Cloudera Enterprise Reference Architecture for AWS deployments
Table of Contents

ABSTRACT .......................................................................................................................................................... 1
CLOUDERA ON AWS ........................................................................................................................................... 1
AWS OVERVIEW ................................................................................................................................................. 2
   Elastic Compute Cloud (EC2) ................................................................................................................................. 2
   Simple Storage Service (S3) ................................................................................................................................... 2
   Relational Database Service (RDS) ........................................................................................................................ 2
   Elastic Block Store (EBS) ........................................................................................................................................ 3
   Direct Connect ....................................................................................................................................................... 3
   Virtual Private Cloud ........................................................................................................................................... 3
DEPLOYMENT ARCHITECTURE ............................................................................................................................ 3
   DEPLOYMENT TOPOLOGIES .............................................................................................................................................. 3
      Public Subnet Deployments ................................................................................................................................... 3
      Private Subnet Deployments ................................................................................................................................ 4
WORKLOADS, ROLES, AND INSTANCE TYPES ........................................................................................................................ 6
   Management Nodes ............................................................................................................................................. 6
   Worker Nodes ....................................................................................................................................................... 7
   Edge Nodes ........................................................................................................................................................... 7
   Current EC2 Instances ........................................................................................................................................... 7
   Previous Generation EC2 Instances ....................................................................................................................... 8
REGIONS AND AVAILABILITY ZONES .................................................................................................................................... 8
NETWORKING, CONNECTIVITY, AND SECURITY ..................................................................................................................... 9
   VPC ........................................................................................................................................................................ 9
   Connectivity to the Internet and Other AWS Services ........................................................................................... 9
   Private Data Center Connectivity .......................................................................................................................... 9
   Security Groups ..................................................................................................................................................... 9
   Placement Groups ............................................................................................................................................... 11
SUPPORTED AMIS .................................................................................................................................................. 11
STORAGE OPTIONS AND CONFIGURATION ......................................................................................................................... 11
   Instance Storage .................................................................................................................................................. 11
   Simple Storage Service ........................................................................................................................................ 11
   Root Device ......................................................................................................................................................... 11
Abstract
An organization’s requirements for a big-data solution are simple: Acquire and combine any amount or type of data in its original fidelity, in one place, for as long as necessary, and deliver insights to all kinds of users, as quickly as possible.

Cloudera, an enterprise data management company, introduced the concept of the enterprise data hub (EDH): a central system to store and work with all data. The EDH has the flexibility to run a variety of enterprise workloads (for example, batch processing, interactive SQL, enterprise search, and advanced analytics) while meeting enterprise requirements such as integrations to existing systems, robust security, governance, data protection, and management. The EDH is the emerging center of enterprise data management. EDH builds on Cloudera Enterprise, which consists of the open source Cloudera Distribution including Apache Hadoop (CDH), a suite of management software and enterprise-class support.

In addition to needing an enterprise data hub, enterprises are looking to move or add this powerful data management infrastructure to the cloud for operation efficiency, cost reduction, compute and capacity flexibility, and speed and agility.

As organizations embrace Hadoop-powered big data deployments in cloud environments, they also want enterprise-grade security, management tools, and technical support—all of which are part of Cloudera Enterprise.

Customers of Cloudera and Amazon Web Services (AWS) can now run the EDH in the AWS public cloud, leveraging the power of the Cloudera Enterprise platform and the flexibility of the AWS cloud.

Cloudera on AWS
Cloudera makes it possible for organizations to deploy the Cloudera solution as an EDH in the AWS cloud. This joint solution combines Cloudera’s expertise in large-scale data management and analytics with AWS’s expertise in cloud computing.

This joint solution provides the following benefits:

Flexible Deployment, Faster Time to Insight
Running Cloudera Enterprise on AWS provides the greatest flexibility in deploying Hadoop. Customers can now bypass prolonged infrastructure selection and procurement processes to rapidly implement the Cloudera big data platform and realize tangible business value from their data immediately. Hadoop excels at large-scale data management, and the AWS cloud provides infrastructure services on demand.

Scalable Data Management
At large organizations, it can take weeks or even months to add new nodes to a traditional data cluster. By deploying Cloudera Enterprise in AWS, enterprises can effectively shorten rest-to-growth cycles to scale their data hubs as their business grows.

**On-demand Processing Power**
While Hadoop focuses on collocating compute to disk, many processes benefit from increased compute power. Deploying Hadoop on Amazon allows a fast ramp-up and ramp-down based on specific workloads—flexibility that is difficult to obtain with on-premise deployment.

**Improved Efficiency and Increased Cost Savings**
Deploying in AWS eliminates the need for dedicated resources to maintain a traditional data center, enabling organizations to focus instead on core competencies. As annual data growth for the average enterprise continues to skyrocket, even relatively new data management systems can strain under the demands of modern high-performance workloads. By moving their data-management platform to the cloud, enterprises can avoid costly annual investments in on-premises data infrastructure to support new enterprise data growth, applications, and workloads.

In this white paper, we provide an overview of best practices for running Cloudera on AWS and leveraging different AWS services such as EC2, S3, and RDS.

**AWS Overview**
AWS is the leading public cloud infrastructure provider. Their offerings consist of several different services, ranging from storage to compute, to higher up the stack for automated scaling, messaging, queuing, and other services. Cloudera Enterprise deployments can use the following service offerings.

**Elastic Compute Cloud (EC2)**
With Elastic Compute Cloud (EC2), users can rent virtual machines of different configurations, on demand, for the time required. For this deployment, EC2 instances are the equivalent of servers that run Hadoop. EC2 offers several different types of instances with different pricing options. For Cloudera Enterprise deployments, each individual node in the cluster conceptually maps to an individual server. A list of supported instance types and the roles that they play in a Cloudera Enterprise deployment are described later in this document.

**Simple Storage Service (S3)**
Simple Storage Service (S3) allows users to store and retrieve various sized data objects using simple API calls. S3 is designed for 99.999999999% durability and 99.99% availability. S3 provides only storage; there is no compute element. The compute service is provided by EC2, which is independent of S3.

**Relational Database Service (RDS)**
Relational Database Service (RDS) allows users to provision different types of managed relational database instances, including Oracle and MySQL. RDS handles database management tasks, such as backups for a user-defined retention period, point-in-time recovery,
patch management, and replication, allowing users to pursue higher value application development or database refinements.

**Elastic Block Store (EBS)**

Elastic Block Store (EBS) provides block-level storage volumes that can be used as network attached disks with EC2 instances. Users can provision volumes of different capacities and IOPS guarantees. Unlike S3, these volumes can be mounted as network attached storage to EC2 instances and have an independent persistence lifecycle; that is, they can be made to persist even after the EC2 instance has been shut down. At a later point, the same EBS volume can be attached to a different EC2 instance. EBS volumes can also be snapshotted to S3 for higher durability guarantees. EBS is optimized primarily for random access patterns.

**Direct Connect**

Use **Direct Connect** to establish direct connectivity between your data center and AWS region. You can configure direct connect links with different bandwidths based on your requirement. With this service, you can consider AWS infrastructure as an extension to your data center.

**Virtual Private Cloud**

With **Virtual Private Cloud (VPC)**, you can logically isolate a section of the AWS cloud and provision services inside of that isolated network. Using VPC is recommended to provision services inside AWS and is enabled by default for all new accounts. VPC has various configuration options for accessibility to the Internet and other AWS services. You can create public-facing subnets in VPC, where the instances can have direct access to the public Internet gateway and other AWS services. Instances can be provisioned in private subnets too, where their access to the Internet and other AWS services can be restricted or managed through network address translation (NAT). RDS instances can be accessed from within a VPC.

**Deployment Architecture**

**Deployment Topologies**

Two kinds of Cloudera Enterprise deployments are supported in AWS, both within VPC but with different accessibility:

1. Cluster inside a public subnet in VPC
2. Cluster inside a private subnet in VPC

Choosing between the public subnet and private subnet deployments depends predominantly on the accessibility of the cluster, both inbound and outbound, and the bandwidth required for outbound access.

**Public Subnet Deployments**

A public subnet in this context is a subnet with a route to the Internet gateway. Instances provisioned in public subnets inside VPC can have direct access to the Internet as well as to other AWS services such as RDS and S3. If your cluster must access S3 for data transfers, or ingest from sources on the Internet, your cluster should be deployed in a public subnet. This gives each instance full bandwidth access to the Internet and other AWS services. Unless it’s a requirement, we don’t recommend opening full access to your cluster from the Internet. Using security groups (discussed later), you can configure your cluster to have access to other AWS
services but not to the Internet.

**Private Subnet Deployments**

Instances provisioned in private subnets inside VPC don’t have direct access to the Internet or to other AWS services. To access the Internet, they must go through a NAT instance in the public subnet. If your cluster does not require full bandwidth access to the Internet or to other AWS services, you should deploy in a private subnet.

In both cases, you can set up VPN or Direct Connect between your corporate network and AWS. This makes AWS look like an extension to your network, and the Cloudera Enterprise deployment is accessible as if it were on servers in your own data center.

Deployment in the public subnet looks like this:

Deployment in the private subnet looks like this:

The accessibility of your Cloudera Enterprise cluster is defined by the VPC configuration and depends on the security requirements and the workload. Typically, there are edge/client nodes
that have direct access to the cluster. Users go through these edge nodes via client applications to interact with the cluster and the data residing there. These edge nodes could be running a web application for real-time serving workloads, BI tools, or simply the Hadoop command-line client used to submit or interact with HDFS. The public subnet deployment with edge nodes looks like this:

The edge nodes in a private subnet deployment could be in the public subnet, depending on how they must be accessed. The figure above shows them in the private subnet as one deployment option.
The edge nodes can be EC2 instances in your VPC or servers in your own data center. Cloudera recommends allowing access to the Cloudera Enterprise cluster via edge nodes only. You can configure this in the security groups for the instances that you provision. The rest of this document describes the various options in detail.

workloads, Roles, and Instance Types
In this reference architecture, we consider different kinds of workloads that are run on top of an enterprise data hub and make recommendations on the EC2 instances that are suitable for each of these workload types. The recommendations consider new- and old-generation instance types, with storage options including magnetic disks and SSDs. You choose instance types based on the workload you run on the cluster. You should also do a cost-performance analysis.

Cloudera currently supports RHEL 6.4 AMIs on CDH 4.5 and higher and CDH 5.

The following shows a matrix of workload categories and services that are typically combined for the workload type:

<table>
<thead>
<tr>
<th>Workload Type</th>
<th>Typical Services</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>● MapReduce</td>
<td>Suitable for workloads that are predominantly batch oriented in nature and involved MapReduce or Spark.</td>
</tr>
<tr>
<td></td>
<td>● YARN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Spark</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Hive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Pig</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Crunch</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>● HBase</td>
<td>Suitable for higher resource-consuming services and production workloads but limited to only one of these running at any time.</td>
</tr>
<tr>
<td></td>
<td>● Solr</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Impala</td>
<td></td>
</tr>
<tr>
<td>High/Full EDH</td>
<td>● All CDH services</td>
<td>Full-scale production workloads with multiple services running in parallel on a multi-tenant cluster.</td>
</tr>
</tbody>
</table>

Management Nodes
Management nodes for a Cloudera Enterprise deployment run the management services, which include:

- Cloudera Manager
- JobTracker
- Standby JobTracker
- NameNode
- Standby NameNode
- JournalNodes
- HBase Master
- ZooKeeper
- Oozie

Worker Nodes
Worker nodes for a Cloudera Enterprise deployment run worker services, which include:
- DataNode
- TaskTracker
- HBase RegionServer
- Impala Daemons
- Solr Servers

Edge Nodes
Hadoop client services run on edge nodes. They are also known as gateway services and include:
- Third-party tools
- Hadoop command-line client
- Hive command-line client
- Impala command-line client
- Flume agents
- Hue Server

The following matrix show the different workload categories, instance types, and roles they are suited for in a cluster.

Current EC2 Instances

<table>
<thead>
<tr>
<th>Workload Type</th>
<th>Typical Services</th>
<th>Instances for Management Nodes</th>
<th>Instances for Worker Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>- MapReduce</td>
<td>c3.8xlarge</td>
<td>c3.8xlarge</td>
</tr>
<tr>
<td></td>
<td>- YARN</td>
<td>d2.2xlarge</td>
<td>d2.8xlarge</td>
</tr>
<tr>
<td></td>
<td>- Spark</td>
<td>i2.2xlarge</td>
<td>i2.2xlarge</td>
</tr>
<tr>
<td></td>
<td>- Hive</td>
<td>i2.4xlarge</td>
<td>i2.4xlarge</td>
</tr>
<tr>
<td></td>
<td>- Pig</td>
<td>i2.8xlarge</td>
<td>i2.8xlarge</td>
</tr>
<tr>
<td></td>
<td>- Crunch</td>
<td>m2.4xlarge</td>
<td>r3.8xlarge</td>
</tr>
<tr>
<td>Medium</td>
<td>- HBase</td>
<td>c3.8xlarge</td>
<td>d2.8xlarge</td>
</tr>
</tbody>
</table>
A detailed list of configurations for the different instance types is available on the EC2 instance types page.

**Previous Generation EC2 Instances**

<table>
<thead>
<tr>
<th>Workload Type</th>
<th>Typical Services</th>
<th>Instances for Management Nodes</th>
<th>Instances for Worker Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>● MapReduce ● YARN ● Spark ● Hive ● Pig ● Crunch</td>
<td>c1.xlarge cc2.8xlarge hi1.4xlarge hs1.8xlarge m1.large m1.xlarge m2.2xlarge</td>
<td>c1.xlarge cc2.8xlarge hi1.4xlarge hs1.8xlarge m1.large m1.xlarge m2.4xlarge</td>
</tr>
<tr>
<td>Medium</td>
<td>● HBase ● Solr ● Impala</td>
<td>cc2.8xlarge hi1.4xlarge hs1.8xlarge m1.xlarge m2.4xlarge</td>
<td>cc2.8xlarge hi1.4xlarge hs1.8xlarge</td>
</tr>
<tr>
<td>High / Full EDH workloads</td>
<td>● All CDH services</td>
<td>cc2.8xlarge hs1.8xlarge</td>
<td>cc2.8xlarge hs1.8xlarge i2.8xlarge</td>
</tr>
</tbody>
</table>

A detailed list of configurations for the different instance types is available on the EC2 Previous Generation Instances page.

**Regions and Availability Zones**

Regions are self-contained geographical locations where AWS services are deployed. Regions have their own deployment of each service. Each service within a region has its own endpoint that you can interact with to use the service.

Regions contain availability zones, which are isolated locations within a general geographical location. Some regions have more availability zones than others. While provisioning, you can choose specific availability zones or let AWS pick for you.
Cloudera EDH deployments are restricted to single availability zones. Clusters spanning availability zones and regions are not supported.

**Networking, Connectivity, and Security**

**VPC**

VPC has several different configuration options. See the VPC documentation for detailed explanation of the options and choose based on your networking requirements. You can deploy Cloudera Enterprise clusters in either public or private subnets. In both cases, the instances forming the cluster should not be assigned a publicly addressable IP unless they must be accessible from the Internet or other AWS services. If you assign public IP addresses to the instances and want to block incoming traffic, you can use security groups.

**Connectivity to the Internet and Other AWS Services**

Deploying the instances in a public subnet allows them to have access to the Internet for outgoing traffic as well as to other AWS services, such as S3 and RDS. Clusters that require data transfer between other AWS services (especially S3) and HDFS should be deployed in a public subnet with public IP addresses assigned so that they can directly transfer data to those services. Configure the security group for the cluster nodes to block incoming connections to the cluster instances.

Clusters that don’t need heavy data transfer between other AWS services or the Internet and HDFS should be launched in the private subnet. These clusters still might need access to services like RDS or software repositories for updates. Do this by provisioning a NAT instance in the public subnet, allowing access outside the private subnet into the public domain. Cloudera does not recommend using the NAT instance for large-scale data movement.

If you completely disconnect the cluster from the Internet, you block access for software updates as well as to other AWS services, which makes maintenance difficult. If you are required to completely lock down any external access because you don’t want to keep the NAT instance running all the time, Cloudera recommends starting a NAT instance when external access is required and stopping it when activities are complete.

**Private Data Center Connectivity**

You can establish connectivity between your data center and the VPC hosting your Cloudera Enterprise cluster by using a VPN or Direct Connect. We recommend using Direct Connect so that there is a dedicated link between the two networks with lower latency, higher bandwidth, security and encryption via IPSec. If you don't need high bandwidth and low latency connectivity between your data center and AWS, connecting to EC2 through the Internet is sufficient and Direct Connect may not be required.

**Security Groups**

Security Groups are analogous to firewalls. You can define rules for EC2 instances and define allowable traffic, IP addresses, and port ranges. Instances can belong to multiple security groups. Cloudera Enterprise deployments require the following security groups:
Cluster
This security group blocks all inbound traffic except that coming from the security group containing the Flume nodes and edge nodes. You can allow outbound traffic for Internet access during installation and upgrade time and disable it thereafter. You can also allow outbound traffic if you intend to access large volumes of Internet-based data sources like S3.

Flume Nodes
This security group is for instances running Flume agents. Outbound traffic to the Cluster security group must be allowed, and inbound traffic from sources from which Flume is receiving data must be allowed.

Edge Nodes
This security group is for instances running client applications. Outbound traffic to the Cluster security group must be allowed, and incoming traffic from IP addresses that interact with client applications as well the cluster itself must be allowed.

Each of these security groups can be implemented in public or private subnets depending on the access requirements highlighted above.

A full deployment looks like the following:

Source systems are where the data is being ingested from using Flume. You’ll have flume sources deployed on those machines.
End users are the end clients that interact with the applications running on the edge nodes that can interact with the Cloudera Enterprise cluster.

**Placement Groups**
As described in the AWS documentation [here](#), Placement Groups are a logical grouping of EC2 instances within an availability zone, where instances are provisioned such that the network between them has higher throughput and lower latency. AWS accomplishes this by provisioning instances as close to each other as possible. This limits the pool of instances available for provisioning but guarantees uniform network performance. Cloudera recommends provisioning the worker nodes of the cluster within a placement group. Master and edge nodes can be outside the placement group unless you need high throughput and low latency between those and the cluster—or example, if you are moving large amounts of data or expect low-latency responses between the edge nodes and the cluster.

**Supported AMIs**
Amazon Machine Images (AMIs) are the virtual machine images that run on EC2 instances.
These consist of the operating system and any other software that the AMI creator bundles into them. Cloudera Enterprise deployments in AWS support Red Hat AMIs. HVM and PV AMIs available. For certain instances like cc2.8xlarge, you use Hardware Assisted Virtualization (HVM) AMIs, whereas for instances like m1.xlarge, you use Paravirtualization (PV) AMIs. For instances like hs1.8xlarge where there’s a choice, Cloudera recommends that you use HVM. You can find a list of the Red Hat AMIs for each region [here](#). The only version currently supported is Red Hat 6.4.

**Storage Options and Configuration**
AWS offers different storage options that vary in performance, durability, and cost.

**Instance Storage**
EC2 instances have storage attached at the instance level, similar to disks on a physical server. The storage is virtualized and referred to as *ephemeral storage* because the lifetime of the storage is the same as the lifetime of your EC2 instance. If you stop or terminate the EC2 instance, the storage is lost. The storage not lost on restarts, however. Different EC2 instances have different amounts of instance storage, as highlighted above. For long-running Cloudera Enterprise clusters, the HDFS data directories should use the instance storage, which all the benefits of shipping compute close to the storage and not reading remotely over the network.

**Simple Storage Service**
We strongly recommend using S3 to keep a copy of the data you have in HDFS for disaster recovery. The durability and availability guarantees make it ideal for a cold backup that you can restore in case the primary HDFS cluster goes down. For a hot backup, you need a second HDFS cluster holding a copy of your data.

**Root Device**
We recommend using EBS volumes as root devices for the EC2 instances. When instantiating the instances, you can define the root device size. The root device size for Cloudera Enterprise...
clusters should be at least 500 GB to allow parcels and logs to be stored. You don’t need to use any instance storage for the root device; it can be 100% utilized for HDFS data directories.

**Capacity Planning**
Using AWS allows you to scale your Cloudera Enterprise cluster up and down easily. If your storage or compute requirements change, you can provision and deprovision instances and meet your requirements quickly, without buying physical servers. However, some advance planning makes operations easier. You must plan for whether your workloads need a high amount of storage capacity or not. The supported EC2 instances have different amounts of memory, storage, and compute, and deciding which instance type and generation make up your initial deployment depends on the storage and workload requirement. The operational cost of your cluster depends on the type and number of instances you choose.

**Low Storage Density**
For use cases with lower storage requirements, using cc2.8xlarge is recommended. It provides a lower amount of storage per instance but has a high amount of compute and memory resources to go with it. The cc.8xlarge instances have 4 x 840 GB raw instance storage.

**High Storage Density**
For use cases with higher storage requirements, using d2.8xlarge is recommended. These provide a high amount of storage per instance, but the compute is lower than cc2.8xlarge instances. The d2.8xlarge instances have 24 x 2 TB instance storage.

**Reserved Instances**
AWS offers the ability to reserve EC2 instances up front and pay a lower per-hour price. This is beneficial for users that are using EC2 instances for the foreseeable future and will keep them on a majority of the time. Reserving instances can drive down the TCO significantly of long-running Cloudera Enterprise clusters. There are different options for reserving instances in terms of the time period of the reservation and the utilization of each instance. See the AWS documentation to plan instance reservation.

**Relational Databases**
Cloudera Enterprise deployments require relational databases for the following components:
- Cloudera Manager databases
- Hive and Impala metastore
- Hue database
- Oozie database
- SqoopServer2 database

For operating relational databases in AWS, you can either provision EC2 instances and install and manage your own database instances, or you can use RDS. The list of supported database types and versions is available [here](#).

With all the considerations highlighted so far, a deployment in AWS would look like (for both private and public subnets):
The next section addresses preparation and installation of the cluster.

Installation and Software Configuration

Preparation

Provisioning Instances
To provision EC2 instances, first define the VPC configurations based on your requirements on aspects like access to Internet, other AWS services, and connectivity to your corporate network. After that, you can use the EC2 command-line API tool to provision instances or the AWS management console. To provision instances, you must create a keypair with which you will later be able to log into the instances. In Red Hat AMIs, you'll be able to use this keypair to log in as ec2-user, which has sudo privileges for administrative tasks. While provisioning instances, make sure to specify the following:

- Red Hat 6.4 AMI
- Root device size of at least 500GB
- All ephemeral storage devices to be attached to the instances
- Tags to indicate the role that the instance will play later. This makes identifying instances easier.

Along with instances, databases must be provisioned (RDS or self managed). If you are provisioning in a public subnet, RDS instances can be accessed directly. If you are deploying in a private subnet, you either need a NAT instance to access RDS instances, or you must set up...
database instances on EC2 inside the private subnet. The database credentials are required during Cloudera Enterprise installation.

**Setting Up Instances**
Once the instances are provisioned, you must perform the following to get them ready for deploying CE:

- Disable iptables
- Disable SElinux
- Format and mount the instance storage
- Resize the root volume if it does not show full capacity

For more information on operating system preparation and configuration, see the Cloudera Manager installation instructions available [here](#).

**Deploying Cloudera Enterprise**
To deploy Cloudera Enterprise, log into the instance that you have elected to host Cloudera Manager and follow installation instructions available [here](#).

**Cloudera Enterprise Configuration Considerations**

**HDFS**

**Durability**
For Cloudera Enterprise deployments in AWS, the supported storage option is the ephemeral storage. HDFS on EBS volumes is not a supported configuration. Data stored on ephemeral storage is lost if instances are stopped, terminated, or go down for some other reason. Data persists on restarts, however. Data durability in HDFS can be guaranteed by keeping replication at 3. Cloudera does not recommend lowering the replication factor.

Secondly, a persistent copy of all data should be maintained in S3 to guard against cases where you can lose all three copies of the data. Do this by either writing to S3 at ingest time or distcp-ing datasets from HDFS afterwards.

**Availability**
HDFS availability can be accomplished by deploying the NameNode with high availability with at least three JournalNodes.

**ZooKeeper**
We recommend running at least three ZooKeeper servers for availability and durability.

**Flume**
For durability in Flume agents, use memory channel or file channel. Flume’s memory channel offers increased performance at the cost of no data durability guarantees. File channels offer a higher level of durability guarantee because the data is persisted on disk in the form of files. Cloudera supports file channels on ephemeral storage as well as EBS. If the EC2 instance goes down, the data on the ephemeral storage is lost. For guaranteed data delivery, use EBS-backed storage for the Flume file channel.
Summary
Cloudera and AWS allow users to deploy and use Cloudera Enterprise on AWS infrastructure, combining the scalability and functionality of the Cloudera Enterprise suite of products with the flexibility and economics of the AWS cloud. This whitepaper provided reference configurations for Cloudera Enterprise deployments in AWS. These configurations leverage different AWS services such as EC2, S3, and RDS.

References

Cloudera Enterprise
Cloudera homepage
Cloudera Enterprise documentation
Cloudera Enterprise support

Amazon Web Services
AWS homepage
EC2 homepage
EC2 instance lifecycle
S3 homepage
RDS homepage
VPC homepage
Direct Connect homepage
EC2 networking and security
Red Hat certified AMIs
AWS developer tools
AWS support