page 330** - describes how to view historical information about disk utilization by user, user group, and by directory and view cluster job activity user, group, or job ID. These reports are aggregated over selected time periods (hourly, daily, weekly, and so on) and can be exported as XLS or CSV files. You can also manage HDFS directories as well, including searching and setting quotas.
- **Troubleshooting Cluster Configuration and Operation on page 336** - contains solutions to some common problems that prevent you from using Cloudera Manager and describes how to use Cloudera Manager log and notification management tools to diagnose problems.
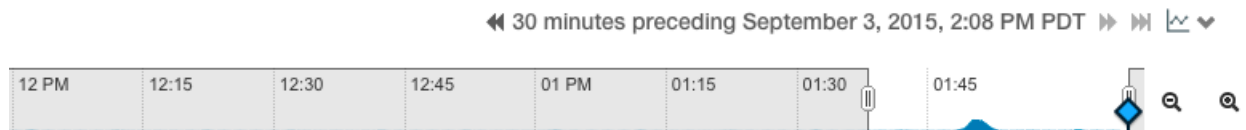
Time Line

The Time Line appears on many pages in Cloudera Manager. When you view the top level service and Hosts tabs, the Time Line shows status and health only for a specific point in time. When you are viewing the Logs and Events tabs, and when you are viewing the Status, Commands, Audits, Jobs, Applications, and Queries pages of individual services, roles, and hosts, the Time Line appears as a Time Range Selector, which lets you highlight a range of time over which to view historical data.

Click the (⟋) icon at the far right to turn on and turn off the display of the Time Line.

Cloudera Manager displays timestamped data using the time zone of the host where Cloudera Manager server is running. The time zone information can be found under the **Support** > **About** menu.

The background chart in the Time Line shows the percentage of CPU utilization on all hosts in the cluster, updated at approximately one-minute intervals, depending on the total visible time range. You can use this graph to identify periods of activity that may be of interest.

In the pages that support a time range selection, the area between the handles shows the selected time range.

◀◀ 30 minutes preceding September 3, 2015, 2:08 PM PDT ▶▶ ▶▶| ⟋ ⌄

| 12 PM | 12:15 | 12:30 | 12:45 | 01 PM | 01:15 | 01:30 | 01:45 | 🔍 🔍 |

There are a variety of ways to change the time range in this mode.

The Reports screen (**Clusters** > **Reports**) does not support the Time Range Selector: the historical reports accessed from the Reports screen have their own time range selection mechanism.

Zooming the Time Line In or Out

Use the Zoom In and Zoom Out buttons (🔍 and 🔍) to zoom the time line graph in or out.

- **Zoom In** shows a shorter time period with more detailed interval segments. Zooming does not change your selected time range. However, the ability to zoom the Time Line can make it easier to use the selector to highlight a time range.
- **Zoom Out** lets you show a longer time period on the time range graph (with correspondingly less granular segmentation).

### Selecting a Point In Time or a Time Range

Depending on what page the Time Line appears, you can select a point in time or a time range. There are two ways to look at information about your cluster—its current status and health, or its status and health at some point (or during some interval) in the past. When you are looking at a point in the past, some functions may not be available. For example, on a Service Status page, the **Actions** menu (where you can take actions like stopping, starting, or restarting services or roles) is accessible only when you are looking at **Current** status.

Selecting a Point in Time

Status information on pages such as the service **Status** pages, reflects the state at a single point in time (a snapshot of the health and status). When displayed data is from a single point in time (a snapshot), the panel or column displays a small version of the Time Marker icon (◆) in the panel. This indicates that the data corresponds to the time at the location of the Time Marker on the Time Line.

By default, the status is shown at the current time. If you specify an earlier point on the time range graph, you see the status as it was at the selected point in the past.

- When the Time Marker is set to the current time, it is blue (◆).
- When the Time Marker is set to a time in the past, it is orange (◆).
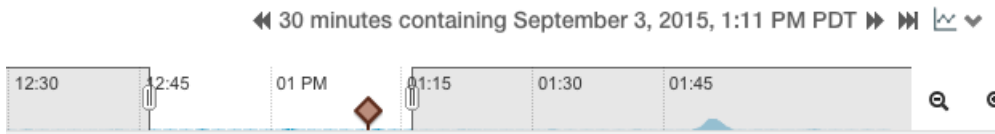
You can select the point in time in one of the following ways:

- By moving the Time Marker (◆)
- When the Time Marker is set to a past time, you can quickly switch back to view the current time using the Now button (⏭).
- By clicking the date, choosing the date and time, and clicking **Apply**.

Selecting a Time Range

Pages such as the Logs, Events, and Activities show data over a time range rather than at a single point. These default to showing the past 30 minutes of data (ending at the current time). The charts that appear on the individual Service Status and Host Status pages also show data over a time range. For this type of display, there are several ways to select a time range of interest:

- Drag one (or both) edges of the time range handles to expand or contract the range.
- Choose a duration by clicking a duration link

  30m 1h 2h 6h 12h 1d 7d 30d

  and then do one of the following:

  - Click the next ⏭ or ⏮ previous buttons to select the next or previous duration.
  - Click somewhere in the dark portion of the time range to choose the selected duration.

- Click the date range

  ⏮ 30 minutes containing September 3, 2015, 1:11 PM PDT ⏭ ⏮ 〰 ⌄

  to open the time selection widget. Enter a start and end time and click **Apply** to put your choice into effect.

- When you are under the **Clusters** tab with an individual activity selected, a **Zoom to Duration** button is available. This lets you zoom the time selection to include just the time range that corresponds to the duration of your selected activity.

Health Tests

Cloudera Manager monitors the health of the services, roles, and hosts that are running in your clusters using *health tests*. The Cloudera Management Service also provides health tests for its roles. Role-based health tests are enabled by default. For example, a simple health test is whether there's enough disk space in every NameNode data directory. A more complicated health test may evaluate when the last checkpoint for HDFS was compared to a threshold or whether a DataNode is connected to a NameNode. Some of these health tests also aggregate other health tests: in a distributed system like HDFS, it's normal to have a few DataNodes down (assuming you've got dozens of hosts), so we allow for setting thresholds on what percentage of hosts should color the entire service down.

Health tests can return one of three values: **Good**, **Concerning**, and **Bad**. A test returns **Concerning** health if the test falls below a warning threshold. A test returns **Bad** if the test falls below a critical threshold. The overall health of a service or role instance is a roll-up of its health tests. If any health test is **Concerning** (but none are **Bad**) the role's or service's health is **Concerning**; if any health test is **Bad**, the service's or role's health is **Bad**.

In the Cloudera Manager Admin Console, health tests results are indicated with colors: **Good** ✅ , **Concerning** 🟡 , and **Bad** ⛔ .

There are two types of health tests:

- **Pass-fail tests** - there are two types:

  - Compare a property to a yes-no value. For example, whether a service or role started as expected, a DataNode is connected to its NameNode, or a TaskTracker is (or is not) blacklisted.

– Exercise a service lightly to confirm it is working and responsive. HDFS (NameNode role), HBase, and ZooKeeper services perform these tests, which are referred to as "canary" tests.

Both types of pass-fail tests result in the health reported as being either **Good** or **Bad**.

- **Metric tests** - compare a property to a numeric value. For example, the number of file descriptors in use, the amount of disk space used or free, how much time spent in garbage collection, or how many pages were swapped to disk in the previous 15 minutes. In these tests the property is compared to a threshold that determine whether everything is **Good**, (for example, plenty of disk space available), whether it is **Concerning** (disk space getting low), or is **Bad** (a critically low amount of disk space).

By default most health tests are enabled and (if appropriate) configured with reasonable thresholds. You can modify threshold values by editing the monitoring properties under the entity's **Configuration** tab. You can also enable or disable individual or summary health tests, and in some cases specify what should be included in the calculation of overall health for the service, role instance, or host. See Configuring Monitoring Settings on page 223 for more information.

Viewing Health Test Results

Health test results are available in the following locations:

- **Home** > **Status** tab where various health results determine an overall health assessment of the service or role. The overall health of a role or service is a roll-up of its health tests; if any health test is **Bad**, the service's or role's health will be **Bad**. If any health test is **Concerning** (but none are **Bad**) the role's or service's health will be **Concerning**.
- **Hosts** tab, which shows summary result for the hosts.
- **Status** tab - which shows metrics for services, role instances, and hosts. These are reflected in the results shown in the **Health Tests** panel when you have selected a service, role instance, or host.
- The **All Health Issues** tab of the **Home** page displays all health issues. You can sort the display by entity or by Health Test.

For some health test results, you can chart the associated metrics over a time range. See Viewing Service Status on page 233, Viewing Role Instance Status on page 237, and Host Details on page 244 for details.

Suppressing Health Test Results

Cloudera Manager displays warnings when health tests indicate a problem in the cluster. Sometimes these warnings are expected or do not indicate a real problem in your deployment. You can suppress display of these warnings in Cloudera Manager.

You can suppress health test warnings as they appear or before any tests run. Suppressed health tests are hidden in Cloudera Manager and their status does not affect the roll-up of health tests that display for a service, host, or role instance. Suppressed health test warnings remain available in Cloudera Manager, and the tests continue to run but the results are hidden. You can unsuppress a suppressed health test at any time.

> **Note:** *Suppressing* a health test is different than *disabling* a health test. A disabled health test never runs, whereas a suppressed health test runs but its results are hidden.

Suppressing a Health Test

1. Go to the health test you want to suppress. (See Viewing Health Test Results on page 217.)
2. Click the **Suppress...** link to the right of the health test description.

   A dialog box opens where you can enter a comment about the suppression action.

3. Click **Confirm**.

   The display changes to **Suppressing...** while the change is propagated.

Managing Suppressed Health Tests

On pages where you have suppressed validations, you will see a link that says **Show # Suppressed Test**. On this screen, you can:

- Click the **Show # Suppressed Test** link to view all suppressed health tests for the page.
- Click the **Unsuppress...** link to unsuppress the health test.
- Click **Hide Suppressed Tests** to re-hide the suppressed tests.

Configuring Suppression of Health Tests Before Tests Run

1. Go to the service or host with the health test you want to suppress.
2. Click the **Configuration** tab.
3. In the filters on the left, select **Category** > **Suppressions**.

   A list of suppression properties displays. The names of the properties begin with **Suppress Health Test**.

4. Select a health test suppression property to suppress the test.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Viewing a List of Suppressed Health Tests

1. From the **Home** page or the **Status** page of a cluster, select **Configuration** > **Suppressed Health and Configuration Issues**.
2. Select **Status** > **Non-default**.

   A list of suppressed health tests and configuration issues displays.

3. To limit the list to health tests, enter "health test" in the **Search** box.

Unsuppressing Health Tests

You can unsuppress a health test by doing one of the following:

- To unsuppress a single health test where it displays, click the **Unsuppress...** link next to a suppressed test. (You may need to click the **Show # Suppressed Test** link first.)
- To unsuppress one or more health tests from the configuration screen:

   1. Go to the service or host with the health test you want to unsuppress.
   2. Select **Status** > **Non-default**.

      A list of suppressed health tests and configuration issues displays.

   3. (Optional) Type the name of the health test in the **Search** box to locate it.
   4. Clear the suppression property for the health test.
   5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Cloudera Manager Admin Console Home Page

When you start the [Cloudera Manager Admin Console](#) on page 16, the **Home** > **Status** tab displays.

You can also go to the **Home** > **Status** tab by clicking the Cloudera Manager logo in the top navigation bar.

Status

The Status tab contains:

- **Clusters** - The clusters being managed by Cloudera Manager. Each cluster is displayed either in summary form or in full form depending on the configuration of the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property. When the number of clusters exceeds the value of the property, only cluster summary information displays.

  - **Summary Form** - A list of links to cluster status pages. Click **Customize** to jump to the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property.
  - **Full Form** - A separate section for each cluster containing a link to the cluster status page and a table containing links to the Hosts page and the status pages of the services running in the cluster.

  Each service row in the table has a menu of actions that you select by clicking

  and can contain one or more of the following indicators:

Indicator	Meaning	Description
❶2	Health issue	Indicates that the service has at least one health issue. The indicator shows the number of health issues at the highest severity level. If there are Bad health test results, the indicator is red. If there are no Bad health test results, but Concerning test results exist, then the indicator is yellow. No indicator is shown if there are no Bad or Concerning health test results.
		Important: If there is one Bad health test result and two Concerning health results, there will be three health issues, but the number will be one.
		Click the indicator to display the **Health Issues** pop-up dialog box.
		By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show *n* concerning issue(s)**

Indicator	Meaning	Description
		link.Click the link to display the Status page containing with details about the health test result.
✖ 4	Configuration issue	Indicates that the service has at least one configuration issue. The indicator shows the number of configuration issues at the highest severity level. If there are configuration errors, the indicator is red. If there are no errors but configuration warnings exist, then the indicator is yellow. No indicator is shown if there are no configuration notifications. ❗ **Important:** If there is one configuration error and two configuration warnings, there will be three configuration issues, but the number will be one. Click the indicator to display the **Configuration Issues** pop-up dialog box. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show _n_ warning(s)** link.Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.See Managing Services on page 193.
⏻ Restart Needed ↻ Refresh Needed	Configuration modified	Indicates that at least one of a service's roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. Click the indicator to display the Stale Configurations on page 77 page.To bring the cluster up-to-date, click the Refresh or Restart button on the Stale Configurations page or follow the instructions in Refreshing a Cluster on page 85, Restarting a Cluster on page 85, or Restarting Services and Instances after Configuration Changes on page 64.
🖳	Client configuration redeployment required	Indicates that the client configuration for a service should be redeployed. Click the indicator to display the Stale Configurations on page 77 page.To bring the cluster up-to-date, click the Deploy Client Configuration button on the Stale Configurations page or follow the instructions in Manually Redeploying Client Configuration Files on page 80.

– **Cloudera Management Service** - A table containing a link to the Cloudera Manager Service. The Cloudera Manager Service has a menu of actions that you select by clicking

[▼] .

– **Charts** - A set of charts (dashboard) that summarize resource utilization (IO, CPU usage) and processing metrics.

Click a line, stack area, scatter, or bar chart to expand it into a full-page view with a legend for the individual charted entities as well more fine-grained axes divisions.

By default the time scale of a dashboard is 30 minutes. To change the time scale, click a duration link

30m 1h 2h 6h 12h 1d 7d 30d   at the top-right of the dashboard.

To set the dashboard type, click [✎ ▼] and select one of the following:

- **Custom** - displays a custom dashboard.
- **Default** - displays a default dashboard.
- **Reset** - resets the custom dashboard to the predefined set of charts, discarding any customizations.

All Health Issues

Displays all health issues by cluster. The number badge has the same semantics as the per service health issues reported on the Status tab.

- By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show *n* concerning issue(s)** link.
- To group the health test results by entity or health test, click the buttons on the **Organize by Entity/Organize by Health Test** switch.
- Click the link to display the Status page containing with details about the health test result.

All Configuration Issues

Displays all configuration issues by cluster. The number badge has the same semantics as the per service configuration issues reported on the Status tab. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show *n* warning(s)** link. Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.

All Recent Commands

Displays all commands run recently across the clusters. A badge indicates how many recent commands are still running. Click the command link to display details about the command and child commands. See also [Viewing Running and Recent Commands](#) on page 241.

Starting and Logging into the Cloudera Manager Admin Console

1. In a web browser, go to `http://<server_host>:7180`, where *<server_host>* is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   The login screen for Cloudera Manager Admin Console displays.

2. Log into Cloudera Manager Admin Console using the [credentials](#) assigned by your administrator. User accounts are assigned [roles](#) that constrain the features available to you.

   > **Note:** You can configure the Cloudera Manager Admin Console to automatically log out a user after a configurable period of time. See [Automatic Logout](#) on page 21.

Displaying the Cloudera Manager Server Version and Server Time

To display the version, build number, and time for the Cloudera Manager Server:

1. Open the Cloudera Manager Admin Console.
2. Select **Support** > **About**.

Viewing Charts for Cluster, Service, Role, and Host Instances

For cluster, service, role, and host instances you can see [dashboards](#) of [charts](#) of various metrics relevant to the entity you are viewing. While the metrics displayed are different for each entity, the basic functionality works in the same way.

The **Home** > **Status** tab for clusters and the Status tab for a service, role, or host display dashboards containing a limited set of charts.

The Status page Charts Library tab displays a dashboard containing a much larger set of charts, organized by categories such as process charts, host charts, CPU charts, and so on, depending on the entity (service, role, or host) that you are viewing.

A custom dashboard is displayed by default when you view the Status tab for an entity. You can switch between [custom and default](#) dashboards by using the edit button

[edit icon ▾]

to the upper right of the chart.

Displaying Information from Charts

There are various ways to display information from charts.

- Click the ⚙ icon at the top right to see a menu for opening the chart in the Chart Builder or exporting its data.
- Change the size of a chart on a dashboard by dragging the lower-right corner of the chart.
- Hovering with the mouse over a stream on a chart (for example, a line on a line chart) opens a small pop-up window that displays information about that stream. Move the mouse horizontally to see the data values change in the small pop-up window, based on the time represented at the mouse's position along the chart's horizontal axis. Click any stream within the chart to display a larger pop-up window that includes additional information for the stream at the point in time where the mouse was clicked. At the bottom of the large pop-up window is a button for viewing the Cloudera Manager page for the entity (service, host, role, query, or application) associated with the chart, if applicable (**View Service**, **View Host**, and so on). Click the button **View Entity Chart** to display a chart for the stream on its own page. If the chart displays more than one stream, the new chart displays only the stream that was selected when the button was clicked.
- The chart page includes an editable text field containing a default title based on the select statement that was used to create the chart. This title will be used if you save the chart as a dashboard. Type a new title for the chart into this field, if desired.

Exporting Data from Charts

The menu displayed by clicking the ⚙ icon at the top right includes the selections **Export JSON**, and **Export CSV**.

- Click **Export JSON** to display the chart data in JSON format in a new browser window.
- Click **Export CSV** to open a **Save** dialog box enabling you to save the data as a CSV file, choose a program to open the CSV, or open the file with your system's default program for editing and displaying CSV files.

> **Note:** Time values that appear in Cloudera Manager charts reflect the time zone setting on the Cloudera Manager client machine, but time values returned by the Cloudera Manager API (including those that appear in JSON and CSV files exported from charts) reflect Coordinated Universal Time (UTC). For more information on the timestamp format, see the Cloudera Manager API documentation, for example, `ApiTimeSeriesData.java`.

Adding and Removing Charts from a Dashboard

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

- With a custom dashboard, the menu displayed by clicking the ⚙ icon at the top right includes the selection **Remove** for users with the required roles. The **Remove** button does not appear in the menu when the default dashboard is used because the default dashboard does not allow removing the original charts. Use the edit button

  [edit icon ▾]

  to the upper right of the chart to switch between custom and default dashboards.
- Charts can also be added to a custom dashboard. Click the ⚙ icon at the top right and click **Add to Dashboard**. You can add the chart to an existing dashboard by selecting **Add chart to an existing custom or system dashboard** and selecting the dashboard name. Add the chart to a new dashboard by clicking **Add chart to a new custom dashboard** and enter a new name in the **Dashboard Name** field.

Creating Triggers from Charts

Minimum Required Role: [Full Administrator](#)

- For many charts, the menu opened with the ⚙ icon will also include **Create Trigger**. Triggers allow you to define actions to be taken when a specified condition is met. For information on creating triggers, see [Triggers](#) on page 294.

Configuring Monitoring Settings

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

There are several types of monitoring settings you can configure in Cloudera Manager:

- Health tests - For a service or role for which monitoring is provided, you can enable and disable selected health tests and events, configure how those health tests factor into the overall health of the service, and modify thresholds for the status of certain health tests. For hosts you can disable or enable selected health tests, modify thresholds, and enable or disable health alerts.
- Free space - For hosts, you can set threshold-based monitoring of free space in the various directories on the hosts Cloudera Manager monitors.
- Activities - For MapReduce, YARN, and Impala services, you can configure aspects of how Cloudera Manager monitors activities, applications, and queries.
- Alerts - For all roles you can configure health alerts and configuration change alerts. You can also configure some service specific alerts and how alerts are delivered.
- Log events - For all roles you can configure logging thresholds, log directories, log event capture, when log messages become events, and when to generate log alerts.
- Monitoring roles - For the Cloudera Management Service you can configure monitoring settings for the monitoring roles themselves—enable and disable health tests on the monitoring processes as well as configuring some general settings related to events and alerts (specifically with the Event Server and Alert Publisher). Each of the Cloudera Management Service roles has its own parameters that can be modified to specify how much data is retained by that service. For some monitoring functions, the amount of retained data can grow very large, so it may become necessary to adjust the limits.

For general information about modifying configuration settings, see Modifying Configuration Properties Using Cloudera Manager on page 60.

Configuring Health Monitoring

The initial health monitoring configuration is handled during the installation and configuration of your cluster, and most monitoring parameters have default settings. However, you can set or modify these at any time.

Depending on the service or role you select, and the configuration category, you can enable or disable health tests, determine when health tests cause alerts, or determine whether specific health tests are used in computing the overall health of a role or service. In most cases you can disable these "roll-up" health tests separately from the individual health tests.

As a rule, a health test whose result is considered "Concerning" or "Bad" is forwarded as an event to the Event Server. That includes health tests whose results are based on configured Warning or Critical thresholds, as well pass-fail type health tests. An event is also published when the health test result returns to normal.

You can control when an individual health test is forwarded as an event or as an alert by modifying the threshold values for the relevant health test.

Configuring Service Monitoring

1. Select **Clusters** > *cluster_name* > *service_name*.
2. Click the **Configuration** tab.
3. Select **Scope** > *service_name* **(Service-Wide)**.
4. Select **Category** > **Monitoring**.
5. Locate the property to change or search for it by typing its name in the Search box.
6. Configure the property.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

8. Return to the Home page by clicking the Cloudera Manager logo.
9. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Host Monitoring

1. Click the **Hosts** tab.
2. Select a host.
3. Click the **Configuration** tab.
4. Select **Scope** > **All**.
5. Click the **Monitoring** category.
6. Configure the property.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
8. Return to the Home page by clicking the Cloudera Manager logo.
9. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Directory Monitoring

Cloudera Manager can perform threshold-based monitoring of free space in the various directories on the hosts it monitors—such as log directories or checkpoint directories (for the Secondary NameNode).

These thresholds can be set in one of two ways—as absolute thresholds (in terms of MiB and GiB, and so on) or as percentages of space. As with other threshold properties, you can set values that trigger events at both the Warning and Critical levels.

If you set both thresholds, the Absolute Threshold setting is used.

Configuring Activity Monitoring

The Activity Monitor monitors the MapReduce MRv1 jobs running on your cluster. This also includes the higher-level activities, such as Pig, Hive, and Oozie workflows that run as MapReduce tasks.

You can monitor for slow-running jobs or jobs that fail, and [alert](#) on these events. To detect jobs that are running too slowly, you must configure a set of [activity duration rules](#) that specify what jobs to monitor, and what the limits on duration are for those jobs. A "slow activity" event occurs when a job exceeds the duration limit configured for it in an activity duration rule. Activity duration rules are not defined by default; you must configure these rules if you want to see events for jobs that exceed the duration defined by these rules.

To configure Activity Monitor settings:

1. Go to the MapReduce service.
2. Click the **Configuration** tab.
3. Select **Scope** > ***MapReduce service_name*** **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Specify one or more [activity duration rules](#).
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Activity Duration Rules

An ***activity duration rule*** is a regular expression (used to match an activity name (that is, a Job ID)) combined with a run time limit which the job should not exceed. You can add as many rules as you like, one per line, in the **Activity Duration Rules** property.

The format of each rule is *regex=number* where the *regex* is a [regular expression](#) to match against the activity name, and *number* is the job duration limit, in minutes. When a new activity starts, each *regex* expression is tested against the name of the activity for a match.

The list of rules is tested in order, and the first match found is used. For example, if the rule set is:

```
foo=10
bar=20
```

any activity named "foo" would be marked slow if it ran for more than 10 minutes. Any activity named "bar" would be marked slow if it ran for more than 20 minutes.

Since Java regular expressions can be used, if the rule set is:

```
foo.*=10
bar=20
```

any activity with a name that starts with "foo" (for example, fool, food, foot) matches the first rule.

If there is no match for an activity, then that activity is not monitored for job duration. However, you can add a "catch-all" as the last rule that always matches any name:

```
foo.*=10
bar=20
baz=30
.*=60
```

In this case, any job that runs longer than 60 minutes is marked slow and generates an event.

Configuring YARN Application Monitoring

You can configure the visibility of the YARN application monitoring results.

Configuring Application Visibility

To configure whether admin and non-admin users can view all applications, only that user's applications, or no applications:

1. Go to the YARN service.
2. Click the **Configuration** tab.
3. Select **Scope** > *YARN service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Set the **Applications List Visibility Settings** properties for admin and non-admin users.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Impala Query Monitoring

You can configure the visibility of the Impala query results and the size of the storage allocated to Impala query results.

Configuring Query Visibility

To configure whether admin and non-admin users can view all queries, only that user's queries, or no queries:

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope** > *Impala service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Set the **Visibility Settings** properties for admin and non-admin users.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Impala Query Data Store Maximum Size

The query store stores enough information to make the query searchable through the filter language.

1. Do one of the following:

    - Select **Clusters** > **Cloudera Management Service**.
    - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Configuration** tab.
3. Select **Scope** > **Service Monitor**.
4. Click the **Main** category.
5. In the **Impala Storage** section, set the **firehose_impala_storage_bytes** property. The default is 1 GiB.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

The `firehose_impala_storage_bytes` property determines the approximate amount of disk space dedicated to storing Impala query data. Once the store reaches its maximum size, older data is deleted to make room for newer queries. The disk usage is approximate because data deletion begins only when the limit has been reached.

Configuring Alerts
Enabling Activity Monitor Alerts

You can enable alerts when an activity runs [too slowly](#) or fails.

1. Go to the MapReduce service.
2. Click the **Configuration** tab.
3. Select **Scope** > *MapReduce service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Check the **Alert on Slow Activities** or **Alert on Activity Failure** checkboxes.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Enabling Configuration Change Alerts

Configuration change alerts can be set service wide, or on specific roles for the service.

1. Click a service, role, or host.
2. Click the **Configuration** tab.
3. Select **Scope** > **All**.
4. Click the **Monitoring** category.
5. Check the **Enable Configuration Change Alerts** checkbox.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Enabling HBase Alerts

1. Go to the HBase service.
2. Click the **Configuration** tab.
3. Select **Scope** > *HBase service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Set one of the region or Hbck alerts:

    - Hbck Region Error Count
    - Hbck Error Count

- Hbck Alert Error Codes
- Hbck Slow Run
- Region Health Canary Slow Run
- Canary Unhealthy Region Count
- Canary Unhealthy Region Percentage

6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Health Alerts

Enabling Health Alerts

You can enable alerts when the health of a role or service crosses a threshold.

1. Select **Clusters** > *cluster_name* > *service_name* or open the page for a role.
2. Click the **Configuration** tab.
3. Select **Scope** > *role_name* **or** *service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Check the **Enable Health Alerts for this Role** or **Enable Service Level Health Alerts** checkbox, depending on whether you are configuring a role or a service.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Modifying the Health Threshold

You can configure the threshold when a health alert is raised.

1. Select **Administration** > **Alerts**.
2. Click 🔧 to the right of **Health Alert Threshold**.
3. Select **Scope** > **Event Server**.
4. Click the **Main** category.
5. Select the **Bad** or **Concerning** option.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Alerts Transitioning Out of Alerting Health Threshold

You can configure an alert when a service or role instance transitions from an alerting to a non-alerting health threshold.

1. Select **Administration** > **Alerts**.
2. Click 🔧 to the right of **Alert on Transitions out of Alerting Health**.
3. Select **Scope** > *role_name* **or** *service_name* **(Service-Wide)**.
4. In the category **Event Server Default Group**, check the **Alert on Transitions out of Alerting Health** checkbox.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Return to the Home page by clicking the Cloudera Manager logo.
7. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Log Alerts

You can configure an alert when a daemon emits a log message that matches a specified regular expression. See Configuring Log Alerts on page 230.

Configuring Alert Delivery

You can configure alerts to be delivered by email or sent as SNMP traps. If you choose email delivery, you can add to or modify the list of alert recipient email addresses. You can also send a test alert email. See [Managing Alerts](#) on page 287.

> **Note:** If alerting is enabled for events, you can search for and view alerts in the Events tab, even if you do not have email notification configured.

Configuring Log Events

You can enable or disable the forwarding of selected log events to the Event Server. This is enabled by default, and is a service-wide setting (**Enable Log Event Capture**) for each service for which monitoring is provided. You can enable and disable event capture for CDH services or for the Cloudera Management Service.

> **Important:** We do not recommend logging to a network-mounted file system. If a role is writing its logs across the network, a network failure or the failure of a remote file system can cause that role to freeze up until the network recovers.

Configuring Logs

1. Go to a service.
2. Click the **Configuration** tab.
3. Select *role_name* **(Service-Wide)** > **Logs** .
4. Edit a log property.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Return to the Home page by clicking the Cloudera Manager logo.
7. Click the ⟳ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Logging Thresholds

A logging threshold determines what [level](#) of log message is reported. The available levels are:

- `TRACE` - Informational events finer-grained than DEBUG.
- `DEBUG` - Informational events useful to debug an application.
- `INFO` - Informational events that highlight progress at coarse-grained level.
- `WARN` - Events that indicate a potential problem which is handled by the application.
- `ERROR` - Error events that allows the application to continue running.
- `FATAL` - Very severe error events that typically lead the application to abort.

The number of messages is greater and severity is least for `TRACE`. The default setting is `INFO`.

1. Go to a service.
2. Click the **Configuration** tab.
3. Enter `Logging Threshold` in the **Search** text field.
4. For the desired role group, select a logging threshold level.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Return to the Home page by clicking the Cloudera Manager logo.
7. Click the ⟳ icon that is next to any stale services to invoke the cluster restart wizard.

Configuring Log Directories

1. Do one of the following:
   - Cluster:

1. On the **Home** > **Status** tab, click a cluster name.
2. Select **Configuration** > **Log Directories**.
3. Edit a *role_name* **Log Directory** property.

- Service:

  1. Go to a service.
  2. Click the **Configuration** tab.
  3. Select *role_name* **(Service-Wide)** > **Logs**.
  4. Edit the **Log Directory** property.

2. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
3. Return to the Home page by clicking the Cloudera Manager logo.
4. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Enabling and Disabling Log Event Capture

1. Select **Clusters** > *cluster_name* > *service_name*.
2. Click the **Configuration** tab.
3. Select **Scope** > *service_name* **(Service-Wide)**.
4. Click the **Monitoring** category.
5. Modify the **Enable Log Event Capture** setting.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Return to the Home page by clicking the Cloudera Manager logo.
8. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

You can also modify the rules that determine how log messages are turned into events. Editing these rules is not recommended.

For each role, there are rules that govern how its log messages are turned into events by the custom log4j appender for the role. These are defined in the **Rules to Extract Events from Log Files** property.

Configuring Which Log Messages Become Events

1. Select **Clusters** > *cluster_name* > *service_name*.
2. Click the **Configuration** tab.
3. Enter `Rules to Extract Events from Log Files` in the **Search** text field.
4. Click the **Monitoring** category.
5. Select the role group for the role for which you want to configure log events, or search for "Rules to Extract Events from Log Files". Note that for some roles there may be more than one role group, and you may need to modify all of them. The easiest way to ensure that you have found all occurrences of the property you need to modify is to search for the property by name. Cloudera Manager shows all copies of the property that matches the search filter.
6. In the Content field, edit the rules as needed. Rules can be written as regular expressions.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
8. Return to the Home page by clicking the Cloudera Manager logo.
9. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

Cloudera defines a number of rules by default. For example:

- The line `{"rate": 10, "threshold":"FATAL"}`, means log entries with severity `FATAL` should be forwarded as events, up to 10 a minute.
- The line `{"rate": 0, "exceptiontype": "java.io.EOFException"}`, means log entries with the exception `java.io.EOFException` should always be forwarded as an event.

The syntax for these rules is defined in the **Description** field for this property: the syntax lets you create rules that identify log messages based on log4j severity, message content matching, or the exception type. These rules must result in valid JSON.

> **Note:** Editing these rules is not recommended. Cloudera Manager provides a default set of rules that should be sufficient for most users.

Configuring Log Alerts

You specify that a log event should generate an alert (by setting `"alert":true` in the rule). If you specify a content match, the entire content must match — if you want to match on a partial string, you must provide wildcards as appropriate to allow matching the entire string.

Monitoring Clusters

There are several ways to monitor clusters.

The Clusters tab in the top navigation bar displays each cluster's services in its own section, with the Cloudera Management Service separately below. You can select the following cluster-specific pages: hosts, reports, activities, and resource management.

The **Home** > **Status** tab displays the clusters being managed by Cloudera Manager. Each cluster is displayed either in summary form or in full form depending on the configuration of the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property. When the number of clusters exceeds the value of the property, only cluster summary information displays.

To display a cluster Status page, click the cluster name on the **Home** > **Status** tab Status tab. The cluster Status page displays a table containing links to the Hosts page and the status pages of the services running in the cluster.

Each service row in the table has a menu of actions that you select by clicking

▼

and can contain one or more of the following indicators:

Indicator	Meaning	Description
❶ 2	Health issue	Indicates that the service has at least one health issue. The indicator shows the number of health issues at the highest severity level. If there are Bad health test results, the indicator is red. If there are no Bad health test results, but Concerning test results exist, then the indicator is yellow. No indicator is shown if there are no Bad or Concerning health test results. > **Important:** If there is one Bad health test result and two Concerning health results, there will be three health issues, but the number will be one. Click the indicator to display the **Health Issues** pop-up dialog box. By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show *n* concerning issue(s)** link.Click the link to display the Status page containing with details about the health test result.
⚒ 4	Configuration issue	Indicates that the service has at least one configuration issue. The indicator shows the number of configuration issues at the highest severity level. If there are configuration errors, the indicator is red. If there are no errors but configuration warnings exist, then the indicator is yellow. No indicator is shown if there are no configuration notifications.

Indicator	Meaning	Description
		Important: If there is one configuration error and two configuration warnings, there will be three configuration issues, but the number will be one. Click the indicator to display the **Configuration Issues** pop-up dialog box. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show _n_ warning(s)** link.Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.See [Managing Services](#) on page 193.
⏻ Restart Needed 🔁 Refresh Needed	Configuration modified	Indicates that at least one of a service's roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. Click the indicator to display the [Stale Configurations](#) on page 77 page.To bring the cluster up-to-date, click the Refresh or Restart button on the Stale Configurations page or follow the instructions in [Refreshing a Cluster](#) on page 85, [Restarting a Cluster](#) on page 85, or [Restarting Services and Instances after Configuration Changes](#) on page 64.
📤	Client configuration redeployment required	Indicates that the client configuration for a service should be redeployed. Click the indicator to display the [Stale Configurations](#) on page 77 page.To bring the cluster up-to-date, click the Deploy Client Configuration button on the Stale Configurations page or follow the instructions in [Manually Redeploying Client Configuration Files](#) on page 80.

The right side of the status page displays charts ([dashboard](#)) that summarize resource utilization (IO, CPU usage) and processing metrics. .

> **Note:** If you delete a cluster, the deleted cluster still displays in some charts. This is because the charts also show historical data. Over time, data from the deleted cluster will drop off as older data is replaced by more current data. You can work around this by:
>
> - Waiting for the data from the deleted cluster to drop off.
> - Editing the `where` clause of query for the chart to include only the cluster(s) you are interested in. (For example: `clusterDisplayName=Cluster_1`). You can revert to the original query at a later date, after the data for the deleted cluster has dropped off. See [Charting Time-Series Data](#) on page 303.
> - Deleting all data in the Host Monitor and Service Monitor storage directories and starting from scratch. You will, however, loose all historical data from both current and deleted clusters. See [Configuring Host Monitor Data Storage](#) on page 398 and [Configuring Service Monitor Data Storage](#) on page 397 to learn where the storage directories are located.

Monitoring Services

Cloudera Manager's Service Monitoring feature monitors dozens of service health and performance metrics about the services and role instances running on your cluster:

- Presents health and performance data in a variety of formats including interactive charts
- Monitors metrics against configurable thresholds

- Generates events related to system and service health and critical log entries and makes them available for searching and alerting
- Maintains a complete record of service-related actions and configuration changes

Monitoring Service Status

From a service page, you can:

- Monitor the status of the services running on your clusters.
- Manage the services and roles in your clusters.
- Add new services.
- Access the client configuration files generated by Cloudera Manager that enable Hadoop client users to work with the HDFS, MapReduce, HBase, and YARN services you added. (These configuration files are normally deployed automatically when you install a cluster or add a service).
- View the maintenance mode status of a cluster.

You can also pull down a menu from an individual service name to go directly to one of the tabs for that service to its Status, Instances, Commands, Configuration, Audits, or Charts Library tabs.

Viewing the URLs of the Client Configuration Files

To allow Hadoop client users to work with the services you created, Cloudera Manager generates client configuration files that contain the relevant configuration files with the settings from your services. These files are deployed automatically by Cloudera Manager based on the services you have installed, when you add a service, or when you add a Gateway role on a host.

You can manually download and distribute these client configuration files to the users of a service, if necessary.

The **Actions** > **Client Configuration URLs** command opens a pop-up that displays links to the client configuration zip files created for the services installed in your cluster. You can download these zip files by clicking the link.

The Actions button is not enabled if you are viewing status for a point of time in the past.

See Client Configuration Files on page 79 for more information on this topic.

Viewing the Status of a Service Instance

Do one of the following:

- In **Home** > **Status** tab, select *ClusterName* > *ServiceName*.
- Select **Clusters** > *ClusterName* > *ServiceName*.

This opens the **Status** page where you can view a variety of information about a service and its performance. See Viewing Service Status on page 233 for details.

Viewing the Health and Status of a Role Instance

Click the role instance under the **Role Counts** column.

If there is just one instance of this role, this opens the **Status** tab for the role instance.

If there are multiple instances of a role, clicking the role link under **Role Counts** will open the **Instances** tab for the service, showing instances of the role type you have selected. See Viewing Role Instance Status on page 237 for details.

If you are viewing a point in time in the past, the Role Count links will be greyed out, but still functional. Their behavior will depend on whether historical data is available for the role instance.

Viewing the Maintenance Mode Status of a Cluster

Select **Actions** > **View Maintenance Mode Status...** button to view the status of your cluster in terms of which components (service, roles, or hosts) are in maintenance mode. This pops up a dialog box that shows the components in your cluster that are in maintenance mode, and indicates which are in effective maintenance mode as well as those that have been explicitly placed into maintenance mode. (See Maintenance Mode on page 186 for an explanation of explicit maintenance mode and effective maintenance mode.)

From this dialog box you can select any of the components shown there and remove them from maintenance mode.

If individual services are in maintenance mode, you will see the maintenance mode icon ⟳ next to the **Actions** button for that service.

The Actions button is not enabled if you are viewing status for a point of time in the past.

Viewing Service Status

To view service status, do one of the following:

- In the **Home** > **Status** tab, if the cluster is displayed in full form, click *ServiceName* in a *ClusterName* table.
- In the **Home** > **Status** tab, click *ClusterName* and then click *ServiceName*.
- Select **Clusters** > *ClusterName* > *ServiceName*.

For all service types there is a **Status Summary** that shows, for each configured role, the overall status and health of the role instance(s).

> **Note:** Not all service types provide complete monitoring and health information. Hive, Hue, Oozie, Solr, and YARN (CDH 4 only) only provide the basic Status Summary on page 233.

Each service that supports monitoring provides a set of monitoring properties where you can enable or disable health tests and events, and set thresholds for tests and modify thresholds for the status of certain health tests. For more information see Configuring Monitoring Settings on page 223.

The HDFS, MapReduce, HBase, ZooKeeper, and Flume services also provide additional information: a snapshot of service-specific metrics, health test results, health history, and a set of charts that provide a historical view of metrics of interest.

Viewing Past Status

The health and status information on the Status page represents the state of the service or role instance at a given *point in time*. The charts (and the Logs and Events under Diagnostics) represent the time range selected on the Time Range Selector (which defaults to the past 30 minutes). You can view health, status, and chart historical data by expanding the Time Range Selector (click the mini line chart under "admin") and moving the time marker (◆) to a point in the past.

When you move the time marker to a point in the past (for services and roles that support health history), the entire Status page updates to the time selected. A Now button (▶▶) allows you to quickly return to the current state of the service. The Actions menu is disabled while viewing a past status to ensure that you cannot accidentally act on outdated status information.

See Time Line on page 215 for more details.

Status Summary

The Status Summary shows the status of each service instance being managed by Cloudera Manager. Even services such as Hue, Oozie, or YARN (which are not monitored by Cloudera Manager) show a status summary. The overall status for a service is a roll-up of the health test results for the service and all its role instances. The **Status** can be:

Table 8: Status

Indicator	Status	Description
⏻	Started with outdated configuration	For a service, this indicates the service is running, but at least one of its roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. For a role, this indicates a configuration change has been made that requires a restart, and that restart has not yet occurred. Click the indicator to display the Stale Configurations on page 77 page.

Indicator	Status	Description
	Starting	The entity is starting up but is not yet running.
	Stopping	The entity is stopping but has not stopped yet.
	Stopped	The entity is stopped, as expected.
	Down	The entity is not running, but it is expected to be running.
	History not available	Cloudera Manager is in historical mode, and the entity does not have historical monitoring support. This is the case for services other than HDFS, MapReduce and HBase such as ZooKeeper, Oozie, and Hue.
	None	The entity does not have a status. For example, it is not something that can be running and it cannot have health. Examples are the HDFS Balancer (which runs from the HDFS Rebalance action) or Gateway roles. The Start and Stop commands are not applicable to these instances.
	Good health	The entity is running with good health. For a specific health test, the returned result is normal or within the acceptable range. For a role or service, this means all health tests for that role or service are **Good**.
	Concerning health	The entity is running with concerning health. For a specific health test, the returned result indicates a potential problem. Typically this means the test result has gone above (or below) a configured Warning threshold. For a role or service, this means that at least one health test is **Concerning**.
	Bad health	The entity is running with bad health. For a specific health test, the test failed, or the returned result indicates a serious problem. Typically this means the test result has gone above (or below) a configured Critical threshold. For a role or service, this means that at least one health test is **Bad**.
	Disabled health	The entity is running, but all of its health tests are disabled.
	Unknown health	The status of a service or role instance is unknown. This can occur for a number of reasons, such as the Service Monitor is not running, or connectivity to the Agent doing the health monitoring has been lost.

To see the status of one or more role instances, click the role type link under **Status Summary**. If there is a single instance of the role type, the link directs you to the Status page of the role instance.

If there are multiple role instances (such as for DataNodes, TaskTrackers, and RegionServers), the role type link directs you to the Role Instances page for that role type. Click on each instance, under Role Type, to be taken to the corresponding Status page.

To display the results for each health test that applies to this role type, expand the **Health Tests** filter on the left and expand **Good Health**, **Warnings**, **Bad Health**, or **Disabled Health**. Health test results that have been filtered out by your role type selection appear as unavailable.

Service Summary

Some services (specifically HDFS, MapReduce, HBase, Flume, and ZooKeeper) provide additional statistics about their operation and performance. These are shown in a Summary panel at the left side of the page. The contents of this panel depend on the service:

- The HDFS Summary shows disk space usage.
- The MapReduce Summary shows statistics on slot usage, jobs and so on.
- The Flume Summary provides a link to a page of Flume metric details. See Flume Metric Details on page 235.

- The ZooKeeper Summary provides links to the ZooKeeper role instances (nodes) as well as Zxid information if you have a ZooKeeper Quorum (multiple ZooKeeper servers).

For example:

HDFS Summary

Configured Capacity	▬ 15.1 GiB/244.5 GiB
Quick Links	Replication , Reports , Browse Filesystem , NameNode Web UI (Active) ⬈
Event Search	Alerts ⬈ , Critical ⬈ , All ⬈

Other services such as Hue, Oozie, Impala, and Cloudera Manager itself, do not provide a Service Summary.

Health Tests and Health History

The Health Tests and Health History panels appear for HDFS, MapReduce, HBase, Flume, Impala, ZooKeeper, and the Cloudera Manager Service. Other services such as Hue, Oozie, and YARN do not provide a Health Test panel.

The Health Tests panel shows health test results in an expandable and collapsible list, typically with the specific metrics that the test returned. (You can Expand All or Collapse All from the links at the upper right of the Health Tests panel).

- The color of the text (and the background color of the field) for a Health Test result indicates the status of the results. The tests are sorted by their health status – Good, Concerning, Bad, or Disabled. The entries are collapsed by default. Click the arrow to the left of an entry to expand the entry and display further information.
- Clicking the **Details** link for a health test displays further information about the test, such as the meaning of the test and its possible results, suggestions for actions you can take or how to make configuration changes related to the test. The help text may include a link to the relevant monitoring configuration section for the service. See Configuring Monitoring Settings on page 223 for more information.
- In the Health Tests panel:

  - Clicking ❯ displays the lists of health tests that contributed to the health test.
  - Clicking the **Details** link displays further information about the health test.

- In the Health History panel:

  - Clicking ❯ displays the lists of health tests that contributed to the health history.
  - Clicking the **Show** link moves the time range to the historical time period.

Charts

HDFS, MapReduce, HBase, ZooKeeper, Flume, and Cloudera Management Service all display charts of some of the critical metrics related to their performance and health. Other services such as Hive, Hue, Oozie, and Solr do not provide charts.

See Viewing Charts for Cluster, Service, Role, and Host Instances on page 221 for detailed information on the charts that are presented, and the ability to search and display metrics of your choice.

Flume Metric Details

From the Flume Service Status page, click the **Flume Metric Details** link in the **Flume Summary** panel to display details of the Flume agent roles.

On this page you can view a variety of metrics about the Channels, Sources and Sinks you have configured for your various Flume agents. You can view both current and historical metrics on this page.

The **Channels** section shows the metrics for all the channel components in the Flume service. These include metrics related to the channel capacity and throughput.

The **Sinks** section shows metrics for all the sink components in the Flume service. These include event drain statistics as well as connection failure metrics.

The **Sources** section shows metrics for all the source components in the Flume service.

This page maintains the same navigation bar as the Flume service status page, so you can go directly to any of the other tabs (Instances, Commands, Configuration, or Audits).

Viewing Service Instance Details

1. Do one of the following:

   - In the **Home** > **Status** tab, if the cluster is displayed in full form, click *ServiceName* in a *ClusterName* table.
   - In the **Home** > **Status** tab, click *ClusterName* and then click *ServiceName*.
   - Select **Clusters** > ***ClusterName*** > ***ServiceName***.

2. Click the **Instances** tab on the service's navigation bar. This shows all instances of all role types configured for the selected service.

You can also go directly to the Instances page to view instances of a specific role type by clicking one of the links under the **Role Counts** column. This will show only instances of the role type you selected.

The Instances page displays the results of the configuration validation checks it performs for all the role instances for this service.

> **Note:** The information on this page is always the **Current** information for the selected service and roles. This page does not support a historical view: thus, the Time Range Selector is not available.

The information on this page shows:

- The name of the role instance. Click the name to view the role status for that role.
- The host on which it is running. Click the hostname to view the host status details for the host.
- The rack assignment.
- The status. A single value summarizing the state and health of the role instance.
- Whether the role is currently in maintenance mode. If the role has been set into maintenance mode explicitly, you will see the following icon ( ). If it is in effective maintenance mode due to the service or its host having been set into maintenance mode, the icon will be this ( ).
- Whether the role is currently decommissioned.

You can sort or filter the Instances list by criteria in any of the displayed columns:

- **Sort**

  1. Click the column header by which you want to sort. A small arrow indicates whether the sort is in ascending or descending order.
  2. Click the column header again to reverse the sort order.

- **Filter** - Type a property value in the Search box or select the value from the facets at the left of the page.

Role Instance Reference

The following tables contain reference information on the status, role state, and health columns for role instances.

Table 9: Status

Indicator	Status	Description
⏻	Started with outdated configuration	For a service, this indicates the service is running, but at least one of its roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. For a role, this indicates a configuration change has been made that requires a restart, and that restart has not yet occurred. Click the indicator to display the Stale Configurations on page 77 page.
	Starting	The entity is starting up but is not yet running.
	Stopping	The entity is stopping but has not stopped yet.
⏹	Stopped	The entity is stopped, as expected.
⊖	Down	The entity is not running, but it is expected to be running.
❓	History not available	Cloudera Manager is in historical mode, and the entity does not have historical monitoring support. This is the case for services other than HDFS, MapReduce and HBase such as ZooKeeper, Oozie, and Hue.
●	None	The entity does not have a status. For example, it is not something that can be running and it cannot have health. Examples are the HDFS Balancer (which runs from the HDFS Rebalance action) or Gateway roles. The Start and Stop commands are not applicable to these instances.
✅	Good health	The entity is running with good health. For a specific health test, the returned result is normal or within the acceptable range. For a role or service, this means all health tests for that role or service are **Good**.
🟡	Concerning health	The entity is running with concerning health. For a specific health test, the returned result indicates a potential problem. Typically this means the test result has gone above (or below) a configured Warning threshold. For a role or service, this means that at least one health test is **Concerning**.
❗	Bad health	The entity is running with bad health. For a specific health test, the test failed, or the returned result indicates a serious problem. Typically this means the test result has gone above (or below) a configured Critical threshold. For a role or service, this means that at least one health test is **Bad**.
⊘	Disabled health	The entity is running, but all of its health tests are disabled.
❓	Unknown health	The status of a service or role instance is unknown. This can occur for a number of reasons, such as the Service Monitor is not running, or connectivity to the Agent doing the health monitoring has been lost.

Viewing Role Instance Status

To view status for a role instance:

1. Select a service instance to display the **Status** page for that service.
2. Click the **Instances** tab.
3. From the list of roles, select one to display that role instance's **Status** page.

The Actions Menu

Minimum Required Role: Operator (also provided by **Configurator**, **Cluster Administrator**, **Full Administrator**)

The **Actions** menu provides a list of commands relevant to the role type you are viewing. These commands typically include Stopping, Starting, or Restarting the role instance, accessing the Web UI for the role, and may include many other commands, depending on the role you are viewing.

The **Actions** menu is available from the Role Status page only when you are viewing **Current** time status. The menu is disabled if you are viewing a point of time in the past.

Viewing Past Status

The status and health information shown on this page represents the state of the service or role instance at a given point in time. The exceptions are the charts tabs, which show information for the time range currently selected on the Time Range Selector (which defaults to the past 30 minutes). By default, the information shown on this page is for the current time. You can view status for a past point in time simply by moving the time marker (♦) to a point in the past.

When you move the time marker to a point in the past (for Services/Roles that support health history), the Health Status clearly indicates that it is referring to a past time. A Now button (▶▶) enables you to quickly switch to view the current state of the service. In addition, the Actions menu is disabled while you are viewing status in the past – to ensure that you cannot accidentally take an action based on outdated status information. See Time Line on page 215 for more details.

You can also view past status by clicking the **Show** link in the Health Tests and Health History on page 238 panel.

Summary

The Summary panel provides basic information about the role instance, where it resides, and the health of its host.

All role types provide the **Summary** panel. Some role instances related to HDFS, MapReduce, and HBase also provide a Health Tests panel and associated charts.

Health Tests and Health History

The Health Tests and Health History panels are shown for roles that are related to HDFS, MapReduce, or HBase. Roles related to other services such as Hue, ZooKeeper, Oozie, and Cloudera Manager itself, do not provide a Health Tests panel. The Health Tests panel shows health test results in an expandable/collapsible list, typically with the specific metrics that the test returned. (You can Expand All or Collapse All from the links at the upper right of the Health Tests panel).

- The color of the text (and the background color of the field) for a Health Test result indicates the status of the results. The tests are sorted by their health status – Good, Concerning, Bad, or Disabled. The entries are collapsed by default. Click the arrow to the left of an entry to expand the entry and display further information.
- Clicking the **Details** link for a health test displays further information about the test, such as the meaning of the test and its possible results, suggestions for actions you can take or how to make configuration changes related to the test. The help text may include a link to the relevant monitoring configuration section for the service. See Configuring Monitoring Settings on page 223 for more information.
- In the Health Tests panel:
    - Clicking ❯ displays the lists of health tests that contributed to the health test.
    - Clicking the **Details** link displays further information about the health test.

- In the Health History panel:
    - Clicking ❯ displays the lists of health tests that contributed to the health history.
    - Clicking the **Show** link moves the time range to the historical time period.

Status Summary

The Status Summary panel reports a roll-up of the status of all the roles.

Charts

Charts are shown for roles that are related to HDFS, MapReduce, HBase, ZooKeeper, Flume, and Cloudera Management Service. Roles related to other services such as Hue, Hive, Oozie, and YARN, do not provide charts.

See [Viewing Charts for Cluster, Service, Role, and Host Instances](#) on page 221 for detailed information on the charts that are presented, and the ability to search and display metrics of your choice.

The Processes Tab

To view the processes running for a role instance:

1. Select a service instance to display the Status page for that service.
2. Click the **Instances** tab.
3. From the list of roles, select one to display that role instance's Status page.
4. Click the **Processes** tab.

The Processes page shows the processes that run as part of this service role, with a variety of metrics about those processes.

- To see the location of a process' configuration files, and to view the Environment variable settings, click the **Show** link under **Configuration Files/Environment**.
- If the process provides a Web UI (as is the case for the NameNode, for example) click the link to open the Web UI for that process
- To see the most recent log entries, click the **Show Recent Logs** link.
- To see the full log, stderr, or stdout log files, click the appropriate links.

Running Diagnostic Commands for Roles

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

Cloudera Manager allows administrators to run the following diagnostic utility tools against most Java-based role processes:

- List Open Files (`lsof`) - Lists the open files of the process.
- Collect Stack Traces (`jstack`) - Captures Java thread stack traces for the process.
- Heap Dump (`jmap`) - Captures a heap dump for the process.
- Heap Histogram (`jmap -histo`) - Produces a histogram of the heap for the process.

These commands are found on the **Actions** menu of the Cloudera Manager page for the instance of the role. For example, to run diagnostics commands for the HDFS active NameNode, perform these steps:

1. Click the HDFS service on the **Home** > **Status** tab or select it on the Clusters menu.
2. Click **Instances** > **NameNode (Active)**.
3. Click the **Actions** menu.
4. Choose one of the diagnostics commands listed in the lower section of the menu.
5. Click the button in the confirmation dialog box to confirm your choice.
6. When the command is executed, click **Download Result Data** and save the file to view the command output.

Periodic Stacks Collection

Periodic stacks collection allows you to enable and configure the periodic collection of thread stack traces in Cloudera Manager. When stacks collection is enabled for a role, call stacks are output to a log file at regular intervals. The logs can help with diagnosis of performance issues such as deadlock, slow processing, or excessive numbers of threads.

Stacks collection may impact performance for the processes being collected as well as other processes on the host, and is turned off by default. For troubleshooting performance issues, you may be asked by Cloudera Support to enable stacks collection and send the resulting logs to Cloudera for analysis.

Stacks collection is available for the majority of roles in Cloudera Manager. For the HDFS service, for example, you can enable stacks collection for the DataNode, NameNode, Failover Controller, HttpFS, JournalNode, and NFS Gateway. If the **Stacks Collection** category does not appear in the role's configuration settings, the feature is not available for that role.

Configuring Periodic Stacks Collection

To enable and configure periodic stacks collection, open the Cloudera Manager page for a specific service or role. Access the configuration settings in one of the following ways:

- From the service page in Cloudera Manager:
  - Click the **Configuration** tab.
  - Select **Scope** > **NameNode**.
  - Select **Category** > **Stacks Collection**.

- From the service page in Cloudera Manager:
  - Click the **Instances** tab.
  - Click the **Configuration** tab.
  - Select **Scope** > *role type*.
  - Select **Category** > **Stacks Collection**.

The configuration settings are as follows:

- **Stacks Collection Enabled** - Whether or not periodic stacks collection is enabled.
- **Stacks Collection Directory** - The directory in which stack logs will be placed. If not set, stacks will be logged into a `stacks` subdirectory of the role's log directory.
- **Stacks Collection Frequency** - The frequency with which stacks will be collected.
- **Stacks Collection Data Retention** - The amount of stacks data that will be retained. When the retention limit is reached, the oldest data will be deleted.
- **Stacks Collection Method** - The method that will be used to collect stacks. The jstack option involves periodically running the jstack command against the role's daemon process. The servlet method is available for those roles with an HTTP server endpoint that exposes the current stacks traces of all threads. When the servlet method is selected, that HTTP endpoint is periodically scraped.

As an example, to configure stacks collection for an HDFS NameNode, perform the following steps:

1. Go to the HDFS service page.
2. Click the **Configuration** tab.
3. Select **Scope** > **NameNode**.
4. Select **Category** > **Stacks Collection**.
5. Locate the property or search for it by typing its name in the Search box.
6. Modify the configuration settings if desired.
7. Click **Save Changes**.

Stacks collection configuration settings are stored in a per-role configuration file called `cloudera-stacks-monitor.properties`. Cloudera Manager reads the configuration file and coordinates stack collection. Changes to the configuration settings take effect after a short delay. It is not necessary to restart the role.

Viewing and Downloading Stacks Logs

Stacks are collected and logged to a compressed, rotated log file. A certain amount of the log data is in an uncompressed file. When that file reaches a limit, the file is rotated and bzip2 compressed. Once the total number of files exceeds the configured retention limit, the oldest files are deleted.

Collected stacks data is available for download through the Cloudera Manager UI and API. To view or download stacks logs through the UI, perform the following steps:

1. On the service page, click the **Instances** tab.
2. Click the role in the **Role Type** column.
3. In the Summary section of the role page, click **Stacks Logs**.
4. Click **Stacks Log File** to view the most recent stacks file. Click **Download Stacks Logs** to download a zipped bundle of the stacks logs.

Viewing Running and Recent Commands

Viewing Running and Recent Commands For a Cluster

The indicator positioned just to the left of the Search field on the right side of the Admin Console main navigation bar displays the number of commands currently running for all services or roles. To display the running commands, click the indicator.

To display all commands that have run and finished recently, do one of the following:

- Click the **All Recent Commands** button in the window that pops up when you click the indicator. This command displays information on all running and recent commands in the same form, as described below.
- Click the Cloudera Manager logo in the main navigation bar and click the **All Recent Commands** tab.

Select a value from the pager to control how many commands are listed, or click the arrows to view pages.

The command indicator shows the number of commands running on all clusters you are managing. Likewise, **All Recent Commands** shows all commands that were run and finished within the search time range you specified, across all your managed clusters.

Viewing Running and Recent Commands for a Service or Role

For a selected service or role instance, the **Commands** tab shows which commands are running or have been run for that instance, and what the status, progress, and results are. For example, if you go to the HDFS service shortly after you have installed your cluster and look at the **Commands** tab, you will see recent commands that created the directories, started the HDFS role instances (the NameNode, Secondary NameNode, and DataNode instances), and the command that initially formatted HDFS on the NameNode. This information is useful if a service or role seems to be taking a long time to start up or shut down, or if services or roles are not running or do not appear to have been started correctly. You can view both the status and progress of currently running commands, as well as the status and results of commands run in the past.

1. Click the **Clusters** tab on the top navigation bar.
2. Click the service name to go to the Status tab for that service.
3. For a role instance, click the **Instances** tab and select the role instance name to go to its Status tab.
4. Click the **Commands** tab.

Command Details

The details available for a command depend on whether the command is running or recently completed.

Running Commands

The Running Commands area shows commands that are in progress.

While the status of the command is **In Progress**, an **Abort** button displays so that you can abort the command if necessary.

The Commands status information is updated automatically while the command is running.

After the command has finished running (all its subcommands have finished), the status is updated, the **Abort** buttons disappear, and the information for **Recent Commands** appears as described below.

Recent Commands

The Recent Commands area shows commands that were run and finished within the search time range you specified.

If no commands were run during the selected time range, you can double the time range selection by clicking the **Try expanding the time range selection** link. If you are in the "current time" mode, the beginning time will move; if you

are looking at a time range in the past, both the beginning and ending times of the range are changed. You can also change the time range using the options described in Time Line on page 215.

Select a value from the pager Display 25 ⬍ Per Page | << < **1 - 25** > to control how many commands are listed, or click the arrows to view pages.

Commands are shown with the most recent ones at the top.

The icon associated with the status (which typically includes the time that the command finished) plus the result message tells you whether the command succeeded ✔ or failed ⊗. If the command failed, it indicates if it was one of the subcommands that actually failed. In many cases, multiple subcommands result from the top level command.

The **First Run** command runs during the initial startup of your cluster. Click this link to view the command history of the cluster startup.

Command Details

In the Running Commands dialog box or Recent Commands page, click a command in the **Command** column to display its details and any subcommands. The page title is the name of the command.

The **Summary** section at the top shows information about the command:

- The current status
- The context, which can be a cluster, service, host, or role
- The time the command started
- The duration of the command
- A message about the command completion
- If the context is a role, links to role instance logs

The **Details** section shows how many steps, if any, the selected command has and lists any subcommands.

Expand a command to view subcommands. In the Running Commands dialog box, each subcommand also has an **Abort** button that is present as long as the subcommand is in progress.

You can perform the following actions:

- Select the option to display all the subcommands or only failed or running commands.
- Click the link in the **Context** column to go to the **Status** page for the component (host, service, or role instance) to which this command is related.
- Click a **Role Log** tab to display the log for that role, and `stdout` and `stderr` if available for the role.

Monitoring Resource Management

Statically allocating resources using cgroups is configurable through a single *static service pool* wizard. You allocate services as a percentage of total resources, and the wizard configures the cgroups.

Monitoring Static Service Pools

Static service pools isolate the services in your cluster from one another, so that load on one service has a bounded impact on other services. Services are allocated a static percentage of total resources—CPU, memory, and I/O weight—which are not shared with other services. When you configure static service pools, Cloudera Manager computes recommended memory, CPU, and I/O configurations for the worker roles of the services that correspond to the percentage assigned to each service. Static service pools are implemented per role group within a cluster, using Linux control groups (cgroups) and cooperative memory limits (for example, Java maximum heap sizes). Static service pools can be used to control access to resources by HBase, HDFS, Impala, MapReduce, Solr, Spark, YARN, and add-on services. Static service pools are not enabled by default.

Viewing Static Service Pools

Select **Clusters** > *Cluster name* > **Static Service Pools**.If the cluster has a YARN service, the Static Service Pools Status tab displays and shows whether resource management is enabled for the cluster, and the currently configured service pools.

Static Service Pool Status

The Status tab of the Static Service Pools page contains a list of current services that can or have been allocated resources and a set of resource usage charts for the cluster.

Click **Historical Data** to display detailed resource usage charts for each service.

Click a duration link   30m 1h 2h 6h 12h 1d 7d 30d   at the top right of the charts to change the time period for which the resource usage displays.

Monitoring Dynamic Resource Pools

A **dynamic resource pool** is a named configuration of resources and a policy for scheduling the resources among YARN applications and Impala queries running in the pool. Dynamic resource pools allow you to schedule and allocate resources to YARN applications and Impala queries based on a user's access to specific pools and the resources available to those pools. If a pool's allocation is not in use, it can be [preempted](#) and distributed to other pools. Otherwise, a pool receives a share of resources according to the pool's weight. Access control lists (ACLs) restrict who can submit work to dynamic resource pools and administer them.

Viewing Dynamic Resource Pools

1. Go to the YARN service.
2. Click the **Resource Pools** tab.

Click a duration link   30m 1h 2h 6h 12h 1d 7d 30d   at the top right of the charts to change the time period for which the resource usage displays.

- **Status** - a summary of the virtual CPU cores and memory that can be allocated by the YARN scheduler.
- **Resource Pools Usage** - a list of pools that have been explicitly configured and pools created by YARN, and properties of the pools. The **Configuration** link takes you to the [Dynamic Resource Pool Configuration](#) page.

    - **Allocated Memory** - The memory assigned to the pool that is currently allocated to applications and queries.
    - **Allocated VCores** - The number of virtual CPU cores assigned to the pool that are currently allocated to applications and queries.
    - **Allocated Containers** - The number of YARN containers assigned to the pool whose resources have been allocated.
    - **Pending Containers** - The number of YARN containers assigned to the pool whose resources are pending.

Monitoring Hosts

Cloudera Manager's Host Monitoring features let you manage and monitor the status of the hosts in your clusters.

Viewing All Hosts

To display summary information about all the hosts managed by Cloudera Manager, click **Hosts** in the main navigation bar. The All Hosts page displays with a list of all the hosts managed by Cloudera Manager.

The list of hosts shows the overall status of the Cloudera Manager-managed hosts in your cluster.

- The information provided varies depending on which columns are selected. To change the columns, click the **Columns: *n* Selected** drop-down and select the checkboxes next to the columns to display.
- Click ❯ to the left of the number of roles to list all the role instances running on that host.

- Filter the hosts list by entering search terms (hostname, IP address, or role) in the search box separated by commas or spaces. Use quotes for exact matches (for example, strings that contain spaces, such as a role name) and brackets to search for ranges. Hosts that match any of the search terms are displayed. For example:

```
hostname[1-3], hostname8 hostname9, "hostname.example.com"
hostname.example.com "HDFS DataNode"
```

- You can also search for hosts by selecting a value from the facets in the **Filters** section at the left of the page.
- If the Configuring Agent Heartbeat and Health Status Options on page 34 are configured as follows:

  - Send Agent heartbeat every *x*
  - Set health status to Concerning if the Agent heartbeats fail *y*
  - Set health status to Bad if the Agent heartbeats fail *z*

  The value *v* for a host's Last Heartbeat facet is computed as follows:

  - $v < x * y$ = Good
  - $v >= x * y$ and $<= x * z$ = Concerning
  - $v >= x * z$ = Bad

Role Assignments

You can view the assignment of roles to hosts as follows:

1. Click the **Roles** tab.
2. Click a cluster name or **All Clusters**.

Disks Overview

Click the **Disks Overview** tab to display an overview of the status of all disks in the deployment. The statistics exposed match or build on those in `iostat`, and are shown in a series of histograms that by default cover every physical disk in the system.

Adjust the endpoints of the time line to see the statistics for different time periods. Specify a filter in the box to limit the displayed data. For example, to see the disks for a single rack rack1, set the filter to: `logicalPartition = false and rackId = "rack1"` and click **Filter**. Click a histogram to drill down and identify outliers. Mouse over the graph and click ↗ to display additional information about the chart.

Viewing the Hosts in a Cluster

Do one of the following:

- Select **Clusters** > *Cluster name* > **Hosts**.
- In the Home screen, click ▤ Hosts in a full form cluster table.

The All Hosts page displays with a list of the hosts filtered by the cluster name.

Viewing Individual Hosts

You can view detailed information about an individual host—resources (CPU/memory/storage) used and available, which processes it is running, details about the host agent, and much more—by clicking a host link on the All Hosts page. See Host Details on page 244.

Host Details

You can view detailed information about each host, including:

- Name, IP address, rack ID
- Health status of the host and last time the Cloudera Manager Agent sent a heartbeat to the Cloudera Manager Server
- Number of cores

- System load averages for the past 1, 5, and 15 minutes
- Memory usage
- File system disks, their mount points, and usage
- Health test results for the host
- Charts showing a variety of metrics and health test results over time.
- Role instances running on the host and their health
- CPU, memory, and disk resources used for each role instance

To view detailed host information:

1. Click the **Hosts** tab.
2. Click the name of one of the hosts. The Status page is displayed for the host you selected.
3. Click tabs to access specific categories of information. Each tab provides various categories of information about the host, its services, components, and configuration.

From the status page you can view details about several categories of information.

Status

The Status page is displayed when a host is initially selected and provides summary information about the status of the selected host. Use this page to gain a general understanding of work being done by the system, the configuration, and health status.

If this host has been decommissioned or is in maintenance mode, you will see the following icon(s) (,) in the top bar of the page next to the status message.

Details

This panel provides basic system configuration such as the host's IP address, rack, health status summary, and disk and CPU resources. This information summarizes much of the detailed information provided in other panes on this tab. To view details about the Host agent, click the Host Agent link in the Details section.

Health Tests

Cloudera Manager monitors a variety of metrics that are used to indicate whether a host is functioning as expected. The Health Tests panel shows health test results in an expandable/collapsible list, typically with the specific metrics that the test returned. (You can Expand All or Collapse All from the links at the upper right of the Health Tests panel).

- The color of the text (and the background color of the field) for a health test result indicates the status of the results. The tests are sorted by their health status – Good, Concerning, Bad, or Disabled. The list of entries for good and disabled health tests are collapsed by default; however, Bad or Concerning results are shown expanded.
- The text of a health test also acts as a link to further information about the test. Clicking the text will pop up a window with further information, such as the meaning of the test and its possible results, suggestions for actions you can take or how to make configuration changes related to the test. The help text for a health test also provides a link to the relevant monitoring configuration section for the service. See Configuring Monitoring Settings on page 223 for more information.

Health History

The Health History provides a record of state transitions of the health tests for the host.

- Click the arrow symbol at the left to view the description of the health test state change.
- Click the **View** link to open a new page that shows the state of the host at the time of the transition. In this view some of the status settings are greyed out, as they reflect a time in the past, not the current status.

File Systems

The File systems panel provides information about disks, their mount points and usage. Use this information to determine if additional disk space is required.

Roles

Use the Roles panel to see the role instances running on the selected host, as well as each instance's status and health. Hosts are configured with one or more role instances, each of which corresponds to a service. The role indicates which daemon runs on the host. Some examples of roles include the NameNode, Secondary NameNode, Balancer, JobTrackers, DataNodes, RegionServers and so on. Typically a host will run multiple roles in support of the various services running in the cluster.

Clicking the role name takes you to the role instance's status page.

You can delete a role from the host from the Instances tab of the Service page for the parent service of the role. You can add a role to a host in the same way. See Role Instances on page 209.

Charts

Charts are shown for each host instance in your cluster.

See Viewing Charts for Cluster, Service, Role, and Host Instances on page 221 for detailed information on the charts that are presented, and the ability to search and display metrics of your choice.

Processes

The Processes page provides information about each of the processes that are currently running on this host. Use this page to access management web UIs, check process status, and access log information.

> **Note:** The Processes page may display exited startup processes. Such processes are cleaned up within a day.

The Processes tab includes a variety of categories of information.

- **Service** - The name of the service. Clicking the service name takes you to the service status page. Using the triangle to the right of the service name, you can directly access the tabs on the role page (such as the Instances, Commands, Configuration, Audits, or Charts Library tabs).
- **Instance** - The role instance on this host that is associated with the service. Clicking the role name takes you to the role instance's status page. Using the triangle to the right of the role name, you can directly access the tabs on the role page (such as the Processes, Commands, Configuration, Audits, or Charts Library tabs) as well as the status page for the parent service of the role.
- **Name** - The process name.
- **Links** - Links to management interfaces for this role instance on this system. These is not available in all cases.
- **Status** - The current status for the process. Statuses include stopped, starting, running, and paused.
- **PID** - The unique process identifier.
- **Uptime** - The length of time this process has been running.
- **Full log file** - A link to the full log (a file external to Cloudera Manager) for this host log entries for this host.
- **Stderr** - A link to the stderr log (a file external to Cloudera Manager) for this host.
- **Stdout** - A link to the stdout log (a file external to Cloudera Manager) for this host.

Resources

The Resources page provides information about the resources (CPU, memory, disk, and ports) used by every service and role instance running on the selected host.

Each entry on this page lists:

- The service name
- The name of the particular instance of this service
- A brief description of the resource
- The amount of the resource being consumed or the settings for the resource

The resource information provided depends on the type of resource:

- **CPU** - An approximate percentage of the CPU resource consumed.
- **Memory** - The number of bytes consumed.
- **Disk** - The disk location where this service stores information.
- **Ports** - The port number being used by the service to establish network connections.

Commands

The Commands page shows you running or recent commands for the host you are viewing. See <u>Viewing Running and Recent Commands</u> on page 241 for more information.

Configuration

Minimum Required Role: <u>Full Administrator</u>

The Configuration page for a host lets you set properties for the selected host. You can set properties in the following categories:

- **Advanced** - Advanced configuration properties. These include the Java Home Directory, which explicitly sets the value of `JAVA_HOME` for all processes. This overrides the auto-detection logic that is normally used.
- **Monitoring** - Monitoring properties for this host. The monitoring settings you make on this page will override the global host monitoring settings you make on the Configuration tab of the Hosts page. You can configure monitoring properties for:
    - health check thresholds
    - the amount of free space on the filesystem containing the Cloudera Manager Agent's log and process directories
    - a variety of conditions related to memory usage and other properties
    - alerts for health check events

  For some monitoring properties, you can set thresholds as either a percentage or an absolute value (in bytes).

- **Other** - Other configuration properties.
- **Parcels** - Configuration properties related to parcels. Includes the **Parcel Director** property, the directory that parcels will be installed into on this host. If the `parcel_dir` variable is set in the Agent's `config.ini` file, it will override this value.
- **Resource Management** - Enables resource management using control groups (cgroups).

For more information, see the description for each or property or see <u>Modifying Configuration Properties Using Cloudera Manager</u> on page 60.

Components

The Components page lists every component installed on this host. This may include components that have been installed but have not been added as a service (such as YARN, Flume, or Impala).

This includes the following information:

- **Component** - The name of the component.
- **Version** - The version of CDH from which each component came.
- **Component Version** - The detailed version number for each component.

Audits

The Audits page lets you filter for audit events related to this host. See <u>Lifecycle and Security Auditing</u> on page 301 for more information.

Charts Library

The Charts Library page for a host instance provides charts for all metrics kept for that host instance, organized by category. Each category is collapsible/expandable. See <u>Viewing Charts for Cluster, Service, Role, and Host Instances</u> on page 221 for more information.

Host Inspector

You can use the host inspector to gather information about hosts that Cloudera Manager is currently managing. You can review this information to better understand system status and troubleshoot any existing issues. For example, you might use this information to investigate potential DNS misconfiguration.

The inspector runs tests to gather information for functional areas including:

- Networking
- System time
- User and group configuration
- HDFS settings
- Component versions

Common cases in which this information is useful include:

- Installing components
- Upgrading components
- Adding hosts to a cluster
- Removing hosts from a cluster

Running the Host Inspector

1. Click the **Hosts** tab and select **All Hosts**.
2. Click the **Inspect All Hosts** button. Cloudera Manager begins several tasks to inspect the managed hosts.
3. After the inspection completes, click **Download Result Data** or **Show Inspector Results** to review the results.

The results of the inspection displays a list of all the validations and their results, and a summary of all the components installed on your managed hosts.

If the validation process finds problems, the **Validations** section will indicate the problem. In some cases the message may indicate actions you can take to resolve the problem. If an issue exists on multiple hosts, you may be able to view the list of occurrences by clicking a small triangle that appears at the end of the message.

The **Version Summary** section shows all the components that are available from Cloudera, their versions (if known) and the CDH distribution to which they belong.

Viewing Past Host Inspector Results

You can view the results of a past host inspection by looking for the Host Inspector command using the **Recent Commands** feature.

1.

   Click the Running Commands indicator (   ) just to the left of the Search box at the right hand side of the navigation bar.
2. Click the **Recent Commands** button.
3. If the command is too far in the past, you can use the Time Range Selector to move the time range back to cover the time period you want.
4. When you find the Host Inspector command, click its name to display its subcommands.
5. Click the **Show Inspector Results** button to view the report.

See <u>Viewing Running and Recent Commands</u> on page 241 for more information about viewing past command activity.

Monitoring Activities

Cloudera Manager's activity monitoring capability monitors the MapReduce, Pig, Hive, Oozie, and streaming jobs, Impala queries, and YARN applications running or that have run on your cluster. When the individual jobs are part of larger workflows (using Oozie, Hive, or Pig), these jobs are aggregated into MapReduce jobs that can be monitored as a whole, as well as by the component jobs.

If you are running multiple clusters, there will be a separate link in the Clusters tab for each cluster's MapReduce activities, Impala queries, and YARN applications.

The following sections describe how to view and monitor activities that run on your cluster.

Monitoring MapReduce Jobs

A MapReduce job is a unit of processing (query or transformation) on the data stored within a Hadoop cluster. You can view information about the different jobs that have run in your cluster during a selected time span.

- The list of jobs provides specific metrics about the jobs that were submitted, were running, or finished within the time frame you select.
- You can select charts that show a variety of metrics of interest, either for the cluster as a whole or for individual jobs.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details).

> **Note:** Activity Monitor treats the original job start time as immutable. If a job is resubmitted due to failover it will retain its original start time.

You can select an activity and drill down to look at the jobs and tasks spawned by that job:

- View the children (MapReduce jobs) of a Pig or Hive activity.
- View the task attempts generated by a MapReduce job.
- View the children (MapReduce, Pig, or Hive activities) of an Oozie job.
- View the activity or job statistics in a detail report format.
- Compare the selected activity to a set of other similar activities, to determine if the selected activity showed anomalous behavior. For example, if a standard job suddenly runs much longer than usual, this may indicate issues with your cluster.
- Display the distribution of task attempts that made up a job, by different metrics compared to task duration. You can use this, for example, to determine if tasks running on a certain host are performing slower than average.
- Kill a running job, if necessary.

> **Note:** Some activity data is sampled at one-minute intervals. This means that if you run a very short job that both starts and ends within the sampling interval, it may not be detected by the Activity Monitor, and thus will not appear in the Activities list or charts.

Viewing and Filtering MapReduce Activities

This section describes the various actions you can perform in the MapReduce Activities page:

- Viewing MapReduce Activities on page 249
- Selecting Columns to Show in the Activities List on page 251
- Sorting the Activities List on page 251
- Filtering the Activities List on page 251
- Activity Charts on page 252

Viewing MapReduce Activities

1. Select **Clusters** > *Cluster name* > *MapReduce service name* **Jobs**. The *MapReduce service name* page displays a list of activities. The columns in the Activities list show statistics about the performance of and resources used by each activity (and you can modify the default display by adding or removing columns).

   - The leftmost column holds a shortcut menu button (▼). Click this button to display a menu of commands relevant to the job shown in that row. The possible commands are:

Children	For a Pig, Hive or Oozie activity, takes you to the **Children** tab of the individual activity page. You can also go to this page by clicking the activity ID in the activity list. This command only appears for Pig, Hive or Oozie activities.
Tasks	For a MapReduce job, takes you to the **Tasks** tab of the individual job page. You can also go to this page by clicking the job ID in the activity or activity children list. This command only appears for a MapReduce job.
Details	Takes you to the **Details** tab where you can view the activity or job statistics in report form.
Compare	Takes you to the **Compare** tab where you can see how the selected activity compares to other similar activities in terms of a wide variety of metrics.
Task Distribution	Takes you to the **Task Distribution** tab where you can view the distribution of task attempts that made up this job, by amount of data and task duration. This command is available for MapReduce and Streaming jobs.
Kill Job	A pop-up asks for confirmation that you want to kill the job. This command is available only for MapReduce and Streaming jobs.

-
  The second column shows a chart icon (   ). Select this to chart statistics for the job. If there are charts showing similar statistics for the cluster or for other jobs, the statistics for the job are added to the chart. See Activity Charts on page 252 for more details.
- The third column shows the status of the job, if the activity is a MapReduce job:

🏷	The job has been submitted.
▶	The job has been started.
✔	The job is assumed to have succeeded.
✓	The job has finished successfully.
❓	The job's final state is unknown.
⏸	The job has been suspended.
✕	The job has failed.
⚐	The job has been killed.

- The fourth column shows the type of activity:

🖥	MapReduce job
🐷	Pig job
🐝	Hive job
🔵	Oozie job
🐵	Streaming job

Selecting Columns to Show in the Activities List

In the Activities list, you can display or hide any of the statistics that Cloudera Manager collects. By default only a subset of the possible statistics are displayed.

1. Click the **Select Columns to Display** icon (▥). A pop-up panel lets you turn on or off a variety of metrics that may be of interest.
2. Check or uncheck the columns you want to include or remove from the display. As you check or uncheck an item, its column immediately appears or disappears from the display.
3. Click the ✕ in the upper right corner to close the panel.

> **Note:** You cannot hide the shortcut menu or chart icon columns. Also, column selections are retained only for the current session.

Sorting the Activities List

You can sort the Activities list by the contents of any column:

1. Click the column header to initiate a sort. The small arrow that appears next to the column header indicates the sort direction.
2. Click the column header to reverse the sort direction.

Filtering the Activities List

You can filter the list of activities based on values of any of the metrics that are available. You can also easily filter for certain common queries from the drop-down menu next to the Search button at the top of the Activities list. By default, it is set to show **All Activities**.

To use one of the predefined filters:

- Click the
  ▾
  to the right of the **Search** button and select the filter you want to run. There are predefined filters to search by job type (for example Pig activities, MapReduce jobs, and so on) or for running, failed, or long-running activities.

To create a filter:

1. Click the
   ▾
   to the right of the **Search** button and select **Custom**.
2. Select a metric from the drop-down list in the first field; you can create a filter based on any of the available metrics.
3. Once you select a metric, fill in the rest of the fields; your choices depend on the type of metric you have selected. Use the percent character `%` as a wildcard in a string; for example, `Id matches job%0001` will look for any MapReduce job ID with suffix `0001`.
4. To create a compound filter, click the plus icon at the end of the filter row to add another row. If you combine filter criteria, all criteria must be true for an activity to match.
5. To remove a filter criteria from a compound filter, click the minus icon at the end of the filter row. Removing the last row removes the filter.
6. To include any children of a Pig, Hive, or Oozie activity in your search results, check the **Include Child Activities** checkbox. Otherwise, only the top-level activity will be included, even if one or more child activities matched the filter criteria.
7. Click the **Search** button (which appears when you start creating the filter) to run the filter.

> **Note:** Filters are remembered across user sessions — that is, if you log out the filter will be preserved and will still be active when you log back in. Newly-submitted activities will appear in the Activity List only if they match the filter criteria.

Activity Charts

By default the charts show aggregated statistics about the performance of the cluster: Tasks Running, CPU Usage, and Memory Usage. There are additional charts you can enable from a pop-up panel. You can also superimpose individual job statistics on any of the displayed charts.

Most charts display multiple metrics within the same chart. For example, the **Tasks Running** chart shows two metrics: **Cluster, Running Maps** and **Cluster, Running Reduces** in the same chart. Each metric appears in a different color.

- To see the exact values at a given point in time, move the cursor over the chart – a movable vertical line pinpoints a specific time, and a tooltip shows you the values at that point.
- You can use the time range selector at the top of the page to zoom in – the chart display will follow. In order to zoom out, you can use the Time Range Selector at the top of the page or click the link below the chart.

To select additional charts:

1. Click ![icon] at the top right of the chart panel to open the Customize dialog box.
2. Check or uncheck the boxes next to the charts you want to show or hide.

To show or hide cluster-wide statistics:

- Check or uncheck the **Cluster** checkbox at the top of the Charts panel.

To chart statistics for an individual job:

- Click the chart icon (![icon]) in the row next to the job you want to show on the charts. The job ID will appear in the top bar next to the Cluster checkbox, and the statistics will appear on the appropriate chart.
- To remove a job's statistics from the chart, click the ✖ next to the job ID in the top bar of the chart.

> **Note:** Chart selections are retained only for the current session.

To expand, contract, or hide the charts

- Move the cursor over the divider between the Activities list and the charts, grab it and drag to expand or contract the chart area compared to the Activities list.
- Drag the divider all the way to the right to hide the charts, or all the way to the left to hide the Activities list.

Viewing the Jobs in a Pig, Oozie, or Hive Activity

The Activity **Children** tab shows the same information as does the Activities tab, except that it shows only jobs that are children of a selected Pig, Hive or Oozie activity. In addition, from this tab you can view the details of the Pig, Hive or Oozie activity as a whole, and compare it to similar activities.

1. Click the **Activities** tab.
2. Click the Pig, Hive or Oozie activity you want to inspect. This presents a list of the jobs that make up the Pig, Hive or Oozie activity.

The functions under the **Children** tab are the same as those seen under the **Activities** tab. You can filter the job list, show and hide columns in the job list, show and hide charts and plot job statistics on those charts.

- Click an individual job to view Task information and other information for that child. See Viewing and Filtering MapReduce Activities on page 249 for details of how the functions on this page work.

In addition, viewing a Pig, Hive or Oozie activity provides the following tabs:

- The **Details** tab shows Activity details in a report form. See Viewing Activity Details in a Report Format for more information.
- The **Compare** tab compares this activity to other similar activity. The main difference between this and a comparison for a single MapReduce activity is that the comparison is done looking at other activities of the same type (Pig,

Hive or Oozie) but does include the child jobs of the activity. See [Comparing Similar Activities](#) for an explanation of that tab.

Task Attempts

The Tasks tab contains a list of the Map and Reduce task attempts that make up a job.

Viewing a Job's Task Attempts

1. From the **Clusters** tab, in the section marked **Other**, select the activity you want to inspect.

   • If the activity is a MapReduce job, the **Tasks** tab opens.
   • If the activity is a Pig, Hive, or Oozie activity, select the job you want to inspect from the activity's **Children** tab to open the **Tasks** tab.

The columns shown under the **Tasks** tab display statistics about the performance of and resources used by the task attempts spawned by the selected job. By default only a subset of the possible metrics are displayed — you can modify the columns that are displayed to add or remove the columns in the display.

• The status of an attempt is shown in the Attempt Status column:

▶	The attempt is running.
✓	The attempt has succeeded.
⊗	The attempt has failed.
🔖	The attempt has been unassigned.
⚒	The attempt has been killed.
❓	The attempt's final state is unknown.

• Click the task ID to view details of the individual task.

You can use the **Zoom to Duration** button to zoom the Time Range Selector to the exact time range spanned by the activity whose tasks you are viewing.

Selecting Columns to Show in the Tasks List

In the Tasks list, you can display or hide any of the metrics the Cloudera Manager collects for task attempts. By default a subset of the possible metrics are displayed.

1. Click the **Select Columns to Display** icon (▦). A pop-up panel lets you turn on or off a variety of metrics that may be of interest.
2. Check or uncheck the columns you want to include or remove from the display. As you check or uncheck an item, its column immediately appears or disappears from the display.
3. Click the ✕ in the upper right corner to close the panel.

Sorting the Tasks List

You can sort the tasks list by any of the information displayed in the list:

1. Click the column header to initiate a sort. The small arrow that appears next to the column header indicates the sort direction.
2. Click the column header to reverse the sort direction.

Filtering the Tasks List

You can filter the list of tasks based on values of any of the metrics that are available.

To use one of the predefined filters:

- Click the

  ▼

  to the right of the **Search** button and select the filter you want to run. There are predefined filters to search by job type (for example Pig activities, MapReduce jobs, and so on) or for running, failed, or long-running activities.

To create a filter:

1. Click the

   ▼

   to the right of the **Search** button and select **Custom**.
2. Select a metric from the drop-down list in the first field; you can create a filter based on any of the available metrics.
3. Once you select a metric, fill in the rest of the fields; your choices depend on the type of metric you have selected. Use the percent character `%` as a wildcard in a string; for example, `Id matches job%0001` will look for any MapReduce job ID with suffix `0001`.
4. To create a compound filter, click the plus icon at the end of the filter row to add another row. If you combine filter criteria, all criteria must be true for an activity to match.
5. To remove a filter criteria from a compound filter, click the minus icon at the end of the filter row. Removing the last row removes the filter.
6. To include any children of a Pig, Hive, or Oozie activity in your search results, check the **Include Child Activities** checkbox. Otherwise, only the top-level activity will be included, even if one or more child activities matched the filter criteria.
7. Click the **Search** button (which appears when you start creating the filter) to run the filter.

> 📝 **Note:** The filter persists only for this user session — when you log out, tasks list filter is removed.

Viewing Activity Details in a Report Format

The Details tab for an activity shows the job or activity statistics in a report format.

To view activity details for an individual MapReduce job:

1. Select a MapReduce job from the Clusters tab *or* Select a Pig, Hive or Oozie activity, then select a MapReduce job from the **Children** tab.
2. Select the **Details** tab after the job page is displayed.

This display information about the individual MapReduce job in a report format.

From this page you can also access the **Job Details** and **Job Configuration** pages on the JobTracker web UI.

- Click the **Job Details** link at the top of the report to be taken to the job details web page on the JobTracker host.
- Click the **Job Configuration** link to be taken to the job configuration web page on the JobTracker host.

To view activity details for a Pig, Hive, or Oozie activity:

1. Select a Pig, Hive or Oozie activity.
2. Select the **Details** tab after the list of child jobs is displayed.

This displays information about the Pig, Oozie, or Hive job as a whole.

Note that this the same data you would see for the activity if you displayed all possible columns in the Activities list.

Comparing Similar Activities

It can be useful to compare the performance of similar activities if, for example, you suspect that a job is performing differently than other similar jobs that have run in the past.

The **Compare** tab shows you the performance of the selected job compared with the performance of other similar jobs. Cloudera Manager identifies jobs that are similar to each other (jobs that are basically running the same code – the same Map and Reduce classes, for example).

To compare an activity to other similar activities:

1. Select the job or activity from the Activities list.
2. Click the **Compare** tab.

The activity comparison feature compares performance and resource statistics of the selected job to the mean value of those statistics across a set of the most recent similar jobs. The table provides visual indicators of how the selected job deviates from the mean calculated for the sample set of jobs, as well as providing the actual statistics for the selected job and the set of the similar jobs used to calculate the mean.

- **The first row** in the comparison table displays a set of visual indicators of how the selected job deviates from the mean of all the similar jobs (the combined Average values). This is displayed for each statistic for which a comparison makes sense. The diagram in the ID column shows the elements of the indicator, as follows:

  - The line at the midpoint of the bar represents the mean value of all similar jobs. The colored portion of the bar indicates the degree of deviation of your selected job from the mean. The top and bottom of the bar represent two standard deviations (plus or minus) from the mean.
  - For a given metric, if the value for your selected job is within two standard deviations of the mean, the colored portion of the bar is blue.
  - If a metric for your selected job is more than two standard deviations from the mean, the colored portion of the bar is red.

- **The following rows** show the actual values for other similar jobs. These are the sets of values that were used to calculate the mean values shown in the Combined Averages row. The most recent ten similar jobs are used to calculate the average job statistics, and these are the jobs that are shown in the table.

Viewing the Distribution of Task Attempts

The Task Distribution tab provides a graphical view of the performance of the Map and Reduce tasks that make up a job.

To display the task distribution metrics for a job:

1. Do one of the following:

   - Select a MapReduce job from the **Activities** list.
   - Select a job from the **Children** tab of a Pig, Hive, or Oozie activity.

2. Click the **Task Distribution** tab.

The chart that appears initially shows the distribution of Map Input Records by Duration; you can change the Y-axis to chart a number of different metrics.

You can use the **Zoom to Duration** button to zoom the Time Range Selector to the exact time range spanned by the activity whose tasks you are viewing.

The Task Distribution Chart

The Task Distribution chart shows the distribution of attempts according to their duration on the X-axis and a number of different metrics on the Y-axis. Each cell represents the number of tasks whose performance statistics fall within the parameters of the cell.

The Task Distribution chart is useful for detecting tasks that are outliers in your job, either because of skew, or because of faulty TaskTrackers. The chart can clearly show if some tasks deviate significantly from the majority of task attempts.

Normally, the distribution of tasks will be fairly concentrated. If, for example, some Reducers receive much more data than others, that will be represented by having two discrete sections of density on the graph. That suggests that there may be a problem with the user code, or that there's skew in the underlying data. Alternately, if the input sizes of various Map or Reduce tasks are the same, but the time it takes to process them varies widely, it might mean that certain TaskTrackers are performing more poorly than others.

You can click in a cell and see a list of the TaskTrackers that correspond to the tasks whose performance falls within the cell.

The X-axis show the task duration is seconds. From the drop-down you can chose different metrics for the Y-axis: Input or Output records or bytes for Map tasks, or the number of CPU seconds for the user who ran the job:

- Map Input Records vs. Duration
- Map Output Records vs. Duration
- Map Input Bytes vs. Duration
- Map Output Bytes vs. Duration
- Map Total User CPU seconds vs. Duration
- Reduce Input Records vs. Duration
- Reduce Output Records vs. Duration
- Reduce Total User CPU seconds vs. Duration

TaskTracker Hosts

To the right of the chart is a table that shows the TaskTracker hosts that processed the tasks in the selected cell, along with the number of task attempts each host executed.

You can select a cell in the table to view the TaskTracker hosts that correspond to the tasks in the cell.

- The area above the TaskTracker table shows the type of task and range of data volume (or User CPUs) and duration times for the task attempts that fall within the cell.
- The table itself shows the TaskTracker hosts that executed the tasks that are represented within the cell, and the number of task attempts run on that host.

Clicking a TaskTracker hostname takes you to the Role Status page for that TaskTracker instance.

Monitoring Impala Queries

The Impala Queries page displays information about Impala queries that are running and have run in your cluster. You can filter the queries by time period and by specifying simple filtering expressions.

> **Note:** The Impala query monitoring feature requires Impala 1.0.1 and higher.

Viewing Queries

1. Do one of the following:

   - Select **Clusters** > **Cluster name** > **Impala service name Queries**.
   - On the **Home** > **Status** tab, select **Impala service name** and click the **Queries** tab.

   The Impala queries run during the selected time range display in the Results Tab on page 257.

You can also perform the following actions on this page:

Table 10: Viewing Queries Actions

Action	Description
Filter the displayed queries	Create filter expressions manually, select preconfigured filters, or use the **Workload Summary** section to build a query interactively. See Filtering Queries on page 258.
Select additional attributes for display.	Click **Select Attributes**. Selected attributes also display as available filters in the **Workload Summary** section. To display information about attributes, hover over a field label. See Filter Attributes on page 260.

Action	Description
	Only attributes that support filtering appear in the **Workload Summary** section. See the Table 11: Attributes on page 260 table.
View a histogram of the attribute values.	Click the ⊞ icon to the right of each attribute displayed in the **Workload Summary** section.
Display charts based on the filter expression and selected attributes.	Cick the **Charts** tab.
View charts that help identify whether Impala best practices are being followed.	Click the **Best Practices** link.
Export a JSON file with the query results that you can use for further analysis.	Click **Export**.

Configuring Impala Query Monitoring

You can configure the visibility of the Impala query results and the size of the storage allocated to Impala query results.

For information on how to configure whether admin and non-admin users can view all queries, only that user's queries, or no queries, see Configuring Query Visibility on page 225.

Query information is stored in-memory in a ring buffer. If you restart Service Monitor, all queries are lost, and older queries eventually are dropped. For information on how to configure the query store, see Configuring Impala Query Data Store Maximum Size on page 226.

Impala Best Practices

To open the Impala Best Practices page, click the **Best Practices** tab on the Impala service page. The page contains charts that include description of each best practice and how to determine if it is being followed. See the Impala documentation for more detail on each best practice and for additional best practices.

Adjust the time range to see data on queries run at different times. Click the charts to get more detail on individual queries. Use the filter box at the top right of the Best Practices page to adjust which data is shown on the page. For example, to see just the queries that took more than ten seconds, make the filter `query_duration > 10s`.

Create a trigger based on any best practice by choosing **Create Trigger** from the individual chart drop-down menu.

Results Tab

Queries appear on the **Results** tab, with the most recent at the top. Each query has summary and detail information. A query summary includes the following default attributes: start and end timestamps, statement, duration, rows produced, user, coordinator, database, and query type. For example:

You can add additional attributes to the summary by clicking the Attribute Selector. In each query summary, the query statement is truncated if it is too long to display. To display the entire statement, click ❯. The query entry expands to

display the entire query string. To collapse the query display, click ⌄. To display information about query attributes and possible values, hover over a field in a query. For example:
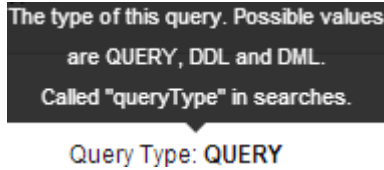
A running job displays a progress bar under the starting timestamp:
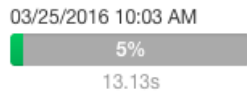
If an error occurred while processing the query, **Error** displays under the complete timestamp.

Use the Actions drop-down menu [≡▾] to the right of each query listing to do the following. (Not all options display, depending on the type of job.)

- Query Details – Opens a details page for the job. See query details.
- User's Impala Queries – Displays a list of queries run by the user for the current job.
- Cancel (running queries only) – Cancel a running query (administrators only). Canceling a running query creates an audit event. When you cancel a query, **⊘ Canceled** replaces the progress bar.
- Queries in the same YARN pool – Displays queries that use the same resource pool.

Filtering Queries

You filter queries by selecting a time range and specifying a filter expression in the search box.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details).

Filter Expressions

Filter expressions specify which entries should display when you run the filter. The simplest expression consists of three components:

- **Attribute** - Query language name of the attribute.
- **Operator** - Type of comparison between the attribute and the attribute value. Cloudera Manager supports the standard comparator operators =, !=, >, <, >=, <=, and RLIKE. (RLIKE performs regular expression matching as specified in the Java Pattern class documentation.) Numeric values can be compared with all operators. String values can be compared with =, !=, and RLIKE. Boolean values can be compared with = and !=.
- **Value** - The value of the attribute. The value depends on the type of the attribute. For a Boolean value, specify either true or false. When specifying a string value, enclose the value in double quotes.

You create compound filter expressions using the AND and OR operators. When more than one operator is used in an expression, AND is evaluated first, then OR. To change the order of evaluation, enclose subexpressions in parentheses.

Compound Expressions

To find all the queries issued by the root user that produced over 100 rows, use the expression:

```
user = "root" AND rowsProduced > 100
```

To find all the executing queries issued by users Jack or Jill, use the expression:

```
executing = true AND (user = "Jack" OR user = "Jill")
```

Choosing and Running a Filter

1. Do one of the following:

   - **Select a Suggested or Recently Run Filter**

     Click the

     to the right of the **Search** button to display a list of sample and recently run filters, and select a filter. The filter text displays in the text box.

   - **Construct a Filter from the Workload Summary Attributes**

     Optionally, click **Select Attributes** to display a dialog box where you can chose which attributes to display in the **Workload Summary** section. Select the checkbox next to one or more attributes, and click **Close**.

     The attributes display in the **Workload Summary** section along with values or ranges of values that you can filter on. The values and ranges display as links with checkboxes. Select one or more checkboxes to add the range or value to the query. Click a link to run a query on that value or range. For example:

     bytes_streamed < 60.0 AND memory_aggregate_peak < 100000.0

     Workload Summary
     (For Completed Queries)

     Aggregate Peak Memory Usage

☑	12 KiB - 97.7 KiB	18
☐	97.7 KiB - 976.6 KiB	8
☐	976.6 KiB - 9.5 MiB	55
☐	9.5 MiB - 95.4 MiB	222
☐	95.4 MiB - 953.7 MiB	62
☐	953.7 MiB - 9.3 GiB	42
☐	9.3 GiB - 34.1 GiB	9

     Bytes Streamed

☑	0 B - 60 B	36
☐	60 B - 600 B	173
☐	600 B - 5.9 KiB	49
☐	5.9 KiB - 58.6 KiB	66
☐	58.6 KiB - 585.9 KiB	96
☐	585.9 KiB - 5.7 MiB	19
☐	5.7 MiB - 5.1 GiB	9

   - **Type a Filter**

     1. Start typing or press **Spacebar** in the text box. As you type, filter attributes matching the typed letter display. If you press **Spacebar**, standard filter attributes display. These suggestions are part of typeahead, which helps build valid queries. For information about the attribute name and supported values for each field, hover over the field in an existing query.
     2. Select an attribute and press **Enter**.
     3. Press **Spacebar** to display a drop-down list of operators.
     4. Select an operator and press **Enter**.
     5. Specify an attribute value in one of the following ways:

        - For attribute values that support typeahead, press **Spacebar** to display a drop-down list of values and press **Enter**.

- Type a value.

2. Click in the text box and press **Enter** or click **Search**. The list displays the results that match the specified filter. The Workload Summary section refreshes to show only the values for the selected filter. The filter is added to the Recently Run list.

Filter Attributes

The following table includes available filter attributes and their names in Cloudera Manager, types, and descriptions.

> **Note:** Only attributes for which the **Supports Filtering?** column value is TRUE appear in the **Workload Summary** section.

Table 11: Attributes

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Admission Result (admission_result)	STRING	TRUE	The result of admission, whether immediately, queued, rejected, or timed out. Called 'admission_result' in searches.
Admission Wait Time (admission_wait)	MILLISECONDS	TRUE	The time from submission for admission to completion of admission. Called 'admission_wait' in searches.
Aggregate Peak Memory Usage (memory_aggregate_peak)	BYTES	TRUE	The highest amount of memory allocated by this query at a particular time across all nodes. Called 'memory_aggregate_peak' in searches.
Bytes Streamed (bytes_streamed)	BYTES	TRUE	The total number of bytes sent between Impala Daemons while processing this query. Called 'bytes_streamed' in searches.
Client Fetch Wait Time (client_fetch_wait_time)	MILLISECONDS	TRUE	The total amount of time the query spent waiting for the client to fetch row data. Called 'client_fetch_wait_time' in searches.
Client Fetch Wait Time Percentage (client_fetch_wait_time_percentage)	NUMBER	TRUE	The total amount of time the query spent waiting for the client to fetch row data divided by the query duration. Called 'client_fetch_wait_time_percentage' in searches.
Connected User (connected_user)	STRING	TRUE	The user who created the Impala session that issued this query. This is distinct from 'user' only if delegation is in use. Called 'connected_user' in searches.
Coordinator (coordinator_host_id)	STRING	TRUE	The host coordinating this query. Called 'coordinator_host_id' in searches.
Database (database)	STRING	TRUE	The database on which the query was run. Called 'database' in searches.
DDL Type (ddl_type)	STRING	TRUE	The type of DDL query. Called 'ddl_type' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Delegated User (delegated_user)	STRING	TRUE	The effective user for the query. This is set only if delegation is in use. Called 'delegated_user' in searches.
Duration (query_duration)	MILLISECONDS	TRUE	The duration of the query in milliseconds. Called 'query_duration' in searches.
Estimated per Node Peak Memory (estimated_per_node_peak_memory)	BYTES	TRUE	The planning process's estimate of per-node peak memory usage for the query. Called 'estimated_per_node_peak_memory' in searches.
Executing (executing)	BOOLEAN	FALSE	Whether the query is currently executing. Called 'executing' in searches.
File Formats (file_formats)	STRING	FALSE	An alphabetically sorted list of all the file formats used in the query. Called 'file_formats' in searches.
HBase Bytes Read (hbase_bytes_read)	BYTES	TRUE	The total number of bytes read from HBase by this query. Called 'hbase_bytes_read' in searches.
HBase Scanner Average Read Throughput (hbase_scanner_average_bytes_read_per_second)	BYTES PER SECOND	TRUE	The average HBase scanner read throughput for this query. This is computed by dividing the total bytes read from HBase by the total time spent reading by all HBase scanners. Called 'hbase_scanner_average_bytes_read_per_second' in searches.
HDFS Average Scan Range (hdfs_average_scan_range)	BYTES	TRUE	The average HDFS scan range size for this query. HDFS scan nodes that contained only a single scan range are not included in this computation. Low numbers for a query might indicate reading many small files which negatively impacts performance. Called 'hdfs_average_scan_range' in searches.
HDFS Bytes Read (hdfs_bytes_read)	BYTES	TRUE	The total number of bytes read from HDFS by this query. Called 'hdfs_bytes_read' in searches.
HDFS Bytes Read From Cache (hdfs_bytes_read_from_cache)	BYTES	TRUE	The total number of bytes read from HDFS that were read from the HDFS cache. This is only for completed queries. Called 'hdfs_bytes_read_from_cache' in searches.
HDFS Bytes Read From Cache Percentage (hdfs_bytes_read_from_cache_percentage)	NUMBER	TRUE	The percentage of all bytes read by this query that were read from the HDFS cache. This is only for completed queries. Called 'hdfs_bytes_read_from_cache_percentage' in searches.
HDFS Bytes Skipped (hdfs_bytes_skipped)	BYTES	TRUE	The total number of bytes that had to be skipped by this query while reading from HDFS. Any number above zero may indicate a problem. Called 'hdfs_bytes_skipped' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
HDFS Bytes Written (hdfs_bytes_written)	BYTES	TRUE	The total number of bytes written to HDFS by this query. Called 'hdfs_bytes_written' in searches.
HDFS Local Bytes Read (hdfs_bytes_read_local)	BYTES	TRUE	The total number of local bytes read from HDFS by this query. This is only for completed queries. Called 'hdfs_bytes_read_local' in searches.
HDFS Local Bytes Read Percentage (hdfs_bytes_read_local_percentage)	NUMBER	TRUE	The percentage of all bytes read from HDFS by this query that were local. This is only for completed queries. Called 'hdfs_bytes_read_local_percentage' in searches.
HDFS Remote Bytes Read (hdfs_bytes_read_remote)	BYTES	TRUE	The total number of remote bytes read from HDFS by this query. This is only for completed queries. Called 'hdfs_bytes_read_remote' in searches.
HDFS Remote Bytes Read Percentage (hdfs_bytes_read_remote_percentage)	NUMBER	TRUE	The percentage of all bytes read from HDFS by this query that were remote. This is only for completed queries. Called 'hdfs_bytes_read_remote_percentage' in searches.
HDFS Scanner Average Read Throughput (hdfs_scanner_average_bytes_read_per_second)	BYTES PER SECOND	TRUE	The average HDFS scanner read throughput for this query. This is computed by dividing the total bytes read from HDFS by the total time spent reading by all HDFS scanners. Called 'hdfs_scanner_average_bytes_read_per_second' in searches.
HDFS Short Circuit Bytes Read (hdfs_bytes_read_short_circuit)	BYTES	TRUE	The total number of bytes read from HDFS by this query that used short-circuit reads. This is only for completed queries. Called 'hdfs_bytes_read_short_circuit' in searches.
HDFS Short Circuit Bytes Read Percentage (hdfs_bytes_read_short_circuit_percentage)	NUMBER	TRUE	The percentage of all bytes read from HDFS by this query that used short-circuit reads. This is only for completed queries. Called 'hdfs_bytes_read_short_circuit_percentage' in searches.
Impala Version (impala_version)	STRING	TRUE	The version of the Impala Daemon coordinating this query. Called 'impala_version' in searches.
Memory Accrual (memory_accrual)	BYTE_SECONDS	TRUE	The total accrued memory usage by the query. This is computed by multiplying the average aggregate memory usage of the query by the query's duration. Called 'memory_accrual' in searches.
Memory Spilled (memory_spilled)	BYTES	TRUE	Amount of memory spilled to disk. Called 'memory_spilled' in searches.
Network Address (network_address)	STRING	TRUE	The network address that issued this query. Called 'network_address' in searches.
Node with Peak Memory Usage	STRING	TRUE	The node with the highest peak memory usage for this query. See Per Node Peak Memory Usage for the

Display Name (Attribute Name)	Type	Supports Filtering?	Description
(memory_per_node_peak_node)			actual peak value. Called 'memory_per_node_peak_node' in searches.
Out of Memory (oom)	BOOLEAN	TRUE	Whether the query ran out of memory. Called 'oom' in searches.
Per Node Peak Memory Usage (memory_per_node_peak)	BYTES	TRUE	The highest amount of memory allocated by any single node that participated in this query. See Node with Peak Memory Usage for the name of the peak node. Called 'memory_per_node_peak' in searches.
Planning Wait Time (planning_wait_time)	MILLISECONDS	TRUE	The total amount of time the query spent waiting for planning to complete. Called 'planning_wait_time' in searches.
Planning Wait Time Percentage (planning_wait_time_percentage)	NUMBER	TRUE	The total amount of time the query spent waiting for planning to complete divided by the query duration. Called 'planning_wait_time_percentage' in searches.
Pool (pool)	STRING	TRUE	The name of the resource pool in which this query executed. Called 'pool' in searches. If YARN is in use, this corresponds to a YARN pool. Within YARN, a pool is referred to as a queue.
Query ID (query_id)	STRING	FALSE	The id of this query. Called 'query_id' in searches.
Query State (query_state)	STRING	TRUE	The current state of the query (running, finished, and so on). Called 'query_state' in searches.
Query Status (query_status)	STRING	TRUE	The status of the query. If the query hasn't failed the status will be 'OK', otherwise it will provide more information on the cause of the failure. Called 'query_status' in searches.
Query Type (query_type)	STRING	TRUE	The type of the query's SQL statement (DML, DDL, Query). Called 'query_type' in searches.
Resource Reservation Wait Time (resources_reserved_wait_time)	MILLISECONDS	TRUE	The total amount of time the query spent waiting for pool resources to become available . Called 'resources_reserved_wait_time' in searches.
Resource Reservation Wait Time Percentage (resources_reserved_wait_time_percentage)	NUMBER	TRUE	The total amount of time the query spent waiting for pool resources to become available divided by the query duration. Called 'resources_reserved_wait_time_percentage' in searches.
Rows Inserted (rows_inserted)	NUMBER	TRUE	The number of rows inserted by the query. Called 'rows_inserted' in searches.
Rows Produced (rows_produced)	NUMBER	TRUE	The number of rows produced by the query. Called 'rows_produced' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Service Name (service_name)	STRING	FALSE	The name of the Impala service. Called 'service_name' in searches.
Session ID (session_id)	STRING	TRUE	The ID of the session that issued this query. Called 'session_id' in searches.
Session Type (session_type)	STRING	TRUE	The type of the session that issued this query. Called 'session_type' in searches.
Statement (statement)	STRING	FALSE	The query's SQL statement. Called 'statement' in searches.
Statistics Missing (stats_missing)	BOOLEAN	TRUE	Whether the query was flagged with missing table or column statistics warning during the planning process. Called 'stats_missing' in searches.
Threads: CPU Time (thread_cpu_time)	MILLISECONDS	TRUE	The sum of the CPU time used by all threads of the query. Called 'thread_cpu_time' in searches.
Threads: CPU Time Percentage (thread_cpu_time_percentage)	NUMBER	TRUE	The sum of the CPU time used by all threads of the query divided by the total thread time. Called 'thread_cpu_time_percentage' in searches.
Threads: Network Receive Wait Time (thread_network_receive_wait_time)	MILLISECONDS	TRUE	The sum of the time spent waiting to receive data over the network by all threads of the query. A query will almost always have some threads waiting to receive data from other nodes in the query's execution tree. Unlike other wait times, network receive wait time does not usually indicate an opportunity for improving a query's performance. Called 'thread_network_receive_wait_time' in searches.
Threads: Network Receive Wait Time Percentage (thread_network_receive_wait_time_percentage)	NUMBER	TRUE	The sum of the time spent waiting to receive data over the network by all threads of the query divided by the total thread time. A query will almost always have some threads waiting to receive data from other nodes in the query's execution tree. Unlike other wait times, network receive wait time does not usually indicate an opportunity for improving a query's performance. Called 'thread_network_receive_wait_time_percentage' in searches.
Threads: Network Send Wait Time (thread_network_send_wait_time)	MILLISECONDS	TRUE	The sum of the time spent waiting to send data over the network by all threads of the query. Called 'thread_network_send_wait_time' in searches.
Threads: Network Send Wait Time Percentage (thread_network_send_wait_time_percentage)	NUMBER	TRUE	The sum of the time spent waiting to send data over the network by all threads of the query divided by the total thread time. Called 'thread_network_send_wait_time_percentage' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Threads: Storage Wait Time (thread_storage_wait_time)	MILLISECONDS	TRUE	The sum of the time spent waiting for storage by all threads of the query. Called 'thread_storage_wait_time' in searches.
Threads: Storage Wait Time Percentage (thread_storage_wait_time_percentage)	NUMBER	TRUE	The sum of the time spent waiting for storage by all threads of the query divided by the total thread time. Called 'thread_storage_wait_time_percentage' in searches.
Threads: Total Time (thread_total_time)	MILLISECONDS	TRUE	The sum of thread CPU, storage wait and network wait times used by all threads of the query. Called 'thread_total_time' in searches.
User (user)	STRING	TRUE	The effective user for the query. This is the delegated user if delegation is in use. Otherwise, this is the connected user. Called 'user' in searches.
Work CPU Time (cm_cpu_milliseconds)	MILLISECONDS	TRUE	Attribute measuring the sum of CPU time used by all threads of the query, in milliseconds. Called 'work_cpu_time' in searches. For Impala queries, CPU time is calculated based on the 'TotalCpuTime' metric. For YARN MapReduce applications, this is calculated from the 'cpu_milliseconds' metric.

Examples

Consider the following filter expressions: `user = "root"`, `rowsProduced > 0`, `fileFormats RLIKE ".TEXT.*"`, and `executing = true`. In the examples:

- The filter attributes are `user`, `rowsProduced`, `fileFormats`, and `executing`.
- The operators are `=`, `>`, and `RLIKE`.
- The filter values are `root`, `0`, `.TEXT.*`, and `true`.

Query Details

The **Query Details** page contains the low-level details of how a SQL query is processed through Impala. The initial information on the page can help you tune the performance of some kinds of queries, primarily those involving joins. The more detailed information on the page is primarily for troubleshooting with the assistance of Cloudera Support; you might be asked to attach the contents of the page to a trouble ticket.

The **Query Details** page displays the following information that is also available in Query Profile:

- Query Plan
- Query Info
- Query Timeline on page 266
- Planner Timeline on page 266
- Query Fragments

To download the contents of the query details, select one of the following:

- **Download Profile...** or **Download Profile...** > **Download Text Profile...** - to download a text version of the query detail.
- **Download Profile...** > **Download Thrift Encoded Profile...** - to download a binary version of the query detail.

Query Plan

The Query Plan section can help you diagnose and tune performance issues with queries. This information is especially useful to understand performance issues with join queries, such as inefficient order of tables in the SQL statement, lack of table and column statistics, and the need for query hints to specify a more efficient join mechanism. You can also learn valuable information about how queries are processed for partitioned tables.

The information in this section corresponds to the output of the `EXPLAIN` statement for the Impala query. Each fragment shown in the query plan corresponds to a processing step that is performed by the central coordinator host or distributed across the hosts in the cluster.

Query Timeline

The Query Timeline section reports statistics about the execution time for phases of the query.

Planner Timeline

The Planner Timeline reports statistics about the execution time for phases of the query planner.

Query Info

The Query Info section reports the attributes of the query, start and end time, duration, and statistics about HDFS access. You can hover over an attribute for information about the attribute name and supported values (for enumerated values). For example:
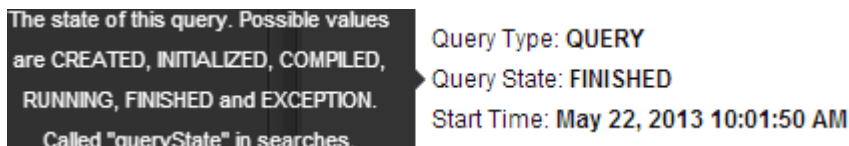
Query Fragments

The Query Fragments section reports detailed low-level statistics for each query plan fragment, involving physical aspects such as CPU utilization, disk I/O, and network traffic. This is the primary information that Cloudera Support might use to help troubleshoot performance issues and diagnose bugs. The details for each fragment display on separate tabs.

Monitoring YARN Applications

The YARN Applications page displays information about the YARN jobs that are running and have run in your cluster. You can filter the jobs by time period and by specifying simple filtering expressions.

Viewing Jobs

1. Do one of the following:

    - Select **Clusters** > *Cluster name* > *YARN service name* **Applications**.
    - On the **Home** > **Status** tab, select *YARN service name* and click the **Applications** tab.

The YARN jobs run during the selected time range display in the Results Tab on page 267. The results displayed can be filtered by creating filter expressions.

You can also perform the following actions on this page:

Table 12: Viewing Jobs Actions

Action	Description
Filter jobs that display.	Create filter expressions manually, select preconfigured filters, or use the **Workload Summary** section to build a query interactively. See [Filtering Jobs](#) on page 268.
Select additional attributes for display.	Click **Select Attributes**. Selected attributes also display as available filters in the **Workload Summary** section. To display information about attributes, hover over a field label. See [Filter Attributes](#) on page 269 Only attributes that support filtering appear in the **Workload Summary** section. See the [Table 13: Attributes](#) on page 270 table.
View a histogram of the attribute values.	Click the ▄ icon to the right of each attribute displayed in the **Workload Summary** section.
Display charts based on the filter expression and selected attributes.	Click the **Charts** tab.
Send a YARN application diagnostic bundle to Cloudera support.	Click **Collect Diagnostics Data**. See [Sending Diagnostic Data to Cloudera for YARN Applications](#) on page 279.
Export a JSON file with the query results that you can use for further analysis.	Click **Export**.

Configuring YARN Application Monitoring

You can configure the visibility of the YARN application monitoring results.

For information on how to configure whether admin and non-admin users can view all applications, only that user's applications, or no applications, see [Configuring Application Visibility](#) on page 225.

Results Tab

Jobs are ordered with the most recent at the top. Each job has summary and detail information. A job summary includes start and end timestamps, query (if the job is part of a Hive query) name, pool, job type, job ID, and user. For example:
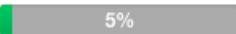
```
03/11/2016 5:30 PM -        insert into traffic_lights_complex...street2(Stage-1)
03/11/2016 5:30 PM          Hive Query String:  ❯ insert into traffic_lights_complex select id, street1, street2, collect_list(named_struct('incident_i...
                            ID:  job_1455752426632_0029          Type:  MAPREDUCE
                            User:   foo                          Pool:     root.foo
                            Duration:  14.53s                    CPU Time:  4.3s
                            File Bytes Read:  144 B              File Bytes Written:  465.6 KiB
                            HDFS Bytes Read:  22.7 KiB           HDFS Bytes Written:  1.7 KiB
                            Memory Allocation:  9.3M
```

A running job displays a progress bar under the start timestamp:

```
03/25/2016 10:03 AM
        5%
     13.13s
```

Use the Actions drop-down menu [▤ ▾] to the right of each job listing to do the following. (Not all options display, depending on the type of job.)

- Application Details – Open a details page for the job.
- Collect Diagnostic Data – Send a YARN application diagnostic bundle to Cloudera support.

- Similar MR2 Jobs – Display a list of similar MapReduce 2 jobs.
- User's YARN Applications – Display a list of all jobs run by the user of the current job.
- View on JobHistory Server – View the application in the YARN JobHistory Server.
- Kill (running jobs only) – Kill a job (administrators only). Killing a job creates an [audit](#) event. When you kill a job, **⊘ Killed** replaces the progress bar.
- Applications in Hive Query (Hive jobs only)
- Applications in Oozie Workflow (Oozie jobs only)
- Applications in Pig Script (Pig jobs only)

Filtering Jobs

You filter jobs by selecting a time range and specifying a filter expression in the search box.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See [Time Line](#) on page 215 for details).

Filter Expressions

Filter expressions specify which entries should display when you run the filter. The simplest expression consists of three components:

- **Attribute** - Query language name of the attribute.
- **Operator** - Type of comparison between the attribute and the attribute value. Cloudera Manager supports the standard comparator operators =, !=, >, <, >=, <=, and RLIKE. (RLIKE performs regular expression matching as specified in the Java [Pattern](#) class documentation.) Numeric values can be compared with all operators. String values can be compared with =, !=, and RLIKE. Boolean values can be compared with = and !=.
- **Value** - The value of the attribute. The value depends on the type of the attribute. For a Boolean value, specify either true or false. When specifying a string value, enclose the value in double quotes.

You create compound filter expressions using the AND and OR operators. When more than one operator is used in an expression, AND is evaluated first, then OR. To change the order of evaluation, enclose subexpressions in parentheses.

Compound Expressions

To find all the jobs issued by the root user that ran for longer than ten seconds, use the expression:

```
user = "root" AND application_duration >= 100000.0
```

To find all the jobs that had more than 200 maps issued by users Jack or Jill, use the expression:

```
maps_completed >= 200.0 AND (user = "Jack" OR user = "Jill")
```

Choosing and Running a Filter

1. Do one of the following:

   - **Select a Suggested or Recently Run Filter**

     Click the

     ▾

     to the right of the **Search** button to display a list of sample and recently run filters, and select a filter. The filter text displays in the text box.

   - **Construct a Filter from the Workload Summary Attributes**

     Optionally, click **Select Attributes** to display a dialog box where you can chose attributes to display in the **Workload Summary** section. Select the checkbox next to one or more attributes and click **Close**. Only attributes that support filtering appear in the **Workload Summary** section. See the [Table 13: Attributes](#) on page 270 table.

The attributes display in the **Workload Summary** section along with values or ranges of values that you can filter on. The values and ranges display as links with checkboxes. Select one or more checkboxes to add the range or value to the query. Click a link to run a query on that value or range. For example:

state = "SUCCEEDED" AND allocated_memory_seconds < 180000.0

Workload Summary
(For Completed Applications)

Allocated Memory Seconds

☑ 120K - 180K   1
☐ 240K - 300K   1
☐ 360K - 420K   1

Allocated VCore Seconds

☐ 120 - 180   1
☐ 240 - 300   1
☐ 360 - 420   1

Application State

☑ SUCCEEDED   2
☐ KILLED   1

CPU Time

- **Type a Filter**

  1. Start typing or press **Spacebar** in the text box. As you type, filter attributes matching the typed letter display. If you press **Spacebar**, standard filter attributes display. These suggestions are part of typeahead, which helps build valid queries. For information about the attribute name and supported values for each field, hover over the field in an existing query.
  2. Select an attribute and press **Enter**.
  3. Press **Spacebar** to display a drop-down list of operators.
  4. Select an operator and press **Enter**.
  5. Specify an attribute value in one of the following ways:

     - For attribute values that support typeahead, press **Spacebar** to display a drop-down list of values and press **Enter**.
     - Type a value.

2. Click in the text box and press **Enter** or click **Search**. The list displays the results that match the specified filter. If the histograms are showing, they are redrawn to show only the values for the selected filter. The filter is added to the Recently Run list.

Filter Attributes

Filter attributes, their names as they are displayed in Cloudera Manager, their types, and descriptions, are enumerated below.

> **Note:** Only attributes where the **Supports Filtering?** column value is TRUE appear in the **Workload Summary** section.

Table 13: Attributes

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Allocated Memory (allocated_mb)	NUMBER	FALSE	The sum of memory in MB allocated to the application's running containers. Called 'allocated_mb' in searches.
Allocated Memory Seconds (allocated_memory_seconds)	NUMBER	TRUE	The amount of memory the application has allocated (megabyte-seconds). Called 'allocated_memory_seconds' in searches.
Allocated VCores (allocated_vcores)	NUMBER	FALSE	The sum of virtual cores allocated to the application's running containers. Called 'allocated_vcores' in searches.
Allocated VCore Seconds (allocated_vcore_seconds)	NUMBER	TRUE	The amount of CPU resources the application has allocated (virtual core-seconds). Called 'allocated_vcore_seconds' in searches.
Application ID (application_id)	STRING	FALSE	The ID of the YARN application. Called 'application_id' in searches.
Application State (state)	STRING	TRUE	The state of this YARN application. This reflects the ResourceManager state while the application is running and the JobHistory Server state after the application has completed. Called 'state' in searches.
Application Tags (application_tags)	STRING	FALSE	A list of tags for the application. Called 'application_tags' in searches.
Application Type (application_type)	STRING	TRUE	The type of the YARN application. Called 'application_type' in searches.
Bytes Read (bytes_read)	BYTES	TRUE	Bytes read. Called 'bytes_read' in searches.
Bytes Written (bytes_written)	BYTES	TRUE	Bytes written. Called 'bytes_written' in searches.
Combine Input Records (combine_input_records)	NUMBER	TRUE	Combine input records. Called 'combine_input_records' in searches.
Combine Output Records (combine_output_records)	NUMBER	TRUE	Combine output records. Called 'combine_output_records' in searches.
Committed Heap (committed_heap_bytes)	BYTES	TRUE	Total committed heap usage. Called 'committed_heap_bytes' in searches.
Completed Maps and Reduces (tasks_completed)	NUMBER	TRUE	The number of completed map and reduce tasks in this MapReduce job. Called 'tasks_completed' in searches. Available only for running jobs.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
CPU Allocation (`vcores_millis`)	NUMBER	TRUE	CPU allocation. This is the sum of 'vcores_millis_maps' and 'vcores_millis_reduces'. Called 'vcores_millis' in searches.
CPU Time (`cpu_milliseconds`)	MILLISECONDS	TRUE	CPU time. Called 'cpu_milliseconds' in searches.
Data Local Maps (`data_local_maps`)	NUMBER	TRUE	Data local maps. Called 'data_local_maps' in searches.
Data Local Maps Percentage (`data_local_maps_percentage`)	NUMBER	TRUE	The number of data local maps as a percentage of the total number of maps. Called 'data_local_maps_percentage' in searches.
Diagnostics (`diagnostics`)	STRING	FALSE	Diagnostic information on the YARN application. If the diagnostic information is long, this may only contain the beginning of the information. Called 'diagnostics' in searches.
Duration (`application_duration`)	MILLISECONDS	TRUE	How long YARN took to run this application. Called 'application_duration' in searches.
Executing (`executing`)	BOOLEAN	FALSE	Whether the YARN application is currently running. Called 'executing' in searches.
Failed Map and Reduce Attempts (`failed_tasks_attempts`)	NUMBER	TRUE	The number of failed map and reduce attempts for this MapReduce job. Called 'failed_tasks_attempts' in searches. Available only for failed jobs.
Failed Map Attempts (`failed_map_attempts`)	NUMBER	TRUE	The number of failed map attempts for this MapReduce job. Called 'failed_map_attempts' in searches. Available only for running jobs.
Failed Maps (`num_failed_maps`)	NUMBER	TRUE	Failed maps. Called 'num_failed_maps' in searches.
Failed Reduce Attempts (`failed_reduce_attempts`)	NUMBER	TRUE	The number of failed reduce attempts for this MapReduce job. Called 'failed_reduce_attempts' in searches. Available only for running jobs.
Failed Reduces (`num_failed_reduces`)	NUMBER	TRUE	Failed reduces. Called 'num_failed_reduces' in searches.
Failed Shuffles (`failed_shuffle`)	NUMBER	TRUE	Failed shuffles. Called 'failed_shuffle' in searches.
Failed Tasks (`num_failed_tasks`)	NUMBER	TRUE	The total number of failed tasks. This is the sum of 'num_failed_maps' and 'num_failed_reduces'. Called 'num_failed_tasks' in searches.
Fallow Map Slots Time (`fallow_slots_millis_maps`)	MILLISECONDS	TRUE	Fallow map slots time. Called 'fallow_slots_millis_maps' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Fallow Reduce Slots Time (`fallow_slots_millis_reduces`)	MILLISECONDS	TRUE	Fallow reduce slots time. Called 'fallow_slots_millis_reduces' in searches.
Fallow Slots Time (`fallow_slots_millis`)	MILLISECONDS	TRUE	Total fallow slots time. This is the sum of 'fallow_slots_millis_maps' and 'fallow_slots_millis_reduces'. Called 'fallow_slots_millis' in searches.
File Bytes Read (`file_bytes_read`)	BYTES	TRUE	File bytes read. Called 'file_bytes_read' in searches.
File Bytes Written (`file_bytes_written`)	BYTES	TRUE	File bytes written. Called 'file_bytes_written' in searches.
File Large Read Operations (`file_large_read_ops`)	NUMBER	TRUE	File large read operations. Called 'file_large_read_ops' in searches.
File Read Operations (`file_read_ops`)	NUMBER	TRUE	File read operations. Called 'file_read_ops' in searches.
File Write Operations (`file_write_ops`)	NUMBER	TRUE	File write operations. Called 'file_large_write_ops' in searches.
Garbage Collection Time (`gc_time_millis`)	MILLISECONDS	TRUE	Garbage collection time. Called 'gc_time_millis' in searches.
HDFS Bytes Read (`hdfs_bytes_read`)	BYTES	TRUE	HDFS bytes read. Called 'hdfs_bytes_read' in searches.
HDFS Bytes Written (`hdfs_bytes_written`)	BYTES	TRUE	HDFS bytes written. Called 'hdfs_bytes_written' in searches.
HDFS Large Read Operations (`hdfs_large_read_ops`)	NUMBER	TRUE	HDFS large read operations. Called 'hdfs_large_read_ops' in searches.
HDFS Read Operations (`hdfs_read_ops`)	NUMBER	TRUE	HDFS read operations. Called 'hdfs_read_ops' in searches.
HDFS Write Operations (`hdfs_write_ops`)	NUMBER	TRUE	HDFS write operations. Called 'hdfs_write_ops' in searches.
Hive Query ID (`hive_query_id`)	STRING	FALSE	If this MapReduce job ran as a part of a Hive query, this field contains the ID of the Hive query. Called 'hive_query_id' in searches.
Hive Query String (`hive_query_string`)	STRING	TRUE	If this MapReduce job ran as a part of a Hive query, this field contains the string of the query. Called 'hive_query_string' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Hive Sentry Subject Name `(hive_sentry_subject_name)`	STRING	TRUE	If this MapReduce job ran as a part of a Hive query on a secured cluster using impersonation, this field contains the name of the user that initiated the query. Called 'hive_sentry_subject_name' in searches.
Input Directory `(input_dir)`	STRING	TRUE	The input directory for this MapReduce job. Called 'input_dir' in searches.
Input Split Bytes `(split_raw_bytes)`	BYTES	TRUE	Input split bytes. Called 'split_raw_bytes' in searches.
Killed Map and Reduce Attempts `(killed_tasks_attempts)`	NUMBER	TRUE	The number of map and reduce attempts that were killed by user(s) for this MapReduce job. Called 'killed_tasks_attempts' in searches. Available only for killed jobs.
Killed Map Attempts `(killed_map_attempts)`	NUMBER	TRUE	The number of map attempts killed by user(s) for this MapReduce job. Called 'killed_map_attempts' in searches. Available only for running jobs.
Killed Reduce Attempts `(killed_reduce_attempts)`	NUMBER	TRUE	The number of reduce attempts killed by user(s) for this MapReduce job. Called 'killed_reduce_attempts' in searches. Available only for running jobs.
Launched Map Tasks `(total_launched_maps)`	NUMBER	TRUE	Launched map tasks. Called 'total_launched_maps' in searches.
Launched Reduce Tasks `(total_launched_reduces)`	NUMBER	TRUE	Launched reduce tasks. Called 'total_launched_reduces' in searches.
Map and Reduce Attempts in NEW State `(new_tasks_attempts)`	NUMBER	TRUE	The number of map and reduce attempts in NEW state for this MapReduce job. Called 'new_tasks_attempts' in searches. Available only for running jobs.
Map Attempts in NEW State `(new_map_attempts)`	NUMBER	TRUE	The number of map attempts in NEW state for this MapReduce job. Called 'new_map_attempts' in searches. Available only for running jobs.
Map Class `(mapper_class)`	STRING	TRUE	The class used by the map tasks in this MapReduce job. Called 'mapper_class' in searches. You can search for the mapper class using the class name alone, for example 'QuasiMonteCarlo$QmcMapper', or the fully qualified classname, for example, 'org.apache.hadoop.examples.QuasiMonteCarlo$QmcMapper'.
Map CPU Allocation `(vcores_millis_maps)`	NUMBER	TRUE	Map CPU allocation. Called 'vcores_millis_maps' in searches.
Map Input Records `(map_input_records)`	NUMBER	TRUE	Map input records. Called 'map_input_records' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Map Memory Allocation (`mb_millis_maps`)	NUMBER	TRUE	Map memory allocation. Called 'mb_millis_maps' in searches.
Map Output Bytes (`map_output_bytes`)	BYTES	TRUE	Map output bytes. Called 'map_output_bytes' in searches.
Map Output Materialized Bytes (`map_output_materialized_bytes`)	BYTES	TRUE	Map output materialized bytes. Called 'map_output_materialized_bytes' in searches.
Map Output Records (`map_output_records`)	NUMBER	TRUE	Map output records. Called 'map_output_records' in searches.
Map Progress (`map_progress`)	NUMBER	TRUE	The percentage of maps completed for this MapReduce job. Called 'map_progress' in searches. Available only for running jobs.
Maps Completed (`maps_completed`)	NUMBER	TRUE	The number of map tasks completed as a part of this MapReduce job. Called 'maps_completed' in searches.
Map Slots Time (`slots_millis_maps`)	MILLISECONDS	TRUE	Total time spent by all maps in occupied slots. Called 'slots_millis_maps' in searches.
Maps Pending (`maps_pending`)	NUMBER	TRUE	The number of maps waiting to be run for this MapReduce job. Called 'maps_pending' in searches. Available only for running jobs.
Maps Running (`maps_running`)	NUMBER	TRUE	The number of maps currently running for this MapReduce job. Called 'maps_running' in searches. Available only for running jobs.
Maps Total (`maps_total`)	NUMBER	TRUE	The number of Map tasks in this MapReduce job. Called 'maps_total' in searches.
Memory Allocation (`mb_millis`)	NUMBER	TRUE	Total memory allocation. This is the sum of 'mb_millis_maps' and 'mb_millis_reduces'. Called 'mb_millis' in searches.
Merged Map Outputs (`merged_map_outputs`)	NUMBER	TRUE	Merged map outputs. Called 'merged_map_outputs' in searches.
Name (`name`)	STRING	TRUE	Name of the YARN application. Called 'name' in searches.
Oozie Workflow ID (`oozie_id`)	STRING	FALSE	If this MapReduce job ran as a part of an Oozie workflow, this field contains the ID of the Oozie workflow. Called 'oozie_id' in searches.
Other Local Maps (`other_local_maps`)	NUMBER	TRUE	Other local maps. Called 'other_local_maps' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Other Local Maps Percentage (`other_local_maps_percentage`)	NUMBER	TRUE	The number of other local maps as a percentage of the total number of maps. Called 'other_local_maps_percentage' in searches.
Output Directory (`output_dir`)	STRING	TRUE	The output directory for this MapReduce job. Called 'output_dir' in searches.
Pending Maps and Reduces (`tasks_pending`)	NUMBER	TRUE	The number of maps and reduces waiting to be run for this MapReduce job. Called 'tasks_pending' in searches. Available only for running jobs.
Physical Memory (`physical_memory_bytes`)	BYTES	TRUE	Physical memory. Called 'physical_memory_bytes' in searches.
Pig Script ID (`pig_id`)	STRING	FALSE	If this MapReduce job ran as a part of a Pig script, this field contains the ID of the Pig script. Called 'pig_id' in searches.
Pool (`pool`)	STRING	TRUE	The name of the resource pool in which this application ran. Called 'pool' in searches. Within YARN, a pool is referred to as a queue.
Progress (`progress`)	NUMBER	TRUE	The progress reported by the application. Called 'progress' in searches.
Rack Local Maps (`rack_local_maps`)	NUMBER	TRUE	Rack local maps. Called 'rack_local_maps' in searches.
Rack Local Maps Percentage (`rack_local_maps_percentage`)	NUMBER	TRUE	The number of rack local maps as a percentage of the total number of maps. Called 'rack_local_maps_percentage' in searches.
Reduce Attempts in NEW State (`new_reduce_attempts`)	NUMBER	TRUE	The number of reduce attempts in NEW state for this MapReduce job. Called 'new_reduce_attempts' in searches. Available only for running jobs.
Reduce Class (`reducer_class`)	STRING	TRUE	The class used by the reduce tasks in this MapReduce job. Called 'reducer_class' in searches. You can search for the reducer class using the class name alone, for example 'QuasiMonteCarlo$QmcReducer', or fully qualified classname, for example, 'org.apache. hadoop.examples.QuasiMonteCarlo$QmcReducer'.
Reduce CPU Allocation (`vcores_millis_reduces`)	NUMBER	TRUE	Reduce CPU allocation. Called 'vcores_millis_reduces' in searches.
Reduce Input Groups (`reduce_input_groups`)	NUMBER	TRUE	Reduce input groups. Called 'reduce_input_groups' in searches.
Reduce Input Records (`reduce_input_records`)	NUMBER	TRUE	Reduce input records. Called 'reduce_input_records' in searches.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Reduce Memory Allocation (mb_millis_reduces)	NUMBER	TRUE	Reduce memory allocation. Called 'mb_millis_reduces' in searches.
Reduce Output Records (reduce_output_records)	NUMBER	TRUE	Reduce output records. Called 'reduce_output_records' in searches.
Reduce Progress (reduce_progress)	NUMBER	TRUE	The percentage of reduces completed for this MapReduce job. Called 'reduce_progress' in searches. Available only for running jobs.
Reduces Completed (reduces_completed)	NUMBER	TRUE	The number of reduce tasks completed as a part of this MapReduce job. Called 'reduces_completed' in searches.
Reduce Shuffle Bytes (reduce_shuffle_bytes)	BYTES	TRUE	Reduce shuffle bytes. Called 'reduce_shuffle_bytes' in searches.
Reduce Slots Time (slots_millis_reduces)	MILLISECONDS	TRUE	Total time spent by all reduces in occupied slots. Called 'slots_millis_reduces' in searches.
Reduces Pending (reduces_pending)	NUMBER	TRUE	The number of reduces waiting to be run for this MapReduce job. Called 'reduces_pending' in searches. Available only for running jobs.
Reduces Running (reduces_running)	NUMBER	TRUE	The number of reduces currently running for this MapReduce job. Called 'reduces_running' in searches. Available only for running jobs.
Reduces Total (reduces_total)	NUMBER	TRUE	The number of reduce tasks in this MapReduce job. Called 'reduces_total' in searches.
Running Containers (running_containers)	NUMBER	FALSE	The number of containers currently running for the application. Called 'running_containers' in searches.
Running Map and Reduce Attempts (running_tasks_attempts)	NUMBER	TRUE	The number of map and reduce attempts currently running for this MapReduce job. Called 'running_tasks_attempts' in searches. Available only for running jobs.
Running Map Attempts (running_map_attempts)	NUMBER	TRUE	The number of running map attempts for this MapReduce job. Called 'running_map_attempts' in searches. Available only for running jobs.
Running MapReduce Application Information Retrieval Duration. (running_application_info_retrieval_time)	NUMBER	TRUE	How long it took, in seconds, to retrieve information about the MapReduce application.
Running Maps and Reduces (tasks_running)	NUMBER	TRUE	The number of maps and reduces currently running for this MapReduce job. Called 'tasks_running' in searches. Available only for running jobs.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Running Reduce Attempts (running_reduce_attempts)	NUMBER	TRUE	The number of running reduce attempts for this MapReduce job. Called 'running_reduce_attempts' in searches. Available only for running jobs.
Service Name (service_name)	STRING	FALSE	The name of the YARN service. Called 'service_name' in searches.
Shuffle Bad ID Errors (shuffle_errors_bad_id)	NUMBER	TRUE	Shuffle bad ID errors. Called 'shuffle_errors_bad_id' in searches.
Shuffle Connection Errors (shuffle_errors_connection)	NUMBER	TRUE	Shuffle connection errors. Called 'shuffle_errors_connection' in searches.
Shuffled Maps (shuffled_maps)	NUMBER	TRUE	Shuffled maps. Called 'shuffled_maps' in searches.
Shuffle IO Errors (shuffle_errors_io)	NUMBER	TRUE	Shuffle IO errors. Called 'shuffle_errors_io' in searches.
Shuffle Wrong Length Errors (shuffle_errors_wrong_length)	NUMBER	TRUE	Shuffle wrong length errors. Called 'shuffle_errors_wrong_length' in searches.
Shuffle Wrong Map Errors (shuffle_errors_wrong_map)	NUMBER	TRUE	Shuffle wrong map errors. Called 'shuffle_errors_wrong_map' in searches.
Shuffle Wrong Reduce Errors (shuffle_errors_wrong_reduce)	NUMBER	TRUE	Shuffle wrong reduce errors. Called 'shuffle_errors_wrong_reduce' in searches.
Slots Time (slots_millis)	MILLISECONDS	TRUE	Total slots time. This is the sum of 'slots_millis_maps' and 'slots_millis_reduces'. Called 'slots_millis' in searches.
Spilled Records (spilled_records)	NUMBER	TRUE	Spilled Records. Called 'spilled_records' in searches.
Successful Map and Reduce Attempts (successful_tasks_attempts)	NUMBER	TRUE	The number of successful map and reduce attempts for this MapReduce job. Called 'successful_tasks_attempts' in searches. Available only for successful jobs.
Successful Map Attempts (successful_map_attempts)	NUMBER	TRUE	The number of successful map attempts for this MapReduce job. Called 'successful_map_attempts' in searches. Available only for running jobs.
Successful Reduce Attempts (successful_reduce_attempts)	NUMBER	TRUE	The number of successful reduce attempts for this MapReduce job. Called 'successful_reduce_attempts' in searches. Available only for running jobs.
Total Maps and Reduces Number (total_task_num)	NUMBER	TRUE	The number of map and reduce tasks in this MapReduce job. Called 'tasks_total' in searches. Available only for running jobs.

Display Name (Attribute Name)	Type	Supports Filtering?	Description
Total Tasks (total_launched_tasks)	NUMBER	TRUE	The total number of tasks. This is the sum of 'total_launched_maps' and 'total_launched_reduces'. Called 'total_launched_tasks' in searches.
Tracking Url (tracking_url)	STRING	FALSE	The MapReduce application tracking URL.
Uberized Job (uberized)	BOOLEAN	FALSE	Whether this MapReduce job is uberized - running completely in the ApplicationMaster. Called 'uberized' in searches. Available only for running jobs.
Unused Memory Seconds (unused_memory_seconds)	NUMBER	TRUE	The amount of memory the application has allocated but not used (megabyte-seconds). This metric is calculated hourly if container usage metric aggregation is enabled. Called 'unused_memory_seconds' in searches.
Unused VCore Seconds (unused_vcore_seconds)	NUMBER	TRUE	The amount of CPU resources the application has allocated but not used (virtual core-seconds). This metric is calculated hourly if container usage metric aggregation is enabled. Called 'unused_vcore_seconds' in searches.
Used Memory Max (used_memory_max)	NUMBER	TRUE	The maximum container memory usage for a YARN application. This metric is calculated hourly if container usage metric aggregation is enabled and a Cloudera Manager Container Usage Metrics Directory is specified. For information about how to enable metric aggregation and the Container Usage Metrics Directory, see Enabling the Cluster Utilization Report on page 402.
User (user)	STRING	TRUE	The user who ran the YARN application. Called 'user' in searches.
Virtual Memory (virtual_memory_bytes)	BYTES	TRUE	Virtual memory. Called 'virtual_memory_bytes' in searches.
Work CPU Time (cm_cpu_milliseconds)	MILLISECONDS	TRUE	Attribute measuring the sum of CPU time used by all threads of the query, in milliseconds. Called 'work_cpu_time' in searches. For Impala queries, CPU time is calculated based on the 'TotalCpuTime' metric. For YARN MapReduce applications, this is calculated from the 'cpu_milliseconds' metric.

Examples

Consider the following filter expressions: user = "root", rowsProduced > 0, fileFormats RLIKE ".TEXT.*", and executing = true. In the examples:

- The filter attributes are user, rowsProduced, fileFormats, and executing.
- The operators are =, >, and RLIKE.

- The filter values are `root`, `0`, `.TEXT.*`, and `true`.

Sending Diagnostic Data to Cloudera for YARN Applications

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

You can send diagnostic data collected from YARN applications, including metadata, configurations, and log data, to Cloudera Support for analysis. Include a support ticket number if one exists to enable Cloudera Support to address the issue more quickly and efficiently. To send YARN application diagnostic data, perform the following steps:

1. From the YARN page in Cloudera Manager, click the **Applications** menu.
2. Collect diagnostics data. There are two ways to do this:

   - To collect data from all applications that are visible in the list, click the top **Collect Diagnostics Data** button on the upper right, above the list of YARN applications.
   - To collect data from only one specific application, click the down arrow on the right-hand end of the row of the application and select **Collect Diagnostics Data**.

3. In the **Send YARN Applications Diagnostic Data** dialog box, provide the following information:

   - If applicable, the Cloudera Support ticket number of the issue being experienced on the cluster.
   - Optionally, add a comment to help the support team understand the issue.

4. Click the checkbox **Send Diagnostic Data to Cloudera**.
5. Click the button **Collect and Send Diagnostic Data**.

> **Note:** Passwords from configuration will not be retrieved.

Monitoring Spark Applications

To obtain information about Spark application behavior you can consult cluster manager logs and the Spark web application UI. These two methods provide complementary information. Logs enable you to see fine grained events in the lifecycle of an application. The web UI provides both a broad overview of the various aspects of Spark application behavior and fine grained metrics. This section provides an overview of both methods.

For further information on Spark monitoring, see Monitoring and Instrumentation.

Viewing and Debugging Spark Applications Using Logs

To see overview information about all running Spark applications:

- Go to the YARN applications page in the Cloudera Manager Admin Console.

  To debug Spark applications running on YARN, view the logs for the NodeManager role:

  1. Open the log event viewer.
  2. Filter the event stream to choose a time window, log level, and display the NodeManager source.

3. For any event, click **View Log File** to view the entire log file.

Visualizing Spark Applications Using the Web Application UI

Every Spark application launches a web application UI that displays useful information about the application:

- An event timeline that displays the relative ordering and interleaving of application events. The timeline view is available on three levels: across all jobs, within one job, and within one stage. The timeline also shows executor allocation and deallocation.
- A list of stages and tasks.
- The execution directed acyclic graph (DAG) for each job.
- A summary of RDD sizes and memory usage.
- Environment - runtime information, property settings, library paths.
- Information about Spark SQL jobs.

The web UI is available in different ways depending on whether the application is running or has completed.

Accessing the Web UI of a Running Spark Application

To access the web application UI of a running Spark application, open `http://spark_driver_host:4040` in a web browser. If multiple applications are running on the same host, the web application binds to successive ports beginning with 4040 (4041, 4042, and so on). The web application is available only for the *duration of the application*.

Accessing the Web UI of a Completed Spark Application

To access the web application UI of a completed Spark application, do the following:

1. Open the Spark History Server UI in one of the following ways:

   - Open the URL `http://spark_history_server_host:18088`.
   - Open the UI in the Cloudera Manager Admin Console:

     1. Go to the Spark service.
     2. Click the **History Server Web UI** link.

   The History Server displays a list of completed applications.
2. In the list of applications, click an **App ID** link. The application UI displays.

> **Note:** In CDH 5.10 and higher, and in CDK 2.x Powered By Apache Spark, the **Storage** tab of the Spark History Server is always blank. To see storage information while an application is running, use the web UI of the application as described in the previous section. After the application is finished, storage information is not available.

Example Spark Application Web Application

Consider a job consisting of a set of transformation to join data from an accounts dataset with a weblogs dataset in order to determine the total number of web hits for every account and then an action write the result to HDFS. In this example, the write is performed twice, resulting in two jobs. To view the application UI, in the History Server click the link in the App ID column:

Spark 1.6.0 **History Server**
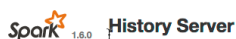
Event log directory: hdfs://vc0136.halxg.cloudera.com:8020/user/spark/applicationHistory

Showing 1-20 of 148

App ID	App Name	Started	Completed	Duration	Spark User	Last Updated
application_1463513516522_0731	PySparkShell	2016/06/15 18:41:54	2016/06/15 18:45:32	3.6 min	sparktest	2016/06/15 18:45:32

The following screenshot shows the timeline of the events in the application including the jobs that were run and the allocation and deallocation of executors. Each job shows the last action, `saveAsTextFile`, run for the job. The timeline shows that the application acquires executors over the course of running the first job. After the second job finishes, the executors become idle and are returned to the cluster.

You can manipulate the timeline as follows:

- Pan - Press and hold the left mouse button and swipe left and right.
- Zoom - Select the **Enable zooming** checkbox and scroll the mouse up and down.

To view the details for Job 0, click the link in the Description column. The following screenshot shows details of each stage in Job 0 and the DAG visualization. Zooming in shows finer detail for the segment from 28 to 42 seconds:

Clicking a stage shows further details and metrics:

Details for Stage 1 (Attempt 0)

Total Time Across All Tasks: 5 s
Locality Level Summary: Node local: 9; Process local: 6
Input Size / Records: 191.8 KB / 97324
Shuffle Read: 78.6 KB / 216
Shuffle Write: 2.8 MB / 1023

▸ DAG Visualization
▸ Show Additional Metrics
▾ Event Timeline
☐ Enable zooming
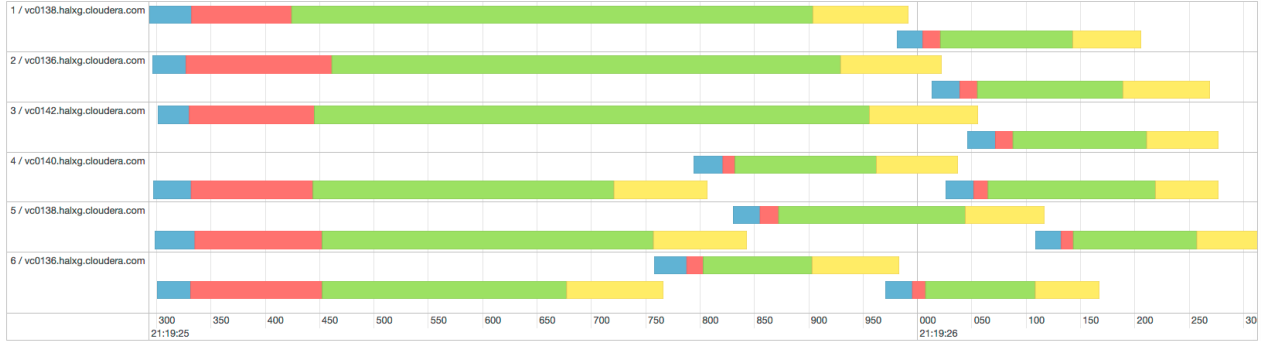
Summary Metrics for 15 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	0.2 s	0.2 s	0.2 s	0.4 s	0.6 s
GC Time	0 ms	0 ms	0 ms	0 ms	0 ms
Input Size / Records	0.0 B / 0	0.0 B / 0	0.0 B / 1	0.0 B / 8187	63.9 KB / 24336
Shuffle Read Size / Records	0.0 B / 0	0.0 B / 0	0.0 B / 0	13.0 KB / 36	13.7 KB / 36
Shuffle Write Size / Records	299.0 B / 1	12.8 KB / 20	13.4 KB / 20	223.8 KB / 135	723.9 KB / 165

Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Succeeded Tasks	Input Size / Records	Shuffle Read Size / Records	Shuffle Write Size / Records
1	vc0138.halxg.cloudera.com:37289	0.9 s	2	0	2	0.0 B / 24253	13.0 KB / 36	734.7 KB / 185
2	vc0136.halxg.cloudera.com:43238	1.0 s	2	0	2	0.0 B / 24336	13.0 KB / 36	737.0 KB / 185
3	vc0142.halxg.cloudera.com:47133	1.0 s	2	0	2	0.0 B / 24266	13.7 KB / 36	737.7 KB / 185
4	vc0140.halxg.cloudera.com:59624	1 s	3	0	3	127.9 KB / 16362	13.3 KB / 36	456.9 KB / 290
5	vc0138.halxg.cloudera.com:33036	1 s	3	0	3	63.9 KB / 8104	25.6 KB / 72	245.7 KB / 175
6	vc0136.halxg.cloudera.com:56747	0.9 s	3	0	3	0.0 B / 3	0.0 B / 0	900.0 B / 3

The web page for Job 1 shows how preceding stages are skipped because Spark retains the results from those stages:

Details for Job 1

Status: SUCCEEDED
Completed Stages: 1
Skipped Stages: 2

▸ Event Timeline
▾ DAG Visualization
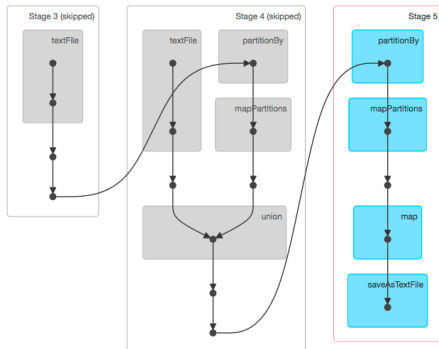
Completed Stages (1)

Stage Id	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
5	saveAsTextFile at NativeMethodAccessorImpl.java:-2	+details	2016/06/15 18:43:49	0.8 s	12/12		55.3 KB	3.1 MB	

Skipped Stages (2)

Stage Id	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	join at <stdin>:1	+details	Unknown	Unknown	0/12				
3	reduceByKey at <stdin>:6	+details	Unknown	Unknown	0/6				

Example Spark SQL Web Application

In addition to the screens described above, the web application UI of an application that uses the Spark SQL API also has an SQL tab. Consider an application that loads the contents of two tables into a pair of DataFrames, joins the tables, and then shows the result. After you click the application ID, the SQL tab displays the final action in the query:

If you click the `show` link you see the DAG of the job. Clicking the **Details** link on this page displays the logical query plan:

Example Spark Streaming Web Application

> **Note:** The following example demonstrates the Spark driver web UI. Streaming information is not captured in the Spark History Server.

The Spark driver web application UI also supports displaying the behavior of streaming applications in the **Streaming** tab. If you run the example described in Spark Streaming Example, and provide three bursts of data, the top of the tab displays a series of visualizations of the statistics summarizing the overall behavior of the streaming application:
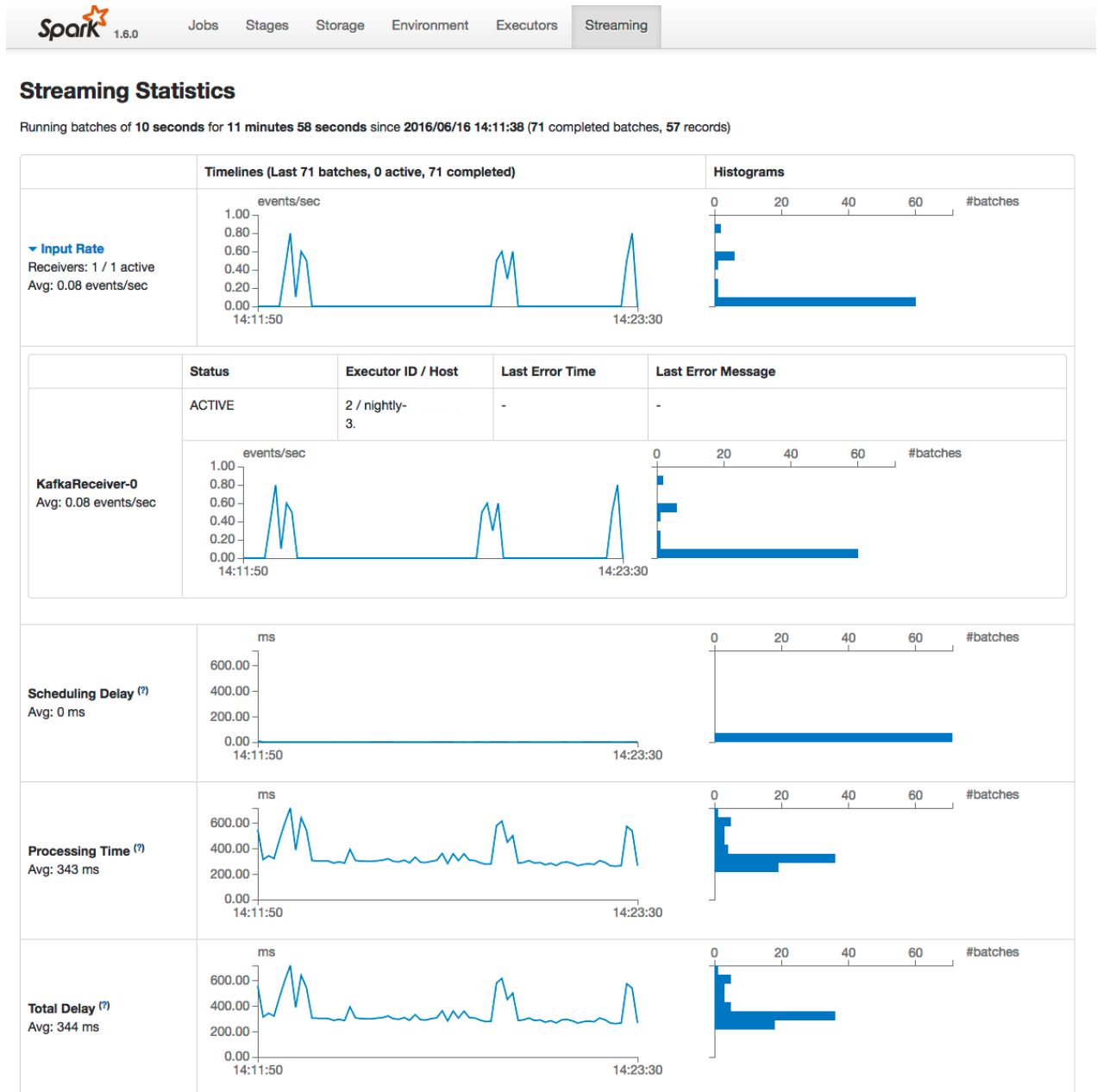
The application has one receiver that processed 3 bursts of event batches, which can be observed in the events, processing time, and delay graphs. Further down the page you can view details of individual batches:

Active Batches (0)

Batch Time	Input Size	Scheduling Delay (?)	Processing Time (?)	Output Ops: Succeeded/Total	Status

Completed Batches (last 71 out of 71)

Batch Time	Input Size	Scheduling Delay (?)	Processing Time (?)	Total Delay (?)	Output Ops: Succeeded/Total
2016/06/16 14:23:30	0 events	1 ms	0.3 s	0.3 s	1/1
2016/06/16 14:23:20	8 events	1 ms	0.5 s	0.5 s	1/1
2016/06/16 14:23:10	5 events	1 ms	0.6 s	0.6 s	1/1
2016/06/16 14:23:00	0 events	0 ms	0.3 s	0.3 s	1/1

To view the details of a specific batch, click a link in the **Batch Time** column. Clicking the **2016/06/16 14:23:20** link with 8 events in the batch, provides the following details:

Details of batch at 2016/06/16 14:23:20

Batch Duration: 10 s
Input data size: 8 records
Scheduling delay: 1 ms
Processing time: 0.5 s
Total delay: 0.5 s

Output Op Id	Description		Output Op Duration	Status	Job Id	Job Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total	Error
0	callForeachRDD at NativeMethodAccessorImpl.java:-2	+details	0.5 s	Succeeded	-	-	-	-	-

Events

An *event* is a record that something of interest has occurred – a service's health has changed state, a log message (of the appropriate severity) has been logged, and so on. Many events are enabled and configured by default.

From the Events page you can filter for events for services or role instances, hosts, users, commands, and much more. You can also search against the content information returned by the event.

The Event Server aggregates relevant events and makes them available for alerting and for searching. This way, you have a view into the history of all relevant events that occur cluster-wide.

Cloudera Manager supports the following categories of events:

Category	Description
ACTIVITY_EVENT	Generated by the Activity Monitor; specifically, for jobs that fail, or that run slowly (as determined by comparison with duration limits). In order to monitor your workload for slow-running jobs, you must specify Activity Duration Rules on page 224.
AUDIT_EVENT	Generated by actions performed • In Cloudera Manager, such as creating, configuring, starting, stopping, and deleting services or roles • By services that are being audited by Cloudera Navigator.
HBASE	Generated by HBase with the exception of log messages, which have the LOG_MESSAGE category.
HEALTH_CHECK	Indicate that certain health test activities have occurred, or that health test results have met specific conditions (thresholds). Thresholds for various health tests can be set under the **Configuration** tabs for HBase, HDFS, Impala, and MapReduce service instances, at both the service and role level. See Configuring Health Monitoring on page 223 for more information.
LOG_MESSAGE	Generated for certain types of log messages from HDFS, MapReduce, and HBase services and roles. Log events are created when a log entry matches a set of rules for identifying messages of interest. The default set of rules is based on Cloudera experience supporting Hadoop clusters. You can configure additional log event rules if necessary.
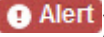
Category	Description
SYSTEM	Generated by system events such as parcel availability.

Viewing Events

The **Events** page lets you display events and alerts that have occurred within a time range you select anywhere in your clusters. From the Events page you can filter for events for services or role instances, hosts, users, commands, and much more. You can also search against the content information returned by the event.

To view events, click the **Diagnostics** tab on the top navigation bar, then select **Events**.

Events List

Event entries are ordered (within the time range you've selected) with the most recent at the top. If the event generated an alert, that is indicated by a red alert icon (ⓘ Alert) in the entry.

This page supports infinite scrolling: you can scroll to the end of the displayed results and the page will fetch more results and add them to the end of the list automatically.

To display event details, click ❯**Expand** at the right side of the event entry.

Clicking the ↻**View** link at the far right of the entry has different results depending on the category of the entry:

- **ACTIVITY_EVENT** - Displays the activity Details page.
- **AUDIT_EVENT** - If the event was a restart, displays the service's Commands page. If the event was a configuration change, the Revision Details dialog box displays.
- **HBASE** - Displays a health report or log details.
- **HEALTH_CHECK** - Displays the status page of the role instance.
- **LOG_MESSAGE** - Displays the event's log entry. You can also click ❯**Expand** to display details of the entry, then click the **URL** link. When you perform one of these actions the time range in the Time Line is shifted to the time the event occurred.
- **SYSTEM** - Displays the Parcels page.

Filtering Events

You filter events by selecting a time range and adding filters.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details). The time it takes to perform a search will typically increase for a longer time range, as the number of events to be searched will be larger.

Adding a Filter

To add a filter, do one of the following:

- Click the ▼ icon that displays next to a property when you hover in one of the event entries. A filter containing the property, operator, and its value is added to the list of filters at the left and Cloudera Manager redisplays all events that match the filter.
- Click the **Add a filter** link. A filter control is added to the list of filters.

  1. Choose a property in the drop-down list. You can search by properties such as Username, Service, Command, or Role. The properties vary depending on the service or role.
  2. If the property allows it, choose an operator in the operator drop-down list.
  3. Type a property value in the value text field. For some properties you can include multiple values in the value field. For example, you can create a filter like `Category = HEALTH_CHECK LOG_MESSAGE`. To drop individual values, click the ✕ to the right of the value. For properties where the list of values is finite and known, you can start typing and then select from a drop-down list of potential matches.
  4. Click **Search**. The log displays all events that match the filter criteria.

5. Click ✚ to add more filters and repeat steps 1 through 4.

> **Note:** You can filter on a string by adding a filter, selecting the property CONTENT, operator =, and typing the string to search for in the value field.

Removing a Filter

1. Click the ━ at the right of the filter. The filter is removed.
2. Click **Search**. The log displays all events that match the filter criteria.
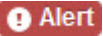
Re-running a Search

To re-run a recently performed search, click

▾

to the right of the Search button and select a search.

Alerts

An **_alert_** is an event that is considered especially noteworthy and is triggered by a selected event. Alerts are shown with an **❶ Alert** badge when they appear in a list of events. You can configure the Alert Publisher to send alert notifications by email or by SNMP trap to a trap receiver.

Service instances of type HDFS, MapReduce, and HBase (and their associated roles) can generate alerts if so configured. Alerts can also be configured for the monitoring roles that are a part of the Cloudera Management Service.

The settings to enable or disable specific alerts are found under the Configuration tab for the services to which they pertain. See Configuring Alerts on page 226 and for more information on setting up alerting.

For information about configuring the Alert Publisher to send email or SNMP notifications for alerts, see Configuring Alert Delivery on page 228.

Viewing What Alerts are Enabled and Disabled

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Do one of the following:

- Select **Administration** > **Alerts**.
- Display the All Alerts Summary page:

    1. Do one of the following:

        - Select **Clusters** > **Cloudera Management Service**.
        - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

    2. Click the **Instances** tab.
    3. Click an **Alert Publisher** role.
    4. Click the **Quick Links** tab and choose **All Alerts Summary**.

Managing Alerts

Minimum Required Role: Full Administrator

The **Administration** > **Alerts** page provides a summary of the settings for alerts in your clusters.

Alert Type The left column lets you select by alert type (Health, Log, or Activity) and within that by service instance. In the case of Health alerts, you can look at alerts for Hosts as well. You can select an individual service to see just the alert settings for that service.

Health/Log/Activity Alert Settings Depending on your selection in the left column, the right hand column show you the list of alerts that are enabled or disabled for the selected service type.

To change the alert settings for a service, click **Edit** next to the service name. This will take you to the Monitoring section of the Configuration tab for the service. From here you can enable or disable alerts and configure thresholds as needed.

Recipients You can also view the list of recipients configured for the enabled alerts.

Configuring Alert Delivery

When you install Cloudera Manager you can configure the mail server you will use with the Alert Publisher. However, if you need to change these settings, you can do so under the Alert Publisher section of the Management Services configuration tab. Under the Alert Publisher role of the Cloudera Manager Management Service, you can configure email or SNMP delivery of alert notifications and you can also configure a custom script that runs in response to an alert.

Configuring Alert Email Delivery

Minimum Required Role: [Full Administrator](#)

Sending A Test Alert E-mail

Select the **Administration** > **Alerts** tab and click the **Send Test Alert** link.

Configuring the List Of Alert Recipient Email Addresses

1. Select the **Administration** > **Alerts** tab and click **Edit** to the right of **Recipient(s)**.
2. Select **Scope** > **Alert Publisher**.
3. Select **Category** > **Main**.
4. Locate the **Alerts: Mail Message Recipients** property or search for it by typing its name in the Search box.
5. Configure the **Alerts: Mail Message Recipients** property.
6. Click the **Save Changes** button at the top of the page to save your settings.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See [Modifying Configuration Properties Using Cloudera Manager](#) on page 60.

7. Restart the Alert Publisher role.

Configuring Alert Email Properties

1. [Display the Cloudera Management Service](#) status page.
2. Click the **Configuration** tab.
3. Select **Scope** > **Alert Publisher**.
4. Select **Category** > **Main** to see the list of properties. To receive email alerts, you must set (or verify) the following settings:

   - Enable email alerts
   - Email protocol to use.
   - Your mail server hostname and port.
   - The username and password of the email user that will be logged into the mail server as the "sender" of the alert emails.
   - A comma-separated list of email addresses that will be the recipients of alert emails.
   - The format of the email alert message. Select **json** if you need the message to be parsed by a script or program.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See [Modifying Configuration Properties Using Cloudera Manager](#) on page 60.

5. Click the **Save Changes** button at the top of the page to save your settings.
6. Restart the Alert Publisher role.

Configuring Alert SNMP Delivery

Minimum Required Role: [Full Administrator](#)

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See [Managing Licenses](#) on page 37 for more information.

Enabling, Configuring, and Disabling SNMP Traps

1. Before you enable SNMP traps, configure the trap receiver (Network Management System or SNMP server) with the Cloudera MIB.
2. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

3. Click the **Configuration** tab.
4. Select **Scope** > **Alert Publisher** > **SNMP**.
5. Select **Category** > **SNMP**

   - Enter the DNS name or IP address of the Network Management System (SNMP server) acting as the trap receiver in the **SNMP NMS Hostname** property.
   - In the **SNMP Security Level** property, select the version of SNMP you are using: SNMPv2, SNMPv3 without authentication and without privacy (`noAuthNoPriv`), or SNMPv3 with authentication and without privacy (`authNoPriv`) and specify the required properties:

     – SNMPv2 - SNMPv2 Community String.
     – SNMPv3 without authentication (`noAuthNoPriv`) - SNMP Server Engine Id and SNMP Security UserName.
     – SNMPv3 with authentication (`authNoPriv`) - SNMP Server Engine Id, SNMP Security UserName, SNMP Authentication Protocol, and SNMP Authentication Protocol Pass Phrase.

   - You can also change other settings such as the port, retry, or timeout values.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See [Modifying Configuration Properties Using Cloudera Manager](#) on page 60.

6. Click **Save Changes** when you are done.
7. Restart the Alert Publisher role.

To disable SNMP traps, remove the hostname from the **SNMP NMS Hostname** property (alert.snmp.server.hostname).

Viewing the Cloudera MIB

1. Do one of the following:

   - Select **Clusters** > **Cloudera Management Service**.
   - On the **Home** > **Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.

2. Click the **Configuration** tab.
3. Select **Scope** > **Alert Publisher** > **SNMP**.
4. Select **Category** > **SNMP**.

5. Locate the **SNMP NMS Hostname** property and click the **?** icon to display the property description.

6. Click the **SMNP Mib** link.

Configuring Custom Alert Scripts

Minimum Required Role: [Full Administrator](#)

> **!** **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See [Managing Licenses](#) on page 37 for more information.

You can configure the Alert Publisher to run a user-written script in response to an [alert](#). The Alert Publisher passes a single argument to the script that is a UTF-8 [JSON file](#) containing a list of alerts. The script runs on the host where the Alert Publisher service is running and must have read and execute permissions for the **cloudera-scm** user. Only one instance of a script runs at a time. The standard out and standard error messages from the script are logged to the Alert Publisher log file.

You use the **Alert Publisher: Maximum Batch Size** and **Alert Publisher: Maximum Batch interval** to configure when the Alert Publisher delivers alerts. See [Configuring Alerts](#) on page 226.

To configure the Alert Publisher to deliver alerts using a script:

1. Save the script on the host where the Alert Publisher role is running.

2. Change the owner of the file to `cloudera-scm` and set its permissions to read and execute:

```
$ sudo chown cloudera-scm:cloudera-scm path_to_script
$ sudo chmod u+rx path_to_script
```

3. Open the Cloudera Manager Admin console and select **Clusters** > **Cloudera Management Service**.

4. Click the **Configuration** tab.

5. Select **Scope** > **Alert Publisher** .

6. Enter the path to the script in the **Custom Alert Script** property.

7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Sample JSON Alert File

When a custom script runs, it passes a JSON file that contains the alerts. For example:

```
[ {
  "body" : {
    "alert" : {
      "content" : "The health test result for MAPREDUCE_HA_JOB_TRACKER_HEALTH has become
 bad: JobTracker summary: myCluster.com (Availability: Active, Health: Bad). This health
 test reflects the health of the active JobTracker.",
      "timestamp" : {
        "iso8601" : "2015-06-11T03:52:56Z",
        "epochMs" : 1433994776083
      },
      "source" :
"http://myCluster.com:7180/cmf/eventRedirect/89521139-0859-4bef-bf65-eb141e63dbba",
      "attributes" : {
        "__persist_timestamp" : [ "1433994776172" ],
        "ALERT_SUPPRESSED" : [ "false" ],
        "HEALTH_TEST_NAME" : [ "MAPREDUCE_HA_JOB_TRACKER_HEALTH" ],
        "SEVERITY" : [ "CRITICAL" ],
        "HEALTH_TEST_RESULTS" : [ {
          "content" : "The health test result for MAPREDUCE_HA_JOB_TRACKER_HEALTH has
become bad: JobTracker summary: myCluster.com (Availability: Active, Health: Bad). This
 health test reflects the health of the active JobTracker.",
          "testName" : "MAPREDUCE_HA_JOB_TRACKER_HEALTH",
          "eventCode" : "EV_SERVICE_HEALTH_CHECK_BAD",
          "severity" : "CRITICAL"
        } ],
        "CLUSTER_DISPLAY_NAME" : [ "Cluster 1" ],
        "ALERT" : [ "true" ],
        "CATEGORY" : [ "HEALTH_CHECK" ],
```

```
             "BAD_TEST_RESULTS" : [ "1" ],
             "SERVICE_TYPE" : [ "MAPREDUCE" ],
             "EVENTCODE" : [ "EV_SERVICE_HEALTH_CHECK_BAD", "EV_SERVICE_HEALTH_CHECK_GOOD"
],
             "ALERT_SUMMARY" : [ "The health of service MAPREDUCE-1 has become bad." ],
             "CLUSTER_ID" : [ "1" ],
             "SERVICE" : [ "MAPREDUCE-1" ],
             "__uuid" : [ "89521139-0859-4bef-bf65-eb141e63dbba" ],
             "CLUSTER" : [ "Cluster 1" ],
             "CURRENT_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test result
 for MAPREDUCE_HA_JOB_TRACKER_HEALTH has become bad: JobTracker summary: myCluster.com
 (Availability: Active, Health: Bad). This health test reflects the health of the active

JobTracker.\",\"testName\":\"MAPREDUCE_HA_JOB_TRACKER_HEALTH\",\"eventCode\":\"EV_SERVICE_HEALTH_CHECK_BAD\",\"severity\":\"CRITICAL\"}",
  "{\"content\":\"The health test result for MAPREDUCE_TASK_TRACKERS_HEALTHY has become
 good: Healthy TaskTracker: 3. Concerning TaskTracker: 0. Total TaskTracker: 3. Percent
 healthy: 100.00%. Percent healthy or concerning:
100.00%.\",\"testName\":\"MAPREDUCE_TASK_TRACKERS_HEALTHY\",\"eventCode\":\"EV_SERVICE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}"
  ],
             "PREVIOUS_HEALTH_SUMMARY" : [ "GREEN" ],
             "CURRENT_HEALTH_SUMMARY" : [ "RED" ],
             "MONITOR_STARTUP" : [ "false" ],
             "PREVIOUS_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test
result for MAPREDUCE_HA_JOB_TRACKER_HEALTH has become good: JobTracker summary:
myCluster.com (Availability: Active, Health:
Good)\",\"testName\":\"MAPREDUCE_HA_JOB_TRACKER_HEALTH\",\"eventCode\":\"EV_SERVICE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
  "{\"content\":\"The health test result for MAPREDUCE_TASK_TRACKERS_HEALTHY has become
 good: Healthy TaskTracker: 3. Concerning TaskTracker: 0. Total TaskTracker: 3. Percent
 healthy: 100.00%. Percent healthy or concerning:
100.00%.\",\"testName\":\"MAPREDUCE_TASK_TRACKERS_HEALTHY\",\"eventCode\":\"EV_SERVICE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}"
  ],
             "SERVICE_DISPLAY_NAME" : [ "MAPREDUCE-1" ]
        }
      }
    },
    "header" : {
      "type" : "alert",
      "version" : 2
    }
}, {
    "body" : {
      "alert" : {
        "content" : "The health test result for JOB_TRACKER_SCM_HEALTH has become bad:
This role's process exited. This role is supposed to be started.",
        "timestamp" : {
          "iso8601" : "2015-06-11T03:52:56Z",
          "epochMs" : 1433994776083
        },
        "source" :
"http://myCluster.com:7180/cmf/eventRedirect/67b4d1c4-791b-428e-a9ea-8a09d4885f5d",
        "attributes" : {
          "__persist_timestamp" : [ "1433994776173" ],
          "ALERT_SUPPRESSED" : [ "false" ],
          "HEALTH_TEST_NAME" : [ "JOB_TRACKER_SCM_HEALTH" ],
          "SEVERITY" : [ "CRITICAL" ],
          "ROLE" : [ "MAPREDUCE-1-JOBTRACKER-10624c438dee9f17211d3f33fa899957" ],
          "HEALTH_TEST_RESULTS" : [ {
            "content" : "The health test result for JOB_TRACKER_SCM_HEALTH has become bad:
 This role's process exited. This role is supposed to be started.",
            "testName" : "JOB_TRACKER_SCM_HEALTH",
            "eventCode" : "EV_ROLE_HEALTH_CHECK_BAD",
            "severity" : "CRITICAL"
          } ],
          "CLUSTER_DISPLAY_NAME" : [ "Cluster 1" ],
          "HOST_IDS" : [ "75e763c2-8d22-47a1-8c80-501751ae0db7" ],
          "ALERT" : [ "true" ],
          "ROLE_TYPE" : [ "JOBTRACKER" ],
          "CATEGORY" : [ "HEALTH_CHECK" ],
          "BAD_TEST_RESULTS" : [ "1" ],
          "SERVICE_TYPE" : [ "MAPREDUCE" ],
          "EVENTCODE" : [ "EV_ROLE_HEALTH_CHECK_BAD", "EV_ROLE_HEALTH_CHECK_GOOD",
"EV_ROLE_HEALTH_CHECK_DISABLED" ],
          "ALERT_SUMMARY" : [ "The health of role jobtracker (nightly-1) has become bad."
```

```
        ],
            "CLUSTER_ID" : [ "1" ],
            "SERVICE" : [ "MAPREDUCE-1" ],
            "__uuid" : [ "67b4d1c4-791b-428e-a9ea-8a09d4885f5d" ],
            "CLUSTER" : [ "Cluster 1" ],
            "CURRENT_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test result
 for JOB_TRACKER_SCM_HEALTH has become bad: This role's process exited. This role is
supposed to be
started.\",\"testName\":\"JOB_TRACKER_SCM_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_BAD\",\"severity\":\"CRITICAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become
good: This role encountered 0 unexpected exit(s) in the previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_UNEXPECTED_EXITS\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_FILE_DESCRIPTOR has become good:
 Open file descriptors: 244. File descriptor limit: 32,768. Percentage in use:
0.74%.\",\"testName\":\"JOB_TRACKER_FILE_DESCRIPTOR\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_SWAP_MEMORY_USAGE has become
good: 0 B of swap memory is being used by this role's
process.\",\"testName\":\"JOB_TRACKER_SWAP_MEMORY_USAGE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE has
become good: This role's Log Directory (/var/log/hadoop-0.20-mapreduce) is on a filesystem
 with more than 20.00% of its space
free.\",\"testName\":\"JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_HOST_HEALTH has become good:
The health of this role's host is
good.\",\"testName\":\"JOB_TRACKER_HOST_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_WEB_METRIC_COLLECTION has become
 good: The web server of this role is responding with metrics. The most recent collection
 took 49
millisecond(s).\",\"testName\":\"JOB_TRACKER_WEB_METRIC_COLLECTION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_GC_DURATION has become good:
Average time spent in garbage collection was 0 second(s) (0.00%) per minute over the
previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_GC_DURATION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE
 has become disabled: Test disabled because role is not configured to dump heap when
out of memory. Test of whether this role's heap dump directory has enough free
space.\",\"testName\":\"JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_DISABLED\",\"severity\":\"INFORMATIONAL\"}"
    ],
            "CURRENT_HEALTH_SUMMARY" : [ "RED" ],
            "PREVIOUS_HEALTH_SUMMARY" : [ "GREEN" ],
            "MONITOR_STARTUP" : [ "false" ],
            "ROLE_DISPLAY_NAME" : [ "jobtracker (nightly-1)" ],
            "PREVIOUS_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test
result for JOB_TRACKER_SCM_HEALTH has become good: This role's status is as expected.
The role is
started.\",\"testName\":\"JOB_TRACKER_SCM_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become
good: This role encountered 0 unexpected exit(s) in the previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_UNEXPECTED_EXITS\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_FILE_DESCRIPTOR has become good:
 Open file descriptors: 244. File descriptor limit: 32,768. Percentage in use:
0.74%.\",\"testName\":\"JOB_TRACKER_FILE_DESCRIPTOR\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_SWAP_MEMORY_USAGE has become
good: 0 B of swap memory is being used by this role's
process.\",\"testName\":\"JOB_TRACKER_SWAP_MEMORY_USAGE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE has
become good: This role's Log Directory (/var/log/hadoop-0.20-mapreduce) is on a filesystem
 with more than 20.00% of its space
free.\",\"testName\":\"JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_HOST_HEALTH has become good:
The health of this role's host is
good.\",\"testName\":\"JOB_TRACKER_HOST_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_WEB_METRIC_COLLECTION has become
 good: The web server of this role is responding with metrics. The most recent collection
 took 49
millisecond(s).\",\"testName\":\"JOB_TRACKER_WEB_METRIC_COLLECTION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_GC_DURATION has become good:
Average time spent in garbage collection was 0 second(s) (0.00%) per minute over the
previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_GC_DURATION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
    "{\"content\":\"The health test result for JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE
 has become disabled: Test disabled because role is not configured to dump heap when
out of memory. Test of whether this role's heap dump directory has enough free
```

```
space.\",\"testName\":\"JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_DISABLED\",\"severity\":\"INFORMATIONAL\"}"
    ],
            "SERVICE_DISPLAY_NAME" : [ "MAPREDUCE-1" ],
            "HOSTS" : [ "myCluster.com" ]
        }
      }
    },
    "header" : {
      "type" : "alert",
      "version" : 2
    }
}, {
    "body" : {
      "alert" : {
        "content" : "The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become
bad: This role encountered 1 unexpected exit(s) in the previous 5 minute(s).This included
 1 exit(s) due to OutOfMemory errors. Critical threshold: any.",
        "timestamp" : {
          "iso8601" : "2015-06-11T03:53:41Z",
          "epochMs" : 1433994821940
        },
        "source" :
"http://myCluster.com:7180/cmf/eventRedirect/b8c4468d-08c2-4b5b-9bda-2bef892ba3f5",
        "attributes" : {
          "__persist_timestamp" : [ "1433994822027" ],
          "ALERT_SUPPRESSED" : [ "false" ],
          "HEALTH_TEST_NAME" : [ "JOB_TRACKER_UNEXPECTED_EXITS" ],
          "SEVERITY" : [ "CRITICAL" ],
          "ROLE" : [ "MAPREDUCE-1-JOBTRACKER-10624c438dee9f17211d3f33fa899957" ],
          "HEALTH_TEST_RESULTS" : [ {
          "content" : "The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become
 bad: This role encountered 1 unexpected exit(s) in the previous 5 minute(s).This included
 1 exit(s) due to OutOfMemory errors. Critical threshold: any.",
            "testName" : "JOB_TRACKER_UNEXPECTED_EXITS",
            "eventCode" : "EV_ROLE_HEALTH_CHECK_BAD",
            "severity" : "CRITICAL"
          } ],
          "CLUSTER_DISPLAY_NAME" : [ "Cluster 1" ],
          "HOST_IDS" : [ "75e763c2-8d22-47a1-8c80-501751ae0db7" ],
          "ALERT" : [ "true" ],
          "ROLE_TYPE" : [ "JOBTRACKER" ],
          "CATEGORY" : [ "HEALTH_CHECK" ],
          "BAD_TEST_RESULTS" : [ "1" ],
          "SERVICE_TYPE" : [ "MAPREDUCE" ],
          "EVENTCODE" : [ "EV_ROLE_HEALTH_CHECK_BAD", "EV_ROLE_HEALTH_CHECK_GOOD",
"EV_ROLE_HEALTH_CHECK_DISABLED" ],
          "ALERT_SUMMARY" : [ "The health of role jobtracker (nightly-1) has become bad."
  ],
          "CLUSTER_ID" : [ "1" ],
          "SERVICE" : [ "MAPREDUCE-1" ],
          "__uuid" : [ "b8c4468d-08c2-4b5b-9bda-2bef892ba3f5" ],
          "CLUSTER" : [ "Cluster 1" ],
          "CURRENT_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test result
 for JOB_TRACKER_SCM_HEALTH has become bad: This role's process exited. This role is
supposed to be
started.\",\"testName\":\"JOB_TRACKER_SCM_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_BAD\",\"severity\":\"CRITICAL\"}",
 "{\"content\":\"The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become bad:
 This role encountered 1 unexpected exit(s) in the previous 5 minute(s).This included
1 exit(s) due to OutOfMemory errors. Critical threshold:
any.\",\"testName\":\"JOB_TRACKER_UNEXPECTED_EXITS\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_BAD\",\"severity\":\"CRITICAL\"}",
 "{\"content\":\"The health test result for JOB_TRACKER_FILE_DESCRIPTOR has become good:
 Open file descriptors: 244. File descriptor limit: 32,768. Percentage in use:
0.74%.\",\"testName\":\"JOB_TRACKER_FILE_DESCRIPTOR\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
 "{\"content\":\"The health test result for JOB_TRACKER_SWAP_MEMORY_USAGE has become
good: 0 B of swap memory is being used by this role's
process.\",\"testName\":\"JOB_TRACKER_SWAP_MEMORY_USAGE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
 "{\"content\":\"The health test result for JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE has
become good: This role's Log Directory (/var/log/hadoop-0.20-mapreduce) is on a filesystem
 with more than 20.00% of its space
free.\",\"testName\":\"JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
 "{\"content\":\"The health test result for JOB_TRACKER_HOST_HEALTH has become good:
The health of this role's host is
good.\",\"testName\":\"JOB_TRACKER_HOST_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
```

```
          "{\"content\":\"The health test result for JOB_TRACKER_WEB_METRIC_COLLECTION has become
          good: The web server of this role is responding with metrics. The most recent collection
          took 49
millisecond(s).\",\"testName\":\"JOB_TRACKER_WEB_METRIC_COLLECTION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_GC_DURATION has become good:
Average time spent in garbage collection was 0 second(s) (0.00%) per minute over the
previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_GC_DURATION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE
          has become disabled: Test disabled because role is not configured to dump heap when
out of memory. Test of whether this role's heap dump directory has enough free
space.\",\"testName\":\"JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_DISABLED\",\"severity\":\"INFORMATIONAL\"}"
          ],
        "CURRENT_HEALTH_SUMMARY" : [ "RED" ],
        "PREVIOUS_HEALTH_SUMMARY" : [ "RED" ],
        "MONITOR_STARTUP" : [ "false" ],
        "ROLE_DISPLAY_NAME" : [ "jobtracker (nightly-1)" ],
        "PREVIOUS_COMPLETE_HEALTH_TEST_RESULTS" : [ "{\"content\":\"The health test
result for JOB_TRACKER_SCM_HEALTH has become bad: This role's process exited. This role
          is supposed to be
started.\",\"testName\":\"JOB_TRACKER_SCM_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_BAD\",\"severity\":\"CRITICAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_UNEXPECTED_EXITS has become
good: This role encountered 0 unexpected exit(s) in the previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_UNEXPECTED_EXITS\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_FILE_DESCRIPTOR has become good:
          Open file descriptors: 244. File descriptor limit: 32,768. Percentage in use:
0.74%.\",\"testName\":\"JOB_TRACKER_FILE_DESCRIPTOR\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_SWAP_MEMORY_USAGE has become
good: 0 B of swap memory is being used by this role's
process.\",\"testName\":\"JOB_TRACKER_SWAP_MEMORY_USAGE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE has
become good: This role's Log Directory (/var/log/hadoop-0.20-mapreduce) is on a filesystem
          with more than 20.00% of its space
free.\",\"testName\":\"JOB_TRACKER_LOG_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_HOST_HEALTH has become good:
The health of this role's host is
good.\",\"testName\":\"JOB_TRACKER_HOST_HEALTH\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_WEB_METRIC_COLLECTION has become
          good: The web server of this role is responding with metrics. The most recent collection
          took 49
millisecond(s).\",\"testName\":\"JOB_TRACKER_WEB_METRIC_COLLECTION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_GC_DURATION has become good:
Average time spent in garbage collection was 0 second(s) (0.00%) per minute over the
previous 5
minute(s).\",\"testName\":\"JOB_TRACKER_GC_DURATION\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_GOOD\",\"severity\":\"INFORMATIONAL\"}",
          "{\"content\":\"The health test result for JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE
          has become disabled: Test disabled because role is not configured to dump heap when
out of memory. Test of whether this role's heap dump directory has enough free
space.\",\"testName\":\"JOB_TRACKER_HEAP_DUMP_DIRECTORY_FREE_SPACE\",\"eventCode\":\"EV_ROLE_HEALTH_CHECK_DISABLED\",\"severity\":\"INFORMATIONAL\"}"
          ],
        "SERVICE_DISPLAY_NAME" : [ "MAPREDUCE-1" ],
        "HOSTS" : [ "myCluster.com" ]
      }
    }
  },
  "header" : {
    "type" : "alert",
    "version" : 2
  }
} ]
```

Triggers

A *trigger* is a statement that specifies an action to be taken when one or more specified conditions are met for a service, role, role configuration group, or host. The conditions are expressed as a [tsquery statement](), and the action to be taken is to change the health for the service, role, role configuration group, or host to either Concerning (yellow) or Bad (red).

Triggers can be created for services, roles, role configuration groups, or hosts. Create a trigger by doing one of the following:

- Directly editing the configuration for the service, role (or role configuration group), or host configuration.
- Clicking **Create Trigger** on the drop-down menu for most charts. Note that the Create Trigger command is not available on the drop-down menu for charts where no context (role, service, and so on) is defined, such as on the **Home** > **Status** tab.

> **Important:** Because triggers are a new and evolving feature, backward compatibility between releases is not guaranteed at this time.

- Use the **Create Trigger** expression builder. See [Creating a Trigger Using the Expression Editor](#) on page 296.

The Structure of Triggers

A trigger is defined by a JSON formatted string that includes four parts:

- Name
- Expression
- Stream threshold
- Whether or not the trigger should be enabled

Each of the four parts of a trigger is described in the following sections.

Name (required)

A trigger's name must be unique in the context for which the trigger is defined. That is, there cannot be two triggers for the same service or role with the same name. Different services or different roles can have triggers with the same name.

Expression (required)

A trigger expression takes the form:

```
IF (CONDITIONS) DO HEALTH_ACTION
```

When the conditions of the trigger are met, the trigger is considered to be firing. A condition is any valid tsquery statement. In most cases conditions employ stream filters to filter out streams below or above a certain threshold. For example, the following tsquery can be used to retrieve the streams for DataNodes with more than 500 open file descriptors:

```
SELECT fd_open WHERE roleType=DataNode AND last(fd_open) > 500
```

The stream filter used here, `last(fd_open) > 50`, is composed of four parts:

- A scalar producing function "last" that takes a stream and returns its last data point
- A metric to operate on
- A comparator
- A scalar value

Other scalar producing functions are available, like `min` or `max`, and they can be combined to create arbitrarily complex expressions:

```
last(moving_avg(fd_open)) >= 500
```

See the [tsquery documentation](#) for more details.

Conditions can be combined using the logical operators `AND` and `OR`. For example, here is a trigger expression with two conditions:

```
IF ((SELECT fd_open WHERE roleType=DataNode AND last(fd_open) > 500) OR (SELECT fd_open
  WHERE roleType=NameNode AND last(fd_open) > 500)) DO health:bad
```

A condition is met if it returns more than the number of streams specified by the `streamThreshold` (see below). A trigger fires if the logical evaluation of all of its conditions results in a met condition. When a trigger fires, two actions can be taken: `health:concerning` or `health:bad`. These actions change the health of the entity on which the trigger is defined.

Stream Threshold (optional)

The stream threshold determines the number of streams that need to be returned by the tsquery before the condition is met. The default is 0; that is, if the tsquery returns any results the condition will be met. For example if the stream threshold is set to 10 and the condition is `SELECT fd_open WHERE roleType=DataNode AND last(fd_open) > 500` the condition will be considered *met* only if there are at least 10 DataNodes that have more than 500 file descriptor opened, so at least 10 streams were returned by the tsquery.

Enabled (optional)

Whether the trigger is enabled. The default is `true`, (enabled).

Trigger Example

The following is a JSON formatted trigger that fires if there are more than 10 DataNodes with more than 500 file descriptors opened:
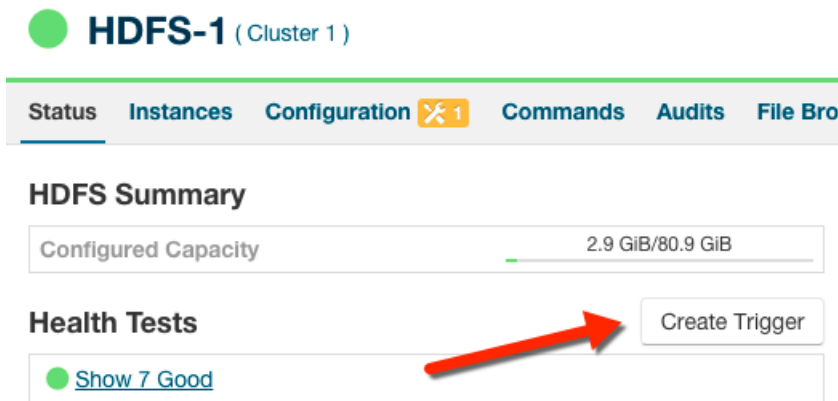
```
[{"triggerName": "sample-trigger", "triggerExpression": "IF (SELECT fd_open WHERE roleType
= DataNode and last(fd_open) > 500) DO health:bad", "streamThreshold": 10, "enabled":
"true"}]
```

Creating a Trigger Using the Expression Editor

The **Create New Trigger** screen allows you to use a graphical editor to build the JSON string that defines a trigger. You can use the expression editor section to build the tsquery statement, or you can edit the tsquery statement manually. Triggers use the <u>tsquery Language</u> on page 315 to create trigger expressions.

To create a trigger using the expression editor:

1. Go to a service, role, role configuration group, or host configuration page and click the **Create Trigger** button next to the **Health Test** section.

The **Create New Trigger** screen displays.

As you build the trigger, the actual query text displays to the right, along with a preview of a chart returned by the query.

2. Enter a <u>name</u> for the trigger in the **Name** field.
3. Build the <u>Metric Expressions</u> on page 316:

   a. Select the function to use in your expression, either **Last**, **Min**, or **Max**.
   b. Select the metric by typing its name in the **Metric** field. A list of available metrics displays as you type.

   c. Select the operator, either **>**, **>=**, **=**, **<**, or **<=**.

   d. Enter the value to use for the comparison in the **Value** field.

   e. (Optional) Click the **+** icon to add additional expressions. Additional expressions are added to the query using the logical operator **AND**.

4. (Optional) Create a predicate for the query. Under **Attribute Conditions**, click the **+** icon to add an attribute condition.

   A set of fields displays that you use to build an expression for the predicate.

   a. Type the attribute name in the **Attribute** field. A list of attributes displays as you type.

   b. Select the operator, either **=** or **RLIKE**.

   c. Enter the value for the comparison in the **Value** field.

   d. (Optional) Click the **+** icon to add additional expressions. Additional expressions are added to the predicate using the logical operator **AND**.

5. Select an **Action** from the drop-down menu to define the action taken when the trigger fires:

   • **Mark as bad** (red)
   • **Mark as concerning** (yellow)

6. Enter a value for the **Stream Threshold**. Leave the value set to 0 to include all streams; enter an integer to set the number of streams required to meet the condition. See Stream Threshold.

7. Select **Enabled** to enable the trigger. If you disable the trigger, it does not run.

8. (Optional) Select **Suppressed**. A suppressed trigger still runs but does not impact the health display of the owning entity.

9. Verify your expression:

   In the area to the right of the expression builder, in the **Preview** section, the expression you have built displays. A chart also displays the result of the query. Click **Show Filtered Streams** to see all streams. Click **Hide Filtered Streams** to hide streams that do not meet the Stream Threshold (optional) on page 296.

   You can edit your trigger using the fields in the expression builder, or you can click the **Edit Manually** link to display a text box in which you can manually edit the trigger. Click **Use Editor** to return to the expression builder.

   > ⓘ **Important:** If you select **Edit Manually**, changes you make manually do not appear in the expression builder when you click **Use Editor**.

10. Click **Create Trigger** to save your trigger.

Editing, Deleting, Suppressing, or Deleting a Trigger

1. Go to the service, role, role configuration group, or host configuration page where the trigger was created. (For example: select **Clusters** > **HDFS**.)

2. In the **Health Tests** section, click the trigger name. (You may need to click a **Show ...** link to expand the list of triggers.)

   A page displays showing the query and chart for the trigger. Click **Show Filtered Streams** to see all streams. Click **Hide Filtered Streams** to hide streams that do not meet the Stream Threshold (optional) on page 296.

3. Click the **Actions** drop-down menu and select one of the following actions:

   • **Edit Trigger**

     A page opens where you can edit the query. Click **Save Trigger** to save your changes.

   • **Disable Trigger** or **Enable Trigger**
   • **Suppress Trigger** or **Unsuppress Trigger**
   • **Delete Trigger**

Cloudera Manager Trigger Use Cases

Cloudera Manager allows you to monitor cluster performance. Some indicators require timely attention to keep your data safe. Triggers let you track occurrence and severity of issues so that you can fix problems before they result in system failures.

The conditions you create for your trigger can be quite complex, but they do not need to be in order to be useful. This topic describes two triggers that alert you when you are approaching capacity limits for your cluster.

Creating a Trigger for Memory Capacity

A common use case is to monitor memory usage, and trigger a warning if your system is approaching its upper limit.

To create a memory usage trigger, do the following.

1. In Cloudera Manager, go to the **Hosts** page.
2. Click a link in the **Name** column to open a host status page.
3. In the **Health Tests** section, click **Create Trigger**.
4. On the **New Trigger** page, enter the name `Resident Memory In Use`.
5. In **Expression**, set these metric conditions.

   a. **Scalar Function**: `Min`.
   b. **Metric**: `mem_rss`.
   c. **Comparator**: **>** (greater than).
   d. **Scalar Value**: `1.75GB`. (This is a low value for demonstration purposes, so that it will trigger the action. In practice, use a value that more accurately reflects the memory limits of your cluster.)

6. Set **Action** to `Mark as bad`.

   As you work, the **Preview** shows the resulting chart and current status of your host.

New Trigger

Name

Resident Memory in Use

Expression

METRIC CONDITIONS

| Min ⬍ | mem_rss | | > ⬍ | 1.75GB |

+

ATTRIBUTE CONDITIONS

+

ACTION

Mark as bad ⬍

---

7. Scroll down and choose whether to apply this trigger to **All hosts**.

8. Click **Create Trigger**.

Creating a Trigger for CPU Capacity

Another key indicator for performance is CPU capacity. Ideally, you will consistently use most of your allocated CPU resources on a regular basis without exceeding capacity.

To create a CPU capacity trigger, do the following.

1. In Cloudera Manager, go to the **Hosts** page.

2. Click a link in the **Name** column to open a host status page.

3. In the **Health Tests** section, click **Create Trigger**.

4. On the **New Trigger** page, enter the name `CPU Capacity`.

5. In **Expression**, set these metric conditions.

   a. **Scalar Function**: `Min`.

   b. **Metric**: `cpu_percent`.

   c. **Comparator**: **>** (greater than).

   d. **Scalar Value**: `90`.

6. Set **Action** to `Mark as concerning`.

7. Click **Create Trigger**.

This trigger fires whenever the CPU percentage exceeds 90%. However, just exceeding 90% of available CPU resources is not necessarily a bad thing. What would be of more concern is if CPU resources were to consistently exceed 90% over an extended period. You can modify the trigger to evaluate the average CPU usage over time.

To modify the trigger to capture high CPU usage in a five minute window, do the following.

1. In Cloudera Manager, go to the **Hosts** page.

2. Click a link in the **Name** column to open the host status page.

3. In the **Health Tests** section, click the **Show *n* Good** link.

4. Click the link for **CPU Capacity**.

5. Choose **Actions** > **Edit Trigger**.

   The metric expression function for average over time, `moving_avg`, is not available from the pop-up menu in the editor. You can edit the expression directly using tsquery language.

6. Above the **Expression** editor, click **Edit manually**.

7. Revise the expression as follows.

```
IF (select cpu_percent where entityName=$HOSTID and min(moving_avg(cpu_percent, 300) )
 > 90)
     DO health:concerning
```

Edit Trigger

New Trigger

Name

A trigger's name mus
trigger is defined. Th
same service or role
different roles can ha

CPU Capacity

Expression   Use editor

The trigger's express

```
IF (select cpu_percent where entityName=$HOSTID and min(moving_avg(cpu_percent, 300) ) > 90)
    DO health:concerning
```

```
IF (CONDITIONS)
```

A condition is any va
employ stream filters
certain threshold. Fo
used to retrieve the s
open file descriptors

```
SELECT fd_open
last(fd_open) >
```

Conditions can be co
and OR. For example
conditions:

```
IF ((SELECT fd_
last(fd_open) >
roleType=NameNo
health:bad
```

A condition is met if
number of streams (
evaluation of all of it

When a trigger is firi
'health:concerning' o
health of the entity o

8. Click **Save Trigger**.

For more information on defining triggers, see [Triggers](#) on page 294.

For more information on writing custom queries, see [tsquery Language](#) on page 315.

For the complete list of available metrics, see [Cloudera Manager Metrics](#).

Lifecycle and Security Auditing

Minimum Required Role: [Auditor](#) (also provided by **Full Administrator**)

An *audit event* is an event that describes an action that has been taken for a cluster, host, license, parcel, role, service or user.

Cloudera Manager records cluster, host, license, parcel, role, and service *lifecycle events* (activate, create, delete, deploy, download, install, start, stop, update, upgrade, and so on), user *security-related events* (add and delete user,

login failed and succeeded), and provides an audit UI and API to view, filter, and export such events. The Cloudera Manager audit log does not track the progress or results of commands (such as starting or stopping a service or creating a directory for a service), it just notes the command that was executed and the user who executed it. To view the progress or results of a command, follow the procedures in Viewing Running and Recent Commands on page 241.

The Cloudera Navigator Audit Server records *service access events* and the Cloudera Navigator Metadata Server provides an audit UI and API to view, filter, and export both service access events and the lifecycle and security events retrieved from Cloudera Manager. For information on Cloudera Navigator auditing features, see Exploring Audit Data.

Viewing Audit Events

You can view audit events for a cluster, service, role, or host.

Object	Procedure
Cluster	1. Click the **Audits** tab on the top navigation bar.
Service	1. Click the **Clusters** tab on the top navigation bar. 2. Select a service. 3. Click the **Audits** tab on the service navigation bar.
Role	1. Click the **Clusters** tab on the top navigation bar. 2. Select a service. 3. Click the **Instances** tab on the service navigation bar. 4. Select a role. 5. Click the **Audits** tab on the role navigation bar.
Host	1. Click the **Hosts** tab on the top navigation bar. 2. Select a host. 3. Click the **Audits** tab on the host navigation bar.

Audit event entries are ordered with the most recent at the top.

Audit Event Properties

The following properties can appear in an audit event entry:

- **Date** - Date and time the action was performed.
- **Command** - The action performed.
- **Source** - The object affected by the action.
- **User** - The name of the user that performed the action.
- **IP Address** - The IP address of the client that initiated the action.
- **Host IP Address** - The IP address of the host on which the action was performed.
- **Service** - The name of the service on which the action was performed.
- **Role** - The name of the role on which the action was performed.

Filtering Audit Events

You filter audit events by selecting a time range and adding filters.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details). When you select the time range, the log displays all events in that range. The time it takes to perform a search will typically increase for a longer time range, as the number of events to be searched will be larger.

Adding a Filter

To add a filter, do one of the following:

- Click the 🔽 icon that displays next to a property when you hover in one of the event entries. A filter containing the property, operator, and its value is added to the list of filters at the left and Cloudera Manager redisplays all events that match the filter.
- Click the **Add a filter** link. A filter control is added to the list of filters.

  1. Choose a property in the drop-down list. You can search by properties such as Username, Service, Command, or Role. The properties vary depending on the service or role.
  2. If the property allows it, choose an operator in the operator drop-down list.
  3. Type a property value in the value text field. To match a substring, use the `like` operator and specify `%` around the string. For example, to see all the audit events for files created in the folder `/user/joe/out` specify `Source like %/user/joe/out%`.
  4. Click **Search**. The log displays all events that match the filter criteria.
  5. Click ➕ to add more filters and repeat steps [1] through [4].

Removing a Filter

  1. Click the ➖ at the right of the filter. The filter is removed.
  2. Click **Search**. The log displays all events that match the filter criteria.

Downloading Audit Events

You can download audit events in CSV formats.

  1. Specify desired filters and time range.
  2. Click the **Download CSV** button. A file with the following fields is downloaded: `service`, `username`, `command`, `ipAddress`, `resource`, `allowed`, `timestamp`, `operationText`. The structure of the `resource` field depends on the type of the service:

      - HDFS - A file path
      - Hive, Hue, and Impala - *database*:*tablename*
      - HBase - *table family*:*qualifier*

For Hive, Hue, and Impala query and load commands, `operationText` is the query string.

HDFS Service Audit Log

```
service,username,command,ipAddress,resource,allowed,timestamp
hdfs1,cloudera,setPermission,10.20.187.242,/user/hive,false,"2013-02-09T00:59:34.430Z"
hdfs1,cloudera,getfileinfo,10.20.187.242,/user/cloudera,true,"2013-02-09T00:59:22.667Z"
hdfs1,cloudera,getfileinfo,10.20.187.242,/,true,"2013-02-09T00:59:22.658Z"
```

In this example, the first event access was denied, and therefore the `allowed` field has the value `false`.

Charting Time-Series Data

Cloudera Manager enables you to enter a query for a time series, chart the time-series data, group (facet) individual time series if your query produced multiple time series, and save the results as a [dashboard](#).

The following sections have more details on the terminology used, how to query for time-series data, displaying chart details, editing charts, and modifying chart properties.

Terminology

Entity

A Cloudera Manager component that has metrics associated with it, such as a service, role, or host.

Metric

A property that can be measured to quantify the state of an entity or activity, such as the number of open file descriptors or CPU utilization percentage.

Time series

A list of (time, value) pairs that is associated with some (entity, metric) pair such as, `(datanode-1, fd_open)`, `(hostname, cpu_percent)`. In more complex cases, the time series can represent operations on other time series. For example, `(datanode-1 , cpu_user + cpu_system)`.

Facet

A display grouping of a set of time series. By default, when a query returns multiple time series, they are displayed in individual charts. Facets allow you to display the time series in separate charts, in a single chart, or grouped by various attributes of the set of time series.

Building a Chart with Time-Series Data

1. Select **Charts** > **Chart Builder**.
2. Display time series in one of the following ways:

   - **Select a recently used statement**

     1. Click the

        [▾]

        to the right of the **Build Chart** button to display a list of recently run statements and select a statement. The statement text displays in the text box and the chart(s) that display that time series will display.

   - **Select from the list of Chart Examples**

     1. Click the question mark icon

        (?)

        to the right of the **Build Chart** button to display a list of examples with descriptions.
     2. Click **Try it** to create a chart based on the statement text in the example.

   - **Type a new statement**

     1. Press **Spacebar** in the text box. tsquery statement components display in a drop-down list. These suggestions are part of type ahead, which helps build valid queries. Scroll to the desired component and click **Enter**. Continue choosing query components by pressing **Spacebar** and **Enter** until the tsquery statement is complete.

For example, the query `SELECT jvm_heap_used_mb where clusterId = 1` could return a set of charts like the following:
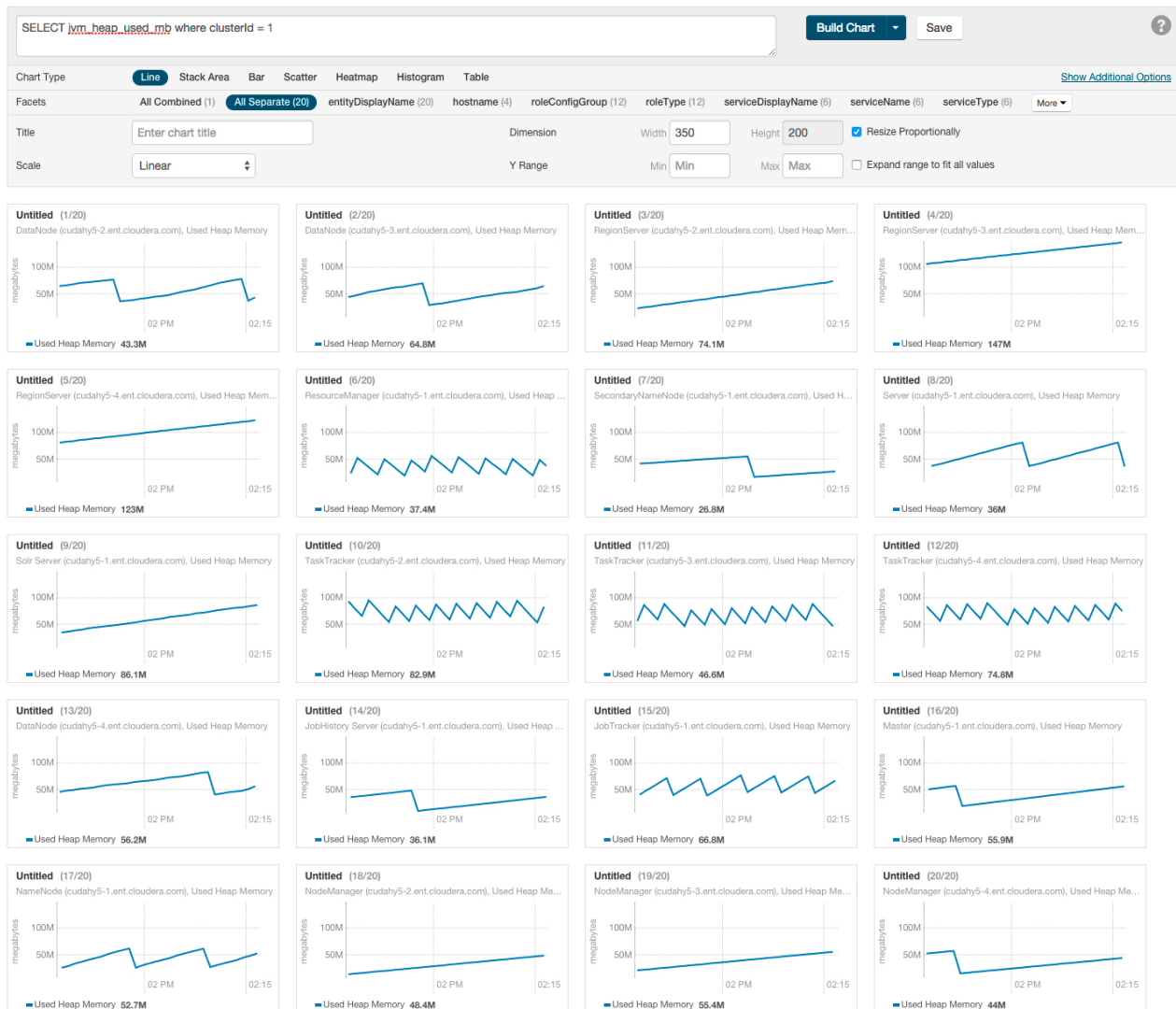
Configuring Time-Series Query Results

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

A time-series query returns one or more time series or scalar values. By default a maximum of 250 time series will be returned.

To change this value:

1. Select **Administration** > **Settings**.
2. In the **Advanced** category, set the **Maximum Number Of Time-Series Streams Returned Per Time-Series Query** or the **Maximum Number of Time-Series Streams Returned Per Heatmap** property.
3. Click **Save Changes**.

Using Context-Sensitive Variables in Charts

When editing charts from a service, role or host status or charts page, or when adding a chart to a status page, a set of context-sensitive variables (each beginning with '$') will be displayed below the query box on the Chart Builder page. For example, you might see variables similar to those in the query below:

Notice the $HOSTNAME portion of the query string. $HOSTNAME is a variable that will be resolved to a specific value based on the page before the query is actually issued. In this case, $HOSTNAME will become nightly53-2.ent.cloudera.com.

The chart below shows an example of the output of a similar query.

Context-sensitive variables are useful since they allow portable queries to be written. For example the query above may be on the host status page or any role status page to display the appropriate host's swap rate. Variables cannot be used in queries that are part of user-defined dashboards since those dashboards have no service, role or host context.

Chart Properties

By default, the time-series data retrieved by the tsquery is displayed on its own chart, using a **Line** style chart, a default size, and a default minimum and maximum for the Y-axis. You can change the chart type, facet the data, set the chart scale and size, and set X- and Y-axis ranges.

Changing the Chart Type

To change the chart type, click one of the chart types on the left:

- **Line** - Displays the points in the time series as continuous line.
- **Stack Area** - Displays the points in the time series as continuous line and the area under the line filled in.
- **Bar** - Displays each the value of the metric averaged over a second as a bar.
- **Scatter** - Displays the points in the time series as dots.
- **Heatmap** - Displays a metric thermometer and grid of colored squares. The thermometer displays buckets that represent a range of metric values and a color coding for the bucket. Each square represents an entity and the color of the square represents the value of a metric within a range. The following heatmap shows the last value of the resident memory for the NodeManager, ImpalaD, DataNode, and RegionServer roles.
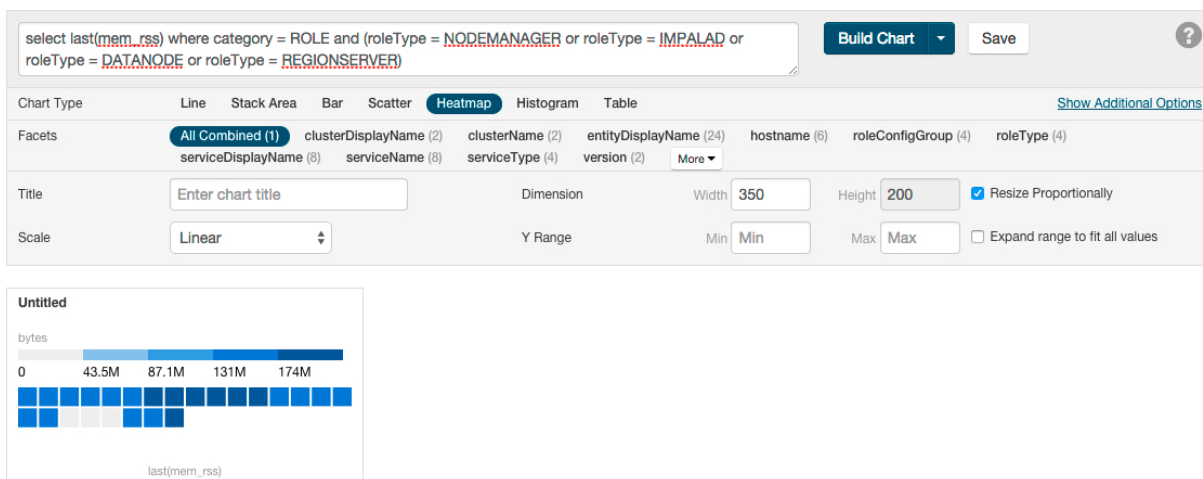
- **Histogram** - Displays the time series values as a set of bars where each bar represents a range of metric values and the height of the bar represents the number of entities whose value falls within the range. The following histogram shows the number of roles in each range of the last value of the resident memory.
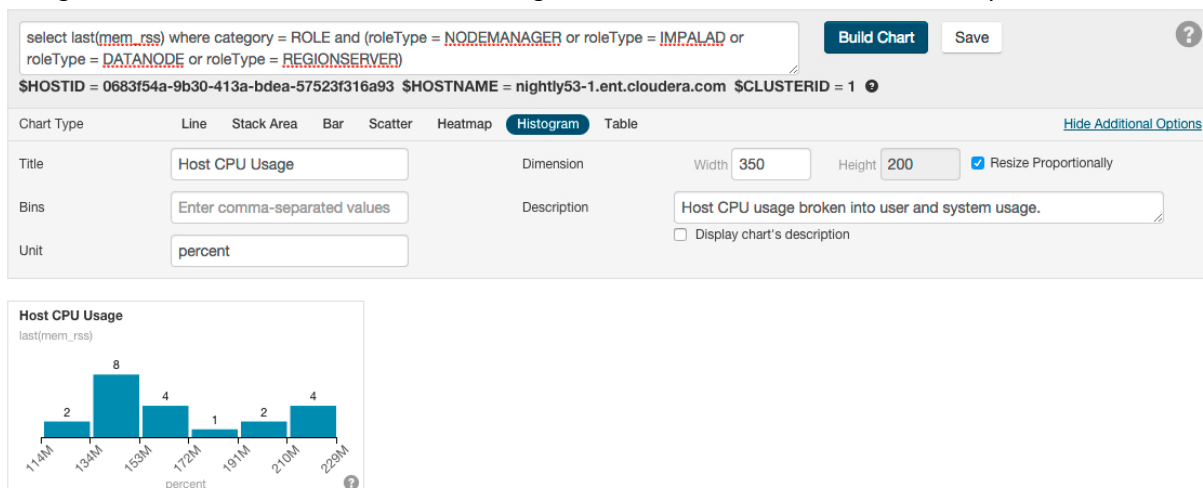
- **Table** - Displays the time series values as a table with each row containing the data for a single time value.

> **Note:** Heatmaps and histograms render charts for a single point as opposed to time series charts that render a series of points. For queries that return time series, Cloudera Manager will generate the heatmap or histogram based on the last recorded point in the series, and will issue the warning: "Query returned more than one value per stream. Only the last value was used." To eliminate this warning, use a scalar returning function to choose a point. For example, use `select last(cpu_percent)` to use the last point or `select max(cpu_percent)` to use the maximum value (in the selected time range).

Grouping (Faceting) Time Series

A time-series plot for a service, role, or host may actually be a composite of multiple individual time series. For example, the query `SELECT jvm_heap_used_mb where clusterId = 1` returns time-series data for the JVM heap used. Each time series has hostname, role type, metric, and entity name attributes. By default each attribute is displayed all on a single chart.

Using facets, you can combine time series based their attributes. To change the organization of the chart data, click one of the facets in the facet section in the upper part of the screen. The number in parentheses indicates how many charts will be displayed for that facet. As shown in the image below if the **serviceName** facet is selected for the JVM heap query, the time series is grouped into six charts, one chart each for each service name. The charts for service types with multiple roles contain multiple lines (for example, HBase, HDFS) while services that have only one role (for

example, ZooKeeper) contain just a single line. When a chart contains multiple lines, each entity is identified by a different color line.
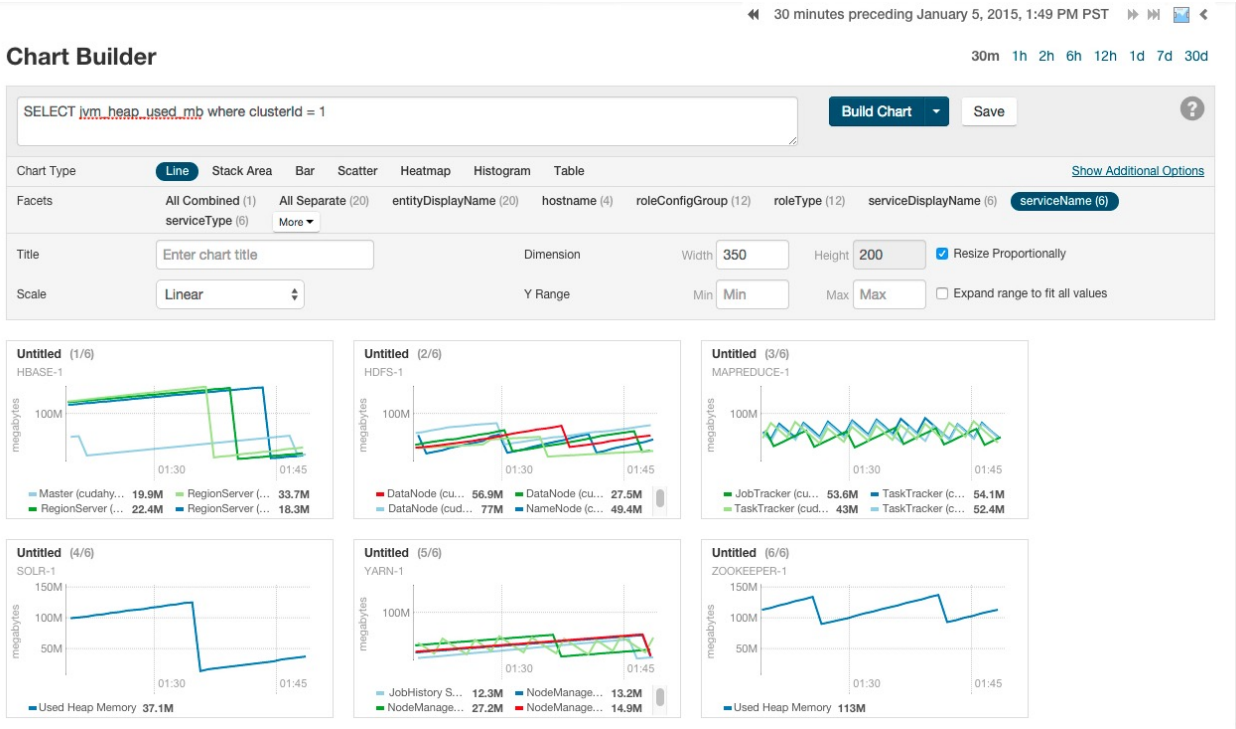
Changing Scale

You can set the scale of the chart to linear, logarithmic, and power.

Changing Dimensions

You can change the size of your charts by modifying the values in the **Dimension** fields. They change in 50-pixel increments when you click the up or down arrows, and you can type values in as long as they are multiples of 50. If you have multiple charts, depending on the dimensions you specify and the size of your browser window, your charts may appear in rows of multiple charts. If the **Resize Proportionally** checkbox is checked, you can modify one dimension and the other will be modified automatically to maintain the chart's width and height proportions.

The following chart shows the same query as the previous chart, but with **All Combined** selected (which shows all time series in a single chart) and with the Dimension values increased to expand the chart.
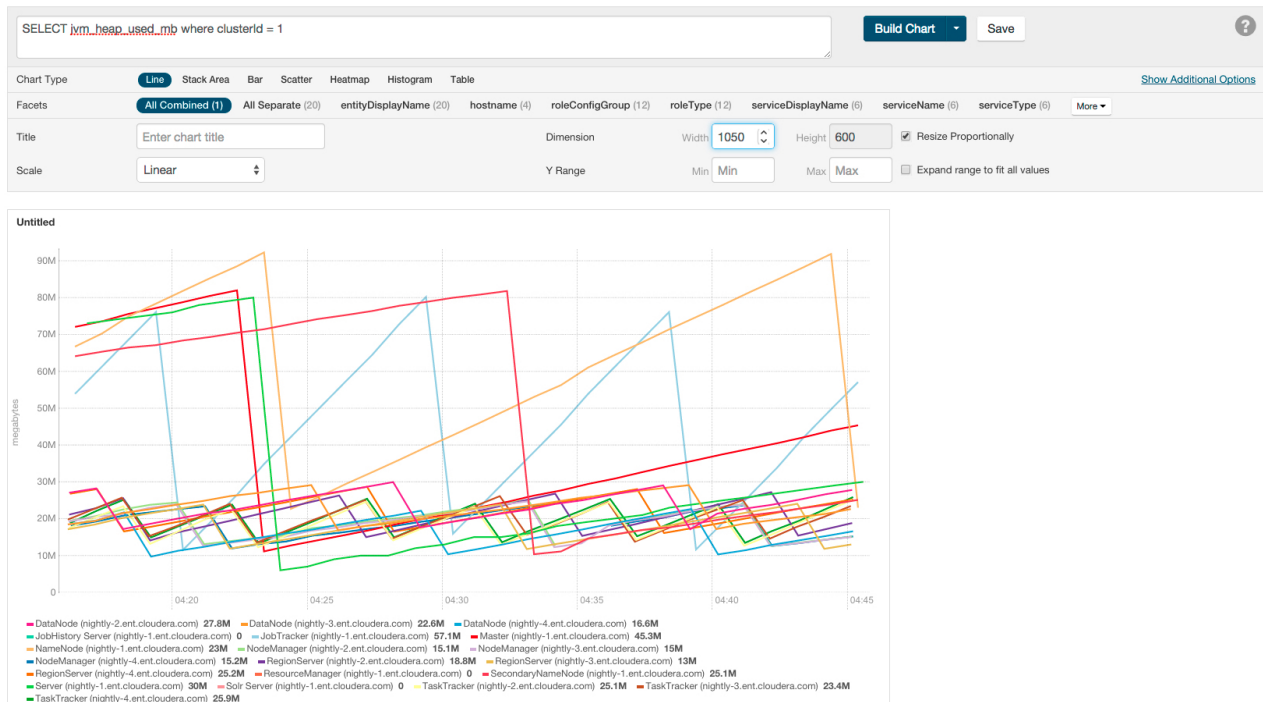
Changing Axes

You can change the Y-axis range using the **Y Range** minimum and maximum fields.

The X-axis is based on clock time, and by default shows the last hour of data. You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details).

Displaying Chart Details

When you move your mouse over a chart, its background turns gray, indicating that you can act upon it.

- Moving the mouse to a data point on a line, stack area, or bar chart shows the details about that data point in a pop-up tooltip.
- Click a line, stack area, scatter, or bar chart to expand it into a full-page view with a legend for the individual charted entities as well more fine-grained axes divisions.
    - If there are multiple entities in the chart, you can
        - Check and uncheck the legend item to hide or show the time series for the entities on the chart.

            ✓ ☐ vda (tcdn5-4.ent.cloudera.com), await_time ⌅ View   ✓ ■ vda (tcdn5-3.ent.cloudera.com), await_time ⌅ View

        - If there are service, role, or host instances in the chart, click the ⌅**View** link to display the instance's **Status** page.
    - Click the **Close** button to return to the regular chart view.

- **Heatmap** - Clicking a square in a heatmap displays a line chart of the time series for that entity.
- **Histogram** -
    - Mousing over the upper right corner of a histogram and clicking ⤢ opens a pop-up containing the query that generated the chart, an expanded view of the chart, a list of entity names and links to the entities whose metrics are represented by the histogram bars, and the value of the metric for each entity. For example, clicking the following histogram

displays the following:

- Clicking a bar in the expanded histogram displays a line chart of the time series from which the histogram was generated:
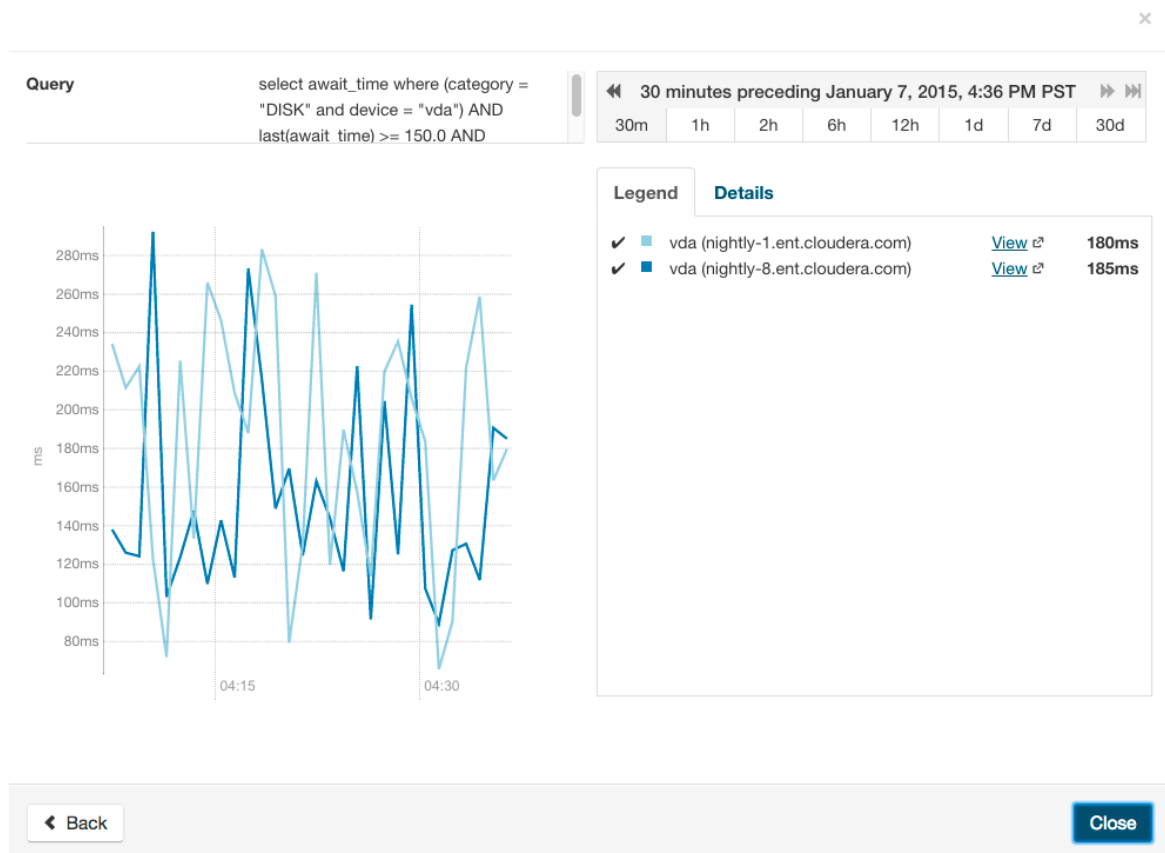
Clicking the **< Back** link at the bottom left of the line chart returns to the expanded histogram.

Editing a Chart

You can edit a chart from the [custom dashboard](#) and save it back into the same or another existing dashboard, or to a new custom dashboard. Editing a chart only affects the copy of the chart in the current dashboard – if you have copied the chart into other dashboards, those charts are not affected by your edits.

1. Move the cursor over the chart, and click the gear icon
   ⚙
   at the top right.
2. Click ⌨**Open in Chart Builder**. This opens the **Chart Builder** page with the chart you selected already displayed.
3. Edit the chart's select statement and click **Build Chart**.

Saving a Chart

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

After editing a chart you can save it to a new or existing custom dashboard.

1. Modify the chart's properties and click **Build Chart**.
2. Click **Save** to open the **Save Chart** dialog box, and select one of the following:

   a. Update chart in current dashboard: <name of current dashboard>.
   b. Add chart to another dashboard.
   c. Add chart to a new custom dashboard.

3. Click **Save Chart**.
4. Click **View Dashboard** to go to the dashboard where the chart has been saved.

See the following topics for more information:

- [Saving Charts to a New Dashboard](#) on page 313
- [Saving Charts to an Existing Dashboard](#) on page 314

Saving a chart only affects the copy of the chart in the dashboard where you save it – if you have previously copied the chart into other dashboards, those charts are not affected by your edits.

Users with Viewer, Limited Operator, or Operator user roles can edit charts and view the results, but cannot save them to a dashboard.

Obtaining Time-Series Data Using the API

Time-series data can be obtained using the Cloudera Manager API. For details about using a tsquery statement to obtain time-series data, see the `/timeseries` API documentation at `http://cmServerHost:7180/static/apidocs/path__timeseries.html`. To see the API call that returns the time-series data for an existing chart, click the blue down-arrow at the upper-right corner of the chart and click **Export JSON**. A new web browser window opens, displaying the time-series data in JSON format. The query string of the URL for that window displays the API call that retrieved the time-series data.

Dashboards

A **dashboard** is a set of [charts](#). This topic covers:

Dashboard Types

A **default dashboard** is a predefined set of charts that you cannot change. In a default dashboard you can:

- [Display chart details](#).
- [Edit a chart](#) and then [save back to a new or existing custom dashboard](#).

A **custom dashboard** contains a set of charts that you can change. In a custom dashboard you can:

- [Display chart details](#).
- [Edit a chart](#) and then [save back to a new or existing custom dashboard](#).
- [Save a chart](#), make any modifications, and then [save to a new or existing dashboard](#).
- [Remove a chart](#).

When you first display a page containing charts it has a custom dashboard with the same charts as a default dashboard.

Creating a Dashboard

1. Do one of the following:
   - Select **Charts** > **New Dashboard**.
   - Select **Charts** > **Manage Dashboards** and click **Create Dashboard**.
   - [Save a chart to a new dashboard](#).

2. Specify a name and optionally a duration.

   Conventional dashboard names follow the patterns given by one of the following Java regular expressions:

   ```
   Pattern.compile("^(.+):(\\d):" + STATUS_VIEW_SUFFIX + "$");
   ```

   ```
   Pattern.compile("^(.+):" + STATUS_VIEW_SUFFIX + "$");
   ```

   Examples of expected names are: `HOST:STATUS_VIEW`, `MGMT:STATUS_VIEW`, or `HDFS:5:STATUS_VIEW`. If the dashboard name does not match the expected pattern, a warning will be displayed in the server log.

3. Click **Create Dashboard**.

Managing Dashboards

To manage dashboards, select **Charts** > **Manage Dashboards**. You can create, clone, edit, export, import, and remove dashboards.

- **Create Dashboard** - create a new dashboard.
- **Clone** - clones an existing dashboard.
- **Edit** - edit an existing dashboard.
- **Export** - exports the specifications for the dashboard as a JSON file.
- **Import Dashboard** - reads an exported JSON file and recreates the dashboard.
- **Remove** - deletes the dashboard.

Configuring Dashboards

You can change the time scale of a dashboard, switch between default and custom dashboards, and reset a custom dashboard.

Setting the Time Scale of a Dashboard

By default the time scale of a dashboard is 30 minutes. To change the time scale, click a duration link

30m 1h 2h 6h 12h 1d 7d 30d   at the top-right of the dashboard.

Setting the Dashboard Type

To set the dashboard type, click and select one of the following:

- **Custom** - displays a custom dashboard.
- **Default** - displays a default dashboard.
- **Reset** - resets the custom dashboard to the predefined set of charts, discarding any customizations.

Saving Charts to Dashboards

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

You can save the charts and their configurations (type, dimension, and y-axis minimum and maximum) to a new dashboard or to an existing dashboard.

If your tsquery statement resulted in multiple charts, those charts are saved as a unit (either to a new or existing dashboard). You cannot edit the individual plots in that set of charts, but you can edit the set as a whole. A single edit button appears for the set that you saved — typically on the last chart in the set.

You can edit a copy of the individual charts in the set, but the edited copy does not change the original chart in the dashboard from which it was copied.

Saving Charts to a New Dashboard

1. Optionally modify the chart properties.
2. If the chart was created with the Chart Builder, optionally type a name for the chart in the Title field.
3. Do one of the following:

   - New chart - Click **Save**.
   - Existing chart - Move the cursor over the chart, and click the ⚙ icon at the top right.

4. Optionally edit the chart name.
5. Select the **Add chart to a new custom dashboard** option.
6. Enter a dashboard name.
7. Click **Save Chart**. The new dashboard appears on the menu under the top-level **Charts** tab.

Saving Charts to an Existing Dashboard

1. Optionally modify the chart properties.
2. If the chart was created with the Chart Builder, optionally type a name for the chart in the Title field.
3. Do one of the following:

   - New chart - Click **Save**.
   - Existing chart - Move the cursor over the chart, and click the ⚙ icon at the top right.

4. Optionally edit the chart name.
5. Select the **Add chart to an existing custom or system dashboard** option.
6. Select a dashboard from the **Dashboard Name** drop-down list.
7. Click **Save Chart**. The chart is added (appended) to the dashboard you select.

Adding a New Chart to the Home Page Custom Dashboard

You can add new charts to the custom dashboard on the **Home** > **Status** tab.

1. Click [✎ ▾] and select **Add From Chart Builder** - displays the **Add Chart To** *Dashboard* page, with variables preset for the specific cluster where you want to add the dashboard.

   a. Click the question mark icon [?] to the right of the **Build Chart** button and select a metric from the **List of Metrics**, type a metric name or description into the **Basic** text field, or type a query into the **Advanced** field.
   b. Click **Build Chart**. The charts that result from your query are displayed, and you can modify their chart type, combine them using facets, change their size and so on.

2. Click **Add**.

Adding a New Chart to the Custom Dashboard

You can add new charts to the custom dashboard on the Status tab of a service, host, or role.

1. Click [✎ ▾] and select one of the following:

   - **Add From Charts Library** - displays the charts page.

     1. Select one or more charts.

   - **Add From Chart Builder** - displays the **Add Chart To** *Dashboard* page, with variables preset for the specific service, role, or host where you want to add the dashboard.

     1. Click the question mark icon [?] to the right of the **Build Chart** button and select a metric from the **List of Metrics**, type a metric name or description into the **Basic** text field, or type a query into the **Advanced** field.
     2. Click **Build Chart**. The charts that result from your query are displayed, and you can modify their chart type, combine them using facets, change their size and so on.

2. Click **Add**.

> **Note:** If the query you've chosen has resulted in multiple charts, all the charts are added to the dashboard as a set. Although the individual charts in this set can be copied, you can only edit the set as a whole.

Removing a Chart from a Custom Dashboard

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

1. Move the cursor over the chart, and click the ⚙ icon at the top right.
2. Click **Remove**.

Moving and Resizing Charts on a Dashboard

You can move or resize the charts on a dashboard:

- Drag charts to a dashboard to change their relative positions.
- Change the size of a chart on a dashboard by dragging the lower-right corner of the chart.

tsquery Language

The tsquery language is used to specify statements for retrieving time-series data from the Cloudera Manager time-series datastore.

Before diving into the tsquery language specification, here's how you perform some common queries using the tsquery language:

1. Retrieve time series for all metrics for all DataNodes.

```
select * where roleType=DATANODE
```

2. Retrieve `cpu_user_rate` metric time series for all DataNodes.

```
select cpu_user_rate where roleType=DATANODE
```

3. Retrieve the `jvm_heap_used_mb` metric time series divided by 1024 and the `jvm_heap_committed` metric time series divided by 1024 for all roles running on the host named "my host".

```
select jvm_heap_used_mb/1024, jvm_heap_committed_mb/1024 where category=ROLE and
hostname="my host"
```

4. Retrieve the `jvm_total_threads` and `jvm_blocked_threads` metric time series for all entities for which Cloudera Manager collects these two metrics.

```
select jvm_total_threads,jvm_blocked_threads
```

tsquery Syntax

A tsquery statement has the following structure:

```
SELECT [metric expression]WHERE [predicate]
```

Note the following properties of tsquery statements:

- The statement `select *` is invalid.
- Tokens are case insensitive. For example, `Select`, `select`, and `SeLeCt` are all equivalent to `SELECT`.
- Multiple statements can be concatenated with semi-colons. Thus example 3 can be written as:

```
select jvm_heap_used_mb/1024 where category=ROLE and hostname=myhost; select
jvm_heap_commited_mb/1024 where category=ROLE and hostname=myhost
```

- The metric expression can be replaced with an asterisk (*), as shown in example 1. In that case, all metrics that are applicable for selected entities, such as `DATANODE` in example 1, are returned.
- The predicate can be omitted, as shown in example 4. In such cases, time series for all entities for which the metrics are appropriate are returned. For this query you would see the `jvm_new_threads` metric for NameNodes, DataNodes, TaskTrackers, and so on.

Metric Expressions

A *metric expression* generates the time series. It is a comma-delimited list of one or more metric expression statements. A *metric expression statement* is the name of a metric, a metric expression function, or a scalar value, joined by one or more metric expression operators.

See the FAQ on page 322 which answers questions concerning how to discover metrics and use cases for scalar values.

Metric expressions support the binary operators: +, -, *, /.

Here are some examples of metric expressions:

- `jvm_heap_used_mb, cpu_user, 5`
- `1000 * jvm_gc_time_ms / jvm_gc_count`
- `total_cpu_user + total_cpu_system`
- `max(total_cpu_user)`

Metric Expression Functions

Metric expressions support the functions listed in the following table. A function can return a time series or a scalar computed from a time series.

Functions that return scalars must be used for heatmap charts.

Function	Returns Scalar?	Description
`avg(metric expression)`	N	Computes a simple average for a time series.
`count_service_roles()`	Y	Returns the number of roles. There are three variants of this function: - `count_service_roles(roleType, roleState)` - Returns the number of roles of the specified `roleType` and `roleState`. For example, `count_service_roles(datanode, running)` returns the number of running DataNodes. - `count_service_roles(roleType)` - Returns the number of roles with the specified `roleType`. - `count_service_roles()` - Return the number of roles. For example, `select events_critical where count_service_roles() > 100` returns the `event_critical` metric when the number of roles is greater than 100.
`dt(metric expression)`	N	Derivative with negative values. The change of the underlying metric expression per second. For example: `dt(jvm_gc_count)`.
`dt0(metric expression)`	N	Derivative where negative values are skipped (useful for dealing with counter resets). The change of the underlying metric expression per second. For example: `dt0(jvm_gc_time_ms) / 10`.
`getClusterFact(string factName, double defaultValue)`	Y	Retrieves a fact about a cluster. Currently supports one fact: `numCores`. If the number of cores cannot be determined, `defaultValue` is returned.
`getHostFact(string factName, double defaultValue)`	Y	Retrieves a fact about a host. Currently supports one fact: `numCores`. If the number of cores cannot be determined, `defaultValue` is returned. For example, `select dt(total_cpu_user) / getHostFact(numCores, 2) where category=HOST` divides the results of `dt(total_cpu_user)` by the current number of cores for each host.

Function	Returns Scalar?	Description
		The following query computes the percentage of total user and system CPU usage each role is using on the host. It first computes the CPU seconds per second for the number of cores used by taking the derivative of the total user and system CPU times. It normalizes the result to the number of cores on the host by using the `getHostFact` function and multiplies the result by 100 to get the percentage. ```select dt0(total_cpu_user)/getHostFact(numCores,1)*100, dt0(total_cpu_system)/getHostFact(numCores,1)*100 where category=ROLE and clusterId=1```
greatest(*metric expression*, *scalar metric expression*)	N	Compares two metric expressions, one of which one is a scalar metric expression. Returns a time series where each point is the result of evaluating `max(`*point*`, `*scalar metric expression*`)`.
integral(*metric expression*)	N	Computes the integral value for a stream and returns a time-series stream within which each data point is the integral value of the corresponding data point from the original stream. For example, `select integral(maps_failed_rate)` will return the count of the failed number of maps.
counter_delta(*metric expression*)	N	Computes the difference in counter value for a stream and returns a time-series stream within which each data point is the difference in counter value of the corresponding data point from the counter value of previous data point in the original stream. For example: `select counter_delta(maps_failed_rate)` returns the count of the failed number of maps. This method is more accurate than the `integral()` function. However there are a few caveats: • This function is only implemented for single time-series streams. For streams of cross-entity aggregates, continue to use the `integral()` function. • If you apply this method for time-series streams which was created using a version of Cloudera Manager older than 5.7, Cloudera Manager fills in the older data points using the `integral()` function.
last(*metric expression*)	Y	Returns the last point of a time series. For example, to use the last point of the `cpu_percent` metric time series, use the expression `select last(cpu_percent)`.
least(*metric expression*, *scalar metric expression*)	N	Compares two metric expressions, of which one is a scalar metric expression. Returns a time series where each point is the result of evaluating `min(`*point*`, `*scalar metric expression*`)`.
max(*metric expression*)	Y	Computes the maximum value of the time series. For example, `select max(cpu_percent)`.
min(*metric expression*)	Y	Computes the minimum value of the time series.
moving_avg(*metric expression*, time_window_sec)	N	Computes the moving average for a time series over a time window `time_window_sec` specified in seconds (2, 0.1, and so on)
stats(*metric expression*, *stats name*)	N	Some time-series streams have additional statistics for each data point. These include rollup time-series streams, cross-entity aggregates, and rate metrics. The following statistics are available for rollup and cross-entity aggregates: max, min, avg, std_dev, and sample. For rate metrics, the underlying counter value is available using the "counter" statistics. For

Function	Returns Scalar?	Description
		example, `stats(fd_open_across_datanodes, max)` or `stats(swap_out_rate, counter)`.
`sum(metric expression)`	Y	Computes the sum value of the time-series.

Predicates

A **predicate** limits the number of streams in the returned series and can take one of the following forms:

- *time_series_attribute operator value*, where

  - *time_series_attribute* is one of the supported [attributes](#).
  - *operator* is one of = and `rlike`
  - *value* is an attribute value subject to the following constraints:

    - For attributes values that contain spaces or values of attributes of the form `xxxName` such as `displayName`, use quoted strings.
    - The value for the `rlike` operator must be specified in quotes. For example: `hostname rlike "host[0-3]+.*"`.
    - *value* can be any regular expression as specified in regular expression constructs in the Java [Pattern](#) class documentation.

- *`scalar_producing_function(metric_expression) comparator number`*, where

  - *scalar_producing_function* is any [function](#) that takes a time series and produces a scalar. For example, `min` or `max`.
  - *metric_expression* is a valid metric expression. For example, `total_cpu_user + total_cpu_system`.
  - *comparator* is a comparison operator: <, <=, =, !=, >=, >.
  - *number* is any number expression or a number expression with units. For example, `5`, `5mb`, `5s` are all valid number expressions. The valid units are:

    - Time - `ms` (milliseconds), `s` (seconds), `m` (minutes), `h` (hours), and `d` (days).
    - Bytes -`b` (bytes), `kb` or `kib` (kilobytes), `mb` or `mib` (megabytes), `gb` or `gib` (gigabytes), `tb` or `tib` (terabytes), and `pb` or `pib` (petabytes)
    - Bytes per second - Bytes and Time: `bps`, `kbps`, `kibps`, `mbps`, `mibps`, and so on. For example, 5 kilobytes per second is `5 kbps`.
    - Bytes time - Bytes and Time combined: `bms`, `bs`, `bm`, `bh`, `bd`, `kms`, `ks`, and so on. For example, 5 kilobytes seconds is `5 ks` or `5 kis`.

You use the `AND` and `OR` operators to compose compound predicates.

Example Statements with Compound Predicates

1. Retrieve all time series for all metrics for DataNodes or TaskTrackers.

```
select * where roleType=DATANODE or roleType=TASKTRACKER
```

2. Retrieve all time series for all metrics for DataNodes or TaskTrackers that are running on host named "myhost".

```
select * where (roleType=DATANODE or roleType=TASKTRACKER) and hostname=myhost
```

3. Retrieve the `total_cpu_user` metric time series for all hosts with names that match the regular expression "host[0-3]+.*"

```
select total_cpu_user where category=role and hostname rlike "host[0-3]+.*"
```

Example Statements with Predicates with Scalar Producing Functions

1. Return the entities where the last count of Java VM garbage collections was greater than 10:

```
select jvm_gc_count where last(jvm_gc_count) > 10
```

2. Return the number of open file descriptors where processes have more than 500Mb of `mem_rss`:

```
select fd_open where min(mem_rss) > 500Mb
```

Filtering by Day of Week or Hour of Day

You can add an expression to the predicate of a tsquery statement that limits the stream to specified days of the week or to a range of hours in each day.

By Day – Limits the stream to selected days of the week.

The `day in ()` expression takes an argument with a comma-separated list of days of the week, enclosed in parentheses. The days of the week are numbered 1 through 7; 1 = Monday, 2 = Tuesday, and so on. Use the following syntax:

```
day in (#, #, ...)
```

For example, the following expression limits the stream to events that occurred only on weekdays:

```
day in (1,2,3,4,5)
```

By Hour – Limits the stream to a range of hours each day.

The `hour in` expression takes an argument with a range of hours separated by a colon and enclosed in square brackets. Valid values are integers 0–23:

```
hour in [#:#]
```

For example, the following expression limits the stream to events that occur only between 9:00 a.m. and 5:00 p.m.:

```
hour in [9:17]
```

Add the day or time range expression after the `WHERE` clause. Do not use the `AND` keyword. For example:

```
select fd_open where category = ROLE and roleType = SERVICEMONITOR day in (1,2,3,4,5)
```

You can also combine `day in` and `hour in` expressions. Always put the `day` expression before the `hour` expression. The following example limits the stream to weekdays between 9:00 a.m. and 5:00 p.m.:

```
select fd_open where category = ROLE and roleType = SERVICEMONITOR day in (1,2,3,4,5)
hour in [9:17]
```

Time Series Attributes

Attribute names and most attribute values are case insensitive. `displayName` and `serviceType` are two attributes whose values are *case sensitive*.

Name	Description
`active`	Indicates whether the entities to be retrieved must be active. A nonactive entity is an entity that has been removed or deleted from the cluster. The default is to retrieve only active entities (that is, `active=true`). To access time series for deleted or removed entities, specify `active=false` in the query. For example: `SELECT fd_open WHERE roleType=DATANODE and active=false`

Name	Description
agentName	A Flume agent name.
applicationName	One of the Cloudera Manager monitoring daemon names.
cacheId	The HDFS cache directive ID.
category	The category of the entities returned by the query: CLUSTER, DIRECTORY, DISK, FILESYSTEM, FLUME_SOURCE, FLUME_CHANNEL, FLUME_SINK, HOST, HTABLE, IMPALA_QUERY_STREAM, NETWORK_INTERFACE, ROLE, SERVICE, USER, YARN_APPLICATION_STREAM, YARN_QUEUE. Some metrics are collected for more than one type of entity. For example, total_cpu_user is collected for entities of category HOST and ROLE. To retrieve the data only for hosts use: select total_cpu_user where category=HOST The ROLE category applies to all role types (see roleType attribute). The SERVICE category applies to all service types (see serviceType attribute). For example, to retrieve the committed heap for all roles on host1 use: select jvm_committed_heap_mb where category=ROLE and hostname="host1"
clusterDisplayName	The user-defined display name of a cluster.
clusterName	The cluster ID. To specify the cluster by its display name, use the clusterDisplayName attribute.
componentName	A Flume component name. For example, channel1, sink1.
device	A disk device name. For example, sda.
entityName	A display name plus unique identifier. For example: HDFS-1-DATANODE-692d141f436ce70aac080aedbe83f887.
expired	A Boolean that indicates whether an HDFS cache directive expired.
groupName	A user group name.
hbaseNamespace	The name of the HBase namespace.
hostId	The canonical identifier for a host in Cloudera Manager. It is unique and immutable. For example: 3d645222-2f7e-4895-ae51-cd43b91f1e7a.
hostname	A hostname.
hregionName	The HBase region name. For example, 4cd887662e5c2f3cd5dd227bb03dd760.
hregionStartTimeMs	Milliseconds from UNIX epoch since Cloudera Manager monitoring started collecting metrics for the HBase region.
htableName	The name of an HBase table.
iface	A network interface name. For example, eth0.
logicalPartition	A Boolean indicating whether or not the disk is a logical partition. Applies to disk entity types.
mountpoint	A mount point name. For example, /var, /mnt/homes.
nameserviceName	The name of the HDFS nameservice.
ownerName	The owner username.
partition	A partition name. Applies to partition entity types.

Name	Description
path	A filesystem path associated with the time-series entity.
poolName	A pool name. For example, `hdfs cache pool`, `yarn pools`.
queueName	The name of a YARN queue.
rackId	A Rack ID. For example, `/default`.
roleConfigGroup	The role group that a role belongs to.
roleName	The role ID. For example, `HBASE-1-REGIONSERVER-0b0ad09537621923e2b460e5495569e7`.
roleState	The role state: BUSY, HISTORY_NOT_AVAILABLE, NA, RUNNING, STARTING, STOPPED, STOPPING, UNKNOWN
roleType	The role type: ACTIVITYMONITOR, AGENT, ALERTPUBLISHER, BEESWAX_SERVER, CATALOGSERVER, DATANODE, EVENTSERVER, FAILOVERCONTROLLER, HBASE_INDEXER, HBASERESTSERVER, HBASETHRIFTSERVER, HIVEMETASTORE, HIVESERVER2, HOSTMONITOR, HTTPFS, HUESERVER, IMPALAD, JOBHISTORY,JOBTRACKER, JOURNALNODE, KT_RENEWER, LLAMA, MASTER, NAVIGATOR, REGIONSERVER, SERVICEMONITOR, NAMENODE, NODEMANAGER, REPORTSMANAGER, SECONDARYNAMENODE, SERVER, SOLR_SERVER, SQOOP_SERVER, STATESTORE, TASKTRACKER.
rollup	The time-series store table rollup type.
schedulerType	The scheduler type associated with the pool service.
serviceDisplayName	The user-defined display name of a service entity.
serviceName	The service ID. To specify a service by its display name use the `serviceDisplayName` attribute.
serviceState	The service state: HISTORY_NOT_AVAILABLE, NA, RUNNING, STARTING, STOPPED, STOPPING, UNKNOWN
serviceType	The service type: ACCUMULO,FLUME, HDFS, HBASE, HIVE, HUE, IMPALA, KS_INDEXER, MAPREDUCE, MGMT, OOZIE,SOLR, SPARK,SQOOP,YARN, ZOOKEEPER.
solrCollectionName	The Solr collection name. For example, `my_collection`.
solrReplicaName	The Solr replica name. For example, `my_collection_shard1_replica1`.
solrShardName	The Solr shard name. For example, `shard1`.
systemTable	A boolean indicating whether the HBase table is a system table or not.
tableName	The name of a table.
userName	The name of the user.
version	The version of the cluster. The value can be any of the supported CDH major versions.

Time Series Entities and their Attributes

The following table shows the entities and associated attributes that can appear in the predicate ("where" clause) of a tsquery statement.

Entity	Attributes
All Roles	roleType, hostId, hostname, rackId, serviceType, serviceName
All Services	serviceName, serviceType, clusterId, version, serviceDisplayName, clusterDisplayName

Entity	Attributes
Agent	roleType, hostId, hostname, rackId, serviceType, serviceName, clusterId, version, agentName, serviceDisplayName, clusterDisplayName
Cluster	clusterId, version, clusterDisplayName
Directory	roleName, hostId, path, roleType, hostname, rackId, serviceType, serviceName, clusterId, version, agentName, hostname, clusterDisplayName
Disk	device, logicalPartition, hostId, rackId, clusterId, version, hostname, clusterDisplayName
File System	hostId, mountpoint, rackId, clusterId, version, partition, hostname, clusterDisplayName
Flume Channel	serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType, clusterId, version, agentName, serviceDisplayName, clusterDisplayName
Flume Sink	serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType, clusterId, version, agentName, serviceDisplayName, clusterDisplayName
Flume Source	serviceName, hostId, rackId, roleName, flumeComponent, roleType, serviceType, clusterId, version, agentName, serviceDisplayName, clusterDisplayName
HDFS Cache Pool	serviceName, poolName, nameserviceName, serviceType, clusterId, version, groupName, ownerName, serviceDisplayName, clusterDisplayName
HNamespace	serviceName, namespaceName, serviceType, clusterId, version, serviceDisplayName, clusterDisplayName
Host	hostId, rackId, clusterId, version, hostname, clusterDisplayName
HRegion	htableName, hregionName, hregionStartTimeMs, namespaceName, serviceName, tableName, serviceType, clusterId, version, roleType, hostname, roleName, hostId, rackId , serviceDisplayName, clusterDisplayName
HTable	namespaceName, serviceName, tableName, serviceType, clusterId, version, serviceDisplayName, clusterDisplayName
Network Interface	hostId, networkInterface, rackId, clusterId, version, hostname, clusterDisplayName
Rack	rackId
Service	serviceName, serviceType, clusterId, serviceDisplayName
Solr Collection	serviceName, serviceType, clusterId, version, serviceDisplayName, clusterDisplayName
Solr Replica	serviceName, solrShardName, solrReplicaName, solrCollectionName, serviceType, clusterId, version, roleType, hostId, hostname, rackId, roleName, serviceDisplayName, clusterDisplayName
Solr Shard	serviceName, solrCollectionName, solrShardName, serviceType, clusterId, version, serviceDisplayName, clusterDisplayName
Time Series Table	tableName, roleName, roleType, applicationName, rollup, path
User	userName
YARN Pool	serviceName, queueName, schedulerType

FAQ

How do I compare information across hosts?

1. Click **Hosts** in the top navigation bar and click a host link.
2. In the Charts pane, choose a chart, for example **Host CPU Usage** and select ⚙ and then **Open in Chart Builder**.
3. In the text box, remove the `where entityName=$HOSTID` clause and click **Build Chart**.

4. In the Facets list, click **hostname** to compare the values across hosts.
5. Configure the time scale, minimums and maximums, and dimension. For example:
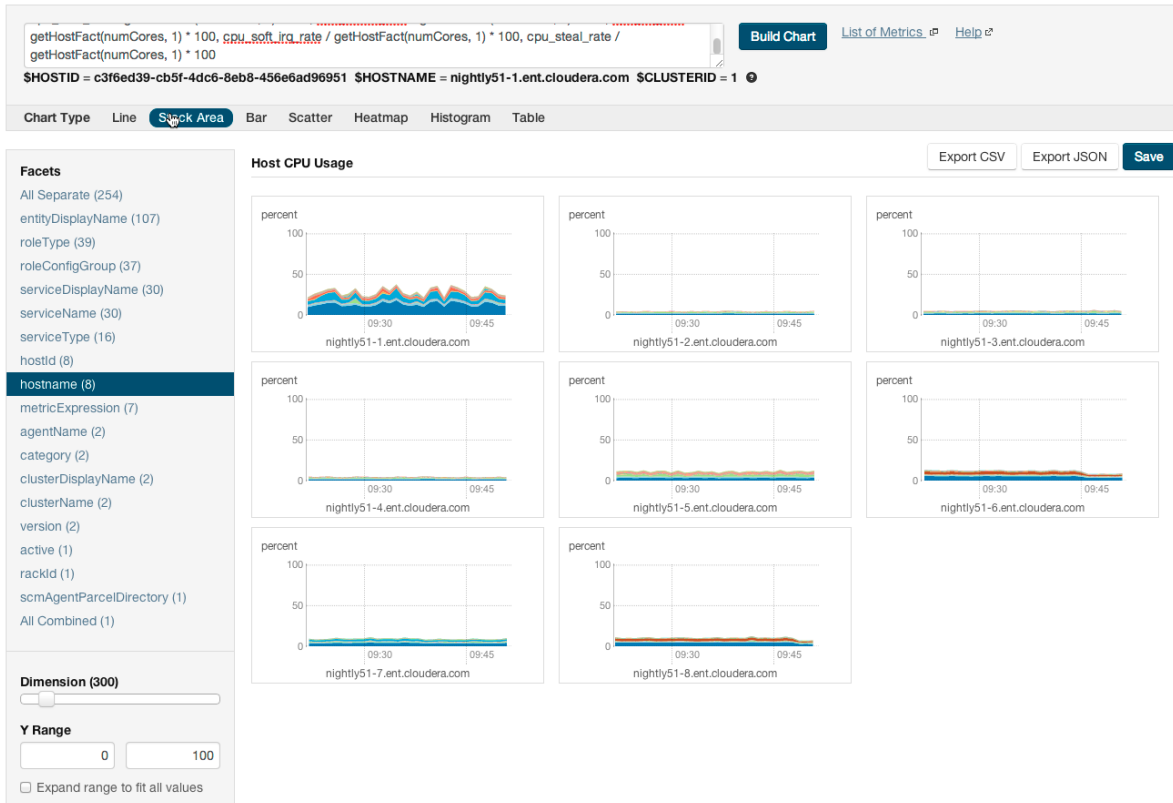
How do I compare all disk IO for all the DataNodes that belong to a specific HDFS service?

Use a query of the form:

```
select bytes_read, bytes_written where roleType=DATANODE and serviceName=hdfs1
```

replacing `hdfs1` with your HDFS service name. Then facet by **metricDisplayName** and compare all DataNode `byte_reads` and `byte_writes` metrics at once. See Grouping (Faceting) Time Series on page 307 for more details about faceting.

When would I use a derivative function?

Some metrics represent a counter, for example, `bytes_read`. For such metrics it is sometimes useful to see the rate of change instead of the absolute counter value. Use `dt` or `dt0` derivative functions.

When should I use the `dt0` function?

Some metrics, like `bytes_read` represent a counter that always grows. For such metrics a negative rate means that the counter has been reset (for example, process restarted, host restarted, and so on). Use `dt0` for these metrics.

How do I display a threshold on a chart?

Suppose that you want to retrieve the latencies for all disks on your hosts, compare them, and show a threshold on the chart to easily detect outliers. Use the following query to retrieve the metrics and the threshold:

```
select await_time, await_read_time, await_write_time, 250 where category=disk
```

Then choose **All Combined (1)** in the Facets list. The scalar threshold `250` will also be rendered on the chart:
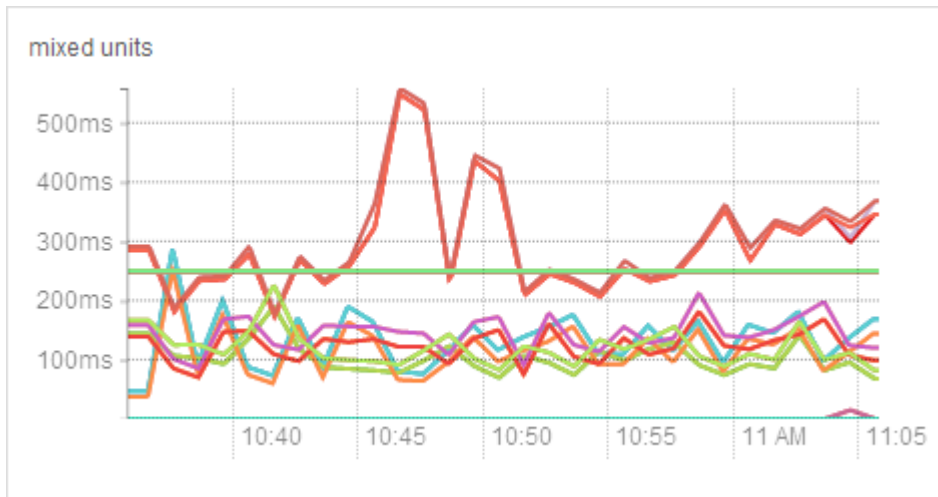
See Grouping (Faceting) Time Series on page 307 for more details about faceting.

I get the warning "The query hit the maximum results limit". How do I work around the limit?

There is a limit on the number of results that can be returned by a query. When a query results in more time-series streams than the limit a warning for "partial results" is issued. To circumvent the problem, reduce the number of metrics you are trying to retrieve or see Configuring Time-Series Query Results on page 305.

You can use the `rlike` operator to limit the query to a subset of entities. For example, instead of

```
select await_time, await_read_time, await_write_time, 250 where category=DISK
```

you can use

```
select await_time, await_read_time, await_write_time, 250 where category=DISK and hostname
rlike "host1[0-9]?.cloudera.com"
```

The latter query retrieves the disk metrics for ten hosts.

How do I discover which metrics are available for which entities?

- Type **Select** in the text box and then press **Space** or continue typing. Metrics matching the letters you type display in a drop-down list.
- Select **Charts** > **Chart Builder**, click the question mark icon
  ⍰
  to the right of the **Build Chart** button and click the **List of Metrics** link
- Retrieve all metrics for the type of entity:

```
select * where roleType=DATANODE
```

Metric Aggregation

In addition to collecting and storing raw metric values, the Cloudera Manager Service Monitor and Host Monitor produce a number of aggregate metrics from the raw metric data. Where a raw data point is a timestamp value pair, an aggregate metric point is a timestamp paired with a bundle of statistics including the minimum, maximum, average, and standard deviation of the data points considered by the aggregate.

Individual metric streams are aggregated across time to produce statistical summaries at different data granularities. For example, an individual metric stream of the number of open file descriptors on a host will be aggregated over time to the ten-minute, hourly, six-hourly, daily and weekly data granularities. A point in the hourly aggregate stream will include the maximum number of open file descriptors seen during that hour, the minimum, the average and so on. When servicing a time-series request, either for the Cloudera Manager UI or API, the Service Monitor and Host Monitor automatically choose the appropriate data granularity based on the time-range requested.

Cross-Time Aggregate Example

Consider the following `fd_open` raw metric values for a host:

```
9:00, 100 fds
9:01, 101 fds
9:02, 102 fds
. . .
9:09, 109 fds
```

The ten minutely cross-time aggregate point covering the ten-minute window from 9:00 - 9:10 would have the following statistics and metadata:

```
min: 100 fds
min timestamp: 9:00
max 109 fds
max timestamp 9:09
mean 104.5 fds
standard deviation: 3.02765 fds
count: 10 points
sample: 109 fds
sample timestamp: 9:09
```

The Service Monitor and Host Monitor also produce cross-entity aggregates for a number of entities in the system. Cross-entity aggregates are produced by considering the metric value of a particular metric across a number of entities of the same type at a particular time. For each stream considered, two metrics are produced. The first tracks statistics such as the minimum, maximum, average and standard deviation across all considered entities as well as the identities of the entities that had the minimum and maximum values. The second tracks the sum of the metric across all considered entities.

An example of the first type of cross-entity aggregate is the `fd_open_across_datanodes` metric. For an HDFS service this metric contains aggregate statistics on the `fd_open` metric value for all the DataNodes in the service. For a rack this metric contains statistics for all the DataNodes within that rack, and so on. An example of the second type of cross-entity aggregate is the `total_fd_open_across_datanodes` metric. For an HDFS service this metric contains the total number of file descriptors open by all the DataNodes in the service. For a rack this metric contains the total number of file descriptors open by all the DataNodes within the rack, and so on. Note that unlike the first type of cross-entity aggregate, this total type of cross-entity aggregate is a simple timestamp, value pair and not a bundle of statistics.

Cross-Entity Aggregate Example

Consider the following `fd_open` raw metric values for a set of ten DataNodes in an HDFS service at a given timestamp:

```
datanode-0, 200 fds
datanode-1, 201 fds
datanode-2, 202 fds
…
datanode-9, 209 fds
```

The cross-entity aggregate `fd_open_across_datanodes` point for that HDFS service at that time would have the following statistics and metadata:

```
min: 200 fds
min entity: datanode-0
max: 209 fds
max entity: datanode-9
mean: 204.5 fds
standard deviation: 3.02765 fds
count: 10 points
sample: 209 fds
sample entity: datanode-9
```

Just like every other metric, cross-entity aggregates are aggregated across time. For example, a point in the hourly aggregate of `fd_open_across_datanodes` for an HDFS service will include the `maximum fd_open` value of any

DataNode in that service over that hour, the average value over the hour, and so on. A point in the hourly aggregate of `total_fd_open_across_datanodes` for an HDFS service will contain statistics on the value of the `total_fd_open_across_datanodes` for that service over the hour.

Presentation of Aggregate Data

Aggregate data points returned from the Cloudera Manager API appear as shown in this section.

A cross-time aggregate:

```
{
        "timestamp" : "2014-02-24T00:00:00.000Z",
        "value" : 0.014541698027508003,
        "type" : "SAMPLE",
        "aggregateStatistics" : {
          "sampleTime" : "2014-02-23T23:59:35.000Z",
          "sampleValue" : 0.0,
          "count" : 360,
          "min" : 0.0,
          "minTime" : "2014-02-23T18:00:35.000Z",
          "max" : 2.9516129032258065,
          "maxTime" : "2014-02-23T19:37:36.000Z",
          "mean" : 0.014541698027508003,
          "stdDev" : 0.17041289765265377
        }
      }
```

A raw cross-entity aggregate:

```
{
        "timestamp" : "2014-03-26T00:50:15.725Z",
        "value" : 3288.0,
        "type" : "SAMPLE",
        "aggregateStatistics" : {
          "sampleTime" : "2014-03-26T00:49:19.000Z",
          "sampleValue" : 7232.0,
          "count" : 4,
          "min" : 1600.0,
          "minTime" : "2014-03-26T00:49:42.000Z",
          "max" : 7232.0,
          "maxTime" : "2014-03-26T00:49:19.000Z",
          "mean" : 3288.0,
          "stdDev" : 2656.7549127961856,
          "crossEntityMetadata" : {
            "maxEntityDisplayName" : "cleroy-9-1.ent.cloudera.com",
            "minEntityDisplayName" : "cleroy-9-4.ent.cloudera.com",
            "numEntities" : 4.0
          }
        }
      }
```

A cross-time, cross-entity aggregate:

```
{
        "timestamp" : "2014-03-11T00:00:00.000Z",
        "value" : 3220.818863879957,
        "type" : "SAMPLE",
        "aggregateStatistics" : {
          "sampleTime" : "2014-03-10T22:28:48.000Z",
          "sampleValue" : 7200.0,
          "count" : 933,
          "min" : 1536.0,
          "minTime" : "2014-03-10T21:02:17.000Z",
          "max" : 7200.0,
          "maxTime" : "2014-03-10T22:28:48.000Z",
          "mean" : 3220.818863879957,
          "stdDev" : 2188.6143063503378,
          "crossEntityMetadata" : {
            "maxEntityDisplayName" : "cleroy-9-1.ent.cloudera.com",
            "minEntityDisplayName" : "cleroy-9-4.ent.cloudera.com",
```

```
            "numEntities" : 3.9787037037037036
        }
    }
}
```

These differ from non-aggregate data points by having the aggregateStatistics structure. Note that the value field in the point structure will always be the same as the aggregteStatistics mean field. The Cloudera Manager UI presents aggregate statistics in a number of ways. First, aggregate statistics are made available in the hover detail and chart popover when dealing with aggregate data. Second, it is possible to turn on and turn off the display of minimum and maximum time-series streams in line charts of aggregate data. These streams are displayed using dotted lines and give a visual indication of the underlying metric values data range over the time considered, entities considered or both. These lines are displayed by default for single stream line charts of aggregate data. For all line charts this behavior can be turned on and turned off using the chart popover.
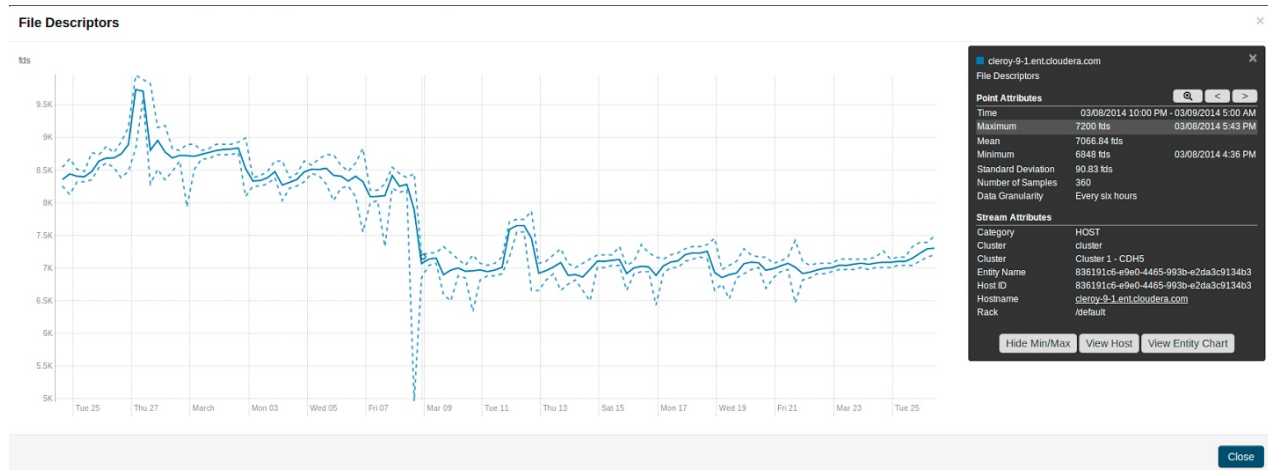
Accessing Aggregate Statistics Through tsquery

The stats function can be used to access aggregate statistics directly in tsquery. For example, `select stats(fd_open_across_datanodes, max) where category = service and serviceDisplayName = "my-hdfs-service"` will return a single time-series stream containing the just the maximum statistic values from the `fd_open_across_datanodes` stream. The following statistics are available through the stats function: `min`, `max`, `avg`, `std_dev`, and `sample`. See tsquery Language for more details on the stats function.

Logs

The Logs page presents log information for Hadoop services, filtered by service, role, host, or search phrase as well log level (severity).

To configure logs, see Configuring Log Events on page 228.

Viewing Logs

1. Select **Diagnostics** > **Logs** on the top navigation bar.
2. Click **Search**.

The logs for all roles display. If any of the hosts cannot be searched, an error message notifies you of the error and the host(s) on which it occurred.

Logs List

Log results are displayed in a list with the following columns:

- **Host** - The host where this log entry appeared. Clicking this link will take you to the Host Status page (see Host Details on page 244).

- **Log Level** - The log level (severity) associated with this log entry.
- **Time** - The date and time this log entry was created.
- **Source** - The class that generated the message.
- **Message** - The message portion of the log entry. Clicking **View Log File** displays the Log Details on page 328 page, which presents a display of the full log, showing the selected message (highlighted) and the 100 messages before and after it in the log.

If there are more results than can be shown on one page (per the Results per Page setting you selected), **Next** and **Prev** buttons let you view additional results.

Filtering Logs

You filter logs by selecting a time range and specifying filter parameters.

You can use the Time Range Selector or a duration link (30m 1h 2h 6h 12h 1d 7d 30d) to set the time range. (See Time Line on page 215 for details). However, logs are, by definition, historical, and are meaningful only in that context. So the Time Marker, used to pinpoint status at a specific point in time, is not available on this page. The Now button ( ) is available.

1. Specify any of the log filter parameters:

   - **Search Phrase** - A string to match against the log message content. The search is case-insensitive, and the string can be a regular expression, such that wildcards and other regular expression primitives are supported.
   - **Select Sources** - A list of all the service instances and roles currently instantiated in your cluster. By default, all services and roles are selected to be included in your log search; the All Sources checkbox lets you select or clear all services and roles in one operation. You can expand each service and limit the search to specific roles by selecting or clearing individual roles.
   - **Hosts** - The hosts to be included in the search. As soon as you start typing a hostname, Cloudera Manager provides a list of hosts that match the partial name. You can add multiple names, separated by commas. The default is to search all hosts.
   - **Minimum Log Level** - The minimum severity level for messages to be included in the search results. Results include all log entries at the selected level or higher. This defaults to WARN (that is, a search will return log entries with severity of WARN, ERROR, or FATAL only.
   - **Additional Settings**

     - **Search Timeout** - A time (in seconds) after which the search will time out. The default is 20 seconds.
     - **Results per Page** - The number of results (log entries) to be displayed per page.

2. Click **Search**. The Logs list displays the log entries that match the specified filter.

Log Details

The Log Details page presents a portion of the full log, showing the selected message (highlighted), and messages before and after it in the log. The page shows you:

- The host
- The role
- The full path and name of the log file you are viewing.
- Messages before and after the one you selected.

The log displays the following information for each message:

- Time - the time the entry was logged
- Log Level - the severity of the entry
- Source - the source class that logged the entry
- Log Message

You can switch to display only messages or all columns using the buttons.

In addition, from the Log Details page you can:

- View the log entries in either expanded or contracted form using the buttons to the left of the date range at the top of the log.
- Download the full log using the **Download Full Log** button at the top right of the page.
- View log details for a different host or for a different role on the current host, by clicking the **Change...** link next to the host or role at the top of the page. In either case this shows a pop-up where you can select the role or host you want to see.

Viewing the Cloudera Manager Server Log

To help you troubleshoot problems, you can view the Cloudera Manager Server log. You can view the logs in the Logs page or in specific pages for the log.

Viewing Cloudera Manager Server Logs in the Logs Page

1. Select **Diagnostics** > **Logs** on the top navigation bar.
2. Click **Select Sources** to display the log source list.
3. Uncheck the **All Sources** checkbox.
4. Click ▶ to the left of Cloudera Manager and select the **Server** checkbox.
5. Click **Search**.

For more information about the Logs page, see Logs on page 327.

Viewing the Cloudera Manager Server Log

1. Select **Diagnostics** > **Server Log** on the top navigation bar.

> **Note:** You can also view the Cloudera Manager Server log at
> `/var/log/cloudera-scm-server/cloudera-scm-server.log` on the Server host.

Viewing the Cloudera Manager Agent Logs

To help you troubleshoot problems, you can view the Cloudera Manager Agent logs. You can view the logs in the Logs page or in specific pages for the logs.

Viewing Cloudera Manager Agent Logs in the Logs Page

1. Select **Diagnostics** > **Logs** on the top navigation bar.
2. Click **Select Sources** to display the log source list.
3. Uncheck the **All Sources** checkbox.
4. Click ▶ to the left of Cloudera Manager and select the **Agent** checkbox.
5. Click **Search**.

For more information about the Logs page, see Logs on page 327.

Viewing the Cloudera Manager Agent Log

1. Click the **Hosts** tab.
2. Click the link for the host where you want to see the Agent log.
3. In the **Details** panel, click the **Details** link in the **Host Agent** field.
4. Click the **Agent Log** link.

You can also view the Cloudera Manager Agent log at `/var/log/cloudera-scm-agent/cloudera-scm-agent.log` on the Agent hosts.

Managing Disk Space for Log Files

All CDH cluster hosts write out separate log files for each role instance assigned to the host. Cluster administrators can monitor and manage the disk space used by these roles and configure log rotation to prevent log files from consuming too much disk space.

Disk Space Requirements

For each role assigned to a host, you should generally provision 2GB of disk space for log files. This recommendation is based on the default values of configuration properties that set the maximum log file size (200MB) and the maximum number of files (10). To calculate the disk space required for each host, multiply the configured maximum size of the log file by the configured maximum number of logs. Perform this calculation for each role on a host and add them together. (Note that Gateway roles do not generate log files.)

To determine the roles assigned to each host, open the Cloudera Manager Admin Console and go to **Hosts** > **All Hosts** and expand the list of roles in the **Roles** column.

Managing Log Files

To manage log file configurations for all role instances of a service:

1. Go to *Service Name* > **Configuration**.
2. Select **Category** > **Logs**.
3. Edit the logging parameters.
4. Click **Save Changes**.

> **Note:** You can also manage these configurations using role groups, which you can use to configure similar hosts with the same configuration values. See Managing Roles on page 209.

There are the parameters you use to manage log files:

Table 14: Log File Properties

Property	Description	Default Value
Role Type **Max Log Size**	Maximum size for a log file before the log file rolls over into a new file.	200 MB
Role Type **Maximum Log File Backups**	Maximum number of rolled-over log files to retain.	10
Role Type **Log Directory**	The path to the directory where the log files are saved.	`/var/log/log_file_name`
Role Type **Logging Threshold** (not available for all roles)	Logging level to limit the number of entries saved in the log file.	Depends on the role.

Reports

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See Managing Licenses on page 37 for more information.

The **Reports** page lets you create reports about the usage of HDFS in your cluster—data size and file count by user, group, or directory. It also lets you report on the MapReduce activity in your cluster, by user.

To display the **Reports** page, select **Clusters** > *Cluster name* > **Reports**.

For users with the Administrator role, the **Search Files and Manage Directories** button on the **Reports** page opens a file browser for searching files, managing directories, and setting quotas.

If you are managing multiple clusters, or have multiple nameservices configured (if high availability or federation is configured) there will be separate reports for each cluster and nameservice.

Directory Usage Report

Minimum Required Role: <u>BDR Administrator</u> (also provided by **Full Administrator** and **Cluster Administrator**)

The directory usage report allows you to browse the HDFS filesystem in a way that is similar to the HDFS <u>File Browser</u> . However, the **Directory Usage Report** also allows you to sort the listings and select multiple items and perform actions on them. You can also view the last access time, the last modified time of any file in a directory, and the total size of all files in the directory. This usage information is updated on an hourly basis.

You can customize the report by adding filters. A number of preconfigured filters are available, and you can create a custom filter.

Accessing the Directory Usage Report

To view the directory usage report, click **Clusters** > *Cluster Name* > **Reports** > **Directory Usage**.

You can also access this report from the HDFS File Browser. (To access the File Browser, click **Clusters** > *HDFS service* > **File Browser**.) Click the **Directory Usage** link located in the lower-right portion of the File Browser.

Using the Directory Usage Report

When you first open the report, the top level of the HDFS filesystem displays:

Directory Usage (Reports , Cluster 1 , HDFS-1)   The file system image was last indexed on March 10, 2016 10:47 AM

/ Edit

Filters ▾

Actions for Selected ▾

		▴ Name	⇕ Owner	⇕ Group	⇕ Permission	⇕ Last Access	⇕ Last Modified	⇕	Size	⇕	Raw Size / Quota	⇕	File and Directory Count / Quota
☐	👁 📁	/	hdfs	supergroup	drwxr-xr-x	9:47 AM	5:27 AM		546.4 MiB		1.6 GiB / -		1.1K / 9.2E
☐	👁 📁	hbase	hbase	hbase	drwxr-xr-x	9:42 AM	5:27 AM		3.5 KiB		10.6 KiB / -		42 / -
☐	👁 📁	solr	solr	solr	drwxrwxr-x	-	5:27 AM		0 B		0 B / -		0 / -
☐	👁 📁	tmp	hdfs	supergroup	drwxrwxrwx	9:47 AM	5:47 AM		14 B		42 B / -		16 / -
☐	👁 📁	user	hdfs	supergroup	drwxr-xr-x	5:43 AM	5:47 AM		546.4 MiB		1.6 GiB / -		1.1K / -

Display 50 ⬍ Per Page | <<   <   **1 - 5**   >

Directories highlighted with the 👁 icon in the first column are indexed and usage data is included in the <u>**Current Disk Usage By Directory**</u> and <u>**Historical Disk Usage By Directory**</u> reports.

Click the **Reports** link next to the **Directory Usage** title to go back to the **Reports** menu. You can also click links to go to the cluster and HDFS service home pages.

Click any column header to sort the display.

Click a directory name to view the files and subdirectories in the directory.

Select one or more rows by checking the boxes on the left and then choose an action to perform on the selection from the **Actions for Selected** drop-down menu. You can select the following actions:

- **Manage Quota** – A dialog box opens in which you can set a quota for the number of files or disk space. These values are displayed in columns in the file listing.
- **Include selected directories in disk usage reports** – The selected directories appear in the [Disk Usage Reports](#).
- **Exclude selected directories from disk usage reports** – The selected directories do not appear in the [Disk Usage Reports](#).

Filters

You can use filters to limit the display and to search for files. To apply filters to the directory usage report, click the **Filters** drop-down menu near the top of the page and select one of the following preconfigured filters:
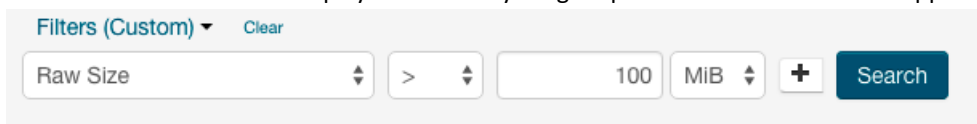
- Large Files
- Large Directories
- By Specific Owner
- By Specific Group
- Old Files
- Old Directories
- Files with Low Replication
- Overpopulated Directories
- Directories with Quotas
- Directories Watched

To modify any of these filters, click the **Customize** link and select new criteria. Click **Clear** to revert to the preconfigured criteria for the filter.

Click the **Search** button to display the report with the filters applied.

You can also select **Custom** from the **Filters** drop-down menu to create a report in which you define the criteria. To create a custom report:

1. Select any of the following criteria from the drop-down menu on the left:

   - Filename
   - Owner
   - Group
   - Path
   - Last Modified
   - Size
   - Diskspace Quota
   - Namespace Quota
   - Last Access
   - File and Directory Count
   - Replication
   - Parent
   - Raw Size

2. Select an operator from the drop-down menu.
3. Enter a value and units of measure for the comparison.
4. Select the units of measure for the comparison from the drop-down menu. (Some criteria do not require units of measure.)
5. Click the ✚ icon to add additional criteria.
6. Click the **Search** button to display the directory usage report with the custom filter applied.

The report changes to display the result of applying the filter. A new column, **Parent** is added that contains the full path to each file or subdirectory.

Disk Usage Reports

There are two types of disk usage reports: **Current Disk Usage By Directory** and **Historical Disk Usage By Directory**. To use these reports, select one or more directories to watch by clicking the 👁 icon for the directory. You can also select multiple directories, and then click **Actions for Selected** > **Include selected directories in disk usage reports**.

For information on using and configuring the Disk Usage Report, see Disk Usage Reports on page 333.

Disk Usage Reports

The following reports show HDFS disk usage statistics, either current or historical, by user, group, or directory.

The **By Directory** reports display information about the directories in the Watched list, so if you are not watching any directories there will be no results found for these reports. You can also specify which directories to watch by selecting them from the Directory Usage Report on page 331.

Viewing Current Disk Usage by User, Group, or Directory

These reports show "current" disk usage in both chart and tabular form. The data for these reports comes from the `fsimage` kept on the NameNode, so the data in a report will be only as current as when the last checkpoint was performed. Typically the checkpoint interval is (by default) once per hour, but if checkpoints are not being performed as frequently, the disk usage report may not be up to date. The disk usage report displays the current usage and does not account for deleted files that only exist in snapshots. These files are included in the usage information when you run the `du` command.

To create a disk usage report:

- Click the report name (link) to produce the resulting report.

Each of these reports show:

Bytes	The logical number of bytes in the files, aggregated by user, group, or directory. This is based on the actual files sizes, not taking replication into account.
Raw Bytes	The physical number of bytes (total disk space in HDFS) used by the files aggregated by user, group, or directory. This does include replication, and so is actually Bytes times the number of replicas.
File and Directory Count	The number of files aggregated by user, group, or directory.

Bytes and Raw Bytes are shown in IEC binary prefix notation (1 GiB = 1 * 2^{30}).

The directories shown in the **Current Disk Usage by Directory** report are the HDFS directories you have set as watched directories. You can add or remove directories to or from the watch list from this report; click the **Search Files and Manage Directories** button at the top right of the set of reports for the cluster or nameservice (see Designating Directories to Include in Disk Usage Reports on page 335).

The report data is also shown in chart format:

- Move the cursor over the graph to highlight a specific period on the graph and see the actual value (data size) for that period.
- You can also move the cursor over the user, group, or directory name (in the graph legend) to highlight the portion of the graph for that name.
- You can right-click within the chart area to save the whole chart display as a single image (a .PNG file) or as a PDF file. You can also print to the printer configured for your browser.

Viewing Historical Disk Usage by User, Group, or Directory

You can use these reports to view disk usage over a time range you define. You can have the usage statistics reported per hour, day, week, month, or year.

To create one of these reports:

- Click the report name (link) to produce the initial report. This generates a report that shows Raw Bytes for the past month, aggregated daily.

To change the report parameters:

- Select the **Start Date** and **End Date** to define the time range of the report.
- Select the **Graph Metric** you want to graph: bytes, raw bytes, or files and directories count.
- In the **Report Period** field, select the period over which you want the metrics aggregated. The default is Daily. This affects both the number of rows in the results table, and the granularity of the data points on the graph.
- Click **Generate Report** to produce a new report.

As with the current reports, the report data is also presented in chart format, and you can use the cursor to view the data shown on the charts, as well as save and print them.

For weekly or monthly reports, the Date indicates the date on which disk usage was measured.

The directories shown in the **Historical Disk Usage by Directory** report are the HDFS directories you have set as watched directories (see Designating Directories to Include in Disk Usage Reports on page 335).

Downloading Reports as CSV and XLS Files

Any report can be downloaded to your local system as an XLS file (Microsoft Excel 97-2003 worksheet) or CSV (comma-separated value) text file.

To download a report, do one of the following:

- From the main page of the Report tab, click CSV or XLS link next to in the column to the right of the report name
- From any report page, click the **Download CSV** or **Download XLS** buttons.

Either of these opens the Open file dialog box where you can open or save the file locally.

Activity, Application, and Query Reports

The Reports page contains links for displaying metrics on the following types of activities in your cluster:

- Disk usage
- MapReduce jobs
- YARN applications
- Impala queries
- HBase tables and namespaces

To view the Reports page, click **Clusters** > *ClusterName* > **Reports**. You can generate a report to view aggregate job activity per hour, day, week, month, or year, by user or for all users.

1. Click the **Start Date** and **End Date** fields and choose a date from the date control.
2. In the **Report Period** drop-down, select the period over which you want the metrics aggregated. Default is Daily.
3. Click **Generate Report**.

For weekly reports, the Date column indicates the year and week number (for example, 2013-01 through 2013-52). For monthly reports, the Date column indicates the year and month by number (2013-01 through 2013-12).

The File Browser

Minimum Required Role: BDR Administrator (also provided by **Full Administrator** and **Cluster Administrator**)

The **File Browser** tab on the HDFS service page lets you browse and search the HDFS namespace and manage your files and directories. The File Browser page initially displays the root directory of the HDFS file system in the gray panel

at the top and its immediate subdirectories below. Click any directory to drill down into the contents of that directory or to select that directory for available actions.

Searching Within the File System

To search the file system, click **Custom report** in the **Reports** section. The **Choose** drop down lets you select from custom search criteria such as filename, owner, file size, and so on. The file and directory listings are taken from the `fsimage` stored on the NameNode, so the listings will be only as current as the last checkpoint. Typically the checkpoint interval is (by default) once per hour, but if checkpoints are not being performed as frequently, the listings may not be up to date.

To search the file system:

1. From the HDFS service page, select the **File Browser** tab.
2. Click **Choose** and do one of the following:

   • Select a predefined query. Depending on what you select, you may be presented with different fields to fill in or different views of the file system. For example, selecting **Size** will provide a choice of arithmetic operators and fields where you provide the size to be used as the search criteria.

     1. Select a property in the **Choose...** drop-down.
     2. Select an operator.
     3. Specify a value.
     4. Click ✚ to add another criteria (all of which must be satisfied for a file to be considered a match) and repeat the preceding steps.

3. Click the **Generate Report** button to generate a custom report containing the search results.

If you search within a directory, only files within that directory will be found. For example, if you browse `/user` and do a search, you might find `/user/foo/file`, but you will not find `/bar/baz`.

Enabling Snapshots

To enable snapshots for an HDFS directory and its contents, see <u>Managing HDFS Snapshots</u> on page 543.

Setting Quotas

To set quotas for an HDFS directory and its contents, see <u>Setting HDFS Quotas</u> on page 109.

Designating Directories to Include in Disk Usage Reports

1. To add or remove directories from the directory-based Disk Usage reports, navigate through the file system to see the directory you want to add. You can include a directory at any level without including its parent.
2. Check the checkbox **Include this directory in Disk Usage reports**. As long as the checkbox is checked, the directory appears in the usage reports. To discontinue inclusion of the directory in Disk Usage reports, clear the checkbox.

Downloading HDFS Directory Access Permission Reports

The Directory Access By Group feature in the User Access category on the Reports page is a Cloudera data management feature.

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

For each HDFS service, you can download a report that details the HDFS directories a group has permission to access.

1. In the Cloudera Manager Admin Console, click **Clusters** > **ClusterName** > **Reports**.
2. In the Directory Access by Group row, click **CSV** or **XLS**. The Download User Access Report pop-up displays.

   a. In the pop-up, type a group and directory.
   b. Click **Download**. A report of the selected type will be generated containing the following information – path, owner, permissions, and size – for each directory contained in the specified directory that the specified group has access to.

Troubleshooting Cluster Configuration and Operation

This section contains solutions to some common problems that prevent you from using Cloudera Manager and describes how to use Cloudera Manager log and notification management tools to diagnose problems.

Solutions to Common Problems

Symptom	Reason	Solution
Cloudera Manager		
The Cloudera Manager service will not be running as it exited abnormally. Running `service cloudera-scm-server status` will print following message "cloudera-scm-server dead but pid file exists". The Cloudera Manager Server log file /var/log/cloudera-scm-server/cloudera-scm-server.log will have a stacktrace with "`java.lang.OutOfMemoryError`" logged.	Out of memory.	Examine the heap dump that the Cloudera Manager Server creates when it runs out of memory. The heap dump file is created in the `/tmp` directory, has file extension `.hprof` and file permission of 600. Its owner and group will be the owner and group of the Cloudera Manager server process, normally `cloudera-scm:cloudera-scm`.
You are unable to start service on the Cloudera Manager server, that is, `service cloudera-scm-server start` does not work and there are errors in the log file located at /var/log/cloudera-scm-server/cloudera-scm-server.log	The server has been disconnected from the database or the database has stopped responding or has shut down.	Go to `/etc/cloudera-scm-server/db.properties` and make sure the database you are trying to connect to is listed there and has been started.
Logs include `APPARENT DEADLOCK` entries for c3p0.	These deadlock messages are cause by the c3p0 process not making progress at the expected rate. This can indicate either that c3p0 is deadlocked or that its progress is slow enough to trigger these messages. In many cases, progress is occurring and these messages should not be seen as catastrophic.	There are a variety of ways to react to these log entries. • You may ignore these messages if system performance is not otherwise affected. Because these entries often occur during slow progress, they may be ignored in some cases. • You may modify the timer triggers. If c3p0 is making slow progress, increasing the period of time during which progress is evaluated stop the log entries from occurring. The default time between Timer triggers is 10 seconds and is configurable indirectly by configuring `maxAdministrativeTaskTime`. For more information, see maxAdministrativeTaskTime. • You may increase the number of threads in the c3p0 pool, thereby increasing the resources available to make progress on tasks. For more information, see numHelperThreads.
Starting Services		

Symptom	Reason	Solution
After you click the **Start** button to start a service, the **Finished** status does not display. This may not be merely a case of the status not getting displayed. It could be for a number of reasons such as network connectivity issues or subcommand failures.	The host is disconnected from the Server, as will be indicated by missing heartbeats on the **Hosts** tab.	• Look at the logs for the service for causes of the problem. • Restart the Agents on the hosts where the heartbeats are missing.
	Subcommands failed resulting in errors in the log file indicating that either the command timed out or the target port was already occupied	• Look at the log file at `/var/log/cloudera-scm-server/cloudera-scm-server.log` for more details on the errors. For example, if the port is already occupied you should see an "Address in use" error. • Go to the **Hosts > Status** tab. Click the **Name** of the host you want to inspect. Now go to the **Processes** tab and check the **Stdout/Stderr** logs to diagnose the cause of the failure. For example, if any binaries are missing or if Java could not be found.
After you click **Start** to start a service, the **Finished** status displays but there are error messages. The subcommands to start service components (such as JobTracker and one or more TaskTrackers) do not start.	A port specified in the **Configuration** tab of the service is already being used in your cluster. For example, the JobTracker port is in use by another process.	Enter an available port number in the port property (such as **JobTracker port**) in the **Configuration** tab of the service.
	There are incorrect directories specified in the **Configuration** tab of the service (such as the log directory).	Enter correct directories in the **Configuration** tab of the service.
Job is Failing	No space left on device.	One approach is to use a system monitoring tool such as Nagios to alert on the disk space or quickly check disk space across all systems. If you do not have Nagios or equivalent you can do the following to determine the source of the space issue: In the JobTracker Web UI, drill down from the job, to the map or reduce, to the task attempt details to see which TaskTracker the task executed and failed on due to disk space. For example: `http://JTHost:50030/taskdetails.jsp?tipid=TaskID`. You can see on which host the task is failing in the Machine column. In the NameNode Web UI, inspect the % used column on the NameNode Live Nodes page: `http://namenode:50070/dfsnodelist.jsp?whatNodes=LIVE`.
Send Test Alert and Diagnose SMTP Errors		

Symptom	Reason	Solution
You have enabled sending alerts from the Cloudera Manager Admin Console, however, Cloudera Manager does not seem to be sending any alerts. Using the **Send Test Alert** link under **Administration** > **Alerts** shows success even though you do not receive an alert email.	There is possibly a mismatch of protocol or port numbers between your mail server and the Alert Publisher. For example, if the Alert Publisher is sending alerts to SMTPS on port 465 and your mail servers are not configured for SMTPS, you wouldn't receive any alerts.	Use the following steps to make changes to the Alert Publisher configuration: 1. In the Cloudera Manager Admin Console, click the Cloudera Management Service. 2. Click the **Configuration** tab. 3. Select **Scope** > **Alert Publisher**. 4. Click the **Main** category. 5. Change **Alerts: Mail Server Protocol** to `smtp` (or `smtps`). 6. Click the **Ports and Addresses** category and change **Alerts: Mail Server TCP Port** to `25` (or to `465` for SMTPS) 7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes. 8. Restart the Alert Publisher.

Logs and Events

For information about problems, check the logs and events:

- Logs on page 327 present log information for services, filtered by role, host, or keywords as well log level (severity).
- Viewing the Cloudera Manager Server Log on page 329 contains information on the server and host agents.
- The Events tab lets you search for and display events and alerts that have occurred within a selected time range filtered by service, hosts, or keywords.

Monitoring Reference

Performance Management

This section describes mechanisms and best practices for improving performance.

Related Information

- [Tuning Impala for Performance](#)

Optimizing Performance in CDH

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

This section provides solutions to some performance problems, and describes configuration best practices.

> **Important:** Work with your network administrators and hardware vendors to ensure that you have the proper NIC firmware, drivers, and configurations in place and that your network performs properly. Cloudera recognizes that network setup and upgrade are challenging problems, and will do its best to share useful experiences.

Disable the tuned Service

If your cluster hosts are running RHEL/CentOS 7.x, disable the "tuned" service by running the following commands:

1. Ensure that the tuned service is started:

```
systemctl start tuned
```

2. Turn the tuned service off:

```
tuned-adm off
```

3. Ensure that there are no active profiles:

```
tuned-adm list
```

The output should contain the following line:

```
No current active profile
```

4. Shutdown and disable the tuned service:

```
systemctl stop tuned
systemctl disable tuned
```

Disabling Transparent Hugepages (THP)

Most Linux platforms supported by CDH include a feature called **_transparent hugepages_**, which interacts poorly with Hadoop workloads and can seriously degrade performance.

Symptom: `top` and other system monitoring tools show a large percentage of the CPU usage classified as "system CPU". If system CPU usage is 30% or more of the total CPU usage, your system may be experiencing this issue.

To see whether transparent hugepages are enabled, run the following commands and check the output:

```
$ cat defrag_file_pathname
$ cat enabled_file_pathname
```

- [always] never means that transparent hugepages is enabled.
- always [never] means that transparent hugepages is disabled.

To disable Transparent Hugepages, perform the following steps on all cluster hosts:

1. (**Required for hosts running RHEL/CentOS 7.x.**) To disable transparent hugepages on reboot, add the following commands to the /etc/rc.d/rc.local file on all cluster hosts:

   - **RHEL/CentOS 7.x:**

   ```
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
   echo never > /sys/kernel/mm/transparent_hugepage/defrag
   ```

   - **RHEL/CentOS 6.x**

   ```
   echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
   echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
   ```

   - **Ubuntu/Debian, OL, SLES**:

   ```
   echo never > /sys/kernel/mm/transparent_hugepage/defrag
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
   ```

   Modify the permissions of the rc.local file:

   ```
   chmod +x /etc/rc.d/rc.local
   ```

2. If your cluster hosts are running RHEL/CentOS 7.x, modify the GRUB configuration to disable THP:

   - Add the following line to the GRUB_CMDLINE_LINUX options in the /etc/default/grub file:

   ```
   transparent_hugepage=never
   ```

   - Run the following command:

   ```
   grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

3. <u>Disable the tuned service</u>, as described above.

You can also disable transparent hugepages interactively (but remember this will not survive a reboot).

To disable transparent hugepages temporarily as root:

```
# echo 'never' > defrag_file_pathname
# echo 'never' > enabled_file_pathname
```

To disable transparent hugepages temporarily using sudo:

```
$ sudo sh -c "echo 'never' > defrag_file_pathname"
$ sudo sh -c "echo 'never' > enabled_file_pathname"
```

Setting the vm.swappiness Linux Kernel Parameter

The Linux kernel parameter, vm.swappiness, is a value from 0-100 that controls the swapping of application data (as anonymous pages) from physical memory to virtual memory on disk. The higher the value, the more aggressively

inactive processes are swapped out from physical memory. The lower the value, the less they are swapped, forcing filesystem buffers to be emptied.

On most systems, `vm.swappiness` is set to 60 by default. This is not suitable for Hadoop clusters because processes are sometimes swapped even when enough memory is available. This can cause lengthy garbage collection pauses for important system daemons, affecting stability and performance.

Cloudera recommends that you set `vm.swappiness` to a value between 1 and 10, **preferably 0**, for minimum swapping on systems where the RHEL kernel is 2.6.32-642.el6 or higher.

To view your current setting for `vm.swappiness`, run:

```
cat /proc/sys/vm/swappiness
```

To set `vm.swappiness` to `1`, run:

```
sudo sysctl -w vm.swappiness=1
```

Improving Performance in Shuffle Handler and IFile Reader

The MapReduce shuffle handler and IFile reader use native Linux calls, (`posix_fadvise`(2) and `sync_data_range`), on Linux systems with Hadoop native libraries installed.

Shuffle Handler

You can improve MapReduce shuffle handler performance by enabling shuffle readahead. This causes the TaskTracker or Node Manager to pre-fetch map output before sending it over the socket to the reducer.

- To enable this feature for YARN, set `mapreduce.shuffle.manage.os.cache`, to `true` (default). To further tune performance, adjust the value of `mapreduce.shuffle.readahead.bytes`. The default value is 4 MB.
- To enable this feature for MapReduce, set the `mapred.tasktracker.shuffle.fadvise` to `true` (default). To further tune performance, adjust the value of `mapred.tasktracker.shuffle.readahead.bytes`. The default value is 4 MB.

IFile Reader

Enabling IFile readahead increases the performance of merge operations. To enable this feature for either MRv1 or YARN, set `mapreduce.ifile.readahead` to `true` (default). To further tune the performance, adjust the value of `mapreduce.ifile.readahead.bytes`. The default value is 4MB.

Best Practices for MapReduce Configuration

The configuration settings described below can reduce inherent latencies in MapReduce execution. You set these values in `mapred-site.xml`.

Send a heartbeat as soon as a task finishes

Set `mapreduce.tasktracker.outofband.heartbeat` to `true` for TaskTracker to send an out-of-band heartbeat on task completion to reduce latency. The default value is `false`:

```
<property>
    <name>mapreduce.tasktracker.outofband.heartbeat</name>
    <value>true</value>
</property>
```

Reduce the interval for JobClient status reports on single node systems

The `jobclient.progress.monitor.poll.interval` property defines the interval (in milliseconds) at which JobClient reports status to the console and checks for job completion. The default value is 1000 milliseconds; you may

want to set this to a lower value to make tests run faster on a single-node cluster. Adjusting this value on a large production cluster may lead to unwanted client-server traffic.

```
<property>
    <name>jobclient.progress.monitor.poll.interval</name>
    <value>10</value>
</property>
```

Tune the JobTracker heartbeat interval

Tuning the minimum interval for the TaskTracker-to-JobTracker heartbeat to a smaller value may improve MapReduce performance on small clusters.

```
<property>
    <name>mapreduce.jobtracker.heartbeat.interval.min</name>
    <value>10</value>
</property>
```

Start MapReduce JVMs immediately

The `mapred.reduce.slowstart.completed.maps` property specifies the proportion of Map tasks in a job that must be completed before any Reduce tasks are scheduled. For small jobs that require fast turnaround, setting this value to 0 can improve performance; larger values (as high as 50%) may be appropriate for larger jobs.

```
<property>
    <name>mapred.reduce.slowstart.completed.maps</name>
    <value>0</value>
</property>
```

Tips and Best Practices for Jobs

This section describes changes you can make at the job level.

Use the Distributed Cache to Transfer the Job JAR

Use the distributed cache to transfer the job JAR rather than using the `JobConf(Class)` constructor and the `JobConf.setJar()` and `JobConf.setJarByClass()` methods.

To add JARs to the classpath, use `-libjars jar1,jar2`. This copies the local JAR files to HDFS and uses the distributed cache mechanism to ensure they are available on the task nodes and added to the task classpath.

The advantage of this, over `JobConf.setJar`, is that if the JAR is on a task node, it does not need to be copied again if a second task from the same job runs on that node, though it will still need to be copied from the launch machine to HDFS.

> **Note:** `-libjars` works only if your MapReduce driver uses [ToolRunner](#). If it does not, you would need to use the DistributedCache APIs (Cloudera does not recommend this).

For more information, see item 1 in the blog post [How to Include Third-Party Libraries in Your MapReduce Job](#).

Changing the Logging Level on a Job (MRv1)

You can change the logging level for an individual job. You do this by setting the following properties in the job configuration ([JobConf](#)):

- `mapreduce.map.log.level`
- `mapreduce.reduce.log.level`

Valid values are `NONE`, `INFO`, `WARN`, `DEBUG`, `TRACE`, and `ALL`.

Example:

```
JobConf conf = new JobConf();
...

conf.set("mapreduce.map.log.level", "DEBUG");
conf.set("mapreduce.reduce.log.level", "TRACE");
...
```

Decrease Reserve Space

By default, the `ext3` and `ext4` filesystems reserve 5% space for use by the root user. This reserved space counts as `Non DFS Used`. To view the reserved space use the `tune2fs` command:

```
# tune2fs -l /dev/sde1 | egrep "Block size:|Reserved block count"
Reserved block count:   36628312
Block size:             4096
```

The Reserved block count is the number of `ext3`/`ext4` filesystem blocks that are reserved. The block size is the size in bytes. In this example, 150 GB (139.72 Gigabytes) are reserved on this filesystem.

Cloudera recommends reducing the root user block reservation from 5% to 1% for the DataNode volumes. To set reserved space to 1% with the `tune2fs` command:

```
# tune2fs -m 1 /dev/sde1
```

Choosing and Configuring Data Compression

For an overview of compression, see [Data Compression](#).

Guidelines for Choosing a Compression Type

- GZIP compression uses more CPU resources than Snappy or LZO, but provides a higher compression ratio. GZip is often a good choice for *cold data*, which is accessed infrequently. Snappy or LZO are a better choice for *hot data*, which is accessed frequently.
- BZip2 can also produce more compression than GZip for some types of files, at the cost of some speed when compressing and decompressing. HBase does not support BZip2 compression.
- Snappy often performs better than LZO. It is worth running tests to see if you detect a significant difference.
- For MapReduce and Spark, if you need your compressed data to be splittable, BZip2 and LZO formats can be split. Snappy and GZip blocks are not splittable, but files with Snappy blocks inside a container file format such as SequenceFile or Avro can be split. Snappy is intended to be used with a container format, like SequenceFiles or Avro data files, rather than being used directly on plain text, for example, since the latter is not splittable and cannot be processed in parallel. Splittability is not relevant to HBase data.
- For MapReduce, you can compress either the intermediate data, the output, or both. Adjust the parameters you provide for the MapReduce job accordingly. The following examples compress both the intermediate data and the output. MR2 is shown first, followed by MR1.

    – **MRv2**

```
hadoop jar hadoop-examples-.jar sort "-Dmapreduce.compress.map.output=true"
"-Dmapreduce.map.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec"
"-Dmapreduce.output.compress=true"
"-Dmapreduce.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input output
```

    – **MRv1**

```
hadoop jar hadoop-examples-.jar sort "-Dmapred.compress.map.output=true"
"-Dmapred.map.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec"
```

```
"-Dmapred.output.compress=true"
"-Dmapred.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input output
```

Configuring Data Compression

To configure support for LZO using Cloudera Manager, you must install the GPL Extras parcel, then configure services to use it. See [Installing the GPL Extras Parcel](#) and [Configuring Services to Use the GPL Extras Parcel](#) on page 168.

Tuning the Solr Server

Solr performance tuning is a complex task. The following sections provide more details.

Setting Java System Properties for Solr

Several of the following sections refer to Java system properties. These properties are set differently depending on whether or not you are using Cloudera Manager.

To set Java system properties for Solr in Cloudera Manager:

1. **Solr service** > **Configuration** > **Category** > **Advanced**
2. Add the property to **Java Configuration Options for Solr Server** using the format `-D<property_name>=<value>`. For example, to set `solr.hdfs.blockcache.slab.count` to `100`, add the following:

```
-Dsolr.hdfs.blockcache.slab.count=100
```

Garbage collection options, such as `-XX:+PrintGCTimeStamps`, can also be set here. Use spaces to separate multiple parameters.

To set Java system properties in unmanaged environments:

Add or modify the `JAVA_OPTS` environment variable in `/etc/default/solr`. For example:

```
JAVA_OPTS="-Xmx10g -XX:MaxDirectMemorySize=20g \
-XX:+UseLargePages -Dsolr.hdfs.blockcache.slab.count=100" \
-XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:+PrintGCDetails
```

Tuning to Complete During Setup

Some tuning is best completed during the setup of you system or may require some re-indexing.

Configuring Lucene Version Requirements

You can configure Solr to use a specific version of Lucene. This can help ensure that the Lucene version that Search uses includes the latest features and bug fixes. At the time that a version of Solr ships, Solr is typically configured to use the appropriate Lucene version, in which case there is no need to change this setting. If a subsequent Lucene update occurs, you can configure the Lucene version requirements by directly editing the `luceneMatchVersion` element in the `solrconfig.xml` file. Versions are typically of the form `x.y`, such as `4.4`. For example, to specify version 4.4, you would ensure the following setting exists in `solrconfig.xml`:

```
<luceneMatchVersion>4.4</luceneMatchVersion>
```

Designing the Schema

When constructing a schema, use data types that most accurately describe the data that the fields will contain. For example:

- Use the `tdate` type for dates. Do this instead of representing dates as strings.

- Consider using the `text` type that applies to your language, instead of using `String`. For example, you might use `text_en`. `Text` types support returning results for subsets of an entry. For example, querying on "john" would find "John Smith", whereas with the string type, only exact matches are returned.
- For IDs, use the string type.

Configuring the Heap Size

Set the Java heap size for the Solr Server to at least 16 GB for production environments. For more information on memory requirements, see [Guidelines for Deploying Cloudera Search](#).

General Tuning

The following tuning categories can be completed at any time. It is less important to implement these changes before beginning to use your system.

General Tips

- Enabling multi-threaded faceting can provide better performance for field faceting. When multi-threaded faceting is enabled, field faceting tasks are completed in a parallel with a thread working on every field faceting task simultaneously. Performance improvements do not occur in all cases, but improvements are likely when all of the following are true:
  - The system uses highly concurrent hardware.
  - Faceting operations apply to large data sets over multiple fields.
  - There is not an unusually high number of queries occurring simultaneously on the system. Systems that are lightly loaded or that are mainly engaged with ingestion and indexing may be helped by multi-threaded faceting; for example, a system ingesting articles and being queried by a researcher. Systems heavily loaded by user queries are less likely to be helped by multi-threaded faceting; for example, an e-commerce site with heavy user-traffic.

  > **Note:** Multi-threaded faceting only applies to field faceting and not to query faceting.
  >
  > - Field faceting identifies the number of unique entries for a field. For example, multi-threaded faceting could be used to simultaneously facet for the number of unique entries for the fields, "color" and "size". In such a case, there would be two threads, and each thread would work on faceting one of the two fields.
  > - Query faceting identifies the number of unique entries that match a query for a field. For example, query faceting could be used to find the number of unique entries in the "size" field are between 1 and 5. Multi-threaded faceting does not apply to these operations.

  To enable multi-threaded faceting, add `facet-threads` to queries. For example, to use up to 1000 threads, you might use a query as follows:

  ```
  http://localhost:8983/solr/collection1/select?q=*:*&facet=true&fl=id&facet.field=f0_ws&facet.threads=1000
  ```

  If `facet-threads` is omitted or set to 0, faceting is single-threaded. If `facet-threads` is set to a negative value, such as -1, multi-threaded faceting will use as many threads as there are fields to facet up to the maximum number of threads possible on the system.

- If your environment does not require Near Real Time (NRT), turn off soft auto-commit in `solrconfig.xml`.
- In most cases, do not change the default [batchSize](#) setting of 1000. If you are working with especially large documents, you may consider decreasing the batch size.
- To help identify any garbage collector (GC) issues, enable GC logging in production. The overhead is low and the JVM supports GC log rolling as of 1.6.0_34.
  - The minimum recommended GC logging flags are: `-XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:+PrintGCDetails`.

- – To rotate the GC logs: `-Xloggc: -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles= -XX:GCLogFileSize=.`

    For Cloudera Manager environments, you can set these flags at **Solr service** > **Configuration** > **Category** > **Java Configuration Options for Solr Server**.

    For unmanaged environments, you can configure Java options by adding or modifying the `JAVA_OPTS` environment variable in `/etc/default/solr`:

```
JAVA_OPTS="-Xmx10g -XX:MaxDirectMemorySize=20g \
-XX:+UseLargePages -Dsolr.hdfs.blockcache.slab.count=100" \
-XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:+PrintGCDetails
```

Solr and HDFS - the Block Cache

⚠️ **Warning:** Do not enable the Solr HDFS write cache, because it can lead to index corruption.

Cloudera Search enables Solr to store indexes in an HDFS filesystem. To maintain performance, an HDFS block cache has been implemented using Least Recently Used (LRU) semantics. This enables Solr to cache HDFS index files on read and write, storing the portions of the file in JVM direct memory (off heap) by default, or optionally in the JVM heap.

Batch jobs typically do not use the cache, while Solr servers (when serving queries or indexing documents) should. When running indexing using MapReduce, the MR jobs themselves do not use the block cache. Block write caching is turned off by default and should be left disabled.

Tuning of this cache is complex and best practices are continually being refined. In general, allocate a cache that is about 10-20% of the amount of memory available on the system. For example, when running HDFS and Solr on a host with 96 GB of memory, allocate 10-20 GB of memory using `solr.hdfs.blockcache.slab.count`. As index sizes grow you may need to tune this parameter to maintain optimal performance.

📝 **Note:** Block cache metrics are currently unavailable.

Configuration

The following parameters control caching. They can be configured at the Solr process level by setting the respective Java system property or by editing `solrconfig.xml` directly. For more information on setting Java system properties, see Setting Java System Properties for Solr on page 344.

If the parameters are set at the collection level (using `solrconfig.xml`), the first collection loaded by the Solr server takes precedence, and block cache settings in all other collections are ignored. Because you cannot control the order in which collections are loaded, you must make sure to set identical block cache settings in every collection `solrconfig.xml`. Block cache parameters set at the collection level in `solrconfig.xml` also take precedence over parameters at the process level.

Parameter	Cloudera Manager Setting	Default	Description
`solr.hdfs.blockcache.global`	Not directly configurable. Cloudera Manager automatically enables the global block cache. To override this setting, you must	true	If enabled, one HDFS block cache is used for each collection on a host. If `blockcache.global` is disabled, each SolrCore on a host creates its own private HDFS block cache. Enabling this parameter simplifies managing HDFS block cache memory.

Parameter	Cloudera Manager Setting	Default	Description
	use the **Solr Service Environment Advanced Configuration Snippet (Safety Valve)**.		
solr.hdfs.blockcache.enabled	**HDFS Block Cache**	true	Enable the block cache.
solr.hdfs.blockcache.read.enabled	Not directly configurable. If the block cache is enabled, Cloudera Manager automatically enables the read cache. To override this setting, you must use the **Solr Service Environment Advanced Configuration Snippet (Safety Valve)**.	true	Enable the read cache.
solr.hdfs.blockcache.write.enabled	Not directly configurable. If the block cache is enabled, Cloudera Manager automatically disables the write cache. ⚠ **Warning:** Do not enable the Solr HDFS write cache, because it can lead to [index corruption]().	false	Enable the write cache.

Parameter	Cloudera Manager Setting	Default	Description
`solr.hdfs.blockcache.direct.memory.allocation`	**HDFS Block Cache Off-Heap Memory**	true	Enable direct memory allocation. If this is false, heap is used.
`solr.hdfs.blockcache.blocksperbank`	**HDFS Block Cache Blocks per Slab**	16384	Number of blocks per cache slab. The size of the cache is 8 KB (the block size) times the number of blocks per slab times the number of slabs.
`solr.hdfs.blockcache.slab.count`	**HDFS Block Cache Number of Slabs**	1	Number of slabs per block cache. The size of the cache is 8 KB (the block size) times the number of blocks per slab times the number of slabs.

> **Note:**
>
> Increasing the direct memory cache size may make it necessary to increase the maximum direct memory size allowed by the JVM. Each Solr slab allocates memory, which is 128 MB by default, as well as allocating some additional direct memory overhead. Therefore, ensure that the `MaxDirectMemorySize` is set comfortably above the value expected for slabs alone. The amount of additional memory required varies according to multiple factors, but for most cases, setting `MaxDirectMemorySize` to at least 20-30% more than the total memory configured for slabs is sufficient. Setting `MaxDirectMemorySize` to the number of slabs multiplied by the slab size does not provide enough memory.
>
> **To set `MaxDirectMemorySize` using Cloudera Manager**:
>
> 1. Go to the Solr service.
> 2. Click the **Configuration** tab.
> 3. In the Search box, type **Java Direct Memory Size of Solr Server in Bytes**.
> 4. Set the new direct memory value.
> 5. Restart Solr servers after editing the parameter.
>
> **To set `MaxDirectMemorySize` in unmanaged environments**:
>
> 1. Add `-XX:MaxDirectMemorySize=20g` to the `JAVA_OPTS` environment variable in `/etc/default/solr`.
> 2. Restart Solr servers:
>
> ```
> sudo service solr-server restart
> ```

Solr HDFS optimizes caching when performing NRT indexing using Lucene's `NRTCachingDirectory`.

Lucene caches a newly created segment if both of the following conditions are true:

- The segment is the result of a flush or a merge and the estimated size of the merged segment is <= `solr.hdfs.nrtcachingdirectory.maxmergesizemb`.
- The total cached bytes is <= `solr.hdfs.nrtcachingdirectory.maxcachedmb`.

The following parameters control NRT caching behavior:

Parameter	Default	Description
`solr.hdfs.nrtcachingdirectory.enable`	true	Whether to enable the NRTCachingDirectory.
`solr.hdfs.nrtcachingdirectory.maxcachedmb`	192	Size of the cache in megabytes.

Parameter	Default	Description
`solr.hdfs.nrtcachingdirectory.maxmergesizemb`	16	Maximum segment size to cache.

Here is an example of `solrconfig.xml` with defaults:

```
<directoryFactory name="DirectoryFactory">
   <bool name="solr.hdfs.blockcache.enabled">${solr.hdfs.blockcache.enabled:true}</bool>

   <int name="solr.hdfs.blockcache.slab.count">${solr.hdfs.blockcache.slab.count:1}</int>

   <bool
name="solr.hdfs.blockcache.direct.memory.allocation">${solr.hdfs.blockcache.direct.memory.allocation:true}</bool>

   <int
name="solr.hdfs.blockcache.blocksperbank">${solr.hdfs.blockcache.blocksperbank:16384}</int>

   <bool
name="solr.hdfs.blockcache.read.enabled">${solr.hdfs.blockcache.read.enabled:true}</bool>

   <bool
name="solr.hdfs.nrtcachingdirectory.enable">${solr.hdfs.nrtcachingdirectory.enable:true}</bool>

   <int
name="solr.hdfs.nrtcachingdirectory.maxmergesizemb">${solr.hdfs.nrtcachingdirectory.maxmergesizemb:16}</int>

   <int
name="solr.hdfs.nrtcachingdirectory.maxcachedmb">${solr.hdfs.nrtcachingdirectory.maxcachedmb:192}</int>
</directoryFactory>
```

The following example illustrates passing Java options by editing the `/etc/default/solr` or `/opt/cloudera/parcels/CDH-*/etc/default/solr` configuration file:

```
JAVA_OPTS="-Xmx10g -XX:MaxDirectMemorySize=20g -XX:+UseLargePages \
-Dsolr.hdfs.blockcache.slab.count=100"
```

For better performance, Cloudera recommends setting the Linux swap space on all Solr server hosts as shown below:

- Minimize swappiness:

```
sudo sysctl vm.swappiness=1
```

- Disable swap space until next reboot:

```
sudo swapoff -a
```

Garbage Collection

Choose different garbage collection options for best performance in different environments. Some garbage collection options typically chosen include:

- **Concurrent low pause collector**: Use this collector in most cases. This collector attempts to minimize "Stop the World" events. Avoiding these events can reduce connection timeouts, such as with ZooKeeper, and may improve user experience. This collector is enabled using the Java system property `-XX:+UseConcMarkSweepGC`.
- **Throughput collector**: Consider this collector if raw throughput is more important than user experience. This collector typically uses more "Stop the World" events so this may negatively affect user experience and connection timeouts such as ZooKeeper heartbeats. This collector is enabled using the Java system property `-XX:+UseParallelGC`. If `UseParallelGC` "Stop the World" events create problems, such as ZooKeeper timeouts, consider using the `UseParNewGC` collector as an alternative collector with similar throughput benefits.

For information on setting Java system properties, see Setting Java System Properties for Solr on page 344.

You can also affect garbage collection behavior by increasing the Eden space to accommodate new objects. With additional Eden space, garbage collection does not need to run as frequently on new objects.

You can adjust the degree to which different data is replicated.

Replication Settings

> ✎ **Note:** Do not adjust HDFS replication settings for Solr in most cases.

To adjust the Solr replication factor for index files stored in HDFS:

- **Cloudera Manager:**

    1. Go to **Solr service** > **Configuration** > **Category** > **Advanced**.
    2. Click the plus sign next to **Solr Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** to add a new property with the following values:

        Name: `dfs.replication`

        Value: 2

    3. Click **Save Changes**.
    4. Restart the Solr service (**Solr service** > **Actions** > **Restart**).

- **Unmanaged:**

    1. Configure the `solr.hdfs.confdir` system property to refer to the Solr HDFS configuration files. Typically the value is `/etc/solrhdfs/`. For information on setting Java system properties, see Setting Java System Properties for Solr on page 344.
    2. Set the DFS replication value in the HDFS configuration file at the location you specified in the previous step. For example, to set the replication value to `2`, you would change the `dfs.replication` setting as follows:

```
<property>
  <name>dfs.replication<name>
  <value>2<value>
<property>
```

    3. Restart the Solr service:

```
sudo service solr-server restart
```

Replicas

If you have sufficient additional hardware, add more replicas for a linear boost of query throughput. Note that adding replicas may slow write performance on the first replica, but otherwise this should have minimal negative consequences.

Transaction Log Replication

Beginning with CDH 5.4.1, Search supports configurable transaction log replication levels for replication logs stored in HDFS. Cloudera recommends leaving the value unchanged at 3 or, barring that, setting it to at least 2.

Configure the transaction log replication factor for a collection by modifying the `tlogDfsReplication` setting in `solrconfig.xml`. The `tlogDfsReplication` is a new setting in the `updateLog` settings area. An excerpt of the `solrconfig.xml` file where the transaction log replication factor is set is as follows:

```
<updateHandler class="solr.DirectUpdateHandler2">

    <!-- Enables a transaction log, used for real-time get, durability, and
         and solr cloud replica recovery.  The log can grow as big as
         uncommitted changes to the index, so use of a hard autoCommit
         is recommended (see below).
         "dir" - the target directory for transaction logs, defaults to the
                 solr data directory.  -->
```

```
<updateLog>
  <str name="dir">${solr.ulog.dir:}</str>
  <int name="tlogDfsReplication">${solr.ulog.tlogDfsReplication:3}</int>
  <int name="numVersionBuckets">${solr.ulog.numVersionBuckets:65536}</int>
</updateLog>
```

The default replication level is 3. For clusters with fewer than three DataNodes (such as proof-of-concept clusters), reduce this number to the amount of DataNodes in the cluster. Changing the replication level only applies to new transaction logs.

Initial testing shows no significant performance regression for common use cases.

Shards

In some cases, oversharding can help improve performance including intake speed. If your environment includes massively parallel hardware and you want to use these available resources, consider oversharding. You might increase the number of replicas per host from 1 to 2 or 3. Making such changes creates complex interactions, so you should continue to monitor your system's performance to ensure that the benefits of oversharding do not outweigh the costs.

Commits

Changing commit values may improve performance in some situation. These changes result in tradeoffs and may not be beneficial in all cases.

- For hard commit values, the default value of 60000 (60 seconds) is typically effective, though changing this value to 120 seconds may improve performance in some cases. Note that setting this value to higher values, such as 600 seconds may result in undesirable performance tradeoffs.
- Consider increasing the auto-commit value from 15000 (15 seconds) to 120000 (120 seconds).
- Enable soft commits and set the value to the largest value that meets your requirements. The default value of 1000 (1 second) is too aggressive for some environments.

Other Resources

- General information on Solr caching is available on the [Query Settings in SolrConfig](#) page in the Solr Reference Guide.
- Information on issues that influence performance is available on the [SolrPerformanceFactors](#) page on the Solr Wiki.
- [Resource Management](#) describes how to use Cloudera Manager to manage resources, for example with Linux cgroups.
- For information on improving querying performance, see [How to make searching faster](#).
- For information on improving indexing performance, see [How to make indexing faster](#).

Tuning Apache Spark Applications

This topic describes various aspects in tuning the performance and scalability of Apache Spark applications. For general Spark tuning advice, consult in the upstream Spark documentation. This topic focuses on performance aspects that are especially relevant when using Spark in the context of CDH clusters.

During tuning, monitor application behavior to determine the effect of tuning actions. You might see improvements that are directly relevant to the performance of your job, such as reduction in CPU usage, or reductions in resource usage that improve overall scalability within a multi-tenant cluster.

For information on monitoring Spark applications, see [Monitoring Spark Applications](#) on page 279 .

Tuning Spark Shuffle Operations

A Spark dataset comprises a fixed number of partitions, each of which comprises a number of records. For the datasets returned by *narrow* transformations, such as `map` and `filter`, the records required to compute the records in a single partition reside in a *single partition* in the parent dataset. Each object is only dependent on a single object in the parent.

Operations such as `coalesce` can result in a task processing multiple input partitions, but the transformation is still considered narrow because the input records used to compute any single output record can still only reside in a limited subset of the partitions.

Spark also supports transformations with **wide** dependencies, such as `groupByKey` and `reduceByKey`. In these dependencies, the data required to compute the records in a single partition can reside in *many partitions* of the parent dataset. To perform these transformations, all of the tuples with the same key must end up in the same partition, processed by the same task. To satisfy this requirement, Spark performs a *shuffle*, which transfers data around the cluster and results in a new [stage](#) with a new set of partitions.

For example, consider the following code:

```
sc.textFile("someFile.txt").map(mapFunc).flatMap(flatMapFunc).filter(filterFunc).count()
```

It runs a single action, `count`, which depends on a sequence of three transformations on a dataset derived from a text file. This code runs in a single stage, because none of the outputs of these three transformations depend on data that comes from different partitions than their inputs.

In contrast, this Scala code finds how many times each character appears in all the words that appear more than 1,000 times in a text file:

```
val tokenized = sc.textFile(args(0)).flatMap(_.split(' '))
val wordCounts = tokenized.map((_, 1)).reduceByKey(_ + _)
val filtered = wordCounts.filter(_._2 >= 1000)
val charCounts = filtered.flatMap(_._1.toCharArray).map((_, 1)).reduceByKey(_ + _)
charCounts.collect()
```

This example has three stages. The two `reduceByKey` transformations each trigger stage boundaries, because computing their outputs requires repartitioning the data by keys.

A final example is this more complicated transformation graph, which includes a `join` transformation with multiple dependencies:
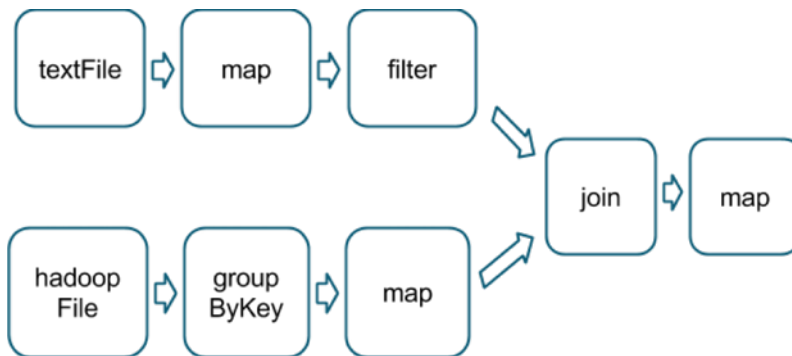
The pink boxes show the resulting stage graph used to run it:
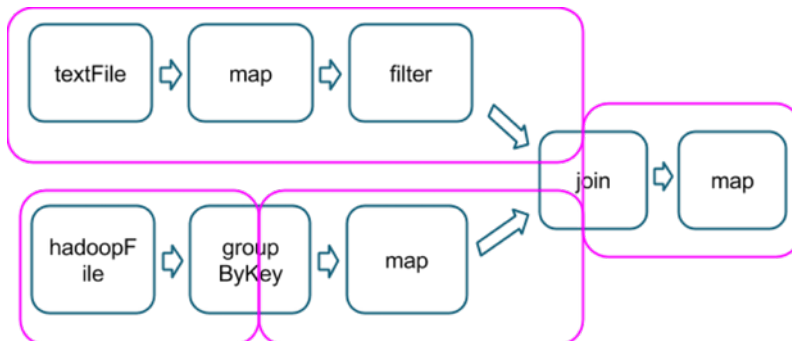
At each stage boundary, data is written to disk by tasks in the parent stages and then fetched over the network by tasks in the child stage. Because they incur high disk and network I/O, stage boundaries can be expensive and should be avoided when possible. The number of data partitions in a parent stage may be different than the number of partitions in a child stage. Transformations that can trigger a stage boundary typically accept a `numPartitions` argument, which specifies into how many partitions to split the data in the child stage. Just as the number of reducers is an important parameter in MapReduce jobs, the number of partitions at stage boundaries can determine an application's performance. describes how to tune this number.

Choosing Transformations to Minimize Shuffles

You can usually choose from many arrangements of actions and transformations that produce the same results. However, not all these arrangements result in the same performance. Avoiding common pitfalls and picking the right arrangement can significantly improve an application's performance.

When choosing an arrangement of transformations, minimize the number of shuffles and the amount of data shuffled. Shuffles are expensive operations; all shuffle data must be written to disk and then transferred over the network. `repartition`, `join`, `cogroup`, and any of the `*By` or `*ByKey` transformations can result in shuffles. Not all these transformations are equal, however, and you should avoid the following patterns:

- `groupByKey` when performing an associative reductive operation. For example, `rdd.groupByKey().mapValues(_.sum)` produces the same result as `rdd.reduceByKey(_ + _)`. However, the former transfers the entire dataset across the network, while the latter computes local sums for each key in each partition and combines those local sums into larger sums after shuffling.

- `reduceByKey` when the input and output value types are *different*. For example, consider writing a transformation that finds all the unique strings corresponding to each key. You could use `map` to transform each element into a `Set` and then combine the `Set`s with `reduceByKey`:

```
rdd.map(kv => (kv._1, new Set[String]() + kv._2)).reduceByKey(_ ++ _)
```

    This results in unnecessary object creation because a new set must be allocated for each record.

    Instead, use `aggregateByKey`, which performs the map-side aggregation more efficiently:

```
val zero = new collection.mutable.Set[String]()
rdd.aggregateByKey(zero)((set, v) => set += v,(set1, set2) => set1 ++= set2)
```

  - `flatMap-join-groupBy`. When two datasets are already grouped by key and you want to join them and keep them grouped, use `cogroup`. This avoids the overhead associated with unpacking and repacking the groups.

When Shuffles Do Not Occur

In some circumstances, the transformations described previously *do not* result in shuffles. Spark does not shuffle when a previous transformation has already partitioned the data according to the *same partitioner*. Consider the following flow:

```
rdd1 = someRdd.reduceByKey(...)
rdd2 = someOtherRdd.reduceByKey(...)
rdd3 = rdd1.join(rdd2)
```

Because no partitioner is passed to `reduceByKey`, the default partitioner is used, resulting in `rdd1` and `rdd2` both being hash-partitioned. These two `reduceByKey` transformations result in two shuffles. If the datasets have the same number of partitions, a join requires no additional shuffling. Because the datasets are partitioned identically, the set of keys in any single partition of `rdd1` can only occur in a single partition of `rdd2`. Therefore, the contents of any single output partition of `rdd3` depends only on the contents of a single partition in `rdd1` and single partition in `rdd2`, and a third shuffle is not required.

For example, if `someRdd` has four partitions, `someOtherRdd` has two partitions, and both the `reduceByKey`s use three partitions, the set of tasks that run would look like this:
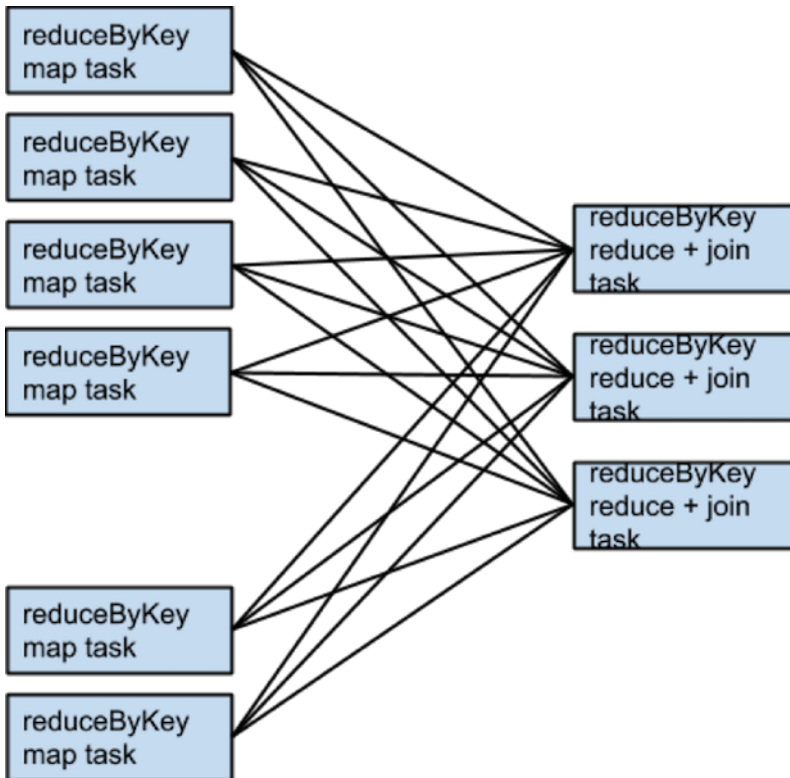
If `rdd1` and `rdd2` use different partitioners or use the default (hash) partitioner with different numbers of partitions, only one of the datasets (the one with the fewer number of partitions) needs to be reshuffled for the join:
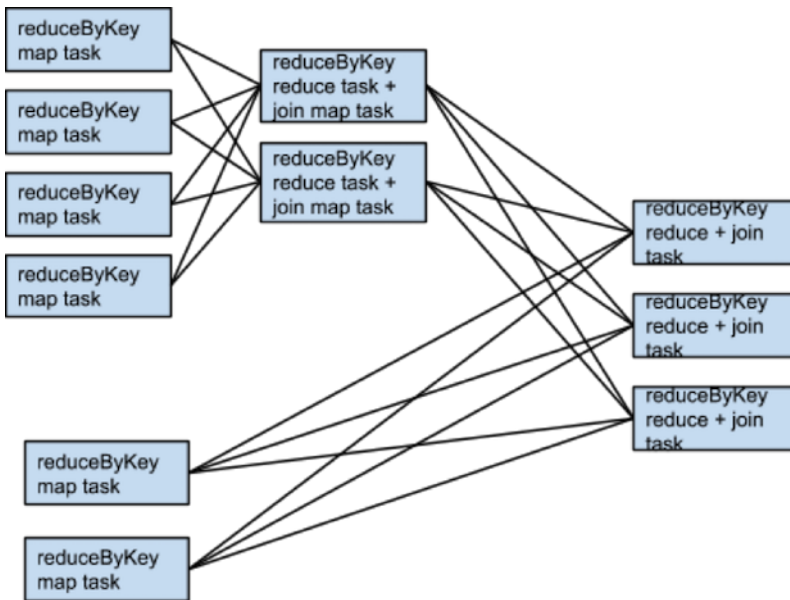
To avoid shuffles when joining two datasets, you can use broadcast variables. When one of the datasets is small enough to fit in memory in a single executor, it can be loaded into a hash table on the driver and then broadcast to every executor. A map transformation can then reference the hash table to do lookups.

When to Add a Shuffle Transformation

The rule of minimizing the number of shuffles has some exceptions.

An extra shuffle can be advantageous when it increases parallelism. For example, if your data arrives in a few large unsplittable files, the partitioning dictated by the `InputFormat` might place large numbers of records in each partition,

while not generating enough partitions to use all available cores. In this case, invoking repartition with a high number of partitions (which triggers a shuffle) after loading the data allows the transformations that follow to use more of the cluster's CPU.

Another example arises when using the `reduce` or `aggregate` action to aggregate data into the driver. When aggregating over a high number of partitions, the computation can quickly become bottlenecked on a single thread in the driver merging all the results together. To lighten the load on the driver, first use `reduceByKey` or `aggregateByKey` to perform a round of distributed aggregation that divides the dataset into a smaller number of partitions. The values in each partition are merged with each other in parallel, before being sent to the driver for a final round of aggregation. See `treeReduce` and `treeAggregate` for examples of how to do that.

This method is especially useful when the aggregation is already grouped by a key. For example, consider an application that counts the occurrences of each word in a corpus and pulls the results into the driver as a map. One approach, which can be accomplished with the `aggregate` action, is to compute a local map at each partition and then merge the maps at the driver. The alternative approach, which can be accomplished with `aggregateByKey`, is to perform the count in a fully distributed way, and then simply `collectAsMap` the results to the driver.

Secondary Sort

The [repartitionAndSortWithinPartitions](#) transformation repartitions the dataset according to a partitioner and, within each resulting partition, sorts records by their keys. This transformation pushes sorting down into the shuffle machinery, where large amounts of data can be spilled efficiently and sorting can be combined with other operations.

For example, Apache Hive on Spark uses this transformation inside its `join` implementation. It also acts as a vital building block in the *secondary sort* pattern, in which you group records by key and then, when iterating over the values that correspond to a key, have them appear in a particular order. This scenario occurs in algorithms that need to group events by user and then analyze the events for each user, based on the time they occurred.

Tuning Resource Allocation

For background information on how Spark applications use the YARN cluster manager, see [Running Spark Applications on YARN](#).

The two main resources that Spark and YARN manage are CPU and memory. Disk and network I/O affect Spark performance as well, but neither Spark nor YARN actively manage them.

Every Spark executor in an application has the same fixed number of cores and same fixed heap size. Specify the number of cores with the `--executor-cores` command-line flag, or by setting the `spark.executor.cores` property. Similarly, control the heap size with the `--executor-memory` flag or the `spark.executor.memory` property. The `cores` property controls the number of concurrent tasks an executor can run. For example, set `--executor-cores` 5 for each executor to run a maximum of five tasks at the same time. The memory property controls the amount of data Spark can cache, as well as the maximum sizes of the shuffle data structures used for grouping, aggregations, and joins.

[Dynamic allocation](#), which adds and removes executors dynamically, is enabled by default. To explicitly control the number of executors, you can override dynamic allocation by setting the `--num-executors` command-line flag or `spark.executor.instances` configuration property.

Consider also how the resources requested by Spark fit into resources YARN has available. The relevant YARN properties are:

- `yarn.nodemanager.resource.memory-mb` controls the maximum sum of memory used by the containers on each host.
- `yarn.nodemanager.resource.cpu-vcores` controls the maximum sum of cores used by the containers on each host.

Requesting five executor cores results in a request to YARN for five cores. The memory requested from YARN is more complex for two reasons:

- The `--executor-memory`/`spark.executor.memory` property controls the executor heap size, but executors can also use some memory off heap, for example, Java NIO direct buffers. The value of the

`spark.yarn.executor.memoryOverhead` property is added to the executor memory to determine the full memory request to YARN for each executor. It defaults to `max`(384, .1 * `spark.executor.memory`).

- YARN may round the requested memory up slightly. The `yarn.scheduler.minimum-allocation-mb` and `yarn.scheduler.increment-allocation-mb` properties control the minimum and increment request values, respectively.

The following diagram (not to scale with defaults) shows the hierarchy of memory properties in Spark and YARN:
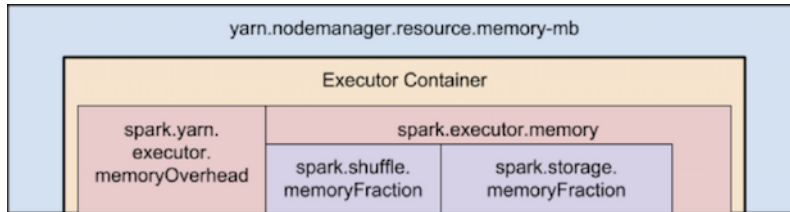
Keep the following in mind when sizing Spark executors:

- The ApplicationMaster, which is a non-executor container that can request containers from YARN, requires memory and CPU that must be accounted for. In **client** deployment mode, they default to 1024 MB and one core. In **cluster** deployment mode, the ApplicationMaster runs the driver, so consider bolstering its resources with the `--driver-memory` and `--driver-cores` flags.
- Running executors with too much memory often results in excessive garbage-collection delays. For a single executor, use 64 GB as an upper limit.
- The HDFS client has difficulty processing many concurrent threads. At most, five tasks per executor can achieve full write throughput, so keep the number of cores per executor below that number.
- Running tiny executors (with a single core and just enough memory needed to run a single task, for example) offsets the benefits of running multiple tasks in a single JVM. For example, broadcast variables must be replicated once on each executor, so many small executors results in many more copies of the data.

Resource Tuning Example

Consider a cluster with six hosts running NodeManagers, each equipped with 16 cores and 64 GB of memory.

The NodeManager capacities, `yarn.nodemanager.resource.memory-mb` and `yarn.nodemanager.resource.cpu-vcores`, should be set to 63 * 1024 = 64512 (megabytes) and 15, respectively. Avoid allocating 100% of the resources to YARN containers because the host needs some resources to run the OS and Hadoop daemons. In this case, leave one GB and one core for these system processes. Cloudera Manager accounts for these and configures these YARN properties automatically.

You might consider using `--num-executors 6 --executor-cores 15 --executor-memory 63G`. However, this approach does not work:

- 63 GB plus the executor memory overhead does not fit within the 63 GB capacity of the NodeManagers.
- The ApplicationMaster uses a core on one of the hosts, so there is no room for a 15-core executor on that host.
- 15 cores per executor can lead to bad HDFS I/O throughput.

Instead, use `--num-executors 17 --executor-cores 5 --executor-memory 19G`:

- This results in three executors on all hosts except for the one with the ApplicationMaster, which has two executors.
- `--executor-memory` is computed as (63/3 executors per host) = 21. 21 * 0.07 = 1.47. 21 - 1.47 ~ 19.

Tuning the Number of Partitions

Spark has limited capacity to determine optimal parallelism. Every Spark stage has a number of tasks, each of which processes data sequentially. The number of tasks per stage is the most important parameter in determining performance.

As described in Spark Execution Model, Spark groups datasets into stages. The number of tasks in a stage is the same as the number of partitions in the last dataset in the stage. The number of partitions in a dataset is the same as the number of partitions in the datasets on which it depends, with the following exceptions:

- The `coalesce` transformation creates a dataset with *fewer* partitions than its parent dataset.

- The `union` transformation creates a dataset with the *sum* of its parents' number of partitions.
- The `cartesian` transformation creates a dataset with the *product* of its parents' number of partitions.

Datasets with no parents, such as those produced by `textFile` or `hadoopFile`, have their partitions determined by the underlying MapReduce `InputFormat` used. Typically, there is a partition for each HDFS block being read. The number of partitions for datasets produced by `parallelize` are specified in the method, or `spark.default.parallelism` if not specified. To determine the number of partitions in an dataset, call `rdd.partitions().size()`.

If the number of tasks is smaller than number of slots available to run them, CPU usage is suboptimal. In addition, more memory is used by any aggregation operations that occur in each task. In `join`, `cogroup`, or `*ByKey` operations, objects are held in hashmaps or in-memory buffers to group or sort. `join`, `cogroup`, and `groupByKey` use these data structures in the tasks for the stages that are on the fetching side of the shuffles they trigger. `reduceByKey` and `aggregateByKey` use data structures in the tasks for the stages on both sides of the shuffles they trigger. If the records in these aggregation operations exceed memory, the following issues can occur:

- Increased garbage collection, which can lead to pauses in computation.
- Spilling data to disk, causing disk I/O and sorting, which leads to job stalls.

To increase the number of partitions if the stage is reading from Hadoop:

- Use the `repartition` transformation, which triggers a shuffle.
- Configure your `InputFormat` to create more splits.
- Write the input data to HDFS with a smaller block size.

If the stage is receiving input from another stage, the transformation that triggered the stage boundary accepts a `numPartitions` argument:

```
val rdd2 = rdd1.reduceByKey(_ + _, numPartitions = X)
```

Determining the optimal value for *X* requires experimentation. Find the number of partitions in the parent dataset, and then multiply that by 1.5 until performance stops improving.

You can also calculate *X* using a formula, but some quantities in the formula are difficult to calculate. The main goal is to run enough tasks so that the data destined for each task fits in the memory available to that task. The memory available to each task is:

```
(spark.executor.memory * spark.shuffle.memoryFraction * spark.shuffle.safetyFraction)/
spark.executor.cores
```

`memoryFraction` and `safetyFraction` default to 0.2 and 0.8 respectively.

The in-memory size of the total shuffle data is more difficult to determine. The closest heuristic is to find the ratio between shuffle spill memory and the shuffle spill disk for a stage that ran. Then, multiply the total shuffle write by this number. However, this can be compounded if the stage is performing a reduction:

```
(observed shuffle write) * (observed shuffle spill memory) * (spark.executor.cores)/
(observed shuffle spill disk) * (spark.executor.memory) * (spark.shuffle.memoryFraction)
 * (spark.shuffle.safetyFraction)
```

Then, round up slightly, because too many partitions is usually better than too few.

When in doubt, err on the side of a larger number of tasks (and thus partitions). This contrasts with recommendations for MapReduce, which unlike Spark, has a high startup overhead for tasks.

Reducing the Size of Data Structures

Data flows through Spark in the form of records. A record has two representations: a deserialized Java object representation and a serialized binary representation. In general, Spark uses the deserialized representation for records in memory and the serialized representation for records stored on disk or transferred over the network. For sort-based shuffles, in-memory shuffle data is stored in serialized form.

The `spark.serializer` property controls the serializer used to convert between these two representations. Cloudera recommends using the Kryo serializer, `org.apache.spark.serializer.KryoSerializer`.

The footprint of your records in these two representations has a significant impact on Spark performance. Review the data types that are passed and look for places to reduce their size. Large deserialized objects result in Spark spilling data to disk more often and reduces the number of deserialized records Spark can cache (for example, at the `MEMORY` storage level). The Apache Spark tuning guide describes how to [reduce the size of such objects](#). Large serialized objects result in greater disk and network I/O, as well as reduce the number of serialized records Spark can cache (for example, at the `MEMORY_SER` storage level.) Make sure to register any custom classes you use with the `SparkConf#registerKryoClasses` API.

Choosing Data Formats

When storing data on disk, use an extensible binary format like [Avro](#), [Parquet](#), Thrift, or Protobuf and store in a [sequence file](#).

Tuning YARN

This topic applies to YARN clusters only, and describes how to tune and optimize YARN for your cluster.
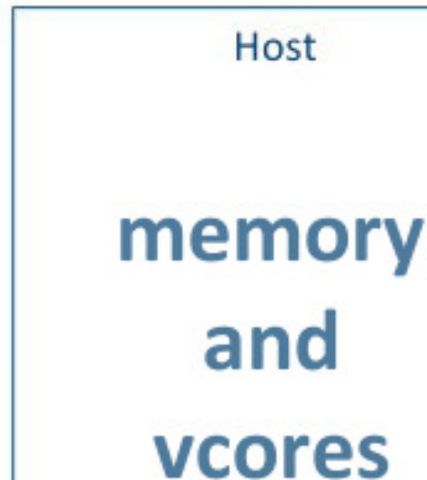
> **Note:** Download the Cloudera [YARN tuning spreadsheet](#) to help calculate YARN configurations. For a short video overview, see [Tuning YARN Applications](#).

Overview

This overview provides an abstract description of a YARN cluster and the goals of YARN tuning.

A YARN cluster is composed of host machines. Hosts provide memory and CPU resources. A *vcore*, or virtual core, is a usage share of a host CPU.

Tuning YARN consists primarily of optimally defining *containers* on your worker hosts. You can think of a container as a rectangular graph consisting of memory and vcores. Containers perform tasks.
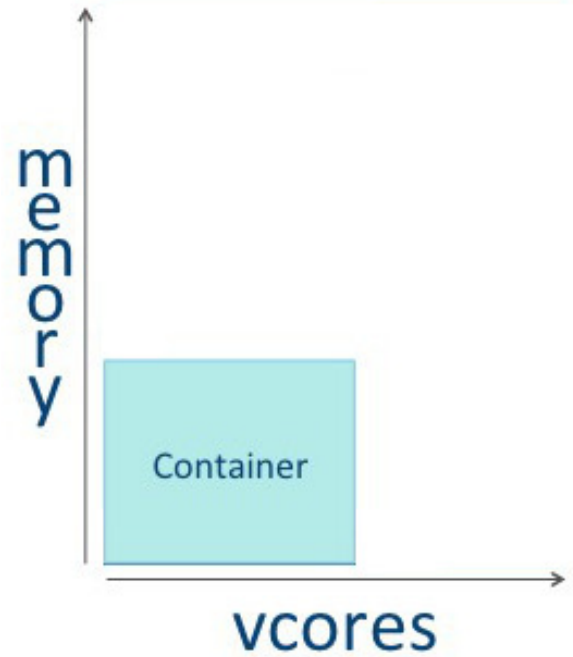
Some tasks use a great deal of memory, with minimal processing on a large volume of data.
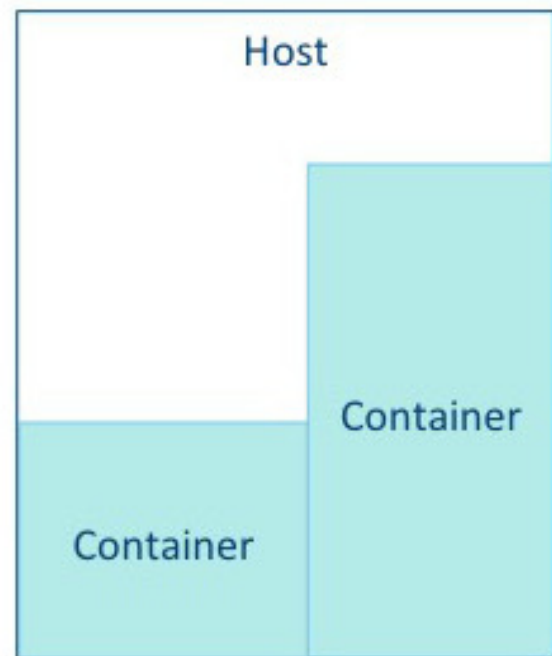
Other tasks require a great deal of processing power, but use less memory. For example, a Monte Carlo Simulation that evaluates many possible "what if?" scenarios uses a great deal of processing power on a relatively small dataset.
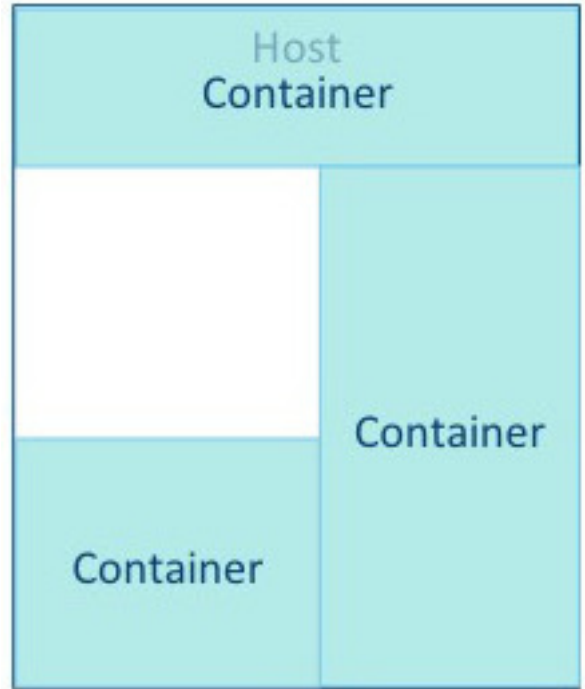
The YARN ResourceManager allocates memory and vcores to use all available resources in the most efficient way possible. Ideally, few or no resources are left idle.
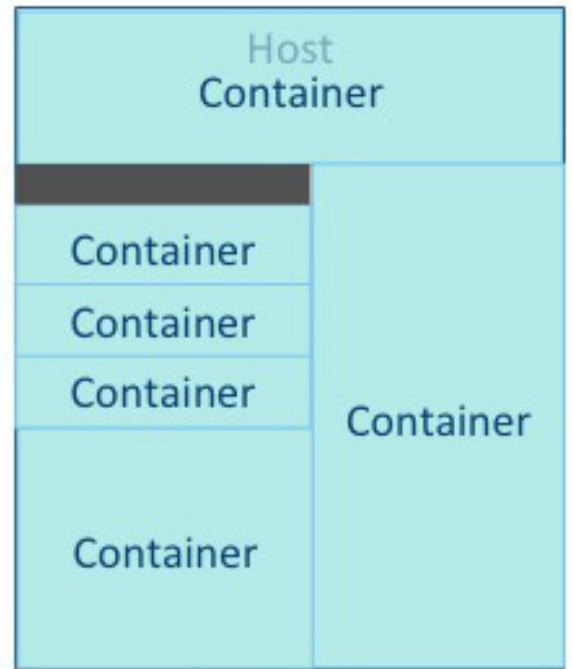
An *application* is a YARN client program consisting of one or more tasks. Typically, a task uses all of the available resources in the container. A task cannot consume more than its designated allocation, ensuring that it cannot use all of the host CPU cycles or exceed its memory allotment.	
Tune your YARN hosts to optimize the use of vcores and memory by configuring your containers to use all available resources beyond those required for overhead and other services.	
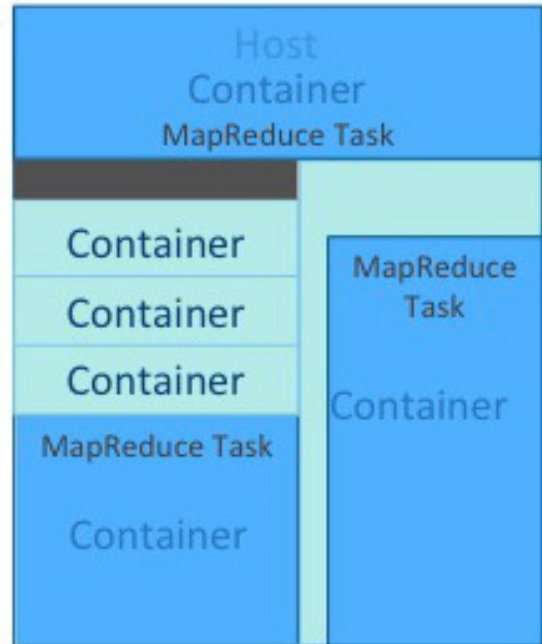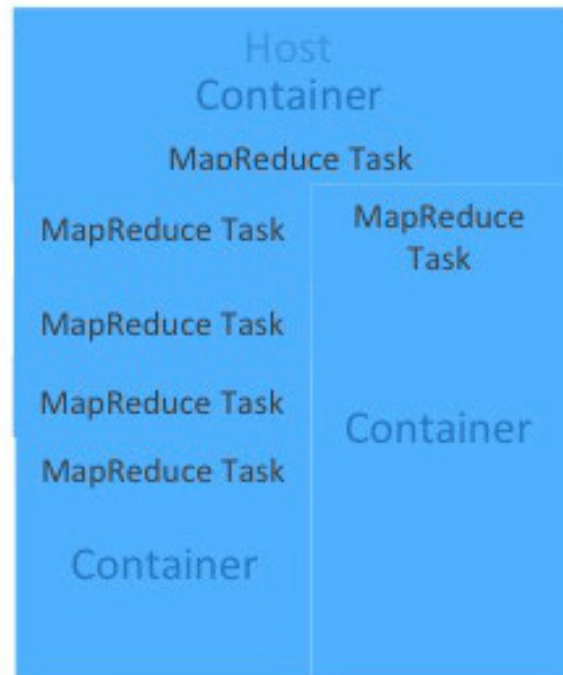
YARN tuning has three phases. The phases correspond to the tabs in the YARN tuning spreadsheet.

1. Cluster configuration, where you configure your hosts.
2. YARN configuration, where you quantify memory and vcores.
3. MapReduce configuration, where you allocate minimum and maximum resources for specific map and reduce tasks.

YARN and MapReduce have many configurable properties. For a complete list, see Cloudera Manager Configuration Properties. The YARN tuning spreadsheet lists the essential subset of these properties that are most likely to improve performance for common MapReduce applications.

Cluster Configuration

In the Cluster Configuration tab, you define the worker host configuration and cluster size for your YARN implementation.

Step 1: Worker Host Configuration

Step 1 is to define the configuration for a single worker host computer in your cluster.

STEP 1: Worker Host Configuration

Enter your likely machine configuration in the input boxes below. If you are uncertain what machines you plan on buying, put in some minimum values that will suit what you expect to buy. Last updated early 2016.

Host Components	Quantity	Description
RAM	256	Gigabytes
CPU	48	8 CPUs: 6 cores, 3.5 GHz, 15MB cache
HDD (Hard Disk Drive)	36	12x3TB SATA III Hard Drives in JBOD Configuration
Ethernet	2	1 Gigabit Ethernet

As with any system, the more memory and CPU resources available, the faster the cluster can process large amounts of data. A machine with 8 CPUs, each with 6 cores, provides 48 vcores per host.

3 TB hard drives in a 2-unit server installation with 12 available slots in JBOD (Just a Bunch Of Disks) configuration is a reasonable balance of performance and pricing at the time the spreadsheet was created. The cost of storage decreases over time, so you might consider 4 TB disks. Larger disks are expensive and not required for all use cases.

Two 1-Gigabit Ethernet ports provide sufficient throughput at the time the spreadsheet was published, but 10-Gigabit Ethernet ports are an option where price is of less concern than speed.

Step 2: Worker Host Planning

Step 2 is to allocate resources on each worker machine.

STEP 2: Worker Host Planning

Now that you have your base Host configuration from Step 1, use the table below to allocate resources, mainly CPU and memory, to the various software components that run on the host.

Service	Category	CPU (cores)	Memory (MB)	CM Static Service %	Notes
Operating System	Overhead	1	8192	N/A	Most operating systems use 4-8GB minimum.
Cloudera Manager agent	Overhead	1	1024	N/A	Allocate 1GB for Cloudera Manager agents, which track resource usage on a host.
Other services	Overhead	0	0	N/A	Enter the required cores or memory for services not listed above.
HDFS DataNode	CDH	1	1024	4	Allocate 1GB for the HDFS DataNode.
Impala daemon	CDH	0	0	0	(Optional Service) Suggestion: Allocate at least 16GB memory when using Impala.
Hbase RegionServer	CDH	0	0	0	(Optional Service) Suggestion: Allocate no more than 12-16GB memory when using HBase Region Servers.
Solr Server	CDH	0	0	0	(Optional Service) Suggestion: Minimum 1GB for Solr server. More will be necessary depending on index sizes.
YARN NodeManager	CDH	1	1024	N/A	Allocate 1GB for the YARN NodeManager.
Available Resources		44	250880		
Physical Cores to Vcores Multiplier		4			Set this ratio based on the expected number of concurrent threads per core. Use 1 for CPU intensive tasks up to 4 for standard I/O bound tasks.
YARN Available Vcores		176			This value will be used in STEP 4 for YARN Configuration
YARN Available Memory			250880		This value will be used in STEP 4 for YARN Configuration

Start with at least 8 GB for your operating system, and 1 GB for Cloudera Manager. If services outside of CDH require additional resources, add those numbers under Other Services.

The HDFS DataNode uses a minimum of 1 core and about 1 GB of memory. The same requirements apply to the YARN NodeManager.

The spreadsheet lists three optional services. For Impala, allocate at least 16 GB for the daemon. HBase RegionServer requires 12-16 GB of memory. Solr Server requires a minimum of 1 GB of memory.

Any remaining resources are available for YARN applications (Spark and MapReduce). In this example, 44 CPU cores are available. Set the multiplier for vcores you want on each physical core to calculate the total available vcores.

Step 3: Cluster Size

Having defined the specifications for each host in your cluster, enter the number of worker hosts needed to support your business case. To see the benefits of parallel computing, set the number of hosts to a minimum of 10.

STEP 3: Cluster Size

Enter the number of nodes you have (or expect to have) in the cluster

	Quantity
Number of Worker Hosts in the cluster	10

YARN Configuration

On the YARN Configuration tab, you verify your available resources and set minimum and maximum limits for each container.

Steps 4 and 5: Verify Settings

Step 4 pulls forward the memory and vcore numbers from step 2. Step 5 shows the total memory and vcores for the cluster.

STEP 4: YARN Configuration on Cluster

These are the first set of configuration values for your cluster. You can set these values in YARN->Configuration in Cloudera Manager.

YARN Configuration Property	Value	
yarn.nodemanager.resource.cpu-vcores	176	Copied from STEP 2 "Available Resources"
yarn.nodemanager.resource.memory-mb	250880	Copied from STEP 2 "Available Resources"

STEP 5: Verify YARN Settings on Cluster

Go to the Resource Manager Web UI (usually http://<ResourceManagerIP>:8088/ and verify the "Memory Total" and "Vcores Total" matches the values above. If your machine has no bad nodes, then the numbers should match exactly.

Resource Manager Property to Check	Value	Note
Expected Value for "Vcores Total"	1760	Calculated from STEP 2 "YARN Available Vcores" and STEP 3
Expected Value for "Memory Total" (in GB)	2450	Calculated from STEP 2 "YARN Available Memory" and STEP 3

Step 6: Verify Container Settings on Cluster

In step 6, you can change the four values that impact the size of your containers.

The minimum number of vcores should be 1. When additional vcores are required, adding 1 at a time should result in the most efficient allocation. Set the maximum number of vcore reservations for a container to ensure that no single task consumes all available resources.

Set the minimum and maximum reservations for memory. The increment should be the smallest amount that can impact performance. Here, the minimum is approximately 1 GB, the maximum is approximately 8 GB, and the increment is 512 MB.

STEP 6: Verify Container Settings on Cluster

In order to have YARN jobs run cleanly, you need to configure the container properties.

YARN Container Configuration Property (Vcores)	Value	Description
yarn.scheduler.minimum-allocation-vcores	1	Minimum vcore reservation for a container
yarn-scheduler.maximum-allocation-vcores	32	Maximum vcore reservation for a container
yarn.scheduler.increment-allocation-vcores	1	Vcore allocations must be a multiple of this value

YARN Container Configuration Property (Memory)	Value	Description
yarn.scheduler.minimum-allocation-mb	1024	Minimum memory reservation for a container
yarn.scheduler.maximum-allocation-mb	8192	Maximum memory reservation for a container
yarn.scheduler.increment-allocation-mb	512	Memory allocations must be a multiple of this value

Step 6A: Cluster Container Capacity

Step 6A lets you validate the minimum and maximum number of containers in your cluster, based on the numbers you entered.

Step 6A: Cluster Container Capacity

This section will tell you the capacity of your cluster (in terms of containers).

Cluster Container Estimates	Value
Largest number of containers, based on memory configuration	2450
Smallest number of containers, based on memory configuration	306
Largest number of containers, based on vcore configuration	1760
Smallest number of containers, based on vcore configuration	55

Step 6B: Container Sanity Checking

Step 6B lets you see at a glance whether you have over-allocated resources.

STEP 6B: Container Sanity Checking

This section will do some basic checking of your container parameters in STEP 6 against the hosts.

Sanity Check	Check Status	Description
Vcore Max >= Vcore Min	GOOD	yarn.scheduler.maximum-alloca
Memory Max >= Memory Min	GOOD	yarn.scheduler.maximum-alloca
VCoreMin >= 0	GOOD	yarn.scheduler.minimum-alloca
VCoreMin <= HostsVCores	GOOD	yarn.scheduler.minimum-alloca
VCoreMax >= 1	GOOD	yarn.scheduler.maximum-alloca
VCoreMax <= HostsVcores	GOOD	yarn.scheduler.maximum-alloca
Memory Min < 1024 MB	GOOD	If yarn.scheduler.minimum-allo
Memory Max <= HostsMemory	GOOD	yarn.scheduler.maximum-alloca

MapReduce Configuration

On the MapReduce Configuration tab, you can plan for increased task-specific memory capacity.

Step 7: MapReduce Configuration

You can increase the memory allocation for the ApplicationMaster, map tasks, and reduce tasks. The minimum vcore allocation for any task is always 1. The Spill/Sort memory allocation of 256 should be sufficient, and should be (rarely) increased if you determine that frequent spills to disk are hurting job performance.

STEP 7: MapReduce Configuration

Property	Property Type	Component	Value	Description
yarn.app.mapreduce.am.resource.cpu-vcores	Config	Application Master	1	AM container vcore reservation
yarn.app.mapreduce.am.resource.mb	Config	Application Master	1024	AM container memory reservation
mapreduce.map.cpu.vcores	Config	Map Task	1	Map task vcore reservation
mapreduce.map.memory.mb	Config	Map Task	1024	Map task memory reservation
mapreduce.reduce.cpu.vcores	Config	Reduce Task	1	Reduce task vcore reservation
mapreduce.reduce.memory.mb	Config	Reduce Task	1024	Reduce task memory reservation
mapreduce.task.io.sort.mb	Config	Spill/Sort (Map Task)	256	Spill/Sort memory reservation

Step 7A: MapReduce Sanity Checking

Step 7A lets you verify at a glance that all of your minimum and maximum resource allocations are within the parameters you set.

STEP 7A: MapReduce Sanity Checking

Sanity check MapReduce settings against container minimum/maximum properties.

Application Master Sanity Checks	Value	Description
yarn.app.mapreduce.am.resource.cpu-vcores >= container min	GOOD	Make sure ApplicationMaster vcore request fits within container limits
yarn.app.mapreduce.am.resource.cpu-vcores <= container max	GOOD	Ditto
yarn.app.mapreduce.am.resource.mb >= container min	GOOD	Make sure ApplicationMaster memory request fits within container limits
yarn.app.mapreduce.am.resource.mb <= container max	GOOD	Ditto

Map Task Sanity Checks	Value	Description
mapreduce.map.cpu.vcores >= container min	GOOD	Make sure Map Task vcore request fits within container limits
mapreduce.map.cpu.vcores <= container max	GOOD	Ditto
mapreduce.map.cpu.memory.mb >= container min	GOOD	Make sure Map Task memory request fits within container limits
mapreduce.map.cpu.memory.mb <= container max	GOOD	Ditto

Reduce Task Sanity Checks	Value	Description
mapreduce.reduce.cpu.vcores >= container min	GOOD	Make sure Reduce Task vcore request fits within container limits
mapreduce.reduce.cpu.vcores <= container max	GOOD	Ditto
mapreduce.reduce.cpu.memory.mb >= container min	GOOD	Make sure Reduce Task memory request fits within container limits
mapreduce.reduce.cpu.memory.mb <= container max	GOOD	Ditto

Configuring Your Cluster In Cloudera Manager

When you are satisfied with the cluster configuration estimates, use the values in the spreadsheet to set the corresponding properties in Cloudera Manager. For more information, see Modifying Configuration Properties Using Cloudera Manager on page 60.

Table 15: Cloudera Manager Property Correspondence

Step	YARN/MapReduce Property	Cloudera Manager Equivalent
4	yarn.nodemanager.resource.cpu-vcores	Container Virtual CPU Cores
4	yarn.nodemanager.resource.memory-mb	Container Memory
6	yarn.scheduler.minimum-allocation-vcores	Container Virtual CPU Cores Minimum
6	yarn.scheduler.maximum-allocation-vcores	Container Virtual CPU Cores Maximum
6	yarn.scheduler.increment-allocation-vcores	Container Virtual CPU Cores Increment
6	yarn.scheduler.minimum-allocation-mb	Container Memory Minimum
6	yarn.scheduler.maximum-allocation-mb	Container Memory Maximum
6	yarn.scheduler.increment-allocation-mb	Container Memory Increment
7	yarn.app.mapreduce.am.resource.cpu-vcores	ApplicationMaster Virtual CPU Cores

Step	YARN/MapReduce Property	Cloudera Manager Equivalent
7	`yarn.app.mapreduce.am.resource.mb`	ApplicationMaster Memory
7	`mapreduce.map.cpu.vcores`	Map Task CPU Virtual Cores
7	`mapreduce.map.memory.mb`	Map Task Memory
7	`mapreduce.reduce.cpu.vcores`	Reduce Task CPU Virtual Cores
7	`mapreduce.reduce.memory.mb`	Reduce Task Memory
7	`mapreduce.task.io.sort.mb`	I/O Sort Memory

In CDH 5.5 and higher, the common mapreduce parameters `mapreduce.map.java.opts`, `mapreduce.reduce.java.opts`, and `yarn.app.mapreduce.am.command-opts` are configured for you automatically based on the *Heap to Container Size Ratio*.

Resource Management

Resource management helps ensure predictable behavior by defining the impact of different services on cluster resources. Use resource management to:

- Guarantee completion in a reasonable time frame for critical workloads.
- Support reasonable cluster scheduling between groups of users based on fair allocation of resources per group.
- Prevent users from depriving other users access to the cluster.

Cloudera Manager Resource Management

Cloudera Manager provides two methods for allocating cluster resources to services: static and dynamic.

Static Allocation

Statically allocating resources using cgroups is configurable through a single *static service pool* wizard. You allocate services as a percentage of total resources, and the wizard configures the cgroups.

For example, the following figure illustrates static pools for HBase, HDFS, Impala, and YARN services that are respectively assigned 20%, 30%, 20%, and 30% of cluster resources.
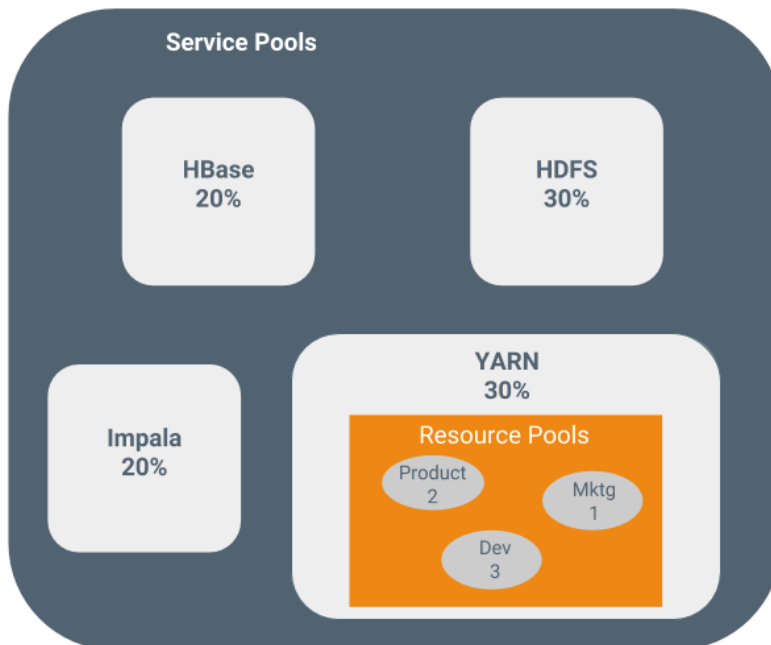
Dynamic Allocation

You can dynamically apportion resources that are statically allocated to YARN and Impala by using *dynamic resource pools*.

Depending on the version of CDH you are using, dynamic resource pools in Cloudera Manager support the following scenarios:

- **YARN** - YARN manages the virtual cores, memory, running applications, maximum resources for undeclared children (for parent pools), and scheduling policy for each pool. In the preceding diagram, three dynamic resource pools—Dev, Product, and Mktg with weights 3, 2, and 1 respectively—are defined for YARN. If an application starts

and is assigned to the Product pool, and other applications are using the Dev and Mktg pools, the Product resource pool receives 30% x 2/6 (or 10%) of the total cluster resources. If no applications are using the Dev and Mktg pools, the YARN Product pool is allocated 30% of the cluster resources.

- **Impala** - Impala manages memory for pools running queries and limits the number of running and queued queries in each pool.

The scenarios in which YARN manages resources map to the YARN [scheduler](#) policy. The scenarios in which Impala independently manages resources use Impala [admission control](#).

Static Service Pools

Static service pools isolate the services in your cluster from one another, so that load on one service has a bounded impact on other services. Services are allocated a static percentage of total resources—CPU, memory, and I/O weight—which are not shared with other services. When you configure static service pools, Cloudera Manager computes recommended memory, CPU, and I/O configurations for the worker roles of the services that correspond to the percentage assigned to each service. Static service pools are implemented per role group within a cluster, using [Linux control groups (cgroups)](#) and cooperative memory limits (for example, Java maximum heap sizes). Static service pools can be used to control access to resources by HBase, HDFS, Impala, MapReduce, Solr, Spark, YARN, and [add-on](#) services. Static service pools are not enabled by default.

> **Note:**
>
> - I/O allocation only works when [short-circuit reads](#) are enabled.
> - I/O allocation does not handle write side I/O because cgroups in the Linux kernel do not currently support buffered writes.

Viewing Static Service Pool Status

Select **Clusters** > *Cluster name* > **Static Service Pools**.If the cluster has a YARN service, the Static Service Pools Status tab displays and shows whether resource management is enabled for the cluster, and the currently configured service pools.

See [Monitoring Static Service Pools](#) on page 242 for more information.

Enabling and Configuring Static Service Pools

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

1. Select **Clusters** > *Cluster name* > **Static Service Pools**.
2. Click the **Configuration** tab. The **Step 1 of 4: Basic Allocation Setup** page displays. In each field in the basic allocation table, enter the percentage of resources to give to each service. The total must add up to 100%. Click **Continue** to proceed.
3. **Step 2: Review Changes** - The allocation of resources for each resource type and role displays with the new values as well as the values previously in effect. The values for each role are set by role group; if there is more than one role group for a given role type (for example, for RegionServers or DataNodes) then resources will be allocated separately for the hosts in each role group. Take note of changed settings. If you have previously customized these settings, check these over carefully.

   - Click the ❯ to the right of each percentage to display the allocations for a single service. Click ❯ to the right of the Total (100%) to view all the allocations in a single page.
   - Click the **Back** button to go to the previous page and change your allocations.

   When you are satisfied with the allocations, click **Continue**.

4. **Step 3 of 4: Restart Services** - To apply the new allocation percentages, click **Restart Now** to restart the cluster. To skip this step, click **Restart Later**. If HDFS High Availability is enabled, you will have the option to choose a [rolling restart](#).

5. **Step 4 of 4: Progress** displays the status of the restart commands. Click **Finished** after the restart commands complete.

6. After you enable static service pools, there are three additional tasks.

    a. Delete everything under the local directory path on NodeManager hosts. The local directory path is configurable, and can be verified in Cloudera Manager with **YARN** > **Configuration** > **NodeManager Local Directories**.

    b. Enable cgroups for resource management. You can enable cgroups in Cloudera Manager with **Yarn** > **Configuration** > **Use CGroups for Resource Management**.

    c. If you are using the optional Impala scratch directory, delete all files in the Impala scratch directory. The directory path is configurable, and can be verified in Cloudera Manager with **Impala** > **Configuration** > **Impala Daemon Scratch Directories**.

Disabling Static Service Pools

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

To disable static service pools, disable cgroup-based resource management for all hosts in all clusters:

1. In the main navigation bar, click **Hosts**.
2. Click the **Configuration** tab.
3. Select **Scope** > **Resource Management**.
4. Clear the **Enable Cgroup-based Resource Management** property.
5. Click **Save Changes**.
6. Restart all services.

Static resource management is disabled, but the percentages you set when you configured the pools, and all the changed settings (for example, heap sizes), *are retained* by the services. The percentages and settings will also be used when you re-enable static service pools. If you want to revert to the settings you had before static service pools were enabled, follow the procedures in <u>Viewing and Reverting Configuration Changes</u> on page 80.

Linux Control Groups (cgroups)

Minimum Required Role: Full Administrator

Cloudera Manager supports the Linux control groups (cgroups) kernel feature. With cgroups, administrators can impose per-resource restrictions and limits on services and roles. This provides the ability to allocate resources using cgroups to enable isolation of compute frameworks from one another. Resource allocation is implemented by setting properties for the services and roles.

Linux Distribution Support

Cgroups are a feature of the Linux kernel, and as such, support depends on the host's Linux distribution and version as shown in the following tables. If a distribution lacks support for a given parameter, changes to the parameter have no effect.

Table 16: RHEL-compatible

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Red Hat Enterprise Linux, CentOS, and Oracle Enterprise Linux 7	■	■	■	■
Red Hat Enterprise Linux, CentOS, and Oracle Enterprise Linux 6	■	■	■	■

Table 17: SLES

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
SUSE Linux Enterprise Server 11	■	■	■	■

Table 18: Ubuntu

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Ubuntu 14.04 LTS	■	■	■	■
Ubuntu 12.04 LTS	■	■	■	■
Ubuntu 10.04 LTS	■		■	■

Table 19: Debian

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Debian 7.1	■			
Debian 7.0	■			
Debian 6.0	■			

The exact level of support can be found in the Cloudera Manager Agent log file, shortly after the Agent has started. See Viewing the Cloudera Manager Server Log on page 329 to find the Agent log. In the log file, look for an entry like this:

```
Found cgroups capabilities: {
'has_memory': True,
'default_memory_limit_in_bytes': 9223372036854775807,
'writable_cgroup_dot_procs': True,
'has_cpu': True,
'default_blkio_weight': 1000,
'default_cpu_shares': 1024,
'has_blkio': True}
```

The has_cpu and similar entries correspond directly to support for the CPU, I/O, and memory parameters.

Further Reading

- https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt
- https://www.kernel.org/doc/Documentation/cgroup-v1/blkio-controller.txt
- https://www.kernel.org/doc/Documentation/cgroup-v1/memory.txt
- Managing System Resources on Red Hat Enterprise Linux 6
- Managing System Resources on Red Hat Enterprise Linux 7

Resource Management with Control Groups

To use cgroups, you must enable cgroup-based resource management under the host resource management configuration properties. However, if you configure static service pools, this property is set as part of that process.

Enabling Resource Management

Cgroups-based resource management can be enabled for all hosts, or on a per-host basis.

1. If you have upgraded from a version of Cloudera Manager older than Cloudera Manager 4.5, restart every Cloudera Manager Agent before using cgroups-based resource management:

a. Stop all services, including the Cloudera Management Service.

b. On each cluster host, run as root:

- RHEL-compatible 7 and higher:

```
$ sudo service cloudera-scm-agent next_stop_hard
$ sudo service cloudera-scm-agent restart
```

- All other Linux distributions:

```
$ sudo service cloudera-scm-agent hard_restart
```

c. Start all services.

2. Click the **Hosts** tab.

3. Optionally click the link of the host where you want to enable cgroups.

4. Click the **Configuration** tab.

5. Select **Category** > **Resource Management**.

6. Select **Enable Cgroup-based Resource Management**.

7. Restart all roles on the host or hosts.

Limitations

- Role group and role instance override cgroup-based resource management parameters must be saved one at a time. Otherwise some of the changes that should be reflected dynamically will be ignored.
- The role group abstraction is an imperfect fit for resource management parameters, where the goal is often to take a numeric value for a host resource and distribute it amongst running roles. The role group represents a "horizontal" slice: the same role across a set of hosts. However, the cluster is often viewed in terms of "vertical" slices, each being a combination of worker roles (such as TaskTracker, DataNode, RegionServer, Impala Daemon, and so on). Nothing in Cloudera Manager guarantees that these disparate horizontal slices are "aligned" (meaning, that the role assignment is identical across hosts). If they are unaligned, some of the role group values will be incorrect on unaligned hosts. For example a host whose role groups have been configured with memory limits but that's missing a role will probably have unassigned memory.

Configuring Resource Parameters

After enabling cgroups, you can restrict and limit the resource consumption of roles (or role groups) on a per-resource basis. All of these parameters can be found in the Cloudera Manager Admin Console, under the Resource Management category:

- **CPU Shares** - The more CPU shares given to a role, the larger its share of the CPU when under contention. Until processes on the host (including both roles managed by Cloudera Manager and other system processes) are contending for all of the CPUs, this will have no effect. When there is contention, those processes with higher CPU shares will be given more CPU time. The effect is linear: a process with 4 CPU shares will be given roughly twice as much CPU time as a process with 2 CPU shares.

  Updates to this parameter are dynamically reflected in the running role.

- **I/O Weight** - The greater the I/O weight, the higher priority will be given to I/O requests made by the role when I/O is under contention (either by roles managed by Cloudera Manager or by other system processes).

  This only affects read requests; write requests remain unprioritized. The Linux I/O scheduler controls when buffered writes are flushed to disk, based on time and quantity thresholds. It continually flushes buffered writes from multiple sources, not certain prioritized processes.

  Updates to this parameter are dynamically reflected in the running role.

- **Memory Soft Limit** - When the limit is reached, the kernel will reclaim pages charged to the process if and only if the host is facing memory pressure. If reclaiming fails, the kernel may kill the process. Both anonymous as well as page cache pages contribute to the limit.

After updating this parameter, you must restart the role for changes to take effect.

- **Memory Hard Limit** - When a role's resident set size (RSS) exceeds the value of this parameter, the kernel will swap out some of the role's memory. If it is unable to do so, it will kill the process. The kernel measures memory consumption in a manner that does not necessarily match what the `top` or `ps` report for RSS, so expect that this limit is a rough approximation.

  After updating this parameter, you must restart the role for changes to take effect.

Example: Protecting Production MapReduce Jobs from Impala Queries

Suppose you have MapReduce deployed in production and want to roll out Impala without affecting production MapReduce jobs. For simplicity, we will make the following assumptions:

- The cluster is using homogenous hardware
- Each worker host has two cores
- Each worker host has 8 GB of RAM
- Each worker host is running a DataNode, TaskTracker, and an Impala Daemon
- Each role type is in a single role group
- Cgroups-based resource management has been enabled on all hosts

Action	Procedure
CPU	1. Leave DataNode and TaskTracker role group CPU shares at 1024. 2. Set Impala Daemon role group's CPU shares to 256. 3. The TaskTracker role group should be configured with a Maximum Number of Simultaneous Map Tasks of 2 and a Maximum Number of Simultaneous Reduce Tasks of 1. This yields an upper bound of three MapReduce tasks at any given time; this is an important detail for memory sizing.
Memory	1. Set Impala Daemon role group memory limit to 1024 MB. 2. Leave DataNode maximum Java heap size at 1 GB. 3. Leave TaskTracker maximum Java heap size at 1 GB. 4. Leave MapReduce Child Java Maximum Heap Size for Gateway at 1 GB. 5. Leave cgroups hard memory limits alone. We'll rely on "cooperative" memory limits exclusively, as they yield a nicer user experience than the cgroups-based hard memory limits.
I/O	1. Leave DataNode and TaskTracker role group I/O weight at 500. 2. Impala Daemon role group I/O weight is set to 125.

When you're done with configuration, restart all services for these changes to take effect. The results are:

1. When MapReduce jobs are running, all Impala queries together will consume up to a fifth of the cluster's CPU resources.
2. Individual Impala Daemons will not consume more than 1 GB of RAM. If this figure is exceeded, new queries will be cancelled.
3. DataNodes and TaskTrackers can consume up to 1 GB of RAM each.
4. We expect up to 3 MapReduce tasks at a given time, each with a maximum heap size of 1 GB of RAM. That's up to 3 GB for MapReduce tasks.
5. The remainder of each host's available RAM (6 GB) is reserved for other host processes.
6. When MapReduce jobs are running, read requests issued by Impala queries will receive a fifth of the priority of either HDFS read requests or MapReduce read requests.

Dynamic Resource Pools

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

A *dynamic resource pool* is a named configuration of resources and a policy for scheduling the resources among YARN applications and Impala queries running in the pool. Dynamic resource pools allow you to schedule and allocate resources to YARN applications and Impala queries based on a user's access to specific pools and the resources available to those pools. If a pool's allocation is not in use, it can be [preempted](#) and distributed to other pools. Otherwise, a pool receives a share of resources according to the pool's weight. Access control lists (ACLs) restrict who can submit work to dynamic resource pools and administer them.

Configuration Sets define the allocation of resources across pools that can be active at a given time. For example, you can define "daytime" and "off hour" configuration sets, for which you specify different resource allocations during the daytime and for the remaining time of the week.

A *scheduling rule* defines when a [configuration set](#) is active. The configurations in the configuration set are propagated to the [fair scheduler](#) allocation file as required by the schedule. The updated files are stored in the YARN ResourceManager configuration directory `/var/run/cloudera-scm-agent/process/nn-yarn-RESOURCEMANAGER` on the host running the ResourceManager role. See [Server and Client Configuration](#).

The resources available for sharing are subject to the allocations made for each service if [static service pools](#) (cgroups) are enforced. For example, if the static pool for YARN is 75% of the total cluster resources, resource pools will use only 75% of resources.

After you create or edit dynamic resource pool settings, **Refresh Dynamics Resource Pools** and **Discard Changes** buttons display. Click **Refresh Dynamics Resource Pools** to propagate the settings to the [fair scheduler](#) allocation file (by default, `fair-scheduler.xml`). The updated files are stored in the YARN ResourceManager configuration directory `/var/run/cloudera-scm-agent/process/nn-yarn-RESOURCEMANAGER` on the host running the ResourceManager role. See [Server and Client Configuration](#).

For information on determining how allocated resources are used, see [Cluster Utilization Reports](#) on page 400.

Pool Hierarchy

YARN resource pools can be nested, with subpools restricted by the settings of their parent pool. This allows you to specify a set of pools whose resources are limited by the resources of a parent. Each subpool can have its own resource restrictions; if those restrictions fall within the configuration of the parent pool, the limits for the subpool take effect. If the limits for the subpool exceed those of the parent, the parent pool limits take precedence.

You create a parent pool either by configuring it as a parent or by [creating a subpool](#) under the pool. Once a pool is a parent, you cannot submit jobs to that pool; they must be submitted to a subpool.

Managing Dynamic Resource Pools

The main entry point for using dynamic resource pools with Cloudera Manager is the **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration** menu item.

Viewing Dynamic Resource Pool Configuration

Depending on which resource management scenario described in [Cloudera Manager Resource Management](#) on page 367 is in effect, the dynamic resource pool configuration overview displays the following tabs:

- **YARN** - Weight, Virtual Cores, Min and Max Memory, Max Running Apps, Max Resources for Undeclared Children, and Scheduling Policy
- **Impala Admission Control** - Max Memory, Max Running Queries, Max Queued Queries, Queue Timeout, and Default Query Memory Limit

To view dynamic resource pool configuration:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service [enabled for dynamic resource pools](#), the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** or **Impala Admission Control** tab.

Creating a YARN Dynamic Resource Pool

There is always a resource pool named `root.default`. By default, all YARN applications run in this pool. You create additional pools when your workload includes identifiable groups of applications (such as from a particular application, or a particular group within your organization) that have their own requirements.

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** tab.
3. Click **Create Resource Pool**. The Create Resource Pool dialog box displays, showing the Resource Limits tab.
4. Specify a name and resource limits for the pool:

   - In the **Resource Pool Name** field, specify a unique pool name containing only alphanumeric characters. If referencing a user or group name that contains a ".", replace the "." with "_dot_".
   - To make the pool a parent pool, select the **Parent Pool** checkbox.
   - Define a configuration set by assigning values to the listed properties. Specify a weight that indicates that pool's share of resources relative to other pools, minimum and maximum for virtual cores and memory, and a limit on the number of applications that can run simultaneously in the pool. If this is a parent pool, you can also set the maximum resources for undeclared children.
   - Click the **Scheduling Policy** tab and select a policy:

     - **Dominant Resource Fairness (DRF) (default)** - An extension of fair scheduling for more than one resource. DRF determines CPU and memory resource shares based on the availability of those resources and the job requirements.
     - **Fair (FAIR)** - Determines resource shares based on memory.
     - **First-In, First-Out (FIFO)** - Determines resource shares based on when a job was added.

   - If you have enabled Fair Scheduler preemption, click the **Preemption** tab and optionally set a preemption timeout to specify how long a job in this pool must wait before it can preempt resources from jobs in other pools. To enable preemption, follow the procedure in Enabling and Disabling Fair Scheduler Preemption on page 387.
   - If you have enabled ACLs and specified users or groups, optionally click the **Submission** and **Administration Access Control** tabs to specify which users and groups can submit applications and which users can view all and kill applications. By default, anyone can submit, view all, and kill applications. To restrict these permissions, select **Allow these users and groups** and provide a comma-delimited list of users and groups in the Users and Groups fields respectively.

5. Click **Create**.
6. Click **Refresh Dynamic Resource Pools**.

Enabling and Disabling Dynamic Resource Pools for Impala

By default, admission control and dynamic resource pools for Impala are disabled. Only when both are enabled does the Impala Admission Control tab appear in the Dynamic Resource Pool Configuration tab. To enable and disable Impala dynamic resource pools, follow the procedure in Enabling and Disabling Impala Admission Control Using Cloudera Manager on page 397.

Creating an Impala Dynamic Resource Pool

There is always a resource pool designated as `root.default`. By default, all Impala queries run in this pool when the dynamic resource pool feature is enabled for Impala. You create additional pools when your workload includes identifiable groups of queries (such as from a particular application, or a particular group within your organization) that have their own requirements for concurrency, memory use, or service level agreement (SLA). Each pool has its own settings related to memory, number of queries, and timeout interval.

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **Impala Admission Control** tab.

3. Click **Create Resource Pool**.

4. Specify a name and resource limits for the pool:

   - In the **Resource Pool Name** field, type a unique name containing only alphanumeric characters.
   - Optionally click the **Submission Access Control** tab to specify which users and groups can submit queries. By default, anyone can submit queries. To restrict this permission, select the **Allow these users and groups** option and provide a comma-delimited list of users and groups in the Users and Groups fields respectively.

5. Click **Create**.

6. Click **Refresh Dynamic Resource Pools**.

Choosing Settings for an Impala Dynamic Resource Pool

Impala dynamic resource pools support the following settings:

Max Memory

The maximum amount of aggregate memory, cluster-wide, that can be used by all queries running concurrently in the pool. In conjunction with **Default Query Memory Limit** and the number of Impala hosts in the cluster, Impala determines the expected maximum memory used by all queries in the pool and holds back any further queries once the estimated upper limit is reached.

> **Note:** If you specify **Max Memory** for an Impala dynamic resource pool, you must also specify the **Default Query Memory Limit**. **Max Memory** relies on the **Default Query Memory Limit** to produce a reliable estimate of overall memory consumption for a query.

For example, consider the following scenario:

- The cluster is running `impalad` daemons on five DataNodes.
- A dynamic resource pool has **Max Memory** set to 100 GB.
- The **Default Query Memory Limit** for the pool is 10 GB. Therefore, any query running in this pool could use up to 50 GB of memory (default query memory limit * number of Impala nodes).
- The maximum number of queries that Impala executes concurrently within this dynamic resource pool is two, which is the most that could be accomodated within the 100 GB **Max Memory** cluster-wide limit.
- There is no memory penalty if queries use less memory than the **Default Query Memory Limit** per-host setting or the **Max Memory** cluster-wide limit. These values are only used to estimate how many queries can be run concurrently within the resource constraints for the pool.

Max Running Queries

The maximum number of queries that can run concurrently in this pool. The default value is unlimited. Any queries for this pool that exceed **Max Running Queries** are added to the admission control queue until other queries finish. You can use **Max Running Queries** in the early stages of resource management, when you do not have extensive data about query memory usage, to determine if the cluster performs better overall if throttling is applied to Impala queries.

For a workload with many small queries, you typically specify a high value for this setting, or leave the default setting of "unlimited". For a workload with expensive queries, where some number of concurrent queries saturate the memory, I/O, CPU, or network capacity of the cluster, set the value low enough that the cluster resources are not overcommitted for Impala.

Once you have enabled memory-based admission control using other pool settings, you can still use **Max Running Queries** as a safeguard. If queries exceed either the total estimated memory or the maximum number of concurrent queries, they are added to the queue.

Max Queued Queries

The maximum number of queries that can be in the admission control queue for a pool at any one time. Additional queries cannot be added to the queue until other queries begin running, reducing the queue size. The default value is 200. Typically, this value does not need to be adjusted. If a large number of queries are queued, you can change other parameters such as the timeout interval, or you can increase the pool capacity.

Queue Timeout

The amount of time, in milliseconds, that a query waits in the admission control queue for this pool before being canceled. The default value is 60,000 (60 seconds).

It the following cases, **Queue Timeout** is not significant and you can specify a high value to avoid canceling queries unexpectedly:

- In a low-concurrency workload, where few or no queries are queued
- In an environment without a strict SLA, where it does not matter if queries occasionally take longer than usual because they are held in admission control

You might also need to increase the value to use Impala with some business intelligence tools that have their own timeout intervals for queries.

In a high-concurrency workload, especially for queries with a tight SLA, long wait times in admission control can cause a serious problem. For example, if a query needs to run in 10 seconds, and you have tuned it so that it runs in 8 seconds, it violates its SLA if it waits in the admission control queue longer than 2 seconds. In a case like this, set a low timeout value and monitor how many queries are cancelled because of timeouts. This technique helps you to discover capacity, tuning, and scaling problems early, and helps avoid wasting resources by running expensive queries that have already missed their SLA.

If you identify some queries that can have a high timeout value, and others that benefit from a low timeout value, you can create separate pools with different values for this setting.

Default Query Memory Limit

The equivalent of setting the `MEM_LIMIT` query option for each query that runs in this pool. This value represents the maximum memory the query can use on each host. If the query exceeds that memory on a host, it activates spill-to-disk, and could be canceled if available memory is too low. Impala multiplies this default memory limit value by the number of Impala hosts in the cluster to estimate how many queries fit within the total RAM represented by **Max Memory**, which represents a cluster-wide limit.

Cloning a Resource Pool

To create a new pool with the same properties as an existing pool, you can clone a pool:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** or **Impala Admission Control** tab.
3. Click ▼ at the right of a resource pool row and select **Clone**.
4. Specify a name for the pool and edit pool properties as desired.
5. Click **Create**.
6. Click **Refresh Dynamic Resource Pools**.

Deleting a Resource Pool

To delete a pool:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** or **Impala Admission Control** tab.
3. Click ▼ at the right of a resource pool row and select **Delete**.
4. Click **OK**.
5. Click **Refresh Dynamic Resource Pools**.

Editing Dynamic Resource Pools

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** or **Impala Admission Control** tab.
3. Click **Edit** at the right of a resource pool row. Edit the properties.
4. If you have enabled ACLs and specified users or groups, optionally click the **Submission** and **Administration Access Control** tabs to specify which users and groups can submit applications and which users can view all and kill applications. By default, anyone can submit, view all, and kill applications. To restrict these permissions, select **Allow these users and groups** and provide a comma-delimited list of users and groups in the Users and Groups fields respectively.
5. Click **Save**.
6. Click **Refresh Dynamic Resource Pools**.

YARN Pool Status and Configuration Options

Viewing Dynamic Resource Pool Status

Do one of the following:

- **Cluster** menu

    1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. The **YARN** > **Resource Pools** tab displays.
    2. Click **View Dynamic Resource Pool Status**.

- YARN service

    1. Go to the YARN service.
    2. Click the **Resource Pools** tab.

See Monitoring Dynamic Resource Pools on page 243 for more information.

Configuring Default YARN Fair Scheduler Properties

For information on the default properties, see Configuring the Fair Scheduler on page 384.

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. The **YARN** > **Resource Pools** tab displays.
2. Click the **YARN** tab.
3. Click the **Default Settings** button.
4. Specify the default scheduling policy, maximum applications, and preemption timeout properties.
5. Click **Save**.
6. Click **Refresh Dynamic Resource Pools**.

Creating a Resource Subpool

YARN resource pools can form a nested hierarchy. To manually create a subpool:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. The **YARN** > **Resource Pools** tab displays.
2. Click ▼ at the right of a resource pool row and select **Create Subpool**. Configure subpool properties.
3. Click **Create**.
4. Click **Refresh Dynamic Resource Pools**.

Setting YARN User Limits

Pool properties determine the maximum number of applications that can run in a pool. To limit the number of applications specific users can run at the same time:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. The **YARN** > **Resource Pools** tab displays.

2. Click the **User Limits** tab. The table displays a list of users and the maximum number of jobs each user can submit.

3. Click **Add User Limit**.

4. Specify a username. containing only alphanumeric characters. If referencing a user or group name that contains a ".", replace the "." with "_dot_".

5. Specify the maximum number of running applications.

6. Click **Create**.

7. Click **Refresh Dynamic Resource Pools**.

Configuring ACLs

To configure which users and groups can submit and kill applications in any resource pool:

1. Enable ACLs.

2. In Cloudera Manager select the YARN service.

3. Click the Configuration tab.

4. Search for **Admin ACL** property, and specify which users and groups can submit and kill applications.

> **Note:** Group names in YARN Resource Manager ACLs are case sensitive. Consequently, if you specify an uppercase group name in the ACL, it will not match the group name resolved from the Active Directory because the Active Directory group name is resolved in lowercase.

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

6. Return to the Home page by clicking the Cloudera Manager logo.

7. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

8. Click **Restart Stale Services**.

9. Click **Restart Now**.

10. Click **Finish**.

Enabling ACLs

To specify whether ACLs are checked:

1. In Cloudera Manager select the YARN service.

2. Click the **Configuration** tab.

3. Search for the **Enable ResourceManager ACL** property, and select the YARN service.

4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

5. Return to the Home page by clicking the Cloudera Manager logo.

6. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

7. Click **Restart Stale Services**.

8. Click **Restart Now**.

9. Click **Finish**.

Defining Configuration Sets

Configuration Sets define the allocation of resources across pools that can be active at a given time. For example, you can define "daytime" and "off hour" configuration sets, for which you specify different resource allocations during the daytime and for the remaining time of the week.

You define configuration set activity by creating scheduling rules. Once you have defined configuration sets, you can configure limits such as weight, minimum and maximum memory and virtual cores, and maximum running applications.

Specifying Resource Limit Properties

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.

2. Click the **Resource Pools** tab.

3. For each resource pool, click **Edit**.

    a. Select a configuration set name.

    b. Edit the configuration set properties and click **Save**.

4. Click **Refresh Dynamic Resource Pools**.

Example Configuration Sets

The **Daytime** configuration set assigns the **production** pool five times the resources of the **development** pool:

The **Off Hour** configuration set assigns the **production** and **development** pools an equal amount of resources:

See example scheduling rules for these configuration sets.

Viewing the Properties of a Configuration Set

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.

2. In the **Configuration Sets** drop-down list, select a configuration set. The properties of each pool for that configuration set display.

Scheduling Configuration Sets

A **scheduling rule** defines when a [configuration set](#) is active. The configurations in the configuration set are propagated to the [fair scheduler](#) allocation file as required by the schedule. The updated files are stored in the YARN ResourceManager configuration directory `/var/run/cloudera-scm-agent/process/`*nn*`-yarn-RESOURCEMANAGER` on the host running the ResourceManager role. See [Server and Client Configuration](#).

Example Scheduling Rules

Consider the example [Daytime and Off Hour](#) configuration sets. To specify that the **Daytime** configuration set is active every weekday from 8:00 a.m. to 5:00 p.m. and the **Off Hour** configuration set is active all other times (evenings and weekends), define the following rules:

	Scheduling Rule	Configuration Set
1	Repeats weekly on Monday, Tuesday, Wednesday, Thursday, Friday from 8:00 AM to 5:00 PM (PST), starting 03/08/2016.	**Daytime**
2	Repeats weekly on Monday, Tuesday, Wednesday, Thursday, Friday from 5:00 PM to 8:00 AM (PST), starting 03/08/2016.	**Off Hour**
3	Repeats weekly on Sunday, Saturday from 12:00 AM to 12:00 AM (PST), starting 03/08/2016.	**Off Hour**

Adding a Scheduling Rule

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service [enabled for dynamic resource pools](#), the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **Scheduling Rules** tab.
3. Click **Create Scheduling Rule**.
4. In the Configuration Set field, choose the configuration set to which the rule applies. Select **Create New** or **Use Existing**.
5. If you create a new configuration set, type a name in the Name field.

   If you use an existing configuration set, select one from the drop-down list.

6. Configure the rule to repeat or not:
   - To repeat the rule, keep the **Repeat** field selected and specify the repeat frequency. If the frequency is weekly, specify the repeat day or days.
   - If the rule does not repeat, clear the **Repeat** field, click the left side of the **on** field to display a drop-down calendar where you set the starting date and time. When you specify the date and time, a default time window of two hours is set in the right side of the **on** field. Click the right side to adjust the date and time.
7. Click **Create**.
8. Click **Refresh Dynamic Resource Pools**.

Reordering Scheduling Rules

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service [enabled for dynamic resource pools](#), the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **Scheduling Rules** tab.
3. Click **Reorder Scheduling Rules**.
4. Click **Move Up** or **Move Down** in a rule row.
5. Click **Save**.
6. Click **Refresh Dynamic Resource Pools**.

Editing a Scheduling Rule

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service [enabled for dynamic resource pools](#), the **Impala Admission Control** > **Resource Pools** tab displays.

   2. Click **Scheduling Rules**.

   3. Click **Edit** at the right of a rule.

   4. Edit the rule.

   5. Click **Save**.

   6. Click **Refresh Dynamic Resource Pools**.

Deleting a Scheduling Rule

   1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.

   2. Click **Scheduling Rules**.

   3. Click ▼ at the right of a rule and select **Delete**.

   4. Click **OK**.

   5. Click **Refresh Dynamic Resource Pools**.

Assigning Applications and Queries to Resource Pools

You can specify at run time that an application or query should run in a named resource pool. To specify a pool at run time for a YARN application, provide the pool name in the `mapreduce.job.queuename` property. To specify a pool at run time for an Impala query, provide the pool name in the REQUEST_POOL option.

Cloudera Manager allows you to specify a set of ordered rules for assigning applications and queries to pools. You can also specify default pool settings directly in the YARN fair scheduler configuration.

Some rules allow to you specify that the pool be created in the dynamic resource pool configuration if it does not already exist. Allowing pools to be created is optional. If a rule is satisfied and you do not create a pool, YARN runs the job "ad hoc" in a pool to which resources are not assigned or managed.

If *no rule is satisfied* when the application or query runs, the YARN application or Impala query is rejected.

Placement Rule Ordering and Evaluation

Pool placement rules are evaluated in the order in which they appear in the placement rule list. When a job is submitted, the rules are evaluated, and the first matching rule is used to determine the pool in which the job is run.

If a rule is always satisfied, subsequent rules are not evaluated. Rules that are never evaluated appear in struck-though gray text. For an example see Example Placement Rules on page 383.

By default, pool placement rules are ordered in reverse order of when they were added; the last rule added appears first. You can easily reorder rules.

Creating Pool Placement Rules

Placement rules determine the pools to which applications and queries are assigned. At installation, Cloudera Manager provides a set of default rules and rule ordering.

To create a placement rule:

   1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.

   2. Click the **YARN** or **Impala Admission Control** tab.

   3. Click the **Placement Rules** tab.

   4. Click **Create Placement Rule**. In the **Type** drop-down list, select a rule that specifies the name of pool and its position in the pool hierarchy:

      • **YARN**

- **specified at run time** - Use `root.[pool name]`, where `pool name` is the name of the pool specified at run time.
      - **root.users.[username]** - Use the parent pool `root.users` in a pool named by the user submitting the application. The `root.users` parent pool and this rule are created by default. However, on upgrading from Cloudera Manager 5.7, neither the pool or placement rule is added.
      - **root.default** - Use `root.default`.
      - **root.[pool name]** - Use `root.pool name`, where `pool name` is the name you specify in the **Pool Name** field that displays after you select the rule.
      - **root.[primary group]** - Use the pool that matches the primary group of the user submitting the application.
      - **root.[secondary group]** - Use the pool that matches one of the secondary groups of the user that submitted the application.
      - **root.[username]** - Use the pool that matches the name of the user that submitted the application.
      - **root.[primary group].[username]** - Use the parent pool that matches the primary group of the user that submitted the application and then a subpool that matches the username.
      - **root.[secondary group].[username]** - Use the parent pool that matches one of the secondary groups of the user that submitted the application and then a subpool that matches the username.

- **Impala**

  The following rules are applied in the order that they are listed. If the selected pool does not exist, the `root.default` pool is used.

  1. **specified at run time** - Use the `REQUEST_POOL` query option. For example, `SET REQUEST_POOL=root.[pool name]`.
  2. **root.[username]** - Use the pool that matches the name of the user that submitted the query. This is not recommended.
  3. **root.[primary group]** - Use the pool that matches the primary group of the user that submitted the query.
  4. **root.[secondary group]** - Use the pool that matches one of the secondary groups of the user that submitted the query.
  5. **root.default** - Use `root.default`.

  > **Note:** Currently, it's not possible to map a username to a resource pool with different name. For example, you cannot map the `user1` user to the `root.pool1` resource pool.

  For example, consider the following three Impala users.

  - username1: test1, secondarygroupname: testgroup

  - username2: test2, secondarygroupname: testgroup

  - username3: test3, secondarygroupname: testgroup

  This cluster had 2 dynamic pools: `root.default` and `root.testgroup`.

  With the above placement rule, query submitted by all three users will be mapped to the `root.testgroup` resource pool. If the `root.testgroup` pool is removed, the queries submitted by the users will be mapped to the `root.default` pool.

  For more information about these rules, see the description of the `queuePlacementPolicy` element in [Allocation File Format](#).

5. **(YARN only)** To indicate that the pool should be created if it does not exist when the application runs, check the **Create pool if it does not exist** checkbox.
6. Click **Create**. The rule is added to the top of the placement rule list and becomes the first rule evaluated.
7. Click **Refresh Dynamic Resource Pools**.

Default Placement Rules and Order

The default placement rules and order for YARN are:

1. Use the pool specified at run time and create the pool if it does not exist.
2. Use the pool root.users.[username].
3. Use the pool root.default.

Also see Example Placement Rules on page 383.

The default rules and order for Impala are:

1. Use the pool specified at run time, only if the pool exists.
2. Use the pool root.default.

Reordering Pool Placement Rules

To change the order in which pool placement rules are evaluated:

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays. If the cluster has an Impala service enabled for dynamic resource pools, the **Impala Admission Control** > **Resource Pools** tab displays.
2. Click the **YARN** or **Impala Admission Control** tab.
3. Click the **Placement Rules** tab.
4. Click **Reorder Placement Rules**.
5. Click **Move Up** or **Move Down** in a rule row.
6. Click **Save**.
7. Click **Refresh Dynamic Resource Pools**.

Example Placement Rules

The following figures show the default pool placement rule setting for YARN:

If a pool is specified at run time, that pool is used for the job and the pool is created if it did not exist. If no pool is specified at run time, a pool named according to the user submitting the job within the `root.users` parent pool is used. If that pool cannot be used (for example, because the `root.users` pool is a leaf pool), pool `root.default` is used.

If you move rule 2 down (which specifies to run the job in a pool named after the user running the job nested within the parent pool `root.users`), rule 2 becomes disabled because the previous rule (Use the pool `root.default`) is always satisfied.

	Placement Rule		
1	Use the pool specified at runtime and create the pool if it does not exist.	⬆ Move Up	⬇ Move Down
2	Use the pool root.default. This rule is always satisfied. Subsequent rules are not used.	⬆ Move Up	⬇ Move Down
3	Use the pool root.users.[username].	⬆ Move Up	⬇ Move Down

Reorder Placement Rules

YARN (MRv2) and MapReduce (MRv1) Schedulers

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

A *scheduler* determines which jobs run, where and when they run, and the resources allocated to the jobs. The YARN (MRv2) and MapReduce (MRv1) computation frameworks support the following schedulers:

- **FIFO** - Allocates resources based on arrival time.
- **Fair** - Allocates resources to weighted pools, with fair sharing within each pool. When configuring the scheduling policy of a pool, Domain Resource Fairness (DRF) is a type of fair scheduler.

  - CDH Fair Scheduler

- **Capacity** - Allocates resources to pools, with FIFO scheduling within each pool.

  - CDH Capacity Scheduler

The scheduler defaults for YARN and MapReduce 1 are:

- **YARN** - Cloudera Manager and CDH set the default to Fair Scheduler. Cloudera recommends Fair Scheduler.

  In YARN, the scheduler is responsible for allocating resources to the various running applications subject to familiar constraints of capacities, queues, and so on. The scheduler performs its scheduling function based on resource requirements of the applications; it does so based on the abstract notion of a resource *container* that incorporates elements such as memory, CPU, disk, and network.

  The YARN scheduler has a pluggable policy, which is responsible for partitioning cluster resources among the various queues, applications, and so on.

  You can manually configure the scheduler type. If you choose the Fair Scheduler, see Configuring the Fair Scheduler on page 384 for information on how to manually configure it. Alternatively you can use Cloudera Manager dynamic allocation to manage scheduler configuration.

- **MapReduce 1** - Cloudera Manager and CDH 5 set the default scheduler to FIFO for backward compatibility, however Cloudera recommends Fair Scheduler.

Configuring the Fair Scheduler

The Fair Scheduler is the Cloudera recommended scheduler option. The Fair Scheduler controls how resources are allocated to *pool*s (or *queue*s) and how jobs are assigned to pools. Jobs can also be explicitly submitted to pools; to submit an job to a specific pool, you specify the `mapreduce.job.queuename` property.

Pools have policies for preempting running jobs, so that when there is contention for resources, jobs that have high priority or have been waiting a long time to run are allowed to run.

Fair Scheduler configuration is maintained in two files: `yarn-site.xml` and `fair-scheduler.xml`. Detailed information on the available properties is available at Fair Scheduler Configuration. When you change the contents of `yarn-site.xml`, you must restart the YARN service.

In Cloudera Manager the Dynamic Resource Pools Configuration screen provides an enhanced interface for configuring the Fair Scheduler. In addition to allowing you to configure resource allocation properties, you can define schedules for changing the values of the properties. Cloudera Manager automatically updates Fair Scheduler configuration files according to the schedule.

Table 20: yarn-site.xml Properties

Property	Description
yarn.scheduler.fair.allow-undeclared-pools	When set to **true**, the Fair Scheduler uses the username as the default pool name, in the event that a pool name is not specified. When set to **false**, all applications are run in a shared pool, called **default**. Default: true.
yarn.scheduler.fair.user-as-default-queue	When set to **true**, pools specified in applications but not explicitly configured, are created at runtime with default settings. When set to **false**, applications specifying pools not explicitly configured run in a pool named **default**. This setting applies when an application explicitly specifies a pool and when the application runs in a pool named with the username associated with the application. Default: true.
yarn.scheduler.fair.preemption	When enabled, under certain conditions, the Fair Scheduler preempts applications in other pools. Preemption guarantees that production applications are not starved while also allowing the cluster to be used for experimental and research applications. To minimize wasted computation, the Fair Scheduler preempts the most recently launched applications. Default: false.
yarn.scheduler.fair.preemption.cluster-utilization-threshold	The cluster utilization threshold above which preemption is triggered. If the cluster utilization is under this threshold, preemption is not triggered even if there are starved queues. The utilization is computed as the maximum ratio of usage to capacity among all resources. Default: .8.

For example:

```
...
  <property>
    <name>yarn.scheduler.fair.allow-undeclared-pools</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.scheduler.fair.user-as-default-queue</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.scheduler.fair.preemption</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.scheduler.fair.preemption.cluster-utilization-threshold</name>
    <value>0.8</value>
  </property>
...
```

Figure 5: yarn-site.xml Example

Table 21: fair-scheduler.xml Properties

Element	Subelement	Description
queuePlacementPolicy		Policy for assigning jobs to resource pools. In Cloudera Manager this property is configured using [placement rules](#).
userMaxAppsDefault		Default running app limit for a user whose limit is not otherwise specified. In Cloudera Manager this property is configured using [user limits](#).
queueMaxAppsDefault		Default running app limit for pools; overridden by the equivalent element in a pool.
queueMaxAMShareDefault		Default ApplicationMaster resource limit for the pool; overridden by the equivalent element in a pool.
defaultFairSharePreemptionThreshold		Fair share preemption threshold for pools; overridden by the equivalent element in a pool. The threshold value is between 0 and 1. If set to x and the fair share of the pool is F, resources are preempted from other pools if the allocation is less than (x * F).
defaultFairSharePreemptionTimeout		Default number of seconds a resource pool is under its fair share before it will preempt containers to take resources from other resource pools; overridden by the equivalent element in a pool. If this parameter is not set, and fairSharePreemptionTimeout is not set for a given queue or its ancestor queues, pre-emption by this queue will never occur, even if pre-emption is enabled. Default timeout is 2^64 milliseconds.
defaultMinSharePreemptionTimeout		Default number of seconds a resource pool is under its minimum share before it will preempt containers to take resources from other resource pools; overridden by the equivalent element in a pool.
defaultQueueSchedulingPolicy		Default scheduling policy for pools; overridden by the equivalent element in a pool. Default: drf.
queue		Name of a [dynamic resource pool](#).
	weight	Weight given to the resource pool when determining how to allocate resources relative to other resource pools. In Cloudera Manager this property is configured using [configuration sets](#).
	schedulingPolicy	Policy to determine how resources are allocated to the resource pool: fair, fifo, or drf.
	aclSubmitApps	Users and groups that can submit jobs to the pool.
	aclAdministerApps	Users and groups that can administer the pool.
	minResources, maxResources	Minimum and maximum share of resources that can allocated to the resource pool in the form X mb, Y vcores. Values computed by the weight settings are limited by (or constrained by) the minimum and maximum values.
	maxAMShare	Fraction of the resource pool's fair share that can be used to run ApplicationMasters. For example, if set to 1.0, then ApplicationMasters in the pool can take up to 100% of both the memory and CPU fair share. The value of -1.0 disables this feature, and the ApplicationMaster share is not checked. The default value is 0.5.
	maxRunningApps	See default elements.

Element	Subelement	Description
	fairSharePreemptionThreshold	See default elements.
	fairSharePreemptionTimeout	See default elements.
	minSharePreemptionTimeout	See default elements.

For example:

```
<allocations>
    <queue name="root">
        <weight>1.0</weight>
        <schedulingPolicy>drf</schedulingPolicy>
        <aclSubmitApps> </aclSubmitApps>
        <aclAdministerApps>*</aclAdministerApps>
        <queue name="production">
            <minResources>1024 mb, 10 vcores</minResources>
            <maxResources>5120 mb, 20 vcores</maxResources>
            <weight>4.0</weight>
            <schedulingPolicy>drf</schedulingPolicy>
            <aclSubmitApps>*</aclSubmitApps>
            <aclAdministerApps>*</aclAdministerApps>
        </queue>
        <queue name="development">
            <weight>1.0</weight>
            <schedulingPolicy>drf</schedulingPolicy>
            <aclSubmitApps>*</aclSubmitApps>
            <aclAdministerApps>*</aclAdministerApps>
        </queue>
    </queue>
    <defaultQueueSchedulingPolicy>drf</defaultQueueSchedulingPolicy>
    <queuePlacementPolicy>
        <rule name="specified" create="true"/>
        <rule name="user" create="true"/>
    </queuePlacementPolicy>
</allocations>
```

Figure 6: fair-scheduler.xml

Dynamic resource pools allow you to configure scheduler properties. See Configuring Default YARN Fair Scheduler Properties on page 377.

Enabling and Disabling Fair Scheduler Preemption

You can enable the Fair Scheduler to preempt applications in other pools if a pool's fair or minimum share is not met for some period of time. When you create a pool, you can specify whether preemption is allowed and whether a specific pool can be preempted. Fair scheduler preemption is controlled by several properties. For more information, see Configuring the Fair Scheduler on page 384.

Enabling and Disabling Preemption Using Cloudera Manager

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

Enabling Preemption

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays.
2. Click **Default Settings**.
3. Click the **Enable Fair Scheduler Preemption** link.
4. Select the **ResourceManager Default Group** checkbox.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Return to the Home page by clicking the Cloudera Manager logo.
7. Click the ⏻ icon that is next to any stale services to invoke the cluster restart wizard.

8. Click **Restart Stale Services**.
9. Click **Restart Now**.
10. Click **Finish**.

Disabling Preemption

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays.
2. In Cloudera Manager select the YARN service.
3. Clear the **Enable Fair Scheduler Preemption** checkbox.
4. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
5. Return to the Home page by clicking the Cloudera Manager logo.
6. Click the ⟳ icon that is next to any stale services to invoke the cluster restart wizard.
7. Click **Restart Stale Services**.
8. Click **Restart Now**.
9. Click **Finish**.

Enabling and Disabling Preemption for a Pool

1. Select **Clusters** > *Cluster name* > **Dynamic Resource Pool Configuration**. If the cluster has a YARN service, the **YARN** > **Resource Pools** tab displays.
2. In a pool row, click **Edit**.
3. Click the **Preemption** tab.
4. Select or clear **Preemptable**.
5. Click **Save**.
6. Click **Refresh Dynamic Resource Pools**.

Resource Management for Impala

> **Note:**
>
> The use of the Llama component for integrated resource management within YARN is no longer supported with and higher. The Llama support code is removed entirely in and higher.
>
> For clusters running Impala alongside other data management components, you define static service pools to define the resources available to Impala and other components. Then within the area allocated for Impala, you can create dynamic service pools, each with its own settings for the Impala admission control feature.

You can limit the CPU and memory resources used by Impala, to manage and prioritize workloads on clusters that run jobs from many Hadoop components.

How Resource Limits Are Enforced

- If Cloudera Manager Static Partitioning is used, it creates a cgroup in which Impala runs. This cgroup limits CPU, network, and IO according to the static partitioning policy.
- Limits on memory usage are enforced by Impala's process memory limit (the `MEM_LIMIT` query option setting). The admission control feature checks this setting to decide how many queries can be safely run at the same time. Then the Impala daemon enforces the limit by activating the spill-to-disk mechanism when necessary, or cancelling a query altogether if the limit is exceeded at runtime.

impala-shell Query Options for Resource Management

Before issuing SQL statements through the `impala-shell` interpreter, you can use the `SET` command to configure the following parameters related to resource management:

- EXPLAIN_LEVEL Query Option
- MEM_LIMIT Query Option

Limitations of Resource Management for Impala

The `MEM_LIMIT` query option, and the other resource-related query options, are settable through the ODBC or JDBC interfaces in Impala 2.0 and higher. This is a former limitation that is now lifted.

Admission Control and Query Queuing

Admission control is an Impala feature that imposes limits on concurrent SQL queries, to avoid resource usage spikes and out-of-memory conditions on busy CDH clusters. It is a form of "throttling". New queries are accepted and executed until certain conditions are met, such as too many queries or too much total memory used across the cluster. When one of these thresholds is reached, incoming queries wait to begin execution. These queries are queued and are admitted (that is, begin executing) when the resources become available.

In addition to the threshold values for currently executing queries, you can place limits on the maximum number of queries that are queued (waiting) and a limit on the amount of time they might wait before returning with an error. These queue settings let you ensure that queries do not wait indefinitely, so that you can detect and correct "starvation" scenarios.

Queries, DML statements, and some DDL statements, including `CREATE TABLE AS SELECT` and `COMPUTE STATS` are affected by admission control.

Enable this feature if your cluster is underutilized at some times and overutilized at others. Overutilization is indicated by performance bottlenecks and queries being cancelled due to out-of-memory conditions, when those same queries are successful and perform well during times with less concurrent load. Admission control works as a safeguard to avoid out-of-memory conditions during heavy concurrent usage.

> **Note:**
>
> The use of the Llama component for integrated resource management within YARN is no longer supported with and higher. The Llama support code is removed entirely in and higher.
>
> For clusters running Impala alongside other data management components, you define static service pools to define the resources available to Impala and other components. Then within the area allocated for Impala, you can create dynamic service pools, each with its own settings for the Impala admission control feature.

Overview of Impala Admission Control

On a busy CDH cluster, you might find there is an optimal number of Impala queries that run concurrently. For example, when the I/O capacity is fully utilized by I/O-intensive queries, you might not find any throughput benefit in running more concurrent queries. By allowing some queries to run at full speed while others wait, rather than having all queries contend for resources and run slowly, admission control can result in higher overall throughput.

For another example, consider a memory-bound workload such as many large joins or aggregation queries. Each such query could briefly use many gigabytes of memory to process intermediate results. Because Impala by default cancels queries that exceed the specified memory limit, running multiple large-scale queries at once might require re-running some queries that are cancelled. In this case, admission control improves the reliability and stability of the overall workload by only allowing as many concurrent queries as the overall memory of the cluster can accommodate.

The admission control feature lets you set an upper limit on the number of concurrent Impala queries and on the memory used by those queries. Any additional queries are queued until the earlier ones finish, rather than being cancelled or running slowly and causing contention. As other queries finish, the queued queries are allowed to proceed.

In and higher, you can specify these limits and thresholds for each pool rather than globally. That way, you can balance the resource usage and throughput between steady well-defined workloads, rare resource-intensive queries, and ad hoc exploratory queries.

For more details on the internal workings of admission control, see How Impala Schedules and Enforces Limits on Concurrent Queries on page 391.

Concurrent Queries and Admission Control

One way to limit resource usage through admission control is to set an upper limit on the number of concurrent queries. This is the initial technique you might use when you do not have extensive information about memory usage for your workload. This setting can be specified separately for each dynamic resource pool.

You can combine this setting with the memory-based approach described in Memory Limits and Admission Control on page 390. If either the maximum number of or the expected memory usage of the concurrent queries is exceeded, subsequent queries are queued until the concurrent workload falls below the threshold again.

See for information about all these dynamic resource pool settings, how to use them together, and how to divide different parts of your workload among different pools.

Memory Limits and Admission Control

Each dynamic resource pool can have an upper limit on the cluster-wide memory used by queries executing in that pool. This is the technique to use once you have a stable workload with well-understood memory requirements.

Always specify the **Default Query Memory Limit** for the expected maximum amount of RAM that a query might require on each host, which is equivalent to setting the `MEM_LIMIT` query option for every query run in that pool. That value affects the execution of each query, preventing it from overallocating memory on each host, and potentially activating the spill-to-disk mechanism or cancelling the query when necessary.

Optionally, specify the **Max Memory** setting, a cluster-wide limit that determines how many queries can be safely run concurrently, based on the upper memory limit per host multiplied by the number of Impala nodes in the cluster.

For example, consider the following scenario:

- The cluster is running `impalad` daemons on five DataNodes.
- A dynamic resource pool has **Max Memory** set to 100 GB.
- The **Default Query Memory Limit** for the pool is 10 GB. Therefore, any query running in this pool could use up to 50 GB of memory (default query memory limit * number of Impala nodes).
- The maximum number of queries that Impala executes concurrently within this dynamic resource pool is two, which is the most that could be accomodated within the 100 GB **Max Memory** cluster-wide limit.
- There is no memory penalty if queries use less memory than the **Default Query Memory Limit** per-host setting or the **Max Memory** cluster-wide limit. These values are only used to estimate how many queries can be run concurrently within the resource constraints for the pool.

> **Note:** If you specify **Max Memory** for an Impala dynamic resource pool, you must also specify the **Default Query Memory Limit**. **Max Memory** relies on the **Default Query Memory Limit** to produce a reliable estimate of overall memory consumption for a query.

You can combine the memory-based settings with the upper limit on concurrent queries described in Concurrent Queries and Admission Control on page 390. If either the maximum number of or the expected memory usage of the concurrent queries is exceeded, subsequent queries are queued until the concurrent workload falls below the threshold again.

See for information about all these dynamic resource pool settings, how to use them together, and how to divide different parts of your workload among different pools.

How Impala Admission Control Relates to Other Resource Management Tools

The admission control feature is similar in some ways to the Cloudera Manager static partitioning feature, as well as the YARN resource management framework. These features can be used separately or together. This section describes

some similarities and differences, to help you decide which combination of resource management features to use for Impala.

Admission control is a lightweight, decentralized system that is suitable for workloads consisting primarily of Impala queries and other SQL statements. It sets "soft" limits that smooth out Impala memory usage during times of heavy load, rather than taking an all-or-nothing approach that cancels jobs that are too resource-intensive.

Because the admission control system does not interact with other Hadoop workloads such as MapReduce jobs, you might use YARN with static service pools on CDH clusters where resources are shared between Impala and other Hadoop components. This configuration is recommended when using Impala in a **multitenant** cluster. Devote a percentage of cluster resources to Impala, and allocate another percentage for MapReduce and other batch-style workloads. Let admission control handle the concurrency and memory usage for the Impala work within the cluster, and let YARN manage the work for other components within the cluster. In this scenario, Impala's resources are not managed by YARN.

The Impala admission control feature uses the same configuration mechanism as the YARN resource manager to map users to pools and authenticate them.

Although the Impala admission control feature uses a `fair-scheduler.xml` configuration file behind the scenes, this file does not depend on which scheduler is used for YARN. You still use this file, and Cloudera Manager can generate it for you, even when YARN is using the capacity scheduler.

How Impala Schedules and Enforces Limits on Concurrent Queries

The admission control system is decentralized, embedded in each Impala daemon and communicating through the statestore mechanism. Although the limits you set for memory usage and number of concurrent queries apply cluster-wide, each Impala daemon makes its own decisions about whether to allow each query to run immediately or to queue it for a less-busy time. These decisions are fast, meaning the admission control mechanism is low-overhead, but might be imprecise during times of heavy load across many coordinators. There could be times when the more queries were queued (in aggregate across the cluster) than the specified limit, or when number of admitted queries exceeds the expected number. Thus, you typically err on the high side for the size of the queue, because there is not a big penalty for having a large number of queued queries; and you typically err on the low side for configuring memory resources, to leave some headroom in case more queries are admitted than expected, without running out of memory and being cancelled as a result.

To avoid a large backlog of queued requests, you can set an upper limit on the size of the queue for queries that are queued. When the number of queued queries exceeds this limit, further queries are cancelled rather than being queued. You can also configure a timeout period per pool, after which queued queries are cancelled, to avoid indefinite waits. If a cluster reaches this state where queries are cancelled due to too many concurrent requests or long waits for query execution to begin, that is a signal for an administrator to take action, either by provisioning more resources, scheduling work on the cluster to smooth out the load, or by doing Impala performance tuning to enable higher throughput.

How Admission Control works with Impala Clients (JDBC, ODBC, HiveServer2)

Most aspects of admission control work transparently with client interfaces such as JDBC and ODBC:

- If a SQL statement is put into a queue rather than running immediately, the API call blocks until the statement is dequeued and begins execution. At that point, the client program can request to fetch results, which might also block until results become available.
- If a SQL statement is cancelled because it has been queued for too long or because it exceeded the memory limit during execution, the error is returned to the client program with a descriptive error message.

In Impala 2.0 and higher, you can submit a SQL `SET` statement from the client application to change the `REQUEST_POOL` query option. This option lets you submit queries to different resource pools, as described in REQUEST_POOL Query Option.

At any time, the set of queued queries could include queries submitted through multiple different Impala daemon hosts. All the queries submitted through a particular host will be executed in order, so a `CREATE TABLE` followed by an `INSERT` on the same table would succeed. Queries submitted through different hosts are not guaranteed to be executed in the order they were received. Therefore, if you are using load-balancing or other round-robin scheduling where different statements are submitted through different hosts, set up all table structures ahead of time so that

the statements controlled by the queuing system are primarily queries, where order is not significant. Or, if a sequence of statements needs to happen in strict order (such as an `INSERT` followed by a `SELECT`), submit all those statements through a single session, while connected to the same Impala daemon host.

Admission control has the following limitations or special behavior when used with JDBC or ODBC applications:

- The other resource-related query options, `RESERVATION_REQUEST_TIMEOUT` and `V_CPU_CORES`, are no longer used. Those query options only applied to using Impala with Llama, which is no longer supported.

SQL and Schema Considerations for Admission Control

When queries complete quickly and are tuned for optimal memory usage, there is less chance of performance or capacity problems during times of heavy load. Before setting up admission control, tune your Impala queries to ensure that the query plans are efficient and the memory estimates are accurate. Understanding the nature of your workload, and which queries are the most resource-intensive, helps you to plan how to divide the queries into different pools and decide what limits to define for each pool.

For large tables, especially those involved in join queries, keep their statistics up to date after loading substantial amounts of new data or adding new partitions. Use the `COMPUTE STATS` statement for unpartitioned tables, and `COMPUTE INCREMENTAL STATS` for partitioned tables.

When you use dynamic resource pools with a **Max Memory** setting enabled, you typically override the memory estimates that Impala makes based on the statistics from the `COMPUTE STATS` statement. You either set the `MEM_LIMIT` query option within a particular session to set an upper memory limit for queries within that session, or a default `MEM_LIMIT` setting for all queries processed by the `impalad` instance, or a default `MEM_LIMIT` setting for all queries assigned to a particular dynamic resource pool. By designating a consistent memory limit for a set of similar queries that use the same resource pool, you avoid unnecessary query queuing or out-of-memory conditions that can arise during high-concurrency workloads when memory estimates for some queries are inaccurate.

Follow other steps from [Tuning Impala for Performance](#) to tune your queries.

Configuring Admission Control

The configuration options for admission control range from the simple (a single resource pool with a single set of options) to the complex (multiple resource pools with different options, each pool handling queries for a different set of users and groups). Cloudera recommends configuring the settings through the Cloudera Manager user interface.

> **Important:** Although the following options are still present in the Cloudera Manager interface under the **Admission Control** configuration settings dialog, If possible, avoid using them in and higher. These settings only apply if you enable admission control but leave dynamic resource pools disabled. In and higher, prefer to set up dynamic resource pools and customize the settings for each pool, as described in *Creating an Impala Dynamic Resource Pool* and *Editing Dynamic Resource Pools* in .

Impala Service Flags for Admission Control (Advanced)

The following Impala configuration options let you adjust the settings of the admission control feature. When supplying the options on the `impalad` command line, prepend the option name with `--`.

queue_wait_timeout_ms

 Purpose: Maximum amount of time (in milliseconds) that a request waits to be admitted before timing out.

 Type: `int64`

 Default: `60000`

default_pool_max_requests

 Purpose: Maximum number of concurrent outstanding requests allowed to run before incoming requests are queued. Because this limit applies cluster-wide, but each Impala node makes independent decisions to run queries immediately or queue them, it is a soft limit; the overall number of concurrent queries might be slightly higher during times of heavy load. A negative value indicates no limit. Ignored if `fair_scheduler_config_path` and `llama_site_path` are set.

Type: `int64`

Default: -1, meaning unlimited (prior to the default was 200)

default_pool_max_queued

Purpose: Maximum number of requests allowed to be queued before rejecting requests. Because this limit applies cluster-wide, but each Impala node makes independent decisions to run queries immediately or queue them, it is a soft limit; the overall number of queued queries might be slightly higher during times of heavy load. A negative value or 0 indicates requests are always rejected once the maximum concurrent requests are executing. Ignored if `fair_scheduler_config_path` and `llama_site_path` are set.

Type: `int64`

Default: unlimited

default_pool_mem_limit

Purpose: Maximum amount of memory (across the entire cluster) that all outstanding requests in this pool can use before new requests to this pool are queued. Specified in bytes, megabytes, or gigabytes by a number followed by the suffix `b` (optional), `m`, or `g`, either uppercase or lowercase. You can specify floating-point values for megabytes and gigabytes, to represent fractional numbers such as `1.5`. You can also specify it as a percentage of the physical memory by specifying the suffix `%`. 0 or no setting indicates no limit. Defaults to bytes if no unit is given. Because this limit applies cluster-wide, but each Impala node makes independent decisions to run queries immediately or queue them, it is a soft limit; the overall memory used by concurrent queries might be slightly higher during times of heavy load. Ignored if `fair_scheduler_config_path` and `llama_site_path` are set.

> **Note:** Impala relies on the statistics produced by the `COMPUTE STATS` statement to estimate memory usage for each query. See COMPUTE STATS Statement for guidelines about how and when to use this statement.

Type: string

Default: `""` (empty string, meaning unlimited)

disable_pool_max_requests

Purpose: Disables all per-pool limits on the maximum number of running requests.

Type: Boolean

Default: `false`

disable_pool_mem_limits

Purpose: Disables all per-pool mem limits.

Type: Boolean

Default: `false`

fair_scheduler_allocation_path

Purpose: Path to the fair scheduler allocation file (`fair-scheduler.xml`).

Type: string

Default: `""` (empty string)

Usage notes: Admission control only uses a small subset of the settings that can go in this file, as described below. For details about all the Fair Scheduler configuration settings, see the Apache wiki.

llama_site_path

Purpose: Path to the configuration file used by admission control (`llama-site.xml`). If set, `fair_scheduler_allocation_path` must also be set.

Type: string

Default: `" "` (empty string)

Usage notes: Admission control only uses a few of the settings that can go in this file, as described below.

Configuring Admission Control Using Cloudera Manager

In Cloudera Manager, you can configure pools to manage queued Impala queries, and the options for the limit on number of concurrent queries and how to handle queries that exceed the limit. For details, see Managing Resources with Cloudera Manager.

Configuring Admission Control Using the Command Line

To configure admission control, use a combination of startup options for the Impala daemon and edit or create the configuration files `fair-scheduler.xml` and `llama-site.xml`.

For a straightforward configuration using a single resource pool named `default`, you can specify configuration options on the command line and skip the `fair-scheduler.xml` and `llama-site.xml` configuration files.

For an advanced configuration with multiple resource pools using different settings, set up the `fair-scheduler.xml` and `llama-site.xml` configuration files manually. Provide the paths to each one using the `impalad` command-line options, `--fair_scheduler_allocation_path` and `--llama_site_path` respectively.

The Impala admission control feature only uses the Fair Scheduler configuration settings to determine how to map users and groups to different resource pools. For example, you might set up different resource pools with separate memory limits, and maximum number of concurrent and queued queries, for different categories of users within your organization. For details about all the Fair Scheduler configuration settings, see the Apache wiki.

The Impala admission control feature only uses a small subset of possible settings from the `llama-site.xml` configuration file:

```
llama.am.throttling.maximum.placed.reservations.queue_name
llama.am.throttling.maximum.queued.reservations.queue_name
impala.admission-control.pool-default-query-options.queue_name
impala.admission-control.pool-queue-timeout-ms.queue_name
```

The `impala.admission-control.pool-queue-timeout-ms` setting specifies the timeout value for this pool, in milliseconds. The`impala.admission-control.pool-default-query-options` settings designates the default query options for all queries that run in this pool. Its argument value is a comma-delimited string of 'key=value' pairs, for example, `'key1=val1,key2=val2'`. For example, this is where you might set a default memory limit for all queries in the pool, using an argument such as `MEM_LIMIT=5G`.

The `impala.admission-control.*` configuration settings are available in and higher.

Examples of Admission Control Configurations

Example Admission Control Configurations Using Cloudera Manager

For full instructions about configuring dynamic resource pools through Cloudera Manager, see .

Example Admission Control Configurations Using Configuration Files

For clusters not managed by Cloudera Manager, here are sample `fair-scheduler.xml` and `llama-site.xml` files that define resource pools `root.default`, `root.development`, and `root.production`. These sample files are stripped down: in a real deployment they might contain other settings for use with various aspects of the YARN component. The settings shown here are the significant ones for the Impala admission control feature.

fair-scheduler.xml:

Although Impala does not use the `vcores` value, you must still specify it to satisfy YARN requirements for the file contents.

Each `<aclSubmitApps>` tag (other than the one for `root`) contains a comma-separated list of users, then a space, then a comma-separated list of groups; these are the users and groups allowed to submit Impala statements to the corresponding resource pool.

If you leave the `<aclSubmitApps>` element empty for a pool, nobody can submit directly to that pool; child pools can specify their own `<aclSubmitApps>` values to authorize users and groups to submit to those pools.

```xml
<allocations>
    <queue name="root">
        <aclSubmitApps> </aclSubmitApps>
        <queue name="default">
            <maxResources>50000 mb, 0 vcores</maxResources>
            <aclSubmitApps>*</aclSubmitApps>
        </queue>
        <queue name="development">
            <maxResources>200000 mb, 0 vcores</maxResources>
            <aclSubmitApps>user1,user2 dev,ops,admin</aclSubmitApps>
        </queue>
        <queue name="production">
            <maxResources>1000000 mb, 0 vcores</maxResources>
            <aclSubmitApps> ops,admin</aclSubmitApps>
        </queue>
    </queue>
    <queuePlacementPolicy>
        <rule name="specified" create="false"/>
        <rule name="default" />
    </queuePlacementPolicy>
</allocations>
```

llama-site.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>llama.am.throttling.maximum.placed.reservations.root.default</name>
    <value>10</value>
  </property>
  <property>
    <name>llama.am.throttling.maximum.queued.reservations.root.default</name>
    <value>50</value>
  </property>
  <property>
    <name>impala.admission-control.pool-default-query-options.root.default</name>
    <value>mem_limit=128m,query_timeout_s=20,max_io_buffers=10</value>
  </property>
  <property>
    <name>impala.admission-control.pool-queue-timeout-ms.root.default</name>
    <value>30000</value>
  </property>
  <property>
    <name>llama.am.throttling.maximum.placed.reservations.root.development</name>
    <value>50</value>
  </property>
  <property>
    <name>llama.am.throttling.maximum.queued.reservations.root.development</name>
    <value>100</value>
  </property>
  <property>
    <name>impala.admission-control.pool-default-query-options.root.development</name>
    <value>mem_limit=256m,query_timeout_s=30,max_io_buffers=10</value>
  </property>
  <property>
    <name>impala.admission-control.pool-queue-timeout-ms.root.development</name>
    <value>15000</value>
  </property>
  <property>
    <name>llama.am.throttling.maximum.placed.reservations.root.production</name>
    <value>100</value>
  </property>
  <property>
    <name>llama.am.throttling.maximum.queued.reservations.root.production</name>
    <value>200</value>
  </property>
<!--
```

```
        Default query options for the 'root.production' pool.
        THIS IS A NEW PARAMETER in CDH 5.7 / Impala 2.5.
        Note that the MEM_LIMIT query option still shows up in here even though it is a
        separate box in the UI. We do that because it is the most important query option

        that people will need (everything else is somewhat advanced).

        MEM_LIMIT takes a per-node memory limit which is specified using one of the
following:
        - '<int>[bB]?'  -> bytes (default if no unit given)
        - '<float>[mM(bB)]' -> megabytes
        - '<float>[gG(bB)]' -> in gigabytes
        E.g. 'MEM_LIMIT=12345' (no unit) means 12345 bytes, and you can append m or g
            to specify megabytes or gigabytes, though that is not required.
-->
  <property>
    <name>impala.admission-control.pool-default-query-options.root.production</name>
    <value>mem_limit=386m,query_timeout_s=30,max_io_buffers=10</value>
  </property>
<!--
  Default queue timeout (ms) for the pool 'root.production'.
  If this isn't set, the process-wide flag is used.
  THIS IS A NEW PARAMETER in CDH 5.7 / Impala 2.5.
-->
  <property>
    <name>impala.admission-control.pool-queue-timeout-ms.root.production</name>
    <value>30000</value>
  </property>
</configuration>
```

Guidelines for Using Admission Control

To see how admission control works for particular queries, examine the profile output for the query. This information is available through the PROFILE statement in impala-shell immediately after running a query in the shell, on the **queries** page of the Impala debug web UI, or in the Impala log file (basic information at log level 1, more detailed information at log level 2). The profile output contains details about the admission decision, such as whether the query was queued or not and which resource pool it was assigned to. It also includes the estimated and actual memory usage for the query, so you can fine-tune the configuration for the memory limits of the resource pools.

Where practical, use Cloudera Manager to configure the admission control parameters. The Cloudera Manager GUI is much simpler than editing the configuration files directly.

Remember that the limits imposed by admission control are "soft" limits. The decentralized nature of this mechanism means that each Impala node makes its own decisions about whether to allow queries to run immediately or to queue them. These decisions rely on information passed back and forth between nodes by the statestore service. If a sudden surge in requests causes more queries than anticipated to run concurrently, then throughput could decrease due to queries spilling to disk or contending for resources; or queries could be cancelled if they exceed the MEM_LIMIT setting while running.

In impala-shell, you can also specify which resource pool to direct queries to by setting the REQUEST_POOL query option.

If you set up different resource pools for different users and groups, consider reusing any classifications you developed for use with Sentry security. See [Enabling Sentry Authorization for Impala](#) for details.

For details about all the Fair Scheduler configuration settings, see [Fair Scheduler Configuration](#), in particular the tags such as <queue> and <aclSubmitApps> to map users and groups to particular resource pools (queues).

Managing Impala Admission Control

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

Admission control is an Impala feature that imposes limits on concurrent SQL queries, to avoid resource usage spikes and out-of-memory conditions on busy CDH clusters. It is a form of "throttling". New queries are accepted and executed until certain conditions are met, such as too many queries or too much total memory used across the cluster. When

one of these thresholds is reached, incoming queries wait to begin execution. These queries are queued and are admitted (that is, begin executing) when the resources become available.

For further information on Impala admission control, see Admission Control and Query Queuing on page 389.

Enabling and Disabling Impala Admission Control Using Cloudera Manager

You perform this task when you have determined that the Impala workload is heavy enough to cause capacity problems on the cluster. The capacity issues could be because of a high volume of concurrent queries, because of heavy-duty join and aggregation queries that require large amounts of memory, or because Impala is being used alongside other Hadoop data management components and the resource usage of Impala must be constrained to work well in a multitenant deployment.

> **Important:**
>
> In and higher, admission control and dynamic resource pools are enabled by default.

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Category** > **Admission Control**.
4. Select or clear both the **Enable Impala Admission Control** checkbox and the **Enable Dynamic Resource Pools** checkbox.
5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
6. Restart the Impala service.

After completing this task, for further configuration settings, customize the configuration settings for the dynamic resource pools, as described in Creating an Impala Dynamic Resource Pool on page 374 and Editing Dynamic Resource Pools on page 377.

Configuring Impala Admission Control Using Cloudera Manager

In and higher, Cloudera recommends configuring admission control at a granular level, using the dynamic resource pools feature, rather than using the cluster-wide settings.

- In the **Admission Control** configuration group for Impala, ignore all the settings below the checkboxes to enable or disable admission control and dynamic resource pools.
- Whether you are creating a new pool or editing the settings of an existing one, follow the decision process in Choosing Settings for an Impala Dynamic Resource Pool on page 375.
- If you are subdividing your workload and do not already have a dynamic resource pool set up for it, follow the procedure in Creating an Impala Dynamic Resource Pool on page 374.
- To edit the properties of an existing dynamic resource pool, follow the procedure in Editing Dynamic Resource Pools on page 377.

Data Storage for Monitoring Data

The Service Monitor and Host Monitor roles in the Cloudera Management Service store time series data, health data, and Impala query and YARN application metadata.

On this page:

Configuring Service Monitor Data Storage

The Service Monitor stores time series data and health data, Impala query metadata, and YARN application metadata. By default, the data is stored in `/var/lib/cloudera-service-monitor/` on the Service Monitor host. You can change this by modifying the **Service Monitor Storage Directory** configuration (`firehose.storage.base.directory`). To change this configuration on an active system, see Moving Monitoring Data on an Active Cluster on page 399.

You can control how much disk space to reserve for the different classes of data the Service Monitor stores by changing the following configuration options:

- Time-series metrics and health data - Time-Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)
- Impala query metadata - Impala Storage (`firehose_impala_storage_bytes` - 1 GB default)
- YARN application metadata - YARN Storage (`firehose_yarn_storage_bytes` - 1 GB default)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see Data Granularity and Time-Series Metric Data on page 398.

The default values are small, so you should examine disk usage after several days of activity to determine how much space is needed.

Configuring Host Monitor Data Storage

The Host Monitor stores time series data and health data. By default, the data is stored in `/var/lib/cloudera-host-monitor/` on the Host Monitor host. You can change this by modifying the **Host Monitor Storage Directory** configuration. To change this configuration on an active system, follow the procedure in Moving Monitoring Data on an Active Cluster on page 399.

You can control how much disk space to reserve for Host Monitor data by changing the following configuration option:

- Time-series metrics and health data: Time Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see Data Granularity and Time-Series Metric Data on page 398.

The default value is small, so you should examine disk usage after several days of activity to determine how much space they need. The **Charts Library** tab on the Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data.

Viewing Host and Service Monitor Data Storage

The Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data:

1. Select **Clusters** > **Cloudera Management Service**.
2. Click the **Charts Library** tab.

Data Granularity and Time-Series Metric Data

The Service Monitor and Host Monitor store time-series metric data in a variety of ways. When the data is received, it is written as-is to the metric store. Over time, the raw data is summarized to and stored at various data granularities. For example, after ten minutes, a summary point is written containing the average of the metric over the period as well as the minimum, the maximum, the standard deviation, and a variety of other statistics. This process is summarized to produce hourly, six-hourly, daily, and weekly summaries. This data summarization procedure applies only to metric data. When the Impala query and YARN application monitoring storage limit is reached, the oldest stored records are deleted.

The Service Monitor and Host Monitor internally manage the amount of overall storage space dedicated to each data granularity level. When the limit for a level is reached, the oldest data points at that level are deleted. Metric data for that time period remains available at the lower granularity levels. For example, when an hourly point for a particular time is deleted to free up space, a daily point still exists covering that hour. Because each of these data granularities consumes significantly less storage than the previous summary level, lower granularity levels can be retained for longer periods of time. With the recommended amount of storage, weekly points can often be retained indefinitely.

Some features, such as detailed display of health results, depend on the presence of raw data. Health history is maintained by the event store dictated by its retention policies.

Moving Monitoring Data on an Active Cluster

You can change where monitoring data is stored on a cluster.

Basic: Changing the Configured Directory

1. Stop the Service Monitor or Host Monitor.
2. Save your old monitoring data and then copy the current directory to the new directory (optional).
3. Update the **Storage Directory** configuration option (`firehose.storage.base.directory`) on the corresponding role configuration page.
4. Start the Service Monitor or Host Monitor.

Advanced: High Performance

For the best performance, and especially for a large cluster, Host Monitor and Service Monitor storage directories should have their own dedicated spindles. In most cases, that provides sufficient performance, but you can divide your data further if needed. You cannot configure this directly with Cloudera Manager; instead, you must use symbolic links.

For example, if all your Service Monitor data is located in `/data/1/service_monitor`, and you want to separate your Impala data from your time series data, you could do the following:

1. Stop the Service Monitor.
2. Move the original Impala data in `/data/1/service_monitor/impala` to the new directory, for example `/data/2/impala_data`.
3. Create a symbolic link from `/data/1/service_monitor/impala` to `/data/2/impala_data` with the following command:

```
ln -s /data/2/impala_data /data/1/service_monitor/impala
```

4. Start the Service Monitor.

Host Monitor and Service Monitor Memory Configuration

You can configure Java heap size and non-Java memory size. The memory recommended for these configuration options depends on the number of hosts in the cluster, the services running on the cluster, and the number of monitored entities. Monitored entities are the objects monitored by the Service Monitor or Host Monitor. As the number of hosts and services increases, the number of monitored entities also increases.

In addition to the memory configured, the Host Monitor and Service Monitor use the Linux page cache. Memory available for page caching on the Host Monitor and Service Monitor hosts improves performance.

To configure memory allocations, determine how many entities are being monitored and then consult the tables below for required and recommended memory configurations.

To determine the number of entities being monitored:

1. Go to **Clusters** > **Cloudera Management Service**.
2. Locate the chart with the title **Cloudera Management Service Monitored Entities**.

   The number of monitored entities for the Host Monitor and Service Monitor displays at the bottom of the chart. In the following example, the Host Monitor has 46 monitored entities and the Service Monitor has 230 monitored entities.
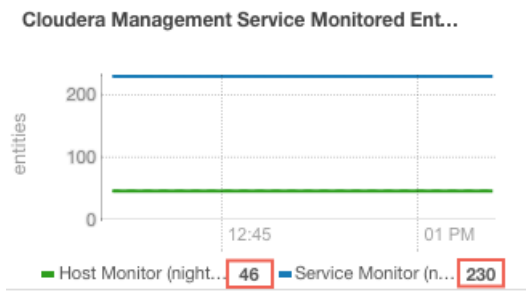
3. Use the number of monitored entities for the Host Monitor to determine its memory requirements and recommendations in the tables below.

4. Use the number of monitored entities for the Service Monitor to determine its memory requirements and recommendations in the tables below.

Clusters with HDFS, YARN, or Impala

Use the recommendations in this table for clusters where the only services having worker roles are HDFS, YARN, or Impala.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-2,000	0-100	1 GB	6 GB
2,000-4,000	100-200	1.5 GB	6 GB
4,000-8,000	200-400	1.5 GB	12 GB
8,000-16,000	400-800	2.5 GB	12 GB
16,000-20,000	800-1,000	3.5 GB	12 GB

Clusters with HBase, Solr, Kafka, or Kudu

Use the recommendations when services such as HBase, Solr, Kafka, or Kudu are deployed in the cluster. These services typically have larger quantities of monitored entities.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-30,000	0-100	2 GB	12 GB
30,000-60,000	100-200	3 GB	12 GB
60,000-120,000	200-400	3.5 GB	12 GB
120,000-240,000	400-800	8 GB	20 GB

Cluster Utilization Reports

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

The **Cluster Utilization Report** screens in Cloudera Manager display aggregated utilization information for YARN and Impala jobs. The reports display CPU utilization, memory utilization, resource allocations made due to the YARN fair scheduler, and Impala queries. The report displays aggregated utilization for the entire cluster and also breaks out

utilization by *tenant*, which is either a user or a resource pool. You can configure the report to display utilization for a range of dates, specific days of the week, and time ranges.

The report displays the current utilization of CPU and memory resources and the resources that were allocated using the Cloudera Manager resource management features. See Resource Management on page 367.

Using the information displayed in the **Cluster Utilization Report**, a CDH cluster administrator can verify that sufficient resources are available for the number and types of jobs running in the cluster. An administrator can use the reports to tune resource allocations so that resources are used efficiently and meet business requirements. Tool tips in the report pages provide suggestions about how to improve performance based on the information displayed in the report. Hover over a label to see these suggestions and other information. For example:
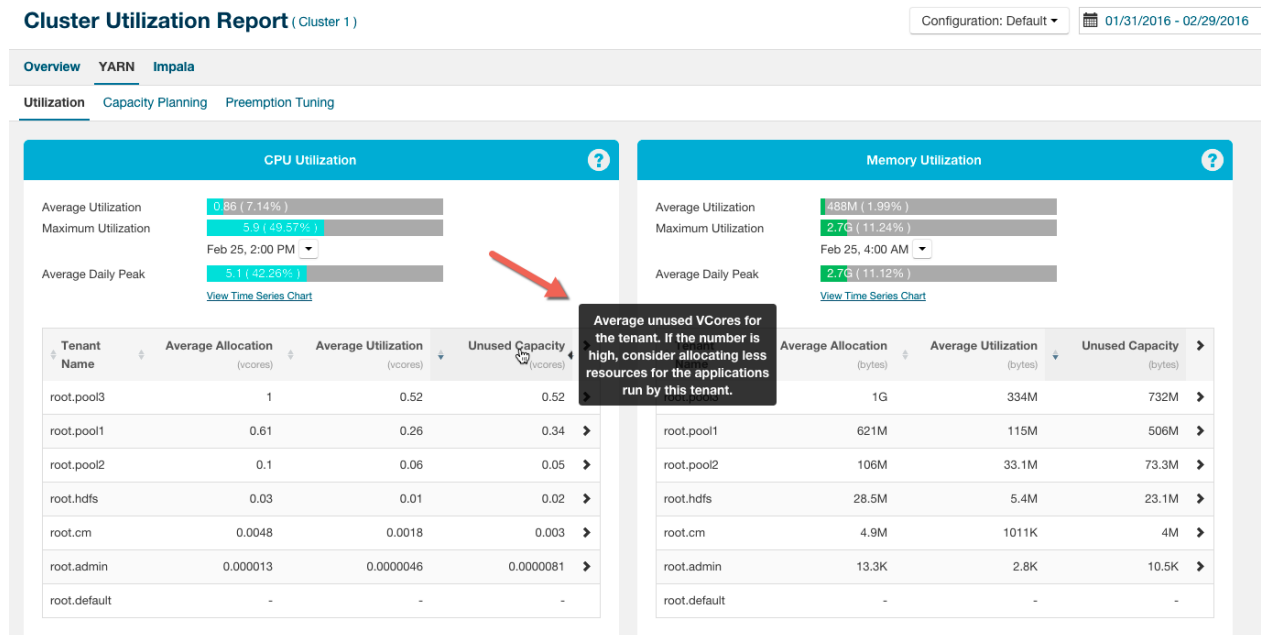
You can tune the following:

- CPU and memory allocations
- Weights for each pool
- Scheduling rules
- Preemption thresholds
- Maximum number of running and queued Impala queries
- Maximum timeout for the queue of Impala queries
- Placement rules
- Number of hosts in a cluster
- Memory capacity of hosts
- Impala Admission Control pool and queue configurations

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See Managing Licenses on page 37 for more information.

If you want to create your own reports with similar functionality, or if you want to export the report results, see Creating a Custom Cluster Utilization Report on page 409.

Configuring the Cluster Utilization Report

This topic describes the prerequisites and configurations required to use the **Cluster Utilization Report**.

Enabling the Cluster Utilization Report

By default, the **Cluster Utilization Report** displays aggregated CPU and memory utilization for an entire CDH cluster and for YARN and Impala utilization. You can also view this utilization by *tenants*, which include Linux users and Dynamic Resource Pools on page 372. To see utilization for a tenant, you must configure the tenant and define resource limits for it.

You must configure several parameters to enable the **Cluster Utilization Report**:

1. Enable YARN utilization metrics collection.

   a. Go to the YARN Service
   b. Click the **Configuration** tab.
   c. Select **Category** > **Monitoring**.
   d. Type `container` in the **Search** box.
   e. Select the **Enable Container Usage Metrics Collection** parameter.
   f. Enter a username for the MapReduce job that collects the metrics in the **Container Usage MapReduce Job User** parameter. The username you enter must be a Linux user on all the cluster hosts. If you are using an Active Directory KDC, the username must also exist in Active Directory. For secure clusters, the user must not be banned or below the minimum user ID. You can view the list of banned users (`banned.users`) and the minimum user ID (`min.user.id`) by selecting **Clusters** > ***YARN service>*** > **Configuration**.

   > **Note:** The user that is configured with the **Container Usage MapReduce Job User** property in the YARN service requires permissions to read the subdirectories of the HDFS directory specified with the **Cloudera Manager Container Usage Metrics Directory** property. The default umask of `022` allows any user to read from that directory. However, if a more strict umask (for example, `027`) is used, then those directories are not readable by any user. In that case the user specified with the **Container Usage MapReduce Job User** property should be added to the same group that owns the subdirectories.
   >
   > For example, if the `/tmp/cmYarnContainerMetrics/20161010` subdirectory is owned by user and group `yarn:hadoop`, the user specified in **Container Usage MapReduce Job User** should be added to the `hadoop` group.

   > **Note:** The directories you specify with the **Cloudera Manager Container Usage Metrics Directory** and **Container Usage Output Directory** properties should not be located in encryption zones.

   g. (Optional) Enter the resource pool in which the container usage collection MapReduce job runs in the **Container Usage MapReduce Job Pool** parameter. Cloudera recommends that you dedicate a resource pool for running this MapReduce job.

   > **Note:** If you specify a custom resource pool, ensure that the placement rules for the cluster allow for it. The first rule must be for resource pools to be specified at run time with the **Create pool if it does not exist** option selected. Alternatively, ensure that the pool you specify already exists. If the placement rule is not properly configured or the resource pool does not already exist, the job may run in a different pool.

   h. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
   i. Click the **Actions** drop-down list and select **Create CM Container Usage Metrics Dir**.
   j. Restart the YARN service:

      a. Go to the YARN service.
      b. Select **Actions** > **Restart**.

2. Enable Impala utilization collection.

      a. Go to the Impala service.

      b. Click the **Configuration** tab.

      c. Select **Category** > **Admission Control**.

      d. Select or clear both the **Enable Impala Admission Control** checkbox and the **Enable Dynamic Resource Pools** checkbox.

      e. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

      f. Restart the Impala service.

Configuring the Cluster Utilization Report

To access the **Cluster Utilization Report**, go to **Clusters** and then select **Utilization Report** for the cluster. The **Overview** tab displays when you first open the report.

The upper-right part of the page has two controls that you use to configure the **Cluster Utilization Report**:

Configuration: Default ▾   🗓 01/31/2016 - 02/29/2016

You can apply a configuration and date range that applies to all tabs in the report:

1. Click the **Configuration** drop-down menu.

2. Select one the configured options, or create a new configuration:

      a. Click **Create New Configuration**.

      b. Enter a **Configuration Name**.

      c. Select the **Tenant Type**, either **Pool** or **User**.

      d. Select the days of the week for which you want to report utilization.

      e. Select **All Day**, or use the drop-down menus to specify a utilization time range for the report.

      f. Click **Create**.

   The configuration you created is now available from the **Configuration** drop-down menu.

3. Select a date range for the report:

      a. Click the date range button.

      b. Select one of the range options (**Today**, **Yesterday**, **Last 7 Days**, **Last 30 Days**, or **This Month**) or click **Custom Range** and select the beginning and ending dates for the date range.

Using the Cluster Utilization Report to Manage Resources

To access the **Cluster Utilization Report**, go to **Clusters** and then select **Utilization Report** for the cluster. The **Overview** tab of the report displays.

> **Note:** The report updates utilization information every hour. The utilization information for Impala and YARN queries does not display in the **Cluster Utilization Report** until captured by the hourly update.

Figure 7: Cluster Utilization Report Overview Tab

The Cluster Utilization Report is divided into the following tabs:

Overview Tab

The **Overview** tab provides a summary of CPU and memory utilization for the entire cluster and also for only YARN applications and Impala queries. Two sections, **CPU Utilization** and **Memory Utilization**, display the following information:

CPU Utilization	Memory Utilization
Overall Cluster Utilization	**Overall Cluster Utilization**
• **Total CPU Cores** – Average number of CPU cores available during the reporting window. • **Average Utilization** – Average CPU utilization for the entire cluster, including resources consumed by user applications and CDH services.	• **Total Physical Memory** – Average physical memory available in the cluster during the reporting window. • **Average Utilization** – Average memory consumption for the entire cluster, including resources consumed by user applications and CDH services.

CPU Utilization	Memory Utilization

• **Maximum Utilization** – Maximum CPU utilization for the entire cluster during the reporting window, including resources consumed by user applications and CDH services. If this value is high, consider adding more hosts to the cluster.
 	
Click the drop-down menu next to the date and select one of the following to view details about jobs running when maximum utilization occurred:
 	
    • **View YARN Applications Running at the Time**
    • **View Impala Queries Running at the Time**
 	
• **Average Daily Peak** – Average daily peak CPU consumption for the entire cluster during the reporting window. This includes resources consumed by user applications and CDH services. The number is computed by averaging the maximum resource consumption for each day of the reporting period.
 	
Click **View Time Series Chart** to view a chart of peak utilization.
	• **Maximum Utilization** – Maximum memory consumption for the entire cluster during the reporting window, including resources consumed by user applications and CDH services. If this value is high, consider adding more hosts to the cluster.
	
	Click the drop-down menu next to the date and select one of the following to view details about jobs running when maximum utilization occurred:
	
	    • **View YARN Applications Running at the Time**
	    • **View Impala Queries Running at the Time**
	
	• **Average Daily Peak** – Average daily peak memory consumption for the entire cluster during the reporting window, including resources consumed by user applications and CDH services. The number is computed by averaging the maximum memory utilization for each day of the reporting period.
	
	Click **View Time Series Chart** to view a chart of peak utilization.

YARN + Impala Utilization
 	
• **Average Utilization** – Average resource consumption by YARN applications and Impala queries that ran on the cluster.
• **Maximum Utilization** – Maximum resource consumption by YARN applications and Impala queries that ran on the cluster.
 	
Click the drop-down menu next to the date and select one of the following to view details about jobs running when maximum utilization occurred:
 	
    • **View YARN Applications Running at the Time**
    • **View Impala Queries Running at the Time**
 	
• **Average Daily Peak** – Average daily peak resource consumption by YARN applications and Impala queries during the reporting window. The number is computed by finding the maximum resource consumption per day and calculating the mean.
 	
Click **View Time Series Chart** to view a chart of peak utilization.
	YARN + Impala Utilization
	
	• **Average Utilization** – Average memory consumption by YARN applications and Impala queries that ran on the cluster.
	• **Maximum Utilization** – Maximum memory consumption for the entire cluster during the reporting window, including resources consumed by user applications and CDH services. If this is high, consider adding more hosts to the cluster.
	
	Click the drop-down menu next to the date and select one of the following to view details about jobs running when maximum utilization occurred:
	
	    • **View YARN Applications Running at the Time**
	    • **View Impala Queries Running at the Time**
	
	• **Average Daily Peak** – Average daily peak memory consumption by YARN applications and Impala queries during the reporting window. The number is computed by finding the maximum resource consumption per day and then calculating the mean.
	
	Click **View Time Series Chart** to view a chart of peak utilization.

Utilization by Tenant Displays overall utilization for each tenant. Tenants can be either pools or users. See [Configuring the Cluster Utilization Report](#) on page 403.	**Utilization by Tenant** Displays overall utilization for each tenant. Tenants can be either pools or users. See [Configuring the Cluster Utilization Report](#) on page 403.

YARN Tab

The **YARN** tab displays CPU and memory utilization for YARN applications on three tabs:

- Utilization Tab on page 406
- Capacity Planning Tab on page 407
- Preemption Tuning Tab on page 407

For information about managing YARN resources, see:

- YARN (MRv2) and MapReduce (MRv1) Schedulers on page 384
- Enabling and Disabling Fair Scheduler Preemption on page 387
- Dynamic Resource Pools on page 372

Utilization Tab

Table 22: Utilization Tab

CPU Utilization	Memory Utilization
Summary section: - **Average Utilization** – Average number of vcores used by YARN applications. The percentage reported is of the total number of vcores configured for YARN. - **Maximum Utilization** – Maximum number of vcores used by YARN applications. The percentage reported is of the total number of vcores configured for YARN. Click the drop-down menu next to the date and select **View YARN Applications Running at the Time** to view details about jobs running when maximum utilization occurred. - **Average Daily Peak** – Average daily peak vcores used by YARN applications. The number is computed by finding the maximum resource consumption per day and calculating the mean. The percentage reported is of the total number of vcores configured for YARN. Click **View Time Series Chart** to view a chart of peak utilization.	Summary section: - **Average Utilization** – Average memory used by YARN applications. The percentage reported is of the total container memory configured for YARN. - **Maximum Utilization** – Maximum memory used by YARN applications. The percentage reported is of the total container memory configured for YARN. Click the drop-down menu next to the date and select **View YARN Applications Running at the Time** to view details about jobs running when maximum utilization occurred. - **Average Daily Peak** – Average daily peak memory used by YARN applications. The number is computed by finding the maximum resource consumption per day and calculating the mean. The percentage reported is of the total container memory configured for YARN. Click **View Time Series Chart** to view a chart of peak utilization.
Utilization by Tenant Displays overall utilization for each tenant. The tenants can be either pools or users. See Configuring the Cluster Utilization Report on page 403 Utilization by tenant is displayed in a table with the following columns: - **Tenant Name** - **Average Allocation** – Average number of vcores allocated to YARN applications of the tenant. The percentage reported is of the total number of vcores configured for YARN. - **Average Utilization** – Average number of vcores used by YARN applications. The percentage reported is of the total number of vcores configured for YARN.	**Utilization by Tenant** Displays overall utilization for each tenant. The tenants can be either pools or users. See Configuring the Cluster Utilization Report on page 403. Utilization by tenant is displayed in a table with the following columns: - **Tenant Name** - **Average Allocation** – Average memory allocated to YARN applications of the tenant. The percentage reported is of the total container memory configured for YARN. - **Average Utilization** – Average memory used by YARN applications. The percentage reported is of the total container memory configured for YARN.

CPU Utilization	Memory Utilization
• **Unused Capacity** – Average unused vcores for the tenant. If this number is high, consider allocating less resources for the applications run by this tenant. Click a column header to sort the table by that column. Click the ❯ icon in the header row of the table to view utilization charts for all tenants. Click ❯ in a row to view CPU utilization for a single tenant.	• **Unused Capacity** – Average unused memory for the tenant. If this number is high, consider allocating less resources for the applications run by this tenant. Click a column header to sort the table by that column. Click the ❯ icon in the header row of the table to view utilization charts for all tenants. Click ❯ in a row to view CPU utilization for a single tenant.

Adjusting YARN Resources

To adjust YARN resources. Go to the YARN service and select **Configuration** > **Category** > **Resource Management** and configure the following properties:

- vcores: **Container Virtual CPU Cores**
- Memory: **Container Memory**

Capacity Planning Tab

The **Capacity Planning Tab** displays a table showing how the weights assigned to YARN Dynamic Resource Pools affect CPU and memory allocations. The table displays the following columns:

- **Tenant Name**
- **CPU Steady Fair Share** – Displays the average number of CPU vcores allocated for each tenant based on the weights assigned to dynamic resource pools.
- **Memory Steady Fair Share** – Displays the average memory allocated for each tenant based on the weights assigned to dynamic resource pools.
- **Wait Ratio During Contention** – The wait ratio is the average percentage of containers in the YARN pool that were pending when there was at least one pending container in the pool. If a pool running critical applications has a high wait ratio, consider increasing the weight of that pool. If several pools in the cluster have a high wait ratio, consider adding more hosts to the cluster.

Click a column header to sort the table by that column.

Preemption Tuning Tab

The **Preemption Tuning** tab displays graphs for each tenant that display the average steady fair share allocations against the average instantaneous fair share allocations and average overall allocations for CPU and memory allocations.

The **CPU** section shows the average allocated vcores, instantaneous fair share of vcores, and steady fair share of vcores whenever the YARN pool was facing contention with resources (times when there was at least one pending container). If the allocated vcores are less than the steady fair share during contention, consider making preemption more aggressive by doing the following:

- Enable fair scheduler preemption.
- Reduce the fair scheduler preemption utilization threshold.
- If you have configured a preemption timeout for a pool (on the **Dynamic Resource Pool Configuration** page (**Clusters** > *cluster name* > **Resource Management** > **Dynamic Resource Pool**), reduce the length of the timeout for pools with a high wait ratio. See

See

The **Memory** section shows the average allocated memory, instantaneous fair share of memory, and steady fair share of memory whenever the YARN pool was facing contention with resources (times when there was at least one pending container). If the allocated memory is less than the Steady Fair Share during contention, consider making preemption more aggressive, as described previously for CPU.

Impala Tab

The **Impala** tab displays CPU and memory utilization for Impala queries using three tabs:

Resource Management

For information about managing Impala resources, see:

Queries Tab

The **Overview** tab displays information about Impala queries.

The top part of the page displays summary information about Impala queries for the entire cluster. The table in the lower part displays the same information by tenant. Both sections display the following:

- **Total** – Total number of queries.

  Click the link with the total to view details and charts about the queries.

- **Avg Wait Time in Queue** – Average time, in milliseconds, spent by a query in an Impala pool while waiting for resources. If this number is high, consider increasing the resources allocated to the pool. If this number is high for several pools, consider increasing the number of hosts in the cluster.

- **Successful** – The number and percentage of queries that finished successfully.

  Click the link with the total to view details and charts about the queries.

- **Memory Limit Exceeded** – Number and percentage of queries that failed due to insufficient memory. If there are such queries, consider increasing the memory allocated to the pool. If there are several pools with such queries, consider increasing the number of hosts in the cluster.

- **Timed Out in Queue** – Number of queries that timed out while waiting for resources in a pool. If there are such queries, consider increasing the maximum number of running queries allowed for the pool. If there are several pools with such queries, consider increasing the number of hosts in the cluster.

- **Rejected** – Number of queries that were rejected by Impala because the pool was full. If this number is high, consider increasing the maximum number of queued queries allowed for the pool. See Admission Control and Query Queuing on page 389.

Click a column header to sort the table by that column.

Peak Memory Usage Tab

This report shows how Impala consumes memory at peak utilization. If utilization is high for a pool, consider adding resources to the pool. If utilization is high for several pools, consider adding more hosts to the cluster.

The **Summary** section of this page displays aggregated peak memory usage information for the entire cluster and the **Utilization by Tenant** section displays peak memory usage by tenant. Both sections display the following:

- **Max Allocated**

  - **Peak Allocation Time** – The time when Impala reserved the maximum amount of memory for queries.

    Click the drop-down list next to the date and time and select **View Impala Queries Running at the Time** to see details about the queries.

  - **Max Allocated** – The maximum memory that was reserved by Impala for executing queries. If the percentage is high, consider increasing the number of hosts in the cluster.

  - **Utilized at the Time** – The amount of memory used by Impala for running queries at the time when maximum memory was reserved.

    Click **View Time Series Chart** to view a chart of peak memory allocations.

  - **Histogram of Allocated Memory at Peak Allocation Time** – Distribution of memory reserved per Impala daemon for executing queries at the time Impala reserved the maximum memory. If some Impala daemons have reserved memory close to the configured limit, consider adding more physical memory to the hosts.

> **Note:** This histogram is generated from the minute-level metrics for Impala daemons. If the minute-level metrics for the timestamp at which peak allocation happened are no longer present in the Cloudera Service Monitor Time-Series Storage, the histogram shows no data. To maintain a longer history for the minute-level metrics, increase the value of the **Time-Series Storage** property for the Cloudera Service Monitor. (Go to the **Cloudera Management Service** > **Configuration** and search for **Time-Series Storage**.)

- **Max Utilized**

  - **Peak Usage Time** – The time when Impala used the maximum amount of memory for queries.

    Click the drop-down list next to the date and time and select **View Impala Queries Running at the Time** to see details about the queries.

  - **Max Utilized** – The maximum memory that was used by Impala for executing queries. If the percentage is high, consider increasing the number of hosts in the cluster.
  - **Reserved at the Time** – The amount of memory reserved by Impala at the time when it was using the maximum memory for executing queries.

    Click **View Time Series Chart** to view a chart of peak memory utilization.

  - **Histogram of Utilized Memory at Peak Usage Time** – Distribution of memory used per Impala daemon for executing queries at the time Impala used the maximum memory. If some Impala daemons are using memory close to the configured limit, consider adding more physical memory to the hosts.

> **Note:** This histogram is generated from the minute-level metrics for Impala daemons. If the minute-level metrics for the timestamp at which peak allocation happened are no longer present in the Cloudera Service Monitor Time-Series Storage, the histogram shows no data. To maintain a longer history for the minute-level metrics, increase the value of the **Time-Series Storage** property for the Cloudera Service Monitor. (Go to the **Cloudera Management Service** > **Configuration** and search for **Time-Series Storage**.)

Spilled Memory Tab

The **Spilled Memory** tab displays information about Impala spilled memory. These disk spills can deteriorate the performance of Impala queries significantly. This report shows the amount of disk spills for Impala queries by tenant. If disk spill is high for a pool, consider adding resources to the pool. If disk spill is high for several pools, consider adding more hosts to the cluster.

For each tenant, the following are displayed:

- **Average Spill** – Average spill per query
- **Maximum Spill** – Maximum memory spilled per hour

Downloading Cluster Utilization Reports Using the Cloudera Manager API

You can download the Cluster Utilization Reports as a JSON file using the Cloudera Manager API. See the [Cloudera Manager REST API documentation](#) for the following API endpoints:

- **Cluster Utilization**: `path__clusters_-clusterName-_utilization.html`
- **Impala Utilization:** `path__clusters_-clusterName-_impalaUtilization.html`
- **YARN Utilization**: `path__clusters_-clusterName-_yarnUtilization.html`

Creating a Custom Cluster Utilization Report

Cloudera Manager provides a [Cluster Utilization Report](#) that displays aggregated utilization information for YARN and Impala jobs. If you wish to export the data from this report, you can build custom reports based on the same metrics data using the Cloudera Manager Admin console or the Cloudera Manager API. This topic describes the metrics and

queries you can use to build these custom reports. These reports all use the [tsquery Language](#) on page 315 to [chart time-series data](#).

Metrics and Queries

For more information about the *Data Granularity* described in these metrics, see [Metric Aggregation](#) on page 324.

Many of the metrics described below use a data granularity of *hourly*. This is not required, but is recommended because some of the YARN utilization metrics are only available hourly and using the hourly granularity allows for consistent reporting.

Cluster-Level CPU and Memory Metrics

Total cluster CPU usage

   Data Granularity: hourly

   Units: percentage

   tsquery:

```
SELECT
    cpu_percent_across_hosts
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Total CPU Cores in the cluster

   Data Granularity: hourly

   Units: CPU cores

   tsquery:

```
SELECT
    total_cores_across_hosts
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Total cluster memory usage

   Data Granularity: hourly

   Units: percentage

   tsquery:

```
SELECT
  100 * total_physical_memory_used_across_hosts/total_physical_memory_total_across_hosts
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Total cluster memory usage

   Time series of total cluster memory usage.

   Data Granularity:hourly

   Units: Byte seconds

   tsquery:

```
SELECT
    total_physical_memory_total_across_hosts
WHERE
```

```
    category=CLUSTER
    AND clusterName=Cluster_Name
```

CPU used by Impala

Time series of total Impala CPU usage in milliseconds.

Data Granularity: hourly

Units: milliseconds

tsquery:

```
SELECT
    counter_delta(impala_query_thread_cpu_time_rate)
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Memory used by Impala

Time series of Impala memory usage

Data Granularity: hourly

Units: byte seconds

tsquery:

```
SELECT
    counter_delta(impala_query_memory_accrual_rate)
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

CPU used by YARN

The `yarn_reports_containers_used_cpu` metric used in this tsquery is generated per hour, therefore the data granularity used for this query is the raw metric value.

Data Granularity: Raw

Units: percent seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_cpu FROM REPORTS
WHERE
    category=SERVICE
    AND clusterName=Cluster_Name
```

Memory used by YARN

Yarn memory usage. The `yarn_reports_containers_used_memory` metric used in this tsquery is generated per hour, therefore the data granularity used for this query is the raw metric value.

Data Granularity: raw metric value

Units: megabyte seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_memory
FROM
    REPORTS
WHERE
```

```
category=SERVICE
AND clusterName=Cluster_Name
```

Pool-Level CPU and Memory Metrics

CPU used by Impala pool

CPU usage for an Impala pool.

Data Granularity: hourly

Units: milliseconds

tsquery:

```
SELECT
    counter_delta(impala_query_thread_cpu_time_rate)
WHERE
    category=IMPALA_POOL
    AND poolName=Pool_Name
```

Memory used by Impala pool

Data Granularity: hourly

Units: byte seconds

tsquery:

```
SELECT
    counter_delta(impala_query_memory_accrual_rate)
WHERE
    category=IMPALA_POOL
    AND poolName=Pool_Name
```

CPU used by YARN pool

Provides CPU metrics per YARN pool and user. You can aggregate a pool-level metric from this query.

Data Granularity: Raw

Units: percent seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_cpu FROM REPORTS
WHERE
    category=YARN_POOL_USER
```

Memory used by YARN pool

Provides memory metrics per YARN pool and user. You can aggregate a pool-level metric from this query.

Data Granularity: hourly

Units: megabyte seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_memory
FROM
    REPORTS
WHERE
    category=YARN_POOL_USER
```

YARN Metrics

YARN VCore usage

Data Granularity: Raw

Units: VCore seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_vcores
FROM
    REPORTS
WHERE
    category=SERVICE
    AND clusterName=Cluster_Name
```

Total VCores available to YARN

Data Granularity: hourly

Units: Number of VCores (Note that this value is not multiplied by the time unit.)

tsquery:

```
SELECT
    total_allocated_vcores_across_yarn_pools + total_available_vcores_across_yarn_pools
WHERE
    category=SERVICE
    AND clusterName=Cluster_Name
```

YARN Memory usage

Data Granularity: Raw

Units: MB seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_memory FROM REPORTS
WHERE
    category=SERVICE
    AND clusterName=Cluster_Name
```

Total memory available to YARN

Data Granularity: hourly

Units: MB (Note that this value is not multiplied by the time unit.)

tsquery:

```
SELECT
    total_available_memory_mb_across_yarn_pools +
total_allocated_memory_mb_across_yarn_pools
WHERE
    category=SERVICE
    AND clusterName=Cluster_Name
```

Pool-level VCore usage

The results of this query return the usage for each user in each pool. To see the total usage for a pool, sum all users of the pool.

Data Granularity: Raw

Units: VCore seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_vcores FROM REPORTS
WHERE
    category=YARN_POOL_USER
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level memory usage

The results of this query return the usage for each user in each pool. To see the total usage for a pool, sum all users of the pool.

Data Granularity: Raw

Units: MB seconds

tsquery:

```
SELECT
    yarn_reports_containers_used_memory FROM REPORTS
WHERE
    category=YARN_POOL_USER
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level allocated VCores

The results of this query return the usage for each user in each pool. To see the total usage for a pool, sum all users of the pool.

Data Granularity: raw metric value

Units: VCore seconds

tsquery:

```
SELECT
    yarn_reports_containers_allocated_vcores FROM REPORTS
WHERE
    category=YARN_POOL_USER
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level allocated memory

The results of this query return the usage for each user in each pool. To see the total usage for a pool, sum all users of the pool.

Data Granularity: raw metric value

Units: megabyte seconds

tsquery:

```
SELECT
    yarn_reports_containers_allocated_memory
FROM
    REPORTS
WHERE
    category=YARN_POOL_USER
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level steady fair share VCore

Data Granularity: hourly

Units: VCores

tsquery:

```
SELECT
    steady_fair_share_vcores
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level fair share VCore

Data Granularity: hourly

Units: VCores

tsquery:

```
SELECT
    fair_share_vcores
WHERE
    category=YARN_POOL
```

Pool-level steady fair share memory

Data Granularity: hourly

Units: MB

tsquery:

```
SELECT
    steady_fair_share_mb
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool-level fair share memory

Data Granularity: hourly

Units: MB

tsquery:

```
SELECT
    fair_share_mb
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Metric indicating contention

Data Granularity: hourly

Units: percentage

tsquery:

```
SELECT
    container_wait_ratio
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Use the following metrics to monitor resource contention.

Pool-level allocated VCores when contention occurs

> **Data Granularity:** hourly
>
> **Units:** VCores
>
> **tsquery:**

```
SELECT
    allocated_vcores_with_pending_containers
WHERE
    category=YARN_POOL
```

> To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool level steady fair share VCores when contention occurs

> **Data Granularity:** hourly
>
> **Units:** VCores
>
> **tsquery:**

```
SELECT
    steady_fair_share_vcores_with_pending_containers
WHERE
    category=YARN_POOL
```

> To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool level fair share VCores when contention occurs

> **Data Granularity:** hourly
>
> **Units:** VCores
>
> **tsquery:**

```
SELECT
    fair_share_vcores_with_pending_containers
WHERE
    category=YARN_POOL
```

> To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool level allocated memory when contention occurs

> **Data Granularity:** hourly
>
> **Units:** MB
>
> **tsquery:**

```
SELECT
    allocated_memory_mb_with_pending_containers
WHERE
    category=YARN_POOL
```

> To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool level steady fair share memory when contention occurs

> **Data Granularity:** hourly
>
> **Units:** MB

tsquery:

```
SELECT
    steady_fair_share_mb_with_pending_containers
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Pool level fair share memory when contention occurs

Data Granularity: hourly

Units: MB

tsquery:

```
SELECT
    fair_share_mb_with_pending_containers
WHERE
    category=YARN_POOL
```

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Impala-Specific Metrics

To view metrics for a specific pool, add `poolName=Pool Name` to the tsquery statement.

Total reserved memory

Data Granularity: hourly

Units: MB seconds

tsquery:

```
SELECT
    total_impala_admission_controller_local_backend_mem_reserved_across_impala_daemon_pools
WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Total used memory

Data Granularity: hourly

Units: MB seconds

tsquery:

```
SELECT
    total_impala_admission_controller_local_backend_mem_usage_across_impala_daemon_pools

WHERE
    category=CLUSTER
    AND clusterName=Cluster_Name
```

Total available memory

Data Granularity: hourly

Units: MB seconds

tsquery:

```
SELECT
    total_mem_tracker_process_limit_across_impalads
WHERE
```

```
       category=CLUSTER
       AND clusterName=Cluster_Name
```

> **Note:** To query for pool-level metrics, change the `category` to `IMPALA-POOL` in the above tsquery statements.

Impala Query Counter Metrics

Include the following in the `SELECT` statement of the tsquery to get information about the rate of Impala queries:

- `counter_delta(queries_ingested_rate)`
- `counter_delta(queries_successful_rate)`
- `counter_delta(queries_rejected_rate)`
- `counter_delta(queries_oom_rate)`
- `counter_delta(queries_timed_out_rate)`
- `counter_delta(impala_query_admission_wait_rate)`
- `counter_delta(impala_query_memory_spilled_rate)`

For example:

```
SELECT
    counter_delta(queries_ingested_rate)
 WHERE
    category=IMPALA_POOL
    AND clusterName=Cluster_Name
    AND serviceName=Service_Name
```

Calculations for reports

All the metrics listed in this topic return a time series of metric values. Depending on the collection frequency of the metric itself and the data granularity you use when issuing tsquery statements, the results return metric values in different frequencies and therefore there are different ways to handle the metric values.

Note the following about how to correctly perform calculations using metric values:

- YARN container metrics are generated once per hour resulting in one raw metric value every hour. Therefore, the most detailed results possible for YARN CPU and memory usage are hourly reports.

- Hourly aggregates are summarized from raw metric values. These aggregates include a set of statistics that include the sum, maximum, minimum, count and other statistics that summarize the raw metric values. When you use the hourly granularity, you lose the single values of the raw metric values. However, you can still get peak usage data for such metrics. For more information, see Metric Aggregation on page 324.

- For some of the YARN metrics described in this topic, the tsquery statement aggregates from the pool and user level to pool level in the Cloudera Manager Cluster Utilization reports. For these queries, because the maximum and minimum for different pool and user combinations are not likely to happen at the same time, there is no way to get the peak usage across pool and user combinations, or at the pool level. The only meaningful results possible are average and sum.

- When calculating CPU/Memory usage percentage, pay attention to the units for each metric. For example, if the cluster consistently has 8 VCores, the total VCore seconds for each hour would be 8 * 3600 VCore seconds. You can then use this adjusted number to compare with the VCore seconds used by YARN or YARN pools.

Retrieving metric data using the Cloudera Manager API

There is a Time series endpoint exposed by the Cloudera Manager REST API. See Cloudera Manager API documentation timeseries Endpoint. The API accepts tsquery statements as input for which metrics need to be retrieved during the specified time window. The API provides functionality to specify the desired data granularity (for example, raw metric values, TEN_MINUTES, HOURLY etc.). Each granularity level of data is maintained in a leveldb table (see Data Granularity

on page 398). This data is aggregated from raw metric values such as minimum, maximum, etc. within the corresponding data window.

For example, if you do not need the metric data at a specific timestamp but care more about the hourly usage, HOURLY data should be good enough. In general, the longer the granular window it is, the less storage it is taking, and thus the longer period of time you are able to keep that level of data without being purged when the storage hits the configured limit. In the case of Cloudera Manager Cluster Utilization Reports, Cloudera Manager generates the reports based on an hourly window.

To view the Cloudera Manager Service Monitor data storage granularities, go to **Clusters** > **Cloudera Management Service** > **Service Monitor** > **Charts Library** > **Service Monitor Storage** and scroll down to see the **Data Duration Covered** table to see the earliest available data points for each level of granularity. The value in the **last(duration_covered)** column indicates the age of the oldest data in the table.

Data Duration Covered

September 14, 2016 3:40 PM	Search
⬥ Entity	last(duration_covered)
impala-query-monitoring - profiles (RAW)	9.4h
impala-query-monitoring - profiles_end_time (RAW)	9.4h
impala-query-monitoring - queries (RAW)	9.4h
service-monitoring - reports_stream (RAW)	9.4h
service-monitoring - reports_type (RAW)	9.4h
service-monitoring - stream (RAW)	9.4h
service-monitoring - subject_ts (RAW)	9.4h
service-monitoring - ts_stream_rollup_PT21600S (SIX_HOURLY)	9.4h
service-monitoring - ts_stream_rollup_PT3600S (HOURLY)	9.4h
service-monitoring - ts_stream_rollup_PT600S (TEN_MINUTELY)	9.4h
service-monitoring - ts_stream_rollup_PT604800S (WEEKLY)	9.4h
service-monitoring - ts_stream_rollup_PT86400S (DAILY)	9.4h
service-monitoring - ts_subject (RAW)	9.4h
service-monitoring - ts_type_rollup_PT21600S (SIX_HOURLY)	9.4h
service-monitoring - ts_type_rollup_PT3600S (HOURLY)	9.4h
service-monitoring - ts_type_rollup_PT600S (TEN_MINUTELY)	9.4h
service-monitoring - ts_type_rollup_PT604800S (WEEKLY)	9.4h
service-monitoring - ts_type_rollup_PT86400S (DAILY)	9.4h
service-monitoring - type (RAW)	9.4h
yarn-application-monitoring - application_details (RAW)	9.4h
yarn-application-monitoring - applications (RAW)	9.4h
yarn-application-monitoring - applications_end_time (RAW)	9.4h

To configure the Time series storage used by the Service Monitor, go to **Clusters** > **Cloudera Management Service** > **Configuration** > **Charts Library** > **Service Monitor Storage** and search for "Time-Series Storage".

Resource Management

Querying metric data using the Cloudera Manager Admin Console

To build charts that query time series data using the Cloudera Manager Admin console, go to **Charts > Chart Builder**. When building charts, it may be useful to choose the data granularity by clicking the **Show additional options** link on the chart builder page and then selecting the **Data Granularity** drop-down list. See Charting Time-Series Data on page 303.

Selecting data granularity in chart builder:

High Availability

This guide is for Apache Hadoop system administrators who want to enable continuous availability by configuring clusters without single points of failure.

Not all Hadoop components currently support highly availability configurations. However, some currently SPOF (single point of failure) components can be configured to restart automatically in the event of a failure (**Auto-Restart Configurable**, in the table below). Some components support high availability implicitly because they comprise distributed processes (identified with an asterisk (*) in the table). In addition, some components depend on external databases which must also be configured to support high availability.

High Availability	Auto-Restart Configurable	Components with External Databases
Alert Publisher	Hive Metastore	Activity Monitor
Cloudera Manager Agent*	Impala catalog service	Cloudera Navigator Audit Server
Cloudera Manager Server	Impala statestore	Cloudera Navigator Metadata Server
Data Node*	Sentry Service	Hive Metastore Server
Event Server	Spark Job History Server	Oozie Server
Flume*	YARN Job History Server	Reports Manager
HBase Master		Sentry Server
Host Monitor		Sqoop Server
Hue (add multiple services, use load balancer)		
Impalad* (add multiple services, use load balancer)		
NameNode		
Navigator Key Trustee		
Node Manager*		
Oozie Server		
RegionServer*		
Reports Manager		
Resource Manager		
Service Monitor		
Solr Server*		
Zookeeper server*		

HDFS High Availability

This section provides an overview of the HDFS high availability (HA) feature and how to configure and manage an HA HDFS cluster.

Introduction to HDFS High Availability

This section assumes that the reader has a general understanding of components in an HDFS cluster. For details, see the Apache HDFS Architecture Guide.

Background

In a standard configuration, the NameNode is a single point of failure (SPOF) in an HDFS cluster. Each cluster has a single NameNode, and if that host or process became unavailable, the cluster as a whole is unavailable until the NameNode is either restarted or brought up on a new host. The Secondary NameNode does not provide failover capability.

The standard configuration reduces the total availability of an HDFS cluster in two major ways:

- In the case of an unplanned event such as a host crash, the cluster is unavailable until an operator restarts the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine result in periods of cluster downtime.

HDFS HA addresses the above problems by providing the option of running two NameNodes in the same cluster, in an active/passive configuration. These are referred to as the active NameNode and the standby NameNode. Unlike the Secondary NameNode, the standby NameNode is hot standby, allowing a fast automatic failover to a new NameNode in the case that a host crashes, or a graceful administrator-initiated failover for the purpose of planned maintenance. You cannot have more than two NameNodes.

Implementation

Cloudera Manager and CDH support Quorum-based Storage to implement HA.

Quorum-based Storage

Quorum-based Storage refers to the HA implementation that uses a Quorum Journal Manager (QJM).

For the standby NameNode to keep its state synchronized with the active NameNode in this implementation, both nodes communicate with a group of separate daemons called JournalNodes. When any namespace modification is performed by the active NameNode, it durably logs a record of the modification to a majority of the JournalNodes. The standby NameNode is capable of reading the edits from the JournalNodes, and is constantly watching them for changes to the edit log. As the standby Node sees the edits, it applies them to its own namespace. In the event of a failover, the standby ensures that it has read all of the edits from the JournalNodes before promoting itself to the active state. This ensures that the namespace state is fully synchronized before a failover occurs.

To provide a fast failover, it is also necessary that the standby NameNode has up-to-date information regarding the location of blocks in the cluster. To achieve this, DataNodes are configured with the location of both NameNodes, and they send block location information and heartbeats to both.

It is vital for the correct operation of an HA cluster that only one of the NameNodes be active at a time. Otherwise, the namespace state would quickly diverge between the two, risking data loss or other incorrect results. To ensure this property and prevent the so-called "split-brain scenario," JournalNodes only ever allow a single NameNode to be a writer at a time. During a failover, the NameNode which is to become active simply takes over the role of writing to the JournalNodes, which effectively prevents the other NameNode from continuing in the active state, allowing the new active NameNode to safely proceed with failover.

> **Note:** Because of this, fencing is not required, but it is still useful; see Enabling HDFS HA on page 424.

Automatic Failover

Automatic failover relies on two additional components in an HDFS: a ZooKeeper quorum, and the `ZKFailoverController` process (abbreviated as ZKFC). In Cloudera Manager, the ZKFC process maps to the HDFS Failover Controller role.

Apache ZooKeeper is a highly available service for maintaining small amounts of coordination data, notifying clients of changes in that data, and monitoring clients for failures. The implementation of HDFS automatic failover relies on ZooKeeper for the following functions:

- **Failure detection** - each of the NameNode machines in the cluster maintains a persistent session in ZooKeeper. If the machine crashes, the ZooKeeper session will expire, notifying the other NameNode that a failover should be triggered.
- **Active NameNode election** - ZooKeeper provides a simple mechanism to exclusively elect a node as active. If the current active NameNode crashes, another node can take a special exclusive lock in ZooKeeper indicating that it should become the next active NameNode.

The `ZKFailoverController` (ZKFC) is a ZooKeeper client that also monitors and manages the state of the NameNode. Each of the hosts that run a NameNode also run a ZKFC. The ZKFC is responsible for:

- **Health monitoring** - the ZKFC contacts its local NameNode on a periodic basis with a health-check command. So long as the NameNode responds promptly with a healthy status, the ZKFC considers the NameNode healthy. If the NameNode has crashed, frozen, or otherwise entered an unhealthy state, the health monitor marks it as unhealthy.
- **ZooKeeper session management** - when the local NameNode is healthy, the ZKFC holds a session open in ZooKeeper. If the local NameNode is active, it also holds a special lock `znode`. This lock uses ZooKeeper's support for "ephemeral" nodes; if the session expires, the lock node is automatically deleted.
- **ZooKeeper-based election** - if the local NameNode is healthy, and the ZKFC sees that no other NameNode currently holds the lock `znode`, it will itself try to acquire the lock. If it succeeds, then it has "won the election", and is responsible for running a failover to make its local NameNode active. The failover process is similar to the manual failover described above: first, the previous active is fenced if necessary, and then the local NameNode transitions to active state.

General Questions about HDFS HA

What does the message "Operation category READ/WRITE is not supported in state standby" mean?

In an HA-enabled cluster, DFS clients cannot know in advance which NameNode is active at a given time. So when a client contacts a NameNode and it happens to be the standby, the READ or WRITE operation will be refused and this message is logged. The client will then automatically contact the other NameNode and try the operation again. As long as there is one active and one standby NameNode in the cluster, this message can be safely ignored.

If an application is configured to contact only one NameNode always, this message indicates that the application is failing to perform any read/write operation. In such situations, the application would need to be modified to use the HA configuration for the cluster. The Jira [HDFS-3447](#) deals with lowering the severity of this message (and similar ones) to DEBUG so as to reduce noise in the logs but is unresolved as of October 2018.

Configuring Hardware for HDFS HA

To deploy an HA cluster using Quorum-based Storage, you should prepare the following:

- NameNode hosts - These are the hosts on which you run the active and standby NameNodes. They should have equivalent hardware to each other, and equivalent hardware to what would be used in a non-HA cluster.
- JournalNode hosts - These are the hosts on which you run the JournalNodes. Cloudera recommends that you deploy the JournalNode daemons on the "master" host or hosts (NameNode, Standby NameNode, JobTracker, and so on) so the JournalNodes' local directories can use the reliable local storage on those machines.
- If co-located on the same host, each JournalNode process and each NameNode process should have its own dedicated disk. You should not use SAN or NAS storage for these directories.
- There must be at least three JournalNode daemons, since edit log modifications must be written to a majority of JournalNodes. This will allow the system to tolerate the failure of a single host. You can also run more than three JournalNodes, but to actually increase the number of failures the system can tolerate, you should run an odd number of JournalNodes, (three, five, seven, and so on). Note that when running with N JournalNodes, the system

can tolerate at most (N - 1) / 2 failures and continue to function normally. If the requisite quorum is not available, the NameNode will not format or start, and you will see an error similar to this:

```
12/10/01 17:34:18 WARN namenode.FSEditLog: Unable to determine input streams from QJM
to [10.0.1.10:8485, 10.0.1.10:8486, 10.0.1.10:8487]. Skipping.
java.io.IOException: Timed out waiting 20000ms for a quorum of nodes to respond.
```

> **Note:** In an HA cluster, the standby NameNode also performs checkpoints of the namespace state, and thus it is not necessary to run a Secondary NameNode, CheckpointNode, or BackupNode in an HA cluster. In fact, to do so would be an error. If you are reconfiguring a non-HA-enabled HDFS cluster to be HA-enabled, you can reuse the hardware which you had previously dedicated to the Secondary NameNode.

Enabling HDFS HA

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

An HDFS high availability (HA) cluster uses two NameNodes—an active NameNode and a standby NameNode. Only one NameNode can be active at any point in time. HDFS HA depends on maintaining a log of all namespace modifications in a location available to both NameNodes, so that in the event of a failure, the standby NameNode has up-to-date information about the edits and location of blocks in the cluster.

> **Important:** Enabling and disabling HA causes a service outage for the HDFS service and *all services* that depend on HDFS. Before enabling or disabling HA, ensure that there are no jobs running on your cluster.

Enabling HDFS HA Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

You can use Cloudera Manager to configure your CDH cluster for HDFS HA and automatic failover. In Cloudera Manager, HA is implemented using Quorum-based storage. Quorum-based storage relies upon a set of JournalNodes, each of which maintains a local edits directory that logs the modifications to the namespace metadata. Enabling HA enables automatic failover as part of the same command.

> **Important:**
> - Enabling or disabling HA causes the previous monitoring history to become unavailable.
> - Some parameters will be automatically set as follows once you have enabled JobTracker HA. If you want to change the value from the default for these parameters, use an advanced configuration snippet.
>
>   - `mapred.jobtracker.restart.recover: true`
>   - `mapred.job.tracker.persist.jobstatus.active: true`
>   - `mapred.ha.automatic-failover.enabled: true`
>   - `mapred.ha.fencing.methods: shell(true)`

Enabling High Availability and Automatic Failover

The **Enable High Availability** workflow leads you through adding a second (standby) NameNode and configuring JournalNodes.

1. Perform all the configuration and setup tasks described under Configuring Hardware for HDFS HA on page 423.

2. Ensure that you have a ZooKeeper service.

3. Go to the HDFS service.

4. Select **Actions** > **Enable High Availability**. A screen showing the hosts that are eligible to run a standby NameNode and the JournalNodes displays.

   a. Specify a name for the nameservice and click **Continue**.

   b. In the **NameNode Hosts** field, click **Select a host**. The host selection dialog box displays.

   c. Check the checkbox next to the hosts where you want the standby NameNode to be set up and click **OK**. The standby NameNode cannot be on the same host as the active NameNode, and the host that is chosen should have the same hardware configuration (RAM, disk space, number of cores, and so on) as the active NameNode.

   d. In the **JournalNode Hosts** field, click **Select hosts**. The host selection dialog box displays.

   e. Check the checkboxes next to an odd number of hosts (a minimum of three) to act as JournalNodes and click **OK**. JournalNodes should be hosted on hosts with similar hardware specification as the NameNodes. Cloudera recommends that you put a JournalNode each on the same hosts as the active and standby NameNodes, and the third JournalNode on similar hardware, such as the JobTracker.

   f. Click **Continue**.

   g. In the **JournalNode Edits Directory** property, enter a directory location for the JournalNode edits directory into the fields for each JournalNode host.

      - You may enter only one directory for each JournalNode. The paths do not need to be the same on every JournalNode.
      - The directories you specify should be empty.
      - The directory owner should be hdfs:hadoop and must have read, write, and execute permission (`drwx------`).

   h. **Extra Options:** Decide whether Cloudera Manager should clear existing data in ZooKeeper, standby NameNode, and JournalNodes. If the directories are not empty (for example, you are re-enabling a previous HA configuration), Cloudera Manager will not automatically delete the contents—you can select to delete the contents by keeping the default checkbox selection. The recommended default is to clear the directories. If you choose not to do so, the data should be in sync across the edits directories of the JournalNodes and should have the same version data as the NameNodes.

   i. Click **Continue**.

   Cloudera Manager executes a set of commands that stop the dependent services, delete, create, and configure roles and directories as required, create a nameservice and failover controller, restart the dependent services, and deploy the new client configuration.

   > **Important:** Some steps, such as formatting the NameNode may report failure if the action was already completed. However, the configuration steps continue to execute after reporting non-critical failed steps.

5. If you want to use other services in a cluster with HA configured, follow the procedures in .

> **Important:** If you change the NameNode Service RPC Port (`dfs.namenode.servicerpc-address`) while automatic failover is enabled, this will cause a mismatch between the NameNode address saved in the ZooKeeper `/hadoop-ha` znode and the NameNode address that the Failover Controller is configured with. This will prevent the Failover Controllers from restarting. If you need to change the NameNode Service RPC Port after Auto Failover has been enabled, you must do the following to re-initialize the znode:
>
> 1. Stop the HDFS service.
> 2. Configure the service RPC port:
>   a. Go to the HDFS service.
>   b. Click the **Configuration** tab.
>   c. Select **Scope** > **NameNode**.
>   d. Select **Category** > **Ports and Addresses**.
>   e. Locate the **NameNode Service RPC Port** property or search for it by typing its name in the Search box.
>   f. Change the port value as needed.
>
>   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See Modifying Configuration Properties Using Cloudera Manager on page 60.
>
> 3. On a ZooKeeper server host, run `zookeeper-client`.
>   a. Execute the following to remove the configured nameservice. This example assumes the name of the nameservice is **nameservice1**. You can identify the nameservice from the **Federation and High Availability** section on the HDFS **Instances** tab:
>
>    ```
>    rmr /hadoop-ha/nameservice1
>    ```
>
> 4. Click the **Instances** tab.
> 5. Select **Actions** > **Initialize High Availability State in ZooKeeper**.
> 6. Start the HDFS service.

Fencing Methods

To ensure that only one NameNode is active at a time, a fencing method is required for the shared edits directory. During a failover, the fencing method is responsible for ensuring that the previous active NameNode no longer has access to the shared edits directory, so that the new active NameNode can safely proceed writing to it.

By default, Cloudera Manager configures HDFS to use a shell fencing method (`shell(true)`).

The fencing parameters are found in the **Service-Wide** > **High Availability** category under the configuration properties for your HDFS service.

Fencing Configuration

`dfs.ha.fencing.methods` - a list of scripts or Java classes which will be used to fence the active NameNode during a failover

It is desirable for correctness of the system that only one NameNode be in the active state at any given time.

When you use Quorum-based Storage, only one NameNode will ever be allowed to write to the JournalNodes, so there is no potential for corrupting the file system metadata in a "split-brain" scenario. This is reflected in the default value of `shell(true)` for the `dfs.ha.fencing.methods`, which does not explicitly try to fence the standby NameNode.

In the absence of explicitly fencing, there is a narrow time window where the previously active NameNode may serve out-of-date responses to reads from clients. This window ends when the previously active NameNode tries to write to the JournalNodes, at which point the NameNode shuts down.

This window of stale read responses is rarely an issue for applications since there is no danger of split-brain corruption. In rare or special cases where strong read consistency is required, use an explicit fencing method such as the `agent-based fencer`.

> **Note:** If you choose to use the agent-based fencing method, you should still configure something `shell(true)` as a fallback fencing option since agent-based fencing fails if the other NameNode is unresponsive.

The fencing methods used during a failover are configured as a carriage-return-separated list, and these will be attempted in order until one of them indicates that fencing has succeeded.

For information on implementing your own custom fencing method, see the `org.apache.hadoop.ha.NodeFencer` class.

Configuring the shell fencing method

`shell` - run an arbitrary shell command to fence the active NameNode

The shell fencing method runs an arbitrary shell command, which you can configure as shown below:

```
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>shell(/path/to/my/script.sh arg1 arg2 ...)</value>
</property>
```

The string between '(' and ')' is passed directly to a `bash` shell and cannot include any closing parentheses.

When executed, the first argument to the configured script will be the address of the NameNode to be fenced, followed by all arguments specified in the configuration.

The shell command will be run with an environment set up to contain all of the current Hadoop configuration variables, with the '_' character replacing any '.' characters in the configuration keys. The configuration used has already had any NameNode-specific configurations promoted to their generic forms - for example `dfs_namenode_rpc-address` will contain the RPC address of the target node, even though the configuration may specify that variable as `dfs.namenode.rpc-address.ns1.nn1`.

The following variables referring to the target node to be fenced are also available:

Variable	Description
`$target_host`	Hostname of the node to be fenced
`$target_port`	IPC port of the node to be fenced
`$target_address`	The above two variables, combined as *host:port*
`$target_nameserviceid`	The nameservice ID of the NameNode to be fenced
`$target_namenodeid`	The NameNode ID of the NameNode to be fenced

You can also use these environment variables as substitutions in the shell command itself. For example:

```
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>shell(/path/to/my/script.sh --nameservice=$target_nameserviceid
$target_host:$target_port)</value>
</property>
```

If the shell command returns an exit code of 0, the fencing is determined to be successful. If it returns any other exit code, the fencing was not successful and the next fencing method in the list will be attempted.

> **Note:** This fencing method does not implement any timeout. If timeouts are necessary, they should be implemented in the shell script itself (for example, by forking a subshell to kill its parent in some number of seconds).

Disabling and Redeploying HDFS HA

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

1. Go to the HDFS service.
2. Select **Actions** > **Disable High Availability**.
3. Select the hosts for the NameNode and the SecondaryNameNode and click **Continue**.
4. Select the HDFS checkpoint directory and click **Continue**.
5. Confirm that you want to take this action.
6. Configuring the Hive Metastore to Use HDFS HA on page 428.

Cloudera Manager ensures that one NameNode is active, and saves the namespace. Then it stops the standby NameNode, creates a SecondaryNameNode, removes the standby NameNode role, and restarts all the HDFS services.

Configuring Other CDH Components to Use HDFS HA

You can use the HDFS high availability NameNodes with other components of CDH.

Configuring HBase to Use HDFS HA

If you configure HBase to use an HA-enabled HDFS instance, Cloudera Manager automatically handles HA configuration for you.

Configuring the Hive Metastore to Use HDFS HA

The Hive metastore can be configured to use HDFS high availability by using Cloudera Manager or by using the command-line for unmanaged clusters.

1. In the Cloudera Manager Admin Console, go to the Hive service.
2. Select **Actions** > **Stop**.

> **Note:** You may want to stop the Hue and Impala services first, if present, as they depend on the Hive service.

   Click **Stop** again to confirm the command.
3. Back up the Hive metastore database.
4. Select **Actions** > **Update Hive Metastore NameNodes** and confirm the command.
5. Select **Actions** > **Start** and click **Start** to confirm the command.
6. Restart the Hue and Impala services if you stopped them prior to updating the metastore.

Configuring Hue to Work with HDFS HA Using Cloudera Manager

1. Add the HttpFS role.
2. After the command has completed, go to the **Hue** service.
3. Click the **Configuration** tab.
4. Locate the **HDFS Web Interface Role** property or search for it by typing its name in the Search box.
5. Select the **HttpFS** role you just created instead of the NameNode role, and save your changes.
6. Restart the Hue service.

Configuring Impala to Work with HDFS HA

1. Complete the steps to reconfigure the Hive metastore database, as described in the preceding section. Impala shares the same underlying database with Hive, to manage metadata for databases, tables, and so on.

2. Issue the `INVALIDATE METADATA` statement from an Impala shell. This one-time operation makes all Impala daemons across the cluster aware of the latest settings for the Hive metastore database. Alternatively, restart the Impala service.

Configuring Oozie to Use HDFS HA

To configure an Oozie workflow to use HDFS HA, use the HDFS nameservice instead of the NameNode URI in the `<name-node>` element of the workflow.

Example:

```
<action name="mr-node">
  <map-reduce>
    <job-tracker>${jobTracker}</job-tracker>
    <name-node>hdfs://ha-nn
```

where *ha-nn* is the value of `dfs.nameservices` in `hdfs-site.xml`.

Administering an HDFS High Availability Cluster

Manually Failing Over to the Standby NameNode

If you are running a HDFS service with HA enabled, you can manually cause the active NameNode to failover to the standby NameNode. This is useful for planned downtime—for hardware changes, configuration changes, or software upgrades of your primary host.

1. Go to the HDFS service.
2. Click the **Instances** tab.
3. Click **Federation and High Availability**.
4. Locate the row for the Nameservice where you want to fail over the NameNode.
5. Select **Actions** > **Manual Failover**. (This option does not appear if HA is not enabled for the cluster.)
6. From the pop-up, select the NameNode that should be made active, then click **Manual Failover**.

> **Note: For advanced use only:** You can set the **Force Failover** checkbox to force the selected NameNode to be active, irrespective of its state or the other NameNode's state. Forcing a failover will first attempt to failover the selected NameNode to active mode and the other NameNode to standby mode. It will do so even if the selected NameNode is in safe mode. If this fails, it will proceed to transition the selected NameNode to active mode. To avoid having two NameNodes be active, use this only if the other NameNode is either definitely stopped, or can be transitioned to standby mode by the first failover step.

7. When all the steps have been completed, click **Finish**.

Cloudera Manager transitions the NameNode you selected to be the active NameNode, and the other NameNode to be the standby NameNode. HDFS should *never* have two active NameNodes.

Moving an HA NameNode to a New Host

See <u>Moving Highly Available NameNode, Failover Controller, and JournalNode Roles Using the Migrate Roles Wizard</u> on page 93.

Other HDFS haadmin Commands

After your HA NameNodes are configured and started, you will have access to some additional commands to administer your HA HDFS cluster. Specifically, you should familiarize yourself with the subcommands of the `hdfs haadmin` command.

This page describes high-level uses of some important subcommands. For specific usage information of each subcommand, you should run `hdfs haadmin -help <command>`.

getServiceState

`getServiceState` - determine whether the given NameNode is active or standby

Connect to the provided NameNode to determine its current state, printing either "standby" or "active" to STDOUT as appropriate. This subcommand might be used by `cron` jobs or monitoring scripts which need to behave differently based on whether the NameNode is currently active or standby.

checkHealth

`checkHealth` - check the health of the given NameNode

Connect to the provided NameNode to check its health. The NameNode is capable of performing some diagnostics on itself, including checking if internal services are running as expected. This command will return 0 if the NameNode is healthy, non-zero otherwise. One might use this command for monitoring purposes.

Using the dfsadmin Command When HA is Enabled

By default, applicable `dfsadmin` command options are run against both active and standby NameNodes. To limit an option to a specific NameNode, use the `-fs` option. For example,

To turn safe mode on for both NameNodes, run:

```
hdfs dfsadmin -safemode enter
```

To turn safe mode on for a single NameNode, run:

```
hdfs dfsadmin -fs hdfs://<host>:<port> -safemode enter
```

For a full list of `dfsadmin` command options, run: `hdfs dfsadmin -help`.

Converting From an NFS-mounted Shared Edits Directory to Quorum-based Storage

Converting a HA configuration from using an NFS-mounted shared edits directory to Quorum-based storage involves disabling the current HA configuration then enabling HA using Quorum-based storage.

1. [Disable HA](#).
2. Although the standby NameNode role is removed, its name directories are not deleted. Empty these directories.
3. [Enable HA with Quorum-based storage](#).

Changing a Nameservice Name for Highly Available HDFS Using Cloudera Manager

For background on HDFS high availability, see [Enabling HDFS HA Using Cloudera Manager](#) on page 424.

Before you start, make note of the name of the active NameNode role instance. You can find the list of NameNode instances on the **Instances** tab for the HDFS service in the Cloudera Manager Admin Console.

Complete the following steps to change the NameService name for HDFS with HA:

1. Stop all services except ZooKeeper.
2. On a ZooKeeper server host, run `zookeeper-client`.

   a. Execute the following to remove the configured nameservice. This example assumes the name of the nameservice is **nameservice1**. You can identify the nameservice from the **Federation and High Availability** section on the HDFS **Instances** tab:

```
rmr /hadoop-ha/nameservice1
```

3. In the Cloudera Manager Admin Console, update the NameNode nameservice name.

   a. Go to the HDFS service.
   b. Click the **Configuration** tab.
   c. Type `nameservice` in the Search field.

    d. For the **NameNode Nameservice** property, type the nameservice name in the NameNode (*instance_name*) field. The name must be unique and can contain only alphanumeric characters.

    e. Type `quorum` in the Search field.

    f. For the **Quorum-based Storage Journal name** property, type the nameservice name in the NameNode (*instance_name*) field.

    g. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

4. Click the **Instances** tab.
5. In the Federation and High Availability pane, select **Actions** > **Initialize High Availability State in ZooKeeper**.
6. Go to the Hive service.
7. Select **Actions** > **Update Hive Metastore NameNodes**.
8. Go to the HDFS service.
9. Click the **Instances** tab.
10. Select the checkboxes next to the JournalNode role instances.
11. Select **Actions for Selected** > **Start**.
12. Click on an active **NameNode** role instance.
13. Select **Actions** > **Initialize Shared Edits Directory**.
14. Click the Cloudera Manager logo to return to the **Home** page.
15. Redeploy client configuration files.
16. Start all services except ZooKeeper.

MapReduce (MRv1) and YARN (MRv2) High Availability

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see [Deprecated Items](#).

This section covers:

YARN (MRv2) ResourceManager High Availability

The YARN ResourceManager is responsible for tracking the resources in a cluster and scheduling applications (for example, MapReduce jobs). Before CDH 5, the ResourceManager was a single point of failure in a YARN cluster. The ResourceManager high availability (HA) feature adds redundancy in the form of an active-standby ResourceManager pair to remove this single point of failure. Furthermore, upon failover from the active ResourceManager to the standby, the applications can resume from the last state saved to the state store; for example, map tasks in a MapReduce job are not run again if a failover to a new active ResourceManager occurs after the completion of the map phase. This allows events such the following to be handled without any significant performance effect on running applications:

- Unplanned events such as machine crashes
- Planned maintenance events such as software or hardware upgrades on the machine running the ResourceManager

ResourceManager HA requires ZooKeeper and HDFS services to be running.

Architecture

ResourceManager HA is implemented by means of an active-standby pair of ResourceManagers. On start-up, each ResourceManager is in the standby state; the process is started, but the state is not loaded. When one of the ResourceManagers is transitioning to the active state, the ResourceManager loads the internal state from the designated state store and starts all the internal services. The stimulus to transition to active comes from either the administrator (through the [CLI)](#) or through the integrated failover controller when [automatic failover](#) is enabled. The subsections that follow provide more details about the components of ResourceManager HA.

ResourceManager Restart

Restarting the ResourceManager allows for the recovery of in-flight applications if recovery is enabled. To achieve this, the ResourceManager stores its internal state, primarily application-related data and tokens, to the `RMStateStore`; the cluster resources are re-constructed when the NodeManagers connect. The available alternatives for the state store are `MemoryRMStateStore` (a memory-based implementation) and `ZKRMStateStore` (ZooKeeper-based implementation). Note that `MemoryRMStateStore` will not work for HA.

Fencing

When running two ResourceManagers, a split-brain situation can arise where both ResourceManagers assume they are active. To avoid this, only a single ResourceManager should be able to perform active operations and the other ResourceManager should be "fenced". The ZooKeeper-based state store (`ZKRMStateStore`) allows only a single ResourceManager to make changes to the stored state, implicitly fencing the other ResourceManager. This is accomplished by the ResourceManager claiming exclusive create-delete permissions on the root znode. The ACLs on the root `znode` are automatically created based on the ACLs configured for the store; in case of secure clusters, Cloudera recommends that you set ACLs for the root host such that both ResourceManagers share read-write-admin access, but have exclusive create-delete access. The fencing is implicit and does not require explicit configuration (as fencing in HDFS and MRv1 does). You can plug in a custom "Fencer" if you choose to – for example, to use a different implementation of the state store.

Configuration and FailoverProxy

In an HA setting, you should configure two ResourceManagers to use different ports (for example, ports on different hosts). To facilitate this, YARN uses the notion of an ResourceManager Identifier (`rm-id`). Each ResourceManager has a unique `rm-id`, and all the RPC configurations (`<rpc-address>`; for example `yarn.resourcemanager.address`) for that ResourceManager can be configured via `<rpc-address>.<rm-id>`. Clients, ApplicationMasters, and NodeManagers use these RPC addresses to talk to the active ResourceManager automatically, even after a failover. To achieve this, they cycle through the list of ResourceManagers in the configuration. This is done automatically and does not require any configuration (as it does in HDFS and MapReduce (MRv1)).

Automatic Failover

By default, ResourceManager HA uses ZKFC (ZooKeeper-based failover controller) for automatic failover in case the active ResourceManager is unreachable or goes down. Internally, the ***StandbyElector*** is used to elect the active ResourceManager. The failover controller runs as part of the ResourceManager.

You can plug in a custom failover controller if you prefer.

Manual Transitions and Failover

You can use the [command-line tool](#) `yarn rmadmin` to transition a particular ResourceManager to active or standby state, to fail over from one ResourceManager to the other, to get the HA state of an ResourceManager, and to monitor an ResourceManager's health.

Configuring YARN (MRv2) ResourceManager High Availability Using Cloudera Manager

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

You can use Cloudera Manager to configure CDH for ResourceManager high availability (HA). Cloudera Manager supports automatic failover of the ResourceManager. It does not provide a mechanism to manually force a failover through the Cloudera Manager user interface.

> **Important:** Enabling or disabling HA will cause the previous monitoring history to become unavailable.

Enabling High Availability

1. Go to the YARN service.
2. Select **Actions** > **Enable High Availability**. A screen showing the hosts that are eligible to run a standby ResourceManager displays. The host where the current ResourceManager is running is not available as a choice.

3. Select the host where you want the standby ResourceManager to be installed, and click **Continue**. Cloudera Manager proceeds to run a set of commands that stop the YARN service, add a standby ResourceManager, initialize the ResourceManager high availability state in ZooKeeper, restart YARN, and redeploy the relevant client configurations.

4. Work preserving recovery is enabled for the ResourceManager by default when you enable ResourceManager HA in Cloudera Manager. For more information, including instructions on disabling work preserving recovery, see Work Preserving Recovery for YARN Components on page 433.

> **Note:** ResourceManager HA does not affect the JobHistory Server (JHS). JHS does not maintain any state, so if the host fails you can simply assign it to a new host. You can also enable process auto-restart by doing the following:
>
> 1. Go to the YARN service.
> 2. Click the **Configuration** tab.
> 3. Select **Scope** > **JobHistory Server**.
> 4. Select **Category** > **Advanced**.
> 5. Locate the **Automatically Restart Process** property or search for it by typing its name in the Search box.
> 6. Click **Edit Individual Values**
> 7. Select the JobHistory Server Default Group.
> 8. Restart the JobHistory Server role.

Disabling High Availability

1. Go to the YARN service.
2. Select **Actions** > **Disable High Availability**. A screen showing the hosts running the ResourceManagers displays.
3. Select which ResourceManager (host) you want to remain as the single ResourceManager, and click **Continue**. Cloudera Manager runs a set of commands that stop the YARN service, remove the standby ResourceManager and the Failover Controller, restart the YARN service, and redeploy client configurations.

Using `yarn rmadmin` to Administer ResourceManager HA

You can use `yarn rmadmin` on the command line to manage your ResourceManager HA deployment. `yarn rmadmin` has the following options related to ResourceManager HA:

```
[-transitionToActive serviceId]
[-transitionToStandby serviceId]
[-getServiceState serviceId]
[-checkHealth <serviceId]
[-help <command>]
```

where *serviceId* is the `rm-id`.

> **Note:** Even though `-help` lists the `-failover` option, it is not supported by `yarn rmadmin`.

Work Preserving Recovery for YARN Components

Minimum Required Role: Configurator (also provided by **Cluster Administrator, Full Administrator**)

With work preserving recovery enabled, if a ResourceManager or NodeManager restarts, no in-flight work is lost. You can configure work preserving recovery separately for a ResourceManager or NodeManager. You can enable work preserving recovery whether or not you use ResourceManager High Availability.

> **Note:** YARN does not support high availability for the JobHistory Server (JHS). If the JHS goes down, Cloudera Manager will restart it automatically.

> **Note:**
>
> After moving the JobHistory Server to a new host, the URLs listed for the JobHistory Server on the ResourceManager web UI still point to the old JobHistory Server. This affects existing jobs only. New jobs started after the move are not affected. For any existing jobs that have the incorrect JobHistory Server URL, there is no option other than to allow the jobs to roll off the history over time. For new jobs, make sure that all clients have the updated `mapred-site.xml` that references the correct JobHistory Server.

Configuring Work Preserving Recovery Using Cloudera Manager

Enabling Work Preserving Recovery on ResourceManager with Cloudera Manager

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

If you use Cloudera Manager and you enable [YARN (MRv2) ResourceManager High Availability](#) on page 431, work preserving recovery is enabled by default for the ResourceManager.

Disabling Work Preserving Recovery on ResourceManager Using Cloudera Manager

To disable Work Preserving Recovery for the ResourceManager:

1. Go to the **YARN** service.
2. Click the **Configuration** tab.
3. Search for `Enable ResourceManager Recovery`.
4. In the **Enable ResourceManager Recovery** field, clear the **ResourceManager Default Group** checkbox.
5. Click **Save Changes**.

Enabling Work Preserving Recovery on NodeManager with Cloudera Manager

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

The default value for the recovery directory is `/var/lib/hadoop-yarn/yarn-nm-recovery`.

Work preserving recovery is enabled by default in Cloudera Manager managed clusters.

These are the steps to enable work preserving recovery for a given NodeManager, if needed:

1. Edit the advanced configuration snippet for `yarn-site.xml` on that NodeManager, and set the value of `yarn.nodemanager.recovery.enabled` to *true*.
2. Configure the directory on the local filesystem where state information is stored when work preserving recovery is enabled.

   a. Go to the **YARN** service.
   b. Click the **Configuration** tab.
   c. Search for `NodeManager Recovery Directory`.
   d. Enter the directory path in the **NodeManager Recovery Directory** field (for example, `/var/lib/hadoop-yarn/yarn-nm-recovery`).
   e. Click **Save Changes**.

Example Configuration for Work Preserving Recovery

Minimum Required Role: [Configurator](#) (also provided by **Cluster Administrator, Full Administrator**)

The following example configuration can be used with a Cloudera Manager advanced configuration snippet. Adjust the configuration to suit your environment.

```
<property>
  <name>yarn.resourcemanager.work-preserving-recovery.enabled</name>
  <value>true</value>
  <description>Whether to enable work preserving recovery for the Resource
Manager</description>
</property>
<property>
  <name>yarn.nodemanager.recovery.enabled</name>
  <value>true</value>
  <description>Whether to enable work preserving recovery for the Node
Manager</description>
</property>
<property>
  <name>yarn.nodemanager.recovery.dir</name>
  <value>/home/cloudera/recovery</value>
  <description>The location for stored state on the Node Manager, if work preserving
recovery
    is enabled.</description>
</property>
<property>
  <name>yarn.nodemanager.address</name>
  <value>0.0.0.0:45454</value>
</property>
```

MapReduce (MRv1) JobTracker High Availability

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Follow the instructions in this section to configure high availability (HA) for JobTracker.

Configuring MapReduce (MRv1) JobTracker High Availability Using Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

You can use Cloudera Manager to configure CDH 4.3 or higher for JobTracker high availability (HA). Although it is possible to configure JobTracker HA with CDH 4.2, it is not recommended. Rolling restart, decommissioning of TaskTrackers, and rolling upgrade of MapReduce from CDH 4.2 to CDH 4.3 are not supported when JobTracker HA is enabled.

Cloudera Manager supports automatic failover of the JobTracker. It does not provide a mechanism to manually force a failover through the Cloudera Manager user interface.

> **Important:** Enabling or disabling JobTracker HA will cause the previous monitoring history to become unavailable.

Enabling JobTracker High Availability

The **Enable High Availability** workflow leads you through adding a second (standby) JobTracker:

1. Go to the MapReduce service.
2. Select **Actions** > **Enable High Availability**. A screen showing the hosts that are eligible to run a standby JobTracker displays. The host where the current JobTracker is running is not available as a choice.
3. Select the host where you want the Standby JobTracker to be installed, and click **Continue**.
4. Enter a directory location on the local filesystem for each JobTracker host. These directories will be used to store job configuration data.

- You may enter more than one directory, though it is not required. The paths do not need to be the same on both JobTracker hosts.
- If the directories you specify do not exist, they will be created with the appropriate permissions. If they already exist, they must be empty and have the appropriate permissions.
- If the directories are not empty, Cloudera Manager will not delete the contents.

5. Optionally use the checkbox under Advanced Options to force initialize the ZooKeeper znode for auto-failover.
6. Click **Continue.** Cloudera Manager runs a set of commands that stop the MapReduce service, add a standby JobTracker and Failover controller, initialize the JobTracker high availability state in ZooKeeper, create the job status directory, restart MapReduce, and redeploy the relevant client configurations.

Disabling JobTracker High Availability

1. Go to the MapReduce service.
2. Select **Actions** > **Disable High Availability**. A screen showing the hosts running the JobTrackers displays.
3. Select which JobTracker (host) you want to remain as the single JobTracker, and click **Continue**. Cloudera Manager runs a set of commands that stop the MapReduce service, remove the standby JobTracker and the Failover Controller, restart the MapReduce service, and redeploy client configurations.

Cloudera Navigator Key Trustee Server High Availability

Key Trustee Server high availability applies to read operations only. If either Key Trustee Server fails, the KeyProvider automatically retries fetching keys from the functioning server. New write operations (for example, creating new encryption keys) are not allowed unless both Key Trustee Servers are operational.

If a Key Trustee Server fails, the following operations are impacted:

- **HDFS Encryption**

  – You cannot create new encryption keys for encryption zones.
  – You can write to and read from existing encryption zones, but you cannot create new zones.

- **Cloudera Navigator Encrypt**

  – You cannot register new Cloudera Navigator Encrypt clients.
  – You can continue reading and writing encrypted data, including creating new mount points, using existing clients.

Cloudera recommends monitoring both Key Trustee Servers. If a Key Trustee Server fails catastrophically, restore it from backup to a new host with the same hostname and IP address as the failed host. See Backing Up and Restoring Key Trustee Server and Clients for more information. Cloudera does not support PostgreSQL promotion to convert a passive Key Trustee Server to an active Key Trustee Server.

Depending on your cluster configuration and the security practices in your organization, you might need to restrict the allowed versions of TLS/SSL used by Key Trustee Server. For details, see Specifying TLS/SSL Minimum Allowed Version and Ciphers.

Configuring Key Trustee Server High Availability Using Cloudera Manager

For new installations, use the **Set up HDFS Data At Rest Encryption** wizard and follow the instructions in Enabling HDFS Encryption Using the Wizard. When prompted, make sure that the **Enable High Availability** option is selected.

If you already have a Key Trustee Server service, and want to enable high availability, use the Add Role Instances wizard for the Key Trustee Server service instead to add the Passive Key Trustee Server and Passive Database roles.

> **Important:** You *must* assign the Key Trustee Server and Database roles to the same host. Assign the Active Key Trustee Server and Active Database roles to one host, and the Passive Key Trustee Server and Passive Database roles to a separate host.

After completing the **Add Role Instances** wizard, the Passive Key Trustee Server and Passive Database roles fail to start. Complete the following manual actions to start these roles:

1. Stop the Key Trustee Server service (**Key Trustee Server service** > **Actions** > **Stop**).
2. Run the **Set Up Key Trustee Server Database** command (**Key Trustee Server service** > **Actions** > **Set Up Key Trustee Server Database**).
3. Run the following command on the Active Key Trustee Server:

```
$ sudo rsync -zcav --exclude .ssl /var/lib/keytrustee/.keytrustee
root@keytrustee02.example.com:/var/lib/keytrustee/.
```

   Replace *keytrustee02.example.com* with the hostname of the Passive Key Trustee Server.

4. Run the following command on the Passive Key Trustee Server:

```
$ sudo ktadmin init
```

5. Start the Key Trustee Server service (**Key Trustee Server service** > **Actions** > **Start**).

   > ❗ **Important:** Starting or restarting the Key Trustee Server service attempts to start the Active Database and Passive Database roles. If the Active Database is not running when the Passive Database attempts to start, the Passive Database fails to start. If this occurs, manually restart the Passive Database role after confirming that the Active Database role is running.

6. Enable synchronous replication (**Key Trustee Server service** > **Actions** > **Setup Enable Synchronous Replication in HA mode**).
7. Restart the Key Trustee Server service (**Key Trustee Server service** > **Actions** > **Restart**).

For parcel-based Key Trustee Server releases 5.8 and higher, Cloudera Manager automatically backs up Key Trustee Server (using the `ktbackup.sh` script) after adding the Key Trustee Server service. It also schedules automatic backups using `cron`. For package-based installations, you must manually back up Key Trustee Server and configure a `cron` job.

Cloudera Manager configures `cron` to run the backup script hourly. The latest 10 backups are retained in `/var/lib/keytrustee` in cleartext. For information about using the backup script and configuring the `cron` job (including how to encrypt backups), see Backing Up Key Trustee Server and Key Trustee KMS Using the ktbackup.sh Script.

Recovering a Key Trustee Server

If a Key Trustee Server fails, restore it from backup as soon as possible. If the Key Trustee Server hosts fails completely, make sure that you restore the Key Trustee Server to a new host with the same hostname and IP address as the failed host.

For more information, see Backing Up and Restoring Key Trustee Server and Clients.

Enabling Key Trustee KMS High Availability

CDH 6 supports Key Trustee KMS high availability. For new installations, you can use the Set up HDFS Data At Rest Encryption wizard to install and configure Key Trustee KMS high availability. If you have an existing standalone Key Trustee KMS service, use the following procedure to enable Key Trustee KMS high availability:

1. Back up the Key Trustee KMS private key and configuration directory. See Backing Up and Restoring Key Trustee Server and Clients for more information.
2. If you do not have a ZooKeeper service in your cluster, add one using the instructions in Adding a Service on page 193.
3. Run the Add Role Instances wizard for the Key Trustee KMS service (**Key Trustee KMS service** > **Actions** > **Add Role Instances**).

4. Click **Select hosts** and check the box for the host where you want to add the additional Key Management Server Proxy role. See [Resource Planning for Data at Rest Encryption](#) for considerations when selecting a host. Click **OK** and then **Continue**.

5. On the **Review Changes** page of the wizard, confirm the authorization code, organization name, and settings, and then click **Finish**.

> **!** **Important:** The initial startup of the KMS instance may fail with the following error message:
>
> ```
> java.io.IOException: Unable to verify private key match between KMS
>  hosts. If the system has been recently upgraded, DO NOT TAKE FURTHER
>  ACTION and contact your support representative as soon as possible.
>  If this is a new installation, verify private key files have been
> synced between all KMS hosts. Aborting to prevent data inconsistency.
> ```
>
> If this occurs, it indicates that the KMS attempted to verify the Key Trustee private key has been synchronized with the new instance, but was unable to because that synchronization has not yet taken place. This is expected behavior at this point in the process. Proceed to the next step, and the new KMS instance will come up when the KMS service is restarted after the synchronization.

6. If it is not already running, start the new KMS instance. Select the new instance and go to **Actions for Selected > Start**.

7. Go to **Key Trustee KMS service** > **Configuration** and make sure that the **ZooKeeper Service** dependency is set to the ZooKeeper service for your cluster.

8. Synchronize the Key Trustee KMS private key.

> **⚠** **Warning:** It is *very important* that you perform this step. Failure to do so leaves Key Trustee KMS in a state where keys are intermittently inaccessible, depending on which Key Trustee KMS host a client interacts with, because cryptographic key material encrypted by one Key Trustee KMS host cannot be decrypted by another. If you are already running multiple Key Trustee KMS hosts with different private keys, immediately [back up](#) all Key Trustee KMS hosts, and contact Cloudera Support for assistance correcting the issue.
>
> If you fail to maintain proper synchronization of private keys between Key Trustee KMS hosts, then the GPG validation check that runs automatically when the Key Trustee KMS is restarted will return the following error and abort the restart operation, forcing you to synchronize private keys before a restart can occur:
>
> ```
> java.io.IOException: Unable to verify private key match between KMS
>  hosts. Verify private key files have been synced
> between all KMS hosts. Aborting to prevent data inconsistency.
> ```

To determine whether the Key Trustee KMS private keys are different, compare the MD5 hash of the private keys. On each Key Trustee KMS host, run the following command:

```
$ md5sum /var/lib/kms-keytrustee/keytrustee/.keytrustee/secring.gpg
```

If the outputs are different, contact Cloudera Support for assistance. Do not attempt to synchronize existing keys. If you overwrite the private key and do not have a backup, any keys encrypted by that private key are permanently inaccessible, and any data encrypted by those keys is permanently irretrievable. If you are configuring Key Trustee KMS high availability for the first time, continue synchronizing the private keys.

Cloudera recommends following security best practices and transferring the private key using offline media, such as a removable USB drive. For convenience (for example, in a development or testing environment where maximum

security is not required), you can copy the private key over the network by running the following `rsync` command on the original Key Trustee KMS host:

```
rsync -zav /var/lib/kms-keytrustee/keytrustee/.keytrustee
root@ktkms02.example.com:/var/lib/kms-keytrustee/keytrustee/.
```

Replace *ktkms02.example.com* with the hostname of the Key Trustee KMS host that you are adding.

9. Restart the Key Trustee KMS service (**Key Trustee KMS service** > **Actions** > **Restart**).
10. Restart the cluster.
11. Redeploy the client configuration (**Home** > **Cluster-wide** ▾ > **Deploy Client Configuration**).
12. Re-run the steps in Validating Hadoop Key Operations.

Enabling Navigator HSM KMS High Availability

CDH 6 supports HSM KMS high availability. For new installations, you can use the Set up HDFS Data At Rest Encryption wizard to install and configure HSM KMS high availability. If you have an existing standalone HSM KMS service, use the following procedure to enable HSM KMS high availability:

1. If you do not have a ZooKeeper service in your cluster, add one using the instructions in Adding a Service on page 193.
2. Run the Add Role Instances wizard for the HSM KMS service (**HSM KMS service** > **Actions** > **Add Role Instances**).
3. Click **Select hosts** and check the box for the host where you want to add the additional Key Management Server Proxy role. See Resource Planning for Data at Rest Encryption for considerations when selecting a host. Click **OK** and then **Continue**.

> ⚠ **Warning:** The same host must be specified for the Navigator HSM KMS metastore and Navigator HSM KMS proxy.

4. On the **Review Changes** page of the wizard, confirm the HSM KMS settings, and then click **Finish**.
5. Go to **HSM KMS service** > **Configuration** and make sure that the **ZooKeeper Service** dependency is set to the ZooKeeper service for your cluster.
6. In the **Add Role Instance** path, the initialize metastore action does not run automatically (as it does for the **Add Service** wizard). When a new metastore instance is added, the initialize metastore action must be run manually before starting the metastore). So, stop both role instances (metastore and proxy) and then run the initialize metastore action.
7. Restart the HSM KMS service (**HSM KMS service** > **Actions** > **Restart**).
8. Restart the cluster.
9. Redeploy the client configuration (**Home** > **Cluster-wide** > **Select from Cluster drop-down menu (arrow icon)** > **Deploy Client Configuration**).
10. Re-run the steps in Validating Hadoop Key Operations.

HSM KMS High Availability Backup and Recovery

When running the HSM KMS in high availability mode, if either of the two nodes fails, a role instance can be assigned to another node and federated into the service by the single remaining active node. In other words, you can bring a node that is part of the cluster, but that is not running HSM KMS role instances, into the service by making it an HSM KMS role instance–more specifically, an HSM KMS proxy role instance and an HSM KMS metastore role instance. So each node acts as an online ("hot" backup) backup of the other. In many cases, this will be sufficient. However, if a manual ("cold" backup) backup of the files necessary to restore the service from scratch is desirable, you can create that as well.

To create a backup, copy the `/var/lib/hsmkp` and `/var/lib/hsmkp-meta` directories on one or more of the nodes running HSM KMS role instances.

To restore from a backup: bring up a completely new instance of the HSM KMS service, and copy the `/var/lib/hsmkp` and `/var/lib/hsmkp-meta` directories from the backup onto the file system of the restored nodes before starting HSM KMS for the first time.

High Availability for Other CDH Components

This section provides information on high availability for CDH components independently of HDFS. See also Configuring Other CDH Components to Use HDFS HA on page 428.

For details about HA for Impala, see Using Impala through a Proxy for High Availability.

For details about HA for Cloudera Search, see Using Search through a Proxy for High Availability.

HBase High Availability

Most aspects of HBase are highly available in a standard configuration. A cluster typically consists of one Master and three or more RegionServers, with data stored in HDFS. To ensure that every component is highly available, configure one or more backup Masters. The backup Masters run on other hosts than the active Master.

Enabling HBase High Availability Using Cloudera Manager

1. Go to the HBase service.
2. Follow the process for adding a role instance and add a backup Master to a different host than the one on which the active Master is running.

HBase Read Replicas

Without read replicas, only one RegionServer services a read request from a client, regardless of whether RegionServers are colocated with other DataNodes that have local access to the same block. This ensures consistency of the data being read. However, a RegionServer can become a bottleneck due to an underperforming RegionServer, network problems, or other reasons that could cause slow reads.

With read replicas enabled, the HMaster distributes read-only copies of regions (*replicas*) to different RegionServers in the cluster. One RegionServer services the default or *primary* replica, which is the only replica which can service write requests. If the RegionServer servicing the primary replica is down, writes will fail.

Other RegionServers serve the *secondary* replicas, follow the primary RegionServer and only see committed updates. The secondary replicas are read-only, and are unable to service write requests. The secondary replicas can be kept up to date by reading the primary replica's HFiles at a set interval or by replication. If they use the first approach, the secondary replicas may not reflect the most recent updates to the data when updates are made and the RegionServer has not yet flushed the memstore to HDFS. If the client receives the read response from a secondary replica, this is indicated by marking the read as "stale". Clients can detect whether or not the read result is stale and react accordingly.

Replicas are placed on different RegionServers, and on different racks when possible. This provides a measure of high availability (HA), as far as reads are concerned. If a RegionServer becomes unavailable, the regions it was serving can still be accessed by clients even before the region is taken over by a different RegionServer, using one of the secondary replicas. The reads may be stale until the entire WAL is processed by the new RegionServer for a given region.

For any given read request, a client can request a faster result even if it comes from a secondary replica, or if consistency is more important than speed, it can ensure that its request is serviced by the primary RegionServer. This allows you to decide the relative importance of consistency and availability, in terms of the CAP Theorem, in the context of your application, or individual aspects of your application, using Timeline Consistency semantics.

Timeline Consistency

Timeline Consistency is a consistency model which allows for a more flexible standard of consistency than the default HBase model of *strong consistency*. A client can indicate the level of consistency it requires for a given read (Get or Scan) operation. The default consistency level is `STRONG`, meaning that the read request is only sent to the RegionServer servicing the region. This is the same behavior as when read replicas are not used. The other possibility, `TIMELINE`, sends the request to all RegionServers with replicas, including the primary. The client accepts the first response, which

includes whether it came from the primary or a secondary RegionServer. If it came from a secondary, the client can choose to verify the read later or not to treat it as definitive.

Keeping Replicas Current

The read replica feature includes two different mechanisms for keeping replicas up to date:

Using a Timer

In this mode, replicas are refreshed at a time interval controlled by the configuration option `hbase.regionserver.storefile.refresh.period`.

Using Replication

In this mode, replicas are kept current between a source and sink cluster using HBase replication. This can potentially allow for faster synchronization than using a timer. Each time a flush occurs on the source cluster, a notification is pushed to the sink clusters for the table. To use replication to keep replicas current, you must first set the column family attribute REGION_MEMSTORE_REPLICATION to `false`, then set the HBase configuration property `hbase.region.replica.replication.enabled` to `true`.

> **Important:** Read-replica updates using replication are not supported for the `hbase:meta` table. Columns of hbase:meta must always have their REGION_MEMSTORE_REPLICATION attribute set to `false`.

Enabling Read Replica Support

> **Important:**
>
> Before you enable read-replica support, make sure to account for their increased heap memory requirements. Although no additional copies of HFile data are created, read-only replicas regions have the same memory footprint as normal regions and need to be considered when calculating the amount of increased heap memory required. For example, if your table requires 8 GB of heap memory, when you enable three replicas, you need about 24 GB of heap memory.

To enable support for read replicas in HBase, you must set several properties.

Table 23: HBase Read Replica Properties

Property Name	Default Value	Description
`hbase.region.replica.replication.enabled`	`false`	The mechanism for refreshing the secondary replicas. If set to `false`, secondary replicas are not guaranteed to be consistent at the row level. Secondary replicas are refreshed at intervals controlled by a timer (`hbase.regionserver.storefile.refresh.period`), and so are guaranteed to be at most that interval of milliseconds behind the primary RegionServer. Secondary replicas read from the HFile in HDFS, and have no access to writes that have not been flushed to the HFile by the primary RegionServer. If `true`, replicas are kept up to date using replication. and the column family has the attribute `REGION_MEMSTORE_REPLICATION` set to `false`, Using replication for read replication of `hbase:meta` is not

Property Name	Default Value	Description
		supported, and `REGION_MEMSTORE_REPLICATION` must always be set to `false` on the column family.
`hbase.regionserver.storefile.refresh.period`	0 (disabled)	The period, in milliseconds, for refreshing the store files for the secondary replicas. The default value of `0` indicates that the feature is disabled. Secondary replicas update their store files from the primary RegionServer at this interval. If refreshes occur too often, this can create a burden for the NameNode. If refreshes occur too infrequently, secondary replicas will be less consistent with the primary RegionServer.
`hbase.master.loadbalancer.class`	`org.apache.hadoop.hbase.master.` `balancer.StochasticLoadBalancer` (the class name is split for formatting purposes)	The Java class used for balancing the load of all HBase clients. The default implementation is the `StochasticLoadBalancer`, which is the only load balancer that supports reading data from secondary RegionServers.
`hbase.ipc.client.allowsInterrupt`	true	Whether or not to enable interruption of RPC threads at the client. The default value of `true` enables primary RegionServers to access data from other regions' secondary replicas.
`hbase.client.primaryCallTimeout.get`	10 ms	The timeout period, in milliseconds, an HBase client's will wait for a response before the read is submitted to a secondary replica if the read request allows timeline consistency. The default value is 10. Lower values increase the number of remote procedure calls while lowering latency.
`hbase.client.primaryCallTimeout.multiget`	10 ms	The timeout period, in milliseconds, before an HBase client's multi-get request, such as `HTable.get(List<GET>))`, is submitted to a secondary replica if the multi-get request allows timeline consistency. Lower values increase the number of remote procedure calls while lowering latency.

Configure Read Replicas Using Cloudera Manager

1. Before you can use replication to keep replicas current, you must set the column attribute `REGION_MEMSTORE_REPLICATION` to `false` for the HBase table, using HBase Shell or the client API. See Activating Read Replicas On a Table on page 444.
2. Select **Clusters** > *HBase*.
3. Click the **Configuration** tab.
4. Select **Scope** > **HBase or HBase Service-Wide**.
5. Select **Category** > **Advanced**.

6. Locate the **HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml** property or search for it by typing its name in the Search box.

7. Using the chart above, create a configuration and paste it into the text field. The following example configuration demonstrates the syntax:

```xml
<property>
  <name>hbase.regionserver.storefile.refresh.period</name>
  <value>0</value>
</property>
<property>
  <name>hbase.ipc.client.allowsInterrupt</name>
  <value>true</value>
  <description>Whether to enable interruption of RPC threads at the client. The default
 value of true is
    required to enable Primary RegionServers to access other RegionServers in secondary
 mode. </description>
</property>
<property>
  <name>hbase.client.primaryCallTimeout.get</name>
  <value>10</value>
</property>
<property>
  <name>hbase.client.primaryCallTimeout.multiget</name>
  <value>10</value>
</property>
```

8. Click **Save Changes** to commit the changes.

9. Restart the HBase service.

Configuring Rack Awareness for Read Replicas

Rack awareness for read replicas is modeled after the mechanism used for rack awareness in Hadoop. Its purpose is to ensure that some replicas are on a different rack than the RegionServer servicing the table. The default implementation, which you can override by setting `hbase.util.ip.to.rack.determiner`, to custom implementation, is `ScriptBasedMapping`, which uses a *topology map* and a *topology script* to enforce distribution of the replicas across racks.

Creating a Topology Map

The topology map assigns hosts to racks. It is read by the topology script. A rack is a logical grouping, and does not necessarily correspond to physical hardware or location. Racks can be nested. If a host is not in the topology map, it is assumed to be a member of the default rack. The following map uses a nested structure, with two data centers which each have two racks. All services on a host that are rack-aware will be affected by the rack settings for the host.

If you use Cloudera Manager, do not create the map manually. Instead, go to **Hosts**, select the hosts to assign to a rack, and select **Actions for Selected** > **Assign Rack**.

```xml
<topology>
  <node name="host1.example.com" rack="/dc1/r1"/>
  <node name="host2.example.com" rack="/dc1/r1"/>
  <node name="host3.example.com" rack="/dc1/r2"/>
  <node name="host4.example.com" rack="/dc1/r2"/>
  <node name="host5.example.com" rack="/dc2/r1"/>
  <node name="host6.example.com" rack="/dc2/r1"/>
  <node name="host7.example.com" rack="/dc2/r2"/>
  <node name="host8.example.com" rack="/dc2/r2"/>
</topology>
```

Creating a Topology Script

The topology script determines rack topology using the topology map. By default, CDH uses `/etc/hadoop/conf.cloudera.YARN-1/topology.py` To use a different script, set `net.topology.script.file.name` to the absolute path of the topology script.

Activating Read Replicas On a Table

After enabling read replica support on your RegionServers, configure the tables for which you want read replicas to be created. Keep in mind that each replica increases the amount of storage used by HBase in HDFS.

At Table Creation

To create a new table with read replication capabilities enabled, set the REGION_REPLICATION property on the table. Use a command like the following, in HBase Shell:

```
hbase> create 'myTable', 'myCF', {REGION_REPLICATION => '3'}
```

By Altering an Existing Table

You can also alter an existing column family to enable or change the number of read replicas it propagates, using a command similar to the following. The change will take effect at the next major compaction.

```
hbase> disable 'myTable'
hbase> alter 'myTable', 'myCF', {REGION_REPLICATION => '3'}
hbase> enable 'myTable'
```

Requesting a Timeline-Consistent Read

To request a timeline-consistent read in your application, use the `get.setConsistency(Consistency.TIMELINE)` method before performing the `Get` or `Scan` operation.

To check whether the result is stale (comes from a secondary replica), use the `isStale()` method of the result object. Use the following examples for reference.

Get Request

```
Get get = new Get(key);
get.setConsistency(Consistency.TIMELINE);
Result result = table.get(get);
```

Scan Request

```
Scan scan = new Scan();
scan.setConsistency(CONSISTENCY.TIMELINE);
ResultScanner scanner = table.getScanner(scan);
Result result = scanner.next();
```

Scan Request to a Specific Replica

This example overrides the normal behavior of sending the read request to all known replicas, and only sends it to the replica specified by ID.

```
Scan scan = new Scan();
scan.setConsistency(CONSISTENCY.TIMELINE);
scan.setReplicaId(2);
ResultScanner scanner = table.getScanner(scan);
Result result = scanner.next();
```

Detecting a Stale Result

```
Result result = table.get(get);
if (result.isStale()) {
    ...
}
```

Getting and Scanning Using HBase Shell

You can also request timeline consistency using HBase Shell, allowing the result to come from a secondary replica.

```
hbase> get 'myTable', 'myRow', {CONSISTENCY => "TIMELINE"}
hbase> scan 'myTable', {CONSISTENCY => 'TIMELINE'}
```

Oozie High Availability

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

In CDH 5, you can configure multiple active Oozie servers against the same database. Oozie high availability is "active-active" or "hot-hot" so that both Oozie servers are active at the same time, with no failover. High availability for Oozie is supported in both MRv1 and MRv2 (YARN).

Requirements for Oozie High Availability

- Multiple active Oozie servers, preferably identically configured.
- JDBC JAR in the same location across all Oozie hosts (for example, /var/lib/oozie/).
- External database that supports multiple concurrent connections, preferably with HA support. The default Derby database does not support multiple concurrent connections.
- ZooKeeper ensemble with distributed locks to control database access, and service discovery for log aggregation.
- Load balancer (preferably with HA support, for example HAProxy), virtual IP, or round-robin DNS to provide a single entry point (of the multiple active servers), and for callbacks from the Application Master or JobTracker.

To enable Kerberos authentication, see Enabling Kerberos Authentication for CDH.

For information on setting up TLS/SSL communication with Oozie HA enabled, see Additional Considerations when Configuring TLS/SSL for Oozie HA.

Configuring Oozie High Availability Using Cloudera Manager

Minimum Required Role: Full Administrator

> **Important:** Enabling or disabling high availability makes the previous monitoring history unavailable.

Enabling Oozie High Availability

1. Ensure that the requirements are satisfied.
2. In the Cloudera Manager Admin Console, go to the Oozie service.
3. Select **Actions** > **Enable High Availability** to see eligible Oozie server hosts. The host running the current Oozie server is not eligible.
4. Select the host on which to install an additional Oozie server and click **Continue**.
5. Update the following fields for the Oozie load balancer:

   - Hostname

     For example:

```
nightly6x-1.vpc.cloudera.com
```

   - HTTP Port

For example:

```
5002
```

- HTTPS Port

For example:

```
5000
```

6. Click **Continue**.

Cloudera Manager stops the Oozie servers, adds another Oozie server, initializes the Oozie server High Availability state in ZooKeeper, configures Hue to reference the Oozie load balancer, and restarts the Oozie servers and dependent services. In addition, Cloudera Manager generates Kerberos credentials for the new Oozie server and regenerates credentials for existing servers.

Disabling Oozie High Availability

1. In the Cloudera Manager Admin Console, go to the Oozie service.
2. Select **Actions** > **Disable High Availability** to see all hosts currently running Oozie servers.
3. Select the one host to run the Oozie server and click **Continue**. Cloudera Manager stops the Oozie service, removes the additional Oozie servers, configures Hue to reference the Oozie service, and restarts the Oozie service and dependent services.

Search High Availability

Load Balancing

Using a proxy server to relay requests to and from the Apache Solr service can help meet availability requirements in production clusters serving many users.

For information on configuring a load balancer for the Solr service, see <u>Using a Load Balancer with Solr</u> on page 140.

Data Ingestion

Mission critical, large-scale online production systems need to make progress without downtime despite some issues. Cloudera Search provides two routes to configurable, highly available, and fault-tolerant data ingestion:

- Near Real Time (NRT) ingestion using the Flume Solr Sink
- MapReduce based batch ingestion using the MapReduceIndexerTool

Production versus Test Mode

Some exceptions are generally transient, in which case the corresponding task can simply be retried. For example, network connection errors or timeouts are recoverable exceptions. Conversely, tasks associated with an unrecoverable exception cannot simply be retried. Corrupt or malformed parser input data, parser bugs, and errors related to unknown Solr schema fields produce unrecoverable exceptions.

Different modes determine how Cloudera Search responds to different types of exceptions.

- **Configuration parameter isProductionMode=false** (Non-production mode or test mode): Default configuration. Cloudera Search throws exceptions to quickly reveal failures, providing better debugging diagnostics to the user.
- **Configuration parameter isProductionMode=true** (Production mode): Cloudera Search logs and ignores unrecoverable exceptions, enabling mission-critical large-scale online production systems to make progress without downtime, despite some issues.

> **Note:** Categorizing exceptions as recoverable or unrecoverable addresses most cases, though it is possible that an unrecoverable exception could be accidentally misclassified as recoverable. Cloudera provides the `isIgnoringRecoverableExceptions` configuration parameter to address such a case. In a production environment, if an unrecoverable exception is discovered that is classified as recoverable, change `isIgnoringRecoverableExceptions` to `true`. Doing so allows systems to make progress and avoid retrying an event forever. This configuration flag should only be enabled if a misclassification bug has been identified. Please report such bugs to Cloudera.

If Cloudera Search throws an exception according the rules described above, the caller, meaning Flume Solr Sink and MapReduceIndexerTool, can catch the exception and retry the task if it meets the criteria for such retries.

Near Real Time Indexing with the Flume Solr Sink

The Flume Solr Sink uses the settings established by the `isProductionMode` and `isIgnoringRecoverableExceptions` parameters. If a SolrSink does nonetheless receive an exception, the SolrSink rolls the transaction back and pauses. This causes the Flume channel, which is essentially a queue, to redeliver the transaction's events to the SolrSink approximately five seconds later. This redelivering of the transaction event retries the ingest to Solr. This process of rolling back, backing off, and retrying continues until ingestion eventually succeeds.

Here is a corresponding example Flume configuration file `flume.conf`:

```
agent.sinks.solrSink.isProductionMode = true
agent.sinks.solrSink.isIgnoringRecoverableExceptions = true
```

In addition, Flume SolrSink automatically attempts to load balance and failover among the hosts of a SolrCloud before it considers the transaction rollback and retry. Load balancing and failover is done with the help of ZooKeeper, which itself can be configured to be highly available.

Further, Cloudera Manager can configure Flume so it automatically restarts if its process crashes.

To tolerate extended periods of Solr downtime, you can configure Flume to use a high-performance transactional persistent queue in the form of a [FileChannel](#). A FileChannel can use any number of local disk drives to buffer significant amounts of data. For example, you might buffer many terabytes of events corresponding to a week of data. Further, using the [Replicating Channel Selector](#) Flume feature, you can configure Flume to replicate the same data both into HDFS as well as into Solr. Doing so ensures that if the Flume SolrSink channel runs out of disk space, data delivery is still delivered to HDFS, and this data can later be ingested from HDFS into Solr using MapReduce.

Many machines with many Flume Solr Sinks and FileChannels can be used in a failover and load balancing configuration to improve high availability and scalability. Flume SolrSink servers can be either co-located with live Solr servers serving end user queries, or Flume SolrSink servers can be deployed on separate industry standard hardware for improved scalability and reliability. By spreading indexing load across a large number of Flume SolrSink servers you can improve scalability. Indexing load can be replicated across multiple Flume SolrSink servers for high availability, for example using Flume features such as [Load balancing Sink Processor](#).

Batch Indexing with MapReduceIndexerTool

The Mappers and Reducers of the MapReduceIndexerTool follow the settings established by the `isProductionMode` and `isIgnoringRecoverableExceptions` parameters. However, if a Mapper or Reducer of the `MapReduceIndexerTool` does receive an exception, it does not retry at all. Instead it lets the MapReduce task fail and relies on the Hadoop Job Tracker to retry failed MapReduce task attempts several times according to standard Hadoop semantics. Cloudera Manager can configure the Hadoop Job Tracker to be highly available. On `MapReduceIndexerTool` startup, all data in the output directory is deleted if that output directory already exists. To retry an entire job that has failed, rerun the program using the same arguments.

For example:

```
hadoop ... MapReduceIndexerTool ... -D isProductionMode=true -D
isIgnoringRecoverableExceptions=true ...
```

Navigator Data Management in a High Availability Environment

Navigator Metadata Server and Navigator Audit Server do not currently support highly availability configurations.

When Cloudera Manager is configured for high availability using a load balancer with an active and passive Cloudera Management Service, you would configure single instances of Navigator Metadata Server and Navigator Audit Server. Therefore, if a failover is triggered, Cloudera Manager may fail over to a host where Cloudera Navigator is not available.

Navigator Configuration in a Highly Available Environment

When Cloudera Manager is configured for high availability, set up Navigator as follows:

Navigator Audit Server

- One instance of Navigator Audit Server.

  Navigator Audit Server does not provide a mechanism for reconciling or synchronizing two independent audit databases.

- RDBMS configured for high availability.

  Use [database-specific mechanisms](#) to ensure high availability.

Navigator Metadata Server

- One instance of Navigator Metadata Server.

  Navigator Metadata Server does not provide a mechanism for reconciling or synchronizing two independent storage directories. Typically in a production environment, the Navigator Metadata Server is installed on a host other than where Cloudera Manager is running to ensure there are enough compute resources and storage.

- RDBMS configured for high availability.

  Use [database-specific mechanisms](#) to ensure high availability.

- Solr storage directory.

  No high availability configuration is supported.

Navigator Behavior in a High Availability Environment

If a failover is triggered, Cloudera Manager may fail over to a host where Cloudera Navigator is not available. Here's the behavior you can expect if some component (Navigator or otherwise) goes down:

Navigator Audit Server

If Navigator Audit Server stops running, audit services will continue to queue audit events. When Navigator Audit Server is available again, Cloudera Manager agents collect the events and pass them to Navigator Audit Server. The same behavior occurs if Navigator Audit Server is running but it can't access the audit database: Navigator Audit Server stops accepting events from the cm_agent(s) and the events are held on the host where the service is running.

There are two potential problems that can occur if Navigator Audit Server (or its underlying RDBMS instance) stays offline:

- Audit events can fill up the local file system on the host where services are running. If Navigator Audit Server is down for a prolonged interval and space becomes an issue, consider archiving audit files elsewhere and replacing them after the audit server processes some of the events.
- After 24 or more hours, Navigator Audit Server runs out of pre-created audit tables in its database. If Navigator Audit Server is offline for more than a day, you may need to manually create the missing audit tables before it can resume processing. See [Processing a backlog of audit logs](#).

Navigator Metadata Server

When Navigator Metadata Server stops running, the embedded Solr instance stops running, or Navigator Metadata Server can't access its database:

- Navigator console is not available.
- Metadata extractors stop collecting metadata from supported services.
- Lineage relations are not calculated.
- Scheduled metadata purge jobs don't run.
- Periodic policy jobs don't run.

While metadata for existing data assets—files, tables, partitions, and so on—will be collected when Navigator Metadata Server restarts, there are some circumstances where it is possible to lose metadata:

- **Operations and operation execution entities** from services that use pull extractors (Yarn, MapReduce, Sqoop, and Oozie) are extracted from the JobHistory server. If Navigator Metadata Server is not running, the JobHistory server persists the information, and it is collected when Navigator Metadata Server starts again. However, it is possible to lose metadata for operation executions if Navigator Metadata Server is stopped long enough that the JobHistory server logs are recycled. Operation executions roll up into operation entities and operations are used to generate lineage: If the missing operation executions correspond to existing operations, there is no impact on lineage relations. If these operation executions are on-going, even these lineage relations are eventually created after Navigator is restored. If the missing operation executions are unique in that they do not correspond to existing operations and later operation executions don't produce the same operation, it is possible to be missing lineage for entities referenced by the missing operation executions.

  Operations and operation executions from push extractors such as for HiveServer2, Impala, and Spark are held at the service until Navigator Metadata Server is restarted and are not at risk for loss during a Navigator Metadata Server outage.

- **Data asset entities** for data assets that are created and destroyed during the outage will not appear in Navigator.

If Navigator Metadata Server (or its underlying RDBMS instance or Solr) stays offline, metadata can fill up the local file system on the host where services are running. If Navigator Metadata Server is down for a prolonged interval and space becomes an issue, consider archiving JobHistory files and service log files elsewhere and replacing them after the metadata server processes some of the metadata.

Configuring Cloudera Manager for High Availability With a Load Balancer

This section provides an example of configuring Cloudera Manager for high availability using a TCP load balancer. The procedures describe how to configure high availability using a specific, open-source load balancer. Depending on the operational requirements of your CDH deployment, you can select a different load balancer. You can use either a hardware or software load balancer, but must be capable of forwarding all Cloudera Manager ports to backing server instances. (See Ports Used by Cloudera Manager and Cloudera Navigator for more information about the ports used by Cloudera Manager.)

This topic discusses Cloudera Manager high availability in the context of *active-passive* configurations only; *active-active* configurations are currently unsupported. For more information about the differences between *active-passive* and *active-active* High Availability, see http://en.wikipedia.org/wiki/High-availability_cluster.

> ⊘ **Important:** Cloudera Support supports all of the configuration and modification to Cloudera software detailed in this document. However, Cloudera Support is unable to assist with issues or failures with the third-party software that is used. Use of any third-party software, or software not directly covered by Cloudera Support, is at the risk of the end user.

Introduction to Cloudera Manager Deployment Architecture

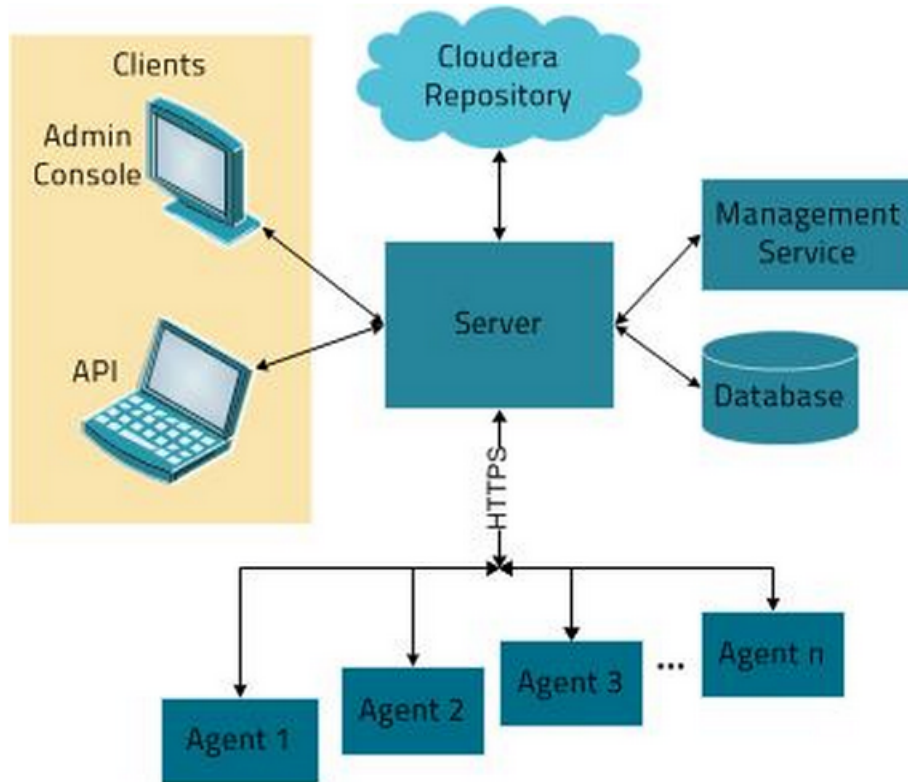Cloudera Manager consists of the following software components:

Figure 8: Cloudera Manager Architecture

- Cloudera Manager Server
- Cloudera Management Service
- Relational databases (several)
- Filesystem-based runtime state storage (used by some services that are part of Cloudera Management Service)
- Cloudera Manager Agent (one instance per each managed host)

You can locate the Cloudera Manager Server and Cloudera Management Service on different hosts (with each role of the Cloudera Management Service, such as the Event Server or the Alert Server and so on, possibly located on different hosts).

Cloudera Manager Server and some of the Cloudera Management Service roles (such as Cloudera Navigator) use a relational database to store their operational data. Some other services (such as the Host Monitor and the Service Monitor) use the filesystem (through LevelDB) to store their data.

High availability in the context of Cloudera Manager involves configuring secondary failover instances for each of these services and also for the persistence components (the relational database and the file system) that support these services. For simplicity, this document assumes that all of the Cloudera Management Service roles are located on a single machine.

The Cloudera Manager Agent software includes an *agent* and a *supervisor* process. The agent process handles RPC communication with Cloudera Manager and with the roles of the Cloudera Management Service, and primarily handles configuration changes to your roles. The supervisor process handles the local Cloudera-deployed process lifecycle and handles auto-restarts (if configured) of failed processes.

Prerequisites for Setting up Cloudera Manager High Availability

Note: MySQL GTID-based replication is not supported.

- A multi-homed TCP load balancer, or two TCP load balancers, capable of proxying requests on specific ports to one server from a set of backing servers.

  - The load balancer does not need to support termination of TLS/SSL connections.
  - This load balancer can be hardware or software based, but should be capable of proxying multiple ports. HTTP/HTTPS-based load balancers are insufficient because Cloudera Manager uses several non-HTTP-based protocols internally.
  - This document uses **HAProxy**, a small, open-source, TCP-capable load balancer, to demonstrate a workable configuration.

- A networked storage device that you can configure to be highly available. Typically this is an NFS store, a SAN device, or a storage array that satisfies the read/write throughput requirements of the Cloudera Management Service. This document assumes the use of NFS due to the simplicity of its configuration and because it is an easy, vendor-neutral illustration.
- The procedures in this document require `ssh` access to all the hosts in the cluster where you are enabling high availability for Cloudera Manager.

The Heartbeat Daemon and Virtual IP Addresses

You may have configured Cloudera Manager high availability by configuring virtual IP addresses and using the Heartbeat daemon (http://www.linux-ha.org/wiki/Heartbeat). The original Heartbeat package is deprecated; however, support and maintenance releases are still available through LinBit (https://www.linbit.com/en/linbit-takes-over-heartbeat-maintenance/).

Cloudera recommends using Corosync and Pacemaker (both currently maintained through ClusterLabs). Corosync is an open-source high-availability tool commonly used in the open-source community.

Editions of this document released for Cloudera Manager4 and CDH 4 also used virtual IP addresses that move as a resource from one host to another on failure. Using virtual IP addresses has several drawbacks:

- Questionable reliance on outdated Address Resolution Protocol (ARP) behavior to ensure that the IP-to-MAC translation works correctly to resolve to the new MAC address on failure.
- Split-brain scenarios that lead to problems with routing.
- A requirement that the virtual IP address subnet be shared between the primary and the secondary hosts, which can be onerous if you deploy your secondaries off site.

Therefore, Cloudera no longer recommend the use of virtual IP addresses, and instead recommends using a dedicated load balancer.

Single-User Mode, TLS, and Kerberos

High availability, as described in this document, supports the following:

- Single-user mode. You must run all commands as the `root` user (unless specified otherwise). These procedures do not alter or modify the behavior of how CDH services function.
- TLS and Kerberized deployments. For more information, see TLS and Kerberos Configuration for Cloudera Manager High Availability on page 479.

Cloudera Manager Failover Protection

A CDH cluster managed by Cloudera Manager can have only one instance of Cloudera Manager active at a time. A Cloudera Manager instance is backed by a single database instance that stores configurations and other operational data.

CDH deployments that use highly available configurations for Cloudera Manager can configure a "standby" instance of Cloudera Manager that takes over automatically if the primary instance fails. In some situations, a second instance of Cloudera Manager may become active during maintenance or upgrade activities or due to operator error. If two instances of Cloudera Manager are active at the same time and attempt to access the same database, data corruption can result, making Cloudera Manager unable to manage the cluster.

Cloudera Manager automatically detects when more than one instance of Cloudera Manager is running and logs a message in the `/var/log/cloudera-scm-server/cloudera-scm-server.log` file. For example:

```
2016-02-17 09:47:27,915 WARN
main:com.cloudera.server.cmf.components.ScmActive:
ScmActive detected spurious CM :
hostname=sysadmin-scm-2.mycompany.com/172.28.197.136,bootup true
2016-02-17 09:47:27,916 WARN
main:com.cloudera.server.cmf.components.ScmActive: ScmActive:
The database is owned by sysadmin-scm-1.mycompany.com/172.28.197.242
2016-02-17 09:47:27,917 ERROR
main:com.cloudera.server.cmf.bootstrap.EntityManagerFactoryBean: ScmActiveat bootup:
The configured database is being used by another instance of Cloudera Manager.
```

In addition, the second instance of Cloudera Manager is automatically shut down, resulting in messages similar to the following in the log file:

```
2016-02-17 09:47:27,919 ERROR main:com.cloudera.server.cmf.Main: Serverfailed.2016-02-17
 09:47:27,919
ERROR main:com.cloudera.server.cmf.Main:
Serverfailed.org.springframework.beans.factory.BeanCreationException:
Error creatingbean with name 'com.cloudera.server.cmf.TrialState':
Cannot resolvereference to bean 'entityManagerFactoryBean' while setting
constructorargument;
nested exception isorg.springframework.beans.factory.BeanCreationException:
Error creatingbean with name 'entityManagerFactoryBean':
FactoryBean threw exception onobject creation; nested exception is
java.lang.RuntimeException: ScmActiveat bootup:
Failed to validate the identity of Cloudera Manager.
```

When a Cloudera Manager instance fails or becomes unavailable and remains offline for more than 30 seconds, any new instance that is deployed claims ownership of the database and continues to manage the cluster normally.

Disabling Automatic Failover Protection

You can disable automatic shutdown by setting a Java option and restarting Cloudera Manager:

1. On the host where Cloudera Manager server is running, open the following file in a text editor:

```
/etc/default/cloudera-scm-server
```

2. Add the following property (separate each property with a space) to the line that begins with `export` `CMF_JAVA_OPTS`:

```
-Dcom.cloudera.server.cmf.components.scmActive.killOnError=false
```

For example:

```
export CMF_JAVA_OPTS="-Xmx2G -XX:MaxPermSize=256m -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/tmp -Dcom.cloudera.server.cmf.components.scmActive.killOnError=false"
```

3. Restart the Cloudera Manager server by running the following command on the Cloudera Manager server host:

```
sudo service cloudera-scm-server restart
```

> **Note:** When you disable automatic shutdown, a message is still logged when more than one instance of Cloudera Manager is running .

High-Level Steps to Configure Cloudera Manager High Availability

To configure Cloudera Manager for high availability, follow these high-level steps. Click each step to see detailed procedures.

> **Important:**
>
> Unless stated otherwise, run all commands mentioned in this topic as the `root` user.
>
> You do not need to stop the CDH cluster to configure Cloudera Manager high availability.

- Step 1: Setting Up Hosts and the Load Balancer on page 453.
- Step 2: Installing and Configuring Cloudera Manager Server for High Availability on page 459.
- Step 3: Installing and Configuring Cloudera Management Service for High Availability on page 462.
- Step 4: Automating Failover with Corosync and Pacemaker on page 468.

Step 1: Setting Up Hosts and the Load Balancer

At a high level, you set up Cloudera Manager Server and Cloudera Management Service *roles* (including Cloudera Navigator) on separate hosts, and make sure that network access to those hosts from other Cloudera services and to the Admin Console occurs through the configured load balancer.

Cloudera Manager Server, Cloudera Navigator, and all of the Cloudera Management Service roles that use a relational database should use an external database server, located off-host. You must make sure that these databases are configured to be highly available. See Database High Availability Configuration on page 479.

You configure other Cloudera Management Service roles (such as the Service Monitor and Host Monitor roles) that use a file-backed storage mechanism to store their data on a shared NFS storage mechanism.
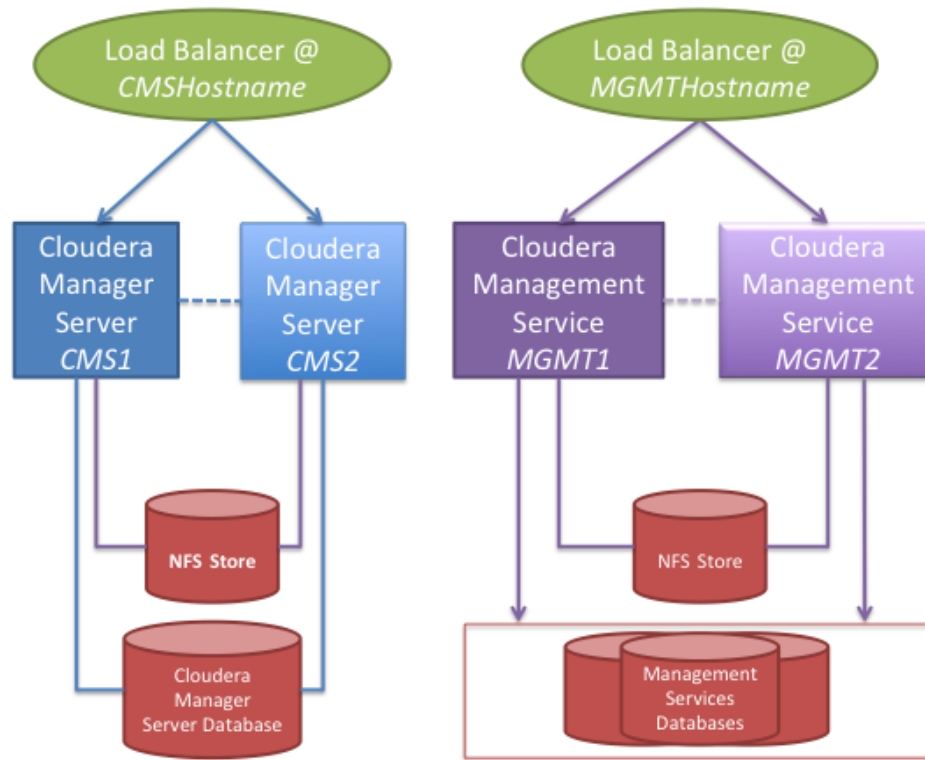
Figure 9: High-level layout of components for Cloudera Manager high availability

Creating Hosts for Primary and Secondary Servers

For this example, Cloudera recommends using four hosts for Cloudera Manager services. All of these hosts must resolve forward and reverse DNS lookups correctly:

- Cloudera Manager Server primary host (hostname: *CMS1*)
- Cloudera Management Service primary host (hostname: *MGMT1*)
- Cloudera Manager Server secondary host (hostname: *CMS2*)
- Cloudera Management Service secondary host (hostname: *MGMT2*)

> **Note:** The hostnames used here are placeholders and are used throughout this document. When configuring your cluster, substitute the actual names of the hosts you use in your environment.

In addition, Cloudera recommends the following:

- Do not host the Cloudera Manager or Cloudera Management Service roles on existing hosts in a CDH cluster, because this complicates failover configuration, and overlapping failure domains can cause problems with fault containment and error tracing.

- Configure both the primary and the secondary hosts using the same host configuration. This helps to ensure that failover does not lead to decreased performance.

- Host the primary and secondary hosts on separate power and network segments within your organization to limit overlapping failure domains.

Setting up the Load Balancer

This procedure demonstrates configuring the load balancer as two separate software load balancers using HAProxy, on two separate hosts for demonstration clarity. (To reduce cost, you might prefer to set up a single load balancer with two network interfaces.) You use one HAProxy host for Cloudera Manager Server and another for the Cloudera Management Service.

> **Note:** HAProxy is used here for demonstration purposes. Production-level performance requirements determine the load balancer that you select for your installation. HAProxy version 1.5.2 is used for these procedures.
>
> HAProxy 1.5.4-2 has a bug that affects the functioning of tcp-check. Cloudera recommends that you use version 1.6.3.

1. Reserve two hostnames in your DNS system, and assign them to each of the load balancer hosts. (The names *CMSHostname*, and *MGMTHostname* are used in this example; substitute the correct hostname for your environment.) These hostnames will be the externally accessible hostnames for Cloudera Manager Server and Cloudera Management Service. (Alternatively, use one load balancer with separate, resolvable IP addresses—one each to back *CMSHostname* and *MGMTHostname* respectively).

   - *CMSHostname* is used to access Cloudera Manager Admin Console.
   - *MGMTHostname* is used for internal access to the Cloudera Management Service from Cloudera Manager Server and Cloudera Manager Agents.

2. Set up two hosts using any supported Linux distribution (RHEL, CentOS, Ubuntu or SUSE; see CDH and Cloudera Manager Supported Operating Systems) with the hostnames listed above. See the HAProxy documentation for recommendations on configuring the hardware of these hosts.

3. Install the version of HAProxy that is recommended for the version of Linux installed on the two hosts:

   RHEL/CentOS:

```
$ yum install haproxy
```

   Ubuntu (use a current Personal Package Archive (PPA) for 1.5 from http://haproxy.debian.net):

```
$ apt-get install haproxy
```

   SUSE:

```
$ zypper install haproxy
```

4. Configure HAProxy to autostart on both the *CMSHostname* and *MGMTHostname* hosts:

   RHEL, CentOS, and SUSE:

```
$ chkconfig haproxy on
```

   Ubuntu:

```
$ update-rc.d haproxy defaults
```

5. Configure HAProxy.

   - On *CMSHostname*, edit the /etc/haproxy/haproxy.cfg files and make sure that the ports listed at Ports Used by Cloudera Manager and Cloudera Navigator for "Cloudera Manager Server" are proxied. For Cloudera Manager, this list includes the following ports as defaults:

     - 7180
     - 7182

    — `7183`

Sample HAProxy Configuration for `CMSHostname`

```
listen cmf :7180
    mode tcp
    option tcplog
    server cmfhttp1 CMS1:7180 check
    server cmfhttp2 CMS2:7180 check

listen cmfavro :7182
    mode tcp
    option tcplog
    server cmfavro1 CMS1:7182 check
    server cmfavro2 CMS2:7182 check

#ssl pass-through, without termination
listen cmfhttps :7183
    mode tcp
    option tcplog
    server cmfhttps1 CMS1:7183 check
    server cmfhttps2 CMS2:7183 check
```

- On *MGMTHostname*, edit the `/etc/haproxy/haproxy.cfg` file and make sure that the ports for Cloudera Management Service are proxied (see [Ports Used by Cloudera Manager and Cloudera Navigator](#)). For Cloudera Manager, this list includes the following ports as defaults:

    — `5678`
    — `7184`
    — `7185`
    — `7186`
    — `7187`
    — `8083`
    — `8084`
    — `8086`
    — `8087`
    — `8091`
    — `9000`
    — `9994`
    — `9995`
    — `9996`
    — `9997`
    — `9998`
    — `9999`
    — `10101`

Example HAProxy Configuration for `MGMTHostname`

```
listen mgmt1 :5678
    mode tcp
    option tcplog
    server mgmt1a MGMT1 check
    server mgmt1b MGMT2 check

listen mgmt2 :7184
    mode tcp
    option tcplog
    server mgmt2a MGMT1 check
    server mgmt2b MGMT2 check

listen mgmt3 :7185
    mode tcp
    option tcplog
```

```
        server mgmt3a MGMT1 check
        server mgmt3b MGMT2 check
listen mgmt4 :7186
    mode tcp
    option tcplog
    server mgmt4a MGMT1 check
    server mgmt4b MGMT2 check
listen mgmt5 :7187
    mode tcp
    option tcplog
    server mgmt5a MGMT1 check
    server mgmt5b MGMT2 check

listen mgmt6 :8083
    mode tcp
    option tcplog
    server mgmt6a MGMT1 check
    server mgmt6b MGMT2 check
listen mgmt7 :8084
    mode tcp
    option tcplog
    server mgmt7a MGMT1 check
    server mgmt7b MGMT2 check
listen mgmt8 :8086
    mode tcp
    option tcplog
    server mgmt8a MGMT1 check
    server mgmt8b MGMT2 check
listen mgmt9 :8087
    mode tcp
    option tcplog
    server mgmt9a MGMT1 check
    server mgmt9b MGMT2 check
listen mgmt10 :8091
    mode tcp
    option tcplog
    server mgmt10a MGMT1 check
    server mgmt10b MGMT2 check
listen mgmt-agent :9000
    mode tcp
    option tcplog
    server mgmt-agenta MGMT1 check
    server mgmt-agentb MGMT2 check
listen mgmt11 :9994
    mode tcp
    option tcplog
    server mgmt11a MGMT1 check
    server mgmt11b MGMT2 check
listen mgmt12 :9995
    mode tcp
    option tcplog
    server mgmt12a MGMT1 check
    server mgmt12b MGMT2 check
listen mgmt13 :9996
    mode tcp
    option tcplog
    server mgmt13a MGMT1 check
    server mgmt13b MGMT2 check
listen mgmt14 :9997
    mode tcp
    option tcplog
    server mgmt14a MGMT1 check
    server mgmt14b MGMT2 check
listen mgmt15 :9998
    mode tcp
    option tcplog
    server mgmt15a MGMT1 check
    server mgmt15b MGMT2 check
listen mgmt16 :9999
    mode tcp
    option tcplog
    server mgmt16a MGMT1 check
    server mgmt16b MGMT2 check
```

```
listen mgmt17 :10101
    mode tcp
    option tcplog
    server mgmt17a MGMT1 check
    server mgmt17b MGMT2 check
```

After updating the configuration, restart HAProxy on both the *MGMTHostname* and *CMSHostname* hosts:

```
$ service haproxy restart
```

Setting up the Database

1. Create databases on your preferred external database server. See [Step 4: Install and Configure Databases](#).

   > **Important:** The embedded Postgres database cannot be configured for high availability and should not be used in a high-availability configuration.

2. Configure your databases to be highly available. Consult the vendor documentation for specific information.

   MySQL, PostgreSQL, and Oracle each have many options for configuring high availability. See [Database High Availability Configuration](#) on page 479 for some external references on configuring high availability for your Cloudera Manager databases.

Setting up an NFS Server

The procedures outlined for setting up the Cloudera Manager Server and Cloudera Management Service hosts presume there is a shared store configured that can be accessed from both the primary and secondary instances of these hosts. This usually requires that this store be accessible over the network, and can be one of a variety of remote storage mechanisms (such as an iSCSI drive, a SAN array, or an NFS server).

   > **Note:** Using NFS as a shared storage mechanism is used here for demonstration purposes. Refer to your Linux distribution documentation on production NFS configuration and security. Production-level performance requirements determine the storage that you select for your installation.

This section describes how to configure an NFS server and assumes that you understand how to configure highly available remote storage devices. Further details are beyond the scope and intent of this guide.

There are no intrinsic limitations on where this NFS server is located, but because overlapping failure domains can cause problems with fault containment and error tracing, Cloudera recommends that you not co-locate the NFS server with any CDH or Cloudera Manager servers or the load-balancer hosts detailed in this document.

1. Install NFS on your designated server:

   RHEL/CentOS

```
$ yum install nfs-utils nfs-utils-lib
```

   Ubuntu

```
$ apt-get install nfs-kernel-server
```

   SUSE

```
$ zypper install nfs-kernel-server
```

2. Start `nfs` and `rpcbind`, and configure them to autostart:

RHEL/CentOS:

```
$ chkconfig nfs on
$ service rpcbind start
$ service nfs start
```

Ubuntu:

```
$ update-rc.d nfs defaults
$ service rpcbind start
$ service nfs-kernel-server
```

SUSE:

```
$ chkconfig nfs on
$ service rpcbind start
$ service nfs-kernel-server start
```

> **Note:** Later sections describe mounting the shared directories and sharing them between the primary and secondary instances.

Step 2: Installing and Configuring Cloudera Manager Server for High Availability

You can use an existing Cloudera Manager installation and extend it to a high-availability configuration, as long as you are not using the embedded PostgreSQL database.

This section describes how to install and configure a failover secondary for Cloudera Manager Server that can take over if the primary fails.

This section does not cover installing instances of Cloudera Manager Agent on *CMS1* or *CMS2* and configuring them to be highly available. See Cloudera Installation Guide.

Setting up NFS Mounts for Cloudera Manager Server

1. Create the following directories on the NFS server you created in a previous step:

```
$ mkdir -p /media/cloudera-scm-server
```

2. Mark these mounts by adding these lines to the /etc/exports file on the NFS server:

```
/media/cloudera-scm-server CMS1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-server CMS2(rw,sync,no_root_squash,no_subtree_check)
```

3. Export the mounts by running the following command on the NFS server:

```
$ exportfs -a
```

4. Set up the filesystem mounts on *CMS1* and *CMS2* hosts:

   a. If you are updating an existing installation for high availability, stop the Cloudera Manager Server if it is running on either of the *CMS1* or *CMS2* hosts by running the following command:

```
$ service cloudera-scm-server stop
```

   b. Make sure that the NFS mount helper is installed:

   RHEL/CentOS:

```
$ yum install nfs-utils-lib
```

Ubuntu:

```
$ apt-get install nfs-common
```

SUSE:

```
$ zypper  install nfs-client
```

    c. Make sure that `rpcbind` is running and has been restarted:

```
$ service rpcbind restart
```

5. Create the mount points on both `CMS1` and `CMS2`:

    a. If you are updating an existing installation for high availability, copy the `/var/lib/cloudera-scm-server` file from your existing Cloudera Manager Server host to the NFS server with the following command (`NFS` refers to the NFS server you created in a previous step):

```
$ scp -r /var/lib/cloudera-scm-server/ NFS:/media/cloudera-scm-server
```

    b. Set up the `/var/lib/cloudera-scm-server` directory on the `CMS1` and `CMS2` hosts:

```
$ rm -rf /var/lib/cloudera-scm-server
$ mkdir -p /var/lib/cloudera-scm-server
```

    c. Mount the following directory to the NFS mounts, on both `CMS1` and `CMS2`:

```
$ mount -t nfs NFS:/media/cloudera-scm-server /var/lib/cloudera-scm-server
```

    d. Set up `fstab` to persist the mounts across restarts by editing the `/etc/fstab` file on `CMS1` and `CMS2` and adding the following lines:

```
NFS:/media/cloudera-scm-server /var/lib/cloudera-scm-server nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

Installing the Primary
Updating an Existing Installation for High Availability

You can retain your existing Cloudera Manager Server as-is, if the deployment meets the following conditions:

- The Cloudera Management Service is located on a single host that is not the host where Cloudera Manager Server runs.
- The data directories for the roles of the Cloudera Management Service are located on a remote storage device (such as an NFS store), and they can be accessed from both primary and secondary installations of the Cloudera Management Service.

If your deployment does not meet these conditions, Cloudera recommends that you uninstall Cloudera Management Services by stopping the existing service and deleting it.

> **Important:** Deleting the Cloudera Management Service leads to loss of all existing data from the Host Monitor and Service Monitor roles that store health and monitoring information for your cluster on the local disk associated with the host(s) where those roles are installed.

To delete and remove the Cloudera Management Service:

1. Open the Cloudera Manager Admin Console and go to the **Home** page.

2. Click **Cloudera Management Service** > **Stop**.

3. Click **Cloudera Management Service** > **Delete**.

Fresh Installation

Follow the instructions in Installing Cloudera Manager, CDH, and Managed Services to install Cloudera Manager Server, but do not add "Cloudera Management Service" to your deployment until you complete Step 3: Installing and Configuring Cloudera Management Service for High Availability on page 462, which describes how to set up the Cloudera Management Service.

See:

- Installing Cloudera Manager, CDH, and Managed Services

You can now start the freshly-installed Cloudera Manager Server on *CMS1*:

```
$ service cloudera-scm-server start
```

Before proceeding, verify that you can access the Cloudera Manager Admin Console at `http://CMS1:7180`.

If you have just installed Cloudera Manager, click the Cloudera Manager logo to skip adding new hosts and to gain access to the Administration menu, which you need for the following steps.

HTTP Referer Configuration

Cloudera recommends that you disable the HTTP Referer check because it causes problems for some proxies and load balancers. Check the configuration manual of your proxy or load balancer to determine if this is necessary.

To disable HTTP Referer in the Cloudera Manager Admin Console:

1. Select **Administration** > **Settings**.
2. Select **Category** > **Security**.
3. Clear the **HTTP Referer Check** property.

Before proceeding, verify that you can access the Cloudera Manager Admin Console through the load balancer at `http://CMSHostname:7180`.

TLS and Kerberos Configuration

To configure Cloudera Manager to use TLS encryption or authentication, or to use Kerberos authentication, see TLS and Kerberos Configuration for Cloudera Manager High Availability on page 479.

Installing the Secondary

Setting up the Cloudera Manager Server secondary requires copying certain files from the primary to ensure that they are consistently initialized.

1. On the *CMS2* host, install the `cloudera-manager-server` package using Installing Cloudera Manager, CDH, and Managed Services.
2. When setting up the database on the secondary, copy the `/etc/cloudera-scm-server/db.properties` file from host *CMS1* to host *CMS2* at `/etc/cloudera-scm-server/db.properties`. For example:

```
$ mkdir -p /etc/cloudera-scm-server
$ scp [<ssh-user>@]CMS1:/etc/cloudera-scm-server/db.properties
/etc/cloudera-scm-server/db.properties
```

3. If you configured Cloudera Manager TLS encryption or authentication, or Kerberos authentication in your primary installation, see TLS and Kerberos Configuration for Cloudera Manager High Availability on page 479 for additional configuration steps.
4. Do not start the `cloudera-scm-server` service on this host yet, and disable autostart on the secondary to avoid automatically starting the service on this host.

   RHEL/CentOS/SUSEL:

```
$ chkconfig cloudera-scm-server off
```

Ubuntu:

```
$ update-rc.d -f cloudera-scm-server remove
```

(You will also disable autostart on the primary when you configure [automatic failover](#) in a later step.) Data corruption can result if both primary and secondary Cloudera Manager Server instances are running at the same time, and it is not supported. :

Testing Failover

Test failover manually by using the following steps:

1. Stop `cloudera-scm-server` on your primary host (*CMS1*):

```
$ service cloudera-scm-server stop
```

2. Start `cloudera-scm-server` on your secondary host (*CMS2*):

```
$ service cloudera-scm-server start
```

3. Wait a few minutes for the service to load, and then access the Cloudera Manager Admin Console through a web browser, using the load-balanced hostname (for example: `http://CMSHostname:CMS_port`).

Now, fail back to the primary before configuring the Cloudera Management Service on your installation:

1. Stop `cloudera-scm-server` on your secondary machine (*CMS2*):

```
$ service cloudera-scm-server stop
```

2. Start `cloudera-scm-server` on your primary machine (*CMS1*):

```
$ service cloudera-scm-server start
```

3. Wait a few minutes for the service to load, and then access the Cloudera Manager Admin Console through a web browser, using the load-balanced hostname (for example: `http://CMSHostname:7180`).

Updating Cloudera Manager Agents to use the Load Balancer

After completing the primary and secondary installation steps listed previously, update the Cloudera Manager Agent configuration on all of the hosts associated with this Cloudera Manager installation, except the *MGMT1*, *MGMT2*, *CMS1*, and *CMS2* hosts, to use the load balancer address:

1. Connect to a shell on each host where CDH processes are installed and running. (The *MGMT1*, *MGMT2*, *CMS1*, and *CMS2* hosts do not need to be modified as part of this step.)
2. Update the `/etc/cloudera-scm-agent/config.ini` file and change the `server_host` line:

```
server_host = <CMSHostname>
```

3. Restart the agent (this command starts the agents if they are not running):

```
$ service cloudera-scm-agent restart
```

Step 3: Installing and Configuring Cloudera Management Service for High Availability

This section demonstrates how to set up shared mounts on *MGMT1* and *MGMT2*, and then install Cloudera Management Service to use those mounts on the primary and secondary servers.

> **!** **Important:** Do not start the primary and secondary servers that are running Cloudera Management Service at the same time. Data corruption can result.

Setting up NFS Mounts for Cloudera Management Service

1. Create directories on the NFS server:

```
$ mkdir -p /media/cloudera-host-monitor
$ mkdir -p /media/cloudera-scm-agent
$ mkdir -p /media/cloudera-scm-eventserver
$ mkdir -p /media/cloudera-scm-headlamp
$ mkdir -p /media/cloudera-service-monitor
$ mkdir -p /media/cloudera-scm-navigator
$ mkdir -p /media/etc-cloudera-scm-agent
```

2. Mark these mounts by adding the following lines to the `/etc/exports` file on the NFS server:

```
/media/cloudera-host-monitor MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-agent MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-eventserver MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-headlamp MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-service-monitor MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-navigator MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/etc-cloudera-scm-agent MGMT1(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-host-monitor MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-agent MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-eventserver MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-headlamp MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-service-monitor MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/cloudera-scm-navigator MGMT2(rw,sync,no_root_squash,no_subtree_check)
/media/etc-cloudera-scm-agent MGMT2(rw,sync,no_root_squash,no_subtree_check)
```

3. Export the mounts running the following command on the NFS server:

```
$ exportfs -a
```

4. Set up the filesystem mounts on *MGMT1* and *MGMT2* hosts:

    a. Make sure that the NFS mount helper is installed:

    RHEL/CentOS:

```
$ yum install nfs-utils-lib
```

    Ubuntu:

```
$ apt-get install nfs-common
```

    SUSE:

```
$ zypper  install nfs-client
```

    b. Create the mount points on both *MGMT1* and *MGMT2*:

```
$ mkdir -p /var/lib/cloudera-host-monitor
$ mkdir -p /var/lib/cloudera-scm-agent
$ mkdir -p /var/lib/cloudera-scm-eventserver
$ mkdir -p /var/lib/cloudera-scm-headlamp
$ mkdir -p /var/lib/cloudera-service-monitor
$ mkdir -p /var/lib/cloudera-scm-navigator
$ mkdir -p /etc/cloudera-scm-agent
```

    c. Mount the following directories to the NFS mounts, on both *MGMT1* and *MGMT2* (*NFS* refers to the server NFS hostname or IP address):

```
$ mount -t nfs NFS:/media/cloudera-host-monitor /var/lib/cloudera-host-monitor
$ mount -t nfs NFS:/media/cloudera-scm-agent /var/lib/cloudera-scm-agent
$ mount -t nfs NFS:/media/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserver
```

```
$ mount -t nfs NFS:/media/cloudera-scm-headlamp /var/lib/cloudera-scm-headlamp
$ mount -t nfs NFS:/media/cloudera-service-monitor /var/lib/cloudera-service-monitor
$ mount -t nfs NFS:/media/cloudera-scm-navigator /var/lib/cloudera-scm-navigator
$ mount -t nfs NFS:/media/etc-cloudera-scm-agent /etc/cloudera-scm-agent
```

5. Set up `fstab` to persist the mounts across restarts. Edit the `/etc/fstab` file and add these lines:

```
NFS:/media/cloudera-host-monitor /var/lib/cloudera-host-monitor nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/cloudera-scm-agent /var/lib/cloudera-scm-agent nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserver nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/cloudera-scm-headlamp /var/lib/cloudera-scm-headlamp nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/cloudera-service-monitor /var/lib/cloudera-service-monitor nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/cloudera-scm-navigator /var/lib/cloudera-scm-navigator nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
NFS:/media/etc-cloudera-scm-agent /etc/cloudera-scm-agent nfs
auto,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

Installing the Primary

1. Connect to a shell on *MGMT1*, and then install the `cloudera-manager-daemons` and `cloudera-manager-agent` packages:

  a. Install packages `cloudera-manager-daemons` and `cloudera-manager-agent` packages using instructions from [Installing Cloudera Manager, CDH, and Managed Services](#).

  b. Install the Oracle Java JDK version that is required for your deployment, if it is not already installed on the host. See [Java Requirements](#).

2. Configure the agent to report its hostname as *<MGMTHostname>* to Cloudera Manager. This ensures that the connections from the Cloudera Manager Agents on the CDH cluster hosts report to the correct Cloudera Management Service host in the event of a failover.

  a. Edit the `/etc/cloudera-scm-agent/config.ini` file to update the following lines:

```
server_host=CMSHostname
listening_hostname=MGMTHostname
```

  b. Edit the `/etc/hosts` file and add *MGMTHostname* as an alias for your public IP address for *MGMT1* by adding a line like this at the end of your `/etc/hosts` file:

```
 MGMT1 IP MGMTHostname
```

  c. Confirm that the alias has taken effect by running the `ping` command. For example:

```
[root@MGMT1 ~]# ping MGMTHostname
PING MGMTHostname (MGMT1 IP) 56(84) bytes of data.
64 bytes from MGMTHostname (MGMT1 IP): icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from MGMTHostname (MGMT1 IP): icmp_seq=2 ttl=64 time=0.018 ms
...
```

  d. Make sure that the `cloudera-scm` user and the `cloudera-scm` group have access to the mounted directories under `/var/lib`, by using the `chown` command on `cloudera-scm`. For example, run the following on *MGMT1*:

```
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-scm-eventserver
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-scm-navigator
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-service-monitor
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-host-monitor
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-scm-agent
$ chown -R cloudera-scm:cloudera-scm /var/lib/cloudera-scm-headlamp
```

> **Note:** The `cloudera-scm` user and the `cloudera-scm` group are the default owners as specified in Cloudera Management Service advanced configuration. If you alter these settings, modify the above `chown` instructions to use the altered user or group name.

    e. Restart the agent on *MGMT1* (this also starts the agent if it is not running):

```
$ service cloudera-scm-agent restart
```

    f. Connect to the Cloudera Manager Admin Console running on *<CMSHostname>* and:

        a. Go to the **Hosts** tab and make sure that a host with name *<MGMTHostname>* is reported. (If it is not available yet, wait for it to show up before you proceed.)

        b. Click **Add Cloudera Management Service**.

    g. Make sure you install all of the roles of the Cloudera Management Service on the host named *MGMTHostname*.

    h. Proceed through the steps to configure the roles of the service to use your database server, and use defaults for the storage directory for Host Monitor or Service Monitor.

    i. After you have completed the steps, wait for the Cloudera Management Service to finish starting, and verify the health status of your clusters as well as the health of the Cloudera Management Service as reported in the Cloudera Manager Admin Console. The health status indicators should be green, as shown:

The service health for Cloudera Management Service might, however, show as red:

In this case, you need to identify whether the health test failure is caused by the **Hostname and Canonical Name Health Check** for the `MGMTHostname` host, which might look like this:

This test can fail in this way because of the way you modified `/etc/hosts` on `MGMT1` and `MGMT2` to allow the resolution of `MGMTHostname` locally. This test can be safely disabled on the `MGMTHostname` host from the Cloudera Manager Admin Console.

j. If you are configuring Kerberos and TLS/SSL, see TLS and Kerberos Configuration for Cloudera Manager High Availability on page 479 for configuration changes as part of this step.

Installing the Secondary

1. Stop all Cloudera Management Service roles using the Cloudera Manager Admin Console:

a. On the **Home** > **Status** tab, click

to the right of **Cloudera Management Service** and select **Stop**.

b. Click **Stop** to confirm. The **Command Details** window shows the progress of stopping the roles.

c. When **Command completed with *n/n* successful subcommands** appears, the task is complete. Click **Close**.

2. Stop the `cloudera-scm-agent` service on `MGMT1`:

```
$ service cloudera-scm-agent stop
```

3. Connect to a shell on MGMT2, and then install `cloudera-manager-daemons` and `cloudera-manager-agent`:

a. Install the `cloudera-manager-daemons` and `cloudera-manager-agent` packages using instructions from Installing Cloudera Manager, CDH, and Managed Services.

b. Install the Oracle Java JDK version that is required for your deployment, if it is not already installed on the host. See Java Requirements.

4. Configure the agent to report its hostname as *MGMTHostname* to Cloudera Manager, as described previously in
Installing the Primary on page 464.

    a. Make sure that `/etc/cloudera-scm-agent/config.ini` has the following lines (because this is a shared
mount with the primary, it should be the same as in the primary installation):

```
server_host=<CMHostname>
listening_hostname=<MGMTHostname>
```

    b. Edit the `/etc/hosts` file and add *MGMTHostname* as an alias for your public IP address for *MGMT2*, by adding
a line like this at the end of your `/etc/hosts` file:

```
<MGMT2-IP> <MGMTHostname>
```

    c. Confirm that the alias is working by running the `ping` command. For example:

```
[root@MGMT2 ~]# ping MGMTHostname
PING MGMTHostname (MGMT2 IP) 56(84) bytes of data.
64 bytes from MGMTHostname (MGMT2 IP): icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from MGMTHostname (MGMT2 IP): icmp_seq=2 ttl=64 time=0.018 ms
```

5. Start the agent on *MGMT2* by running the following command:

```
$ service cloudera-scm-agent start
```

6. Log into the Cloudera Manager Admin Console in a web browser and start all Cloudera Management Service roles.

This starts the Cloudera Management Service on *MGMT2*.

    a. Wait for the Cloudera Manager Admin Console to report that the services have started.

    b. Confirm that the services have started on this host by running the following command on *MGMT2*:

```
$ ps -elf | grep "scm"
```

You should see ten total processes running on that host, including the eight Cloudera Management Service
processes, a Cloudera Manager Agent process, and a Supervisor process.

    c. Test the secondary installation through the Cloudera Management Admin Console, and inspect the health
of the Cloudera Management Service roles, before proceeding.

> **Note:**
>
> Make sure that the UID and GID for the `cloudera-scm` user on the primary and secondary Cloudera
> Management Service hosts are same; this ensures that the correct permissions are available on the
> shared directories after failover.

Failing Back to the Primary

Before finishing the installation, fail back to the primary host (*MGMT1*):

1. Stop the `cloudera-scm-agent` service on *MGMT2*:

```
$ service cloudera-scm-agent hard_stop_confirmed
```

2. Start the `cloudera-scm-agent` service on *MGMT1*:

```
$ service cloudera-scm-agent start
```

Step 4: Automating Failover with Corosync and Pacemaker

Corosync and Pacemaker are popular high-availability utilities that allow you to configure Cloudera Manager to fail over automatically.

This document describes one way to set up clustering using these tools. Actual setup can be done in several ways, depending on the network configuration of your environment.

Prerequisites:

1. Install Pacemaker and Corosync on *CMS1*, *MGMT1*, *CMS2*, and *MGMT2*, using the correct versions for your Linux distribution:

   > **Note:** The versions referred to for setting up automatic failover in this document are Pacemaker 1.1.11 and Corosync 1.4.7. See http://clusterlabs.org/wiki/Install to determine what works best for your Linux distribution.

   RHEL/CentOS:

   ```
   $ yum install pacemaker corosync
   ```

   Ubuntu:

   ```
   $ apt-get install pacemaker corosync
   ```

   SUSE:

   ```
   $ zypper install pacemaker corosync
   ```

2. Make sure that the `crm` tool exists on all of the hosts. This procedure uses the `crm` tool, which works with Pacemaker configuration. If this tool is not installed when you installed Pacemaker (verify this by running `which crm`), you can download and install the tool for your distribution using the instructions at http://crmsh.github.io/installation.

About Corosync and Pacemaker

- By default, Corosync and Pacemaker are not autostarted as part of the boot sequence. Cloudera recommends leaving this as is. If the machine crashes and restarts, manually make sure that failover was successful and determine the cause of the restart before manually starting these processes to achieve higher availability.

  - If the `/etc/default/corosync` file exists, make sure that `START` is set to `yes` in that file:

    ```
    START=yes
    ```

  - Make sure that Corosync is not set to start automatically, by running the following command:

    RHEL/CentOS/SUSE:

    ```
    $ chkconfig corosync off
    ```

    Ubuntu:

    ```
    $ update-rc.d -f corosync remove
    ```

- Note which version of Corosync is installed. The contents of the configuration file for Corosync (`corosync.conf`) that you edit varies based on the version suitable for your distribution. Sample configurations are supplied in this document and are labeled with the Corosync version.
- This document does not demonstrate configuring Corosync with authentication (with `secauth` set to `on`). The Corosync website demonstrates a mechanism to encrypt traffic using symmetric keys.
- Firewall configuration:

Corosync uses UDP transport on ports 5404 and 5405, and these ports must be open for both inbound and outbound traffic on all hosts. If you are using IP tables, run a command similar to the following:

```
$ sudo iptables -I INPUT -m state --state NEW -p udp -m multiport --dports 5404,5405 -j
 ACCEPT
$ sudo iptables -I OUTPUT -m state --state NEW -p udp -m multiport --sports 5404,5405
-j ACCEPT
```

Setting up Cloudera Manager Server

Set up a Corosync cluster over unicast, between *CMS1* and *CMS2*, and make sure that the hosts can "cluster" together. Then, set up Pacemaker to register Cloudera Manager Server as a resource that it monitors and to fail over to the secondary when needed.

Setting up Corosync

1. Edit the `/etc/corosync/corosync.conf` file on *CMS1* and replace the entire contents with the following text (use the correct version for your environment):

   Corosync version 1.x:

```
compatibility: whitetank
totem {
        version: 2
        secauth: off
        interface {
                member {
                        memberaddr: CMS1
                }
                member {
                        memberaddr: CMS2
                }
                ringnumber: 0
                bindnetaddr: CMS1
                mcastport: 5405
        }
        transport: udpu
}

logging {
        fileline: off
        to_logfile: yes
        to_syslog: yes
        logfile: /var/log/cluster/corosync.log
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
        }
}
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  1
        #
}
```

   Corosync version 2.x:

```
totem {
version: 2
secauth: off
cluster_name: cmf
transport: udpu
}

nodelist {
  node {
```

```
          ring0_addr: CMS1
          nodeid: 1
        }
  node {
          ring0_addr: CMS2
          nodeid: 2
        }
}

quorum {
provider: corosync_votequorum
two_node: 1
}
```

2. Edit the `/etc/corosync/corosync.conf` file on *CMS2*, and replace the entire contents with the following text (use the correct version for your environment):

Corosync version 1.x:

```
compatibility: whitetank
totem {
        version: 2
        secauth: off
        interface {
                member {
                        memberaddr: CMS1
                }
                member {
                        memberaddr: CMS2
                }
                ringnumber: 0
                bindnetaddr: CMS2
                mcastport: 5405
        }
        transport: udpu
}

logging {
        fileline: off
        to_logfile: yes
        to_syslog: yes
        logfile: /var/log/cluster/corosync.log
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
        }
}
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  1
        #
}
```

Corosync version 2.x:

```
totem {
version: 2
secauth: off
cluster_name: cmf
transport: udpu
}

nodelist {
  node {
        ring0_addr: CMS1
        nodeid: 1
        }
  node {
```

```
        ring0_addr: CMS2
        nodeid: 2
        }
}

quorum {
provider: corosync_votequorum
two_node: 1
}
```

3. Restart Corosync on *CMS1* and *CMS2* so that the new configuration takes effect:

```
$ service corosync restart
```

Setting up Pacemaker

You use Pacemaker to set up Cloudera Manager Server as a *cluster resource*.

See the Pacemaker configuration reference at
http://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/ for more details about
Pacemaker options.

The following steps demonstrate one way, recommended by Cloudera, to configure Pacemaker for simple use:

1. Disable autostart for Cloudera Manager Server (because you manage its lifecycle through Pacemaker) on both
CMS1 and *CMS2*:

RHEL/CentOS/SUSE:

```
$ chkconfig cloudera-scm-server off
```

Ubuntu:

```
$ update-rc.d -f cloudera-scm-server remove
```

2. Make sure that Pacemaker has been started on both *CMS1* and *CMS2*:

```
$ /etc/init.d/pacemaker start
```

3. Make sure that `crm` reports two nodes in the cluster:

```
# crm status
Last updated: Wed Mar  4 18:55:27 2015
Last change: Wed Mar  4 18:38:40 2015 via crmd on CMS1
Stack: corosync
Current DC: CMS1 (1) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
0 Resources configured
```

4. Change the Pacemaker cluster configuration (on either *CMS1* or *CMS2*):

```
$ crm configure property no-quorum-policy=ignore
$ crm configure property stonith-enabled=false
$ crm configure rsc_defaults resource-stickiness=100
```

These commands do the following:

- Disable quorum checks. (Because there are only two nodes in this cluster, quorum cannot be established.)
- Disable STONITH explicitly (see Enabling STONITH (Shoot the other node in the head) on page 472).
- Reduce the likelihood of the resource being moved among hosts on restarts.

5. Add Cloudera Manager Server as an LSB-managed resource (either on *CMS1* or *CMS2*):

```
$ crm configure primitive cloudera-scm-server lsb:cloudera-scm-server
```

6. Verify that the primitive has been picked up by Pacemaker:

```
$ crm_mon
```

For example:

```
$ crm_mon
Last updated: Tue Jan 27 15:01:35 2015
Last change: Mon Jan 27 14:10:11 2015
Stack: classic openais (with plugin)
Current DC: CMS1 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
1 Resources configured
Online: [ CMS1 CMS2 ]
cloudera-scm-server (lsb:cloudera-scm-server): Started CMS1
```

At this point, Pacemaker manages the status of the `cloudera-scm-server` service on hosts *CMS1* and *CMS2*, ensuring that only one instance is running at a time.

> **Note:** Pacemaker expects all lifecycle actions, such as start and stop, to go through Pacemaker; therefore, running direct `service start` or `service stop` commands breaks that assumption.

Testing Failover with Pacemaker

Test Pacemaker failover by running the following command to move the `cloudera-scm-server` resource to *CMS2*:

```
$ crm resource move cloudera-scm-server <CMS2>
```

Test the resource move by connecting to a shell on *CMS2* and verifying that the `cloudera-scm-server` process is now active on that host. It takes usually a few minutes for the new services to come up on the new host.

Enabling STONITH (Shoot the other node in the head)

The following link provides an explanation of the problem of fencing and ensuring (within reasonable limits) that only one host is running a shared resource at a time:
http://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html-single/Clusters_from_Scratch/index.html#idm140603947390416

As noted in that link, you can use several methods (such as IPMI) to achieve reasonable guarantees on remote host shutdown. Cloudera recommends enabling STONITH, based on the hardware configuration in your environment.

Setting up the Cloudera Manager Service
Setting Up Corosync

1. Edit the `/etc/corosync/corosync.conf` file on *MGMT1* and replace the entire contents with the contents below; make sure to use the correct section for your version of Corosync:

Corosync version 1.x:

```
compatibility: whitetank
totem {
        version: 2
        secauth: off
        interface {
                member {
                        memberaddr: MGMT1
                }
                member {
                        memberaddr: MGMT2
                }
```

```
                    ringnumber: 0
                    bindnetaddr: MGMT1
                    mcastport: 5405
          }
          transport: udpu
}

logging {
          fileline: off
          to_logfile: yes
          to_syslog: yes
          logfile: /var/log/cluster/corosync.log
          debug: off
          timestamp: on
          logger_subsys {
                    subsys: AMF
                    debug: off
          }
}
service {
          # Load the Pacemaker Cluster Resource Manager
          name: pacemaker
          ver:  1
          #
}
```

   Corosync version 2.x:

```
totem {
version: 2
secauth: off
cluster_name: mgmt
transport: udpu
}

nodelist {
   node {
          ring0_addr: MGMT1
          nodeid: 1
          }
   node {
          ring0_addr: MGMT2
          nodeid: 2
          }
}

quorum {
provider: corosync_votequorum
two_node: 1
}
```

   2. Edit the `/etc/corosync/corosync.conf` file on *MGMT2* andf replace the contents with the contents below:

   Corosync version 1.x:

```
compatibility: whitetank
totem {
          version: 2
          secauth: off
          interface {
                    member {
                              memberaddr: MGMT1
                    }
                    member {
                              memberaddr: MGMT2
                    }
                    ringnumber: 0
                    bindnetaddr: MGMT2
                    mcastport: 5405
          }
          transport: udpu
```

```
}
logging {
        fileline: off
        to_logfile: yes
        to_syslog: yes
        logfile: /var/log/cluster/corosync.log
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
        }
}
service {
        # Load the Pacemaker Cluster Resource Manager
        name: pacemaker
        ver:  1
        #
}
```

Corosync version 2.x:

```
totem {
version: 2
secauth: off
cluster_name: mgmt
transport: udpu
}

nodelist {
  node {
        ring0_addr: CMS1
        nodeid: 1
        }
  node {
        ring0_addr: CMS2
        nodeid: 2
        }
}

quorum {
provider: corosync_votequorum
two_node: 1
}
```

3. Restart Corosync on *MGMT1* and *MGMT2* for the new configuration to take effect:

```
$ service corosync restart
```

4. Test whether Corosync has set up a cluster, by using the `corosync-cmapctl` or `corosync-objctl` commands. You should see two members with status **joined**:

```
corosync-objctl | grep "member"
runtime.totem.pg.mrp.srp.members.1.config_version (u64) = 0
runtime.totem.pg.mrp.srp.members.1.ip (str) = r(0) ip(MGMT1)
runtime.totem.pg.mrp.srp.members.1.join_count (u32) = 1
runtime.totem.pg.mrp.srp.members.1.status (str) = joined
runtime.totem.pg.mrp.srp.members.2.config_version (u64) = 0
runtime.totem.pg.mrp.srp.members.2.ip (str) = r(0) ip(MGMT2)
runtime.totem.pg.mrp.srp.members.2.join_count (u32) = 1
runtime.totem.pg.mrp.srp.members.2.status (str) = joined
```

Setting Up Pacemaker

Use Pacemaker to set up Cloudera Management Service as a *cluster resource*.

See the Pacemaker configuration reference at
http://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/ for more information about
Pacemaker options.

Because the lifecycle of Cloudera Management Service is managed through the Cloudera Manager Agent, you configure
the Cloudera Manager Agent to be highly available.

Follow these steps to configure Pacemaker, recommended by Cloudera for simple use:

1. Disable autostart for the Cloudera Manager Agent (because Pacemaker manages its lifecycle) on both *MGMT1* and
   MGMT2:

   RHEL/CentOS/SUSE

   ```
   $ chkconfig cloudera-scm-agent off
   ```

   Ubuntu:

   ```
   $ update-rc.d -f cloudera-scm-agent remove
   ```

2. Make sure that Pacemaker is started on both *MGMT1* and *MGMT2*:

   ```
   $ /etc/init.d/pacemaker start
   ```

3. Make sure that the `crm` command reports two nodes in the cluster; you can run this command on either host:

   ```
   # crm status
   Last updated: Wed Mar  4 18:55:27 2015
   Last change: Wed Mar  4 18:38:40 2015 via crmd on MGMT1
   Stack: corosync
   Current DC: MGMT1 (1) - partition with quorum
   Version: 1.1.10-42f2063
   2 Nodes configured
   0 Resources configured
   ```

4. Change the Pacemaker cluster configuration on either *MGMT1* or *MGMT2*:

   ```
   $ crm configure property no-quorum-policy=ignore
   $ crm configure property stonith-enabled=false
   $ crm configure rsc_defaults resource-stickiness=100
   ```

   As with Cloudera Manager Server Pacemaker configuration, this step disables quorum checks, disables STONITH
   explicitly, and reduces the likelihood of resources being moved between hosts.

5. Create an Open Cluster Framework (OCF) provider on both *MGMT1* and *MGMT2* for Cloudera Manager Agent for
   use with Pacemaker:

   a. Create an OCF directory for creating OCF resources for Cloudera Manager:

   ```
   $ mkdir -p /usr/lib/ocf/resource.d/cm
   ```

   b. Create a Cloudera Manager Agent OCF wrapper as a file at `/usr/lib/ocf/resource.d/cm/agent`, with
      the following content, on both *MGMT1* and *MGMT2*:

      • RHEL-compatible 7 and higher:

   ```
   #!/bin/sh
   ###################################################################
   # CM Agent OCF script
   ###################################################################
   ###################################################################
   # Initialization:
   : ${__OCF_ACTION=$1}
   OCF_SUCCESS=0
   ```

```
        OCF_ERROR=1
        OCF_STOPPED=7
        ####################################################################

        meta_data() {
                cat <<END
        <?xml version="1.0"?>
        <!DOCTYPE resource-agent SYSTEM "ra-api-1.dtd">
        <resource-agent name="Cloudera Manager Agent" version="1.0">
        <version>1.0</version>

        <longdesc lang="en">
         This OCF agent handles simple monitoring, start, stop of the Cloudera
         Manager Agent, intended for use with Pacemaker/corosync for failover.
        </longdesc>
        <shortdesc lang="en">Cloudera Manager Agent OCF script</shortdesc>

        <parameters />

        <actions>
        <action name="start"        timeout="20" />
        <action name="stop"         timeout="20" />
        <action name="monitor"      timeout="20" interval="10" depth="0"/>
        <action name="meta-data"    timeout="5" />
        </actions>
        </resource-agent>
        END
        }

        ####################################################################

        agent_usage() {
        cat <<END
         usage: $0 {start|stop|monitor|meta-data}
         Cloudera Manager Agent HA OCF script - used for managing Cloudera Manager Agent and
        managed processes lifecycle for use with Pacemaker.
        END
        }

        agent_start() {
            service cloudera-scm-agent start
            if [ $? =  0 ]; then
                return $OCF_SUCCESS
            fi
            return $OCF_ERROR
        }

        agent_stop() {
            service cloudera-scm-agent next_stop_hard
            service cloudera-scm-agent stop
            if [ $? =  0 ]; then
                return $OCF_SUCCESS
            fi
            return $OCF_ERROR
        }

        agent_monitor() {
                # Monitor _MUST!_ differentiate correctly between running
                # (SUCCESS), failed (ERROR) or _cleanly_ stopped (NOT RUNNING).
                # That is THREE states, not just yes/no.
                service cloudera-scm-agent status
                if [ $? = 0 ]; then
                     return $OCF_SUCCESS
                fi
                return $OCF_STOPPED
        }


        case $__OCF_ACTION in
        meta-data)      meta_data
                        exit $OCF_SUCCESS
                        ;;
        start)          agent_start;;
```

```
stop)            agent_stop;;
monitor)         agent_monitor;;
usage|help)      agent_usage
                 exit $OCF_SUCCESS
                 ;;
*)               agent_usage
                 exit $OCF_ERR_UNIMPLEMENTED
                 ;;
esac
rc=$?
exit $rc
```

- All other Linux distributions:

```
#!/bin/sh
#######################################################################
# CM Agent OCF script
#######################################################################
#######################################################################
# Initialization:
: ${__OCF_ACTION=$1}
OCF_SUCCESS=0
OCF_ERROR=1
OCF_STOPPED=7
#######################################################################

meta_data() {
        cat <<END
<?xml version="1.0"?>
<!DOCTYPE resource-agent SYSTEM "ra-api-1.dtd">
<resource-agent name="Cloudera Manager Agent" version="1.0">
<version>1.0</version>

<longdesc lang="en">
 This OCF agent handles simple monitoring, start, stop of the Cloudera
 Manager Agent, intended for use with Pacemaker/corosync for failover.
</longdesc>
<shortdesc lang="en">Cloudera Manager Agent OCF script</shortdesc>

<parameters />

<actions>
<action name="start"        timeout="20" />
<action name="stop"         timeout="20" />
<action name="monitor"      timeout="20" interval="10" depth="0"/>
<action name="meta-data"    timeout="5" />
</actions>
</resource-agent>
END
}

#######################################################################

agent_usage() {
cat <<END
 usage: $0 {start|stop|monitor|meta-data}
 Cloudera Manager Agent HA OCF script - used for managing Cloudera Manager Agent and
managed processes lifecycle for use with Pacemaker.
END
}

agent_start() {
    service cloudera-scm-agent start
    if [ $? =  0 ]; then
        return $OCF_SUCCESS
    fi
    return $OCF_ERROR
}

agent_stop() {
    service cloudera-scm-agent hard_stop_confirmed
    if [ $? =  0 ]; then
```

```
        return $OCF_SUCCESS
    fi
    return $OCF_ERROR
}

agent_monitor() {
        # Monitor _MUST!_ differentiate correctly between running
        # (SUCCESS), failed (ERROR) or _cleanly_ stopped (NOT RUNNING).
        # That is THREE states, not just yes/no.
        service cloudera-scm-agent status
        if [ $? = 0 ]; then
            return $OCF_SUCCESS
        fi
        return $OCF_STOPPED
}


case $__OCF_ACTION in
meta-data)      meta_data
                exit $OCF_SUCCESS
                ;;
start)          agent_start;;
stop)           agent_stop;;
monitor)        agent_monitor;;
usage|help)     agent_usage
                exit $OCF_SUCCESS
                ;;
*)              agent_usage
                exit $OCF_ERR_UNIMPLEMENTED
                ;;
esac
rc=$?
exit $rc
```

      c. Run `chmod` on that file to make it executable:

```
$ chmod 770 /usr/lib/ocf/resource.d/cm/agent
```

  6. Test the OCF resource script:

```
$ /usr/lib/ocf/resource.d/cm/agent monitor
```

    This script should return the current running status of the SCM agent.

  7. Add Cloudera Manager Agent as an OCF-managed resource (either on *MGMT1* or *MGMT2*):

```
$ crm configure primitive cloudera-scm-agent ocf:cm:agent
```

  8. Verify that the primitive has been picked up by Pacemaker by running the following command:

```
$ crm_mon
```

    For example:

```
>crm_mon
Last updated: Tue Jan 27 15:01:35 2015
Last change: Mon Jan 27 14:10:11 2015ls /
Stack: classic openais (with plugin)
Current DC: CMS1 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
1 Resources configured
Online: [ MGMT1 MGMT2 ]
cloudera-scm-agent (ocf:cm:agent): Started MGMT2
```

Pacemaker starts managing the status of the `cloudera-scm-agent` service on hosts *MGMT1* and *MGMT2*, ensuring that only one instance is running at a time.

> **Note:** Pacemaker expects that all lifecycle actions, such as start and stop, go through Pacemaker; therefore, running direct `service start` or `service stop` commands on one of the hosts breaks that assumption and could cause Pacemaker to start the service on the other host.

Testing Failover with Pacemaker

Test that Pacemaker can move resources by running the following command, which moves the `cloudera-scm-agent` resource to *MGMT2*:

```
$ crm resource move cloudera-scm-agent MGMT2
```

Test the resource move by connecting to a shell on *MGMT2* and verifying that the `cloudera-scm-agent` and the associated Cloudera Management Services processes are now active on that host. It usually takes a few minutes for the new services to come up on the new host.

Database High Availability Configuration

This section contains additional information you can use when configuring databases for high availability.

Database-Specific Mechanisms

- MariaDB:

  Configuring MariaDB for high availability requires configuring MariaDB for replication. For more information, see https://mariadb.com/kb/en/mariadb/setting-up-replication/.

- MySQL:

  Configuring MySQL for high availability requires configuring MySQL for replication. Replication configuration depends on which version of MySQL you are using. For version 5.1, http://dev.mysql.com/doc/refman/5.1/en/replication-howto.html provides an introduction.

  MySQL GTID-based replication is not supported.

- PostgreSQL:

  PostgreSQL has extensive documentation on high availability, especially for versions 9.0 and higher. For information about options available for version 9.1, see http://www.postgresql.org/docs/9.1/static/high-availability.html.

- Oracle:

  Oracle supports a wide variety of free and paid upgrades to their database technology that support increased availability guarantees, such as their Maximum Availability Architecture (MAA) recommendations. For more information, see http://www.oracle.com/technetwork/database/features/availability/oracle-database-maa-best-practices-155386.html.

Disk-Based Mechanisms

DRBD is an open-source Linux-based disk replication mechanism that works at the individual write level to replicate writes on multiple machines. Although not directly supported by major database vendors (at the time of writing of this document), it provides a way to inexpensively configure redundant distributed disk for disk-consistent databases (such as MySQL, PostgreSQL, and Oracle). For information, see http://drbd.linbit.com.

TLS and Kerberos Configuration for Cloudera Manager High Availability

Cloudera Manager supports TLS for encrypted network communications, and it supports integration with Kerberos for authentication (see Configuring TLS Encryption for Cloudera Manager and Configuring Authentication in Cloudera Manager for details). Configuring TLS- or Kerberos-enabled Cloudera Manager clusters for high availability requires the additional steps discussed below:

- TLS Considerations for Cloudera Manager High Availability
  - Configure Load Balancers for TLS Pass-Through
  - Server Certificate Requirements for HA Deployments
  - Cloudera Manager Agent Host Requirements for HA Deployments

- Kerberos Considerations for Cloudera Manager High Availability
  - Server Configuration Requirements for HA
  - Re-Generate Kerberos Credentials

Example hostnames used throughout Configuring Cloudera Manager for High Availability With a Load Balancer on page 449 are summarized in the table below.

Host description	Example hostnames
Load balancer for Cloudera Manager Server	CMSHostname
Cloudera Manager Server (primary)	CMS1
Cloudera Manager Server (secondary)	CMS2
Load balancer for Cloudera Management Service	MGMTHostname
Cloudera Management Service (primary)	MGMT1
Cloudera Management Service (secondary)	MGMT2

TLS Considerations for Cloudera Manager High Availability

When successfully configured for high availability and for TLS, the Cloudera Manager Admin Console is accessed using the host name or IP address of the load balancer:

```
https://[CMSHostname]:7183
```

This assumes that the load balancer has been set up for TLS pass-through and that the Cloudera Manager Server host has been set up as detailed below.

Configure Load Balancers for TLS Pass-Through

As detailed in Configuring Cloudera Manager for High Availability With a Load Balancer on page 449, high availability for Cloudera Manager Server clusters requires secondary nodes that act as backups for the primary Cloudera Manager Server and Cloudera Management Service nodes, respectively. Only the primary nodes are active at any time, but if these fail, requests are redirected by a load balancer (CMSHostname or MGMTHostname) to the appropriate secondary node.

When the Cloudera Manager Server cluster is configured for TLS in addition to high availability, the load balancers must be configured for TLS *pass-through*—traffic from clients is not decrypted until it receives the actual server host system. Keep this in mind when you are setting up the load balancer for your Cloudera Manager High Availability deployment.

Server Certificate Requirements for HA Deployments

TLS-enabled Cloudera Manager Server clusters require certificates that authenticate the host identity prior to encryption, as detailed in Configuring TLS Encryption for Cloudera Manager. When deploying Cloudera Manager Server for high availability, however, the certificate must identify the load balancer and both primary and secondary nodes (rather than the primary host alone). That means you must create your certificate signing request (CSR) as follows:

- Use the FQDN of the load balancer (for example, (*CMSHostname*) for the CN (common name).
- Use the primary and secondary Cloudera Manager Server host names (for example, *CMS1* and *CMS2*, respectively) for the SubjectAlternativeName (SAN) values.

To create a CSR using these example load balancer and server host names:

```
keytool -genkeypair -alias loadBalProxyCMS  -keyalg RSA -keystore keystoreName.jks \
-keysize 2048 -dname "CN=CMSHostname, OU=Department, O=Example, \
L=City, ST=State, C=US" -storepass password \
-keypass password

keytool -certreq -alias loadBalProxyCMS -keystore keystoreName.jks -file
sigRequestHA_LB_1.csr \
-storepass password -keypass password  -ext san=dns:CMS1,dns:CMS2
```

Alternatively, if the Cloudera Manager Server certificates on the hosts do not specify the load balancer name and SAN names, you can make the following change to the configuration. From the Cloudera Manager Admin Console, go to:

- **Administration** > **Ports and Addresses**
- Enter the FQDN of the load balancer in the **Cloudera Manager Hostname Override**

In addition to using correctly created certificates (or over-riding the hostname), you must:

- Store the keystore and truststore in the same path on both primary and secondary Cloudera Manager Server hosts (*CMS1*, *CMS2*), or point to the same shared network mount point from each host.

Cloudera Manager Agent Host Requirements for HA Deployments

Cloudera Manager Server hosts can present their certificate to agents prior to encrypting the connection (see Enable Server Certificate Verification on Cloudera Manager Agents for details). For a high availability deployment:

- Use the same setting for `verify_cert_file` (in the `/etc/cloudera-scm-agent/config.ini` file) on each agent host system. To simplify the set, share the file path to `verify_cert_file` or copy the files manually as specified in the config file between *MGMT1* and *MGMT2*.

Cloudera Manager Agent hosts can present certificates to requesting processes such as the Cloudera Manager Server prior to encryption (see Configure Agent Certificate Authentication). For a high availability deployment:

- Share the certificate and key for use by all Cloudera Manager Agent host systems on NFS, or copy them to the same path on both *MGMT1* and *MGMT2*.

> **Important:** Restart `cloudera-scm-agent` after making changes to the certificates or other files, or to the configuration.

Kerberos Considerations for Cloudera Manager High Availability

As detailed in Creating Hosts for Primary and Secondary Servers on page 454, primary and secondary nodes that comprise a Cloudera Manager High Availability cluster must be configured the same (only one host is active at any given time). That means that if the cluster uses Kerberos for authentication, the Kerberos configuration on the primary and secondary nodes must also be the same.

Server Configuration Requirements for HA

When configuring high availability for Kerberos-enabled Cloudera Manager clusters, you must:

- Install Kerberos client libraries in the same path on both primary (for example, *CMS1*) and secondary (*CMS2*) Cloudera Manager Server hosts .
- Configure the `/etc/krb5.conf` file identically across the Cloudera Manager Server and Cloudera Management Service hosts (*CMS1*, *CMS2*, *MGMT1*, *MGMT2*).
- If the Cloudera Manager Server primary host (*CMS1*) is configured to store the Cloudera Manager Server KDC access credentials in `/etc/cloudera-scm-server`, use this same path on the secondary host (*CMS2*).

Re-Generate Kerberos Credentials

Configuring the Cloudera Management Service for high availability using an existing Cloudera Manager Server cluster (as discussed in Installing the Primary on page 464) results in the Cloudera Management Service not starting, as shown here:

mgmt: Configuration Issues

🔴 **C** hostmonitor (test01-ha-common-2): Role is missing Kerberos keytab.

🔴 **C** reportsmanager (test01-ha-common-2): Role is missing Kerberos keytab.

🔴 **C** servicemonitor (test01-ha-common-2): Role is missing Kerberos keytab.

🔴 **C** activitymonitor (test01-ha-common-2): Role is missing Kerberos keytab.

🔴 **C** navigatormetaserver (test01-ha-common-2): Role is missing Kerberos keytab.

This is expected. To resolve, re-generate the Kerberos credentials for the roles:

1. Log in to the Cloudera Manager Admin Console.
2. Select **Administration** > **Kerberos** > **Credentials** > **Generate Credentials**.

Backup and Disaster Recovery

Cloudera Manager provides an integrated, easy-to-use management solution for enabling data protection on the Hadoop platform. Cloudera Manager enables you to replicate data across data centers for disaster recovery scenarios. Replications can include data stored in HDFS, data stored in Hive tables, Hive metastore data, and Impala metadata (catalog server metadata) associated with Impala tables registered in the Hive metastore. When critical data is stored on HDFS, Cloudera Manager helps to ensure that the data is available at all times, even in case of complete shutdown of a datacenter.

You can also replicate HDFS data to and from Amazon S3.

You can also use the HBase shell to replicate HBase data. (Cloudera Manager does not manage HBase replications.)

> **Important:** This feature requires a Cloudera Enterprise license. It is not available in Cloudera Express. See Managing Licenses on page 37 for more information.

You can also use Cloudera Manager to schedule, save, and restore snapshots of HDFS directories and HBase tables.

Cloudera Manager provides key functionality in the Cloudera Manager Admin Console:

- **Select** - Choose datasets that are critical for your business operations.
- **Schedule** - Create an appropriate schedule for data replication and snapshots. Trigger replication and snapshots as required for your business needs.
- **Monitor** - Track progress of your snapshots and replication jobs through a central console and easily identify issues or files that failed to be transferred.
- **Alert** - Issue alerts when a snapshot or replication job fails or is aborted so that the problem can be diagnosed quickly.

BDR functions consistently across HDFS and Hive:

- You can set it up on files or directories in HDFS and on External tables in Hive—without manual translation of Hive datasets to HDFS datasets, or vice versa. Hive Metastore information is also replicated.
- Applications that depend on External table definitions stored in Hive, operate on both replica and source as table definitions are updated.

You can also perform a "dry run" to verify configuration and understand the cost of the overall operation before actually copying the entire dataset.

Port Requirements for Backup and Disaster Recovery

Make sure that the following ports are open and accessible on the source hosts to allow communication between the source and destination Cloudera Manager servers and the HDFS, Hive, MapReduce, and YARN hosts:

Table 24:

Service	Default Port
Cloudera Manager Admin Console HTTP	7180
Cloudera Manager Admin Console HTTPS (with TLS enabled)	7183
Cloudera Manager Agent	9000
HDFS NameNode	8020
Key Management Server (KMS)	16000
HDFS DataNode	50010

Service	Default Port
WebHDFS	50070
YARN Resource Manager	8032

See Ports for more information, including how to verify the current values for these ports.

Data Replication

Cloudera Manager enables you to replicate data across data centers for disaster recovery scenarios. Replications can include data stored in HDFS, data stored in Hive tables, Hive metastore data, and Impala metadata (catalog server metadata) associated with Impala tables registered in the Hive metastore. When critical data is stored on HDFS, Cloudera Manager helps to ensure that the data is available at all times, even in case of complete shutdown of a datacenter.

You can also replicate HDFS data to and from Amazon S3 and you can replicate Hive data and metadata to and from Amazon S3.

For an overview of data replication, view this video about Backing Up Data Using Cloudera Manager.

You can also use the HBase shell to replicate HBase data. (Cloudera Manager does not manage HBase replications.)

For recommendations on using data replication and Sentry authorization, see Configuring Sentry to Enable BDR Replication.

> **Note:** BDR does not replicate Sentry RBAC rules. This means that the source and destination clusters can end up with different authorization rules for the replicated content if the clusters do not have identical Sentry RBAC rules.

View a video about Backing up Data Using Cloudera Manager .

Cloudera License Requirements for Replication

Both the *source* and *destination* clusters must have a Cloudera Enterprise license.

Supported and Unsupported Replication Scenarios

Supported Replication Scenarios

Versions

To replicate data to or from clusters managed by Cloudera Manager 6, the source or destination cluster must be managed by Cloudera Manager 5.14.0 or higher. Note that some functionality may not be available in Cloudera Manager 5.14.0+ or 6.0.0+.

Kerberos

BDR supports the following replication scenarios when Kerberos authentication is used on a cluster:

- Secure source to a secure destination.
- Insecure source to insecure destination.

Cloud Storage

BDR supports replicating to or from Amazon S3 and Microsoft Azure ADLS Gen1.

TLS

You can use TLS with BDR. Additionally, BDR supports replication scenarios where TLS is enabled for non-Hadoop services (Hive/Impala) and TLS is disabled Hadoop services (such as HDFS,YARN, and MapReduce).

Unsupported Replication Scenarios

> ✎ **Note:** If you are using Isilon storage for CDH, see Supported Replication Scenarios for Clusters using Isilon Storage.

Versions

Replicating to or from Cloudera Manager 6 managed clusters with Cloudera Manager versions earlier than 5.14.0 are not supported.

Kerberos

BDR does not support the following replication scenarios when Kerberos authentication is used on a cluster:

- Secure source to an insecure destination is not supported.
- Insecure source to a secure destination is not supported.

Replicating Directories with Thousands of Files and Subdirectories

To replicate data that includes a directory with several hundred thousand files or subdirectories:

1. On the destination Cloudera Manager instance, go to the HDFS service page.
2. Click the **Configuration** tab.
3. Select **Scope** > *HDFS service name* **(Service-Wide)** and **Category** > **Advanced**.
4. Locate the **HDFS Replication Advanced Configuration Snippet** property.
5. Increase the heap size by adding a key-value pair, for instance, `HADOOP_CLIENT_OPTS=-Xmx1g`. In this example, `1g` sets the heap size to 1 GB. This value should be adjusted depending on the number of files and directories being replicated.
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

HDFS and Hive/Impala Replication To and From Amazon S3

Minimum Required Role: User Administrator (also provided by **Full Administrator**)

To configure Amazon S3 as a source or destination for HDFS or Hive/Impala replication, you configure **AWS Credentials** that specify the type of authentication to use, the Access Key ID, and Secret Key. See How to Configure AWS Credentials.

After adding the AWS credentials, you can click the **Replication Schedules** link to define a replication schedule. See HDFS Replication on page 487 or Hive/Impala Replication on page 499 for details about creating replication schedules. You can also click **Close** and create the replication schedules later. Select the AWS Credentials account in the **Source** or **Destination** drop-down lists when creating the schedules.

Supported Replication Scenarios for Clusters using Isilon Storage

> ⚠ **Warning:** CDH 6.0 is not currently supported on Dell EMC Isilon.

Note the following when scheduling replication jobs for clusters that use Isilon storage:

- As of CDH 5.8 and higher, Replication is supported for clusters using Kerberos and Isilon storage on the source or destination cluster, or both. See Configuring Replication with Kerberos and Isilon on page 118. Replication between clusters using Isilon storage and Kerberos is not supported in CDH 5.7.
- Make sure that the `hdfs` user is a superuser in the Isilon system. If you specify alternate users with the **Run As** option when creating replication schedules, those users must also be superusers.
- Cloudera recommends that you use the Isilon `root` user for replication jobs. (Specify `root` in the **Run As** field when creating replication schedules.)
- Select the **Skip checksum checks** property when creating replication schedules.
- Clusters that use Isilon storage do not support snapshots. Snapshots are used to ensure data consistency during replications in scenarios where the source files are being modified. Therefore, when replicating from an Isilon

cluster, Cloudera recommends that you do not replicate Hive tables or HDFS files that could be modified before the replication completes.

See Using CDH with Isilon Storage on page 117.

Designating a Replication Source

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

The Cloudera Manager Server that you are logged into is the destination for replications set up using that Cloudera Manager instance. From the Admin Console of this destination Cloudera Manager instance, you can designate a peer Cloudera Manager Server as a source of HDFS and Apache Hive data for replication.

Configuring a Peer Relationship

> **Note:** If your cluster uses SAML Authentication, see Configuring Peers with SAML Authentication on page 487 before configuring a peer.

1. Go to the **Peers** page by selecting **Backup** > **Peers**. If there are no existing peers, you will see only an **Add Peer** button in addition to a short message. If peers already exist, they display in the Peers list.
2. Click the **Add Peer** button.
3. In the **Add Peer** dialog box, provide a name, the URL (including the port) of the Cloudera Manager Server source for the data to be replicated, and the login credentials for that server.

> **Important:** The role assigned to the login on the source server must be either a *User Administrator* or a *Full Administrator*.

Cloudera recommends that TLS/SSL be used. A warning is shown if the URL scheme is `http` instead of `https`. After configuring both peers to use TLS/SSL, add the remote source Cloudera Manager TLS/SSL certificate to the local Cloudera Manager truststore, and vice versa. See Configuring TLS Encryption for Cloudera Manager.

4. Click the **Add Peer** button in the dialog box to create the peer relationship.

The peer is added to the Peers list. Cloudera Manager automatically tests the connection between the Cloudera Manager Server and the peer. You can also click **Test Connectivity** to test the connection. **Test Connectivity** also tests the Kerberos configuration for the clusters. For more information about this part of the test, see Kerberos Connectivity Test on page 516.

Modifying Peers

1. Go to the **Peers** page by selecting **Backup** > **Peers**. If there are no existing peers, you will see only an **Add Peer** button in addition to a short message. If peers already exist, they display in the Peers list.
2. Do one of the following:

   - **Edit**

     1. In the row for the peer, select **Edit**.
     2. Make your changes.
     3. Click **Update Peer** to save your changes.

   - **Delete** - In the row for the peer, click **Delete**.

HDFS and Hive/Impala Replication To and From Amazon S3

Minimum Required Role: User Administrator (also provided by **Full Administrator**)

To configure Amazon S3 as a source or destination for HDFS or Hive/Impala replication, you configure **AWS Credentials** that specify the type of authentication to use, the Access Key ID, and Secret Key. See How to Configure AWS Credentials.

After adding the AWS credentials, you can click the **Replication Schedules** link to define a replication schedule. See HDFS Replication on page 487 or Hive/Impala Replication on page 499 for details about creating replication schedules. You can also click **Close** and create the replication schedules later. Select the AWS Credentials account in the **Source** or **Destination** drop-down lists when creating the schedules.

Configuring Peers with SAML Authentication

If your cluster uses SAML Authentication, do the following before creating a peer:

1. Create a Cloudera Manager user account that has the **User Administrator** or **Full Administrator** role.

   You can also use an existing user that has one of these roles. Since you will only use this user to create the peer relationship, you can delete the user account after adding the peer.

2. Create or modify the peer, as described in this topic.
3. (Optional) Delete the Cloudera Manager user account you just created.

HDFS Replication

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Minimum Required Role: BDR Administrator (also provided by **Full Administrator**)

HDFS replication enables you to copy (replicate) your HDFS data from one HDFS service to another, synchronizing the data set on the *destination* service with the data set on the *source* service, based on a specified replication schedule. You can also replicate HDFS data to and from Amazon S3. The destination service must be managed by the Cloudera Manager Server where the replication is being set up, and the source service can be managed by that same server or by a peer Cloudera Manager Server. You can also replicate HDFS data within a cluster by specifying different source and destination directories.

Remote BDR Replication automatically copies HDFS metadata to the destination cluster as it copies files. HDFS metadata need only be backed up locally. For information about how to backup HDFS metadata locally, see Backing Up and Restoring NameNode Metadata on page 97.

> **Important:** To use HDFS replication, both the destination and source HDFS services must use Kerberos authentication, or both must not use Kerberos authentication. See Enabling Replication Between Clusters with Kerberos Authentication on page 514.

Source Data

While a replication runs, ensure that the source directory is not modified. A file added during replication does not get replicated. If you delete a file during replication, the replication fails.

Additionally, ensure that all files in the directory are closed. Replication fails if source files are open. If you cannot ensure that all source files are closed, you can configure the replication to continue despite errors. Uncheck the **Abort on Error** option for the HDFS replication. For more information, see Configuring Replication of HDFS Data on page 489

After the replication completes, you can view the log for the replication to identify opened files. Ensure these files are closed before the next replication occurs.

Network Latency and Replication

High latency among clusters can cause replication jobs to run more slowly, but does not cause them to fail. For best performance, latency between the source cluster NameNode and the destination cluster NameNode should be less than 80 milliseconds. (You can test latency using the Linux `ping` command.) Cloudera has successfully tested replications with latency of up to 360 milliseconds. As latency increases, replication performance degrades.

Performance and Scalability Limitations

HDFS replication has the following limitations:

- Maximum number of files for a single replication job: 100 million.
- Maximum number of files for a replication schedule that runs more frequently than once in 8 hours: 10 million.
- The throughput of the replication job depends on the absolute read and write throughput of the source and destination clusters.
- Regular rebalancing of your HDFS clusters is required for efficient operation of replications. See HDFS Balancers on page 105.

> **Note:** Cloudera Manager provides downloadable data that you can use to diagnose HDFS replication performance. See Monitoring the Performance of HDFS Replications on page 496.

Replication with Sentry Enabled

If the cluster has Sentry enabled and you are using BDR to replicate files or tables and their permissions, configuration changes to HDFS are required.

The configuration changes are required due to how HDFS manages ACLs. When a user reads ACLs, HDFS provides the ACLs configured in the External Authorization Provider, which is Sentry. If Sentry is not available or it does not manage authorization of the particular resource, such as the file or directory, then HDFS falls back to its own internal ACLs. But when ACLs are written to HDFS, HDFS always writes these internal ACLs even when Sentry is configured. This causes HDFS metadata to be polluted with Sentry ACLs. It can also cause a replication failure in replication when Sentry ACLs are not compatible with HDFS ACLs.

To prevent issues with HDFS and Sentry ACLs, complete the following steps:

1. Create a user account that is only used for BDR jobs since Sentry ACLs will be bypassed for this user.

   For example, create a user named `bdr-only-user`.

2. Configure HDFS on the source cluster:

   a. In the Cloudera Manager Admin Console, select **Clusters** > **<HDFS service>**.
   b. Select **Configuration** and search for the following property: `NameNode Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml`.
   c. Add the following property:

      Name: Use the following property name: `dfs.namenode.inode.attributes.provider.bypass.users`

      Value: Provide the following information: <username>, <username>@<RealmName>

      Replace *<username>* with the user you created in step 1 and *<RealmName>* with the name of the Kerberos realm.

      For example, the user `bdr-only-user` on the realm `elephant` requires the following value:

   ```
   bdr-only-user, bdr-only-user@ElephantRealm
   ```

      Description: This field is optional.

   d. Restart the NameNode.

3. Repeat step 2 on the destination cluster.
4. When you create a replication schedule, specify the user you created in step 1 in the **Run As Username** and **Run on Peer as Username (if available)** fields.

   > **Note:** The **Run As Username** field is used to launch MapReduce job for copying data. **Run on Peer as Username** field is used to run copy listing on source, if different than **Run as Username**.

> **Important:** Make sure to set the value of **Run on Peer as Username** same as **Run as Username**, else BDR reads ACL from the source as `hdfs`, which pulls the Sentry provided ACLs over to the target cluster and applies them to the files in HDFS. It can result in additional usage of NameNode heap in the target cluster.

Configuring Replication of HDFS Data

1. Verify that your cluster conforms to one of the [Supported Replication Scenarios](#).

2. If you are using different Kerberos principals for the source and destination clusters, add the *destination* principal as a proxy user on the *source* cluster. For example, if you are using the `hdfssrc` principal on the source cluster and the `hdfsdest` principal on the destination cluster, add the following properties to the HDFS service **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** property on the *source* cluster:

```
<property>
    <name>hadoop.proxyuser.hdfsdest.groups</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.hdfsdest.hosts</name>
    <value>*</value>
</property>
```

   Deploy the client configuration and restart all services on the *source* cluster.

3. If the source cluster is managed by a different Cloudera Manager server than the destination cluster, [configure a peer relationship](#). If the source or destination is Amazon S3, you must [configure AWS credentials](#).

4. Do one of the following:

   1. Select **Backup** > **Replication Schedules**
   2. Click **Create Schedule** > **HDFS Replication**.

   or

   1. Select **Clusters** > *HDFS Service Name*.
   2. Select **Quick Links** > **Replication**.
   3. Click **Create Schedule** > **HDFS Replication**.

   The **Create HDFS Replication** dialog box displays, and opens displaying the **General** tab. Click the **Peer** or **AWS Credentials** link if your replication job requires them and you need to create these entities.

5. Select the **General** tab to configure the following:

   a. Click the **Name** field and add a unique name for the replication schedule.

   b. Click the **Source** field and select the source HDFS service. You can select HDFS services managed by a peer Cloudera Manager Server, local HDFS services (managed by the Cloudera Manager Server for the Admin Console you are logged into), or you can select AWS Credentials.

   c. Enter the **Source Path** to the directory (or file) you want to replicate. For replication to Amazon S3, enter the path using the following form:

```
s3a://bucket name/path
```

   d. Click the **Destination** field and select the destination HDFS service from the HDFS services managed by the Cloudera Manager Server for the Admin Console you are logged into, or select AWS Credentials.

   e. Enter the **Destination Path** where the source files should be saved. For replication to Amazon S3, enter the path using the following form:

```
s3a://bucket name/path
```

   f. Select a **Schedule**:

- **Immediate** - Run the schedule Immediately.
- **Once** - Run the schedule one time in the future. Set the date and time.
- **Recurring** - Run the schedule periodically in the future. Set the date, time, and interval between runs.

g. Enter the user to run the replication job in the **Run As Username** field. By default this is `hdfs`. If you want to run the job as a different user, enter the user name here. If you are using Kerberos, you *must* provide a user name here, and it must be one with an ID greater than 1000. (You can also configure the minimum user ID number with the **min.user.id** property in the YARN or MapReduce service.) Verify that the user running the job has a home directory, `/user/username`, owned by `username:supergroup` in HDFS. This user must have permissions to read from the source directory and write to the destination directory.

Note the following:

- The User must not be present in the list of banned users specified with the **Banned System Users** property in the YARN configuration (Go to the YARN service, select **Configuration** tab and search for the property). For security purposes, the `hdfs` user is banned by default from running YARN containers.
- The requirement for a user ID that is greater than 1000 can be overridden by adding the user to the "white list" of users that is specified with the **Allowed System Users** property. (Go to the YARN service, select **Configuration** tab and search for the property.)

6. Select the **Resources** tab to configure the following:

- **Scheduler Pool** – (Optional) Enter the name of a resource pool in the field. The value you enter is used by the **MapReduce Service** you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties:
  - MapReduce – Fair scheduler: `mapred.fairscheduler.pool`
  - MapReduce – Capacity scheduler: `queue.name`
  - YARN – `mapreduce.job.queuename`

- **Maximum Map Slots** - Limits for the number of map slots per mapper. The default value is 20.
- **Maximum Bandwidth** - Limits for the bandwidth per mapper. The default is 100 MB.
- **Replication Strategy** - Whether file replication tasks should be distributed among the mappers statically or dynamically. (The default is **Dynamic**.) Static replication distributes file replication tasks among the mappers up front to achieve a uniform distribution based on the file sizes. Dynamic replication distributes file replication tasks in small sets to the mappers, and as each mapper completes its tasks, it dynamically acquires and processes the next unallocated set of tasks.

7. Select the **Advanced Options** tab, to configure the following:

- **Add Exclusion** click the link to exclude one or more paths from the replication.

  The **Regular Expression-Based Path Exclusion** field displays, where you can enter a regular expression-based path. When you add an exclusion, include the snapshotted relative path for the regex. For example, to exclude the `/user/bdr` directory, use the following regular expression, which includes the snapshots for the `bdr` directory:

```
.*/user/\.snapshot/.+/bdr.*
```

  You can add more than one regular expression to exclude.

- **MapReduce Service** - The MapReduce or YARN service to use.
- **Log path** - An alternate path for the logs.
- **Description** - A description of the replication schedule.
- **Error Handling** You can select the following:
  - **Skip Checksum Checks** - Whether to skip checksum checks on the copied files. If checked, checksums are not validated. Checksums are checked by default.

> **Important:**
>
> You must skip checksum checks to prevent replication failure due to non-matching checksums in the following cases:
>
> - Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster.
> - Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster.
> - Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster.
>
> Checksums are used for two purposes:
>
> - To skip replication of files that have already been copied. If **Skip Checksum Checks** is selected, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination.
> - To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data.

- **Skip Listing Checksum Checks** - Whether to skip checksum check when comparing two files to determine whether they are same or not. If skipped, the file size and last modified time are used to determine if files are the same or not. Skipping the check improves performance during the mapper phase. Note that if you select the **Skip Checksum Checks** option, this check is also skipped.
- **Abort on Error** - Whether to abort the job on an error. If selected, files copied up to that point remain on the destination, but no additional files are copied. **Abort on Error** is off by default.

- **Preserve** - Whether to preserve the block size, replication count, permissions (including ACLs), and extended attributes (XAttrs) as they exist on the source file system, or to use the settings as configured on the destination file system. By default source system settings are preserved. When **Permission** is checked, and both the source and destination clusters support ACLs, replication preserves ACLs. Otherwise, ACLs are not replicated. When **Extended attributes** is checked, and both the source and destination clusters support extended attributes, replication preserves them. (This option only displays when both source and destination clusters support extended attributes.)

If you select one or more of the **Preserve** options and you are replicating *to* Amazon S3, the values all of these items are saved in meta data files on S3. When you replicate *from* Amazon S3 to HDFS, you can select which of these options you want to preserve.

> **Note:** To preserve permissions to HDFS, you must be running as a superuser on the *destination* cluster. Use the "Run As Username" option to ensure that is the case.

See [Replication of Encrypted Data](#) on page 517 and [HDFS Transparent Encryption](#).

- **Delete Policy** - Whether files that were deleted on the source should also be deleted from the destination directory. This policy also determines the handling of files in the destination location that are unrelated to the source. Options include:

  - **Keep Deleted Files** - Retains the destination files even when they no longer exist at the source. (This is the default.).
  - **Delete to Trash** - If the HDFS trash is enabled, files are moved to the trash folder. (Not supported when replicating to Amazon S3.)
  - **Delete Permanently** - Uses the least amount of space; use with caution.

- **Alerts** - Whether to generate alerts for various state changes in the replication workflow. You can alert on failure, on start, on success, or when the replication workflow is aborted.

8. Click **Save Schedule**.

   The replication task now appears as a row in the **Replications Schedule** table. (It can take up to 15 seconds for the task to appear.)

   If you selected **Immediate** in the **Schedule** field, the replication job begins running when you click **Save Schedule**.

To specify additional replication tasks, select **Create** > **HDFS Replication**.

> **Note:** If your replication job takes a long time to complete, and files change before the replication finishes, the replication may fail. Consider making the directories snapshottable, so that the replication job creates snapshots of the directories before copying the files and then copies files from these snapshottable directories when executing the replication. See Using Snapshots with Replication on page 513.

Limiting Replication Hosts

You can limit which hosts can run replication processes by specifying a whitelist of hosts. For example, you may not want a host with the Gateway role to run a replication job since the process is resource intensive.

To configure the hosts used for HDFS replication:

1. Click **Clusters** > **HDFS** > **Configuration**.
2. Type `HDFS Replication` in the search box.
3. Locate the **HDFS Replication Environment Advanced Configuration Snippet (Safety Valve)** property.
4. Add the `HOST_WHITELIST` property. Enter a comma-separated list of DataNode hostnames to use for HDFS replication. For example:

```
HOST_WHITELIST=host-1.mycompany.com,host-2.mycompany.com
```

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Viewing Replication Schedules

The **Replications Schedules** page displays a row of information about each scheduled replication job. Each row also displays recent messages regarding the last time the Replication job ran.

Figure 10: Replication Schedules Table

Only one job corresponding to a replication schedule can occur at a time; if another job associated with that same replication schedule starts before the previous one has finished, the second one is canceled.

You can limit the replication jobs that are displayed by selecting filters on the left. If you do not see an expected schedule, adjust or clear the filters. Use the search box to search the list of schedules for path, database, or table names.

The **Replication Schedules** columns are described in the following table.

Table 25: Replication Schedules Table

Column	Description
ID	An internally generated ID number that identifies the schedule. Provides a convenient way to identify a schedule. Click the **ID** column label to sort the replication schedule table by ID.
Name	The unique name you specify when you create a schedule.
Type	The type of replication scheduled, either HDFS or Hive.
Source	The source cluster for the replication.
Destination	The destination cluster for the replication.
Throughput	Average throughput per mapper/file of all the files written. Note that throughput does not include the following information: the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.
Progress	The progress of the replication.

Last Run	The date and time when the replication last ran. Displays **None** if the scheduled replication has not yet been run. Click the date and time link to view the Replication History page for the replication.
	
	Displays one of the following icons:
	
	• ✔ - Successful. Displays the date and time of the last run replication.
	• ⊗ - Failed. Displays the date and time of a failed replication.
	• ● - None. This scheduled replication has not yet run.
	•
	
	    60%
	
	    - Running. Displays a spinner and bar showing the progress of the replication.
	
	Click the **Last Run** column label to sort the Replication Schedules table by the last run date.

Next Run	The date and time when the next replication is scheduled, based on the schedule parameters specified for the schedule. Hover over the date to view additional details about the scheduled replication. Click the **Next Run** column label to sort the Replication Schedules table by the next run date.
Objects	Displays on the bottom line of each row, depending on the type of replication: • **Hive** - A list of tables selected for replication. • **HDFS** - A list of paths selected for replication. For example:

Column	Description

Actions	The following items are available from the **Action** button:
	
	• **Show History** - Opens the **Replication History** page for a replication. See [Viewing Replication History](#) on page 494.
	• **Edit Configuration** - Opens the **Edit Replication Schedule** page.
	• **Dry Run** - Simulates a run of the replication task but does not actually copy any files or tables. After a Dry Run, you can select **Show History**, which opens the **Replication History** page where you can view any error messages and the number and size of files or tables that would be copied in an actual replication.
	• Click **Collect Diagnostic Data** to open the **Send Diagnostic Data** screen, which allows you to collect replication-specific diagnostic data for the last 10 runs of the schedule:
	
	    1. Select **Send Diagnostic Data to Cloudera** to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle.
	    2. Click **Collect and Send Diagnostic Data** to generate the bundle and open the **Replications Diagnostics Command** screen.
	    3. When the command finishes, click **Download Result Data** to download a zip file containing the bundle.
	
	• **Run Now** - Runs the replication task immediately.
	• **Disable | Enable** - Disables or enables the replication schedule. No further replications are scheduled for disabled replication schedules.
	• **Delete** - Deletes the schedule. Deleting a replication schedule does not delete copied files or tables.

• While a job is in progress, the **Last Run** column displays a spinner and progress bar, and each stage of the replication task is indicated in the message beneath the job's row. Click the **Command Details** link to view details about the execution of the command.
• If the job is successful, the number of files copied is indicated. If there have been no changes to a file at the source since the previous job, then that file is *not* copied. As a result, after the initial job, only a subset of the files may actually be copied, and this is indicated in the success message.
• If the job fails, the ⊗ icon displays.
• To view more information about a completed job, select **Actions** > **Show History**. See [Viewing Replication History](#) on page 494.

Enabling, Disabling, or Deleting A Replication Schedule

When you create a new replication schedule, it is automatically enabled. If you disable a replication schedule, it can be re-enabled at a later time.

To enable, disable, or delete a replication schedule:

• Click **Actions** > **Enable\|Disable\|Delete** in the row for a replication schedule.

To enable, disable, or delete multiple replication schedules:

1. Select one or more replication schedules in the table by clicking the check box the in the left column of the table.
2. Click **Actions for Selected** > **Enable\|Disable\|Delete**.

Viewing Replication History

You can view historical details about replication jobs on the **Replication History** page.

To view the history of a replication job:

1. Select **Backup** > **Replication Schedules** to go to the **Replication Schedules** page.
2. Locate the row for the job.
3. Click **Actions** > **Show History**.

Figure 11: Replication History Screen (HDFS)

Figure 12: Replication History Screen (Hive, Failed Replication)

The **Replication History** page displays a table of previously run replication jobs with the following columns:

Table 26: Replication History Table

Column	Description

Start Time	Time when the replication job started.
	
	Expand the display and show details of the replication. In this screen, you can:
	
	• Click the **View** link to open the **Command Details** page, which displays details and messages about each step in the execution of the command. Expand the display for a **Step** to:
	
	  – View the actual command string.
	  – View the Start time and duration of the command.
	  – Click the **Context** link to view the service status page relevant to the command.
	  – Select one of the tabs to view the **Role Log**, **stdout**, and **stderr** for the command.
	
	  See Viewing Running and Recent Commands on page 241.
	
	• Click **Collect Diagnostic Data** to open the **Send Diagnostic Data** screen, which allows you to collect replication-specific diagnostic data for this run of the schedule:
	
	  1. Select **Send Diagnostic Data to Cloudera** to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle.
	  2. Click **Collect and Send Diagnostic Data** to generate the bundle and open the **Replications Diagnostics Command** screen.
	  3. When the command finishes, click **Download Result Data** to download a zip file containing the bundle.
	
	• (HDFS only) Link to view details on the **MapReduce Job** used for the replication. See Viewing and Filtering MapReduce Activities on page 249.
	• (Dry Run only) View the number of **Replicable Files**. Displays the number of files that would be replicated during an actual replication.

Column	Description
	• (Dry Run only) View the number of **Replicable Bytes**. Displays the number of bytes that would be replicated during an actual replication. • Link to download a CSV file containing a **Replication Report**. This file lists the databases and tables that were replicated. • View the number of **Errors** that occurred during the replication. • View the number of **Impala UDFs** replicated. (Displays only for Hive/Impala replications where **Replicate Impala Metadata** is selected.) • Click the link to download a CSV file containing a **Download Listing**. This file lists the files and directories that were replicated. • Click the link to download a CSV file containing **Download Status**. • If a user was specified in the **Run As Username** field when creating the replication job, the selected user displays. • View messages returned from the replication job.
Duration	Amount of time the replication job took to complete.
Outcome	Indicates success or failure of the replication job.
Files Expected	Number of files expected to be copied, based on the parameters of the replication schedule.
Files Copied	Number of files actually copied during the replication.
Tables	(Hive only) Number of tables replicated.
Files Failed	Number of files that failed to be copied during the replication.
Files Deleted	Number of files that were deleted during the replication.
Files Skipped	Number of files skipped during the replication. The replication process skips files that already exist in the destination and have not changed.

HDFS Replication To and From Amazon S3

You can use Cloudera Manager to replicate HDFS data to and from Amazon S3, however you cannot replicate data from one Amazon S3 instance to another using Cloudera Manager. You must have the appropriate credentials to access the Amazon S3 account and you must create buckets in Amazon S3 to store the replicated files.

When you replicate data to cloud storage with BDR, BDR also backs up file metadata, including extended attributes and ACLs.

To configure HDFS replication to Amazon S3:

1. Create **AWS Credentials**. See How to Configure AWS Credentials
2. Create an **HDFS Replication Schedule**. See HDFS Replication on page 487.

Monitoring the Performance of HDFS Replications

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

You can monitor the progress of an HDFS replication schedule using performance data that you download as a CSV file from the Cloudera Manager Admin console. This file contains information about the files being replicated, the average throughput, and other details that can help diagnose performance issues during HDFS replications. You can view this performance data for running HDFS replication jobs and for completed jobs.

To view the performance data for a *running* HDFS replication schedule:

1. Go to **Backup** > **Replication Schedules**.

2. Locate the schedule.

3. Click **Performance Report** and select one of the following options:

- HDFS Performance Summary – Download a summary report of the performance of the running replication job. An HDFS Performance Summary Report includes the last performance sample for each mapper that is working on the replication job.
- HDFS Performance Full – Download a full report of the performance of the running replication job. An HDFS Performance Full report includes all samples taken for all mappers during the full execution of the replication job.

4. To view the data, import the file into a spreadsheet program such as Microsoft Excel.

To view the performance data for a *completed* HDFS replication schedule:

1. Go to **Backup** > **Replication Schedules**.

2. Locate the schedule and click **Actions** > **Show History**.

The **Replication History** page for the replication schedule displays.

3. Click ➤ to expand the display for this schedule.

4. Click **Download CSV** link and select one of the following options:

- **Listing** – a list of files and directories copied during the replication job.
- **Status** - full status report of files where the status of the replication is one of the following:

    - **ERROR** – An error occurred and the file was not copied.
    - **DELETED** – A deleted file.
    - **SKIPPED** – A file where the replication was skipped because it was up-to-date.

- **Error Status Only** – full status report, filtered to show files with errors only.
- **Deleted Status Only** – full status report, filtered to show deleted files only.
- **Skipped Status Only**– full status report, filtered to show skipped files only.
- **Performance** – summary performance report.
- **Full Performance** – full performance report.

See <u>Table 27: HDFS Performance Report Columns</u> on page 498 for a description of the data in the performance reports.

5. To view the data, import the file into a spreadsheet program such as Microsoft Excel.

The performance data is collected every two minutes. Therefore, no data is available during the initial execution of a replication job because not enough samples are available to estimate throughput and other reported data.

The data returned by the CSV files downloaded from the Cloudera Manager Admin console has the following columns:

Table 27: HDFS Performance Report Columns

Performance Data Columns	Description
Timestamp	Time when the performance data was collected
Host	Name of the host where the YARN or MapReduce job was running.
SrcFile	Name of the source file being copied by the MapReduce job.
TgtFile	Name of the file to which the source file was being copied on the target.
BytesCopiedPerFile	Number of bytes copied for the file currently being copied.
TimeElapsedPerFile	Total time elapsed for this copy operation of the file currently being copied.
CurrThroughput	Current throughput in bytes per second.
AvgFileThroughput	Average throughput in bytes per second since the start of the file currently being copied.
TotalSleepTime	Number of seconds the transfer was stalled due to throughput throttling. This is expected to be zero unless the throughput was throttled using the **Maximum Bandwidth** parameter for the replication schedule. (You configure his parameter on the **Advanced** tab when creating or editing a replication schedule.)
AvgMapperThroughput	Average throughput for current mapper. This can include samples of throughput taken for various files copied by this mapper.
BytesCopiedPerMapper	Total bytes copied by this MapReduce job. This can include multiple files.
TimeElapsedPerMapper	Total time elapsed since this MapReduce job started copying files.

A sample CSV file, as presented in Excel, is shown here:

Note the following limitations and known issues:

- If you click the CSV download too soon after the replication job starts, Cloudera Manager returns an empty file or a CSV file that has columns headers only and a message to try later when performance data has actually been collected.

Timestamp	Host	SrcFile	TgtFile	BytesCopiedPerFile	TimeElapsedPerFile	CurrThroughput	AvgFileThroughput	TotalSleepTime	AvgMapperThroughput	BytesCopiedPerMapper	TimeElapsedPerMapper
No performance statistics path available yet: please try again later.											

- If you employ a proxy user with the form `user@domain`, performance data is not available through the links.
- If the replication job only replicates small files that can be transferred in less than a few minutes, no performance statistics are collected.
- For replication schedules that specify the **Dynamic** Replication Strategy, statistics regarding the last file transferred by a MapReduce job hide previous transfers performed by that MapReduce job.
- Only the last trace per MapReduce job is reported in the CSV file.

Hive/Impala Replication

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

Minimum Required Role: BDR Administrator (also provided by **Full Administrator**)

Hive/Impala replication enables you to copy (replicate) your Hive metastore and data from one cluster to another and synchronize the Hive metastore and data set on the *destination* cluster with the source, based on a specified replication schedule. The destination cluster must be managed by the Cloudera Manager Server where the replication is being set up, and the *source* cluster can be managed by that same server or by a peer Cloudera Manager Server.

Configuration notes:

- If the `hadoop.proxyuser.hive.groups` configuration has been changed to restrict access to the Hive Metastore Server to certain users or groups, the `hdfs` group or a group containing the `hdfs` user must also be included in the list of groups specified for Hive/Impala replication to work. This configuration can be specified either on the Hive service as an override, or in the core-site HDFS configuration. This applies to configuration settings on both the source and destination clusters.
- If you configured Synchronizing HDFS ACLs and Sentry Permissions on the target cluster for the directory where HDFS data is copied during Hive/Impala replication, the permissions that were copied during replication, are overwritten by the HDFS ACL synchronization and are not preserved
- If you are using Kerberos to secure your clusters, see Enabling Replication Between Clusters with Kerberos Authentication on page 514 for details about configuring it.

Network Latency and Replication

High latency among clusters can cause replication jobs to run more slowly, but does not cause them to fail. For best performance, latency between the source cluster NameNode and the destination cluster NameNode should be less than 80 milliseconds. (You can test latency using the Linux `ping` command.) Cloudera has successfully tested replications with latency of up to 360 milliseconds. As latency increases, replication performance degrades.

Host Selection for Hive/Impala Replication

If your cluster has Hive non-Gateway roles installed on hosts with limited resources, Hive/Impala replication may use these hosts to run commands for the replication, which can cause the performance of the replication to degrade. To improve performance, you can specify the hosts (a "white list") to use during replication so that the lower-resource hosts are not used.

To configure the hosts used for Hive/Impala Replication:

1. Click **Clusters** > **Hive** > **Configuration**.
2. Type `Hive Replication` in the search box.
3. Locate the **Hive Replication Environment Advanced Configuration Snippet (Safety Valve)** property.

4. Add the `HOST_WHITELIST` property. Enter a comma-separated list of hostnames to use for Hive/Impala replication. For example:

```
HOST_WHITELIST=host-1.mycompany.com,host-2.mycompany.com
```

5. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Hive Tables and DDL Commands

The following applies when using the `drop table` and `truncate table` DDL commands:

- If you configure replication of a Hive table and then later drop that table, the table remains on the destination cluster. The table is not dropped when subsequent replications occur.
- If you drop a table on the destination cluster, and the table is still included in the replication job, the table is re-created on the destination during the replication.
- If you drop a table partition or index on the source cluster, the replication job also drops them on the destination cluster.
- If you truncate a table, and the **Delete Policy** for the replication job is set to **Delete to Trash** or **Delete Permanently**, the corresponding data files are deleted on the destination during a replication.

Replication of Parameters

Parameters of databases, tables, partitions, and indexes are replicated by default during Hive/Impala replications.

You can disable replication of parameters:

1. Log in to the Cloudera Manager Admin Console.
2. Go to the Hive service.
3. Click the **Configuration** tab.
4. Search for "Hive Replication Environment Advanced Configuration Snippet"
5. Add the following parameter:

```
REPLICATE_PARAMETERS=false
```

6. Click **Save Changes**.

Performance and Scalability Limitations

Hive/Impala replication has the following limitations:

- Maximum number of databases: 100
- Maximum number of tables per database: 1,000
- Maximum number of partitions per table: 10,000. See Identify Workload Characteristics That Increase Memory Pressure.
- Maximum total number of tables (across all databases): 10,000
- Maximum total number of partitions (across all tables): 100,000
- Maximum number of indexes per table: 100

Configuring Replication of Hive/Impala Data

1. Verify that your cluster conforms to one of the Supported Replication Scenarios.
2. If the source cluster is managed by a different Cloudera Manager server than the destination cluster, configure a peer relationship. If the source or destination is Amazon S3, you must configure AWS credentials.
3. Do one of the following:

   - From the **Backup** tab, select **Replications**.
   - From the **Clusters** tab, go to the Hive service and select **Quick Links** > **Replication**.

   The Schedules tab of the Replications page displays.

4. Select **Create New Schedule** > **Hive Replication**. The **General** tab displays.

5. Select the **General** tab to configure the following:

> **Note:** If you are replicating to or from Amazon S3, follow the steps under <u>Hive/Impala Replication To and From Amazon S3</u> on page 508 before completing these steps.

a. Use the **Name** field to provide a unique name for the replication schedule.

b. Use the **Source** drop-down list to select the cluster with the Hive service you want to replicate.

c. Use the **Destination** drop-down list to select the destination for the replication. If there is only one Hive service managed by Cloudera Manager available as a destination, this is specified as the destination. If more than one Hive service is managed by this Cloudera Manager, select from among them.

d. Leave **Replicate All** checked to replicate all the Hive databases from the source. To replicate only selected databases, uncheck this option and enter the database name(s) and tables you want to replicate.

- You can specify multiple databases and tables using the plus symbol to add more rows to the specification.
- You can specify multiple databases on a single line by separating their names with the pipe (|) character. For example: `mydbname1|mydbname2|mydbname3`.
- Regular expressions can be used in either database or table fields, as described in the following table:

Regular Expression	Result
`[\w].+`	Any database or table name.
`(?!myname\b).+`	Any database or table except the one named myname.
`db1|db2` `[\w_]+`	All tables of the db1 and db2 databases.

`db1`
`[\w_]+`
 	
Click the "+" button and then enter
 	
`db2`
`[\w_]+`
	All tables of the db1 and db2 databases (alternate method).

e. Select a **Schedule**:

- **Immediate** - Run the schedule Immediately.
- **Once** - Run the schedule one time in the future. Set the date and time.
- **Recurring** - Run the schedule periodically in the future. Set the date, time, and interval between runs.

f. To specify the user that should run the MapReduce job, use the **Run As Username** option. By default, MapReduce jobs run as `hdfs`. To run the MapReduce job as a different user, enter the user name. If you are using Kerberos, you *must* provide a user name here, and it must have an ID greater than 1000.

> **Note:** The user running the MapReduce job should have `read` and `execute` permissions on the Hive warehouse directory on the *source* cluster. If you configure the replication job to preserve permissions, superuser privileges are required on the *destination* cluster.

6. Select the **Resources** tab to configure the following:

- **Scheduler Pool** – (Optional) Enter the name of a resource pool in the field. The value you enter is used by the **MapReduce Service** you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties:

  - MapReduce – Fair scheduler: `mapred.fairscheduler.pool`
  - MapReduce – Capacity scheduler: `queue.name`
  - YARN – `mapreduce.job.queuename`

- **Maximum Map Slots** and **Maximum Bandwidth** – Limits for the number of map slots and for bandwidth per mapper. The default is 100 MB.
- **Replication Strategy** – Whether file replication should be static (the default) or dynamic. Static replication distributes file replication tasks among the mappers up front to achieve a uniform distribution based on file sizes. Dynamic replication distributes file replication tasks in small sets to the mappers, and as each mapper processes its tasks, it dynamically acquires and processes the next unallocated set of tasks.

7. Select the **Advanced** tab to specify an export location, modify the parameters of the MapReduce job that will perform the replication, and set other options. You can select a MapReduce service (if there is more than one in your cluster) and change the following parameters:

   - Uncheck the **Replicate HDFS Files** checkbox to skip replicating the associated data files.
   - If both the source and destination clusters use CDH 5.7.0 or later up to and including 5.11.x, select the **Replicate Impala Metadata** drop-down list and select **No** to avoid redundant replication of Impala metadata. (This option only displays when supported by both source and destination clusters.) You can select the following options for **Replicate Impala Metadata**:

     - **Yes** – replicates the Impala metadata.
     - **No** – does not replicate the Impala metadata.
     - **Auto** – Cloudera Manager determines whether or not to replicate the Impala metadata based on the CDH version.

     To replicate Impala UDFs when the version of CDH managed by Cloudera Manager is 5.7 or lower, see Replicating Data to Impala Clusters on page 512 for information on when to select this option.

   - The **Force Overwrite** option, if checked, forces overwriting data in the destination metastore if incompatible changes are detected. For example, if the destination metastore was modified, and a new partition was added to a table, this option forces deletion of that partition, overwriting the table with the version found on the source.

     > **Important:** If the **Force Overwrite** option is not set, and the Hive/Impala replication process detects incompatible changes on the source cluster, Hive/Impala replication fails. This sometimes occurs with recurring replications, where the metadata associated with an existing database or table on the source cluster changes over time.

   - By default, Hive metadata is exported to a default HDFS location (`/user/${user.name}/.cm/hive`) and then imported from this HDFS file to the destination Hive metastore. In this example, `user.name` is the process user of the HDFS service on the *destination* cluster. To override the default HDFS location for this export file, specify a path in the **Export Path** field.

     > **Note:** In a Kerberized cluster, the HDFS principal on the *source* cluster must have `read`, `write`, and `execute` access to the **Export Path** directory on the *destination* cluster.

   - By default, Hive HDFS data files (for example, `/user/hive/warehouse/db1/t1`) are replicated to a location relative to "/" (in this example, to `/user/hive/warehouse/db1/t1`). To override the default, enter a path in the **HDFS Destination Path** field. For example, if you enter `/ReplicatedData`, the data files would be replicated to `/ReplicatedData/user/hive/warehouse/db1/t1`.
   - Select the **MapReduce Service** to use for this replication (if there is more than one in your cluster).
   - **Log Path** - An alternative path for the logs.
   - **Description** - A description for the replication schedule.
   - **Skip Checksum Checks** - Whether to skip checksum checks, which are performed by default.

     Checksums are used for two purposes:

     - To skip replication of files that have already been copied. If **Skip Checksum Checks** is selected, the replication job skips copying a file if the file lengths and modification times are identical between the source and destination clusters. Otherwise, the job copies the file from the source to the destination.

- To redundantly verify the integrity of data. However, checksums are not required to guarantee accurate transfers between clusters. HDFS data transfers are protected by checksums during transfer and storage hardware also uses checksums to ensure that data is accurately stored. These two mechanisms work together to validate the integrity of the copied data.

- **Skip Listing Checksum Checks** - Whether to skip checksum check when comparing two files to determine whether they are same or not. If skipped, the file size and last modified time are used to determine if files are the same or not. Skipping the check improves performance during the mapper phase. Note that if you select the **Skip Checksum Checks** option, this check is also skipped.
- **Abort on Error** - Whether to abort the job on an error. By selecting the check box, files copied up to that point remain on the destination, but no additional files will be copied. Abort on Error is off by default.
- **Delete Policy** - Whether files that were on the source should also be deleted from the destination directory. Options include:

  - **Keep Deleted Files** - Retains the destination files even when they no longer exist at the source. (This is the default.).
  - **Delete to Trash** - If the HDFS trash is enabled, files are moved to the trash folder. (Not supported when replicating to Amazon S3.)
  - **Delete Permanently** - Uses the least amount of space; use with caution.

- **Preserve** - Whether to preserve the **Block Size**, **Replication Count**, and **Permissions** as they exist on the source file system, or to use the settings as configured on the destination file system. By default, settings are preserved on the source.

  > **Note:** You must be running as a superuser to preserve permissions. Use the "Run As Username" option to ensure that is the case.

- **Alerts** - Whether to generate alerts for various state changes in the replication workflow. You can alert **On Failure**, **On Start**, **On Success**, or **On Abort** (when the replication workflow is aborted).

8. Click **Save Schedule**.

   The replication task appears as a row in the **Replications Schedule** table. See Viewing Replication Schedules on page 504.

To specify additional replication tasks, select **Create** > **Hive Replication**.

> **Note:** If your replication job takes a long time to complete, and tables change before the replication finishes, the replication may fail. Consider making the **Hive Warehouse Directory** and the directories of any external tables snapshottable, so that the replication job creates snapshots of the directories before copying the files. See Using Snapshots with Replication on page 513.

Replication of Impala and Hive User Defined Functions (UDFs)

By default, for clusters where the version of CDH is 5.7 or higher, Impala and Hive UDFs are persisted in the Hive Metastore and are replicated automatically as part of Hive/Impala replications. See User-Defined Functions (UDFs), Replicating Data to Impala Clusters on page 512, and Managing Apache Hive User-Defined Functions.

To replicate Impala UDFs when the version of CDH managed by Cloudera Manager is 5.6 or lower, see Replicating Data to Impala Clusters on page 512 for information on when to select the **Replicate Impala Metadata** option on the **Advanced** tab when creating a Hive/Impala replication schedule.

After a replication has run, you can see the number of Impala and Hive UDFs that were replicated during the last run of the schedule on the **Replication Schedules** page:

Replication Schedules

For previously-run replications, the number of replicated UDFs displays on the **Replication History** page:

Viewing Replication Schedules

The **Replications Schedules** page displays a row of information about each scheduled replication job. Each row also displays recent messages regarding the last time the Replication job ran.

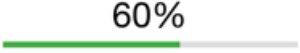**Figure 13: Replication Schedules Table**

Only one job corresponding to a replication schedule can occur at a time; if another job associated with that same replication schedule starts before the previous one has finished, the second one is canceled.

You can limit the replication jobs that are displayed by selecting filters on the left. If you do not see an expected schedule, adjust or clear the filters. Use the search box to search the list of schedules for path, database, or table names.

The **Replication Schedules** columns are described in the following table.

Table 28: Replication Schedules Table

Column	Description
ID	An internally generated ID number that identifies the schedule. Provides a convenient way to identify a schedule. Click the **ID** column label to sort the replication schedule table by ID.

Column	Description
Name	The unique name you specify when you create a schedule.
Type	The type of replication scheduled, either HDFS or Hive.
Source	The source cluster for the replication.
Destination	The destination cluster for the replication.
Throughput	Average throughput per mapper/file of all the files written. Note that throughput does not include the following information: the combined throughput of all mappers and the time taken to perform a checksum on a file after the file is written.
Progress	The progress of the replication.
Last Run	The date and time when the replication last ran. Displays **None** if the scheduled replication has not yet been run. Click the date and time link to view the Replication History page for the replication. Displays one of the following icons: • ✔ - Successful. Displays the date and time of the last run replication. • ⊗ - Failed. Displays the date and time of a failed replication. • ● - None. This scheduled replication has not yet run. • 60% - Running. Displays a spinner and bar showing the progress of the replication. Click the **Last Run** column label to sort the Replication Schedules table by the last run date.
Next Run	The date and time when the next replication is scheduled, based on the schedule parameters specified for the schedule. Hover over the date to view additional details about the scheduled replication. Click the **Next Run** column label to sort the Replication Schedules table by the next run date.
Objects	Displays on the bottom line of each row, depending on the type of replication: • **Hive** - A list of tables selected for replication. • **HDFS** - A list of paths selected for replication. For example:
Actions	The following items are available from the **Action** button: • **Show History** - Opens the **Replication History** page for a replication. See Viewing Replication History on page 494. • **Edit Configuration** - Opens the **Edit Replication Schedule** page. • **Dry Run** - Simulates a run of the replication task but does not actually copy any files or tables. After a Dry Run, you can select **Show History**, which opens the **Replication History** page where you can view any error messages and the number and size of files or tables that would be copied in an actual replication. • Click **Collect Diagnostic Data** to open the **Send Diagnostic Data** screen, which allows you to collect replication-specific diagnostic data for the last 10 runs of the schedule:

Column	Description
	1. Select **Send Diagnostic Data to Cloudera** to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle.
	2. Click **Collect and Send Diagnostic Data** to generate the bundle and open the **Replications Diagnostics Command** screen.
	3. When the command finishes, click **Download Result Data** to download a zip file containing the bundle.
	• **Run Now** - Runs the replication task immediately.
	• **Disable | Enable** - Disables or enables the replication schedule. No further replications are scheduled for disabled replication schedules.
	• **Delete** - Deletes the schedule. Deleting a replication schedule does not delete copied files or tables.

- While a job is in progress, the **Last Run** column displays a spinner and progress bar, and each stage of the replication task is indicated in the message beneath the job's row. Click the **Command Details** link to view details about the execution of the command.
- If the job is successful, the number of files copied is indicated. If there have been no changes to a file at the source since the previous job, then that file is *not* copied. As a result, after the initial job, only a subset of the files may actually be copied, and this is indicated in the success message.
- If the job fails, the ⊗ icon displays.
- To view more information about a completed job, select **Actions** > **Show History**. See

Enabling, Disabling, or Deleting A Replication Schedule

When you create a new replication schedule, it is automatically enabled. If you disable a replication schedule, it can be re-enabled at a later time.

To enable, disable, or delete a replication schedule:

- Click **Actions** > **Enable|Disable|Delete** in the row for a replication schedule.

To enable, disable, or delete multiple replication schedules:

1. Select one or more replication schedules in the table by clicking the check box the in the left column of the table.
2. Click **Actions for Selected** > **Enable|Disable|Delete**.

Viewing Replication History

You can view historical details about replication jobs on the **Replication History** page.

To view the history of a replication job:

1. Select **Backup** > **Replication Schedules** to go to the **Replication Schedules** page.
2. Locate the row for the job.
3. Click **Actions** > **Show History**.

Replication History (Replication Schedules)

		Start Time	Duration	Outcome	Files Expected		Files Copied		Files Failed		Files Deleted	Files Skipped	
Type	HDFS	˅ May 23, 2016 10:04 AM	1 min	Successful	0	(0 B)	0	(0 B)	0	(0 B)	0	0	(0 B)
Source	HDFS-1 (Cluster 1 @ n56u)												
Destination	HDFS-1 (Cluster 1)	Started At	May 23, 2016 10:04 AM			MapReduce Job	job_201605230526_0001						
Next Run	None scheduled.	Duration	a few seconds			HDFS Replication Report	Download Listing CSV Download Status CSV						
		Command Details	View			Run As Username	hdfs						
		Diagnostics	Collect Diagnostic Data										
		Message	HDFS replication succeeded.										

Figure 14: Replication History Screen (HDFS)

Replication History (Replications)

			Files	Files	Files	Files	Files
Start Time	Duration	Outcome	Tables	Expected	Copied	Failed	Deleted	Skipped

Type HIVE
Source HIVE-1 (Cluster 1)
Destination HIVE-2 (Cluster 2)
Next Run None scheduled

Start Time	Duration	Outcome	Tables	Files Expected	Files Copied	Files Failed	Files Deleted	Files Skipped
⌄ September 25, 2015 11:54 AM	0 min	Failed	1	–	–	–	–	–	–

Started At	September 25, 2015 11:54 AM
Duration	a few seconds
Command Details	View
Diagnostics	Collect Diagnostic Data
Errors	2
Impala UDFs	0
Hive Replication Report	Download Results CSV
Message	Hive Replication Failed.

Figure 15: Replication History Screen (Hive, Failed Replication)

The **Replication History** page displays a table of previously run replication jobs with the following columns:

Table 29: Replication History Table

Column	Description
Start Time	Time when the replication job started.

Expand the display and show details of the replication. In this screen, you can:

- Click the **View** link to open the **Command Details** page, which displays details and messages about each step in the execution of the command. Expand the display for a **Step** to:

  - View the actual command string.
  - View the Start time and duration of the command.
  - Click the **Context** link to view the service status page relevant to the command.
  - Select one of the tabs to view the **Role Log**, **stdout**, and **stderr** for the command.

  See Viewing Running and Recent Commands on page 241.

- Click **Collect Diagnostic Data** to open the **Send Diagnostic Data** screen, which allows you to collect replication-specific diagnostic data for this run of the schedule:

  1. Select **Send Diagnostic Data to Cloudera** to automatically send the bundle to Cloudera Support. You can also enter a ticket number and comments when sending the bundle.
  2. Click **Collect and Send Diagnostic Data** to generate the bundle and open the **Replications Diagnostics Command** screen.
  3. When the command finishes, click **Download Result Data** to download a zip file containing the bundle.

- (HDFS only) Link to view details on the **MapReduce Job** used for the replication. See Viewing and Filtering MapReduce Activities on page 249.
- (Dry Run only) View the number of **Replicable Files**. Displays the number of files that would be replicated during an actual replication.
- (Dry Run only) View the number of **Replicable Bytes**. Displays the number of bytes that would be replicated during an actual replication.
- Link to download a CSV file containing a **Replication Report**. This file lists the databases and tables that were replicated.
- View the number of **Errors** that occurred during the replication.
- View the number of **Impala UDFs** replicated. (Displays only for Hive/Impala replications where **Replicate Impala Metadata** is selected.)
- Click the link to download a CSV file containing a **Download Listing**. This file lists the files and directories that were replicated.

Column	Description
	• Click the link to download a CSV file containing **Download Status**. • If a user was specified in the **Run As Username** field when creating the replication job, the selected user displays. • View messages returned from the replication job.
Duration	Amount of time the replication job took to complete.
Outcome	Indicates success or failure of the replication job.
Files Expected	Number of files expected to be copied, based on the parameters of the replication schedule.
Files Copied	Number of files actually copied during the replication.
Tables	(Hive only) Number of tables replicated.
Files Failed	Number of files that failed to be copied during the replication.
Files Deleted	Number of files that were deleted during the replication.
Files Skipped	Number of files skipped during the replication. The replication process skips files that already exist in the destination and have not changed.

Hive/Impala Replication To and From Amazon S3

You can use Cloudera Manager to replicate Hive/Impala data and metadata to and from Amazon S3, however you cannot replicate data from one Amazon S3 instance to another using Cloudera Manager. You must have the appropriate credentials to access the Amazon S3 account and you must create buckets in Amazon S3 to store the replicated files.

When you replicate data to cloud storage with BDR, BDR also backs up file metadata, including extended attributes and ACLs.

To configure Hive/Impala replication to or from Amazon S3:

1. Create **AWS Credentials**. See [How to Configure AWS Credentials](#).

   > **Important:** If AWS S3 access keys are rotated, you must restart Cloudera Manager server; otherwise, Hive replication fails.

2. Select **Backup** > **Replication Schedules**.
3. Click **Create Schedule** > **Hive Replication**.
4. To back up data to S3:

   a. Select the Source cluster from the **Source** drop-down list.

   b. Select the Amazon S3 destination (one of the **AWS Credentials** accounts you created for Amazon S3) from the **Destination** drop-down list.

   c. Enter the path where the data should be copied to in S3. Enter using the following form:

   ```
   s3a://S3_bucket_name/path
   ```

   d. Select one of the following **Replication Options**:

      • **Metadata and Data** – Backs up the Hive data from HDFS and its associated metadata.
      • **Metadata only** – Backs up only the Hive metadata.

5. To restore data from S3:

   a. Select the Amazon S3 source (one of the **AWS Credentials** accounts you created for Amazon S3) from the **Source** drop-down list.

   b. Select the destination cluster from the **Destination** drop-down list.

c. Enter the path to the metadata file (`export.json`) where the data should be copied from in S3. Enter using the following form:

```
s3a://S3_bucket_name/path_to_metadata_file
```

d. Select one of the following **Replication Options**:

- **Metadata and Data** – Restores the Hive data from HDFS from S3 and its associated metadata.
- **Metadata only** – Restores only the Hive metadata.
- **Reference Data From Cloud** – Restores only the Hive tables and references the tables on S3 as a Hive external table. If you drop a table in Hive, the data remains on S3. Only data that was backed up using a Hive/Impala Replication schedule can be restored. However, you can restore a Hive external table that is stored in S3.

6. Complete the configuration of the Hive/Impala replication schedule by following the steps under Configuring Replication of Hive/Impala Data on page 500, beginning with step 5.d on page 501

Monitoring the Performance of Hive/Impala Replications

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

You can monitor the progress of a Hive/Impala replication schedule using performance data that you download as a CSV file from the Cloudera Manager Admin console. This file contains information about the tables and partitions being replicated, the average throughput, and other details that can help diagnose performance issues during Hive/Impala replications. You can view this performance data for running Hive/Impala replication jobs and for completed jobs.

To view the performance data for a *running* Hive/Impala replication schedule:

1. Go to **Backup** > **Replication Schedules**.
2. Locate the row for the schedule.
3. Click **Performance Reports** and select one of the following options:

- **HDFS Performance Summary** – downloads a summary performance report of the HDFS phase of the running Hive replication job.
- **HDFS Performance Full** – downloads a full performance report of the HDFS phase of the running Hive replication job.
- **Hive Performance** – downloads a report of Hive performance.

4. To view the data, import the file into a spreadsheet program such as Microsoft Excel.

To view the performance data for a *completed* Hive/Impala replication schedule:

1. Go to **Backup** > **Replication Schedules**.
2. Locate the schedule and click **Actions** > **Show History**.

The **Replication History** page for the replication schedule displays.

3. Click ❯ to expand the display of the selected schedule.

4. To view performance of the Hive phase, click **Download CSV** next to the **Hive Replication Report** label and select one of the following options:

- **Results** – download a listing of replicated tables.
- **Performance** – download a performance report for the Hive replication.

Replication History (Replication Schedules)

Type HIVE	Source HIVE-1 (Cluster 1 @ n59u)		Destination HIVE-1 (Cluster 1)		Next Run None scheduled.								

Start Time	Duration	Outcome	Tables	Files Expected		Files Copied		Files Failed		Files Deleted		Files Skipped	
❯ December 19, 2016 3:13 PM	4 min	Successful	1	1	(15.4 KiB)	0	(0 B)	0	(0 B)	0		1	(15.4 KiB)

Started At December 19, 2016 3:13 PM
Duration 4 minutes
Command Details View
Diagnostics Collect Diagnostic Data

Hive Export/Import Errors 0
Impala UDFs 0
Hive UDFs 0

HDFS Replication Report Download CSV ▾
Hive Replication Report Download CSV ▾

Results
Performance

Message 1 tables copied.

Display 10 ◆ Per Page | << < 1 - 1 >

> **Note:** The option to download the HDFS Replication Report might not appear if the HDFS phase of the replication skipped all HDFS files because they have not changed, or if the **Replicate HDFS Files** option (located on the **Advanced** tab when creating Hive/Impala replication schedules) is not selected.

See Table 30: Hive Performance Report Columns on page 511 for a description of the data in the HDFS performance reports.

5. To view performance of the HDFS phase, click **Download CSV** next to the **HDFS Replication Report** label and select one of the following options:

- **Listing** – a list of files and directories copied during the replication job.
- **Status** - full status report of files where the status of the replication is one of the following:
  - **ERROR** – An error occurred and the file was not copied.
  - **DELETED** – A deleted file.
  - **SKIPPED** – A file where the replication was skipped because it was up-to-date.
- **Error Status Only** – full status report, filtered to show files with errors only.
- **Deleted Status Only** – full status report, filtered to show deleted files only.
- **Skipped Status Only**– full status report, filtered to show skipped files only.
- **Performance** – summary performance report.
- **Full Performance** – full performance report.

See Table 27: HDFS Performance Report Columns on page 498 for a description of the data in the HDFS performance reports.

6. To view the data, import the file into a spreadsheet program such as Microsoft Excel.

The performance data is collected every two minutes. Therefore, no data is available during the initial execution of a replication job because not enough samples are available to estimate throughput and other reported data.

The data returned by the CSV files downloaded from the Cloudera Manager Admin console has the following structure:

Table 30: Hive Performance Report Columns

Hive Performance Data Columns	Description
Timestamp	Time when the performance data was collected
Host	Name of the host where the YARN or MapReduce job was running.
DbName	Name of the database.
TableName	Name of the table.
TotalElapsedTimeSecs	Number of seconds elapsed from the start of the copy operation.
TotalTableCount	Total number of tables to be copied. The value of the column will be −1 for replications where Cloudera Manager cannot determine the number of tables being changed.
TotalPartitionCount	Total number of partitions to be copied. If the source cluster is running Cloudera Manager 5.9 or lower, this column contains a value of −1 because older releases do not report this information.
DbCount	Current number of databases copied.
DbErrorCount	Number of failed database copy operations.
TableCount	Total number of tables (for all databases) copied so far.
CurrentTableCount	Total number of tables copied for current database.
TableErrorCount	Total number of failed table copy operations.
PartitionCount	Total number of partitions copied so far (for all tables).
CurrPartitionCount	Total number of partitions copied for the current table.
PartitionSkippedCount	Number of partitions skipped because they were copied in the previous run of the replication job.
IndexCount	Total number of index files copied (for all databases).
CurrIndexCount	Total number of index files copied for the current database.

Hive Performance Data Columns	Description
IndexSkippedCount	Number of Index files skipped because they were not altered. Due to a bug in Hive, this value is always zero.
HiveFunctionCount	Number of Hive functions copied.
ImpalaObjectCount	Number of Impala objects copied.

A sample CSV file, as presented in Excel, is shown here:

Timestamp	Host	DbName	TableName	TotalElapsedTimeSecs	TotalTableCount	TotalPartitionCount	DbCount	DbErrorCount	TableCount	CurrentTableCount	TableErrorCount	PartitionCount	CurrPartitionCount	PartitionSkip	IndexCount	CurrIndexCount	IndexSkippedCount	HiveFunctionCount	ImpalaObjCount
22:16.0	TargetHost-3.m	default	null	0	4	-1	1	0	0	0	0	0	0	0	0	0	0	0	0
22:17.6	TargetHost-3.m	null	null	1	4	-1	1	0	4	4	0	4	0	0	0	0	0	0	0

Note the following limitations and known issues:

- If you click the CSV download too soon after the replication job starts, Cloudera Manager returns an empty file or a CSV file that has columns headers only and a message to try later when performance data has actually been collected.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Timestamp	Host	DbName	TableName	TotalElapsed	TotalTableCc	TotalPartiti	DbCount	DbErrorCour	TableCount	CurrentTable	TableErrorCc	Partition
2	No performance statistics available yet: please try again later.												
3													
4													
5													
6													

- If you employ a proxy user with the form `user@domain`, performance data is not available through the links.
- If the replication job only replicates small files that can be transferred in less than a few minutes, no performance statistics are collected.
- For replication schedules that specify the **Dynamic** Replication Strategy, statistics regarding the last file transferred by a MapReduce job hide previous transfers performed by that MapReduce job.
- Only the last trace of each MapReduce job is reported in the CSV file.

Replicating Data to Impala Clusters

Replicating Impala Metadata

> **Note:** This feature is not available if the source and destination clusters run CDH 5.12 or higher. This feature replicated legacy Impala UDFs, which are no longer supported. Impala metadata is replicated as part of regular Hive/Impala replication operations.

Impala metadata replication is performed as a part of Hive replication. Impala replication is only supported between two CDH clusters. The Impala and Hive services must be running on both clusters.

To enable Impala metadata replication, perform the following tasks:

1. Schedule Hive replication as described in Configuring Replication of Hive/Impala Data on page 500.
2. Confirm that the **Replicate Impala Metadata** option is set to **Yes** on the **Advanced** tab in the **Create Hive Replication** dialog.

When you set the Replicate Impala Metadata option to Yes, Impala UDFs (user-defined functions) will be available on the target cluster, just as on the source cluster. As part of replicating the UDFs, the binaries in which they are defined are also replicated.

> **Note:** To run queries or execute DDL statements on tables that have been replicated to a destination cluster, you must run the Impala INVALIDATE METADATA statement on the destination cluster to prevent queries from failing. See INVALIDATE METADATA Statement

Invalidating Impala Metadata

For Impala clusters that do not use LDAP authentication, you can configure Hive/Impala replication jobs to automatically invalidate Impala metadata after replication completes. If the clusters use Sentry, the Impala user should have permissions to run `INVALIDATE METADATA`.

The configuration causes the Hive/Impala replication job to run the Impala `INVALIDATE METADATA` statement per table on the destination cluster after completing the replication. The statement purges the metadata of the replicated tables and views within the destination cluster's Impala upon completion of replication, allowing other Impala clients at the destination to query these tables successfully with accurate results. However, this operation is potentially unsafe if DDL operations are being performed on any of the replicated tables or views while the replication is running. In general, directly modifying replicated data/metadata on the destination is not recommended. Ignoring this can lead to unexpected or incorrect behavior of applications and queries using these tables or views.

> **Note:** If the source contains UDFs, you must run the `INVALIDATE METADATA` statement manually and without any tables specified even if you configure the automatic invalidation.

To configure the option, perform the following tasks:

1. Schedule a Hive/Impala replication as described in Configuring Replication of Hive/Impala Data on page 500.
2. On the **Advanced** tab, select the **Invalidate Impala Metadata on Destination** option.

Alternatively, you can run the `INVALIDATE METADATA` statement manually for replicated tables. For more information about the statement, see INVALIDATE METADATA Statement.

Using Snapshots with Replication

Some replications, especially those that require a long time to finish, can fail because source files are modified during the replication process. You can prevent such failures by using Snapshots in conjunction with Replication. This use of snapshots is automatic with CDH versions 5.0 and higher. To take advantage of this, you must enable the relevant directories for snapshots (also called making the directory *snapshottable*).

When the replication job runs, it checks to see whether the specified source directory is snapshottable. Before replicating any files, the replication job creates point-in-time snapshots of these directories and uses them as the source for file copies. This ensures that the replicated data is consistent with the source data as of the start of the replication job. The latest snapshot for the subsequent runs is retained after the replication process is completed.

A directory is *snapshottable* because it has been enabled for snapshots, or because a parent directory is enabled for snapshots. Subdirectories of a snapshottable directory are included in the snapshot. To enable an HDFS directory for snapshots (to make it snapshottable), see Enabling and Disabling HDFS Snapshots on page 544.

Hive/Impala Replication with Snapshots

If you are using Hive Replication, Cloudera recommends that you make the **Hive Warehouse Directory** snapshottable. The Hive warehouse directory is located in the HDFS file system in the location specified by the `hive.metastore.warehouse.dir` property. (The default location is `/user/hive/warehouse`.) To access this property:

1. Open Cloudera Manager and browse to the Hive service.
2. Click the **Configuration** tab.
3. In the **Search** box, type `hive.metastore.warehouse.dir`.

   The **Hive Warehouse Directory** property displays.

If you are using external tables in Hive, also make the directories hosting any external tables not stored in the Hive warehouse directory snapshottable.

Similarly, if you are using Impala and are replicating any Impala tables using Hive/Impala replication, ensure that the storage locations for the tables and associated databases are also snapshottable. See Enabling and Disabling HDFS Snapshots on page 544.

Enabling Replication Between Clusters with Kerberos Authentication

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

To enable replication between clusters, additional setup steps are required to ensure that the source and destination clusters can communicate.

> ❗ **Important:** Cloudera Backup and Disaster Recovery (BDR) works with clusters in different Kerberos realms even without a Kerberos realm trust relationship. The Cloudera Manager configuration properties **Trusted Kerberos Realms** and **Kerberos Trusted Realms** are used for Cloudera Manager and CDH configuration, and are not related to Kerberos realm trust relationships.
>
> If you are using standalone DistCp between clusters in different Kerberos realms, you must configure a realm trust. For more information, see <u>Distcp between Secure Clusters in Different Kerberos Realms</u> on page 583.

Ports

When using BDR with Kerberos authentication enabled, BDR requires all the ports listed on the following page: <u>Port Requirements for Backup and Disaster Recovery</u> on page 483.

Additionally, the port used for the Kerberos KDC Server and KRB5 services must be open to all hosts on the destination cluster. By default, this is port 88.

Considerations for Realm Names

If the source and destination clusters each use Kerberos for authentication, use one of the following configurations to prevent conflicts when running replication jobs:

- If the clusters do not use the same KDC (Kerberos Key Distribution Center), Cloudera recommends that you use different realm names for each cluster. Additionally, if you are replicating across clusters in two different realms, see the steps for <u>HDFS, Hive, and Impala Replication</u> on page 514 and <u>Hive and Impala Replication in Cloudera Manager 5.11 and Lower</u> on page 515 replication later in this topic to setup trust between those clusters.
- You can use the same realm name if the clusters use the same KDC or different KDCs that are part of a unified realm, for example where one KDC is the master and the other is a slave KDC.
-
  > ✎ **Note:** If you have multiple clusters that are used to segregate production and non-production environments, this configuration could result in principals that have equal permissions in both environments. Make sure that permissions are set appropriately for each type of environment.

> ❗ **Important:** If the source and destination clusters are in the same realm but do not use the same KDC or the KDcs are not part of a unified realm, the replication job will fail.

HDFS, Hive, and Impala Replication

1. On the hosts in the *destination* cluster, ensure that the `krb5.conf` file (typically located at `/etc/kbr5.conf`) on each host has the following information:

   - The KDC information for the *source* cluster's Kerberos realm. For example:

```
[realms]
 SRC.EXAMPLE.COM = {
  kdc = kdc01.src.example.com:88
  admin_server = kdc01.example.com:749
  default_domain = src.example.com
 }
 DST.EXAMPLE.COM = {
  kdc = kdc01.dst.example.com:88
  admin_server = kdc01.dst.example.com:749
```

```
 default_domain = dst.example.com
 }
```

- Realm mapping for the *source* cluster domain. You configure these mappings in the `[domain_realm]` section. For example:

```
[domain_realm]
 .dst.example.com = DST.EXAMPLE.COM
 dst.example.com = DST.EXAMPLE.COM
 .src.example.com = SRC.EXAMPLE.COM
 src.example.com = SRC.EXAMPLE.COM
```

2. On the *destination* cluster, use Cloudera Manager to add the realm of the *source* cluster to the **Trusted Kerberos Realms** configuration property:

   a. Go to the HDFS service.
   b. Click the **Configuration** tab.
   c. In the search field type `Trusted Kerberos` to find the **Trusted Kerberos Realms** property.
   d. Click the plus sign icon, and then enter the *source* cluster realm.
   e. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

3. Go to **Administration** > **Settings**.
4. In the search field, type `domain name`.
5. In the **Domain Name(s)** field, enter any domain or host names you want to map to the destination cluster KDC. Use the plus sign icon to add as many entries as you need. The entries in this property are used to generate the `domain_realm` section in `krb5.conf`.
6. If `domain_realm` is configured in the **Advanced Configuration Snippet (Safety Valve) for remaining krb5.conf**, remove the entries for it.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

Hive and Impala Replication in Cloudera Manager 5.11 and Lower

> **Note:** If the source and destination clusters both run Cloudera Manager 5.12 or higher, you do not need to complete the steps in this section. These additional steps are no longer required for Hive or Impala replication. If you are using Cloudera Manager 5.11 or lower, complete the steps above in HDFS, Hive, and Impala Replication on page 514, and then complete the steps in the following section.

1. Perform the procedure described in the previous section.
2. On the hosts in the *source* cluster, ensure that the `krb5.conf` file on each host has the following information:

   - The kdc information for the *destination* cluster's Kerberos realm.
   - Domain/host-to-realm mapping for the *destination* cluster NameNode hosts.

3. On the *source* cluster, use Cloudera Manager to add the realm of the *destination* cluster to the Trusted Kerberos Realms configuration property.

   a. Go to the HDFS service.
   b. Click the **Configuration** tab.
   c. In the search field type "Trusted Kerberos" to find the **Trusted Kerberos Realms** property.
   d. Enter the destination cluster realm.
   e. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

   It is not necessary to restart any services on the source cluster.

Kerberos Connectivity Test

As part of **Test Connectivity**, Cloudera Manager tests for properly configured Kerberos authentication on the source and destination clusters that run the replication. **Test Connectivity** runs automatically when you add a peer for replication, or you can manually initiate Test Connectivity from the **Actions** menu.

This feature is available when the source and destination clusters run Cloudera Manager 5.12 or later. You can disable the Kerberos connectivity test by setting `feature_flag_test_kerberos_connectivity` to `false` with the Cloudera Manager API: `api/<version>/cm/config`.

If the test detects any issues with the Kerberos configuration, Cloudera Manager provides resolution steps based on whether Cloudera Manager manages the Kerberos configuration file.
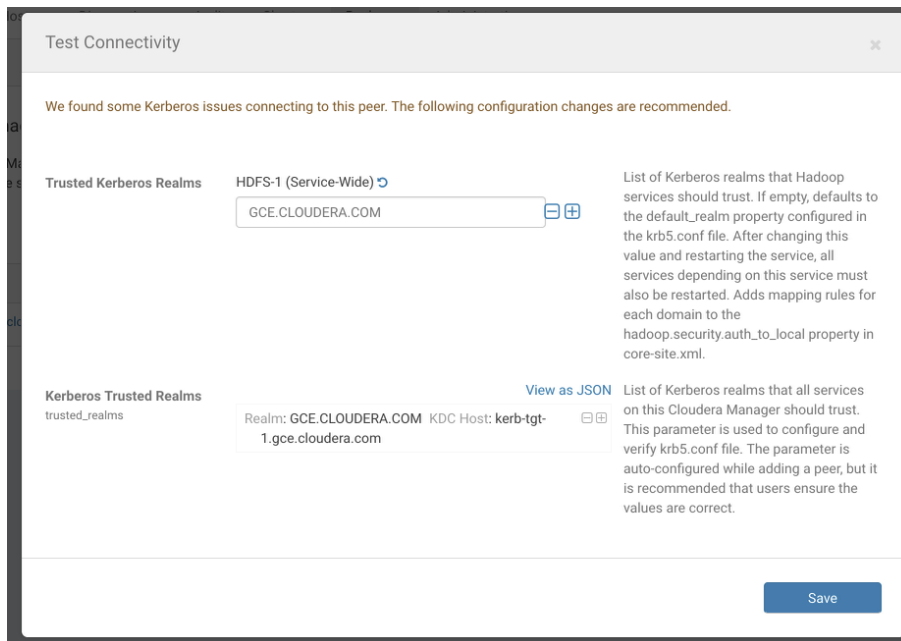
Cloudera Manager tests the following scenarios:

- Whether both clusters have Kerberos enabled. If one cluster uses Kerberos but the other does not, replication is not supported.
- Whether both clusters are in the same Kerberos realm. Clusters in the same realm must share the same KDC or the KDCs must be in a unified realm.
- Whether clusters are in different Kerberos realms. If the clusters are in different realms, the destination cluster must be configured according to the following criteria:
  - Destination HDFS services must have the correct **Trusted Kerberos Realms** setting.
  - The `krb5.conf` file has the correct `domain_realm` mapping on all the hosts.
  - The `krb5.conf` file has the correct `realms` information on all the hosts.
- Whether the local and peer KDC are running on an available port. This port must be open for all hosts in the cluster. The default port is 88.

After Cloudera Manager runs the tests, Cloudera Manager makes recommendations to resolve any Kerberos configuration issues.
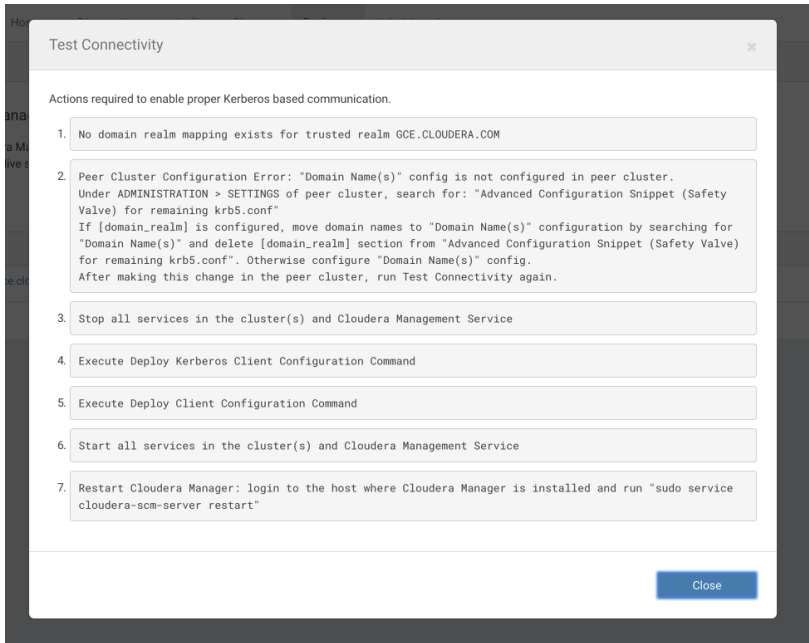
Kerberos Recommendations

If Cloudera Manager manages the Kerberos configuration file, Cloudera Manager configures Kerberos correctly for you and then provides the set of commands that you must manually run to finish configuring the clusters. The following screen shots show the prompts that Cloudera Manager provides in cases of improper configuration:
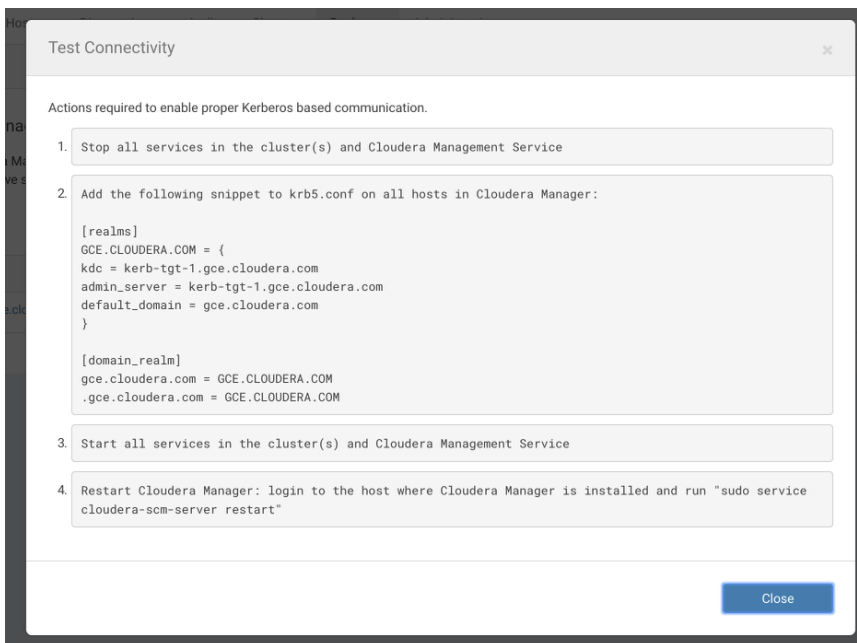
Configuration changes:

Steps to complete configuration:

```
Test Connectivity                                                          ✕

Actions required to enable proper Kerberos based communication.

1.  No domain realm mapping exists for trusted realm GCE.CLOUDERA.COM

2.  Peer Cluster Configuration Error: "Domain Name(s)" config is not configured in peer cluster.
    Under ADMINISTRATION > SETTINGS of peer cluster, search for: "Advanced Configuration Snippet (Safety
    Valve) for remaining krb5.conf"
    If [domain_realm] is configured, move domain names to "Domain Name(s)" configuration by searching for
    "Domain Name(s)" and delete [domain_realm] section from "Advanced Configuration Snippet (Safety Valve)
    for remaining krb5.conf". Otherwise configure "Domain Name(s)" config.
    After making this change in the peer cluster, run Test Connectivity again.

3.  Stop all services in the cluster(s) and Cloudera Management Service

4.  Execute Deploy Kerberos Client Configuration Command

5.  Execute Deploy Client Configuration Command

6.  Start all services in the cluster(s) and Cloudera Management Service

7.  Restart Cloudera Manager: login to the host where Cloudera Manager is installed and run "sudo service
    cloudera-scm-server restart"

                                                                    Close
```

If Cloudera Manager does not manage the Kerberos configuration file, Cloudera manager provides the manual steps required to correct the issue. For example, the following screen shot shows the steps required to properly configure Kerberos:

```
Test Connectivity                                                          ✕

Actions required to enable proper Kerberos based communication.

1.  Stop all services in the cluster(s) and Cloudera Management Service

2.  Add the following snippet to krb5.conf on all hosts in Cloudera Manager:

    [realms]
    GCE.CLOUDERA.COM = {
    kdc = kerb-tgt-1.gce.cloudera.com
    admin_server = kerb-tgt-1.gce.cloudera.com
    default_domain = gce.cloudera.com
    }

    [domain_realm]
    gce.cloudera.com = GCE.CLOUDERA.COM
    .gce.cloudera.com = GCE.CLOUDERA.COM

3.  Start all services in the cluster(s) and Cloudera Management Service

4.  Restart Cloudera Manager: login to the host where Cloudera Manager is installed and run "sudo service
    cloudera-scm-server restart"

                                                                    Close
```

Replication of Encrypted Data

HDFS supports encryption of data at rest (including data accessed through Hive). This topic describes how replication works within and between encryption zones and how to configure replication to avoid failures due to encryption.

Encrypting Data in Transit Between Clusters

A source directory and destination directory may or may not be in an encryption zone. If the destination directory is in an encryption zone, the data on the destination directory is encrypted. If the destination directory is not in an encryption zone, the data on that directory is not encrypted, even if the source directory is in an encryption zone. For

more information about HDFS encryption zones, see [HDFS Transparent Encryption](). Encryption zones are not supported in CDH versions 5.1 or lower.

When you configure encryption zones, you also configure a Key Management Server (KMS) to manage encryption keys. During replication, Cloudera Manager uses TLS/SSL to encrypt the keys when they are transferred from the source cluster to the destination cluster.

When you configure encryption zones, you also configure a Key Management Server (KMS) to manage encryption keys. When a HDFS replication command that specifies an encrypted source directory runs, Cloudera Manager temporarily copies the encryption keys from the source cluster to the destination cluster, using TLS/SSL (if configured for the KMS) to encrypt the keys. Cloudera Manager then uses these keys to decrypt the encrypted files when they are received from the source cluster before writing the files to the destination cluster.

> **Important:** When you configure [HDFS replication](), you must select the **Skip Checksum check** property to prevent replication failure in the following cases:
>
> - Replications from an encrypted zone on the source cluster to an encrypted zone on a destination cluster.
> - Replications from an encryption zone on the source cluster to an unencrypted zone on the destination cluster.
> - Replications from an unencrypted zone on the source cluster to an encrypted zone on the destination cluster.

Even when the source and destination directories are both in encryption zones, the data is decrypted as it is read from the source cluster (using the key for the source encryption zone) and encrypted again when it is written to the destination cluster (using the key for the destination encryption zone). The data transmission is encrypted if you have [configured encryption for HDFS Data Transfer.]()

> **Note:** The decryption and encryption steps happen in the same process on the hosts where the MapReduce jobs that copy the data run. Therefore, data in plain text only exists within the memory of the Mapper task. If a KMS is in use on either the source or destination clusters, and you are using encrypted zones for either the source or destination directories, configure TLS/SSL for the KMS to prevent transferring the key to the mapper task as plain text.

During replication, data travels from the source cluster to the destination cluster using `distcp`. For clusters that use encryption zones, configure encryption of KMS key transfers between the source and destination using TLS/SSL. See [Configuring TLS/SSL for the KMS]().

To configure encryption of data transmission between source and destination clusters:

- Enable TLS/SSL for HDFS clients on both the source and the destination clusters. For instructions, see [Configuring TLS/SSL for HDFS, YARN and MapReduce](). You may also need to configure trust between the SSL certificates on the source and destination.
- Enable TLS/SSL for the two peer Cloudera Manager Servers. See [Configuring TLS Encryption for Cloudera Manager]().
- Encrypt data transfer using HDFS Data Transfer Encryption. See [Configuring Encrypted Transport for HDFS]().

The following blog post provides additional information about encryption with HDFS: [http://blog.cloudera.com/blog/2013/03/how-to-set-up-a-hadoop-cluster-with-network-encryption/]().

Security Considerations

The user you specify with the **Run As** field when scheduling a replication job requires full access to both the key and the data directories being replicated. This is not a recommended best practice for KMS management. If you change permissions in the KMS to enable this requirement, you could accidentally provide access for this user to data in other encryption zones using the same key. If a user is not specified in the **Run As** field, the replication runs as the default user, `hdfs`.

To access encrypted data, the user must be authorized on the KMS for the encryption zones they need to interact with. The user you specify with the **Run As** field when scheduling a replication must have this authorization. The key administrator must add ACLs to the KMS for that user to prevent authorization failure.

Key transfer using the KMS protocol from source to the client uses the REST protocol, which requires that you configure TLS/SSL for the KMS. When TLS/SSL is enabled, keys are not transferred over the network as plain text.

See Encryption Mechanisms Overview.

HBase Replication

If your data is already in an HBase cluster, replication is useful for getting the data into additional HBase clusters. In HBase, cluster replication refers to keeping one cluster state synchronized with that of another cluster, using the write-ahead log (WAL) of the source cluster to propagate the changes. Replication is enabled at column family granularity. Before enabling replication for a column family, create the table and all column families to be replicated, on the destination cluster. Replication is supported both from CDH 5 to CDH 6 and from CDH 6 to CDH 5, the source and destination cluster do not have to run the same major version of CDH.

Cluster replication uses an active-push methodology. An HBase cluster can be a source (also called *active*, meaning that it writes new data), a destination (also called *passive*, meaning that it receives data using replication), or can fulfill both roles at once. Replication is asynchronous, and the goal of replication is consistency.

When data is replicated from one cluster to another, the original source of the data is tracked with a cluster ID, which is part of the metadata. All clusters that have already consumed the data are also tracked. This prevents replication loops.

Common Replication Topologies

- A central source cluster might propagate changes to multiple destination clusters, for failover or due to geographic distribution.
- A source cluster might push changes to a destination cluster, which might also push its own changes back to the original cluster.
- Many different low-latency clusters might push changes to one centralized cluster for backup or resource-intensive data-analytics jobs. The processed data might then be replicated back to the low-latency clusters.
- Multiple levels of replication can be chained together to suit your needs. The following diagram shows a hypothetical scenario. Use the arrows to follow the data paths.
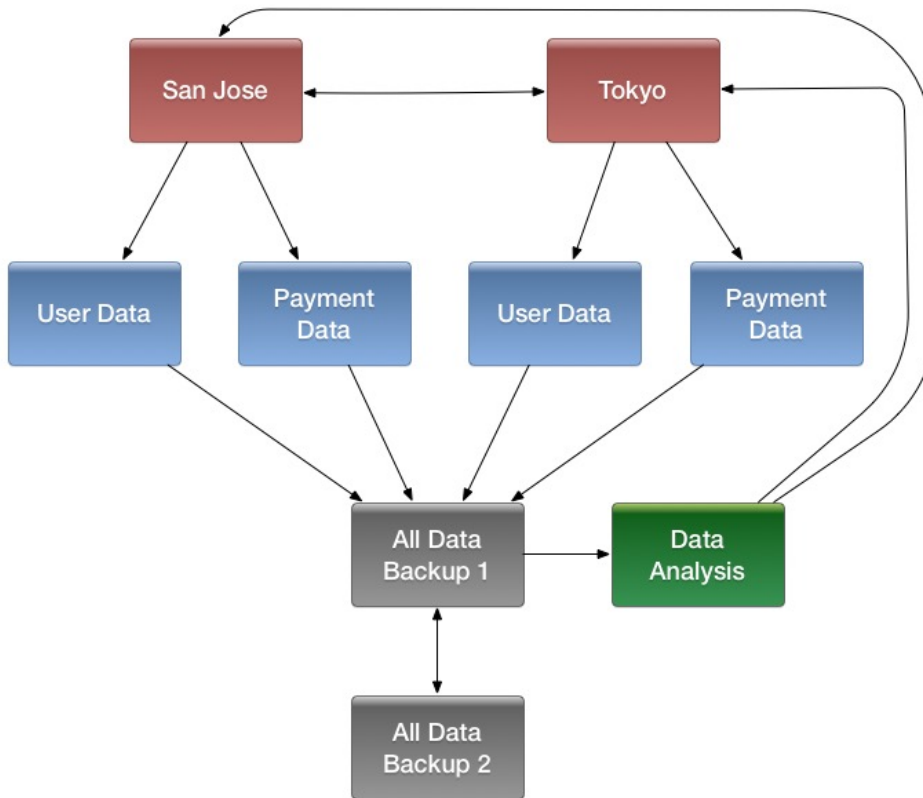
At the top of the diagram, the `San Jose` and `Tokyo` clusters, shown in red, replicate changes to each other, and each also replicates changes to a `User Data` and a `Payment Data` cluster.

Each cluster in the second row, shown in blue, replicates its changes to the `All Data Backup 1` cluster, shown in grey. The `All Data Backup 1` cluster replicates changes to the `All Data Backup 2` cluster (also shown in grey), as well as the `Data Analysis` cluster (shown in green). `All Data Backup 2` also propagates any of its own changes back to `All Data Backup 1`.

The `Data Analysis` cluster runs MapReduce jobs on its data, and then pushes the processed data back to the `San Jose` and `Tokyo` clusters.

Notes about Replication

- The timestamps of the replicated HLog entries are kept intact. In case of a collision (two entries identical as to row key, column family, column qualifier, and timestamp) only the entry arriving later write will be read.
- Increment Column Values (ICVs) are treated as simple puts when they are replicated. In the case where each side of replication is active (new data originates from both sources, which then replicate each other), this may be undesirable, creating identical counters that overwrite one another. (See https://issues.apache.org/jira/browse/HBase-2804.)
- Make sure the source and destination clusters are time-synchronized with each other. Cloudera recommends you use Network Time Protocol (NTP).
- Some changes are not replicated and must be propagated through other means, such as Snapshots or CopyTable.

    - Data that existed in the active cluster before replication was enabled.

    - Operations that bypass the WAL, such as when using BulkLoad or API calls such as `writeToWal(false)`.

    - Table schema modifications.

Requirements

Before configuring replication, make sure your environment meets the following requirements:

- You must manage ZooKeeper yourself. It must not be managed by HBase, and must be available throughout the deployment.
- Each host in both clusters must be able to reach every other host, including those in the ZooKeeper cluster.
- Every table that contains families that are scoped for replication must exist on each cluster and have exactly the same name. If your tables do not yet exist on the destination cluster, see Creating the Empty Table On the Destination Cluster on page 525.
- HBase version 0.92 or greater is required for complex replication topologies, such as active-active.

Deploying HBase Replication

Follow these steps to enable replication from one cluster to another.

> **Important:** You cannot run replication-related HBase commands as an HBase administrator. To run replication-related HBase commands, you must have HBase user permissions. If ZooKeeper uses Kerberos, configure HBase Shell to authenticate to ZooKeeper using Kerberos before attempting to run replication-related commands. No replication-related ACLs are available at this time.

1. Configure and start the source and destination clusters.
2. Create tables with the same names and column families on both the source and destination clusters, so that the destination cluster knows where to store data it receives. All hosts in the source and destination clusters should be reachable to each other. See Creating the Empty Table On the Destination Cluster on page 525.
3. On the source cluster, enable replication in Cloudera Manager, or by setting `hbase.replication` to `true` in `hbase-site.xml`.
4. Obtain Kerberos credentials as the HBase principal. Substitute your `fully.qualified.domain.name` and realm in the following command:

```
$ kinit -k -t /etc/hbase/conf/hbase.keytab
hbase/fully.qualified.domain.name@YOUR-REALM.COM
```

5. On the source cluster, in HBase Shell, add the destination cluster as a peer, using the `add_peer` command. The syntax is as follows:

```
add_peer 'ID', 'CLUSTER_KEY'
```

The ID must be a short integer. To compose the `CLUSTER_KEY`, use the following template:

```
hbase.zookeeper.quorum:hbase.zookeeper.property.clientPort:zookeeper.znode.parent
```

If both clusters use the same ZooKeeper cluster, you must use a different **zookeeper.znode.parent**, because they cannot write in the same folder.

6. On the source cluster, configure each column family to be replicated by setting its `REPLICATION_SCOPE` to 1, using commands such as the following in HBase Shell.

```
hbase> disable 'example_table'
hbase> alter 'example_table', {NAME => 'example_family', REPLICATION_SCOPE => '1'}
hbase> enable 'example_table'
```

7. Verify that replication is occurring by examining the logs on the source cluster for messages such as the following.

```
Considering 1 rs, with ratio 0.1
Getting 1 rs from peer cluster # 0
Choosing peer 192.0.2.49:62020
```

8. To verify the validity of replicated data, use the included `VerifyReplication` MapReduce job on the source cluster, providing it with the ID of the replication peer and table name to verify. Other options are available, such as a time range or specific families to verify.

The command has the following form:

```
hbase org.apache.hadoop.hbase.mapreduce.replication.VerifyReplication
[--starttime=timestamp1] [--stoptime=timestamp] [--families=comma separated list of
families] <peerId> <tablename>
```

The `VerifyReplication` command prints `GOODROWS` and `BADROWS` counters to indicate rows that did and did not replicate correctly.

Replicating Across Three or More Clusters

When configuring replication among three or more clusters, Cloudera recommends you enable `KEEP_DELETED_CELLS` on column families in the destination cluster, where `REPLICATION_SCOPE=1` in the source cluster. The following commands show how to enable this configuration using HBase Shell.

- On the source cluster:

```
create 't1',{NAME=>'f1', REPLICATION_SCOPE=>1}
```

- On the destination cluster:

```
create 't1',{NAME=>'f1', KEEP_DELETED_CELLS=>'true'}
```

Enabling Replication on a Specific Table

To enable replication for a specific table on the source cluster, run the `enable_table_replication <table>` command from the HBase shell on a cluster where a peer has been configured.

Running `enable_table_replication <table>` does the following:

1. Verifies that the table exists on the source cluster.
2. If the table does not exist on the remote cluster, uses the peer configuration to duplicate the table schema (including splits) on the remote cluster.
3. Enables replication on that table.

Configuring Secure Replication

The following steps describe how to set up secure replication between clusters. The steps are the same whether your clusters are all in the same realm or not, with the exception of the last step.

The last step involves setting up custom secure replication configurations per peer. This can be convenient when you need to replicate to a cluster that uses different cross-realm authentication rules than the source cluster. For example, a cluster in Realm A may be allowed to replicate to Realm B and Realm C, but Realm B may not be allowed to replicate to Realm C. If you do not need this feature, skip the last step.

To use this feature, service-level principals and keytabs (specific to HBase) must be specified when you create the cluster peers using HBase Shell.

> **Note:** HBase peer-to-peer replication from a non-Kerberized cluster to a Kerberized cluster is not supported.

1. Set up Kerberos on your cluster, as described in Enabling Kerberos Authentication for CDH.
2. If necessary, configure Kerberos cross-realm authentication.

   - At the command line, use the `list_principals` command to list the `kdc`, `admin_server`, and `default_domain` for each realm.
   - Add this information to each cluster using Cloudera Manager. For each cluster, go to **HDFS** > **Configuration** > **Trusted Kerberos Realms**. Add the target and source. This requires a restart of HDFS.

3. Configure ZooKeeper.

4. Configure the following HDFS parameters on both clusters, in Cloudera Manager or in the listed files if you do not use Cloudera Manager:

> **Note:**
>
> If you use Cloudera Manager to manage your cluster, do not set these properties directly in configuration files, because Cloudera Manager will overwrite or ignore these settings. You must set these properties in Cloudera Manager.
>
> For brevity, the Cloudera Manager setting names are not listed here, but you can search by property name. For instance, in the HDFS service configuration screen, search for **dfs.encrypt.data.transfer**. The **Enable Data Transfer Encryption** setting is shown. Selecting the box is equivalent to setting the value to `true`.

```
<!-- In hdfs-site.xml or advanced configuration snippet -->
<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>true</value>
</property>
<property>
  <name>dfs.data.transfer.protection</name>
  <value>privacy</value>
</property>

<!-- In core-site.xml or advanced configuration snippet -->
<property>
  <name>hadoop.security.authorization</name>
  <value>true</value>
</property>
<property>
  <name>hadoop.rpc.protection</name>
  <value>privacy</value>
</property>
<property>
  <name>hadoop.security.crypto.cipher.suite</name>
  <value>AES/CTR/NoPadding</value>
</property>
<property>
  <name>hadoop.ssl.enabled</name>
  <value>true</value>
</property>
```

5. Configure the following HBase parameters on both clusters, using Cloudera Manager or in `hbase-site.xml` if you do not use Cloudera Managert.

```
<!-- In hbase-site.xml -->
<property>
  <name>hbase.rpc.protection</name>
  <value>privacy</value>
</property>
<property>
  <name>hbase.thrift.security.qop</name>
  <value>auth-conf</value>
</property>
<property>
  <name>hbase.thrift.ssl.enabled</name>
  <value>true</value>
</property>
<property>
  <name>hbase.rest.ssl.enabled</name>
  <value>true</value>
</property>
<property>
  <name>hbase.ssl.enabled</name>
  <value>true</value>
</property>
```

```
<property>
  <name>hbase.security.authentication</name>
  <value>kerberos</value>
</property>
<property>
  <name>hbase.security.authorization</name>
  <value>true</value>
</property>
<property>
  <name>hbase.secure.rpc.engine</name>
  <value>true</value>
</property>
```

6. Add the cluster peers using the simplified `add_peer` syntax, as described in <u>Add Peer</u>.

```
add_peer 'ID', 'CLUSTER_KEY'
```

7. If you need to add any peers which require custom security configuration, modify the `add_peer` syntax, using the following examples as a model.

```
add_peer 'vegas', CLUSTER_KEY => 'zk1.vegas.example.com:2181:/hbase',
        CONFIG => {'hbase.master.kerberos.principal'        => 'hbase/_HOST@TO_VEGAS',

                   'hbase.regionserver.kerberos.principal'    => 'hbase/_HOST@TO_VEGAS',

                   'hbase.regionserver.keytab.file'           =>
'/keytabs/vegas_hbase.keytab',
                   'hbase.master.keytab.file'                 =>
'/keytabs/vegas_hbase.keytab'},
         TABLE_CFS => {"tbl" => [cf1']}
add_peer 'atlanta', CLUSTER_KEY => 'zk1.vegas.example.com:2181:/hbase',
         CONFIG => {'hbase.master.kerberos.principal'          =>
'hbase/_HOST@TO_ATLANTA',
                   'hbase.regionserver.kerberos.principal'    =>
'hbase/_HOST@TO_ATLANTA',
                   'hbase.regionserver.keytab.file'           =>
'/keytabs/atlanta_hbase.keytab',
                   'hbase.master.keytab.file'                 =>
'/keytabs/atlanta_hbase.keytab'},
         TABLE_CFS => {"tbl" => [cf2']}
```

Disabling Replication at the Peer Level

Use the command `disable_peer ("<peerID>")` to disable replication for a specific peer. This will stop replication to the peer, but the logs will be kept for future reference.

> **Note:** This log accumulation is a powerful side effect of the `disable_peer` command and can be used to your advantage. See <u>Initiating Replication When Data Already Exists</u> on page 525.

To re-enable the peer, use the command `enable_peer(<"peerID">)`. Replication resumes.

Examples:

• To disable peer 1:

```
disable_peer("1")
```

• To re-enable peer 1:

```
enable_peer("1")
```

Stopping Replication in an Emergency

If replication is causing serious problems, you can stop it while the clusters are running.

Open the shell on the source cluster and use the `disable_peer` command for each peer, then the `disable_table_replication` command. For example:

```
hbase> disable_peer("1")
hbase> disable_table_replication
```

Already queued edits will be replicated after you use the `disable_table_replication` command, but new entries will not. See [Understanding How WAL Rolling Affects Replication](#) on page 526.

To start replication again, use the `enable_peer` command.

Creating the Empty Table On the Destination Cluster

If the table to be replicated does not yet exist on the destination cluster, you must create it. The easiest way to do this is to extract the schema using HBase Shell.

1. On the source cluster, `describe` the table using HBase Shell. The output below has been reformatted for readability.

```
hbase> describe acme_users

Table acme_users is ENABLED
acme_users
COLUMN FAMILIES DESCRIPTION
{NAME => 'user', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE',
REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE',
MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE',
BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'false'}
```

2. Copy the output and make the following changes:

   - For the TTL, change `FOREVER` to `org.apache.hadoop.hbase.HConstants::FOREVER`.
   - Add the word `CREATE` before the table name.
   - Remove the line `COLUMN FAMILIES DESCRIPTION` and everything above the table name.

   The result is a command like the following:

```
"CREATE 'cme_users' ,

{NAME => 'user', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE',
REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE',
MIN_VERSIONS => '0', TTL => org.apache.hadoop.hbase.HConstants::FOREVER,
KEEP_DELETED_CELLS => 'FALSE',
BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'false'}
```

3. On the destination cluster, paste the command from the previous step into HBase Shell to create the table.

Initiating Replication When Data Already Exists

You may need to start replication from some point in the past. For example, suppose you have a primary HBase cluster in one location and are setting up a disaster-recovery (DR) cluster in another. To initialize the DR cluster, you need to copy over the existing data from the primary to the DR cluster, so that when you need to switch to the DR cluster you have a full copy of the data generated by the primary cluster. Once that is done, replication of new data can proceed as normal.

One way to do this is to take advantage of the write accumulation that happens when a replication peer is disabled.

1. Start replication.
2. Add the destination cluster as a peer and immediately disable it using `disable_peer`.
3. On the source cluster, take a [snapshot](#) of the table and export it. The snapshot command flushes the table from memory for you.
4. On the destination cluster, import and restore the snapshot.

5. Run `enable_peer` to re-enable the destination cluster.

Replicating Pre-existing Data in an Active-Active Deployment

In the case of active-active replication, run the `copyTable` job before starting the replication. (If you start the job after enabling replication, the second cluster will re-send the data to the first cluster, because `copyTable` does not edit the `clusterId` in the mutation objects. The following is one way to accomplish this:

1. Run the `copyTable` job and note the start timestamp of the job.
2. Start replication.
3. Run the `copyTable` job again with a start time equal to the start time you noted in step 1.

This results in some data being pushed back and forth between the two clusters; but it minimizes the amount of data.

Understanding How WAL Rolling Affects Replication

When you add a new peer cluster, it only receives new writes from the source cluster **since the last time the WAL was rolled**.

The following diagram shows the consequences of adding and removing peer clusters with unpredictable WAL rolling occurring. Follow the time line and notice which peer clusters receive which writes. Writes that occurred before the WAL is rolled are **not** retroactively replicated to new peers that were not participating in the cluster before the WAL was rolled.

Configuring Secure HBase Replication

If you want to make HBase Replication secure, follow the instructions under HBase Authentication.

Restoring Data From A Replica

One of the main reasons for replications is to be able to restore data, whether during disaster recovery or for other reasons. During restoration, the *source* and *sink* roles are reversed. The source is the replica cluster, and the sink is the cluster that needs restoration. This can be confusing, especially if you are in the middle of a disaster recovery scenario. The following image illustrates the role reversal between normal production and disaster recovery.

Follow these instructions to recover HBase data from a replicated cluster in a disaster recovery scenario.

1. Change the value of the column family property `REPLICATION_SCOPE` on the sink to `0` for each column to be restored, so that its data will not be replicated during the restore operation.
2. Change the value of the column family property `REPLICATION_SCOPE` on the source to `1` for each column to be restored, so that its data will be replicated.
3. Use the `CopyTable` or `distcp` commands to import the data from the backup to the sink cluster, as outlined in Initiating Replication When Data Already Exists on page 525.
4. Add the sink as a replication peer to the source, using the `add_peer` command as discussed in Deploying HBase Replication on page 521. If you used `distcp` in the previous step, restart or rolling restart both clusters, so that the RegionServers will pick up the new files. If you used `CopyTable`, you do not need to restart the clusters. New data will be replicated as it is written.
5. When restoration is complete, change the `REPLICATION_SCOPE` values back to their values before initiating the restoration.

Verifying that Replication is Working

To verify that HBase replication is working, follow these steps to confirm data has been replicated from a source cluster to a remote destination cluster.

1. Install and configure YARN on the source cluster.

   If YARN cannot be used in the source cluster, configure YARN on the destination cluster to verify replication. If neither the source nor the destination clusters can have YARN installed, you can configure the tool to use local mode; however, performance and consistency could be negatively impacted.

2. Make sure that you have the required permissions:

   - You have sudo permissions to run commands as the hbase user, or a user with admin permissions on both clusters.
   - You are an hbase user configured for submitting jobs with YARN.

   > **Note:** To use the hbase user in a secure cluster, use Cloudera Manager to add the hbase user as a YARN whitelisted user. For a new installation, the hbase user is already added to the whitelisted users. In addition, /user/hbase should exist on HDFS and owned as the hbase user, because YARN will create a job staging directory there.

3. Run the `VerifyReplication` command:

```
src-node$ sudo -u hbase hbase
org.apache.hadoop.hbase.mapreduce.replication.VerifyReplication peer1 table1
```

```
...
        org.apache.hadoop.hbase.mapreduce.replication.VerifyReplication$Verifier$Counters

                BADROWS=2
                CONTENT_DIFFERENT_ROWS=1
                GOODROWS=1
                ONLY_IN_PEER_TABLE_ROWS=1
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
```

The following table describes the `VerifyReplication` counters:

Table 31: VerifyReplication Counters

Counter	Description
GOODROWS	Number of rows. On both clusters, and all values are the same.
CONTENT_DIFFERENT_ROWS	The key is the same on both source and destination clusters for a row, but the value differs.
ONLY_IN_SOURCE_TABLE_ROWS	Rows that are only present in the source cluster but not in the destination cluster.
ONLY_IN_PEER_TABLE_ROWS	Rows that are only present in the destination cluster but not in the source cluster.
BADROWS	Total number of rows that differ from the source and destination clusters; the sum of CONTENT_DIFFERENT_ROWS + ONLY_IN_SOURCE_TABLE_ROWS + ONLY_IN_PEER_TABLE_ROWS

By default, `VerifyReplication` compares the entire content of `table1` on the source cluster against `table1` on the destination cluster that is configured to use the replication peer `peer1`.

Use the following options to define the period of time, versions, or column families

Table 32: VerifyReplication Counters

Option	Description
--starttime=<timestamp>	Beginning of the time range, in milliseconds. Time range is forever if no end time is defined.
--endtime=<timestamp>	End of the time range, in milliseconds.
--versions=<versions>	Number of cell versions to verify.
--families=<cf1,cf2,..>	Families to copy; separated by commas.

The following example, verifies replication only for rows with a timestamp range of one day:

```
src-node$ sudo -u hbase hbase
org.apache.hadoop.hbase.mapreduce.replication.VerifyReplication --starttime=1472499077000
 --endtime=1472585477000 --families=c1  peer1 table1
```

Replication Caveats

- Two variables govern replication: `hbase.replication` as described above under Deploying HBase Replication on page 521, and a replication znode. Stopping replication (using `disable_table_replication` as above) sets the znode to `false`. Two problems can result:

- If you add a new RegionServer to the active cluster while replication is stopped, its current log will not be added to the replication queue, because the replication znode is still set to `false`. If you restart replication at this point (using `enable_peer`), entries in the log will not be replicated.
- Similarly, if a log rolls on an existing RegionServer on the active cluster while replication is stopped, the new log will not be replicated, because the replication znode was set to `false` when the new log was created.

- In the case of a long-running, write-intensive workload, the destination cluster may become unresponsive if its meta-handlers are blocked while performing the replication. CDH has three properties to deal with this problem:

  - `hbase.regionserver.replication.handler.count` - the number of replication handlers in the destination cluster (default is 3). Replication is now handled by separate handlers in the destination cluster to avoid the above-mentioned sluggishness. Increase it to a high value if the ratio of active to passive RegionServers is high.
  - `replication.sink.client.retries.number` - the number of times the HBase replication client at the sink cluster should retry writing the WAL entries (default is 1).
  - `replication.sink.client.ops.timeout` - the timeout for the HBase replication client at the sink cluster (default is 20 seconds).

- For namespaces, tables, column families, or cells with associated ACLs, the ACLs themselves are not replicated. The ACLs need to be re-created manually on the target table. This behavior opens up the possibility for the ACLs could be different in the source and destination cluster.

Snapshots

You can create HBase and HDFS snapshots using Cloudera Manager or by using the command line.

- HBase snapshots allow you to create point-in-time backups of tables without making data copies, and with minimal impact on RegionServers. HBase snapshots are supported for clusters running CDH 4.2 or higher.
- HDFS snapshots allow you to create point-in-time backups of directories or the entire filesystem without actually cloning the data. They can improve data replication performance and prevent errors caused by changes to a source directory. These snapshots appear on the filesystem as read-only directories that can be accessed just like other ordinary directories.

NEW! View a video about [Using Snapshots and Cloudera Manager to Back Up Data](#).

Cloudera Manager Snapshot Policies

Minimum Required Role: [BDR Administrator](#) (also provided by **Full Administrator**)

Cloudera Manager enables the creation of snapshot policies that define the directories or tables to be snapshotted, the intervals at which snapshots should be taken, and the number of snapshots that should be kept for each snapshot interval. For example, you can create a policy that takes both daily and weekly snapshots, and specify that seven daily snapshots and five weekly snapshots should be maintained.

> **Note:** You can improve the reliability of [Data Replication](#) on page 484 by also using snapshots. See [Using Snapshots with Replication](#) on page 513.

Managing Snapshot Policies

> **Note:** You must enable an HDFS directory for snapshots to allow snapshot policies to be created for that directory. To designate a HDFS directory as snapshottable, follow the procedure in [Enabling and Disabling HDFS Snapshots](#) on page 544.

To create a snapshot policy:

1. Select **Backup** > **Snapshot Policies** in the top navigation bar.

Existing snapshot policies are shown in a table. See [Snapshot Policies Page](#) on page 531.

2. To create a new policy, click **Create Snapshot Policy**.
3. From the drop-down list, select the service (HDFS or HBase) and cluster for which you want to create a policy.
4. Provide a name for the policy and, optionally, a description.
5. Specify the directories or tables to include in the snapshot.

> ⊘ **Important:** Do not take snapshots of the root directory.

- For an HDFS service, select the paths of the directories to include in the snapshot. The drop-down list allows you to select only directories that are enabled for snapshotting. If no directories are enabled for snapshotting, a warning displays.

  Click ✚ to add a path and ➖ to remove a path.

- For an HBase service, list the tables to include in your snapshot. You can use a [Java regular expression](#) to specify a set of tables. For example, `finance.*` matchs all tables with names starting with `finance`. You can also create a snapshot for all tables in a given namespace, using the `{namespace}:.*` syntax.

6. Specify the snapshot **Schedule**. You can schedule snapshots hourly, daily, weekly, monthly, or yearly, or any combination of those. Depending on the frequency you select, you can specify the time of day to take the snapshot, the day of the week, day of the month, or month of the year, and the number of snapshots to keep at each interval. Each time unit in the schedule information is shared with the time units of larger granularity. That is, the minute value is shared by all the selected schedules, hour by all the schedules for which hour is applicable, and so on. For example, if you specify that hourly snapshots are taken at the half hour, and daily snapshots taken at the hour 20, the daily snapshot will occur at 20:30.

   To select an interval, check its box. Fields display where you can edit the time and number of snapshots to keep. For example:

Schedule ❓	☐ Hourly
	☐ Daily
	☐ Weekly
	☑ **Monthly**
	Take snapshots every month at 01 ⬍ hour(s) 13 ⬍ minute(s) on 1 day of the month ❓
	Monthly snapshots to keep 1
	☐ Yearly

7. Specify whether **Alerts** should be generated for various state changes in the snapshot workflow. You can alert on failure, on start, on success, or when the snapshot workflow is aborted.
8. Click **Save Policy**.

   The new Policy displays on the **Snapshot Policies** page. See [Snapshot Policies Page](#) on page 531.

To edit or delete a snapshot policy:

1. Select **Backup** > **Snapshot Policies** in the top navigation bar.

   Existing snapshot policies are shown in a table. See [Snapshot Policies Page](#) on page 531.

2. Click the **Actions** menu shown next to a policy and select **Edit** or **Delete**.

Snapshot Policies Page

The policies you add are shown in a table on the **Snapshot Policies** screen. The table displays the following columns:

Table 33: Snapshot Policies

Column	Description
Policy Name	The name of the policy.
Cluster	The cluster that hosts the service (HDFS or HBase).
Service	The service from which the snapshot is taken.
Objects	HDFS Snapshots: The directories included in the snapshot. HBase Snapshots: The tables included in the snapshot.
Last Run	The date and time the snapshot last ran. Click the link to view the Snapshots History page. Also displays the status icon for the last run.
Snapshot Schedule	The type of schedule defined for the snapshot: Hourly, Daily, Weekly, Monthly, or Yearly.
Actions	A drop-down menu with the following options: • **Show History** - Opens the **Snapshots History** page. See Snapshots History on page 531. • **Edit Configuration** - Edit the snapshot policy. • **Delete** - Deletes the snapshot policy. • **Enable** - Enables running of scheduled snapshot jobs. • **Disable** - Disables running of scheduled snapshot jobs.

Snapshots History

The **Snapshots History** page displays information about Snapshot jobs that have been run or attempted. The page displays a table of Snapshot jobs with the following columns:

Table 34: Snapshots History

Column	Description
Start Time	Time when the snapshot job started execution. Click ❯ to display details about the snapshot. For example: Click the **View** link to open the **Managed scheduled snapshots Command** page, which displays details and messages about each step in the execution of the command. For example:

Column	Description
	Manage scheduled snapshots Command ✔ **Summary** Status: **Finished** Context: HBASE-1 Start Time: August 24, 2015 3:58 PM Duration: 20.93 seconds Successfully created/deleted snapshots as per snapshot policy ed2. [Download Result Data] **Details** Completed 1 of 1 step(s). ●All ○Failed Only ○Running Only Step Context Start Time Duration Actions ✔ Creates/deletes snapshots as per schedules defined in backup policies. Process scheduled-snapshots-HBASEee615eed (id=157) on host nightly-1.vpc.cloudera.com (id=1) exited with 0 and expected 0
Outcome	Displays whether the snapshot succeeded or failed.
Paths | Tables Processed	HDFS snapshots: the number of **Paths Processed** for the snapshot. HBase snapshots: the number of **Tables Processed** for the snapshot.
Paths | Tables Unprocessed	HDFS Snapshots: the number of **Paths Unprocessed** for the snapshot. HBase Snapshots: the number of **Tables Unprocessed** for the snapshot.
Snapshots Created	Number of snapshots created.
Snapshots Deleted	Number of snapshots deleted.
Errors During Creation	Displays a list of errors that occurred when creating the snapshot. Each error shows the related path and the error message.
Errors During Deletion	Displays a list of errors that occurred when deleting the snapshot. Each error shows the related path and the error message.

See Managing HDFS Snapshots on page 543 and Managing HBase Snapshots on page 533 for more information about managing snapshots.

Orphaned Snapshots

When a snapshot policy includes a limit on the number of snapshots to keep, Cloudera Manager checks the total number of stored snapshots each time a new snapshot is added, and automatically deletes the oldest existing snapshot if necessary. When a snapshot policy is edited or deleted, files, directories, or tables that were removed from the policy may leave "orphaned" snapshots behind that are not deleted automatically because they are no longer associated with a current snapshot policy. Cloudera Manager never selects these snapshots for automatic deletion because selection for deletion only occurs when the policy creates a *new* snapshot containing those files, directories, or tables.

You can delete snapshots manually through Cloudera Manager or by creating a command-line script that uses the HDFS or HBase snapshot commands. Orphaned snapshots can be hard to locate for manual deletion. Snapshot policies automatically receive the prefix `cm-auto` followed by a globally unique identifier (GUID). You can locate all snapshots for a specific policy by searching for t the prefix `cm-auto-guid` that is unique to that policy.

To avoid orphaned snapshots, delete snapshots before editing or deleting the associated snapshot policy, or record the identifying name for the snapshots you want to delete. This prefix is displayed in the summary of the policy in the policy list and appears in the delete dialog box. Recording the snapshot names, including the associated policy prefix,

is necessary because the prefix associated with a policy cannot be determined after the policy has been deleted, and snapshot names do not contain recognizable references to snapshot policies.

Managing HBase Snapshots

This page demonstrates how to manage HBase snapshots using either Cloudera Manager or the command line.

Managing HBase Snapshots Using Cloudera Manager

For HBase services, you can use the Table Browser tab to view the HBase tables associated with a service on your cluster. You can view the currently saved snapshots for your tables, and delete or restore them. From the HBase Table Browser tab, you can:

- View the HBase tables for which you can take snapshots.
- Initiate immediate (unscheduled) snapshots of a table.
- View the list of saved snapshots currently maintained. These can include one-off immediate snapshots, as well as scheduled policy-based snapshots.
- Delete a saved snapshot.
- Restore from a saved snapshot.
- Restore a table from a saved snapshot to a new table (Restore As).

Browsing HBase Tables

To browse the HBase tables to view snapshot activity:

1. From the **Clusters** tab, select your HBase service.
2. Go to the **Table Browser** tab.

Managing HBase Snapshots

Minimum Required Role: BDR Administrator (also provided by **Full Administrator**)

To take a snapshot:

1. Click a table.
2. Click **Take Snapshot**.
3. Specify the name of the snapshot, and click **Take Snapshot**.

To delete a snapshot, click [▾] and select **Delete**.

To restore a snapshot, click [▾] and select **Restore**.

> ⚠ **Warning:** If you use coprocessors, the coprocessor must be available on the destination cluster before restoring the snapshot.

To restore a snapshot to a new table, select **Restore As** from the menu associated with the snapshot, and provide a name for the new table.

> ⚠ **Warning:** If you "Restore As" to an existing table (that is, specify a table name that already exists), the existing table will be overwritten.

Storing HBase Snapshots on Amazon S3

HBase snapshots can be stored on the cloud storage service Amazon S3 instead of in HDFS.

> **Important:** When HBase snapshots are stored on, or restored from, Amazon S3, a MapReduce (MRv2) job is created to copy the HBase table data and metadata. The YARN service must be running on your Cloudera Manager cluster to use this feature.

To configure HBase to store snapshots on Amazon S3, you must have the following information:

- The *access key ID* for your Amazon S3 account.
- The *secret access key* for your Amazon S3 account.
- The path to the directory in Amazon S3 where you want your HBase snapshots to be stored.

You can improve the transfer of large snapshots to Amazon S3 by increasing the number of nodes due to throughput limitations of EC2 on a per node basis.

Configuring HBase in Cloudera Manager to Store Snapshots in Amazon S3

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Perform the following steps in Cloudera Manager:

1. Open the HBase service page.
2. Select **Scope** > **HBASE (Service-Wide)**.
3. Select **Category** > **Backup**.
4. Type `AWS` in the Search box.
5. Enter your Amazon S3 access key ID in the field **AWS S3 access key ID for remote snapshots**.
6. Enter your Amazon S3 secret access key in the field **AWS S3 secret access key for remote snapshots**.

> **Important:** If AWS S3 access keys are rotated, the Cloudera Manager server must be restarted.

7. Enter the path to the location in Amazon S3 where your HBase snapshots will be stored in the field **Amazon S3 Path for Remote Snapshots**.

> **Warning:** Do not use the Amazon S3 location defined by the path entered in **Amazon S3 Path for Remote Snapshots** for any other purpose, or directly add or delete content there. Doing so risks corrupting the metadata associated with the HBase snapshots stored there. Use this path and Amazon S3 location only through Cloudera Manager, and only for managing HBase snapshots.

8. In a terminal window, log in to your Cloudera Manager cluster at the command line and create a `/user/hbase` directory in HDFS. Change the owner of the directory to `hbase`. For example:

```
hdfs dfs -mkdir /user/hbase
hdfs dfs -chown hbase /user/hbase
```

> **Note:**
>
> Amazon S3 has default rate limitation per prefix per bucket. The throughput can be limited to 3500 requests per second. Consider to use different prefixes on S3 per table namespace, or table if any of the following applies:
>
> - large number of tables
> - tables with a large number of store files or regions
> - frequent snapshot policy

Configuring the Dynamic Resource Pool Used for Exporting and Importing Snapshots in Amazon S3

Dynamic resource pools are used to control the resources available for MapReduce jobs created for HBase snapshots on Amazon S3. By default, MapReduce jobs run against the default dynamic resource pool. To choose a different dynamic resource pool for HBase snapshots stored on Amazon S3, follow these steps:

1. Open the HBase service page.
2. Select **Scope** > **HBASE (Service-Wide)**.
3. Select **Category** > **Backup**.
4. Type `Scheduler` in the Search box.
5. Enter name of a dynamic resource pool in the **Scheduler pool for remote snapshots in AWS S3** property.
6. Click **Save Changes**.

HBase Snapshots on Amazon S3 with Kerberos Enabled

Starting with Cloudera Manager 5.8, YARN should by default allow the `hbase` user to run MapReduce jobs even when Kerberos is enabled. However, this change only applies to new Cloudera Manager deployments, and not if you have upgraded from a previous version to Cloudera Manager 5.8 (or higher).

If Kerberos is enabled on your cluster, and YARN does not allow the `hbase` user to run MapReduce jobs, perform the following steps:

1. Open the YARN service page in Cloudera Manager.
2. Select **Scope** > **NodeManager**.
3. Select **Category** > **Security**.
4. In the **Allowed System Users** property, click the **+** sign and add `hbase` to the list of allowed system users.
5. Click **Save Changes**.
6. Restart the YARN service.

Managing HBase Snapshots on Amazon S3 in Cloudera Manager

Minimum Required Role: BDR Administrator (also provided by **Full Administrator**)

To take HBase snapshots and store them on Amazon S3, perform the following steps:

1. On the HBase service page in Cloudera Manager, click the **Table Browser** tab.
2. Select a table in the Table Browser. If any recent local or remote snapshots already exist, they display on the right side.
3. In the dropdown for the selected table, click **Take Snapshot**.
4. Enter a name in the **Snapshot Name** field of the **Take Snapshot** dialog box.
5. If Amazon S3 storage is configured as described above, the **Take Snapshot** dialog box **Destination** section shows a choice of **Local** or **Remote S3**. Select **Remote S3**.
6. Click **Take Snapshot**.

   While the **Take Snapshot** command is running, a local copy of the snapshot with a name beginning `cm-tmp` followed by an auto-generated filename is displayed in the Table Browser. This local copy is deleted as soon as the remote snapshot has been stored in Amazon S3. If the command fails without being completed, the temporary local snapshot might not be deleted. This copy can be manually deleted or kept as a valid local snapshot. To store a current snapshot in Amazon S3, run the **Take Snapshot** command again, selecting **Remote S3** as the **Destination**, or use the HBase command-line tools to manually export the existing temporary local snapshot to Amazon S3.

Deleting HBase Snapshots from Amazon S3

To delete a snapshot stored in Amazon S3:

1. Select the snapshot in the Table Browser.
2. Click the dropdown arrow for the snapshot.
3. Click **Delete**.

Restoring an HBase Snapshot from Amazon S3

To restore an HBase snapshot that is stored in Amazon S3:

1. Select the table in the Table Browser.
2. Click **Restore Table**.
3. Choose **Remote S3** and select the table to restore.
4. Click **Restore**.

   Cloudera Manager creates a local copy of the remote snapshot with a name beginning with `cm-tmp` followed by an auto-generated filename, and uses that local copy to restore the table in HBase. Cloudera Manager then automatically deletes the local copy. If the **Restore** command fails without completing, the temporary copy might not be deleted and can be seen in the Table Browser. In that case, delete the local temporary copy manually and re-run the **Restore** command to restore the table from Amazon S3.

Restoring an HBase Snapshot from Amazon S3 with a New Name

By restoring an HBase snapshot stored in Amazon S3 with a new name, you clone the table without affecting the existing table in HBase. To do this, perform the following steps:

1. Select the table in the Table Browser.
2. Click **Restore Table From Snapshot As**.
3. In the **Restore As** dialog box, enter a new name for the table in the **Restore As** field.
4. Select **Remote S3** and choose the snapshot in the list of available Amazon S3 snapshots.

Managing Policies for HBase Snapshots in Amazon S3

You can configure policies to automatically create snapshots of HBase tables on an hourly, daily, weekly, monthly or yearly basis. Snapshot policies for HBase snapshots stored in Amazon S3 are configured using the same procedures as for local HBase snapshots. These procedures are described in Cloudera Manager Snapshot Policies. For snapshots stored in Amazon S3, you must also choose **Remote S3** in the **Destination** section of the policy management dialog boxes.

> **Note:** You can only configure a policy as **Local** or **Remote S3** at the time the policy is created and cannot change the setting later. If the setting is wrong, create a new policy.

When you create a snapshot based on a snapshot policy, a local copy of the snapshot is created with a name beginning with `cm-auto` followed by an auto-generated filename. The temporary copy of the snapshot is displayed in the Table Browser and is deleted as soon as the remote snapshot has been stored in Amazon S3. If the snapshot procedure fails without being completed, the temporary local snapshot might not be deleted. This copy can be manually deleted or kept as a valid local snapshot. To export the HBase snapshot to Amazon S3, use the HBase command-line tools to manually export the existing temporary local snapshot to Amazon S3.

Managing HBase Snapshots Using the Command Line

> **Important:**
> - Follow these command-line instructions on systems that do not use Cloudera Manager.
> - This information applies specifically to CDH 6.0.x. See Cloudera Documentation for information specific to other releases.

About HBase Snapshots

In previous HBase releases, the only way to a back up or to clone a table was to use `CopyTable` or `ExportTable`, or to copy all the `hfiles` in HDFS after disabling the table. These methods have disadvantages:

- `CopyTable` and `ExportTable` can degrade RegionServer performance.
- Disabling the table means no reads or writes; this is usually unacceptable.

HBase snapshots allow you to clone a table without making data copies, and with minimal impact on RegionServers. Exporting the table to another cluster does not have any impact on the RegionServers.

Use Cases

- Recovery from user or application errors

  - Useful because it may be some time before the database administrator notices the error.

    > **Note:**
    >
    > The database administrator needs to schedule the intervals at which to take and delete snapshots. Use a script or management tool; HBase does not have this functionality.

  - The database administrator may want to save a snapshot before a major application upgrade or change.

    > **Note:**
    >
    > Snapshots are not primarily used for system upgrade protection because they do not roll back binaries, and would not necessarily prevent bugs or errors in the system or the upgrade.

  - Recovery cases:

    - Roll back to previous snapshot and merge in reverted data.
    - View previous snapshots and selectively merge them into production.

- Backup

  - Capture a copy of the database and store it outside HBase for disaster recovery.
  - Capture previous versions of data for compliance, regulation, and archiving.
  - Export from a snapshot on a live system provides a more consistent view of HBase than `CopyTable` and `ExportTable`.

- Audit or report view of data at a specific time

  - Capture monthly data for compliance.
  - Use for end-of-day/month/quarter reports.

- Application testing

  - Test schema or application changes on similar production data from a snapshot and then discard.

    For example:

    1. Take a snapshot.
    2. Create a new table from the snapshot content (schema and data)
    3. Manipulate the new table by changing the schema, adding and removing rows, and so on. The original table, the snapshot, and the new table remain independent of each other.

- Offload work

  - Capture, copy, and restore data to another site
  - Export data to another cluster

Where Snapshots Are Stored

Snapshot metadata is stored in the `.hbase_snapshot` directory under the `hbase` root directory (`/hbase/.hbase-snapshot`). Each snapshot has its own directory that includes all the references to the `hfiles`, logs, and metadata needed to restore the table.

`hfiles` required by the snapshot are in the
`/hbase/data/<namespace>/<tableName>/<regionName>/<familyName>/` location if the table is still using
them; otherwise, they are in `/hbase/.archive/<namespace>/<tableName>/<regionName>/<familyName>/`.

Zero-Copy Restore and Clone Table

From a snapshot, you can create a new table (`clone` operation) or restore the original table. These two operations do
not involve data copies; instead, a link is created to point to the original `hfiles`.

Changes to a cloned or restored table do not affect the snapshot or (in case of a clone) the original table.

To clone a table to another cluster, you export the snapshot to the other cluster and then run the `clone` operation;
see [Exporting a Snapshot to Another Cluster](#).

Reverting to a Previous HBase Version

Snapshots do not affect HBase backward compatibility if they are not used.

If you use snapshots, backward compatibility is affected as follows:

- If you only take snapshots, you can still revert to a previous HBase version.
- If you use `restore` or `clone`, you cannot revert to a previous version unless the cloned or restored tables have
  no links. Links cannot be detected automatically; you would need to inspect the file system manually.

Storage Considerations

Because `hfiles` are immutable, a snapshot consists of a reference to the files in the table at the moment the snapshot
is taken. No copies of the data are made during the snapshot operation, but copies may be made when a compaction
or deletion is triggered. In this case, if a snapshot has a reference to the files to be removed, the files are moved to an
archive folder, instead of being deleted. This allows the snapshot to be restored in full.

Because no copies are performed, multiple snapshots share the same `hfiles`, butfor tables with lots of updates, and
compactions, each snapshot could have a different set of `hfiles`.

Configuring and Enabling Snapshots

Snapshots are on by default; to disable them, set the `hbase.snapshot.enabled` property in `hbase-site.xml` to
`false`:

```
<property>
    <name>hbase.snapshot.enabled</name>
    <value>
        false
    </value>
</property>
```

To enable snapshots after you have disabled them, set `hbase.snapshot.enabled` to `true`.

> **Note:**
>
> If you have taken snapshots and then decide to disable snapshots, you must delete the snapshots
> before restarting the HBase master; the HBase master will not start if snapshots are disabled and
> snapshots exist.

Snapshots do not affect HBase performance if they are not used.

Shell Commands

You can manage snapshots by using the HBase shell or the HBaseAdmin Java API.

The following table shows actions you can take from the shell.

Action	Shell command	Comments
Take a snapshot of `tableX` called `snapshotX`	`snapshot 'tableX', 'snapshotX'`	Snapshots can be taken while a table is disabled, or while a table is online and serving traffic. • If a table is disabled (using `disable <table>`), an offline snapshot is taken. This snapshot is managed by the master and fully consistent with the state when the table was disabled. This is the simplest and safest method, but it involves a service interruption because the table must be disabled to take the snapshot. • In an online snapshot, the table remains available while the snapshot is taken, and incurs minimal performance degradation of normal read/write loads. This snapshot is managed by the master and run on the RegionServers. The current implementation—simple-flush snapshots—provides no causal consistency guarantees. Despite this shortcoming, it offers the same degree of consistency as `CopyTable` and is a significant improvement.
Restore snapshot `snapshotX` (replaces the source table content)	`restore_snapshot 'snapshotX'`	For emergency use only; see [Restrictions](#). Restoring a snapshot replaces the current version of a table with different version. To run this command, you must disable the target table. The `restore` command takes a snapshot of the table (appending a timestamp code), and then clones data into the original data and removes data not in the snapshot. If the operation succeeds, the target table is enabled. ⚠️ **Warning:** If you use coprocessors, the coprocessor must be available on the destination cluster before restoring the snapshot.
List all available snapshots	`list_snapshots`	
List all available snapshots starting with `'mysnapshot_'` (regular expression)	`list_snapshots 'my_snapshot_.*'`	
Remove a snapshot called `snapshotX`	`delete_snapshot 'snapshotX'`	
Create a new table `tableY` from a snapshot `snapshotX`	`clone_snapshot 'snapshotX', 'tableY'`	Cloning a snapshot creates a new read/write table that serves the data kept at the time of the snapshot. The original table and the cloned table can be modified independently; new data written to one table does not show up on the other.

Taking a Snapshot Using a Shell Script

You can take a snapshot using an operating system shell script, such as a Bash script, in HBase Shell noninteractive mode, which is described in [Accessing HBase by using the HBase Shell](). This example Bash script shows how to take a snapshot in this way. This script is provided as an illustration only; do not use in production.

```
#!/bin/bash
# Take a snapshot of the table passed as an argument
# Usage: snapshot_script.sh table_name
# Names the snapshot in the format snapshot-YYYYMMDD

# Parse the arguments
if [ -z $1 ]||[$1 == '-h' ]; then
 echo "Usage: $0 &lt;table&gt;"
 echo "       $0 -h"
 exit 1
fi

# Modify to suit your environment
export HBASE_PATH=/home/user/hbase
export DATE=`date +"%Y%m%d"`
echo "snapshot '$1', 'snapshot-$DATE'" | $HBASE_PATH/bin/hbase shell -n
status=$?
if [$status -ne 0]; then
 echo "Snapshot may have failed: $status"
fi
exit $status
```

HBase Shell returns an exit code of 0 on successA non-zero exit code indicates the possibility of failure, not a definite failure. Your script should check to see if the snapshot was created before taking the snapshot again, in the event of a reported failure.

Exporting a Snapshot to Another Cluster

You can export any snapshot from one cluster to another. Exporting the snapshot copies the table's hfiles, logs, and the snapshot metadata, from the source cluster to the destination cluster. Specify the `-copy-from` option to copy from a remote cluster to the local cluster or another remote cluster. If you do not specify the `-copy-from` option, the `hbase.rootdir` in the HBase configuration is used, which means that you are exporting from the current cluster. You must specify the `-copy-to` option, to specify the destination cluster.

> **Note:** Snapshots must be enabled on the destination cluster. See [Configuring and Enabling Snapshots]() on page 538.

> **Warning:** If you use coprocessors, the coprocessor must be available on the destination cluster before restoring the snapshot.

The `ExportSnapshot` tool executes a MapReduce Job similar to `distcp` to copy files to the other cluster. It works at file-system level, so the HBase cluster can be offline.

Run `ExportSnapshot` as the `hbase` user or the user that owns the files. If the user, group, or permissions need to be different on the destination cluster than the source cluster, use the `-chuser`, `-chgroup`, or `-chmod` options as in the second example below, or be sure the destination directory has the correct permissions. In the following examples, replace the HDFS server path and port with the appropriate ones for your cluster.

To copy a snapshot called MySnapshot to an HBase cluster srv2 (hdfs://srv2:8020/hbase) using 16 mappers:

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot MySnapshot -copy-to
hdfs://srv2:<hdfs_port>/hbase -mappers 16
```

To export the snapshot and change the ownership of the files during the copy:

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot MySnapshot -copy-to
hdfs://srv2:<hdfs_port>/hbase -chuser MyUser -chgroup MyGroup -chmod 700 -mappers 16
```

You can also use the Java `-D` option in many tools to specify MapReduce or other configuration properties. For example, the following command copies `MY_SNAPSHOT` to `hdfs://cluster2/hbase` using groups of 10 hfiles per mapper:

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot
-Dsnapshot.export.default.map.group=10 -snapshot MY_SNAPSHOT -copy-to
hdfs://cluster2/hbase
```

(The number of mappers is calculated as `TotalNumberOfHFiles`/10.)

To export from one remote cluster to another remote cluster, specify both `-copy-from` and `-copy-to` parameters.

You can then reverse the direction to restore the snapshot back to the first remote cluster.

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot snapshot-test -copy-from
 hdfs://machine1/hbase -copy-to hdfs://machine2/my-backup
```

To specify a different name for the snapshot on the target cluster, use the `-target` option.

```
hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot snapshot-test -copy-from
 hdfs://machine1/hbase -copy-to hdfs://machine2/my-backup -target new-snapshot
```

Restrictions

> ⚠️ **Warning:**
>
> **Do not use merge in combination with snapshots. Merging two regions can cause data loss if snapshots or cloned tables exist for this table.**
>
> The merge is likely to corrupt the snapshot and any tables cloned from the snapshot. If the table has been restored from a snapshot, the merge may also corrupt the table. The snapshot may survive intact if the regions being merged are not in the snapshot, and clones may survive if they do not share files with the original table or snapshot. You can use the `Snapinfo` tool (see Information and Debugging on page 542) to check the status of the snapshot. If the status is `BROKEN`, the snapshot is unusable.

- If you have enabled the `AccessController Coprocessor` for HBase, only a global administrator can take, clone, or restore a snapshot, and these actions do not capture the ACL rights. This means that restoring a table preserves the ACL rights of the existing table, and cloning a table creates a new table that has no ACL rights until the administrator adds them.
- Do not take, clone, or restore a snapshot during a rolling restart. Snapshots require RegionServers to be up; otherwise, the snapshot fails.

> 📝 **Note:** This restriction also applies to a rolling upgrade, which can be done only through Cloudera Manager.

If you are using HBase Replication and you need to restore a snapshot:

> ❗ **Important:**
>
> Snapshot restore is an emergency tool; you need to disable the table and table replication to get to an earlier state, and you may lose data in the process.

If you are using HBase Replication, the replicas will be out of sync when you restore a snapshot. If you need to restore a snapshot, proceed as follows:

1. Disable the table that is the restore target, and stop the replication.
2. Remove the table from both the master and worker clusters.
3. Restore the snapshot on the master cluster.
4. Create the table on the worker cluster and use `CopyTable` to initialize it.

> **Note:**
>
> If this is not an emergency (for example, if you know exactly which rows you have lost), you can create a clone from the snapshot and create a MapReduce job to copy the data that you have lost.
>
> In this case, you do not need to stop replication or disable your main table.

Snapshot Failures

Region moves, splits, and other metadata actions that happen while a snapshot is in progress can cause the snapshot to fail. The software detects and rejects corrupted snapshot attempts.

Information and Debugging

You can use the `SnapshotInfo` tool to get information about a snapshot, including status, files, disk usage, and debugging information.

Examples:

Use the `-h` option to print usage instructions for the `SnapshotInfo` utility.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -h
Usage: bin/hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo [options]
 where [options] are:
  -h|-help               Show this help and exit.
  -remote-dir            Root directory that contains the snapshots.
  -list-snapshots        List all the available snapshots and exit.
  -snapshot NAME         Snapshot to examine.
  -files                 Files and logs list.
  -stats                 Files and logs stats.
  -schema                Describe the snapshotted table.
```

Use the `-list-snapshots` option to list all snapshots and exit.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -list-snapshots
SNAPSHOT                | CREATION TIME        | TABLE NAME
snapshot-test           | 2014-06-24T19:02:54  | test
```

Use the `-remote-dir` option with the `-list-snapshots` option to list snapshots located on a remote system.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -remote-dir
s3a://mybucket/mysnapshot-dir -list-snapshots
SNAPSHOT  |    CREATION TIME  | TABLE NAME
snapshot-test          2014-05-01 10:30     myTable
```

Use the `-snapshot` option to print information about a specific snapshot.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -snapshot test-snapshot
Snapshot Info
----------------------------------------
   Name: test-snapshot
   Type: DISABLED
  Table: test-table
Version: 0
Created: 2012-12-30T11:21:21
*************************************************************
```

Use the `-snapshot` with the `-stats` options to display additional statistics about a snapshot.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -stats -snapshot snapshot-test
Snapshot Info
----------------------------------------
   Name: snapshot-test
   Type: FLUSH
  Table: test
 Format: 0
Created: 2014-06-24T19:02:54


1 HFiles (0 in archive), total size 1.0k (100.00% 1.0k shared with the source table)
```

Use the `-schema` option with the `-snapshot` option to display the schema of a snapshot.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo  -schema -snapshot snapshot-test
Snapshot Info
----------------------------------------
   Name: snapshot-test
   Type: FLUSH
  Table: test
 Format: 0
Created: 2014-06-24T19:02:54

Table Descriptor
----------------------------------------
'test', {NAME => 'cf', DATA_BLOCK_ENCODING => 'FAST_DIFF', BLOOMFILTER => 'ROW',
REPLICATION_SCOPE => '0',
COMPRESSION => 'GZ', VERSIONS => '1', TTL => 'FOREVER', MIN_VERSIONS => '0',
KEEP_DELETED_CELLS => 'false',
BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
```

Use the `–files` option with the `-snapshot` option to list information about files contained in a snapshot.

```
$ hbase org.apache.hadoop.hbase.snapshot.SnapshotInfo -snapshot test-snapshot –files
Snapshot Info
----------------------------------------
   Name: test-snapshot
   Type: DISABLED
  Table: test-table
Version: 0
Created: 2012-12-30T11:21:21

Snapshot Files
----------------------------------------
  52.4k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/bdf29c39da2a4f2b81889eb4f7b18107
 (archive)
  52.4k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/1e06029d0a2a4a709051b417aec88291
 (archive)
  86.8k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/506f601e14dc4c74a058be5843b99577
 (archive)
  52.4k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/5c7f6916ab724eacbcea218a713941c4
 (archive)
 293.4k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/aec5e33a6564441d9bd423e31fc93abb
 (archive)
  52.4k test-table/02ba3a0f8964669520cf96bb4e314c60/cf/97782b2fbf0743edaacd8fef06ba51e4
 (archive)

6 HFiles (6 in archive), total size 589.7k (0.00% 0.0 shared with the source table)
0 Logs, total size 0.0
```

Managing HDFS Snapshots

This topic demonstrates how to manage HDFS snapshots using either Cloudera Manager or the command line.

For HDFS services, use the File Browser tab to view the HDFS directories associated with a service on your cluster. You can view the currently saved snapshots for your files, and delete or restore them. From the HDFS File Browser tab, you can:

- Designate HDFS directories to be "snapshottable" so snapshots can be created for those directories.
- Initiate immediate (unscheduled) snapshots of a HDFS directory.
- View the list of saved snapshots currently being maintained. These can include one-off immediate snapshots, as well as scheduled policy-based snapshots.
- Delete a saved snapshot.
- Restore an HDFS directory or file from a saved snapshot.
- Restore an HDFS directory or file from a saved snapshot to a new directory or file (Restore As).

Before using snapshots, note the following limitations:

- Snapshots that include encrypted directories cannot be restored outside of the zone within which they were created.
- The Cloudera Manager Admin Console cannot perform snapshot operations (such as create, restore, and delete) for HDFS paths with encryption-at-rest enabled. This limitation only affects the Cloudera Manager Admin Console and does not affect CDH command-line tools or actions not performed by the Admin Console, such as BDR replication which uses command-line tools. For more information about snapshot operations, see the Apache HDFS snapshots documentation.

Browsing HDFS Directories

To browse the HDFS directories to view snapshot activity:

1. From the **Clusters** tab, select your CDH HDFS service.
2. Go to the **File Browser** tab.

As you browse the directory structure of your HDFS, basic information about the directory you have selected is shown at the right (owner, group, and so on).

Enabling and Disabling HDFS Snapshots

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

For snapshots to be created, HDFS directories must be enabled for snapshots. You cannot specify a directory as part of a snapshot policy unless it has been enabled for snapshots.

Enabling an HDFS Directory for Snapshots

1. From the **Clusters** tab, select your CDH HDFS service.
2. Go to the **File Browser** tab.
3. Go to the directory you want to enable for snapshots.
4. In the File Browser, click the drop-down menu next to the full file path and select **Enable Snapshots**:

> **Note:** Once you enable snapshots for a directory, you cannot enable snapshots on any of its subdirectories. Snapshots can be taken only on directories that have snapshots enabled.

Disabling a Snapshottable Directory

To disable snapshots for a directory that has snapshots enabled, use **Disable Snapshots** from the drop-down menu button at the upper right. If snapshots of the directory exist, they must be deleted before snapshots can be disabled.

Taking and Deleting HDFS Snapshots

Minimum Required Role: Full Administrator

To manage HDFS snapshots, first enable an HDFS directory for snapshots.

Taking Snapshots

> **Note:** You can also schedule snapshots to occur regularly by creating a Snapshot Policy.

1. From the **Clusters** tab, select your CDH HDFS service.
2. Go to the **File Browser** tab.
3. Go to the directory with the snapshot you want to restore.
4. Click the drop-down menu next to the full path name and select **Take Snapshot**.

   The **Take Snapshot** screen displays.

5. Enter a name for the snapshot.
6. Click **OK**.

   The **Take Snapshot** button is present, enabling an immediate snapshot of the directory.

7. To take a snapshot, click **Take Snapshot**, specify the name of the snapshot, and click **Take Snapshot**. The snapshot is added to the snapshot list.

   Any snapshots that have been taken are listed by the time at which they were taken, along with their names and a menu button.

Deleting Snapshots

1. From the **Clusters** tab, select your CDH HDFS service.
2. Go to the **File Browser** tab.
3. Go to the directory with the snapshot you want to delete.
4. In the list of snapshots, locate the snapshot you want to delete and click ⏷.
5. Select **Delete**.

Restoring Snapshots

Before you restore from a snapshot, ensure that there is adequate disk space.

1. From the **Clusters** tab, select your CDH HDFS service.
2. Go to the **File Browser** tab.
3. Go to the directory you want to restore.
4. In the File Browser, click the drop-down menu next to the full file path (to the right of the file browser listings) and select one of the following:

   - **Restore Directory From Snapshot**
   - **Restore Directory From Snapshot As...**

   The **Restore Snapshot** screen displays.

5. If you selected **Restore Directory From Snapshot As...**, enter the username to apply when restoring the snapshot.
6. Select one of the following:

   - **Use HDFS 'copy' command** - This option executes more slowly and does not require credentials in a secure cluster. It copies the contents of the snapshot as a subdirectory or as files within the target directory.
   - **Use DistCp / MapReduce** - This options executes more quickly and requires credentials (Run As) in secure clusters. It merges the target directory with the contents of the source snapshot. When you select this option,

the following additional fields, which are similar to those available when configuring a replication, display under **More Options**:

- **MapReduce Service** - The MapReduce or YARN service to use.
- **Scheduler Pool** – (Optional) Enter the name of a resource pool in the field. The value you enter is used by the **MapReduce Service** you specified when Cloudera Manager executes the MapReduce job for the replication. The job specifies the value using one of these properties:

  - MapReduce – Fair scheduler: `mapred.fairscheduler.pool`
  - MapReduce – Capacity scheduler: `queue.name`
  - YARN – `mapreduce.job.queuename`

- Enter the user to run the replication job in the **Run As Username** field. By default this is `hdfs`. If you want to run the job as a different user, enter the user name here. If you are using Kerberos, you *must* provide a user name here, and it must be one with an ID greater than 1000. (You can also configure the minimum user ID number with the **min.user.id** property in the YARN or MapReduce service.) Verify that the user running the job has a home directory, `/user/`*username*, owned by *username*`:supergroup` in HDFS. This user must have permissions to read from the source directory and write to the destination directory.

  Note the following:

  - The User must not be present in the list of banned users specified with the **Banned System Users** property in the YARN configuration (Go to the YARN service, select **Configuration** tab and search for the property). For security purposes, the `hdfs` user is banned by default from running YARN containers.
  - The requirement for a user ID that is greater than 1000 can be overridden by adding the user to the "white list" of users that is specified with the **Allowed System Users** property. (Go to the YARN service, select **Configuration** tab and search for the property.)

- **Log path** - An alternate path for the logs.
- **Maximum Map Slots** - Limits for the number of map slots per mapper. The default value is 20.
- **Abort on Error** - Whether to abort the job on an error. If selected, files copied up to that point remain on the destination, but no additional files are copied. **Abort on Error** is off by default.
- **Skip Checksum Checks** - Whether to skip checksum checks (the default is to perform them). If checked, checksum validation will not be performed.

  You must select the this property to prevent failure when restoring snapshots in the following cases:

  - Restoring a snapshot within a single encryption zone.
  - Restoring a snapshot from one encryption zone to a different encryption zone.
  - Restoring a snapshot from an unencrypted zone to an encrypted zone.

  See [HDFS Transparent Encryption](#).

- **Delete Policy** - Whether files that were deleted on the source should also be deleted from the destination directory. This policy also determines the handling of files in the destination location that are unrelated to the source. Options include:

  - **Keep Deleted Files** - Retains the destination files even when they no longer exist at the source. (This is the default.).
  - **Delete to Trash** - If the HDFS trash is enabled, files are moved to the trash folder. (Not supported when replicating to Amazon S3.)
  - **Delete Permanently** - Uses the least amount of space; use with caution.

- **Preserve** - Whether to preserve the block size, replication count, permissions (including ACLs), and extended attributes (XAttrs) as they exist on the source file system, or to use the settings as configured on the destination file system. By default source system settings are preserved. When **Permission** is checked, and both the source and destination clusters support ACLs, replication preserves ACLs. Otherwise, ACLs are not replicated. When **Extended attributes** is checked, and both the source and destination

clusters support extended attributes, replication preserves them. (This option only displays when both source and destination clusters support extended attributes.)

If you select one or more of the **Preserve** options and you are replicating *to* Amazon S3, the values all of these items are saved in meta data files on S3. When you replicate *from* Amazon S3 to HDFS, you can select which of these options you want to preserve.

> **Note:** To preserve permissions to HDFS, you must be running as a superuser on the *destination* cluster. Use the "Run As Username" option to ensure that is the case.

See [Replication of Encrypted Data](#) on page 517 and [HDFS Transparent Encryption](#).

BDR Tutorials

Cloudera Backup and Disaster Recovery (BDR) is available with a Cloudera Enterprise license. Enterprise BDR lets you replicate data from one cluster to another, or from one directory path to another on the same or on a different cluster. In case of data loss, the backup replica can be used to restore data to the production cluster.

The time to start thinking about how to restore data is long before you might ever need to do so. These BDR tutorials take you step-by-step through the process of backing up an example production cluster. The example backup replication schedules are for one-time replication that makes a backup copy of Hive datasets or of HDFS files, respectively, on another cluster designated as a backup cluster.

The restore processes detailed in each tutorial also take you step-by-step through the process of restoring data using two different general approaches:

- [How To Back Up and Restore Apache Hive Data Using Cloudera Enterprise BDR](#) on page 547 highlights a *one-off* data recovery scenario in which you create the replication schedule immediately after a data loss and use it to restore data.
- [How To Back Up and Restore HDFS Data Using Cloudera Enterprise BDR](#) on page 559 shows you how pre-configure replication schedules so they are available when needed.

Use either or both of these tutorials to help plan your own backup and restore strategy.

How To Back Up and Restore Apache Hive Data Using Cloudera Enterprise BDR

Cloudera Enterprise Backup and Disaster Recovery (BDR) uses replication schedules to copy data from one cluster to another, enabling the second cluster to provide a backup for the first.

This tutorial shows you how to configure replication schedules to back up Apache Hive data and to restore data from the backup cluster when needed.

Creating replication schedules for backup and restore requires:

- A license for Cloudera Enterprise. Cloudera Enterprise BDR is available from the **Backup** menu of Cloudera Manager Admin Console when licensed for Enterprise.
- The **BDR Administrator** or **Full Administrator** role on the clusters involved (typically, a production cluster and a backup cluster).

Best Practices for Back Up and Restore

When configuring replication schedules for Hive back up and restore, follow these guidelines:

- Make sure that the time-frames configured for replication schedules allow the replication process to complete.
- Test your replication schedules for both back up and restore before relying on them in a production environment. To test a restore replication schedule in a production environment, use a different HDFS destination path for the Hive data files than that used for the replica.

- Enable only one replication schedule for the same dataset at the same time. That means you must first disable the backup replication schedule before enabling or creating a restore replication schedule for the same dataset, and vice versa.
- Enable snapshots on the HDFS file system containing the Hive data files. This ensures consistency if changes are still being made during the replication process. See Using Snapshots with Replication on page 513 for details.

About the Example Clusters

This guide uses the two example clusters listed in the following table:

Production cluster	Backup cluster
http://prod-db-example-1.vpc.cloudera.com	http://backup-example-1.vpc.cloudera.com
Source cluster for a backup replication schedule.	Destination cluster for backup replication schedule.
Destination cluster for a restore replication schedule.	Source cluster for a restore replication schedule.
To restore data from a backup cluster, set peer relationship to backup cluster and configure replication schedule from this cluster.	To back up a production cluster, set peer relationship and configure replication schedule from this cluster.

The example clusters are not configured to use Kerberos, nor do they use external accounts for cloud storage on Amazon Web Services (AWS). The name of the example production and the example backup cluster have each been changed from the default "Cluster 1" name to *Production DB (Main)* and *Offsite-Backup*, respectively.

The example production cluster contains the Hive *default* database and an example database, *us_fda_fea*, which contains data extracted from the US federal government's open data initiative at data.gov. The us_fda_fea database contains three tables as shown in Hue Web UI:

As shown in the screenshot below, snapshots have been configured and enabled on the HDFS system path containing the Hive database files. Using snapshots is one of the Best Practices for Back Up and Restore on page 547. See Using Snapshots with Replication on page 513 for more information.

The example backup cluster (*Offsite-Backup*) has not yet been used for a backup yet, so the Hive default path is empty as shown below:

Backing Up and Restoring Hive Data

This tutorial steps through these two major tasks:

Backup and restore are each configured and managed using Replication Schedules, available from the Backup menu on Cloudera Manager Admin Console:

> **Note:** Screenshots in this guide show version 5.9 of the Cloudera Manager Admin Console.

The backup and restore processes are configured, managed, and executed using replication schedules. Each replication schedule identifies a *source* and a *destination* for the given replication process. The replication process uses a *pull* model. When the replication process runs, the configured destination cluster accesses the given source cluster and transparently performs all tasks needed to recreate the Hive database and tables on the destination cluster.

The destination cluster handles configuration and running the schedule. Typically, creating a backup replication schedule takes place on the backup cluster and creating a restore replication schedule takes place on the production cluster. Thus, as shown in this tutorial, the example production cluster, *Production DB (Main)*, is the *source* for the backup replication schedule and the *destination* for the restore replication schedule.

Creating a Backup

Defining the backup replication schedule starts from the Cloudera Manager Admin Console on the destination cluster. For this example, the *destination* cluster is the cluster being used as the backup and the *source* is the example production cluster. To create the backup, follow these steps:

- Step 1: Establish a Peer Relationship to the Production Cluster on page 551
- Step 2: Configure the Replication Schedule on page 552
- Step 3: Verify Successful Replication on page 553

Step 1: Establish a Peer Relationship to the Production Cluster

You must have the BDR Administrator or Full Administrator role on both clusters to define a Peer relationship and perform all subsequent steps.

1. Log in to Cloudera Manager Admin Console on the master node of the backup cluster.
2. Click the **Backup** tab and select **Peers** from the menu.
3. On the Peers page, click **Add Peers**.
4. On the Add Peer page:

   a. **Peer Name** - Enter a meaningful name for the cluster that you want to back up, such as *Production DB*. This peer name becomes available in the next step, to be selected as the source for the replication.

   b. **Peer URL** - Enter the URL (including port number) for the Cloudera Manager Admin Console running on the master node of the cluster.

   c. **Peer Admin Username** - Enter the name of the administrator account for the cluster.

   d. **Peer Admin Password** - Enter the password for the administrator account.

5. Click **Add Peer** to save your settings, connect to the production cluster, and establish this peer relationship.

The Peers page re-displays, showing the Status column as Connected (note the green check-mark):

You can now create a schedule to replicate Hive files from production to the backup cluster.

Step 2: Configure the Replication Schedule

From the Cloudera Manager Admin Console on the backup cluster:

1. Click the **Backup** tab and select **Replication Schedules** from the menu.
2. On the Replication Schedules page, select **Create Schedule** > **Hive Replication**.
3. On the Create Hive Replication page, click the General tab to display the default schedule options:

   a. **Source** - Make sure the Hive node selected in the drop-down is the production cluster (the cluster to be backed up).

   b. **Destination** - This is the cluster you are logged into, the backup cluster. Select the Hive node on the cluster.

   c. **Databases** - Select Replicate All to re-create all Hive databases from the production system to the backup. Or deselect Replicate All and enter specific database name and tables to back up select databases or tables only.

   d. **Schedule** - Immediate. For production environments, change this to Recurring and set an appropriate time-frame that can backup the selected dataset completely. For example, do not set an hourly schedule if it takes two hours to back up the dataset.

   e. **Run As Username** - Leave as Default.

   f. **Scheduler Pool** - Leave as Default.

4. Click Save Schedule.

The files are replicated from the source cluster to the backup cluster immediately.

> **Note:** When you configure a replication schedule to back up Hive data on a regular basis, make sure that the schedule allows for each backup to complete. For example, do not create a schedule to back up every hour if it takes two hours to complete a full backup.

When the process completes, the Replication Schedules page re-displays, showing a green check-mark and time-stamp in the Last Run column:

When you set up your own schedules in your actual production environment, the Next Run column will likely also contain a date and time according to your specifications for the schedule.

Step 3: Verify Successful Replication

You can verify that data has been replicated by using Hive commands, the HDFS File Browser, or the Hue Web UI (shown below):

The Hive database is now on both the production and the backup clusters—the source and destination of the backup Replication Schedule, respectively.

At this point if the production cluster has a catastrophic data loss, you can use the backup replica to restore the database to the production cluster.

For example, assume that the us_fda_fea database was inadvertently deleted from the example production cluster as shown in the Hue Web UI:

Whenever you first discover an issue (data loss, corruption) with production data, *immediately disable* any existing backup replication schedules. Disabling the backup replication schedule prevents corrupt or missing data from being replicated over an existing backup replica is why it is the first step in the restore process detailed in the next set of steps.

Restoring Data from the Backup Cluster

Restoring data from a backup cluster takes place on the production cluster but requires that the backup replication schedule is first disabled. The process includes these steps:

Step 1: Disable Backup Replication Schedule

At the Cloudera Manager Admin Console on the backup cluster:

1. Select **Backup** > **Replication Schedules**.
2. On the **Replication Schedules** page, select the schedule.
3. From the **Actions** drop-down menu, select **Disable**.

When the Replication Schedules pages refreshes, the word Disabled displays in the Next Run column for the schedule.

With the backup replication schedule temporarily disabled, move to the production cluster to create and run the replication schedule to restore the data as detailed in the remaining steps.

Step 2: Establish a Peer Relationship to the Backup Cluster

Log in to Cloudera Manager Admin Console on the master node of the production cluster.

1. Click the **Backup** tab and select **Peers** from the menu.
2. On the **Peers** page, click **Add Peers** button.
3. On the **Add Peer** page:
   a. **Peer Name** - Enter a meaningful name for the cluster from which to obtain the backup data.
   b. **Peer URL** - Enter the URL for the Cloudera Manager Admin Console (running on the master node of the cluster).
   c. **Peer Admin Username** - Enter the administrator user name for the backup cluster.

4. Click **Add Peer** to save your settings. The production cluster connects to the backup cluster, establishes the peer relationship, and tests the connection.

The Peers page redisplays and lists the peer name, URL, and shows its Status (Connected) as shown below:

Step 3: Configure the Restore Replication Schedule

From the Cloudera Manager Admin Console on the production cluster:

1. Click the **Backup** tab and select **Replication Schedules** from the menu.
2. On the **Replication Schedules** page, select **Create Schedule** > **Hive Replication**.
3. On the General settings tab of the Create Hive Replication page:

   a. **Source** - The backup cluster from which to pull the data.
   b. **Destination** - The production cluster that needs the data restored.
   c. **Databases** - Select Replicate All.
   d. **Schedule** - Immediate.
   e. **Run As Username** - Leave as Default.
   f. **Scheduler Pool** - Leave as Default.

   > **Important:** Be sure that the *source* is the cluster where your backup is stored and the *destination* is the cluster containing lost or damaged data that you want to replace with the backup.

4. Click the **Advanced** tab:

5. Select **Force Overwrite** so that the backup cluster's metadata replaces the metadata on the production cluster. The assumption is that the production cluster's dataset has been corrupted.

> ❗ **Important:** The Force Overwrite setting can destroy tables or entries created after the backup completed. Do not use this setting unless you are certain that you want to overwrite the destination cluster with data from the source.

6. Click **Save Schedule**.

This schedule runs one time and restores your Hive databases. When the process completes, the Replication Schedules page displays the time-stamp and check-mark in the Last Run column for the schedule.

Step 4: Disable the Replication Schedule

Immediately disable the Replication Schedule used for the restore as soon as it completes. From the Replication Schedules page:

1. Select the replication schedule that just completed.
2. From the **Actions** drop-down menu, select **Disable**.

The page refreshes and displays **Disabled** in the Next Run column.

Step 5: Verify Successful Replication

Use the Hive command-line, the HDFS File Browser, or the Hue Web UI to verify successful data restore:

The restore process is complete. In a production environment, assuming the restored Hive database and tables are as you want them in a temporary path, you can re-configure the replication schedule to restore the data to the original path.

Step 6: Re-enable the Backup Replication Schedule

On the backup cluster, log in to the Cloudera Manager Admin Console.

1. Select **Backup** > **Replication Schedules**.
2. On the Replication Schedules page, select the schedule created at the beginning of this process.
3. From the **Actions** drop-down menu, select **Enable**.

The restore process is complete.

In actual production environments, create replication schedules that regularly back up your production clusters. To restore data, create replication schedules as shown in this tutorial.

Alternatively, you can define replication schedules in advance but leave them disabled. See How To Back Up and Restore HDFS Data Using Cloudera Enterprise BDR on page 559 for details.

See Backup and Disaster Recovery on page 483 and BDR Tutorials on page 547 or more information about Cloudera Enterprise BDR.

How To Back Up and Restore HDFS Data Using Cloudera Enterprise BDR

Cloudera Enterprise Backup and Disaster Recovery (BDR) uses replication schedules to copy data from one cluster to another, enabling the second cluster to provide a backup for the first. In case of any data loss, the second cluster—the backup—can be used to restore data to production.

This tutorial shows you how to create a replication schedule to copy HDFS data from one cluster to another for a backup, and how to create and test a replication schedule that you can use to restore data when needed in the future.

Creating replication schedules for backup and restore requires:

- A license for Cloudera Enterprise. Cloudera Enterprise BDR is available from the **Backup** menu of Cloudera Manager Admin Console when licensed for Enterprise.
- The **BDR Administrator** or **Full Administrator** role on the clusters involved (typically, a production cluster and a backup cluster).

Best Practices for Back Up and Restore

When configuring replication schedules for HDFS back up and restore, follow these guidelines:

- Make sure that the time-frames configured for replication schedules allow for the replication process to complete.
- Create a restore replication schedule in advance but leave it disabled, as shown in this tutorial.
- Test your replication schedules for both back up and restore before relying on them in a production environment.
- Enable only one replication schedule for the same dataset at the same time. That means you must first disable the backup replication schedule before enabling or creating a restore replication schedule for the same dataset, and vice versa.
- Enable snapshots on the HDFS file system. Snapshots ensure consistency if changes are still being made to data during the replication process. See Using Snapshots with Replication on page 513 for details.

About the Example Clusters

This tutorial uses the two example clusters listed in the table. The nodes shown are the master nodes for the clusters.

Production cluster	Backup cluster
cloudera-bdr-src-{1..4}.cloud.computers.com	cloudera-bdr-tgt-{1..4}.cloud.computers.com
http://cloudera-bdr-src-1.cloud.computers.com	http://cloudera-bdr-tgt-1.cloud.conputers.com
Source cluster for a backup replication schedule.	Destination cluster for a backup replication schedule.
Destination for a restore replication schedule.	Source for a restore replication schedule.
For restore, set peer relationship from this cluster.	For backup, set peer relationship from this cluster. To create an initial backup, set Schedule to Immediate.

The example clusters are not configured to use Kerberos, nor do they use external accounts for cloud storage on Amazon Web Services (AWS).

The example production cluster contains nine HDFS files in the `/user/cloudera` path:

```
● ● ▢   hdfs@cloudera-bdr-src-1:~
[hdfs@cloudera-bdr-src-1 ~]$ hadoop fs -ls /user/cloudera
Found 9 items
-rw-r--r--   3 hdfs supergroup 16106127360 2016-09-01 14:17 /user/cloudera/large.0
-rw-r--r--   3 hdfs supergroup  5368709120 2016-09-01 14:29 /user/cloudera/medium.0
-rw-r--r--   3 hdfs supergroup  5368709120 2016-09-01 14:33 /user/cloudera/medium.1
-rw-r--r--   3 hdfs supergroup  5368709120 2016-09-01 14:35 /user/cloudera/medium.2
-rw-r--r--   3 hdfs supergroup  1073741824 2016-09-01 14:36 /user/cloudera/small.0
-rw-r--r--   3 hdfs supergroup  1073741824 2016-09-01 14:36 /user/cloudera/small.1
-rw-r--r--   3 hdfs supergroup  1073741824 2016-09-01 14:47 /user/cloudera/small.2
-rw-r--r--   3 hdfs supergroup  1073741824 2016-09-01 14:48 /user/cloudera/small.3
-rw-r--r--   3 hdfs supergroup  1073741824 2016-09-01 14:48 /user/cloudera/small.4
[hdfs@cloudera-bdr-src-1 ~]$ 
```

The example backup cluster has not been used as the destination of a replication schedule yet, so the HDFS file system has no `/user/cloudera` directory:

```
● ● ▢   hdfs@cloudera-bdr-tgt-1:~
[hdfs@cloudera-bdr-tgt-1 ~]$ hadoop fs -ls /user/cloudera
ls: `/user/cloudera': No such file or directory
[hdfs@cloudera-bdr-tgt-1 ~]$ 
```

Backing Up and Restoring HDFS Data

This tutorial steps through the processes of creating and running a backup replication schedule, and creating a restore replication schedule designed for future use.

Preparing for disaster recovery includes these three major tasks:

- Backing Up HDFS Files on page 561
- Configuring a Restore Replication Schedule on page 564
- Recovering from Catastrophic Data Loss on page 567

Backup and restore are both configured and managed using Replication Schedules, available from the Backup menu on Cloudera Manager Admin Console:

> **Note:** Screenshots in this guide show version 5.9 of the Cloudera Manager Admin Console.

The backup and restore processes are configured, managed, and executed using replication schedules. Each replication schedule identifies a *source* and a *destination* for the given replication process. The replication process uses a *pull* model. When the replication process runs, the configured destination cluster accesses the given source cluster and transparently performs all tasks needed to copy the HDFS files to the destination cluster.

The destination cluster handles configuration and running the schedule. Typically, creating a backup replication schedule takes place on the backup cluster and creating a restore replication schedule takes place on the production cluster. Thus, as shown in this tutorial, the example production cluster, *cloudera-bdr-src-{1..4}.cloud.computers.com*, is the source for the backup replication schedule and the destination for the restore replication schedule.

Backing Up HDFS Files

The backup process begins at the Cloudera Manager Admin Console on the cluster designated as the backup, and includes these steps:

- [Step 1: Establish a Peer Relationship to the Production Cluster](#) on page 561
- [Step 2: Configure the Replication Schedule for the Backup](#) on page 562
- [Step 3: Verify Successful Replication](#) on page 563

Step 1: Establish a Peer Relationship to the Production Cluster

For a backup, the destination is the backup cluster, and the source is the production cluster.

The cluster establishing the peer relationship gains access to the source cluster and can run the export command, list HDFS files, and read files for copying them to the destination cluster. These are all the actions performed by the replication process whenever the defined schedule goes into action.

Defining the replication starts from the Cloudera Manager Admin Console on the backup cluster.

1. Log in to Cloudera Manager Admin Console on the backup cluster.
2. Click the **Backup** tab and select **Peers** from the menu.
3. On the Peers page, click **Add Peer**:

4. On the Add Peer page:

- **a. Peer Name** - Enter a meaningful name for the cluster that you want to back up. This peer name becomes available in the next step, to be selected as the source for the replication.
- **b. Peer URL** - Enter the URL for the Cloudera Manager Admin Console running on the master node of the cluster.
- **c. Peer Admin Username** - Enter the administrator user account for the production cluster.
- **d. Peer Admin Password** - Enter the password for the administrator account for the production cluster.

5. Click **Add Peer** to save your settings, connect to the production cluster, and establish this peer relationship.

After the system establishes and verifies the connection to the peer, the Peers page re-displays, showing the Status column as Connected (note the green check-mark):

With the peer relationship established from destination to source, create a schedule to replicate HDFS files from the source (production cluster) to the destination (backup cluster).

Step 2: Configure the Replication Schedule for the Backup

From the Cloudera Manager Admin Console on the backup cluster:

1. Click the **Backup** tab and select **Replication Schedules** from the menu.

2. On the Replication Schedules page, select **Create Schedule** > **HDFS Replication**.

3. On the Create HDFS Replication page, click the **General** tab to display the default schedule options:

- **a. Source** - Make sure the cluster node selected in the drop-down is the production cluster (the cluster to be backed up).
- **b. Source Path** - Specify the directory name on the production cluster holding the files to back up. Use an asterisk (*) on the directory name to specify that only the explicit directory and no others should be created on the destination. Without the asterisk, directories may be nested inside a containing directory on the destination.
- **c. Destination** - Select the cluster to use as the target of the replication process, typically, the cluster to which you have logged in and the cluster to which you want to backup HDFS data.
- **d. Destination Path** - Specify a directory name on the backup cluster.
- **e. Schedule** - Immediate.
- **f. Run As Username** - Leave as Default.
- **g. Scheduler Pool** - Leave as Default.

4. Click **Save Schedule**.

The files are replicated from the source cluster to the backup cluster immediately.

When the process completes, the Replication Schedules page re-displays, showing a green check-mark and time-stamp in the Last Run column.

Step 3: Verify Successful Replication

Verify that the HDFS files are now on the backup cluster by using the HDFS File Browser or the `hadoop fs -ls` command (shown below):

The HDFS files are now on both the production cluster and the backup cluster.

You can now use the backup as a source for a restore onto the production system when needed.

Configuring a Restore Replication Schedule

To restore data from a backup, configure a Restore Schedule on the cluster to which you want the data pulled. For this example, the replication schedule is created on the example production cluster and designed to pull HDFS files from the backup cluster to a different path on the production cluster, to enable testing the restore process in advance of any failure.

> **Note:** This approach lets you step through and test the process prior to using the schedule in a production environment.

Setting up a schedule for a restore follows the same pattern as setting up the backup, but with all actions initiated using the Cloudera Manager Admin Console on the production cluster.

To set up and test a replication schedule to restore HDFS from an existing backup copy, follow these steps:

- Step 1: Establish a Peer Relationship to the Backup Cluster on page 564
- Step 2: Configure Replication Schedule to Test the Restore on page 565
- Step 3: Disable the Replication Schedule on page 566
- Step 4: Test the Restore Replication Schedule on page 566
- Step 5: Verify Successful Data Restoration on page 567

Step 1: Establish a Peer Relationship to the Backup Cluster

To restore HDFS files on the production cluster, establish a Peer relationship from the destination to the source for the restore.

Log in to the Cloudera Manager Admin Console on the production cluster.

1. Click the Backup tab and select **Peers** from the menu.
2. On the Peers page, click **Add Peers**.

3. On the Add Peer page:

   a. **Peer Name** - Enter a meaningful name for the cluster that you want to restore data from, for example, *BackupCluster*.
   b. **Peer URL** - Enter the URL for the Cloudera Manager Admin Console running on the master node of the cluster.
   c. **Peer Admin Username** - Enter the administrator user name for the peer cluster.
   d. **Peer Admin Password** - Enter the password for the administrator user account.

4. Click **Add Peer** to save the settings, connect to the backup cluster, and establish the peer relationship.

Once the system connects and tests the peer relationship, the Peers page lists its name, URL, and Status (Connected):

The backup cluster is now available as a peer, for use in a Replication Schedule.

Step 2: Configure Replication Schedule to Test the Restore

The goal in these steps is to create a replication schedule that can be used when needed, in the future, but to leave it in a disabled state. However, because Replication Schedules cannot be created in a disabled state, you initially set the date far into the future and then disable the schedule in a subsequent step.

From the Cloudera Manager Admin Console on the production cluster:

1. Click the **Backup** tab and select **Replication Schedules** from the menu.
2. On the Replication Schedules page, select **Create Schedule** > **HDFS Replication**.
3. On the General settings tab of the Create HDFS Replication page:

    a. **Source** - Make sure the cluster node selected in the drop-down is the backup cluster.
    b. **Source Path** - Specify the directory name that you want to back up. Use an asterisk (*) on the directory name to specify that only the explicit directory and no others should be created on the destination.
    c. **Destination** - Select the cluster to use as the target for the replication process.
    d. **Destination Path** - Specify a directory name on the backup cluster.

    > **Important:** The path is a new directory on the example production cluster.

    e. **Schedule** - Set to a time far in the future, so that the schedule does not run as soon as it is saved.
    f. **Run As Username** - Leave as Default.
    g. **Scheduler Pool** - Leave as Default.

4. Click **Save Schedule**.

The Replication Schedule is saved and displays in the Replication Schedules list, with the future date listed in the Next Run column.

Before continuing, immediately disable this newly created Replication Schedule.

Step 3: Disable the Replication Schedule

1. On the Replication Schedules page, select the replication schedule.
2. From the **Actions** drop-down menu, select **Disable**. When the page refreshes, Disabled displays in the Next Run column:

You can leave restore Replication Schedules pre-configured and disabled in this way so they are ready to use in the event of a catastrophic data loss. Before relying on this approach, test the schedule.

Step 4: Test the Restore Replication Schedule

The Replication Schedule defined in the example restores data to a specific directory path identified for the purpose of restoration (`/user/cloudera-restored`) rather than targeting the original source directory path.

From the Cloudera Manager Admin Console on the production cluster, with Replication Schedules page displayed:

1. On the Replication Schedules page, select the disabled replication schedule.
2. Select **Actions** > **Run Now**.

When the replication process completes, disable the Replication Schedule once again:

1. On the Replication Schedules page, select the replication schedule.
2. Select **Actions** > **Disable** .

You can now verify that the files have been replicated to the destination directory path.

Step 5: Verify Successful Data Restoration

To manually verify that your data has been restored to the source cluster, you can use the HDFS File Browser or the `hadoop` command-line, as shown here:

Compare the restored data in `/user/cloudera-restored` to the data in `/user/cloudera` to validate that the restore schedule operates as expected.

At this point, until you need to actually restore production HDFS data files, you can leave the Replication Schedule disabled.

> **Important:** In a production environment, only one Replication Schedule for a given dataset should be active at the same time. In this example, the Replication Schedule that created the backup has not been disabled yet, because the HDFS files from the backup cluster were restored to a different path on the example production cluster.

Recovering from Catastrophic Data Loss

With backup and restore Replication Schedules set up and validated, you can restore data when production data has been erroneously deleted, as shown in this screenshot:

In the event of an actual data loss on a production cluster, you should first disable any existing replication schedules for the affected datasets *before* activating a replication schedule for the restore, to avoid overwriting existing replicas on the backup cluster with defective files.

- Step 1: Disable the Backup Replication Schedule on page 568
- Step 2: Edit the Existing Replication Schedule on page 568
- Step 3: Run the Restore Replication Schedule on page 569
- Step 4: Return the Restore Replication Schedule to a Disabled State on page 569
- Step 5: Re-enable the Backup Replication Schedule on page 569

Step 1: Disable the Backup Replication Schedule

Disabling any existing replication schedule for HDFS backups can help prevent the replication of lost or corrupted data files over existing backups.

At the Cloudera Manager Admin Console on the backup cluster:

1. Select **Backup** > **Replication Schedules**.
2. On the Replication Schedules page, select the schedule.
3. From the Actions drop-down menu, select **Disable**:

When the Replication Schedules pages refreshes, you see **Disabled** in the **Next Run** column for the schedule.

Step 2: Edit the Existing Replication Schedule

With the replication schedule disabled, you can edit the replication schedule verified previously (Step 4: Test the Restore Replication Schedule on page 566) and restore the data to the production cluster.

From the Cloudera Manager Admin Console on the production cluster:

1. Click the **Backup** tab and select **Replication Schedules** from the menu.
2. On the Replication Schedules page, select **Create Schedule** > **HDFS Replication**.
3. On the General settings tab of the Create HDFS Replication page:

    a. **Source** - Name of the backup cluster from which to pull the data. For this example, the source is the backup cluster.

    b. **Source Path** - The path on the backup cluster that contains the data you want to restore. Use the asterisk (*) at the end of the directory name to prevent extraneous sub-directories being created on the destination.

    c. **Destination** - The name of the cluster on which to restore the data, in which case, the example production cluster.

    d. **Destination Path** - Directory in which to restore the HDFS data files, in this case, the directory on the example production system.

    e. **Schedule** - Once.

    f. **Start Time** - Leave set to the future date and time that you originally defined in Step 2: Configure Replication Schedule to Test the Restore on page 565.

    g. **Run as Username** - Leave set as Default.

4. Click **Save Schedule**.

The settings for the Replication Schedule are saved, and the page refreshes. Because this replication schedule is currently disabled, you must actively run the schedule to restore the data.

Step 3: Run the Restore Replication Schedule

While still on the Replication Schedules page:

1. Select the edited replication schedule.
2. From the **Actions** drop-down menu, select **Run Now**.

The replication schedule executes and restores the HDFS files to the original location.

When the process completes, the Replication Schedules page displays the time-stamp and check-mark in the Last Run column for the schedule.

Step 4: Return the Restore Replication Schedule to a Disabled State

You can now disable the schedule, and after verifying that HDFS data has been successfully restored to the production cluster, you can re-enable the backup schedule. While still displaying the Replication Schedules page on the production cluster:

1. Select the replication schedule used for the restore.
2. Edit its configuration again, to point to a non-production directory.
3. Select **Actions** > **Disable**.

After confirming that the schedule has been disabled—you see **Disabled** in the Next Run column for this schedule—but before re-enabling the backup schedule, verify that the HDFS files have been restored to the production cluster.

Step 5: Re-enable the Backup Replication Schedule

With data restored to the production cluster and the replication schedule on the production cluster disabled, you can re-enable the replication schedule on the backup cluster.

1. At the backup cluster, log in to the Cloudera Manager Admin Console.
2. Select **Backup** > **Replication Schedules**.

3. On the Replication Schedules page, select the appropriate replication schedule to back up the production cluster.

4. From the **Actions** drop-down menu, select **Enable**.

This concludes the tutorial. In an actual production environment, you should configure replication schedules to regularly backup production systems. For restoring files from any backup, you can create and test a replication schedules in advance, as shown in this tutorial.

Alternatively, you can create a replication schedule to restore data specifically when needed. See How To Back Up and Restore Apache Hive Data Using Cloudera Enterprise BDR on page 547 for details.

See Backup and Disaster Recovery on page 483 and BDR Tutorials on page 547 for more information about Cloudera Enterprise BDR.

BDR Automation Examples

You can use the Cloudera Manager API to automate BDR tasks, such as creating a schedule for a replication. This page describes an automated solution for creating, running, and managing HDFS replication schedules in order to minimize Recovery Point Objectives (RPOs) for late arriving data or to automate recovery after disaster recovery.

For more information about the Cloudera Manager API and how to install the API client, see the following:

- Cloudera Manager API on page 24
- Python Client

Automating HDFS Replication Schedules

Automating HDFS replication with the API is a multi-step process that involves the following tasks:

Step 1. Create a Peer

Before you can create or run a replication schedule, you need a peer Cloudera Manager instance. This peer acts as the source Cloudera Manager instance where data is pulled from. See Designating a Replication Source on page 486 for more information.

The following code sample shows you how to create a peer:

```
import cm_client
from cm_client.rest import ApiException
from pprint import pprint

# Configure HTTP basic authorization: basic
cm_client.configuration.username = "[***username***]"
cm_client.configuration.password = "[***password***]"
cm_client.configuration.verify_ssl = False

TARGET_CM_HOST = "[***destination_cluster***]"
SOURCE_CM_URL = "[***source_cluster***]:7180"

# Create an instance of the API class
api_host = TARGET_CM_HOST
port = '7183'
api_version = 'v30'
# Construct base URL for API
# http://cmhost:7180/api/v30api_url = api_host + ':' + port + '/api/' + api_version
api_client = cm_client.ApiClient(api_url)
api_instance = cm_client.CmPeersResourceApi(api_client)

# create peer on the target
# url= is the source
body = cm_client.ApiCmPeer(cloudera_manager_created_user=True,
  name="peer_name_from_api",
  password="admin",
  type="REPLICATION",
  url=SOURCE_CM_URL,
  username="admin")

api_response = api_instance.create_peer(body=body)
```

The above sample creates an API root handle and gets a Cloudera Manager instance from it before creating the peer. To implement a similar solution to the example, keep the following guidelines in mind:

- Replace `<destination_cluster>` with the domain name of the destination, for example `target.cm.cloudera.com`.
- Replace `<source_cluster>` with the domain name of the source, for example `src.cm.cloudera.com:7180/`.
- The user you specify must possess a role that is capable of creating a peer, such as the Cluster Administrator role.

Step 2. Create the HDFS Replication Schedule

After you have add a peer Cloudera Manager instance that functions as the source, you can create a replication schedule:

```python
import cm_client
from cm_client.rest import ApiException
from pprint import pprint

# Configure HTTP basic authorization: basic
cm_client.configuration.username = "[***username***]"
cm_client.configuration.password = "[***password***]"

cm_client.configuration.verify_ssl = False

TARGET_CM_HOST = "[***destination_cluster***]"


# Create an instance of the API class
api_host = TARGET_CM_HOST
port = '7183'
api_version = 'v30'

# Construct base URL for API
# http://cmhost:7180/api/v30api_url = api_host + ':' + port + '/api/' + api_version
api_client = cm_client.ApiClient(api_url)
api_instance = cm_client.ReplicationsResourceApi(api_client)
cluster_name = 'Cluster 1' # str |
service_name = 'HDFS-1' # str | The service name.

body = cm_client.ApiReplicationScheduleList([{
   "displayName" : "ScheduleNameFromAPI",
   "startTime" : "2021-03-11T18:28:18.684Z",
   "interval" : 0,
   "intervalUnit" : "MINUTE",
   "alertOnStart" : False,
   "alertOnSuccess" : False,
   "alertOnFail" : False,
   "alertOnAbort" : False,
   "hdfsArguments" : {
     "sourceService" : {
       "peerName" : "peer_name_from_api",
       "clusterName" : "Cluster 1",
       "serviceName" : "HDFS-1"
     },
     "sourcePath" : "/tmp",
     "destinationPath" : "/tmp",
     "mapreduceServiceName" : "YARN-1",
     "userName" : "d",
     "numMaps" : 20,
     "dryRun" : False,
     "bandwidthPerMap" : 100,
     "abortOnError" : False,
     "removeMissingFiles" : False,
     "preserveReplicationCount" : True,
     "preserveBlockSize" : True,
     "preservePermissions" : False,
     "skipChecksumChecks" : False,
     "skipListingChecksumChecks" : False,
     "skipTrash" : False,
     "replicationStrategy" : "DYNAMIC",
     "preserveXAttrs" : False,
     "exclusionFilters" : [ ],
     "raiseSnapshotDiffFailures" : False
   }}])
```

```
api_response = api_instance.create_schedules(cluster_name, service_name, body=body)
```

The example creates `ApiHdfsReplicationArguments` and populate attributes such as source path, destination name, MapReduce service to use, and others. For the source service, you will need to provide the HDFS service name and cluster name on the source Cloudera Manager instance. See the [API documentation](#) for the complete list of attributes for `ApiHdfsReplicationArguments`.

At the end of the example, `hdfs_args` is used to create an HDFS replication schedule.

Step 3. Run the Replication Schedule

The replication schedule created in step 2 has a frequency of 1 DAY, so the schedule will run at the initial start time every day. You can also manually run the schedule using the following:

```
import cm_client
from cm_client.rest import ApiException
from pprint import pprint

# Configure HTTP basic authorization: basic
cm_client.configuration.username = "[***username***]"
cm_client.configuration.password = "[***password***]"

cm_client.configuration.verify_ssl = False

TARGET_CM_HOST = "[***destination_cluster***]"


# Create an instance of the API class
api_host = TARGET_CM_HOST
port = '7183'
api_version = 'v30'

# Construct base URL for API
# http://cmhost:7180/api/v30api_url = api_host + ':' + port + '/api/' + api_version
api_client = cm_client.ApiClient(api_url)
api_instance = cm_client.ReplicationsResourceApi(api_client)
```

Step 4. Monitor the Schedule

Once you get a command (`cmd`), you can wait for the command to finish and then get the results:

```
import cm_client
from cm_client.rest import ApiException
from pprint import pprint

# Configure HTTP basic authorization: basic
cm_client.configuration.username = "[***username***]"
cm_client.configuration.password = "[***password***]"

cm_client.configuration.verify_ssl = False


TARGET_CM_HOST = "[*** destination_cluster ***]"

# Create an instance of the API class
api_host = TARGET_CM_HOST
port = '7183'
api_version = 'v30'

# Construct base URL for API
# http://cmhost:7180/api/v30api_url = api_host + ':' + port + '/api/' + api_version
api_client = cm_client.ApiClient(api_url)
api_instance = cm_client.ReplicationsResourceApi(api_client)
cluster_name = [*** Cluster 1 ***] # str |
schedule_id = 9 # int | Id of an existing replication schedule.
service_name = [*** HDFS-1 ***] # str | The service name.
limit = 20 # int | Maximum number of commands to retrieve. (optional) (default to 20)
offset = 0 # int | Index of first command to retrieve. (optional) (default to 0)
view = [*** summary ***] # str | The view to materialize. (optional) (default to summary)
```

```
try:
    # Returns a list of commands triggered by a schedule.
    api_response = api_instance.read_history([*** cluster_name ***], [*** schedule_id
***], [*** service_name ***], limit=limit, offset=offset, view=view)
    pprint(api_response)
except ApiException as e:
    print("Exception when calling ReplicationsResourceApi->read_history: %s\n" % e)
```

Configuring Replication to/from Cloud Providers

BDR supports Amazon S3 as HDFS replication sources or destinations. The following example shows you how to use the API to configure BDR to or from Amazon S3.

Step 1. Add a Cloud Account

Instead of adding a peer Cloudera Manager instance like a cluster-to-cluster replication, replicating to or from a cloud provider requires an account for that provider.

The following example shows how to add an S3 account:

```
# Construct base URL for API
# http://cmhost:7180/api/v30api_url = api_host + ':' + port + '/api/' + api_version
api_client = cm_client.ApiClient(api_url)

# create an instance of the API class
api_instance = cm_client.ExternalAccountsResourceApi(cm_client.ApiClient(configuration))

ACCESS_KEY="[*** .... ***]"
SECRET_KEY="[*** .... ***]"
TYPE_NAME = 'AWS_ACCESS_KEY_AUTH'


account_configs = {"items" : [{"name" : "aws_access_key","value" : [***ACCESS_KEY***]},
  {"name" : "aws_secret_key","value" : [***SECRET_KEY***]}]}
body = cm_client.ApiExternalAccount(name="cloudAccount1", display_name="cloudAccount1",
 type_name=TYPE_NAME, account_configs=account_configs)


try:
    # Create a new external account.
    api_response = api_instance.create_account(body=body)
    pprint(api_response)
except ApiException as e:
    print("Exception when calling ExternalAccountsResourceApi->create_account: %s\n" %
 e)
```

Step 2. Create the Replication Schedule

```
CLUSTER_NAME='Cluster-tgt-1'
HDFS_NAME='HDFS-tgt-1'
CLOUD_ACCOUNT='cloudAccount1'
YARN_SERVICE='YARN-1'

hdfs = api_root.get_cluster(CLUSTER_NAME).get_service(HDFS_NAME)

hdfs_cloud_args = ApiHdfsCloudReplicationArguments(None)
hdfs_cloud_args.sourceService = ApiServiceRef(None,
                                    peerName=None,
                                    clusterName=CLUSTER_NAME,
                                    serviceName=HDFS_NAME)
hdfs_cloud_args.sourcePath = '/src/path'
hdfs_cloud_args.destinationPath = 's3a://bucket/target/path/'
hdfs_cloud_args.destinationAccount = CLOUD_ACCOUNT
hdfs_cloud_args.mapreduceServiceName = YARN_SERVICE

# creating a schedule with daily frequency
start = datetime.datetime.now() # The time at which the scheduled activity is triggered
  for the first time.
end = start + datetime.timedelta(days=365) # The time after which the scheduled activity
```

```
  will no longer be triggered.

schedule = hdfs.create_replication_schedule(start, end, "DAY", 1, True, hdfs_args)
```

The example creates `ApiHdfsCloudReplicationArguments`, populates it, and creates an HDFS to S3 backup schedule. In addition to specifying attributes such as the source path and destination path, the example provides `destinationAccount` as `CLOUD_ACCOUNT` and `peerName` as `None` in `sourceService`. The `peerName` is `None` since there is no peer for cloud replication schedules.

`hdfs_cloud_args` is then used to create a HDFS-S3 replication schedule with a frequency of `1` day.

Step 3. Run the Replication Schedule

The replication schedule created in step 2 has a frequency of 1 DAY, so the schedule will run at the initial start time every day. You can also manually run the schedule using the following:

```
cmd = hdfs.trigger_replication_schedule(schedule.id)
```

Step 4. Monitor the Schedule

Once you get a command (`cmd`), you can wait for the command to finish and then get the results:

```
cmd = cmd.wait()
result = hdfs.get_replication_schedule(schedule.id).history[0].hdfsResult
```

Maintaining Replication Schedules

The following actions can be performed on replication schedules that are cluster-to-cluster or cluster to/from a cloud provider:

Get all replication schedules for a given service:

```
schs = hdfs.get_replication_schedules()
```

Get a given replication schedule by schedule id for a given service:

```
sch = hdfs.get_replication_schedule(schedule_id)
```

Delete a given replication schedule by schedule id for a given service:

```
sch = hdfs.delete_replication_schedule(schedule_id)
```

Update a given replication schedule by schedule id for a given service:

```
sch.hdfsArguments.removeMissingFiles = True
sch = hdfs.update_replication_schedule(sch.id, sch)
```

Debugging failures during replication

If a replication job fails, you can download replication diagnostic data for the replication command to troubleshoot and diagnose any issues.

The diagnostic data includes all the logs generated, including the MapReduce logs. You can also upload the logs to a support case for further analysis. Collecting a replication diagnostic bundle is available for API v11+ and Cloudera Manager version 5.5+.

```
args = {}
resp = hdfs.collect_replication_diagnostic_data(schedule_id=schedule.id, args)

# Download replication diagnostic bundle to a temp directory
tmpdir = tempfile.mkdtemp(prefix="support-bundle-replication")
support_bundle_path = os.path.join(tmpdir, "support-bundle.zip")
cm.download_from_url(resp.resultDataUrl, support_bundle_path)
```

Migrating Data between Clusters Using distcp

You can migrate the data from any Hadoop cluster to a CDH 5 cluster by using a tool that copies out data in parallel, such as the DistCp tool offered in CDH 5. The following sections provide information and instructions:

Copying Cluster Data Using DistCp

> **Note:** This page contains references to CDH 5 components or features that have been removed from CDH 6. These references are only applicable if you are managing a CDH 5 cluster with Cloudera Manager 6. For more information, see Deprecated Items.

The distributed copy command, `distcp`, is a general utility for copying large data sets between distributed filesystems within and across clusters. You can also use `distcp` to copy data to and from an Amazon S3 bucket. The `distcp` command submits a regular MapReduce job that performs a file-by-file copy.

To see the `distcp` command options, run the built-in help:

```
$ hadoop distcp
```

> **Important:**
> - Do not run `distcp` as the `hdfs` user which is blacklisted for MapReduce jobs by default.
> - Do not use Hadoop shell commands (such as `cp`, `copyfromlocal`, `put`, `get`) for large copying jobs or you may experience I/O bottlenecks.

DistCp Syntax and Examples

You can use `distcp` to copy files between compatible clusters in either direction. The most basic form of the `distcp` command only requires that information:

```
$ hadoop distcp <source> <destination>
```

Copying Between the Same CDH Version

Use the following syntax:

```
hadoop distcp hdfs://<namenode>:<port> hdfs://<namenode>
```

For example, the following command copies data from `example-source` to `example-dest`:

```
hadoop distcp hdfs://example-source.cloudera.com:50070 hdfs://example-dest.cloudera.com
```

Port 50070 is the default NameNode port for HDFS.

Different but Compatible CDH Major Versions

Run the `distcp` command on the cluster that runs the higher version of CDH, which should be the destination cluster. Use the following syntax:

```
hadoop distcp webhdfs://<namenode>:<port> hdfs://<namenode>
```

Note the `webhdfs` prefix for the remote cluster, which should be your source cluster. You must use `webhdfs` when the clusters run different major versions. When clusters run the same version, you can use the `hdfs` protocol for better performance.

For example, the following command copies data from a lower CDH source cluster named `example-source` to a higher CDH version destination cluster named`example-dest`:

```
hadoop distcp webhdfs://example-source.cloudera.com:50070 hdfs://example-dest.cloudera.com
```

Copying a Specific Path

You can also use a specific path, such as `/hbase` to move HBase data:

```
hadoop distcp hdfs://example-source.cloudera.com:50070/hbase
hdfs://example-dest.cloudera.com/hbase
```

Copying to/from Amazon S3

The following syntax for `distcp` shows how to copy data to/from S3:

```
#Copying from S3
hadoop distcp s3a://<bucket>/<data> hdfs://<namenode>/<directory>/
#Copying to S3
hadoop distcp hdfs://<namenode>/<directory> s3a://<bucket>/<data>
```

This is a basic example of using `distcp` with S3. For more information, see Using DistCp with Amazon S3 on page 578.

Using DistCp with Highly Available Remote Clusters

You can use `distcp` to copy files between highly available clusters by configuring access to the remote cluster with the nameservice ID. To enable support, perform the following steps:

1. Create a new directory and copy the contents of the `/etc/hadoop/conf` directory on the local cluster to this directory. The local cluster is the cluster where you plan to run the `distcp` command.

   Specify this directory for the `--config` parameter when you run the `distcp` command in step 5.

   The following steps use `distcpConf` as the directory name. Substitute the name of the directory you created for `distcpConf`.

2. In the `hdfs-site.xml` file in the `distcpConf` directory, add the nameservice ID for the remote cluster to the `dfs.nameservices` property.

   > **Note:**
   >
   > If the remote cluster has the same nameservice ID as the local cluster, change the remote cluster's nameservice ID. Nameservice names must be unique.
   >
   > For example, if the nameservice name for both clusters is `nameservice1`, change the nameservice ID of the remote cluster to a different ID, such as `externalnameservice`:
   >
   > ```
   > <property>
   >  <name>dfs.nameservices</name>
   >  <value>nameservice1,externalnameservice</value>
   > </property>
   > ```

3. On the remote cluster, find the `hdfs-site.xml` file and copy the properties that refers to the nameservice ID to the end of the `hdfs-site.xml` file in the `distcpConf` directory you created in step 1:

   - `dfs.ha.namenodes.<nameserviceID>`
   - `dfs.client.failover.proxy.provider.<remote nameserviceID>`
   - `dfs.ha.automatic-failover.enabled.<remote nameserviceID>`
   - `dfs.namenode.rpc-address.<nameserviceID>.<namenode1>`
   - `dfs.namenode.servicerpc-address.<nameserviceID>.<namenode1>`
   - `dfs.namenode.http-address.<nameserviceID>.<namenode1>`

- `dfs.namenode.https-address.<nameserviceID>.<namenode1>`
- `dfs.namenode.rpc-address.<nameserviceID>.<namenode2>`
- `dfs.namenode.servicerpc-address.<nameserviceID>.<namenode2>`
- `dfs.namenode.http-address.<nameserviceID>.<namenode2>`
- `dfs.namenode.https-address.<nameserviceID>.<namenode2>`

By default, you can find the `hdfs-site.xml` file in the `/etc/hadoop/conf` directory on a node of the remote cluster.

4. If you changed the nameservice ID for the remote cluster in step 2, update the nameservice ID used in the properties you copied in step 3 with the new nameservice ID, accordingly.

The following example shows the properties copied from the remote cluster with the following values:

- A remote nameservice called `externalnameservice`
- NameNodes called `namenode1` and `namenode2`
- A host named `remotecluster.com`

```
 <property>
<name>dfs.ha.namenodes.externalnameservice</name>
<value>namenode1,namenode2</value>
</property>

<property>
<name>dfs.client.failover.proxy.provider.externalnameservice</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>

<property>
<name>dfs.ha.automatic-failover.enabled.externalnameservice</name>
<value>true</value>
</property>

<property>
<name>dfs.namenode.rpc-address.externalnameservice.namenode1</name>
<value>remotecluster.com:8020</value>
</property>

<property>
<name>dfs.namenode.servicerpc-address.externalnameservice.namenode1</name>
<value>remotecluster.com:8022</value>
</property>

<property>
<name>dfs.namenode.http-address.externalnameservice.namenode1</name>
<value>remotecluster.com:20101</value>
</property>

<property>
<name>dfs.namenode.https-address.externalnameservice.namenode1</name>
<value>remotecluster.com:20102</value>
</property>

<property>
<name>dfs.namenode.rpc-address.externalnameservice.namenode2</name>
<value>remotecluster.com:8020</value>
</property>

<property>
<name>dfs.namenode.servicerpc-address.externalnameservice.namenode2</name>
<value>remotecluster.com:8022</value>
</property>

<property>
<name>dfs.namenode.http-address.externalnameservice.namenode2</name>
<value>remotecluster.com:20101</value>
</property>

<property>
<name>dfs.namenode.https-address.externalnameservice.namenode2</name>
```

```
<value>remotecluster.com:20102</value>
</property>
```

At this point, the `hdfs-site.xml` file in the `distcpConf` directory should have both clusters and 4 NameNode IDs.

5. Depending on the use case, the options specified when you run the `distcp` may differ. Here are some examples:

> **Note:** The remote cluster can be either the source or the target. The examples provided specify the remote cluster as the source.

To copy data from an insecure cluster , run the following command:

```
hadoop --config distcpConf distcp hdfs://<nameservice>/<source_directory> <target
directory>
```

To copy data from a secure cluster, run the following command:

```
hadoop --config distcpConf distcp
-Dmapreduce.job.hdfs-servers.token-renewal.exclude=<nameservice>
hdfs://<nameservice>/<source_directory> <target directory>
```

For example:

```
hadoop --config distcpConf distcp -Dmapreduce.job.hdfs-servers.token-renewal.exclude=ns1
 hdfs://ns1/xyz /tmp/test
```

If the `distcp` source or target are in encryption zones, include the following `distcp` options: `-skipcrccheck -update`. The `distcp` command may fail if you do not include these options when the source or target are in encryption zones because the CRC for the files may differ.

For CDH 5.12.0 and later, `distcp` between clusters that both use HDFS Transparent Encryption, you must include the exclude parameter.

Using DistCp with Amazon S3

You can copy HDFS files to and from an Amazon S3 instance. You must provision an S3 bucket using Amazon Web Services and obtain the access key and secret key. You can pass these credentials on the `distcp` command line, or you can reference a credential store to "hide" sensitive credentials so that they do not appear in the console output, configuration files, or log files.

Amazon S3 block and native filesystems are supported with the `s3a://` protocol.

Example of an Amazon S3 Block Filesystem URI: `s3a://`*bucket_name*`/path/to/file`

S3 credentials can be provided in a configuration file (for example, core-site.xml):

```
<property>
    <name>fs.s3a.access.key</name>
    <value>...</value>
</property>
<property>
    <name>fs.s3a.secret.key</name>
    <value>...</value>
</property>
```

You can also enter the configurations in the **Advanced Configuration Snippet** for `core-site.xml`, which allows Cloudera Manager to manage this configuration. See <u>Custom Configuration</u> on page 74.

You can also provide the credentials on the command line:

```
hadoop distcp -Dfs.s3a.access.key=... -Dfs.s3a.secret.key=... s3a://
```

For example:

```
hadoop distcp -Dfs.s3a.access.key=myAccessKey -Dfs.s3a.secret.key=mySecretKey
/user/hdfs/mydata s3a://myBucket/mydata_backup
```

> **Important:** Entering secrets on the command line is inherently insecure. These secrets may be accessed in log files and other artifacts. Cloudera recommends that you use a credential provider to store secrets. See Using a Credential Provider to Secure S3 Credentials on page 579.

> **Note:** Using the `-diff` option with the `distcp` command requires a DistributedFileSystem on both the source and destination and is not supported when using `distcp` to copy data to or from Amazon S3.

Using a Credential Provider to Secure S3 Credentials

You can run the `distcp` command without having to enter the access key and secret key on the command line. This prevents these credentials from being exposed in console output, log files, configuration files, and other artifacts. Running the command in this way requires that you provision a credential store to securely store the access key and secret key. The credential store file is saved in HDFS.

> **Note:** Using a Credential Provider does not work with MapReduce v1 (MRV1).

To provision credentials in a credential store:

1. Provision the credentials by running the following commands:

```
hadoop credential create fs.s3a.access.key -value access_key -provider
jceks://hdfs/path_to_credential_store_file
hadoop credential create fs.s3a.secret.key -value secret_key -provider
jceks://hdfs/path_to_credential_store_file
```

  For example:

```
hadoop credential create fs.s3a.access.key -value foobar -provider
jceks://hdfs/user/alice/home/keystores/aws.jceks
hadoop credential create fs.s3a.secret.key -value barfoo -provider
jceks://hdfs/user/alice/home/keystores/aws.jceks
```

  You can omit the `-value` option and its value and the command will prompt the user to enter the value.

  For more details on the `hadoop credential` command, see Credential Management (Apache Software Foundation).

2. Copy the contents of the `/etc/hadoop/conf` directory to a working directory.
3. Add the following to the `core-site.xml` file in the working directory:

```
<property>
<name>hadoop.security.credential.provider.path</name>
<value>jceks://hdfs/path_to_credential_store_file</value>
</property>
```

4. Set the `HADOOP_CONF_DIR` environment variable to the location of the working directory:

```
export HADOOP_CONF_DIR=path_to_working_directory
```

After completing these steps, you can run the `distcp` command using the following syntax:

```
hadoop distcp source_path s3a://destination_path
```

You can also reference the credential store on the command line, without having to enter it in a copy of the `core-site.xml` file. You also do not need to set a value for `HADOOP_CONF_DIR`. Use the following syntax:

```
hadoop distcp source_path s3a://bucket_name/destination_path
 -Dhadoop.security.credential.provider.path=jceks://hdfspath_to_credential_store_file
```

There are additional options for the `distcp` command. See [DistCp Guide (Apache Software Foundation)](#).

Examples of DistCP Commands Using the S3 Protocol and Hidden Credentials

Copying files to Amazon S3

```
hadoop distcp /user/hdfs/mydata s3a://myBucket/mydata_backup
```

Copying files from Amazon S3

```
hadoop distcp s3a://myBucket/mydata_backup //user/hdfs/mydata
```

Copying files to Amazon S3 using the `-filters` option to exclude specified source files

You specify a file name with the `-filters` option. The referenced file contains regular expressions, one per line, that define file name patterns to exclude from the `distcp` job. The pattern specified in the regular expression should match the fully-qualified path of the intended files, including the scheme (`hdfs`, `webhdfs`, `s3a`, etc.). For example, the following are valid expressions for excluding files:

```
hdfs://x.y.z:8020/a/b/c
webhdfs://x.y.z:50070/a/b/c
s3a://bucket/a/b/c
```

Reference the file containing the filter expressions using `-filters` option. For example:

```
hadoop distcp -filters /user/joe/myFilters /user/hdfs/mydata s3a://myBucket/mydata_backup
```

Contents of the sample `myFilters` file:

```
.*foo.*
.*/bar/.*
hdfs://x.y.z:8020/tmp/.*
hdfs://x.y.z:8020/tmp1/file1
```

The regular expressions in the `myFilters` exclude the following files:

- `.*foo.*` – excludes paths that contain the string "`foo`".
- `.*/bar/.*` – excludes paths that include a directory named `bar`.
- `hdfs://x.y.z:8020/tmp/.*` – excludes all files in the `/tmp` directory.
- `hdfs://x.y.z:8020/tmp1/file1` – excludes the file `/tmp1/file1`.

Copying files to Amazon S3 with the `-overwrite` option.

The `-overwrite` option overwrites destination files that already exist.

```
hadoop distcp -overwrite /user/hdfs/mydata  s3a://user/mydata_backup
```

For more information about the **-filters**, **-overwrite**, and other options, see [DIstCp Guide: Command Line Options (Apache Software Foundation)](#).

Using DistCp with Microsoft Azure (ADLS)

You can use the `distcp` command to copy data from ADLS to your cluster :

1. Configure connectivity to ADLS using one of the methods described in <u>Configuring ADLS Connectivity</u> on page 603.
2. If you are copying data to or from Amazon S3, also configure connectivity to S3 as described above. See <u>Using DistCp with Amazon S3</u> on page 578
3. Use the following syntax to define the Hadoop Credstore:

```
export HADOOP_CONF_DIR=path_to_working_directory
export HADOOP_CREDSTORE_PASSWORD=hadoop_credstore_password
```

4. Run `distcp` jobs using the following syntax:

   ADLS to local cluster:

```
hadoop distcp adl://store.azuredatalakestore.net/src hdfs://hdfs_destination_path
```

   Local cluster to ADLS:

```
hadoop distcp hdfs://hdfs_destination_path adl://store.azuredatalakestore.net/src
```

   You can also use `distcp` to copy data between Amazon S3 and Microsoft ADLS.

   S3 to ADLS:

```
hadoop distcp s3a://user/my_data adl://Account_Name.azuredatalakestore.net/my_data_backup/
```

   ADL to S3:

```
hadoop distcp s3a://user/my_data adl://Account_Name.azuredatalakestore.net/my_data_backup/
```

Note that when copying data between these remote filesystems, the data is first copied form the source filesystem to the local cluster before being copied to the destination filesystem.

Using DistCp with Microsoft Azure (WASB)

You can use the `distcp` command to copy data from Azure WASB to your cluster :

1. Configure connectivity to Azure by setting the following property in `core-site.xml`.

```
<property>
  <name>fs.azure.account.key.youraccount.blob.core.windows.net</name>
  <value>your_access_key</value>
</property>
```

   Note that in practice, you should never store your Azure access key in cleartext. Protect your Azure credentials using one of the methods described at <u>Configuring Azure Blob Storage Credentials</u>.

2. Run your `distcp` jobs using the following syntax:

```
hadoop distcp wasb://<sample_container>@<sample_account>.blob.core.windows.net/
/hdfs_destination_path
```

Reference

- Upstream Hadoop documentation on <u>Hadoop Support for Azure</u>

Kerberos Setup Guidelines for Distcp between Secure Clusters

The guidelines mentioned in this section are only applicable for the following example deployment:

- You have two clusters, each in a different Kerberos realm (SOURCE and DESTINATION in this example)
- You have data that needs to be copied from SOURCE to DESTINATION

- A Kerberos realm trust exists, either between `SOURCE` and `DESTINATION` (in either direction), or between both `SOURCE` and `DESTINATION` and a common third realm (such as an Active Directory domain).

If your environment matches the one described above, use the following table to configure Kerberos delegation tokens on your cluster so that you can successfully `distcp` across two secure clusters. Based on the direction of the trust between the `SOURCE` and `DESTINATION` clusters, you can use the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to instruct ResourceManagers on either cluster to skip or perform delegation token renewal for NameNode hosts.

> **Note:** For CDH 5.12.0 and later, you must use the `mapreduce.job.hdfs-servers.token-renewal.exclude` parameter if both clusters use the HDFS Transparent Encryption feature.

Environment Type		Kerberos Delegation Token Setting
`SOURCE` trusts `DESTINATION`	Distcp job runs on the `DESTINATION` cluster	You do not need to set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property.
	Distcp job runs on the `SOURCE` cluster	Set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to a comma-separated list of the hostnames of the NameNodes of the `DESTINATION` cluster.
`DESTINATION` trusts `SOURCE`	Distcp job runs on the `DESTINATION` cluster	Set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to a comma-separated list of the hostnames of the NameNodes of the `SOURCE` cluster.
	Distcp job runs on the `SOURCE` cluster	You do not need to set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property.
Both `SOURCE` and `DESTINATION` trust each other	Set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to a comma-separated list of the hostnames of the NameNodes of the `DESTINATION` cluster.	

Neither `SOURCE` nor `DESTINATION` trusts the other	If a common realm is usable (such as Active Directory), set the `mapreduce.job.hdfs-servers.token-renewal.exclude` property to a comma-separated list of hostnames of the NameNodes of the cluster that is *not* running the distcp job. For example, if you are running the job on the `DESTINATION` cluster:
	
	 1. `kinit` on any `DESTINATION` YARN Gateway host using an AD account that can be used on both `SOURCE` and `DESTINATION`.
	 2. Run the distcp job as the `hadoop` user:
	
	 ``
	$ hadoop distcp
	-Ddfs.namenode.kerberos.principal.pattern=* \
	-Dmapreduce.job.hdfs-servers.token-renewal.exclude=SOURCE-nn-host1,SOURCE-nn-host2 \
	hdfs://source-nn-nameservice/source/path \
	/destination/path
	``
	
	 By default, the YARN ResourceManager renews tokens for applications. The `mapreduce.job.hdfs-servers.token-renewal.exclude` property instructs ResourceManagers on either cluster to skip delegation token renewal for NameNode hosts.	

Distcp between Secure Clusters in Different Kerberos Realms

> **Important:** To use DistCp between two secure clusters in different Kerberos realms, you must use a single Kerberos principal that can authenticate to both realms. In other words, a Kerberos realm trust relationship must exist between the source and destination realms. This can be a one-way trust (in either direction), a bi-directional trust, or even multiple one-way trusts where both the source and destination realms trust a third realm (such as an Active Directory domain).
>
> If there is no trust relationship between the source and destination realms, you cannot use DistCp to copy data between the clusters, but you can use Cloudera Backup and Data Recovery (BDR). For more information, see Enabling Replication Between Clusters with Kerberos Authentication on page 514.
>
> Additionally, both clusters must run a supported JDK version. For information about supported JDK versions, see Cloudera Enterprise 6 Requirements and Supported Versions.

This section explains how to copy data between two secure clusters in different Kerberos realms:

Configure Source and Destination Realms in `krb5.conf`

Make sure that the `krb5.conf` file on the client (from which the `distcp` job is submitted) includes realm definitions and mappings for both source and destination realms. For example:

```
[realms]
 QA.EXAMPLE.COM = {
  kdc = kdc01.qa.example.com:88
  admin_server = kdc01.qa.example.com:749
 }

 DEV.EXAMPLE.COM = {
  kdc = kdc01.dev.example.com:88
  admin_server = kdc01.dev.example.com:749
 }

[domain_realm]
.qa.example.com = QA.EXAMPLE.COM
qa.example.com = QA.EXAMPLE.COM
.dev.example.com = DEV.EXAMPLE.COM
dev.example.com = DEV.EXAMPLE.COM
```

Configure HDFS RPC Protection and Acceptable Kerberos Principal Patterns

Set the `hadoop.rpc.protection` property to `authentication` in both clusters. You can modify this property either in `hdfs-site.xml`, or using Cloudera Manager as follows:

1. Open the Cloudera Manager Admin Console.
2. Go to the HDFS service.
3. Click the **Configuration** tab.
4. Select **Scope** > **HDFS-1 (Service-Wide)**.
5. Select **Category** > **Security**.
6. Locate the **Hadoop RPC Protection** property and select `authentication`.
7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

The following steps are not required if the two realms are already set up to trust each other, or have the same principal pattern. However, this isn't usually the case.

Set the `dfs.namenode.kerberos.principal.pattern` property to * to allow distcp irrespective of the principal patterns of the source and destination clusters. You can modify this property either in `hdfs-site.xml` on both clusters, or using Cloudera Manager as follows:

1. Open the Cloudera Manager Admin Console.
2. Go to the HDFS service.
3. Click the **Configuration** tab.
4. Select **Scope** > **Gateway**.

5. Select **Category** > **Advanced**.

6. Edit the **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** property to add:

```
<property>
   <name>dfs.namenode.kerberos.principal.pattern</name>
   <value>*</value>
</property>
```

7. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.

(If TLS/SSL is enabled) Specify Truststore Properties

The following properties must be configured in the `ssl-client.xml` file on the client submitting the distcp job to establish trust between the target and destination clusters.

```
<property>
<name>ssl.client.truststore.location</name>
<value>path_to_truststore</value>
</property>

<property>
<name>ssl.client.truststore.password</name>
<value>XXXXXX</value>
</property>

<property>
<name>ssl.client.truststore.type</name>
<value>jks</value>
</property>
```

Set `HADOOP_CONF` to the Destination Cluster

Set the `HADOOP_CONF` path to be the destination environment. If you are not using HFTP, set the `HADOOP_CONF` path to the source environment instead.

Launch Distcp

Authenticate using `kinit` on the client and then launch the `distcp` job. For example:

```
hadoop distcp hdfs://xyz01.dev.example.com:8020/user/alice
hdfs://abc01.qa.example.com:8020/user/alice
```

If launching `distcp` fails, force Kerberos to use TCP instead of UDP by adding the following parameter to the `krb5.conf` file on the client.

```
[libdefaults]
udp_preference_limit = 1
```

Enabling Fallback Configuration

To enable the fallback configuration, for copying between secure and insecure clusters, add the following to the HDFS configuration file, `core-default.xml`, by using an advanced configuration snippet if you use Cloudera Manager, or editing the file directly otherwise.

```
<property>
   <name>ipc.client.fallback-to-simple-auth-allowed</name>
   <value>true</value>
</property>
```

Protocol Support for Distcp

To use `distcp` to copy data between CDH 6 clusters, use the `hdfs` protocol. When you are copying data between two different but compatible versions, use the `webhdfs` protocol for the remote cluster.

Copying Data between a Secure and an Insecure Cluster using DistCp and WebHDFS

You can use DistCp and WebHDFS to copy data between a secure cluster and an insecure cluster. Note that when doing this, the `distcp` commands should be run from the secure cluster. by doing the following:

1. On the secure cluster, set `ipc.client.fallback-to-simple-auth-allowed` to true in core-site.xml:

```
<property>
  <name>ipc.client.fallback-to-simple-auth-allowed</name>
  <value>true</value>
</property>
```

   Alternatively, you can also pass this as a parameter when you run the `distcp` command. If you want to do that, move onto step 2.

2. On the insecure cluster, add the secured cluster's realm name to the insecure cluster's configuration:

   a. In the Cloudera Manager Admin Console for the insecure cluster, navigate to **Clusters** > **<HDFS cluster>**.
   b. On the **Configuration** tab, search for **Trusted Kerberos Realms** and add the secured cluster's realm name.

      Note that his does not require Kerberos to be enabled but is a necessary step to allow the simple auth fallback to happen in the `hdfs://` protocol.

   c. Save the change.

3. Use commands such as the following *from the secure cluster side only*:

```
#This example uses the insecure cluster as the source and the secure cluster as the
destination
distcp webhdfs://<insecure_namenode>:50070 webhdfs://<secure_namenode>:50470

#This example uses the sefcure cluster as the source and the insecure cluster as the
destination
distcp webhdfs://<secure_namenode>:500470 webhdfs://<insecure_namenode>:50070
```

   If TLS is enabled, replace `webhdfs` with `swebhdfs`.

   If you did not configure `ipc.client.fallback-to-simple-auth-allowed` and want to pass it as a parameter, run commands such as the following from the secure cluster:

```
#This example uses the insecure cluster as the source and the secure cluster (with TLS
 enabled) as the destination cluster. swebhdfs is used instead of webhdfs when TLS is
enabled.
hadoop distcp -D ipc.client.fallback-to-simple-auth-allowed=true
webhdfs://<insecure_namenode>:50070 swebhdfs://<secure_namenode>:50470

#This example uses the secure cluster (with TLS enabled) as the source cluster and the
 insecure cluster as the destination. swebhdfs is used instead of webhdfs when TLS is
enabled.
hadoop distcp -D ipc.client.fallback-to-simple-auth-allowed=true
swebhdfs://<secure_namenode>:50470 webhdfs://<insecure_namenode>:50070
```

Post-migration Verification

After migrating data between the two clusters, it is a good idea to use `hadoop fs -ls /basePath` to verify the permissions, ownership and other aspects of your files, and correct any problems before using the files in your new cluster.

Backing Up Databases

Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data and for managed services that require a database:

- Cloudera Manager Server - Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes. This relatively small database (< 100 MB) is the most important to back up.

  > **Important:** When you restart processes, the configuration for each of the services is redeployed using information saved in the Cloudera Manager database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the Cloudera Manager database to recover the cluster in the event of the loss of this database. For more information, see Backing Up Databases on page 586.

- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large.
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small.
- Activity Monitor - Contains information about past activities. In large clusters, this database can grow large. Configuring an Activity Monitor database is only necessary if a MapReduce service is deployed.
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Hue Server - Contains user account information, job submissions, and Hive queries. Relatively small.
- Sentry Server - Contains authorization metadata. Relatively small.
- Cloudera Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.
- Cloudera Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small.

Backing Up PostgreSQL Databases

To back up a PostgreSQL database, use the same procedure whether the database is embedded or external:

1. Log in to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from `/etc/cloudera-scm-server/db.properties`:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root using the parameters from the preceding step:

```
# pg_dump -h hostname -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +%Y%m%d)
```

4. Enter the password from the `com.cloudera.cmf.db.password` property in step 2.
5. To back up a database created for one of the roles described in Creating Databases for Cloudera Software, on the local host as the *roleuser* user:

```
# pg_dump -h hostname -p 7432 -U roleuser > /tmp/roledb
```

6. Enter the password specified when the database was created.

Backing Up MariaDB Databases

To back up the MariaDB database, run the `mysqldump` command on the MariaDB host, as follows:

```
$ mysqldump -hhostname -uusername -ppassword database > /tmp/database-backup.sql
```

For example, to back up the Activity Monitor database `amon` created in [Creating Databases for Cloudera Software](), on the local host as the root user, with the password `amon_password`:

```
$ mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database `amon` on remote host `myhost.example.com` as the root user, with the password `amon_password`:

```
$ mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

Backing Up MySQL Databases

To back up the MySQL database, run the `mysqldump` command on the MySQL host, as follows:m

```
$ mysqldump -hhostname -uusername -ppassword database > /tmp/database-backup.sql
```

For example, to back up the Activity Monitor database `amon` created in [Creating Databases for Cloudera Software](), on the local host as the root user, with the password `amon_password`:

```
$ mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database `amon` on remote host `myhost.example.com` as the root user, with the password `amon_password`:

```
$ mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backup.sql
```

You can back up all database using the following command:

```
mysqldump --all-databases -ppassword > /tmp/all1/all.sql
```

Backing Up Oracle Databases

For Oracle, work with your database administrator to ensure databases are properly backed up.

Database Vendor Resources

Use the following links to access vendor documentation on backing up and restoring databases.

- **MariaDB 5.5:** http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- **MySQL 5.5:** http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- **MySQL 5.6:** http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- **PostgreSQL 8.4:** https://www.postgresql.org/docs/8.4/static/backup.html
- **PostgreSQL 9.2:** https://www.postgresql.org/docs/9.2/static/backup.html
- **PostgreSQL 9.3:** https://www.postgresql.org/docs/9.3/static/backup.html
- **Oracle 11gR2:** http://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm

Cloudera Navigator Administration

The Cloudera Navigator Data Management component encompasses two distinct roles (service daemons) that run on the Cloudera Management Service on page 53:

- Navigator Audit Server, which tracks, coalesces, and stores events in its database.
- Navigator Metadata Server, which handles all metadata management for the system and supports the policies, user authorization, analytic data, and many other capabilities. The Navigator Metadata Server also exposes the Cloudera Navigator APIs and hosts the Cloudera Navigator console.

These roles are set up and configured typically during the Cloudera Manager installation process as detailed in Installing the Cloudera Navigator Data Management Component.

Many of the on-going administration tasks for the Cloudera Navigator component, specifically the Cloudera Audit Server role and Cloudera Metadata Server role, are handled using the Cloudera Manager Admin Console and require Cloudera Manager Full Administrator or Navigator Administrator privileges. That is, you must log in to the Cloudera Manager Admin Console with an account that has either of those user roles. See Cloudera Navigator User Roles and Privileges reference for details.

See the Cloudera Security guide for these topics:

- Configuring Authentication for Cloudera Navigator
    - Cloudera Navigator and External Authentication
    - Configuring Cloudera Navigator for Active Directory
    - Configuring Cloudera Navigator for LDAP
    - Configuring Cloudera Navigator for SAML

- Configuring TLS/SSL for Navigator Audit Server
- Configuring TLS/SSL for Navigator Metadata Server

For details about system architecture and for configuring, tuning, and systems management, see the Cloudera Navigator Data Management guide, specifically:

- Navigator Audit Server Management
- Navigator Metadata Server Management

Get Started with Amazon S3

The following topics can help you deploy, configure, manage, and secure clusters in the cloud using Amazon S3:

Administration or Setup Tasks

- Configuring the Amazon S3 Connector
- Configuring Transient Hive ETL Jobs to Use the Amazon S3 Filesystem
- How to Configure AWS Credentials
- How to Configure Security for Amazon S3
- Configuring and Managing S3Guard
- Using DistCp with Amazon S3

Component Tasks

Backup and Disaster Recovery

- HDFS Replication To and From Amazon S3
- Hive Replication To and From Amazon S3

Cloudera Navigator

- Cloudera Navigator and S3
- S3 Data Extraction for Navigator

Hue

- How to Enable S3 Cloud Storage
- How to Use S3 as Source or Sink

Hive

- Tuning Apache Hive Performance on the Amazon S3 Filesystem in CDH

Impala

- Using Impala with the Amazon S3 Filesystem
- Specifying Impala Credentials to Access Data in S3
- Specifying Impala Credentials to Access Data in S3 with Cloudera Manager

Spark, YARN, MapReduce, Oozie

- Accessing Data Stored in Amazon S3 through Spark
- Configuring MapReduce to Read/Write with Amazon Web Services
- Configuring Oozie to Enable MapReduce Jobs to Read/Write from Amazon S3
- Using S3 Credentials with YARN, MapReduce, or Spark
- How to Configure a MapReduce Job to Access S3 with an HDFS Credstore

Configuring the Amazon S3 Connector

You can securely configure your cluster to authenticate with Amazon Simple Storage Service (S3) using the Cloudera **S3 Connector Service**. This configuration enables Impala queries to access data in S3 and also enables the Hue S3 Browser. Impala and Hue are automatically configured to authenticate with S3, but applications such as YARN, MapReduce, or Spark must provide their own AWS credentials when submitting jobs. You can define only one Amazon S3 service for each cluster.

Cloudera Manager stores these values securely and does not store them in world-readable locations. The credentials are masked in the Cloudera Manager Admin console, encrypted in the configurations passed to processes managed by Cloudera Manager, and redacted from the logs.

To access this storage, you define AWS Credentials in Cloudera Manager, and then you add the S3 Connector Service and configure it to use the AWS credentials.

Consider using the **S3Guard** feature to address possible issues with the "eventual consistency" guarantee provided by Amazon for data stored in S3. To use the S3Guard feature, you provision an Amazon DynamoDb for use as an additional metadata store to improve performance and guarantee that your queries return the most current data. See Configuring and Managing S3Guard on page 594.

Adding AWS Credentials

Minimum Required Role: User Administrator (also provided by **Full Administrator**)

To connect to Amazon S3, obtain an Access Key and Secret Key from Amazon Web Services, and then add AWS credentials in Cloudera Manager. These keys should permit access to all data in S3 that you want to query with Impala or browse with Hue.

Managing AWS Credentials in Cloudera Manager

To edit or remove AWS credentials:

1. Open Cloudera Manager and go to **Administration > External Accounts**.
2. Select the **AWS Credentials** tab.
3. To remove a credential, in the row for the credential you want to change, click **Actions** > **Remove**.

   You cannot remove a credential that is currently being used by the S3 Connector Service; you must first remove the Connector Service.

4. To edit a credential, in the row for the credential you want to edit, click **Actions** > **Edit Credential**.
5. Edit the fields of the credential as needed and click **Save**.

Adding the S3 Connector Service

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

> **Important:**
> - If all hosts are configured with IAM Role-based Authentication that allows access to S3 and you do not want to use S3Guard, you do not need to add the **S3 Connector Service**.
> - When using the More Secure mode, you must have the Sentry service and Kerberos enabled for the cluster in order add the S3 Connector Service. For secure operation, Cloudera also recommends that you enable TLS for Cloudera Manager.
> - A cluster cannot use the S3 Connector Service and the ADLS Connector Service at the same time. You must remove the old connector service before adding a new one. See Removing the ADLS Connector Service on page 602 or Removing the S3 Connector Service on page 591.

To add the S3 Connector Service using the Cloudera Manager Admin Console:

1. If you have not defined AWS Credentials, add AWS credentials in Cloudera Manager.
2. Go to the cluster where you want to add the Amazon S3 Connector Service.
3. Click **Actions** > **Add Service**.
4. Select **S3 Connector**.
5. Click **Continue**.

   The **Add S3 Connector Service to *Cluster Name*** wizard displays.

The wizard checks your configuration for compatibility with S3 and reports any issues. The wizard does not allow you to continue if you have an invalid configuration. Fix any issues, and then repeat these steps to add the S3 Connector Service.

6. Select a **Credentials Protection Policy**. (Not applicable when **IAM Role-Based Authentication** is used.)

   Choose one of the following:

   • **Less Secure**

     Credentials can be stored in plain text in some configuration files for specific services (currently Hive, Impala, and Hue) in the cluster.

     This configuration is appropriate for unsecure, single-tenant clusters that provide fine-grained access control for data stored in S3.

   • **More Secure**

     Cloudera Manager distributes secrets to a limited set of services (currently Impala and Hue) and enables those services to access S3. It does not distribute these credentials to any other clients or services. See S3 Credentials Security.

   Other configurations that are not sensitive, such as the S3Guard configuration, are included in the configuration of all services and clients as needed.

7. Click **Continue**.
8. Select previously-defined AWS credentials from the **Name** drop-down list.
9. Click **Continue**.

   The **Restart Dependent Services** page displays and indicates the dependent services that need to be restarted.

10. Select **Restart Now** to restart these services. You can also restart these services later. Impala and Hue will not be able to authenticate with S3 until you restart the services.
11. Click **Continue** to complete the addition of the Amazon S3 service. If **Restart Now** is selected, the dependent services are restarted.

Removing the S3 Connector Service

To remove the service:

1. Open Cloudera Manager and go to **Administration** > **External Accounts**.
2. Select the **AWS Credentials** tab.
3. In the row for the credential used for the service, click **Actions** > **Edit Connectivity**.

   The **Connect to Amazon Web Services** dialog box displays.

4. Click **Disable for *Cluster_name***.
5. Click **OK**.

   A message displays saying "The configuration has been updated". You will need to restart any stale services. Click the **View Stale Configurations** link to open the **Stale Configurations page.** Click **Restart Stale Services**.

Using S3 Credentials with YARN, MapReduce, or Spark

This topic describes how to access data stored in S3 for applications that use YARN, MapReduce, or Spark.

You can also copy data using the Hadoop `distcp` command. See Using DistCp with Amazon S3 on page 578.

Referencing Credentials for Clients Using the Amazon S3 Service

If you have selected IAM authentication, no additional steps are needed. If you are not using IAM authentication, use one of the following three options to provide Amazon S3 credentials to clients:

> **Note:** This method of specifying AWS credentials to clients does not completely distribute secrets securely because the credentials are not encrypted. Use caution when operating in a multi-tenant environment.

Programmatic

Specify the credentials in the configuration for the job. This option is most useful for Spark jobs.

Make a modified copy of the configuration files

Make a copy of the configuration files and add the S3 credentials:

1. For YARN and MapReduce jobs, copy the contents of the `/etc/hadoop/conf` directory to a local working directory under the home directory of the host where you will submit the job. For Spark jobs, copy `/etc/spark/conf` to a local directory under the home directory of the host where you will submit the job.
2. Set the permissions for the configuration files appropriately for your environment and ensure that unauthorized users cannot access sensitive configurations in these files.
3. Add the following to the `core-site.xml` file within the `<configuration>` element:

```
<property>
    <name>fs.s3a.access.key</name>
    <value>Amazon S3 Access Key</value>
</property>

<property>
    <name>fs.s3a.secret.key</name>
    <value>Amazon S3 Secret Key</value>
</property>
```

4. Reference these versions of the configuration files when submitting jobs by running the following command:

- **YARN or MapReduce:**

```
export HADOOP_CONF_DIR=path to local configuration directory
```

- **Spark:**

```
export SPARK_CONF_DIR=path to local configuration directory
```

> **Note:** If you update the client configuration files from Cloudera Manager, you must repeat these steps to use the new configurations.

Reference the managed configuration files and add AWS credentials

This option allows you to continue to use the configuration files managed by Cloudera Manager. If you deploy new configuration files, the new values are included by reference in your copy of the configuration files while also maintaining a version of the configuration that contains the Amazon S3 credentials:

1. Create a local directory under your home directory.
2. Copy the configuration files from `/etc/hadoop/conf` to the new directory.
3. Set the permissions for the configuration files appropriately for your environment.
4. Edit each configuration file:

    a. Remove all elements within the `<configuration>` element.
    b. Add an XML `<include>` element within the `<configuration>` element to reference the configuration files managed by Cloudera Manager. For example:

```
<include xmlns="http://www.w3.org/2001/XInclude"
         href="/etc/hadoop/conf/hdfs-site.xml">
```

```
    <fallback />
</include>
```

5. Add the following to the `core-site.xml` file within the `<configuration>` element:

```
<property>
    <name>fs.s3a.access.key</name>
    <value>Amazon S3 Access Key</value>
</property>

<property>
    <name>fs.s3a.secret.key</name>
    <value>Amazon S3 Secret Key</value>
</property>
```

6. Reference these versions of the configuration files when submitting jobs by running the following command:

- **YARN or MapReduce:**

```
export HADOOP_CONF_DIR=path to local configuration directory
```

- **Spark:**

```
export SPARK_CONF_DIR=path to local configuration directory
```

Example `core-site.xml` file:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <include xmlns="http://www.w3.org/2001/XInclude"
    href="/etc/hadoop/conf/core-site.xml">
    <fallback />
  </include>

  <property>
    <name>fs.s3a.access.key</name>
    <value>Amazon S3 Access Key</value>
  </property>

  <property>
    <name>fs.s3a.secret.key</name>
    <value>Amazon S3 Secret Key</value>
  </property>
</configuration>
```

Referencing Amazon S3 in URIs

By default, files are still placed on the local HDFS and not on S3 if the protocol is not specified in the URI. When you have added the Amazon S3 service, use one of the following options to construct the URIs to reference when submitting jobs:

- **Amazon S3**:

```
s3a://bucket_name/path
```

- **HDFS**:

```
hdfs://path
```

or

```
/path
```

For more information about using Impala, Hive, and Spark on S3, see:

- [Using Impala with the Amazon S3 Filesystem](#)
- [Tuning Apache Hive Performance on the Amazon S3 Filesystem in CDH](#)
- [Accessing Data Stored in Amazon S3 through Spark](#)

Using Fast Upload with Amazon S3

Writing data to Amazon S3 is subject to limitations of the s3a `OutputStream` implementation, which buffers the entire file to disk before uploading it to S3. This can cause the upload to proceed very slowly and can require a large amount of temporary disk space on local disks.

You can configure CDH to use the Fast Upload feature. This feature implements several performance improvements and has tunable parameters for buffering to disk (the default) or to memory, tuning the number of threads, and for specifying the disk directories used for buffering.

For more information on this feature, and to learn about the tunable parameters, see [Hadoop-AWS module: Integration with Amazon Web Services](#) .

Enabling Fast Upload using Cloudera Manager

To enable Fast Upload for clusters managed by Cloudera Manager:

1. Go to the HDFS service.
2. Select the **Configuration** tab.
3. Search for "core-site.xml" and locate the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** property.
4. Add the `fs.s3a.fast.upload` property and set it to `true`. See [Setting an Advanced Configuration Snippet](#) on page 76.
5. Set any additional tuning properties in the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** configuration properties.
6. Click **Save Changes**.

    Cloudera Manager will indicate that there are stale services and which services need to be restarted. [Restart the indicated services](#).

Configuring and Managing S3Guard

Minimum Required Role: [User Administrator](#) (also provided by **Full Administrator**)

Data written to Amazon S3 buckets is subject to the "eventual consistency" guarantee provided by Amazon Web Services (AWS), which means that data written to S3 may not be immediately available for queries and listing operations. This can cause failures in multi-step ETL workflows, where data from a previous step is not available to the next step. The S3Guard feature guarantees a consistent view of data stored in Amazon S3 by storing additional metadata in a table residing in an Amazon DynamoDB instance. Depending on the workload, this additional metadata store may also improve performance for Hive, Spark, and Impala jobs.

All processes that modify the S3 bucket that S3Guard is enabled for must use S3Guard. Since S3Guard works by logging metadata changes to an external database, modifying the bucket outside of S3Guard will cause the S3 data and the S3Guard database to go out of sync. This can cause issues such as S3A/S3Guard thinking that files are or are not present despite the bucket having different data.

To enable S3Guard, you set up an Amazon DynamoDB database from Amazon Web Services. Amazon charges an hourly rate for this service based on the capacity you provision. See [Amazon DynamoDB Pricing](#).

When the data stored in S3 eventually becomes consistent (usually within 24 hours or less), the S3Guard metadata is no longer required and you can periodically [prune the S3Guard Metadata](#) stored in the DynamoDB to clear older entries. Pruning can also reduce costs associated with the DynamoDB.

To configure S3Guard in your cluster, you must provide the following:

- Credentials for the [Amazon S3](#) bucket.
- An instance of [Amazon DynamoDB](#) database provisioned from Amazon Web Services.
- The [configured region](#) for the DynamoDB database.
- A CDH cluster managed by Cloudera Manager.

Configuring S3Guard for Cluster Access to S3

1. Specify the AWS credentials for the Amazon S3 instance where you want to enable S3Guard. You can:

   - [Add a new AWS credential](#).

     After adding the credential, the **Edit S3Guard** dialog box displays.

   - Use an existing AWS credential:

     - 1. Go to **Administration** > **AWS Credentials**.
       2. Locate the credential you want to use and click **Actions** > **Edit S3Guard**.

          The **Edit S3Guard** dialog box displays.

2. Select **Enable S3Guard**.
3. Edit the following S3Guard configuration properties:

Table 35: S3Guard Configuration Properties

Property	Description
Automatically Create S3Guard Metadata Table (fs.s3a.s3guard.ddb.table.create) **API Name:** s3guard_table_auto_create	When Yes is selected, the DynamoDB table that stores the S3Guard metadata is automatically created if it does not exist. When No is selected and the table does not exist, running the Prune command, queries, or other jobs on S3 will fail.
S3Guard Metadata Table Name (fs.s3a.s3guard.ddb.table) **API Name:** s3guard_table_name	The name of the DynamoDB table that stores the S3Guard metadata. By default, the table is named s3guard-metadata.
S3Guard Metadata Region Name (fs.s3a.s3guard.ddb.region) **API Name:** s3guard_region	The DynamoDB region to connect to for access to the S3Guard metadata. Set this property to a valid region. See [DynamoDB regions](#).
Expand the **Advanced** section to configure the following properties:	
S3Guard Metadata Pruning Age (fs.s3a.s3guard.cli.prune.age) **API Name:** s3guard_cache_prune_age_ms	Maximum age for S3Guard metadata. Whenever the Prune command runs, entries in the S3Guard metadata cache older than this age will be deleted. You can enter this value in milliseconds, seconds, minutes, hours, or days.

Property	Description
S3Guard Metadata Table Read Capacity (fs.s3a.s3guard.ddb.table.capacity.read) **API Name:** s3guard_table_capacity_read	Provisioned throughput requirements, in capacity units, for read operations from the DynamoDB table used for the S3Guard metadata. This value is only used when creating a new DynamoDB table. After the table is created, you can monitor the throughput and adjust the read capacity using the <u>DynamoDB AWS Management Console</u>. See <u>Provisioned Throughput</u>.
S3Guard Metadata Table Write Capacity (fs.s3a.s3guard.ddb.table.capacity.write) **API Name:** s3guard_table_capacity_write	Provisioned throughput requirements, in capacity units, for write operations to the DynamoDB table used for the S3Guard metadata. This value is only used when creating a new DynamoDB table. After the table is created, you can monitor the throughput and adjust the write capacity as needed using the <u>DynamoDB AWS Management Console</u>. See <u>Provisioned Throughput</u>.

4. Click **Save**.

   The **Connect to Amazon Web Services** dialog box displays.

5. To enable cluster access to S3 using the **S3 Connector Service**, click the **Enable for *Cluster Name*** link in the **Cluster Access to S3** section.

   Follow the prompts to add the **S3 Connector Service**. See <u>Adding the S3 Connector Service</u> on page 590 for details.

> ✎ **Note:** S3Guard is not supported for **Cloud Backup and Restore** and **Cloudera Navigator Access to S3**.

Editing the S3Guard Configuration

To edit or disable the S3Guard configuration:

1. Click **Administration** > **AWS Credentials.**
2. Locate the credential associated with the S3Guard configuration and click **Actions** > **Edit S3Guard**.

   The **Edit S3Guard** dialog box displays.

3. Edit the S3Guard configuration. (To disable S3Guard for this credential, uncheck **Enable S3Guard**.)
4. Click **Save**.

Pruning the S3Guard Metadata

Amazon charges for the amount of data stored in the DynamoDB and the bandwidth used for reads and writes to the database. To optimize costs and improve performance, you can remove stale metadata from the DynamoDB table by running the **Prune** command. Generally, data written to S3 becomes consistent after 24 hours or less, meaning that you only need to maintain metadata in DynamoDB for about one day. You can <u>monitor the usage of DynamoDB using AWS tools</u> to determine how often and when to prune the table.

Running the **Prune** command removes all metadata that is older than the age you specify with the **S3Guard Metadata Pruning Age** property in the S3Guard configuration. You can run this command from the Cloudera Manager Admin Console, or you can create a script to run the Prune command automatically using the Cloudera Manager API. Cloudera recommends that you run that script using a Linux cron job or other scheduling mechanism to regularly prune the metadata.

Running the Prune Command Using Cloudera Manager Admin Console

Minimum Required Role: <u>Cluster Administrator</u> (also provided by **Full Administrator**)

To prune the S3Guard metadata in the DynamoDB table using the Cloudera Manager Admin Console:

1. Go to **Administration** > **AWS Credentials**.
2. Locate the credential associated with the S3 data and click **Actions** > **Run S3 Guard Prune Command**.

Running the Prune Command Using the Cloudera Manager API

Cloudera recommends that you automate running the Prune command by creating a script that uses the Cloudera Manager API to run the command. You can run the command using a REST command, a Python script, or Java class. Configure the script using the Linux `cron` command or another scheduling mechanism to run on a regular schedule.

REST

See the [Rest API Documentation](#).

You can run the Prune command by issuing the following REST request:

```
curl -X POST -u username:password

'Cloudera_Manager_server_URL:port_number/api/vAPI_version_number/externalAccounts/account/Credential_Name/commands/S3GuardPrune'
```

For example, the following request runs the S3Guard prune command on the data associated with the `johnsmith` credential. The response from Cloudera Manager is also displayed (within the curly brackets):

```
curl -X POST -u admin:admin
'http://clusterhost-1.gce.mycompany.com:7180/api/v16/externalAccounts/account/johnsmith/commands/S3GuardPrune'
{
  "id" : 322,
  "name" : "S3GuardPrune",
  "startTime" : "2017-03-20T23:35:55.453Z",
  "active" : true,
  "children" : {
    "items" : [ {
      "id" : 323,
      "name" : "HostS3GuardPrune",
      "startTime" : "2017-03-20T23:35:55.777Z",
      "active" : true,
      "hostRef" : {
        "hostId" : "ff988a15-3749-4178-b167-a60b15f91653"
      }
```

Python

You can also use a Python script to run the Prune command. See [aws.py](#) for the code and usage instructions.

Java

See the [Javadoc](#).

How to Configure a MapReduce Job to Access S3 with an HDFS Credstore

This topic describes how to configure your MapReduce jobs to read and write to Amazon S3 using a custom password for an HDFS Credstore.

1. Copy the contents of the `/etc/hadoop/conf` directory to a local working directory on the host where you will submit the MapReduce job. Use the `--dereference` option when copying the file so that symlinks are correctly resolved. For example:

```
cp -r --dereference /etc/hadoop/conf ~/my_custom_config_directory
```

2. Change the permissions of the directory so that only you have access:

```
chmod go-wrx -R my_custom_config_directory/
```

If you see the following message, you can ignore it:

```
cp: cannot open `/etc/hadoop/conf/container-executor.cfg' for reading: Permission denied
```

3. Add the following to the copy of the `core-site.xml` file in the working directory:

```
<property>
  <name>hadoop.security.credential.provider.path</name>
  <value>jceks://hdfs/user/username/awscreds.jceks</value>
</property>
```

4. Specify a custom Credstore by running the following command on the client host:

```
export HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password
```

5. In the working directory, edit the `mapred-site.xml` file:

    a. Add the following properties:

```
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password</value>
</property>

<property>
  <name>mapred.child.env</name>
  <value>HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password</value>
</property>
```

    b. Add `yarn.app.mapreduce.am.env` and `mapred.child.env` to the comma-separated list of values of the `mapreduce.job.redacted-properties` property. For example (new values shown **bold**):

```
<property>
  <name>mapreduce.job.redacted-properties</name>

<value>fs.s3a.access.key,fs.s3a.secret.key,yarn.app.mapreduce.am.env,mapred.child.env</value>
</property>
```

6. Set the environment variable to point to your working directory:

```
export HADOOP_CONF_DIR=~/path_to_working_directory
```

7. Create the Credstore by running the following commands:

```
hadoop credential create fs.s3a.access.key
hadoop credential create fs.s3a.secret.key
```

You will be prompted to enter the access key and secret key.

8. List the credentials to make sure they were created correctly by running the following command:

```
hadoop credential list
```

9. Submit your job. For example:

- `ls`

```
hdfs dfs -ls s3a://S3_Bucket/
```

- `distcp`

```
hadoop distcp hdfs_path s3a://S3_Bucket/S3_path
```

- `teragen` (package-based installations)

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar teragen 100
s3a://S3_Bucket/teragen_test
```

- `teragen` (parcel-based installations)

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
  teragen 100 s3a://S3_Bucket/teragen_test
```

Getting Started with Microsoft Azure Data Lake Storage (ADLS)

These topics focused on Microsoft ADLS from the core Cloudera Enterprise documentation library can help you deploy, configure, manage, and secure clusters in the cloud. They are listed by broad category:

- Administration or Setup Tasks
- Component-Specific Tasks

Administration or Setup Tasks

- Configuring ADLS Access Using Cloudera Manager on page 600

Component Tasks

DistCp

- Using DistCp with Microsoft Azure (ADLS) on page 580

Hive

- Using Microsoft Azure Data Lake Store with Apache Hive in CDH

Impala

- Using Impala with the Azure Data Lake Store (ADLS)

Oozie

- Configuring Oozie to Enable MapReduce Jobs To Read/Write from Microsoft Azure (ADLS)

Spark, YARN, MapReduce

- Configuring ADLS Connectivity on page 603
- Best Practices for Spark Streaming in the Cloud
- Accessing Data Stored in Azure Data Lake Store (ADLS) through Spark
- Using Spark with Azure Data Lake Storage (ADLS)

Configuring ADLS Access Using Cloudera Manager

Minimum Required Role: User Administrator (also provided by **Full Administrator**)

To configure access to Microsoft Azure Data Lake Store (ADLS) using Cloudera Manager, you configure ADLS credentials using the Cloudera Manager **External Accounts** page and then add the ADLS Connector Service to your cluster. Adding the ADLS Connector Service allows users and administrators of the cluster to seamlessly and securely access ADLS in the following ways:

- Run Hive and Impala queries on tables backed by data stored in ADLS.
- Browse ADLS stores using Hue.

Other cluster services may also gain access to ADLS by having their users provide their own credentials directly using the Hadoop Credential Provider mechanism.

> ⚠ **Important:** You cannot use ADLS as a source or destination for Backup and Disaster recovery or to enable lineage or metadata extraction using Cloudera Navigator.

When you configure credentials using Cloudera Manager, it provides a more secure way to access ADLS using credentials that are not stored in plain-text files. The client configuration files generated by Cloudera Manager based on configured services do not include ADLS credentials. Command-line and API clients must manage access to these credentials outside of Cloudera Manager. Cloudera Manager provides credentials directly to trusted clients such as the Impala daemon and Hue. For access from YARN, MapReduce or Spark, see Configuring ADLS Connectivity on page 603.

Configuring ADLS Credentials in Cloudera Manager

If you have already created your ADLS account and configured ADLS credentials in Cloudera Manager, skip this section and continue with Adding the ADLS Connector Service on page 601.

1. Create your ADLS account. See the Microsoft documentation.
2. Create the Active Directory service principal in the Azure portal. See the Microsoft documentation on creating a service principal. You will need the following to configure ADLS credentials in Cloudera Manager:

   • Client ID
   • Client secret
   • Tenant ID

3. Grant the service principal permission to access the ADLS account. See the Microsoft documentation on Authorization and access control. Review the section, "Using ACLs for operations on file systems" for information about granting the service principal permission to access the account. The service principal should have read, write, and execute permissions.

   You can skip the section on RBAC (role-based access control) because RBAC is used for management and you only need data access.

4. Open Cloudera Manager and go to **Administration** > **External Accounts**.
5. Select the **Azure Credentials** tab.
6. Click **Add AD Service Principal**.
7. In the **Name** field, enter a unique name to identify the credentials in your cluster.
8. Enter the **Client ID**, **Client Secret Key**, and **Tenant ID** that you obtained when creating the ADLS account and service principal.
9. Click **Save**.

   The **Connect to Azure Data Lake Storage** dialog box displays.

10. Click **Enable for *Cluster_Name*** to add the ADLS Connector Service, as described in the next section.

Adding the ADLS Connector Service

Minimum Required Role: Cluster Administrator (also provided by **Full Administrator**)

Use this procedure to add the ADLS Connector Service using Cloudera Manager. If you have not already configured ADLS Credentials in Cloudera Manager, see Configuring ADLS Credentials in Cloudera Manager on page 601 before continuing.

> **Important:** A cluster cannot use the ADLS Connector Service and the S3 Connector Service at the same time. You must remove any current connector services before adding a new one. See Removing the ADLS Connector Service on page 602 or Removing the S3 Connector Service on page 591.

1. In the Cloudera Manager Admin console, go to the cluster where you want to add the ADLS Connector Service.
2. Click **Actions** > **Add Service**.
3. Select **ADLS Connector**.
4. Click **Continue**.

   The **Add ADLS Connector Service to *Cluster Name*** wizard displays.

5. Select the ADLS credential to use with this service from the **Name** drop-down list.

6. Click **Continue**.

   The wizard checks your configuration for compatibility with ADLS and reports any issues. The wizard does not allow you to continue if you have an invalid configuration. Fix any issues, and then repeat these steps to add the ADLS Connector Service.

7. Select a **Credentials Protection Policy**. Choose one of the following:

   • **Less Secure**

   Credentials can be stored in plain text in some configuration files for specific services (currently Impala, Hive, and Hue) in the cluster.

   This configuration is appropriate for single-user clusters or clusters where strict fine-grained access control is not required.

   • **More Secure**

   The **More Secure** option requires that you enable Kerberos and the Apache Sentry Service in the cluster.

   Cloudera Manager distributes secrets to a limited set of services (currently Impala and Hue) and enables those services to access ADLS securely, using encrypted credentials. It does not distribute these credentials to any other clients or services.

   Other ADLS configurations settings that are not sensitive are included in the configuration of all services and clients as needed.

   This configuration is appropriate for secure, multi-tenant clusters that provide fine-grained access control to data stored in ADLS. You can use the Apache Sentry Service to limit access to specific users and applications.

8. Click **Continue**.
9. If you have enabled the Hue service, the **Additional Configuration for Hue** screen displays. Enter the domain name of the **Hue Browser Data Lake Store** in the form: `store_name.azuredatalakestore.net`
10. Click **Continue**.

    The **Restart Dependent Services** page displays and indicates the dependent services that need to be restarted.

11. Select **Restart Now** to restart these services. You can also restart these services later.
12. Click **Continue** to complete the addition of the ADLS Connector Service. If **Restart Now** is selected, the dependent services are restarted. The progress of the restart commands displays.
13. When the commands finish executing, click **Continue**.
14. Click **Finish**.

Managing ADLS Credentials in Cloudera Manager

To edit or remove ADLS credentials:

1. Open Cloudera Manager and go to **Administration** > **External Accounts**.
2. Select the **Azure Credentials** tab.
3. To remove a credential, in the row for the credential you want to change, click **Actions** > **Remove**.

   You cannot remove a credential that is currently being used by the ADLS Connector Service; you must first remove the Connector Service from the cluster.

4. To edit a credential, in the row for the credential you want to edit, click **Actions** > **Edit Credential**.
5. Edit the fields of the credential as needed and click **Save**.

Removing the ADLS Connector Service

To remove the service:

1. Open Cloudera Manager and go to **Administration** > **External Accounts**.
2. Select the **Azure Credentials** tab.

3. In the row for the credential used for the service, click **Actions** > **Edit Connectivity**.

   The **Connect to Azure Data Lake Storage** dialog box displays.

4. Click **Disable for *Cluster_name***.
5. Click **OK**.

   A message displays saying "The configuration has been updated". You will need to restart any stale services. Click the **View Stale Configurations** link to open the **Stale Configurations page.** Click **Restart Stale Services**.

You can also delete the ADLS Connector Service from the Cloudera Manager home page for the cluster. See Deleting Services on page 201.

Configuring ADLS Connectivity

Microsoft Azure Data Lake Store (ADLS) is a massively scalable distributed file system that can be accessed through an HDFS-compatible API. ADLS acts as a persistent storage layer for CDH clusters running on Azure. In contrast to Amazon S3, ADLS more closely resembles native HDFS behavior, providing consistency, file directory structure, and POSIX-compliant ACLs. See the ADLS documentation for conceptual details.

CDH supports using ADLS as a storage layer for MapReduce2 (MRv2 or YARN), Hive, Hive on Spark, Spark 2.1 and higher, and Spark 1.6. Other applications are not supported and may not work, even if they use MapReduce or Spark as their execution engine. Use the steps in this topic to set up a data store to use with these CDH components.

Note the following limitations:

* ADLS is not supported as the default filesystem. Do not set the default file system property (`fs.defaultFS`) to an `adl://` URI. You can still use ADLS as secondary filesystem while HDFS remains the primary filesystem.
* Hadoop Kerberos authentication is supported, but it is separate from the Azure user used for ADLS authentication.

Setting up ADLS to Use with CDH

1. To create your ADLS account, see the Microsoft documentation.
2. Create the service principal in the Azure portal. See the Microsoft documentation on creating a service principal.

   > **Important:**
   >
   > While you are creating the service principal, write down the following values, which you will need in step 4:
   >
   > * The client id.
   > * The client secret.
   > * The refresh URL. To get this value, in the Azure portal, go to **Azure Active Directory > App registrations > Endpoints**. In the Endpoints region, copy the **OAUTH 2.0 TOKEN ENDPOINT**. This is the value you need for the *refresh_URL* in step 4.

3. Grant the service principal permission to access the ADLS account. See the Microsoft documentation on Authorization and access control. Review the section, "Using ACLs for operations on file systems" for information about granting the service principal permission to access the account.

   You can skip the section on RBAC (role-based access control) because RBAC is used for management and you only need data access.

4. Configure your CDH cluster to access your ADLS account. To access ADLS storage from a CDH cluster, you provide values for the following properties when submitting jobs:

Table 36: ADLS Access Properties

Property Description	Property Name
Provider Type	`dfs.adls.oauth2.access.token.provider.type` The value of this property should be `ClientCredential`
Client ID	`dfs.adls.oauth2.client.id`
Client Secret	`dfs.adls.oauth2.credential`
Refresh URL	`dfs.adls.oauth2.refresh.url`

There are several methods you can use to provide these properties to your jobs. There are security and other considerations for each method. Select one of the following methods to access data in ADLS:

- User-Supplied Key for Each Job
- Single Master Key for Cluster-Wide Access
- User-Supplied Key stored in a Hadoop Credential Provider
- Create a Hadoop Credential Provider and reference it in a customized copy of the core-site.xml file for the service

Testing and Using ADLS Access

1. After configuring access, test your configuration by running the following command that lists files in your ADLS account:

```
hadoop fs –ls adl://your_account.azuredatalakestore.net/
```

If your configuration is correct, this command lists the files in your account.

2. After successfully testing your configuration, you can access the ADLS account from MRv2, Hive, Hive on Spark , Spark 1.6, Spark 2.1 and higher, or HBase by using the following URI:

```
adl://your_account.azuredatalakestore.net
```

For additional information and examples of using ADLS access with Hadoop components:

- **Spark:** See Accessing Data Stored in Azure Data Lake Store (ADLS) through Spark
- **distcp:** See Using DistCp with Microsoft Azure (ADLS)
- **TeraGen:**

```
export HADOOP_CONF_DIR=path_to_working_directory
export HADOOP_CREDSTORE_PASSWORD=hadoop_credstore_password
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
 teragen 1000 adl://jzhugeadls.azuredatalakestore.net/tg
```

User-Supplied Key for Each Job

You can pass the ADLS properties on the command line when submitting jobs.

- **Advantages:** No additional configuration is required.
- **Disadvantages:** Credentials will appear in log files, command history and other artifacts, which can be a serious security issue in some deployments.

> **Important:** Cloudera recommends that you only use this method for access to ADLS in development environments or other environments where security is not a concern.

Use the following syntax to run your jobs:

```
hadoop command
    -Ddfs.adls.oauth2.access.token.provider.type=ClientCredential \
    -Ddfs.adls.oauth2.client.id=CLIENT ID \
    -Ddfs.adls.oauth2.credential='CLIENT SECRET' \
    -Ddfs.adls.oauth2.refresh.url=REFRESH URL \


     adl://<store>.azuredatalakestore.net/src hdfs://nn/tgt
```

Single Master Key for Cluster-Wide Access

Use Cloudera Manager to save the values in the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml**.

- **Advantages:** All users can access the ADLS storage
- **Disadvantages:** This is a highly insecure means of providing access to ADLS for the following reasons:
    - The credentials will appear in all Cloudera Manager-managed configuration files for all services in the cluster.
    - The credentials will appear in the Job History server.

> **Important:** Cloudera recommends that you only use this method for access to ADLS in development environments or other environments where security is not a concern.

1. Open the Cloudera Manager Admin Console and go to *Cluster Name* > **Configuration** > **Advanced Configuration Snippets**.
2. Enter the following in the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml**:

```
<property>
  <name>dfs.adls.oauth2.access.token.provider.type</name>
  <value>ClientCredential</value>
</property>
<property>
  <name>dfs.adls.oauth2.client.id</name>
  <value>CLIENT ID</value>
</property>
<property>
  <name>dfs.adls.oauth2.credential</name>
  <value>CLIENT SECRET</value>
</property>
<property>
  <name>dfs.adls.oauth2.refresh.url</name>
  <value>REFRESH URL</value>
</property>
```

3. Click **Save Changes**.
4. Click **Restart Stale Services** so the cluster can read the new configuration information.

User-Supplied Key stored in a Hadoop Credential Provider

- **Advantages:** Credentials are securely stored in the credential provider.
- **Disadvantages:** Works with MapReduce2 and Spark only (Hive, Impala, and HBase are not supported).

1. Create a [Credential Provider.](#)

    a. Create a password for the Hadoop Credential Provider and export it to the environment:

```
export HADOOP_CREDSTORE_PASSWORD=password
```

b. Provision the credentials by running the following commands:

```
hadoop credential create dfs.adls.oauth2.client.id -provider
jceks://hdfs/user/USER_NAME/adls-cred.jceks -value client ID
hadoop credential create dfs.adls.oauth2.credential -provider
jceks://hdfs/user/USER_NAME/adls-cred.jceks -value client secret
hadoop credential create dfs.adls.oauth2.refresh.url -provider
jceks://hdfs/user/USER_NAME/adls-cred.jceks -value refresh URL
```

You can omit the `-value` option and its value and the command will prompt the user to enter the value.

For more details on the `hadoop credential` command, see Credential Management (Apache Software Foundation).

2. Reference the Credential Provider on the command line when submitting jobs:

```
hadoop command
      -Ddfs.adls.oauth2.access.token.provider.type=ClientCredential \

-Dhadoop.security.credential.provider.path=jceks://hdfs/user/USER_NAME/adls-cred.jceks
  \
      adl://<store>.azuredatalakestore.net/
```

Create a Hadoop Credential Provider and reference it in a customized copy of the `core-site.xml` file for the service

- **Advantages:** all users can access the ADLS storage
- **Disadvantages:** you must pass the path to the credential store on the command line.

1. Create a Credential Provider:

a. Create a password for the Hadoop Credential Provider and export it to the environment:

```
export HADOOP_CREDSTORE_PASSWORD=password
```

b. Provision the credentials by running the following commands:

```
hadoop credential create dfs.adls.oauth2.client.id -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value client ID
hadoop credential create dfs.adls.oauth2.credential -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value client secret
hadoop credential create dfs.adls.oauth2.refresh.url -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value refresh URL
```

You can omit the `-value` option and its value and the command will prompt the user to enter the value.

For more details on the `hadoop credential` command, see Credential Management (Apache Software Foundation).

2. Copy the contents of the `/etc/service/conf` directory to a working directory. The *service* can be one of the following verify list:

- `yarn`
- `spark`
- `spark2`

Use the `--dereference` option when copying the file so that symlinks are correctly resolved. For example:

```
cp -r --dereference /etc/spark/conf ~/my_custom_config_directory
```

Change the ownership so that you can edit the files:

```
sudo chown --recursive $USER ~/custom-conf-file/*
```

3. Add the following to the `core-site.xml` file in the working directory:

```
<property>
  <name>hadoop.security.credential.provider.path</name>
  <value>jceks://hdfs/path_to_credential_store_file</value>
</property>
<property>
  <name>dfs.adls.oauth2.access.token.provider.type</name>
  <value>ClientCredential</value>
</property>
```

The value of the *path_to_credential_store_file* should be the same as the value for the `--provider` option in the `hadoop credential create` command described in step 1.

4. Set the `HADOOP_CONF_DIR` environment variable to the location of the working directory:

```
export HADOOP_CONF_DIR=path_to_working_directory
```

Creating a Credential Provider for ADLS

You can use a Hadoop Credential Provider to specify ADLS credentials, which allows you to run jobs without having to enter the access key and secret key on the command line. This prevents these credentials from being exposed in console output, log files, configuration files, and other artifacts. Running the command in this way requires that you provision a credential store to securely store the access key and secret key. The credential store file is saved in HDFS.

To create a credential provider, run the following commands:

1. Create a password for the Hadoop Credential Provider and export it to the environment:

```
export HADOOP_CREDSTORE_PASSWORD=password
```

2. Provision the credentials by running the following commands:

```
hadoop credential create dfs.adls.oauth2.client.id -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value client ID
hadoop credential create dfs.adls.oauth2.credential -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value client secret
hadoop credential create dfs.adls.oauth2.refresh.url -provider
jceks://hdfs/user/USER_NAME/adlskeyfile.jceks -value refresh URL
```

You can omit the `-value` option and its value and the command will prompt the user to enter the value.

For more details on the `hadoop credential` command, see [Credential Management (Apache Software Foundation)](#).

ADLS Configuration Notes

ADLS Trash Folder Behavior

If the `fs.trash.interval` property is set to a value other than zero on your cluster and you do not specify the `-skipTrash` flag with your `rm` command when you remove files, the deleted files are moved to the trash folder in your ADLS account. The trash folder in your ADLS account is located at `adl://your_account.azuredatalakestore.net/user/user_name/.Trash/current/`. For more information about HDFS trash, see [Configuring HDFS Trash](#) on page 103.

User and Group Names Displayed as GUIDs

By default ADLS user and group names are displayed as GUIDs. For example, you receive the following output for these Hadoop commands:

```
$hadoop fs -put /etc/hosts adl://your_account.azuredatalakestore.net/one_file
$hadoop fs -ls adl://your_account.azuredatalakestore.net/one_file
-rw-r--r--  1 94c1b91f-56e8-4527-b107-b52b6352320e cdd5b9e6-b49e-4956-be4b-7bd3ca314b18
   273
2017-04-11 16:38 adl://your_account.azuredatalakestore.net/one_file
```

To display user-friendly names, set the property `adl.feature.ownerandgroup.enableupn` to `true` in the `core-site.xml` file or at the command line. When this property is set to `true` the `-ls` command returns the following output:

```
$hadoop fs -ls adl://your_account.azuredatalakestore.net/one_file
-rw-r--r--  1 YourADLSApp your_login_app    273 2017-04-11 16:38
adl://your_account.azuredatalakestore.net/one_file
```

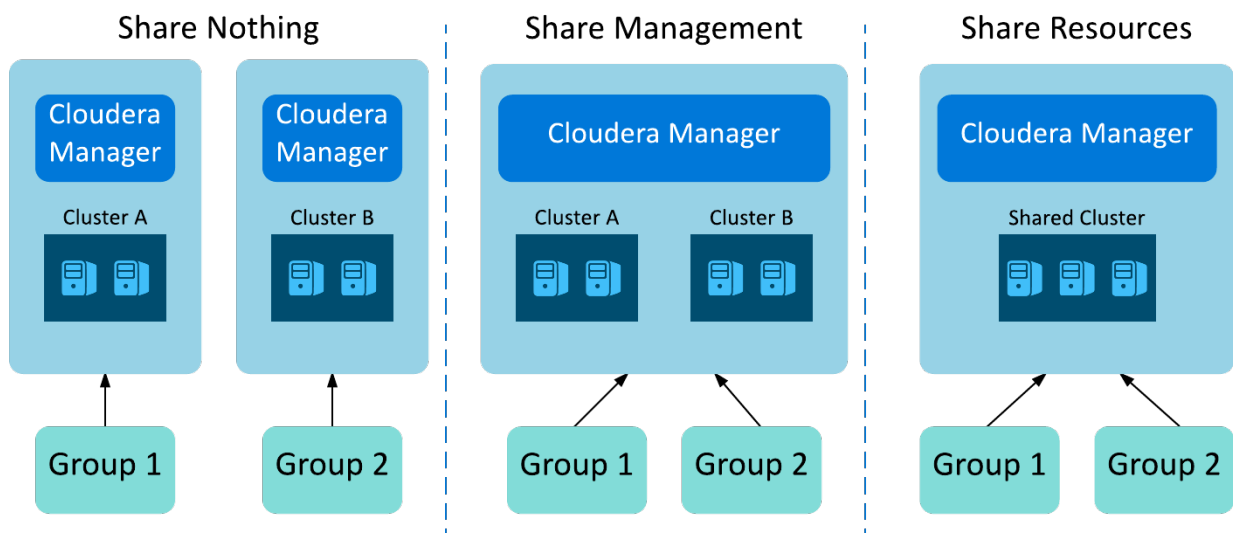# How To Create a Multitenant Enterprise Data Hub

Multitenancy in an enterprise data hub (EDH) lets you share the collective resources of your CDH clusters between user groups without impacting application performance or compromising security.

Advantages of multitenancy include opportunities for data sharing, consolidated operations, improved performance, and better use of resources.

This topic walks through the steps to create a multitenant enterprise data hub:

Choosing an Isolation Model

There are three standard isolation models for an EDH: Share Nothing, Share Management, Share Resources.

Share Nothing

In a *share nothing* architecture, both management and data are completely separate. Nothing is shared between clusters. This architecture does not provide the benefits of multitenancy, but IT teams might find it appropriate based on specific operational realities and governance policies.

For example, specific company contracts or policies might force an enterprise IT team to use this model. Another common example is security and data privacy mandates that restrict the transfer of data sets across geographical boundaries.

Share Management

A *shared management* model offers the benefits of reduced operational overhead without sharing cluster resources or data between groups. This approach is a middle ground, granting some of the benefits of multitenancy while maintaining isolation at the cluster level. This is the preferred choice for environments where full multitenancy is not appropriate. For example, enterprises commonly employ this model for purpose-built, high-priority clusters that cannot risk any performance issues or resource contention, such as an ad serving system or retail personalization, "next offer" engine. While a multitenant EDH always faces the risks of misconfigured applications or malicious users, this model mitigates these risks at the cost of data sharing and resource pooling.

Share Data

The shared resource model uses full multitenancy with all the benefits from consolidated management to shared data and resources. It represents the desired end state for many EDH operators. For example, a biotechnology firm can harness the entire body and insight of research, trial data, and individual perspectives from all its research teams and other departments by employing a full multitenant EDH to store and analyze its information, greatly accelerating innovation through transparency and accessibility.

Balancing Criticality and Commonality

Enterprise IT teams often discover that multitenancy is not necessarily a good fit for their mission-critical workloads running uniquely tailored data sets. Multitenancy can be extremely useful for less critical workloads that employ shared data sets by reducing unnecessary or burdensome data duplication and synchronization. For most situations where business units share data, but the specific workloads are critical to the organization, the overarching business priorities and SLA goals drive the choice of a multitenant or isolated architecture. For some, the risk of latency and resource contention weighs heavily on their performance goals, and would suggest a shared management model. Others consider data visibility paramount, such as for fraud detection and insider threat analysis, which would warrant a shared resource model.

Configuring Security

Once you settle on an isolation model, you can choose the security elements to support your model.

Security for Hadoop is clearly critical in both single tenant and multitenant environments. It establishes the foundation for trusted data and usage among the various actors in the business environment. Without such trust, enterprises cannot rely on the resources and information when making business-critical decisions, which in turn undermines the benefits of operational consolidation and the decreased friction of shared data and insights. Cloudera's EDH provides a rich set of tools and frameworks for security. Key elements of this strategy and its systems include:

- **Authentication**, which proves users are who they say they are.
- **Authorization**, which determines what users are allowed to see and do.
- **Auditing**, which determines who did what, and when.
- **Data Protection**, which encrypts data-at-rest and in-motion.

Cloudera's EDH offers additional tools, such as network connectivity management and data masking. For further information on how IT teams can enable enterprise-grade security measures and policies for multitenant clusters, see Securing Your Enterprise Hadoop Ecosystem. In the context of multitenant administration, security requirements should also include:

- Delegating Security Management
- Managing Auditor Access
- Managing Data Visibility

Delegating Security Management

A central IT team tends to become the bottleneck in granting permissions to individuals and teams to specific data sets when handling large numbers of data sources with different access policies. Organizations can use Apache Sentry, the open source role-based access control (RBAC) system for Hadoop, to delegate permissions management for given data sets. Using this approach, local data administrators are responsible for assigning access for those data sets to the appropriate individuals and teams.

For more information, see Authorization With Apache Sentry.

Managing Auditor Access

For most large multitenant clusters, audit teams typically need access to data audit trails. For example, an audit team might need to monitor usage patterns for irregular behavior such as spikes in request access to credit card details or other sensitive information, without having full access to the cluster and its resources and data. Enterprise IT teams

often adhere to the best practice of "least privilege" and restrict operational access to the minimum data and activity set required. For these cases, Cloudera Navigator provides a data auditor role that partitions the management rights to the cluster so that administrators can grant the audit team access only to the informational data needed, mitigating the impact to operations and security. This approach also answers the common request of audit teams to simplify and focus the application user interface.

For more information, see Cloudera Navigator Auditing.

Managing Data Visibility

Data visibility, in particular for the cluster administrator, is another security requirement that is prominent in most multitenant environments, especially those under strict compliance policies or regulations. Typical security approaches encrypt data, both on-disk and in-use, so that only users with the correct access can view data. Even administrators without proper access cannot view data stored on Hadoop. Cloudera Navigator provides data encryption and enterprise-grade key management with encrypt and key trustee, out-of-the-box.

For more information, see Cloudera Navigator Encryption.

Managing Resource Isolation

IT administrators must manage another crucial aspect of running a multitenant environment: facilitating the fair and equitable usage of finite cluster resources across different users and workloads. Typically, this is a result of aggregating resources to drive improved performance and utilization, a key business driver for multitenancy. Multiple groups within the organization finance the operations of this resource pool to meet this goal. As an outcome of many of these financing models, EDH administrators require systems to grant proportional access to this pool based on the proportion of payment. In addition, a successful multitenant environment employs these tools and frameworks to let users meet SLAs for critical workloads, even in the presence of unpredictable usage stemming from multiple, simultaneous workloads and ill-constructed or misconfigured processes.

Managing Resources

The practical batch processing engine of Hadoop, MapReduce, provides a scheduler framework that administrators can configure to ensure multiple, simultaneous user jobs share physical resources. More specifically, many production environments have successful implementations of the Fair Scheduler. These environments provide maximum utilization while enforcing SLAs through assigned resource minimums.

For more information, see Configuring the Fair Scheduler.

Defining Tenants with Dynamic Resource Pools

With the advent of computing capabilities such as Impala and Cloudera Search and a growing ecosystem of partner applications built to take advantage of CDH and the elements of Cloudera's EDH, cluster faculties and data are increasingly shared with systems that do not operate within the original management framework of MapReduce. A resource management solution must take the full range of these systems into account. To address this challenge, Cloudera's EDH and the underlying Hadoop platform, CDH, ship with the YARN resource management framework. In this context, YARN becomes a building block for computing engines to coordinate consumption and usage reservations to ensure resources are fairly allocated and used. This approach is sometimes referred to as *dynamic partitioning*.

Currently, Impala, MapReduce, and other well designed YARN applications participate in dynamic partitioning in CDH. IT administrators should also consider, with respect to the scheduler capability, how best to regulate tenant access to specified allocations (also known as *pools*) of resources. For example, IT teams might want to balance allocations between the processing needs for their marketing team's near real-time campaign dashboards and their finance department's SLA-driven quarterly compliance and reporting jobs. These administrative needs also extend to multiple applications within a single group. For example, the finance team must balance their quarterly reporting efforts with the daily expense report summaries. To achieve these goals, Hadoop and YARN support Access Control Lists for the various resource schedulers, ensuring that a user (or application) or group of users can only access a given resource pool.

For more information, see Dynamic Resource Pools.

Using Static Partitioning

While dynamic partitioning with YARN offers the IT administrator immense flexibility from a resource management perspective, IT teams operate applications that are not built on the YARN framework or require hard boundaries for resource allocation in order to separate them fully from other services in the cluster. Typically, these applications are purpose-built and by design do not permit this degree of resource flexibility.

To satisfy these business cases, Cloudera's EDH, through Cloudera Manager, supports a static partitioning model, which leverages a technology available on modern Linux operating systems called *container groups* (cgroups). In this model, IT administrators specify policies within the host operating system to restrict a particular service or application to a given allocation of cluster resources. For instance, the IT administrator can choose to partition a cluster by limiting an Apache HBase service to a maximum of 50% of the cluster resources and allotting the remaining 50% to a YARN service and its associated dynamic partitioning in order to accommodate the business SLAs and workloads handled by each of these services.

For more information, see Static Resource Pools.

Using Impala Admission Control

Within the constraints of the static service pool, you can further subdivide Impala's resources using Admission Control.

You use Admission Control to divide usage between Dynamic Resource Pools in multitenant use cases. Allocating resources judiciously allows your most important queries to run faster and more reliably.

For more information, see Managing Impala Admission Control.

Managing Quotas

Fair distribution of resources is essential for keeping tenants happy and productive.

While resource management systems ensure appropriate access and minimum resource amounts for running applications, IT administrators must also carefully govern cluster resources in terms of disk usage in a multitenant environment. Like resource management, disk management is a balance of business objectives and requirements across a range of user communities. The underlying storage foundation of a Hadoop-based EDH, the Hadoop Distributed File System (HDFS), supports quota mechanisms that administrators use to manage space usage by cluster tenants.

HDFS Utilization Reporting

Cloudera Manager reports let you keep track disk usage (storage space information) and directory usage (file metadata, such as size, owner, and last access date). You can use these reports to inform your decisions regarding quotas.

For more information, see Disk Usage Reports and Directory Usage Reports.

Managing Storage Quotas

Administrators can set disk space limits on a per-directory basis. This quota prevents users from accidentally or maliciously consuming too much disk space within the cluster, which can impact the operations of HDFS, similar to other file systems.

For more information, see Setting HDFS Quotas.

Managing Name Quotas

Name quotas are similar to disk quotas. Administrators use them to limit the number of files or subdirectories within a particular directory. This quota helps IT administrators optimize the metadata subsystem (NameNode) within the Hadoop cluster.

Name quotas and disk space quotas are the primary tools for administrators to make sure that tenants have access only to an appropriate portion of disk and name resources, much like the resource allocations mentioned in the previous section, and cannot adversely affect the operations of other tenants through their misuse of the shared file system.

For more information, see Setting HDFS Quotas.

Monitoring and Alerting

Resource and quota management controls are critical to smooth cluster operations. Even with these tools and systems, administrators have to plan for unforeseen situations such as an errant job or process that overwhelms an allotted resource partition for a single group and requires investigation and possible response.

Cloudera Manager provides Hadoop administrators a rich set of reporting and alerting tools that can be used to identify dangerous situations like low disk space conditions; once identified, Cloudera Manager can generate and send alerts to a network operations center (NOC) dashboard or an on-call resource via pager for immediate response.

For more information, see Introduction to Cloudera Manager Monitoring.

Implementing Showback and Chargeback

A common requirement for multitenant environments is the ability to meter the cluster usage of different tenants. As mentioned, one of the key business drivers of multitenancy is the aggregation of resources to improve utilization and performance. The multiple participants build internal budgets to finance this resource pool. In many organizations, IT departments use the metered information to drive showback or chargeback models and illustrate compliance.

Cluster Utilization Reporting

Cluster Utilization Report screens in Cloudera Manager display aggregated utilization information for YARN and Impala jobs. The reports display CPU utilization, memory utilization, resource allocations made due to the YARN fair scheduler, and Impala queries. The report displays aggregated utilization for the entire cluster and also breaks out utilization by tenant. You can configure the report to display utilization for a range of dates, specific days of the week, and time ranges.

Cluster utilization reporting lets you answer key questions such as:

- "How much CPU and memory did each tenant use?"
- "I set up fair scheduler. Did each of my tenants get their fair share?"
- "Which tenants had to wait the longest for their applications to get resources?"
- "Which tenants asked for the most memory but used the least?"
- "When do I need to add nodes to my cluster?"

For more information, see Cluster Utilization Reports.

Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

Appendix: Apache License, Version 2.0

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```