Cloudera Cloud Documentation

Important Notice

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

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 Cloud 1.0
Date: February 22, 2019

Table of Contents

Cloudera Enterprise in the Cloud

Increasingly, customers are evaluating cloud environments to solve their big data challenges. More companies are moving to the cloud as they realize that cloud computing increases efficiency and reduces cost. Customers who use Hadoop in the cloud find that a public cloud infrastructure allows them to quickly provision clusters at a lower cost. They can expand or shrink clusters to only use and pay for the resources they need to meet demand.

When customers make the move to the cloud, they might decide to replicate their applications in the cloud to avoid the expense of re-architecting the applications for the cloud. However, simply running workloads in the cloud the same way that they were run on premises may not take full advantage of native cloud features and can turn out to be an expensive solution.

In most cases, Cloudera recommends architecting applications to efficiently use cloud resources and take advantage of cloud features such as transient and elastic clusters, optimized instance types, and object storage. Workloads that run periodically can take advantage of transient clusters that are automatically terminated when the workload completes. By combining transient clusters with Cloudera's pay-per-use capabilities, companies can run their workloads more efficiently and cost effectively. Object storage in the cloud offers scalability and flexibility in data storage and management.

Cloudera product offerings take advantage of native cloud infrastructure. Cloudera compute engines, including Spark, Hive, and Impala work directly with object storage. Cloudera Altus Director simplifies cluster lifecycle management while exploiting the inherent transient and elastic capabilities of the cloud. Altus Director can deploy CDH and its components in the leading public cloud environments: Amazon Web Service (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

The process of migrating workloads from on premises to the cloud and adapting them to take full advantage of cloud infrastructure tends to involve multiple steps and a significant amount of time. Moreover, some applications may benefit from remaining on premises for an indefinite period of time. As a result, companies often decide to have hybrid operations and run workloads both on premises and in the cloud throughout the transition period. Cloudera products and CDH components are well positioned to support hybrid operations because they provide a consistent experience on premises and in the cloud. Applications that can more efficiently run within the company data center can stay on premises and applications that can take full advantage of the elasticity and low cost storage of the cloud can run in any of the public clouds.

Cloudera has the following products that support workloads on premises and in the cloud:

- **Data Engineering.** Provides CDH components for fast and cost-effective storing and processing of data, including Spark and Navigator.
- **Data Warehouse.** Provides CDH components for discovering, analyzing, and understanding data, including Impala, Hive-on-Spark, and Navigator.
- **Operational Database.** Provides CDH components for building data-driven applications, including HBase, Spark Streaming, and Navigator.
- **Workload Experience Manager (XM).** A hosted application that provides insights to help you gain in-depth understanding of the workloads you send to clusters that are managed by Cloudera Manager on-premises or in the cloud.

Data Engineering

Data engineering and ETL workloads typically do not run continuously, which makes them good candidates for running on transient CDH clusters in the cloud. This type of workload can have variable computing requirements and can benefit from the elasticity of clusters in the cloud.

Cloudera Data Engineering allows you to take advantage of elastic clusters in the cloud to quickly process large volumes of data of all types at a lower cost. Spin up clusters for a short-running data engineering job and terminate the cluster when the job completes.

Typically, data processed in data engineering workloads starts in a raw format in an object storage like Amazon S3. Spark and Hive are the most popular compute engines for data engineering and batch ETL. Spark can process streaming data and analyze the data in real-time. Hive is designed for ad-hoc querying and analysis of large volumes of data. For batch processing, you can use Hive-on-Spark, which supports cloud-native data access and enables you to take full advantage of a shared data layer for ETL and BI analytics workloads in the cloud.

For information about best practices for running data engineering workloads on AWS, see Data Engineering on AWS: Best Practices.

For information about configuring Hive ETL jobs to use Amazon S3, see Configuring Transient Hive ETL Jobs to Use the Amazon S3 Filesystem.

Altus for Data Engineers

Cloudera offers a managed cloud service for data engineering and ETL workloads.

Cloudera Altus is a cloud service platform that enables you to create clusters and run jobs specifically for data science and engineering workloads, including ETL and batch processing jobs. It is designed to provision clusters quickly and to make it easy for you to build and run your data engineering workloads in the cloud.

Altus offers multiple distributed processing engine options, including Hive, Spark, and MapReduce2 (MR2), for different data engineering workloads. The processing engines allow you to manage workloads such as ETL, machine learning, and large scale data processing.

For information about Cloudera Altus, see the Altus documentation.

Data Warehouse

Cloudera Data Warehouse offers high-performance SQL analytics in the cloud. You can use Impala to analyze data stored in shared data platforms such as HDFS and Kudu and cloud-native object storage like Amazon S3.

Depending on the demand for data availability, you can run business analytics jobs on transient or persistent clusters. Batch ETL jobs are more suited to transient clusters that terminate when the batch jobs complete. Data analytics jobs that have high demand run frequently and require persistent clusters. Both types of clusters allow you to grow and shrink resources to support peak and off-peak usage and still provide consistent query performance.

Data warehouse workloads typically have multiple users accessing the same clusters. Multi-tenant clusters have security concerns that must be addressed. Persistent clusters for data analytics jobs require a robust security framework and can use Kerberos and Sentry for user authentication and authorization.

Operational Database

Cloudera Operational Database provides a flexible operational database platform, capable of batch and stream processing. It supports relational SQL and NoSQL storage layers and has the ability to store an unlimited amount of structured and unstructured data.

Operational database jobs use HBase to perform fast searches on very large datasets. They can also use Spark Streaming to feed streaming data into HBase. Typically, operational database jobs run on highly-available long-running clusters backed by local storage with HDFS.

Deploying Cloudera Operational Database in the cloud provides the benefits of low cost and convenience associated with rapid provisioning and decommissioning of clusters. When you move operational database jobs to the cloud, you can quickly provision new clusters without incurring the cost of long-term on-premises infrastructure. With Altus Director, you can easily set up and maintain a secure test and development environment on demand and ease administrative overhead. Snapshots for backup and disaster recovery can be moved to cloud storage such as S3 and provide geographical dispersement and reliability and lower cost.

To manage unexpected and temporary surges in processing volume, such as a high-volume data ingestion or an exceptionally large batch job, you can start a large number of clusters to handle the demand. When the high-volume processing completes, you can terminate the clusters and go back to regular operation.

Cloud deployment also adds flexibility and portability between the on-premises data center and multiple cloud vendors.

Workload XM

Workload XM is a tool that provides insights to help you gain in-depth understanding of the workloads you send to clusters managed by Cloudera Manager on-premises or in the cloud. In addition, it provides information that can be used for troubleshooting failed jobs and optimizing slow jobs that run on those clusters. After a job ends, information about job execution is sent to Workload XM with the Telemetry Publisher, a role in the Cloudera Manager Management Service.

Workload XM uses the information to display metrics about the performance of a job. Additionally, Workload XM compares the current run of a job to previous runs of the same job by creating baselines. You can use the knowledge gained from this information to identify and address abnormal or degraded performance or potential performance improvements.

For more information about setting up and using Workload XM, see the Workload XM documentation.

Download

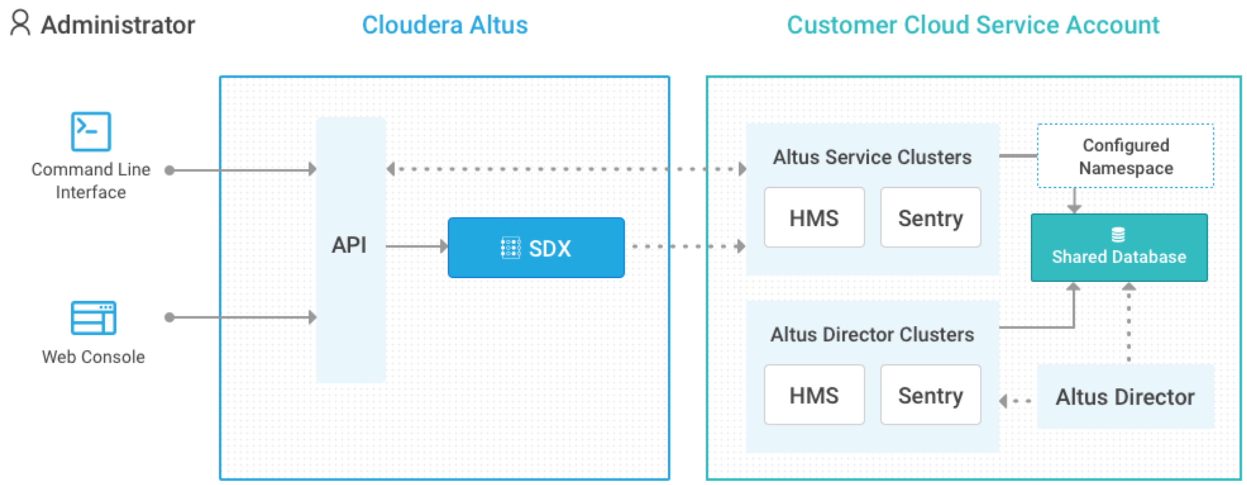
You can download Cloudera Enterprise from the Cloudera downloads page.

Cloudera SDX for Altus: Best Practices and Supported Configuration

Cloudera Shared Data Experience (SDX) for Altus is a centralized framework for data management and governance for CDH clusters in the cloud. While cluster data is stored in cloud storage such as Amazon S3 or Azure Data Lake Store (ADLS), Cloudera SDX externalizes cluster metadata into a shared database available to all workloads and clusters running in the cloud. Workloads in multiple clusters share access to the same consistent data and metadata with no need for replication.

You can set up Cloudera SDX for clusters that you create using Altus Director or Cloudera Manager and share data with Altus services clusters. The Altus SDX service provides configured SDX namespaces that enable you to share Hive metastore (HMS) and Sentry data between Altus Data Warehouse and Altus Data Engineering clusters and clusters that you create with Altus Director or Cloudera Manager.

The following image shows the components in Cloudera SDX and how they interact:



Based on your requirements, you can set up Cloudera SDX to share data between the following types of CDH clusters:

CDH clusters created with Cloudera Manager or Altus Director

You can configure Cloudera SDX for CDH clusters created with Altus Director or Cloudera Manager to share data with other CDH clusters. Set up external databases for Hive metastore (HMS) and Sentry to be shared between CDH clusters created using Altus Director, Cloudera Manager, or Altus cloud services.

Altus Data Engineering or Altus Data Warehouse clusters

Altus services clusters use an Altus SDX namespace to point to databases for shared data and metadata. To share data with clusters created using Altus Director or Cloudera Manager, set up a configured SDX namespace in Altus services to point to the same databases used for the clusters in Altus Director and Cloudera Manager.

Setting Up Cloudera SDX

To share data between CDH clusters, set up the Hive metastore and Sentry databases for the clusters created with Cloudera Manager or Altus Director. Then, in Altus services, create a configured SDX namespace that uses the same Hive metastore and Sentry databases. Altus Data Warehouse or Altus Data Engineering clusters that use the configured SDX namespace can share metadata and access policies with clusters created with Altus Director or Cloudera Manager that use the Hive metastore and Sentry databases.

On AWS, you can manually set up or use Altus Director to set up a database for the Hive metastore and Sentry data. On Azure, manually set up a database for the Hive metastore and Sentry data.

To set up Cloudera SDX for CDH clusters, complete the following steps:

1. **Set up the Hive metastore and Sentry databases and create a cluster using Altus Director or Cloudera Manager.**

2. **Set up a configured SDX namespace and cluster in Altus.**

Setting up the Databases and Clusters with Altus Director or Cloudera Manager

You can use Altus Director to deploy CDH clusters on the cloud. You can also use Cloudera Manager to create and manage clusters.

To set up the databases and clusters in Altus Director or Cloudera Manager, complete the following steps:

1. Set up external databases for the Hive metastore and Sentry databases.

   On AWS, configure Altus Director to create an Amazon Relational Database Service (RDS) instance and set up databases for the Hive metastore and Sentry data. When you configure the RDS, set the configuration parameters according to the number of clusters that will share the RDS. For example, scale up the maximum number of connections to accommodate the number of clusters sharing the database. For more information about setting up a Hive metastore database using RDS, see the blog post, How To Set Up a Shared Amazon RDS as Your Hive Metastore.

   On Azure, install a database server and create databases for the Hive metastore and Sentry.

   Make note of the following details for each database that you create:

   - Host name and port number
   - Database name
   - User name and password for the database administrator

2. Create the cluster using Altus Director or Cloudera Manager.

   If you use Altus Director to create a cluster, follow the SDX Director template. The Kerberos configuration settings in the template are required for SDX. Update the parameters that show a value of *REPLACE-ME* with values appropriate for your setup.

   If you use Cloudera Manager to create a cluster, enable Kerberos so that the clusters can share Sentry access policies.

   On AWS, create the cluster in the same Amazon Virtual Private Cloud (VPC) where you set up the database servers. You can use Altus Director to create the schema in the Hive metastore and Sentry database.

   On Azure, install Altus Director and create the cluster on the same virtual network where you set up the database server.

   When you create the cluster, complete the following tasks:

   - **Configure it to use the Hive metastore and Sentry external databases.**

     Set the database hostname, port number, and login credentials for the Hive metastore and Sentry databases that you set up in Step 1 on page 10. The Hive services in all the clusters must share the same *Server Name for Sentry Authorization* property.

     Restart the Hive and Sentry services.

   - **Set up access from the clusters to your data in object storage.**

     The cluster must have read and write privileges to the object storage so that workloads can create and update data and metadata in the Hive metastore and Sentry databases.

     On AWS, you can enable access to Amazon S3 for the hosts in your cluster in one of the following two ways:

     - Configure the Amazon S3 connector in your cluster.

       Define AWS credentials in Cloudera Manager and then add and configure the S3 Connector Service to use the AWS credentials. For more information about configuring secure access to Amazon S3 through the S3 connector, see Configuring the Amazon S3 Connector

     - Configure all hosts with IAM role-based authentication that allows access to S3.

If you configure role-based authentication, you do not need to add the Amazon S3 connector service or use S3Guard.

On Azure, enable access to ADLS for the hosts in your cluster in one of the following ways:

- Configure the ADLS Connector in your cluster.

  Configure ADLS credentials and add the ADLS Connector service to your cluster. The ADLS connector service provides a secure connection between the cluster and ADLS. For more information about configuring secure access to ADLS through the ADLS connector, see Configuring ADLS Access Using Cloudera Manager.

- Set up a connection from cluster to ADLS through properties in configuration files.

  For more information about configuring connectivity to ADLS through configuration files, see Configuring ADLS Connectivity for CDH.

  Note that using configuration files to provide access to ADLS for jobs running in a cluster can compromise security. Cloudera recommends that you use this method only to access ADLS in development environments or other environments where security is not a concern.

3. Note the URI and administrator credentials for the Hive metastore and Sentry databases.

   You can find the database URI on the Altus Director web UI. On the **Cluster Details** page go to the **Database Servers** section. Click **Learn More** to display the JDBC URLs for the Hive metastore and Sentry databases.

   For each database, make note of the following details:

   - Connection URI
   - User name and password for the database administrator

4. Grant administrator privileges to the Sentry administrator group.

   You can use Hue, Beeline, or impala-shell to set permissions for the Sentry groups that access the Hive metastore. All clusters that use the same Sentry groups have the same access permissions to data and metadata in the Hive metastore.

   Sentry permissions are not granted per cluster: all clusters use the same permissions data. Every Hive service must share the same value in the *Server Name for Sentry Authorization* property.

You can set up Cloudera Navigator to display lineage data from Altus Data Engineering clusters on AWS. For more information, see Cluster Lineage for CDH Clusters on page 13.

Setting up an Altus SDX Namespace and Cluster

Altus SDX is an Altus cloud service that externalizes cluster metadata to provide a consistent view of data for multiple clusters and workloads running on the cloud. You can set up a configured SDX namespace in Altus services that points to an existing Hive metastore and Sentry databases for CDH clusters on the cloud. With a configured SDX namespace, you can share data between clusters created with Altus Director or Cloudera Manager and Altus Data Engineering and Altus Data Warehouse clusters.

To set up a cluster with a configured SDX namespace in Altus, complete the following steps:

1. Identify the Hive metastore and Sentry databases that you want to use for the configured SDX namespace.

   Get the connection URI and administrator credentials for the Hive metastore and Sentry databases that you set up for the cluster in Altus Director.

2. Create a configured SDX namespace in Altus.

   To create a configured SDX namespace, you must be an Altus administrator or have the Altus *SdxAdmin* role.

   When you create the configured SDX namespace, you must provide the connection URI and administrator credentials for the Hive metastore and Sentry databases. In the **Hive Metastore Settings** section, specify the JDBC URI and the database administrator credentials for the Hive metastore database. In the **Sentry Settings** section, specify the JDBC URI and the database administrator credentials for the Sentry database.

For more information about creating a configured SDX namespace, see Creating an SDX Namespace in the Altus documentation.

3. Create an Altus Data Engineering or Data Warehouse cluster and set up the cluster to use the configured SDX namespace.

   Set the following parameters when you create the Altus cluster:

   - ***SDX Namespace***. Specify the configured SDX namespace that uses the Hive metastore and Sentry databases for the clusters in Altus Director.
   - ***Environment***. Specify an environment with the *Secure Clusters* option enabled. The environment must also be set up to allow access to the Hive metastore and Sentry databases.

     If the environment you specify also has the *Cloudera Navigator integration* option enabled, you can view cluster lineage in Cloudera Navigator for Altus Data Engineering clusters on AWS. For more information, see Cluster Lineage for CDH Clusters on page 13.

   Note that a cluster that has the MapReduce2 service cannot use a configured SDX namespace.

4. Grant all privileges to the Sentry administrator group.

   When you create a configured SDX namespace, Altus creates an Altus group and sets it up as an administrator group in Sentry. You, as creator of the configured SDX namespace, are a member of the group by default. If you use the configured SDX namespace with an Altus Data Engineering cluster, Altus also adds the *altus* user account, which runs data engineering jobs, to the group. Members of the Sentry administrator group can create roles and grant privileges in Sentry.

   You must create a role with *all* privileges and assign the role to the SDX Sentry administrator group so that members of the group can create and manage the databases as required.

   To grant all privileges to the SDX Sentry administrator group, run the following commands:

```
create role SentryAdminRoleForAltus;
grant all on server server1 to role SentryAdminRoleForAltus;
grant role SentryAdminRoleForAltus to group SDXSentryAdminGroup;
```

   For Altus Data Engineering clusters, submit a Hive job to run the commands.

   For Altus Data Warehouse clusters, use the Query Editor to run the commands.

5. If you are working with Altus Data Warehouse clusters, set up the users and groups in Altus and synchronize them with the cluster.

   For more information about setting up users and group for Altus Data Warehouse clusters, see Setting Up User Access to an Altus Data Warehouse Cluster.

Guidelines for Using Cloudera SDX

Use the following guidelines when you set up Cloudera SDX for CDH clusters:

When you create databases and tables, use the *LOCATION* attribute to write the data to cloud storage.

When you create a database or table that you want to make accessible to multiple clusters, you must create the database or table in cloud storage. On AWS, create the databases in Amazon S3. In Azure, create the databases in Azure Data Lake Store (ADLS).

Use the *LOCATION* attribute to indicate the location in cloud storage where you want to create the database or table. If you do not provide the location, the database or table is created in a default location in HDFS in the cluster. When the cluster is terminated, the HDFS databases and tables are lost.

The following examples show the CREATE DATABASE statement with the LOCATION attribute pointing to Amazon S3 and ADLS locations:

```
CREATE DATABASE databasename LOCATION s3a://path-to-aws-s3/dir/db
CREATE DATABASE databasename LOCATION adl://path-to-azure-adl/dir/db
```

For more information about creating a database using Impala SQL statements, see CREATE DATABASE Statement.

The following examples show the CREATE TABLE statement with the LOCATION attribute pointing to Amazon S3 and ADLS locations:

```
CREATE EXTERNAL TABLE tablename LOCATION s3a://path-to-aws-s3/dir/table_data
CREATE EXTERNAL TABLE tablename LOCATION adl://path-to-azure-adl/dir/table_data
```

For more information about creating a table using Impala SQL statements, see CREATE TABLE Statement.

To view the location attribute of a database, use the DESCRIBE DATABASE statement:

```
DESCRIBE DATABASE databasename
```

Avoid concurrent updates by multiple clusters to the same schema, table, or partitions in a table.

Cloudera SDX does not manage the metadata updates made by different clusters. It does not have a mechanism to lock the metadata to prevent simultaneous updates by multiple clusters. Data conflicts and errors can arise if multiple clusters sharing an SDX namespace access a dataset at the same time and perform conflicting updates.

For example, problems can arise if multiple clusters concurrently update the same table or partitions within a table or add or change the same schema or database.

Run your workloads in a way that ensures that multiple clusters do not make overlapping data or metadata changes.

Avoid concurrent updates by multiple clusters to the same Sentry groups and roles.

Cloudera SDX does not manage the updates to the Sentry groups and roles made by different clusters. It does not have a mechanism to prevent clusters from concurrently setting or updating permissions on the same metadata. Errors can arise if multiple clusters perform conflicting updates to the same groups and roles at the same time.

Ensure that interim local tables in HDFS are deleted before you terminate a cluster.

When you write interim data to a table stored in HDFS in a cluster, the metadata for the interim files is stored in the Hive metastore tables. If you terminate the cluster, Cloudera SDX does not delete the metadata for these tables from the Hive metastore.

The metadata remains in the Hive metastore tables and can cause data conflicts and errors for other clusters. For example, a job in another cluster that uses the same Hive metastore tables might try to read data in the tables and encounter errors because the HDFS locations do not exist or are not be valid.

To avoid errors with orphaned metadata in the Hive metastore tables, delete all tables created in HDFS before you terminate a cluster.

Cluster Lineage for CDH Clusters

Cloudera Navigator provides a unified view of cluster metadata and lineage across all clusters managed by a Cloudera Manager instance. In Altus Director, you can deploy multiple clusters to be managed by one Cloudera Manager instance. However, when you create clusters with Altus cloud services, each cluster is managed by a separate Cloudera Manager instance.

Altus services provide a Cloudera Navigator integration option that enables Altus Data Engineering clusters to send workload metadata to an Amazon S3 bucket configured as a metadata resource in Cloudera Navigator. On AWS, you can include Altus Data Engineering cluster metadata to the lineage information of clusters managed by another Cloudera Manager.

If you create an Altus Data Engineering cluster that uses an environment with the Cloudera Navigator integration option enabled, Cloudera Navigator can extract metadata from the S3 bucket to generate analytics and data lineage for the cluster. You can configure Altus to send cluster and workload metadata to an S3 bucket that is used by Cloudera Navigator for clusters managed by a Cloudera Manager instance. You can then configure Cloudera Navigator to extract metadata from the shared S3 bucket to display cluster lineage for clusters a Cloudera Manager instance and the Altus Data Engineering cluster.

To view metadata and lineage in Cloudera Navigator for CDH clusters managed by a Cloudera Manager instance and Altus Data Engineering clusters, complete the following steps:

1. In the Cloudera Manager instance that manages the clusters for which you want to display lineage data, enable the option to extract metadata and lineage data from Altus and specify the Amazon S3 bucket for metadata extraction.

   Follow the instructions in Cloudera Navigator Configuration to set up the AWS credentials and connectivity and configure Cloudera Navigator to enable metadata and lineage extraction from Altus services clusters.

2. In Altus services, enable the Cloudera Navigator integration option in the Altus environment of the clusters from which Cloudera Navigator can extract lineage metadata. Specify the Amazon S3 bucket configured as a resource for metadata extraction in Cloudera Navigator.

   The S3 bucket must be the same S3 bucket that is configured for metadata and data lineage for the CDH clusters with which you want to share data.

   For more information about the Cloudera Navigator integration option, see Altus Environment Options in the Altus documentation.

   For more information about creating an environment with the Cloudera Navigator integration option enabled, see Altus Environment Setup for AWS in the Altus documentation.

For more information about using Cloudera Navigator with Altus clusters, see Using Cloudera Navigator with Altus Clusters.

Supported Services and Components

You can set up Cloudera SDX to share data between clusters deployed with Altus Director or Cloudera Manager in AWS or Azure and secure Altus Data Warehouse and Altus Data Engineering clusters.

The following table shows the components supported for clusters in Altus Director and the Altus services that share data and metadata through Cloudera SDX:

Component	Cloudera Manager or Altus Director	Altus Services
Database	• AWS: MySQL databases set up by Director using Amazon Relational Database Service (RDS) or set up manually by the user • Azure: MySQL or PostgreSQL databases set up by the user For more information about the databases that Altus Director supports for Cloudera Manager and CDH, see Supported Software and Distributions.	Configured SDX namespace that uses databases deployed and initialized by Altus Director or Cloudera Manager.
Database server location	• AWS: Set up on the same Amazon Virtual Private Cloud (VPC) as the cluster. • Azure: Set up on the same virtual network as the clusters.	Specify the URI that points to the Hive metastore and Sentry databases for clusters created with Altus Director. The Hive metastore and Sentry databases must be accessible from the Altus services clusters.

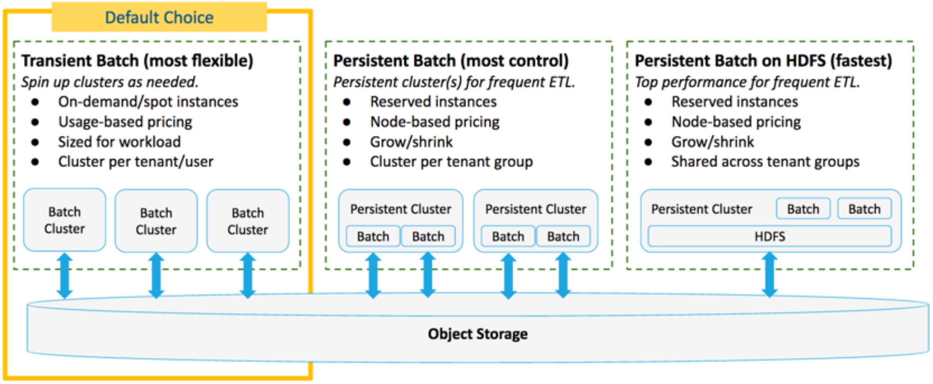
Component	Cloudera Manager or Altus Director	Altus Services
CDH version	• CDH 5.13 or later versions • All clusters must use the same CDH version.	• Altus Data Engineering cluster: CDH 5.13 or later versions • Altus Data Warehouse cluster: CDH 5.14 or later versions
Cloudera Navigator	• Requires CDH 5.13.1 or a later version • Cloudera Navigator must be on the same node as the Cloudera Manager instance that manages the clusters.	• For Altus Data Engineering clusters on AWS only. • Requires CDH 5.13.1 or a later version
File System	• AWS: Amazon S3 • Azure: Azure Data Lake Store (ADLS) Gen 1	• AWS: Amazon S3 • Azure: Azure Data Lake Store (ADLS) Gen 1

Data Engineering on AWS: Best Practices

For most data engineering and ETL workloads, best performance and lowest cost can be achieved using the default recommendations described below.

Basic Architectural Patterns

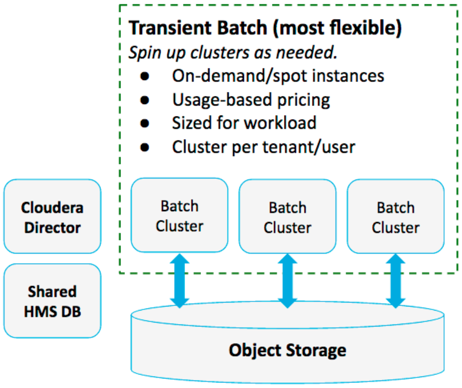
Cloudera recommends the following architectural patterns for three common types of data engineering workloads:



Choose one of these patterns, depending on your particular workloads, to ensure optimal price, performance, and convenience.

Pattern #1: Transient Batch Clusters on Object Storage

Use transient clusters and batch jobs to process data in object storage on demand.



This pattern is ideal when jobs are asynchronous or unpredictable, and run on an irregular basis, for fewer than 50% of weekly hours. This pattern can result in lower cost for two reasons:

- You only spin up clusters as they are needed, and only pay for the cloud resources you use
- You are able to select an instance type for each job, ensuring that jobs run on the most suitable hardware, with maximum efficiency

Benefits:

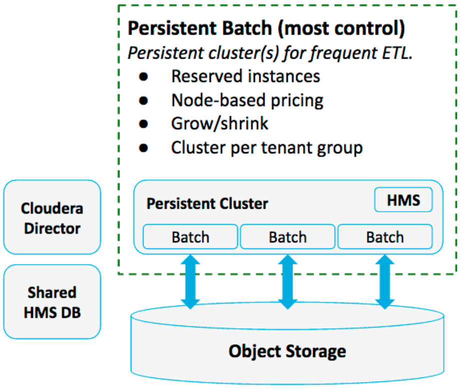
- Enables quick iteration with different instance types and settings
- Instances and software can be tailored to specific workloads
- Workloads run in complete isolation
- You can use spot instances for worker nodes, which lowers costs even further
- You can size your environment optimally, depending on the batch size

Tradeoffs:

- You incur the cost of start and stop time for each cluster
- On-demand instances cost more per hour than long-running instances
- You cannot use Cloudera Navigator with transient instances, since instances are terminated when a job completes

Pattern #2: Persistent Batch Clusters on Object Storage

Use persistent clusters to process data in object storage when your jobs are so frequent that you are able to keep a single cluster working for 50% or more of weekly hours with a series of separate jobs.



This pattern results in a lower cost per job, and works well for homogeneous jobs that can run efficiently with the same cluster setup, using the same hardware and software.

Benefits:

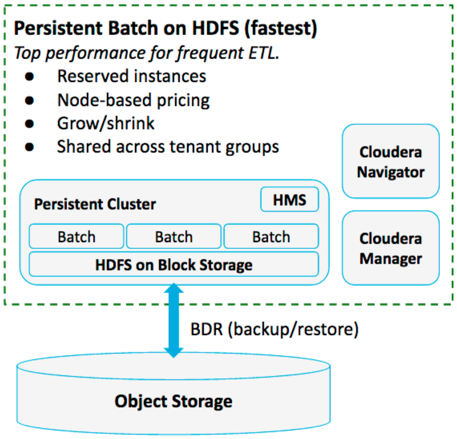
- No costly job time is spent in starting and stopping clusters
- You can use cheaper reserved instances to lower overall cost
- You can grow and shrink your clusters as needed, always maintaining the most cost-effective number of instances
- Cloudera Navigator is supported with Cloudera Enterprise 5.10 and higher

Tradeoffs:

- Less workload isolation
- Less flexibility in terms of instance types and cluster settings

Pattern #3: Persistent Batch Clusters on Local Storage

Use persistent "lift and shift" clusters on data in local HDFS storage for maximum performance.



Here are three common scenarios where this pattern is ideal:

- *Performance*. When you have, for example, AWS workloads that run too slowly on object storage, or Azure workloads that are unsupported with object storage, use block storage with HDFS.
- *Encryption*. Cloudera Enterprise doesn't support Amazon S3 client-side encryption. If you need to manage your own encryption keys, you should use HDFS and encrypt your data there.
- *Efficiency*. With lift-and-shift jobs, you may want to combine data engineering and data warehouse workloads in the same cluster. For more information, refer to Data Warehouse on AWS.

Benefits:

- No costly job time is spent in starting and stopping clusters
- You can use cheaper reserved instances to lower overall cost
- Faster performance per node on local data
- The full Cloudera Enterprise feature set is available, including encryption, lineage, and audit.

Tradeoffs:

- Clusters are less elastic with HDFS than with object storage
- Less workload isolation
- Less flexibility in terms of instance types and cluster settings

Typical Data Engineering Scenario

- You have data stored in AWS S3 in an unprocessed, raw format.
- Data engineers prepare ETL queries in a development environment using some sample of the raw data.
- Final queries go to a production environment where they are executed in recurring transient clusters provisioned by Altus Director.
- Processed data is often read by a data warehouse.

Summary of Default Recommendations

- **Altus Director:** Use Altus Director to deploy Cloudera Manager and provision and scale CDH clusters.
- **Transient clusters:** Recommended for lowest cost if clusters will be busy less than 50% of the time.
- **Master nodes:**
  - Place all master services on a single node, with Cloudera Manager on a separate node.

    > **Note:** Altus Director will do this by default during cluster deployment by creating a **masters** instance group with an instance count of 1. See Deploying Cloudera Manager and CDH on AWS in the Altus Director documentation.

  - Do not use spot instances for master nodes.
- **Worker nodes:**
  - Use more nodes for better performance and maximum S3 bandwidth.
  - For lower cost, use spot instances for worker nodes. Be aware that spot instances are less stable than on-demand instances. See Spot Instances in the AWS documentation for more information.
- **Compute engines:** MapReduce, Hive, Spark, Hive-on-Spark. Use Spark or Hive-on-Spark rather than MapReduce for faster execution.
- **EC2 instance types:** Use m4.2xlarge for workloads with a balance of compute and memory requirements.
  - Use c4.2xlarge for compute-intensive workloads, such as parallel Monte Carlo simulations.
  - Use r3.2xlarge or r4.2xlarge for memory-intensive workloads, such as large cached data structures.
- **Storage:** Use S3 for storage of input data and final output, and use HDFS for storage of intermediate data.

  - Compress all data to improve performance.
  - Avoid small files when defining your partitioning strategy.
  - Use Parquet columnar data format on S3.
  - Impala block size: 256 MB
  - Change S3A to “fs.s3a.block.size” to match block size.

  For information about configuring Hive ETL jobs to use Amazon S3, see Configuring Transient Hive ETL Jobs to Use the Amazon S3 Filesystem in the Cloudera Enterprise documentation.
- **Security:** Launch the cluster in a VPC with strict security groups, as described in Cloudera Enterprise Reference Architecture for AWS Deployments (PDF).

Transient Clusters vs. Permanent Clusters

- On the cloud, you have a choice of transient or permanent clusters. Most batch ETL and data engineering workloads are transient: they are intended to prepare a set of data for some downstream use, and the clusters don't need to stay up 24x7. A transient cluster is launched to run a particular job and is terminated when the job is done. This results in a lower total cost of ownership (TCO), since you only pay for what you use in a cloud environment.
- Transient clusters have additional benefits over permanent clusters besides lowering your Amazon bill for EC2 compute hours. They offer maximum flexibility, enabling you to choose different cluster configurations for different jobs instead of running all jobs on the same permanent cluster with a particular configuration of hardware and a given set of CDH services. With transient clusters, you can experiment with different tools with lower risk and see which work best for your needs. You can also ensure that instance types are ideally suited for each job, depending on factors such as whether your workload is compute intensive or memory intensive.
- When a cluster running transient workloads is used on a very frequent basis, running ETL jobs 50% or more of total weekly hours, a permanent long-running cluster may be more cost effective than a series of transient clusters because it allows you to take advantage of EC2 Reserved Instance pricing instead of more expensive on-demand instances. For more information on EC2 Reserved Instances pricing see Amazon EC2 Reserved Instances Pricing.

Data Storage Considerations

Amazon S3 Object Storage

While not the highest performing storage option, Amazon S3 has considerable advantages, including low cost, fault tolerance, scalability, data persistence, as well as compatibility with other AWS services.

Processing data directly in S3, instead of relying on HDFS, for ETL workloads also increases flexibility by decoupling storage and compute. You can keep your data on S3, process or query it on a transient cluster with a variety of CDH tools, store the output data back on S3, and then access the data later for other purposes after terminating the cluster.

Cloudera’s default recommendation is to use S3 to store initial input and final output data, but to store intermediate results in HDFS. There are three important benefits to this approach:

1. It lowers costs by reducing local HDFS storage requirements.
2. It avoids consistency problems with S3.
3. If you store intermediate results in S3, that data is streamed between every worker node in the cluster and S3, significantly impacting performance.

Important:

Cloudera components writing data to Amazon S3 are constrained by the inherent limitation of S3 known as "eventual consistency." For more information, see Introduction to Amazon S3 in the AWS documentation.

In rare conditions, this limitation of S3 may lead to some data loss when a Spark or Hive job writes output directly to S3.

Solution: Use S3 only for the final output. From your Spark or Hive job, first write the final output to local HDFS on the cluster, and then use distcp to copy the data from HDFS to S3.

In an upcoming CDH release, Cloudera will provide a solution that enables direct writes from a Spark or Hive job to S3 without data loss.

Additional Suggestions

The following are additional suggestions for maximizing performance and minimizing costs on transient clusters for ETL workloads:

- For jobs where I/O is a bottleneck to performance:
  - Preload data from S3 into HDFS if the data does not fit in memory thereby requiring multiple roundtrips to disk. This can speed things up whether HDFS is running on ephemeral disk or on EBS.
  - Use gzip to reduce the size of input data.
  - Consider using Snappy for data compression if your bottleneck is CPU-related.
- On S3, avoid over-partitioning at too fine a granularity, since small files are not handled efficiently on S3. S3 may limit performance if too many files are requested. For more information, see Request Rate and Performance Considerations in the AWS documentation.
- Use Cloudera Manager to monitor workloads. Use one instance of Altus Director per user or user group based on AWS resource permissions.
  - A user group in this context means a set of users who have the same level of permissions to launch EC2 instances or create AWS resources.
  - Deploy Altus Director on an instance with the right IAM role for that group.
- Copy all relevant cluster log files to S3 before destroying the cluster to enable debugging later.
- Use a single cluster to run multiple jobs if the jobs run continuously or as a dependent sequential pipeline, especially if cluster start/stop time exceeds job runtime.

  Note: This suggestion only applies if all jobs are submitted by the same tenant, and all jobs can be run efficiently using the same hardware profile.

- With applications that benefit from low network latency, high network throughput, or both, use placement groups to locate cluster instances close to each other. See Placement Groups in the AWS documentation for more information.

For Workloads Using Cloudera Navigator

If you need to track lineage for workloads with Cloudera Navigator, transient clusters are not supported. Follow these guidelines instead:

- Use a persistent cluster.
- The cluster should be managed by Cloudera Manager.
- The cluster should use HDFS for storage.

Data Warehouse on AWS: Best Practices

A data warehouse job is often performed in two distinct phases:

- A SQL-based ETL/data engineering phase to prepare the data, typically using either Hive on MapReduce or Hive on Spark.
- A BI/SQL analytics phase, typically using Impala and possibly BI tools or SQL applications.

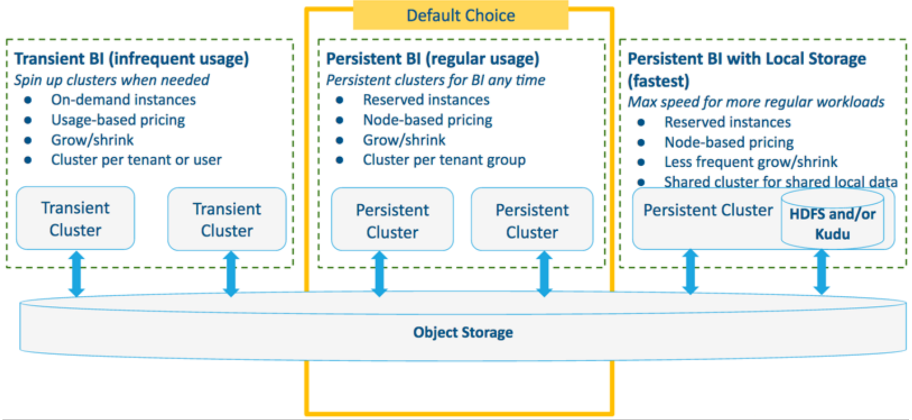
The first of these phases, the ETL or data engineering phase, should generally follow the best practices for data engineering workloads that are described in the previous section of this document, Data Engineering on AWS.

The section will concentrate on the second phase, the BI and SQL analytics phase. This is where reports, visualizations, or applications are produced for people to use in real time, requiring interactive response times and higher concurrency.

For most data warehouse workloads, greatest flexibility and lowest cost can be achieved using the default recommendations described below.

Basic Architectural Patterns

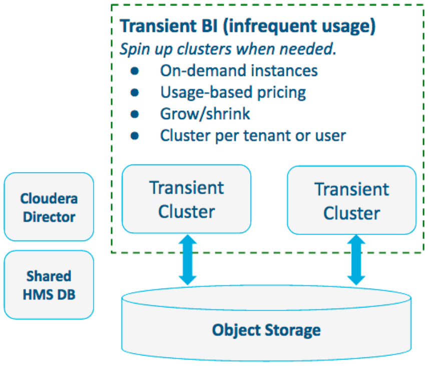
Cloudera recommends the following architectural patterns for three common types of data warehouse workloads:


Default Choice
Transient BI (infrequent usage)
Spin up clusters when needed
On-demand instances
Usage-based pricing
Grow/shrink
Cluster per tenant or user
Transient Cluster
Transient Cluster
Persistent BI (regular usage)
Persistent clusters for BI any time
Reserved instances
Node-based pricing
Grow/shrink
Cluster per tenant group
Persistent Cluster
Persistent Cluster
Persistent BI with Local Storage (fastest)
Max speed for more regular workloads
Reserved instances
Node-based pricing
Less frequent grow/shrink
Shared cluster for shared local data
Persistent Cluster
HDFS and/or Kudu
Object Storage

Choose one of these patterns, depending on your particular workloads, to ensure optimal price, performance, and convenience.

Pattern #1: Transient BI Workloads on Object Storage

For transient business intelligence (BI) workloads where there is infrequent usage, use on-demand instances, and spin up clusters when needed.



This pattern is best when jobs run on an irregular basis, for fewer than 50% of weekly hours. This pattern can result in lower cost for two reasons:

- You only spin up clusters as they are needed, and only pay only for the cloud resources you use
- You are able to select an instance type for each job, ensuring that jobs run on the most suitable hardware, with maximum efficiency

Benefits:

- Enables quick iteration with different instance types and settings
- Instances and software can be tailored to specific workloads
- Workloads run in complete isolation
- You can use spot instances for worker nodes, which lowers costs even further
- You can size your environment optimally, depending on the batch size

Tradeoffs:

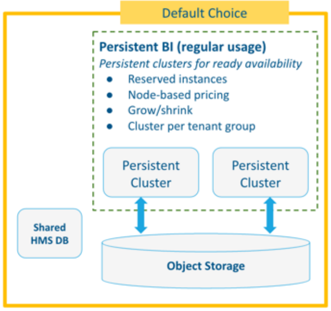
- You incur the cost of start and stop time for each cluster
- On-demand instances cost more per hour than long-running instances
- You cannot use Cloudera Navigator with transient instances, since instances are terminated when a job completes

The following characteristics are typical for this type of cluster:

- On-demand instances
- Usage-based pricing
- Grow and shrink clusters to minimize cost
- Cluster per tenant or user

Pattern #2: Persistent BI on Object Storage

This pattern is usually the best choice when BI workloads run frequently but are flexible and changeable, and are not scheduled for fixed hours.



This is usually the best choice when workloads are:

- Flexible and changing
- Frequent during most working days
- Not scheduled for fixed hours

Benefits:

- Predictable results are readily available
- Full multi-tenant isolation
- Common data can be placed in shared object storage
- Minimal cost can be achieved by growing and shrinking clusters as needed

Tradeoffs:

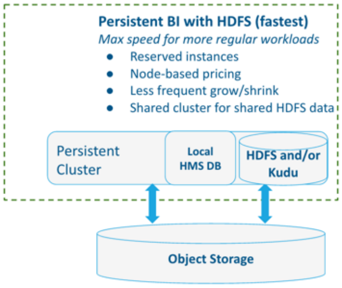
- Per node performance is lower with S3 object storage. You can compensate for this by using more, cheaper nodes.

The following characteristics are typical for this type of cluster:

- Reserved instances
- Node-based pricing
- Grow and shrink clusters to minimize cost
- Cluster per tenant group (where a single cluster is shared by a group of tenants)

Pattern #3: Persistent BI with Locally-Attached Storage

To run frequent, regular BI workloads at maximum speed, Cloudera recommends using persistent clusters and local HDFS or Kudu storage.


Persistent BI with HDFS (fastest)
Max speed for more regular workloads
Reserved instances
Node-based pricing
Less frequent grow/shrink
Shared cluster for shared HDFS data
Persistent Cluster
Local HMS DB
HDFS and/or Kudu
Object Storage

This pattern is the best choice when workloads are:

- Regular and consistent
- Consistently querying common data
- Tight SLAs for performance
- Fast changing data (that needs Kudu)
- Running without object storage (eg. Azure, GCE)

Benefits:

- Faster performance per node on local data
- Ability to query object storage for rest of data

Tradeoffs:

- Less elastic than object stored based clusters
- Less isolation for multi-tenant workloads using same HDFS data
- Cost if there are off-peak hours

The following characteristics are typical for this type of cluster:

- Use of reserved instances
- Node-based pricing
- Less frequent grow/shrink
- Shared cluster for shared local data

Typical Data Warehouse Scenario

A typical scenario for a data warehouse workload is as follows:

- You have data stored in AWS S3 in an unprocessed, raw format.
- Users prepare their ETL tasks in a development environment using some sample of the raw data.
- Final ETL jobs go to a production environment where they are typically executed in recurring transient clusters (as described in the Data Engineering on AWS section).
- Processed data is often served for BI and SQL analytics in one the patterns above.

Basic Guidelines for BI in the Cloud

- **Compute engine:**
  - Impala for most BI workloads
  - See the section Data Engineering on AWS for SQL-based ETL via Hive and Hive-on-Spark
- **Data Formats:**
  - Use Parquet, Snappy, and 256 MB blocks for best IO efficiency, especially on object storage
  - Avoid small files for better performance, especially with object storage
- **Metadata, including Hive Metastore Service (HMS):**
  - Use local RDBMS for clusters that use locally stored HDFS or Kudu data (as described in Pattern#3 above).
  - Use a shared RDBMS for clusters that share common object store data (as described in Pattern #1 and Pattern #2 above).
    - For information on using a shared Amazon RDS as a metastore, see the Cloudera Engineering blog How To Set Up a Shared Amazon RDS as Your Hive Metastore
  - Avoid sharing a RDBMS for non-data sharing clusters
- **Deployment and Administration:**
  - Use either Cloudera Manager or Altus Director for persistent clusters. Cloudera recommends using Altus Director for persistent clusters that will grow and shrink. Use Altus Director 2.4 or higher for this because of enhanced interaction and synchronization between Altus Director and Cloudera Manager.
  - Use Altus Director for transient clusters
  - Deploy a network load balancer (NLB) for persistent clusters as usual, and only when needed for transience
- **Monitor workloads with Cloudera Manager**
- **Security:**
  - Transient BI clusters: configure a VPC with strict security groups
  - Persistent BI clusters: see Security Guidelines for BI in the Cloud on page 24 below.

Security Guidelines for BI in the Cloud

> **Note:** For transient clusters, or for persistent clusters where all cluster users have access to all data in the cluster, you may be able to disregard some of the recommendations in this section and simply configure a VPC with strict security groups.

- **Identity:**
  - Tie the cluster to the corporate user directory (AD or LDAP) using Linux SSSD
- **Authentication:**
  - Kerberos for internal CDH services
  - AD (LDAP) for BI user/tools with Impala
- **Authorization:**
  - Use Sentry for authorization per BI cluster (shared RDBMS with Sentry is not yet available)
- **Encryption:**
  - With AWS, use SSE-S3 server-side encryption

  - If you need HDFS-level encryption or keys outside AWS, then use Persisted Cluster with local storage using Cloudera Manager

Instance Recommendations

The following are instance recommendations for BI workloads:

- For worker nodes, scale-up rather than out:
  - Around 100 nodes for optimal performance
  - If CPU constrained for cost efficiency
  - Use EBS (st1) for spill to disk with r4 instances (size depends on usage)
- For CM master at >50 worker nodes:
  - Add 2nd Cloudera Manager master for CM monitoring services
  - Start with ~50 GB EBS (gp2) for CM
- For greater flexibility and cost savings, consider using common reserved instances for data engineering and data warehouse workloads.

Table 1: Default (object store based)

Type	AWS
Worker nodes	r4.2xlarge
Cloudera Manager master	m4.4xlarge
CDH masters	r4.4xlarge

Table 2: Locally-attached storage (with HDFS or Kudu)

Type	AWS	Azure	Google
Worker nodes	d2.4xlarge (reserved)	DS13v2 + P30 disks	n1-highmem -8
Cloudera Manager master	m4.4xlarge	DS13v2	n1-standard -16
CDH masters	r4.4xlarge	DS14v2	n1-highmem -16

Additional Suggestions

General:

- Be sure to use the same instance type for all worker nodes.

Object storage:

- Default scale-up choice for worker nodes: r4.4xlarge, then r4.8xlarge, and so on.

Locally-attached storage:

- If running with higher concurrency, use d2.8xlarge, DS14v2, or n1-highmem-16 instead for better concurrent performance.
- Cloudera recommends that you do not use d2.2xlarge instances. There is a risk of data loss with this instance type because AWS can put up to 4 nodes on the same host.

> **Note:** Note that Azure and GCE don’t have this data loss risk for lower instances so can run on lower node types.

- If running Azure, set rack to equal fault domain to avoid data loss per Azure Guard Rails.
- Reserve pricing is typically expected for locally-attached storage clusters.
- With on-demand pricing, r4 instances coupled with st1 EBS storage is cheaper than the equivalent d2 instances.
- AWS EBS should use st1 disks for larger disks for best cost-performance and gp2 for smaller EBS volumes because:
  - st1 provides 40 MB/s per TB provisioned up to 500 MB/s (eg 4 TB st1 volume = 160 MB/s)
  - gp2 peaks at 160 MB/s
  - gp2 provides better random IO which should be atypical for Impala workloads
  - gp2 is over 2x the cost of st1

Operational Database on AWS: Best Practices

By running operational database workloads in the cloud, you can leverage inexpensive object storage to manage periodic data inflows. Operational use cases are defined by a unique set of characteristics:

- A dynamic data environment, with rapid read/writes/inserts/updates/deletes.
- An “always on” need for compute resources to meet real-time business demands.
- In most cases, requirement for the more performant EBS over the more cost-effective S3.

Leveraging cloud elasticity by deploying operational databases on public cloud platforms gives you lower cost and greater convenience. Typical uses of operational databases in the cloud include the following:

- Burst ETL to overcome data ingest bottlenecks.
- Rapid provisioning of new persistent clusters.
- Development and testing environments.

This document describes best practices for running operational database workloads.

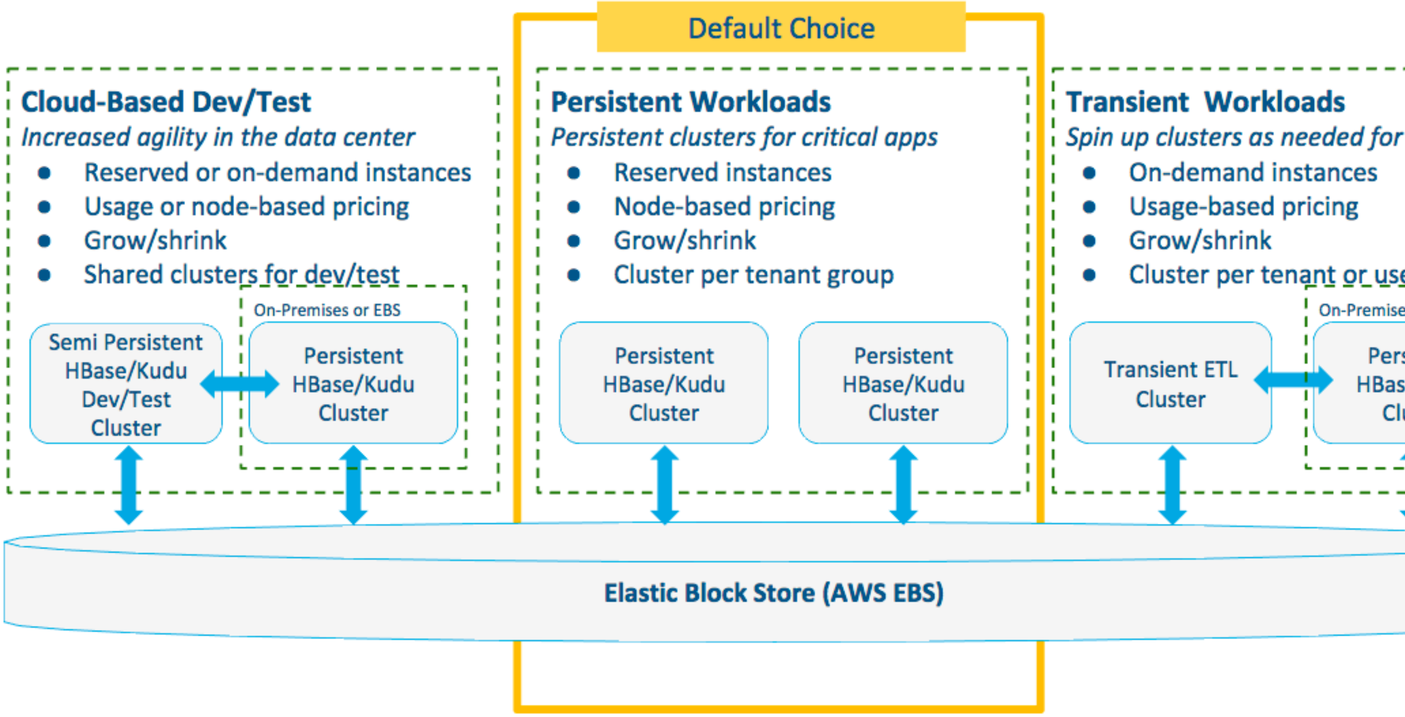
Typical Operational Database Scenario

A typical scenario for an operational database workload is as follows:

- Applications/use cases require access to real-time data
- The underlying pool of structured/unstructured operational data is updated and appended to on an ongoing basis
- The current infrastructure environment was designed to have limited excess capacity to save on costs, but the business runs into short/medium-term scenarios where infrastructure needs outstrip capacity. Users want the flexibility to respond to business needs.

Basic Architectural Patterns

Cloudera recommends the following architectural patterns for three common types of data warehouse workloads:

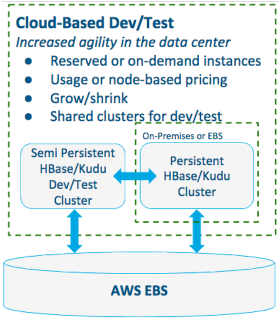

Default Choice
Cloud-Based Dev/Test
Increased agility in the data center
● Reserved or on-demand instances
● Usage or node-based pricing
● Grow/shrink
● Shared clusters for dev/test
On-Premises or EBS
Semi Persistent HBase/Kudu Dev/Test Cluster
Persistent HBase/Kudu Cluster
Persistent Workloads
Persistent clusters for critical apps
● Reserved instances
● Node-based pricing
● Grow/shrink
● Cluster per tenant group
Persistent HBase/Kudu Cluster
Persistent HBase/Kudu Cluster
Transient Workloads
Spin up clusters as needed for
● On-demand instances
● Usage-based pricing
● Grow/shrink
● Cluster per tenant or use
On-Premise
Transient ETL Cluster
Pers
HBas
Cl
Elastic Block Store (AWS EBS)

Choose one of these patterns, depending on your particular workloads, to ensure optimal price, performance, and convenience, keeping the following points in mind:

- Different use case requirements may lead to different choices.
- These options can be used for different use cases side-by-side.
- ETL systems run alongside the BI/Analytics use case, sharing a common data platform.

Pattern #1: Provisioning Cloud-based Dev and Test Environments

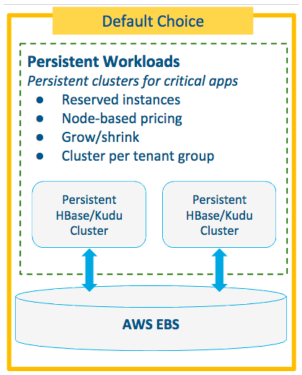
For provisioning development and testing environments.


Cloud-Based Dev/Test
Increased agility in the data center
Reserved or on-demand instances
Usage or node-based pricing
Grow/shrink
Shared clusters for dev/test
On-Premises or EBS
Semi Persistent HBase/Kudu Dev/Test Cluster
Persistent HBase/Kudu Cluster
AWS EBS

- Best when development and testing faces:
  - Long wait periods for development and test environments
  - Constraints with on-premises capacity for dev and testing
  - Difficulty securing real data
- Benefits include:
  - Smaller on-premises data center footprint
  - Scale to exact needs of each dev/testing use case; no one-size-fits-all estimating of on-premises resources
  - Decreased delay in dev and testing, which translates into increased time-to-value of any use case
  - Production-level security means real data is used; real data in dev and testing results in easier production implementation later
- Tradeoffs:
  - In the absence of a development and testing bottleneck, new organizational processes may be needed to determine which dev and test projects receive funding

Pattern #2: Persistent Workloads on Block Storage

For on-demand deployment of operational clusters.



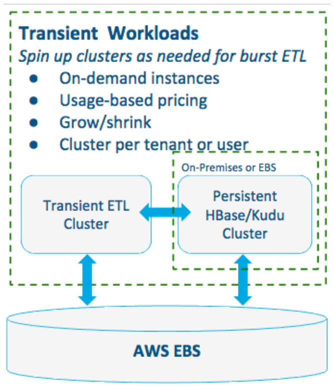
- Best when operational data ingest faces:
  - Unexpected surges in traffic

  - Regular batch jobs growing in size
  - Batch import of net-new data set(s)

- Benefits include:
  - Smaller on-premises or persistent cloud cluster footprint results in lower costs
- Tradeoffs:
  - Increased complexity relative to carrying excess capacity on premises or in persistent cloud clusters, including:
    - Scheduling of ETL process
    - Delays in data inherent in burst ETL vs. streaming ETL
    - Data movement between storage location/types

Pattern #3: Transient Clusters for Burst ETL

Leverage cloud elasticity to overcome ETL bottlenecks at low-cost.



- Best when operational data ingest faces:
  - Unexpected surges in traffic
  - Regular batch jobs growing in size
  - Batch import of net-new data set(s)
- Benefits include:
  - Smaller on-premises or persistent cloud cluster footprint results in lower costs
- Tradeoffs:
  - Increased complexity relative to carrying excess capacity on premises or in persistent cloud clusters, including:
    - Scheduling of ETL process
    - Delays in data inherent in burst ETL vs. streaming ETL
    - Data movement between storage location/types

Storage and Ingestion

This section provides guidelines for storage and data ingestion for operational databases in the cloud.

Storage

- EBS, as opposed to S3, will provide the most performant and cost-efficient storage.
- Apache Kudu is recommended for use cases requiring real-time analytics.
- Apache HBase is recommended for use cases requiring wide tables and unstructured data.

Stream Processing/Data Ingestion

- Spark Streaming, running on a dedicated permanent cluster, is recommended for real-time processing and serving architectures. It should live in the same availability zone as the permanent operational database cluster. Spark Streaming will play a critical role when using Kudu's relational structure.
- Apache Kafka can be used to land non-transformed data into the operational database via Apache Sqoop or Apache Flume.

Default Recommendations

This section provides default recommendations for three common types of operational database clusters.

Default recommendations for always-on clusters

Default recommendations for easy provisioning of always-on clusters. Snapshot backups are typically stored in S3.

Data Nodes

Model	vCPU	Mem (GiB)	Storage (GB)
d2.xlarge	4	30.5	3 x 2000 HDD
d2.2xlarge	8	61	6 x 2000 HDD
d2.4xlarge	16	122	12 x 2000 HDD
d2.8xlarge	36	244	24 x 2000 HDD

Smaller versions are recommended to stagger the impact of full block reports and garbage collection.

Master Nodes

Model	vCPU	Mem (GiB)	Storage (GB)
c3.8xlarge	32	60	2 x 320 SSD

The master node memory should be sized inline with the cluster size, c3.xlarge supports very large cluster sizes but smaller master nodes are possible.

Default recommendations for transient clusters with permanent storage using Altus Director

Default recommendations for transient clusters with permanent storage using Altus Director:

Data Nodes

Model	vCPU	Mem (GiB)	Storage
C4.large	4	30.5	EBS (4000 Mbps dedicated)

Master Nodes:

Model	vCPU	Mem (GiB)	Storage (GB)
c3.8xlarge	32	60	2 x 320 SSD

Storage, throughput workloads (ETL, etc.):

Volume Type	Volume Size	IOPS	Throughput
st1	500 GiB - 16 TiB	500	800 MiB/s

Storage, real time workloads (HBase, etc.):

Volume Type	Volume Size	IOPS	Throughput
io1	4 GiB - 16 TiB	20,000	800 MiB/s

Default recommendations for an always-on Spark streaming cluster

Spark clusters have homogenous nodes; there is no special master node. Default recommendations for an always-on Spark streaming cluster:

Default

Best balance of memory and compute.

Model	vCPU	Mem (GiB)	Storage
m4.2xlarge	8	32	EBS (1000 Mbps dedicated)

Memory-Intensive Workloads

Examples are workloads that cache RDDs/Dataframes or maintain in-memory state via the `updateStateByKey(…)` function.

Model	vCPU	Mem (GiB)	Storage
m3.2xlarge	8	61	160 GB SSD

Compute-Intensive Workloads

Examples are workloads that may perform compute intensive machine learning operations to score incoming events.

Model	vCPU	Mem (GiB)	Storage
c4.2xlarge	8	15	EBS (1000 Mbps dedicated)

Security on AWS: Best Practices

As discussed in Cloudera Enterprise in the Cloud on page 6, Cloudera clusters can be deployed to the cloud using any of the three leading cloud providers, including Amazon Web Services (AWS). Unlike on-premises clusters running securely under the complete control of your IT department, cloud-based clusters require you to manage greater risk.

For example, organizations must connect to the cloud provider and set up the instances to support the cluster, and then install and launch a Cloudera cluster. Whether the cluster runs for 5 minutes or 5 months, it must be configured so that it is available only to the appropriate people and processes, and that it is always available when they need it.

Furthermore, although cloud providers can handle many aspects of security for you, Cloudera adds enterprise security capabilities for clusters beyond what cloud providers can offer. Cloudera customers can be in complete control of their security—managing encryption keys outside the control of the cloud provider or enabling users to authenticate to the cloud through the organization's Active Directory (or other LDAP) server, for example. How can you get the best of both worlds—on premises and cloud—while ensuring security for your system and its data? This guide aims to help you do just that.

This guide focuses on security best practices for Cloudera clusters deployed to the Amazon Web Services (AWS) cloud.

Preliminary Planning

Organizations deploying clusters to any of the public clouds can achieve cost-savings, productivity, high availability, and many other benefits by carefully considering the best practices for different deployment patterns in the context of a specific use case.

In addition to meeting cost-savings and other goals, organizations must also ensure that cloud deployments meet all relevant privacy, integrity, and confidentiality requirements. For example, organizations in highly regulated industries may need to keep extensive audit trails and be able to track data lineage over time.

Identifying the security requirements and how to meet them in your cloud deployment starts by analyzing data inputs and outputs, the workload type, and the user profile:

- Identify the people and processes that need to use the cluster: Are they members of the same division in your organization?
- Identify your users and the specific levels of access to cluster resources and data needed so you can effectively shape your identity, authentication, and authorization requirements.
- Do you need to comply with industry or government regulations for privacy, confidentiality, or other security requirements?
- Do you need to be able to identify distinct users or processes that acted on any data as part of a complete audit trail, for example?
- For a given cluster or for a specific dataset on Amazon S3, should all users who have access be allowed to see all the data? If not, you must set up a multi-tenant cluster.
- Does your organization use an LDAP-based directory (for example, Microsoft Active Directory) for identity management, and if so, do you want to leverage that service when you deploy to the cloud? Or would your organization rather manage an additional set of credentials for all users, just for use in the cloud?
- Identify the locations, format, structure, of data sources.
- Identify encryption mechanisms, keys, and other specific details about how you encrypt data at rest now or plan to in the future.
- Test different sample workloads from your production system to determine the optimal deployment architecture.

These are just some of the questions to consider before deploying any cluster, with security in mind.

This guide highlights best practices for various architectural patterns identified by Cloudera that broadly distinguish between:

- Lifetime of the cluster (transient or persistent)

- Tenancy or usage profile (single-user or multi-tenant)

Other distinguishing characteristics of the architectural patterns are shown in the table below.

Lifetime	Tenancy	Key Components	Data Source/Target
Transient	Single-user	Apache Hive, Apache Spark, Hive on Spark, HDFS	Amazon S3
Persistent	Multi-tenant	Apache Impala, Apache Spark, Hive on Spark, HDFS	Amazon S3
Persistent	Multi-tenant	Apache HBase, Apache Spark	HDFS, Amazon S3

Regardless of type and architectural pattern, all cloud deployments to AWS must first consider network security.

Network Security

Deploying a cluster to the Amazon public cloud starts by configuring and securing the necessary network infrastructure hosted by the cloud provider. For clusters deployed to the Amazon Web Services cloud, this requires an Amazon Virtual Private Cloud (VPC).

Amazon automatically provisions a default VPC for each customer AWS account. The default VPC includes several related networking infrastructure entities, including a default subnet, default security group, default routing table, and so on. The defaults are fine for proof-of-concept deployments, but follow the best practices below for production systems.

Setting up secure networking from your premises to AWS is critical to the security of both your corporate network or data center and the cluster you deploy to the Amazon cloud. Cloudera recommends the following:

- Create and Configure a VPC on page 33
- Create Security Groups on page 34

Create and Configure a VPC

Use Amazon Identity and Access Management (IAM) to create separate user accounts for the various divisions in your organization that will deploy clusters to the cloud. Do not create all your cloud instances under your root Amazon account but instead create an IAM admin user and group.

- Create a VPC. The VPC will support the instances you want to deploy to the cloud, including an instance needed for Altus Director (if you plan to use that deployment tool), and for the specific EC2 instances that you will create to support the cluster or clusters, for specific workloads.
- Create public and private subnets to isolate traffic within the VPC. Plan out the IP addresses you will need for the security groups needed to secure the cluster.
- Add and configure a VPC Endpoint so the private subnet can connect to your Amazon S3 storage.
- Add a VPN (virtual private network) to the VPC to securely connect your on-premises data center to the Amazon cloud. The VPN lets your on-premises network communicate securely with the VPC. Amazon offers four types of VPN Connections, so pick the one that's best for your use case:
  - **AWS hardware VPN:** Supports IPsec VPN connections
  - **AWS Direct Connect**: Use this for a dedicated secure connection between your corporate network and Amazon AWS. This choice requires coordination between your organization's network infrastructure team and Amazon AWS, as well as hardware setup and configuration. See AWS Direct Connect for details.
  - **AWS VPN CloudHub:** Supports multiple remote networks. Use this if you have several remote branch offices, for example, that you want to connect to the VPC.
  - **Software VPN:** Runs on an EC2 instance using third-party software.

Create Security Groups

A security group is the Amazon VPC mechanism that acts as a whitelist for the VPC. The security group contains rules that you define for the port numbers and the protocols allowed for inbound and outbound network traffic. For example, the default VPC has a default security group that allows all outbound traffic but no inbound traffic. Cloudera recommends creating two different security groups in the VPC, as follows:

- Create one security group for the cluster's **edge node** (or nodes, for high availability configurations). Also known as the "**gateway node**," an edge node runs instances of specific gateway roles that let end-users and applications use cluster services.
- Give the edge node security group unlimited outbound access to the public internet.
- Limit inbound access to specific IP addresses from your corporate network or other approved IP addresses.
- Use IP addresses from the public subnet to make specific gateway roles accessible to users.
- Create a second security group for the other nodes of the cluster—the master nodes, worker nodes, and management nodes. The EC2 instances comprising the cluster must be able to communicate with each other through various ports, so these can be configured in the private subnet.
- Give this security group outbound access to the internet, for use with other AWS services and Amazon S3 storage and to access external repositories for software updates. For example, the EC2 instance that's used for Altus Director must be able to access the software repository to download and install the Altus Director software.
- Use private IP addresses for the nodes to communicate internally.
- Do not use a public IP address for Cloudera Manager.

For more information, see:

- Cloudera Enterprise Reference Architecture for AWS Deployments, specifically, the "Networking, Connectivity, and Security" section.
- Getting Started on Amazon Web Services (AWS) in the Altus Director documentation, specifically, Setting up the AWS Environment.
- Amazon Virtual Private Cloud (VPC) and Amazon EC2 (Network and Security) documentation.

Transient Single-User Clusters Using Amazon S3

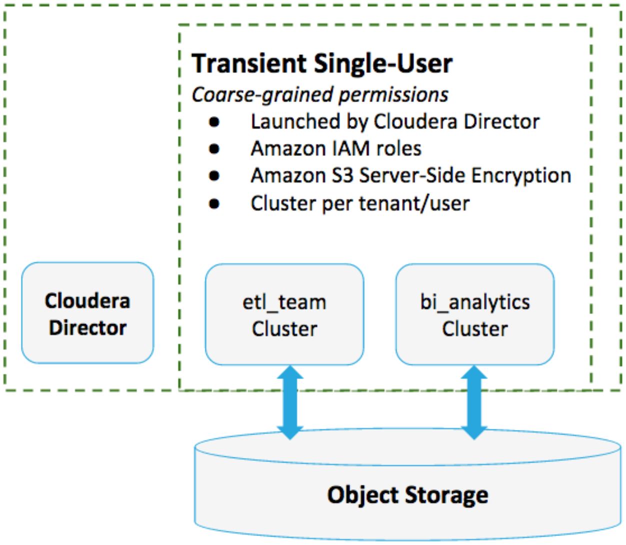
Architectural Pattern	Transient single-user clusters backed by Amazon S3
Cluster Services	Apache Hive, Apache Spark, Spark on Hive, and HDFS
Dependencies	Altus Director running on a persistent Amazon EC2 instance
Use Case	Data engineering, ETL for data warehousing, data pipelines

Transient single-user clusters are ideal for extract, transform, and load (ETL) data pipelines and other workloads that have relatively short durations. Cloud resources are shut down when the workload completes. The processes may be initiated and managed by a single user or a small group, or launched by means of a cron job.

In general, transient single-user clusters make sense for any workload that has a limited lifespan and small number of users with equivalent access privileges. These are also comparatively easy to secure—only coarse-grained authorization privileges are needed because the assumption is that anyone given access to the cluster is entitled to access all of the data associated with that cluster.

Example Use Case

Assume that data from several different applications and database systems, including extracts from an Oracle database, customer reports from Salesforce, and .csv files from legacy systems, is uploaded to Amazon S3 (Object Storage), to a `raw_input` bucket.



Twice a week, a member of the **etl_team** uses Altus Director to launch EC2 instances, spins up the cluster, runs the workload, and then shuts down the cluster when the workload completes. As the job runs on the cluster, results are written back to Amazon S3 to the `etl-results` bucket.

Given the small number of possible users—all of whom have the same permissions to the source data and to the resources needed to run the job—this workload can use Amazon Identity and Access Management for identity, authentication, and authorization within the cluster, and can use Altus Director to launch and manage the cluster when needed.

Identity, Authentication, and Authorization

For transient single-user clusters, use Amazon Identity and Access Management (IAM) to create an IAM role that can be used to launch the EC2 instances and run all aspects of the workload.

The Amazon IAM role takes care of both authentication and authorization for the cluster in the Amazon cloud, and access to the Amazon S3 storage bucket. When you set up an IAM role and use the profile to launch the cluster as described below, any user logging into the cluster can access all data in the Amazon S3 bucket specified in the policy, without the need to provide any other credentials.

The setup process is generally as follows:

1. Use your AWS Management Console to create an IAM role for EC2. When you do this, the console creates the role and an instance profile that will be available to use when you launch your EC2 instances.
2. Add a policy to the Amazon S3 bucket (Bucket Policy) that specifies what authenticated users (the IAM role is authenticated by AWS) can do.

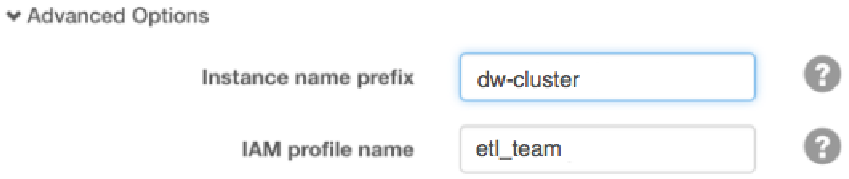
> **Important:** Any user logging in to the cluster can access ***all*** data in the Amazon S3 bucket specified in the policy, without the need to provide any other credentials. Be aware of this exposure and make sure this approach is appropriate for your use case.

For example, here's a policy that let's the IAM role list, read, write, and delete objects on the Amazon S3 `etl-results` bucket. Both *bucket-name* and *bucket-name/** are required in the Resource list, as shown in this example (`etl-results/*`, `etl-results`):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetBucketAcl",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3::: etl-results/*",
                "arn:aws:s3::: etl-results"
            ]
        }
    ]
}
```

3. Launch the EC2 instances using the instance profile created for the IAM role. For ease of deployment, use Altus Director to launch transient single-user clusters. Altus Director provides two different tools for deploying clusters: Altus Director UI or the Altus Director command-line:

    - Altus Director UI is a web-server-hosted console that can be accessed at `https://your.instance.hosting.director.ui:7187`. Enter the profile name in the Advanced Options section of an instance template, as shown here:

      Advanced Options

      Instance name prefix: dw-cluster

      IAM profile name: etl_team

    - Altus Director command-line lets you submit the details in a cluster configuration file to the Altus Director server. For example, here is the `iamProfileName` setting from the sample template for the Cloudera Enterprise Reference Architecture for AWS Deployments (obtain scripts from GitHub's Altus Director scripts section):

```
...
# Name of the IAM Role to use for this instance type
# iamProfileName: iam-profile-REPLACE-ME
iamProfileName: etl_team
...
```

Whether deployed using the Altus Director UI console or the command-line, the EC2 instance is launched using the instance profile (‘iamProfileName’) containing the IAM role, and the result is that your EC2 instances can all use the Amazon S3 bucket associated with the profile.

For more information:

- Provisioning a Cluster on AWS in the Altus Director documentation
- Identities (Users, Groups, and Roles) in Amazon AWS documentation

Encryption (Data in Transit)

Cloudera generally recommends configuring TLS/SSL to encrypt network communications. However, for transient clusters that may have no active management involved and for which the Cloudera Manager instance has been deployed merely to facilitate cluster installation and deployment, TLS/SSL may not be a strict requirement. For short-lived clusters dedicated to limited sets of processing tasks running under the control of a single user account or an IAM role, the setup required for TLS/SSL may be more costly in terms of time than the use case demands.

If you do have a strict requirement to encrypt communications within the cluster, however, you can script this yourself and use Altus Director to automate the process. See How-to: Deploy a Secure Enterprise Data Hub on AWS for more information.

Encryption (Data at Rest)

Encrypt data-at-rest for any production cluster. For clusters backed by Amazon S3, Cloudera supports:

- Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)—Keys are generated and managed by Amazon.
- Server-Side Encryption using Amazon Key Management Server (SSE-KMS)—Keys are created and managed using AWS Key Management Server. Supply the key name in the IAM profile (the IAM user launching the cluster must have permission to use the key).

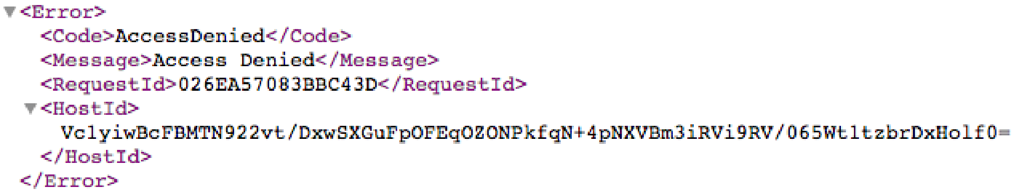
Cloudera clusters fully support both these options, so use the mechanism that makes the most sense for your specific use case.

For more information:

- How to Configure Encryption for Amazon S3 in Cloudera Security
- Writing Encrypted Data to Secure S3 Buckets From Altus Jobs in the Cloudera Altus documentation
- Protecting Data Using Server-Side Encryption in the Amazon AWS documentation

Auditing

Transient clusters that process data pipelines and ETL jobs typically do not require extensive audit trails. You can enable Amazon S3 logging (through the AWS Management Console) to track events that occur on the Amazon S3 storage bucket. However, AWS has limited auditing capabilities, so be aware of the limitations in the context of your workload. For example, according to Amazon (Server Access Logging documentation), although Amazon S3 server logs can provide “an idea of the nature of traffic against your bucket," they are not meant as “a complete accounting of all requests." This example of an Amazon S3 log message captures an unauthorized access request:



```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>026EA57083BBC43D</RequestId>
  ▼<HostId>
     VclyiwBcFBMTN922vt/DxwSXGuFpOFEqOZONPkfqN+4pNXVBm3iRVi9RV/065WtltzbrDxHolf0=
   </HostId>
 </Error>
```

Persistent Multi-Tenant Clusters Using Amazon S3

Architectural Pattern	Persistent multi-tenant clusters backed by Amazon S3
Cluster Services	Apache Impala, Apache Hive, Hive on Spark, and HDFS
Dependencies	Kerberos, Sentry, TLS/SSL,
Use Case	Data Warehouse

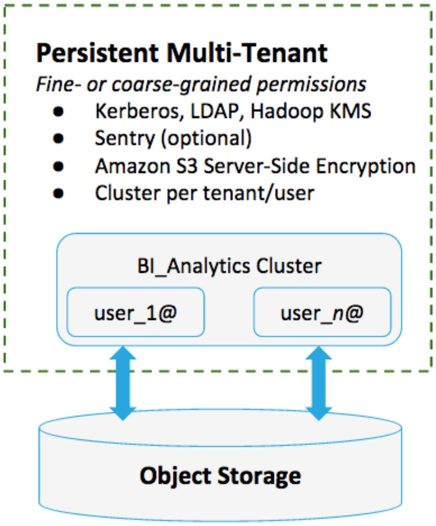
From a security perspective, persistent clusters make sense for use cases in which:

- part (but not all) of a given dataset must be shared but at a very granular level—not only files, but specific columns and rows within tables;
- the number of clusters needed to support the per-user (or per-user group with the same permissions) would be too costly and unmanageable;
- casual users do not want to learn about cloud infrastructure, or spin-up their own clusters, or wait for a cluster to start up; or
- users want to get access to cluster quickly, based on their identity in the organization’s enterprise directory.

The best practices included in this section are the same as Cloudera recommendations for conventional on-premises clusters with some exceptions, where support for the Amazon S3 storage is not yet fully implemented in a Hadoop ecosystem component or security mechanism.

Example Use Case

Results from many ETL pipelines converge in Impala tables hosted on Amazon S3 for use by various subdivisions and teams within the organization. The Impala tables comprise what amounts to a vast data store used to derive everything from financial reports and sales-team bonuses, to operational dashboards and performance metrics for management.


Persistent Multi-Tenant

Fine- or coarse-grained permissions

- Kerberos, LDAP, Hadoop KMS
- Sentry (optional)
- Amazon S3 Server-Side Encryption
- Cluster per tenant/user

Access permissions to the system vary by user, team, and division. Privileges can also vary by subset of data in the tables. See Using Impala with the Amazon S3 Filesystem for details.

Identity

Cloudera recommends using your organization's identity-management infrastructure so you can leverage your organization's existing users and groups instead of creating new ones. Organizations with existing LDAP directory services (Microsoft Active Directory, OpenLDAP) can integrate these with Amazon Web Services. This allows user's directory group membership to determine what data they are allowed to access within the cluster, and it also eliminates the need for separate passwords for end-users. See How-to: Deploy a Secure Enterprise Data Hub on AWS and How-to: Deploy a Secure Enterprise Data Hub on AWS (Part 2) for more information.

For small workgroups, the Linux user/group directory can be used instead of an external enterprise directory.

Authentication

To authenticate users:

- Enable Kerberos and use LDAP/Active Directory to support login credentials. Use your existing Active Directory (see How-to: Deploy a Secure Enterprise Data Hub on AWS for more information).
- Use the Altus Director client in combination with the appropriate cluster configuration file to create and deploy a Kerberized cluster. See How-to: Deploy a Secure Enterprise Data Hub on AWS (Part 2) for an example.
- Do not give non-admin users root access to any of the nodes and do not allow them to directly access or manage any AWS compute resources. Instead, grant non-admin users (through their LDAP credentials) login privileges to specific edge nodes and client services.

> **Important:** End-users, such as BI analysts, access the cluster through an edge node only. They should not have root access.

You can integrate your organization's Active Directory instance with the AWS cloud by hosting your own directory service (domain controller) that connects to your on-premises directory:

- Create an EC2 instance;
- Setup a Windows domain controller (domain controllers host the Active Directory service instance) on that EC2 instance;
- Configure the domain controller to obtain its LDAP objects from your on-premises directory tree.

End-users log in as they normally would, and they are authenticated through the DC in the instance.

Authorization

When all users are allowed to access all data that the cluster can access, using Sentry for authorization is not necessary. However, if different users have different privileges, Cloudera recommends using Sentry for authorization. Apache Sentry is a unified role-based access control (RBAC) service for Hadoop clusters. Sentry includes a service, a backend database, and plug-ins for various components in the Hadoop ecosystem, including Apache Hive, Hive Metastore/HCatalog, Apache Solr, Impala, and HDFS.

Various roles are defined and stored in Sentry's database as mappings between `user:group` identities and Hadoop Group Mappings for the various components.

For example, the data warehouse running on a persistent cluster might be used by dozens of employees with the role `bi_analyst` while the role `analyst_grade_v` may be limited to a handful of senior staff who have spend privileges above $20,000 US. Although these two employee classes have some target systems in common, the privileges on the target component vary widely between the two roles.

Sentry requires Kerberos. Only trusted users (those who have been authenticated by Kerberos) can access the Sentry service to manage the system, define roles, and map roles to permissions.

Currently, each cluster requires its own Sentry service because permissions are managed separately, per cluster. See Authorization with Apache Sentry in Cloudera Security for details.

Encryption (Data in Transit)

In some cases where sensitive data is being used or sent over the network, **all** pathways may need to be encrypted. In other cases, encryption is optional. In order of priority, the following network paths should be encrypted:

1. Encrypt pathways where credentials (such as Kerberos keytabs) flow. This includes Cloudera Manager agent hosts.
2. Encrypt data traffic leaving the cluster, such as to and from Amazon S3 buckets.
3. Encrypt all of the other pathways where data can flow within the cluster.

See How to Configure TLS Encryption for Cloudera Manager in Cloudera Security for details about configuring the Cloudera Manager server and agent host systems for encryption and browser connections to Cloudera Manager Admin Console for HTTPS.

For the specific components running on the cluster, such as Impala, Hive, or Hue, see component-specific details in Cloudera Security, starting with Configuring TLS/SSL Encryption for CDH Services.

In general, however, the process of configuring various services and role instances is much the same for each component. Typically, cluster administrators will use Cloudera Manager to enable TLS/SSL and specify the path and filename for the X.509 artifacts, such as JKS or PEM certificate store, truststore, private key, and password needed for the respective private keystore.

For example, this excerpt from a cluster's `core-site.xml` file shows the property that enables TLS/SSL for connections to Amazon S3 storage:

```
<property>
    <name>fs.s3a.connection.ssl.enabled</name>
    <value>true</value>
    <description>Enables or disables SSL connections to S3.</description>
</property>
```

Encryption (Data at Rest)

Always encrypt data at rest on production clusters. For persistent clusters deployed to the Amazon cloud, that may involve three different types of storage—Amazon S3, local storage (for the EC2 instance), and Amazon EBS (Elastic Block Storage) as auxiliary storage.

Amazon S3 Storage

Enable encryption for Amazon S3 storage using either SSE-KMS or SSE-S3.

Use Server-Side Encryption with AWS KMS–Managed Keys (SSE-KMS) SSE-KMS (supported as of Cloudera Manager/CDH 5.11) to retain complete control over the encryption key material, including generating your own key material using hardware-based solutions if you like. If you choose this approach, you must also set up AWS Key Management Service (AWS KMS).

If you prefer *not* to manage your own encryption keys, use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) (supported as of Cloudera Manager/CDH 5.10) and let AWS automatically assign, manage, and retrieve encryption keys (completely transparently) for your users.

See How to Configure Encryption for Amazon S3 in Cloudera Security for information about configuring either of these encryption mechanisms.

Local Storage (HDFS on the Amazon EC2 Instance)

Encrypt data within the cluster itself by using HDFS Transparent Encryption for the local storage associated with the EC2 instances that comprise the nodes of the cluster. The instance store is ephemeral. It exists for the duration of the EC2 instance, in other words, for the lifetime of the persistent or long-running cluster.

Auxiliary Storage (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) is persistent storage that can be added to an EC2 instance—an EBS volume can be added to a running cluster to support HDFS processing.

Use EBS Encryption for any EBS volumes you add to the cluster. Disk I/O, snapshots from the volume, and the data at rest on the volume itself are all encrypted using the key you configure (with the AWS Key Management Service). See Using EBS Volumes for Cloudera Manager and CDH for more information.

See Protecting Data at Rest in Cloudera Security for an overview of all mechanisms available to Cloudera clusters and HDFS, including encryption for data spills.

Auditing

For persistent clusters backed by Amazon S3:

- Enable Amazon S3 logging to record S3 events emitted by AWS. By default, this feature is disabled but can be enabled through the AWS Management Console for the Amazon S3 bucket. See AWS documentation Server Access Logging for details. According to Amazon, server logs can provide "an idea of the nature of traffic against your bucket" but are not meant as "a complete accounting of all requests."
- Use Cloudera Navigator to audit the local storage for the EC2 instances and any auxiliary storage on EBS volumes. For Amazon S3 storage, Cloudera Navigator currently collects technical metadata only, such as source type (S3), type , bucket, path, region, owner, S3 encryption, and other object-related attributes. See Cloudera Navigator Auditing (in Cloudera Data Management documentation) for information about Cloudera Navigator functionality for HDFS backed clusters.

See Auditing Mechanisms for Cloudera Clusters for an introduction to Cloudera Navigator.

See Cloudera Data Management for configuration and usage details.

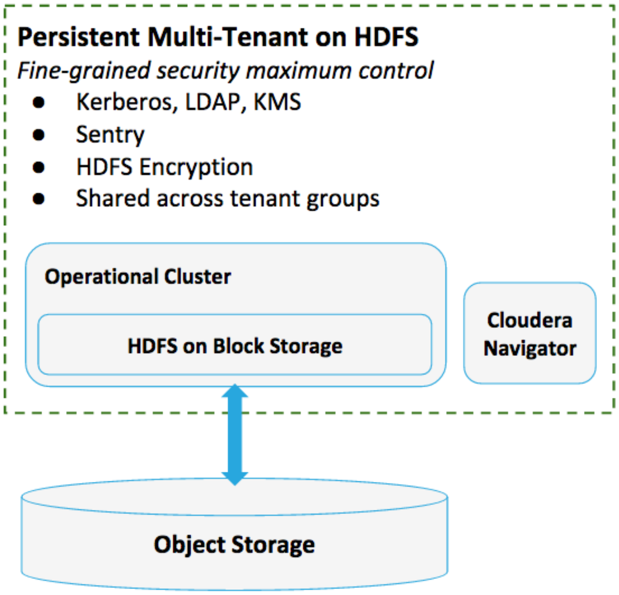
Persistent Multi-Tenant Clusters Using HDFS

Architectural Pattern	Persistent multi-tenant clusters backed by HDFS

Cluster Services	HBase, Apache Hive, Hive on Spark, and HDFS
Dependencies	Kerberos, Sentry, TLS/SSL
Use Case	Operational Database

Persistent multi-tenant clusters backed by HDFS are recommended for operational databases and other use cases that demand exceptionally fast performance, inherent transactional support (that is, without the need to augment Amazon S3's eventual consistency model), or when you want to use client-side encryption available with HDFS transparent encryption and have complete control over all keys.

Example Use Case



For any persistent HDFS-based cluster running in the cloud, Cloudera recommends the same security best practices that apply to on-premise clusters:

Identity	Use your LDAP-compliant directory service (Microsoft Active Directory, OpenLDAP) as discussed in Persistent Multi-Tenant Clusters on Amazon S3 (the approach is much the same). See How-to: Deploy a Secure Enterprise Data Hub on AWS for more details.
Authentication	Use Kerberos (MIT Kerberos, Microsoft Active Directory) for authentication.
Authorization	Use Sentry and RecordService for role-based access control and fine-grained permissions to row- and column-data, respectively.
Encryption (data in transit)	Configure TLS/SSL for all the cluster and for all services and roles. See How to Configure TLS Encryption for Cloudera Manager for details.
Encryption (data at rest)	Use HDFS Transparent Encryption for data-at-rest encryption or Cloudera Navigator Encrypt. In either case, use Cloudera Navigator Key Trustee Server (KTS) to manage your encryption keys. To generate your own key material using a hardware security module (HSM), integrate your HSM with Navigator KTS by adding Cloudera Navigator Key HSM to your setup.
Auditing	Use Cloudera Navigator to log all events that affect data, to meet auditing, data lineage, and governance requirements.

Identity

Organizations that use Microsoft Server for identity, directory, or authentication services can leverage these services when they deploy clusters to AWS. See How-to: Deploy a Secure Enterprise Data Hub on AWS for details.

Authentication

Cloudera recommends the following approach to integrating an on-premises Active Directory server with clusters deployed to the AWS cloud for reduced latency and better security:

- Create an EC2 instance and set up Windows Server as a domain controller on that instance.
  - Create the instance in the same VPC you plan to use for both Altus Director and for the cluster.
- Configure the domain controller so that it obtains its directory service database from your on-premises directory service instance (the Active Directory on-premises forest).

With this setup, when your users log in to the cluster, they obtain their credentials and are authenticated.

For faster deployment, Altus Director client (bootstrap command) or Altus Director server (bootstrap-remote) with the appropriate version of the configuration file (aws.kerberos.sample.conf, modified for your specifics) to create and deploy a cluster pre-configured to integrate with your site's Kerberos key server.

The configuration file assumes that the Kerberos KDC (key distribution center) is already set up and running, and that it is reachable by the Altus Director instance.

> ⚠ **Warning:** Keep Altus Director scripts and configuration files absolutely secure once you add your Kerberos and other highly sensitive—and highly privileged—credentials to the files.

For a complete step-through of using Altus Director to deploy secure clusters—clusters that use TLS/SSL for encrypted communications among all clients, that use Kerberos for authentication, and that leverage existing Active Directory services—see these two Cloudera Engineering Blog posts:

- How-to: Deploy a Secure Enterprise Data Hub on AWS: This post shows you how to integrate any cluster, on-premises or in the AWS cloud, with an existing Active Directory instance. The post also discusses leveraging Centrify with Active Directory for Linux user ID authentication, and how to use Microsoft Active Directory Certificate Services to handle X.509 certificates needed by TLS/SSL.
- How-to: Deploy a Secure Enterprise Data Hub on AWS (Part 2): This post builds on the first, showing you how to use Altus Director to deploy the cluster using configuration files. The post includes a step-through of a custom bootstrap script and how it installs and configures DNS, dependency packages, Java (JDK and cryptography extensions), and other subsystems needed to join the hosts to the Active Directory domain controller, obtain and prepares certificates from ADCS, and completes all other tasks.

See also:

- Types of Kerberos Deployments in the Cloudera Security
- Creating Kerberized Clusters With Altus Director

Authorization

Use Apache Sentry for role-based access to cluster components (see Authorization with Apache Sentry for more background).

Encryption (Data in Transit)

Always configure the cluster and all services to use TLS/SSL whether deployed on-premises or on clusters backed by Amazon S3. The blog post How-to: Deploy a Secure Enterprise Data Hub on AWS includes details, including how to obtain certificates using an offline multi-tier PKI setup (leveraging features of Windows Server). See also How to Configure TLS Encryption for Cloudera Manager.

Encryption (Data at Rest)

Encrypt data at rest for most use cases, especially when data contains patient medical records, customer financials, or any other type of sensitive personally-identifiable information (PII), trade secrets, classified information, and the like.

Cloudera supports using Amazon EBS for HDFS and using EBS encryption for data stored on EBS volumes.

For high availability, set up two Navigator KTS instances. A second on-premises cluster can share the same on-premises KTS as the cloud cluster. See Cloudera Navigator Key Trustee Server High Availability for details.

- Set up Cloudera Navigator Key Trustee Server on-premises.
- Set up the Key Management Service (KMS) to point to the on-premises KTS (using Cloudera Manager). (To mitigate latency between the on-premises cluster and the cloud, Hadoop KMS maintains a local cache of encryption keys).

Auditing

Use Cloudera Navigator to audit service access events. Audit events capture activities occurring at the cluster, host, role, service, or user level, recording the events in a central repository for viewing, filtering, and exporting through the Navigator UI. Event details include dozens of available attributes, such as user name, IP address of the host on which the event occurred, service name, session ID, object type, delegation token ID, and commands invoked (HBase, HDFS, Hive, Hue, Impala, Sentry, Solr, among others).

On-going Security Best Practices

After you deploy production workloads to the cloud, the real test of security begins. Here are some other general recommendations:

- Before running a production workload in the cloud, step-through the perimeter check detailed in How to secure 'Internet exposed' Apache Hadoop.
- Regularly visit the Cloudera Security Bulletins page to find out about known security issues that may arise.
  - Customers with support contracts are notified directly by Cloudera whenever a security issue arises.
  - Customers without support contracts should pro-actively monitor Cloudera Security Bulletins on a regular basis.
- Get active in the Cloudera Community to learn more about clusters and the cloud.
- Read the Cloudera Engineering Blog and get hands-on guidance from Cloudera engineers for a wide range of cluster deployment, management, and security tasks, both on-premises and cloud.

Summary of Security Best Practices

This page summarizes the recommendations made in Security on AWS: Best Practices:

Basics

Your organization's Amazon Web Services account is the root account. Amazon recommends creating a user in IAM (AWS Identity and Access Management) and then creating the VPC under that user's identity for your clusters, rather than using the root account. Here are some other tips:

- Evaluate your cluster's security regularly. Visit the Cloudera Security Bulletins page and the Cloudera Community to find out about known security issues.
- Change any and all default passwords throughout the system after it has been set up and before putting into production.

Networking (Setting Up the VPC)

- Set up a Virtual Private Cloud (VPC) on Amazon into which to create an Amazon EC2 instance and deploy your Cloudera cluster.
- Create a VPN (virtual private network) from your on-premises network to your VPC. Amazon offers several choices, so use the one that makes most sense for the type of deployment.
  - Use AWS Direct Connect for a permanent connection between your organization's network and your Amazon VPC. Direct Connect requires coordination between your organization and Amazon, but is most secure and

extends your corporate network into the Amazon cloud. If you want to federate your on-premises Active Directory instance with authentication in the Amazon cloud, use AWS Direct Connect as your VPN.

- Create at least two subnets in the VPC, one private, one public.
- Create two different security groups in the VPC:
  - Create one security group for the edge node (or nodes) that will support the gateway roles of the cluster, for example, YARN gateway, Flume gateway, and so on. Use public IP addresses (create a public subnet) for the edge node to provide external access to the cluster.
  - Create another security group for the other nodes in the cluster. Use private IP addresses and configure the security group to allow outbound access for all protocols.
- Use a private IP address (an address from the range of your VPC's private subnet) for Cloudera Manager Server.
- Create a persistent EC2 instance for Altus Director server. Altus Director is used to instantiate and control other EC2 instances in this VPC.
- Assign public IP addresses to the gateway node in the cluster (the node configured with various gateway roles, which provide client access to cluster functionality). For example, the gateway node includes HDFS, Hive, Spark 2 on YARN, and YARN gateway role instances.

Transient Cluster, Users with Same Privileges

These recommendations apply mostly to transient clusters and also to any cluster that has no need for granular user permissions or auditing. Assuming all users can have equivalent privileges on the cluster:

Identity, Authentication, and Authorization

When all users can have equivalent privileges to launch a cluster and run a particular workload, especially for transient single-user clusters, using Amazon Identity and Access Management (IAM) takes care of identity, authentication, and authorization from a single service:

- Using the Amazon AWS Console, create an IAM role and configure that role with the permissions needed to create the necessary EC2 instances in the VPC.
- Give this IAM role access to the necessary Amazon S3 bucket.
- In the VPC configuration, give this IAM role access to the edge security group so that it can launch the EC2 instances when needed.
- Give each user group its own Altus Director instance and grant permissions to the IAM role to use Director and create clusters.

Encryption (Data at Rest)

The data at rest encryption mechanisms depend on the type of storage, as follows:

Amazon S3 Data

- For Amazon Simple Storage Service (S3), use Amazon's server-side encryption using S3-managed encryption keys (SSE-S3) or using SSE-KMS. For SSE-KMS, you must also use Amazon's AWS Key Management Service.

Amazon EBS Data

- Use Amazon's AWS Key Management Service to create encryption keys and specify that want an encrypted EBS volume (using that key) when you create the volume. See Amazon's Amazon EBS Encryption and How Amazon Elastic Block Store (Amazon EBS) Uses AWS KMS for more information.

HDFS Storage

- Use native HDFS Transparent Encryption (aka, HDFS Data At Rest Encryption) for client-side transparent end-to-end data encryption for HDFS data.

Persistent Clusters, Users with Different Privileges

These recommendations assume that your organization has Microsoft Active Directory (or other LDAP-compliant directory service) and that you want to leverage the user identity as well as the authentication services available (Kerberos, Active Directory).

Identity

- Use your organization's Microsoft Active Directory or other LDAP-compliant directory service as the basis for identity and authentication.
- Integrate the local hosts' UIDs to the LDAP identity with Linux SSSD (System Security Services daemon), Centrify, or FreeIPA (Identity, Policy, and Audit).

Authentication

- Use Kerberos (Microsoft Active Directory, MIT Kerberos) for internal authentication among the nodes and services in the cluster, and for end-user authentication.

Authorization

- Use Sentry to set role-based access controls on Apache Hive, Apache Solr, and Apache Impala data, and to Hive table data stored in HDFS. Currently, Sentry controls are applied on a per-cluster basis.

Encryption

Encryption needs to be applied to both data in transit and data at rest. Data at rest encryption protects data stored within the cluster.

Encryption (Data in Transit)

Data in transit encrypts network communications between nodes in the cluster and between all the endpoints that need to connect to the cluster. Cloudera clusters use industry standard TLS/SSL to encrypt network communications between RPC client and server processes and between Web components (HTTP/S for browser-based access to Cloudera Manager Admin Console, for example).

- Obtain and install certificates (X.509) for all hosts (EC2 instances) that comprise the cluster, for encrypted communications among the nodes in the cluster

Encryption (Data at Rest)

Encryption mechanisms vary, depending on where the data is stored.

- To manage your own encryption keys, outside Amazon Web Services, use a persistent cluster with local storage (the EC2 instance storage) and manage the cluster using Cloudera Manager rather than Altus Director.

Amazon S3 Data

- Use Amazon server-side encryption (SSE-S3).

HDFS Storage

- Use Transparent Encryption.

Auditing

- Use Cloudera Navigator to track the logs on your Amazon S3 buckets.

Get Started with Amazon S3

These topics focused on Amazon S3 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 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
- 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

- Optimizing Hive Write Performance on Amazon S3

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

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.

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.
```