Cloudera Data Hub

Date published: 2019-12-17
Date modified: 2023-06-27

CLOUDERA

Legal Notice

Contents

Cluster deployment options

You have two basic options when creating a cluster: (1) create a cluster from one of the default or custom cluster definition, or (2) create a custom cluster.

Cluster definitions

Cloudera Data Hub includes a set of prescriptive cluster definitions that can be used to quickly provision clusters for common use cases. These default cluster definitions include prescriptive options related to compute instance types and storage options.

We recommend that you start by reviewing these definitions. If you conclude that they do not match your requirements, you can customize them and then save them as custom cluster definitions.

Custom deployment

Use the custom cluster deployment option if you would like to select specific Cloudera Runtime components for your cluster.

> **Note:**
>
> Currently, this feature is limited and only allows you to select a cluster template that determines which components will be used for your cluster.

Advanced options

The cluster wizard includes a set of advanced options that can be optionally configured. Some of these options require additional configuration prior to cluster creation, so if you would like to use these options, read the cluster planning documentation first.

Default cluster configurations

Cloudera Data Hub includes a set of prescriptive cluster configurations. Each of these default cluster configurations include a cloud-provider specific cluster definition, which primarily defines cloud provider settings. The cluster definition references a cluster template, which defines a number of Cloudera Runtime or Cloudera DataFlow components used for common data analytics and data engineering use cases.

Refer to the topic for each default cluster configuration to view the included services and compatible Runtime versions. These topics include links to documentation that will help you to understand the included components and use the workload cluster.

Many of the cluster components are included in the Cloudera Runtime software distribution. The Streams Messaging, Cloudera Flow Management, and Cloudera Streaming Analytics cluster configurations are part of Cloudera DataFlow for Data Hub and have distinct planning considerations and how-to information. See the Cloudera DataFlow for Data Hub documentation for more details.

You can access the default cluster definitions by clicking Environments, then selecting an environment and clicking the Cluster Definitions tab.

You can access the default cluster templates from Shared ResourcesCluster Templates.

To view details of a cluster definition or cluster template, click on its name. For each cluster definition, you can access a raw JSON file. For each cluster template, you can access a graphical representation ("list view") and a raw JSON file ("raw view") of all cluster host groups and their components.

Related Information
Cloudera DataFlow for Data Hub

Cloudera Runtime

Data Engineering clusters

Learn about the default Data Engineering clusters, including cluster definition and template names, included services, and compatible Cloudera Runtime version.

Data Engineering provides a complete data processing solution, powered by Apache Spark and Apache Hive. Spark and Hive enable fast, scalable, fault-tolerant data engineering and analytics over petabytes of data.

Data Engineering cluster definition

This Data Engineering template includes a standalone deployment of Spark and Hive, as well as Apache Oozie for job scheduling and orchestration, Apache Livy for remote job submission, and Hue and Apache Zeppelin for job authoring and interactive analysis.

Cluster definition names

- Data Engineering for AWS
- Data Engineering HA - Spark3 for AWS

  See the architectural information below for the Data Engineering HA clusters
- Data Engineering Spark3 for AWS

Cluster template name

- Data Engineering: Apache Spark3, Apache Hive, Apache Oozie

  Note: This cluster template was formerly named "Data Engineering: Apache Spark, Apache Hive, Apache Oozie."

- Data Engineering: HA: Apache Spark3, Apache Hive, Apache Oozie

  Note: This cluster template was formerly named "Data Engineering: HA: Apache Spark, Apache Hive, Apache Oozie."

  See the architectural information below for the Data Engineering HA clusters
- Data Engineering: Apache Spark3, Apache Hive, Apache Oozie

  Note: The "Data Engineering: Apache Spark3" cluster template is deleted. Therefore, the "Data Engineering: Apache Spark3, Apache Hive, Apache Oozie" cluster template can be used instead.

Included services

- HDFS
- Hive
- Hue
- Livy
- Spark 3
- Yarn
- Zeppelin
- ZooKeeper
- Oozie is supported for Spark 3 as of Cloudera Runtime version 7.2.18
- Hive Warehouse Connector is supported as of Cloudera Runtime version 7.2.16.

Compatible Cloudera Runtime version

  7.2.16, 7.2.17, 7.2.18

Topology of the Data Engineering cluster

Topology is a set of host groups that are defined in the cluster template and cluster definition used by Data Engineering. Data Engineering uses the following topology:

Host group	Description	Node configuration
Master	The master host group runs the components for managing the cluster resources including Cloudera Manager, Name Node, Resource Manager, as well as other master components such HiveServer2, HMS, Hue etc.	1 For Cloudera Runtime versions earlier than 7.2.14: AWS : m5.4xlarge; gp2 - 100 GB For Cloudera Runtime versions 7.2.14+ DE, DE Spark3, and DE HA: AWS : m5.4xlarge; gp2 - 100 GB
Worker	The worker host group runs the components that are used for executing processing tasks (such as NodeManager) and handling storing data in HDFS such as DataNode.	3 For Cloudera Runtime versions earlier than 7.2.14: AWS : m5.2xlarge; gp2 - 100 GB For Cloudera Runtime versions 7.2.14+ DE and DE Spark3: AWS: r5d.2xlarge - (gp2/EBS volumes) DE HA: AWS: r5d.4xlarge - (gp2/EBS volumes)
Compute	The compute host group can optionally be used for running data processing tasks (such as NodeManager). By default the number of compute nodes is set to 1 for proper configurations of YARN containers. This node group can be scaled down to 0 when there are no compute needs. Additionally, if load-based auto-scaling is enabled with minimum count set to 0, the compute nodegroup will be resized to 0 automatically.	0+ For Cloudera Runtime versions earlier than 7.2.14: AWS : m5.2xlarge; gp2 - 100 GB For Cloudera Runtime versions 7.2.14+ DE and DE Spark3: AWS: r5d.2xlarge - (ephemeral volumes) DE HA: AWS: r5d.4xlarge - (ephemeral volumes)

Host group	Description	Node configuration
		Note: Compute nodes run YARN and require storage only for temporary data - this requirement is fulfilled by instance storage, so making the attached volumes count to 0 by default is more cost-efficient.
Gateway	The gateway host group can optionally be used for connecting to the cluster endpoints like Oozie, Beeline etc. This nodegroup does not run any critical services. This nodegroup resides in the same subnet as the rest of the nodegroups. If additional software binaries are required they could be installed using recipes.	0+ AWS : m5.2xlarge; gp2 - 100 GB

Service configurations			
Master host group Cloudera Manager, HDFS, Hive (on Tez), HMS, Yarn RM, Oozie, Hue, DAS, Zookeeper, Livy, Zeppelin and Sqoop	Gateway host group Configurations for the services on the master node	Worker host group Data Node and YARN NodeManager	Compute group YARN NodeManager

Configurations

Note the following:

- There is a Hive Metastore Service (HMS) running in the cluster that talks to the same database instance as the Data Lake in the environment.
- If you use CLI to create the cluster, you can optionally pass an argument to create an external database for the cluster use such as Cloudera Manager, Oozie, Hue, and DAS. This database is by default embedded in the master node external volume. If you specify the external database to be of type HA or NON_HA, the database will be provisioned in the cloud provider. For all these types of databases the lifecycle is still associated with the cluster, so upon deletion of the cluster, the database will also be deleted.
- The HDFS in this cluster is for storing the intermediary processing data. For resiliency, store the data in the cloud object stores.
- For high availability requirements choose the Data Engineering High Availability cluster shape.

Architecture of the Data Engineering HA for AWS cluster

The Data Engineering HA for AWS and Azure cluster shape provides failure resilience for several of the Data Engineering HA services, including Knox, Oozie, HDFS, HS2, Hue, Livy, YARN, and HMS.

Services that do not yet run in HA mode include Cloudera Manager, DAS, and Zeppelin.
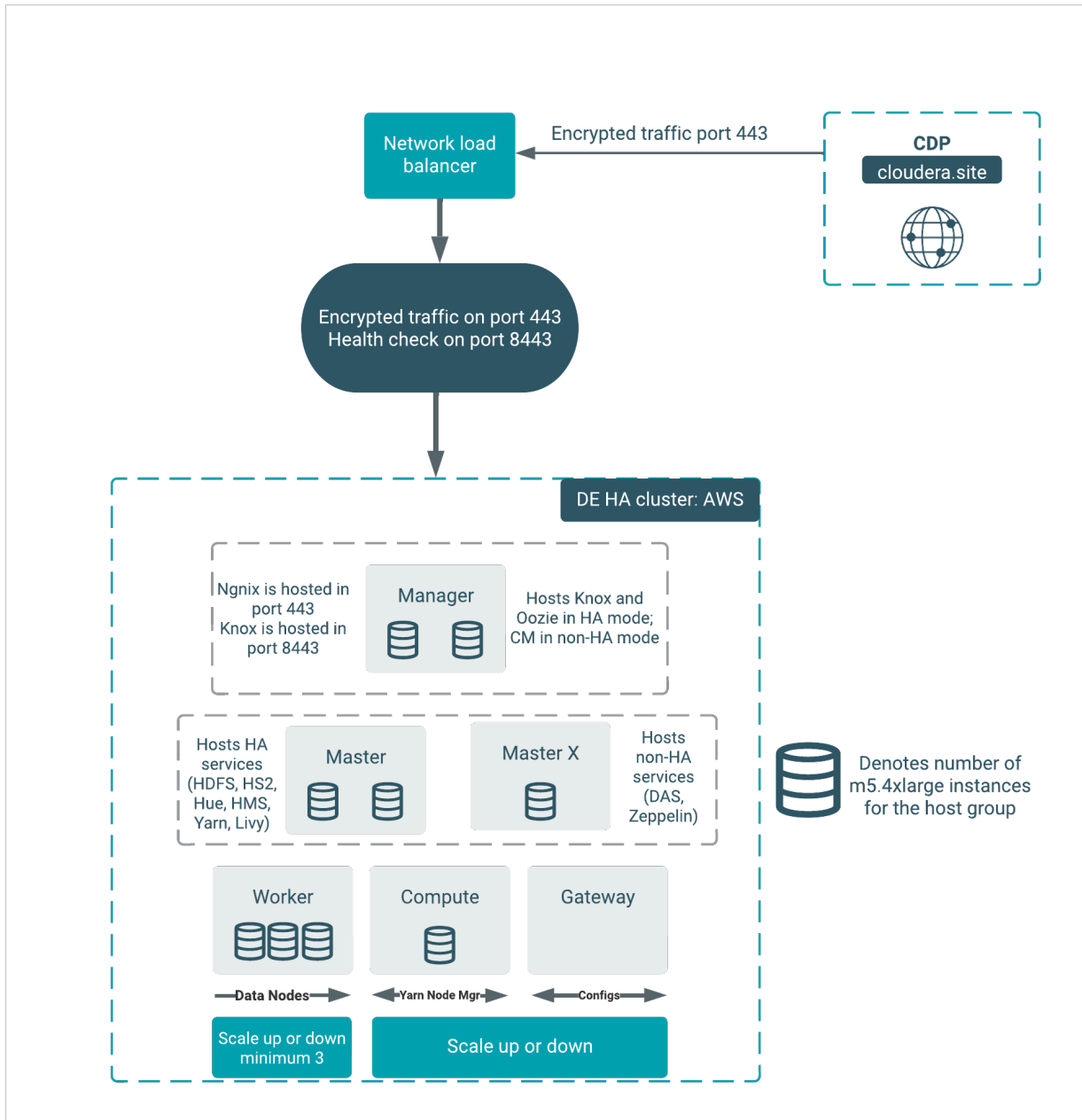
The architecture outlined in the diagram above handles the failure of one node in all of the host groups except for the "masterx" group. See the table below for additional details about the component interactions in failure mode:

Component	Failure	User experience
Knox	One of the Knox services is down	External users will still be able to access all of the UIs, APIs, and JDBC.
Cloudera Manager	The first node in manager host group is down	The cluster operations (such as repair, scaling, and upgrade) will not work.
Cloudera Manager	The second node in the manager host group is down	No impact.
HMS	One of the HMS services is down	No impact.
Hue	One of the Hue services is down in master host group	No impact.

HS2	One of the HS2 services is down in the master host group	External users will still be able to access the Hive service via JDBC. But if Hue was accessing that particular service it will not failover to the other host. The quick fix for Hue is to restart Hue to be able to use Hive functionality.
YARN	One of the YARN services is down	No impact.
HDFS	One of the HDFS services is down	No impact.
Nginx	Nginx in one of the manager hosts is down	Fifty percent of the UI, API, and JDBC calls will be affected. If the entire manager node is down, there is no impact. This is caused by the process of forwarding and health checking that is done by the network load-balancer.
Oozie	One of the Oozie servers is down in the manager host group.	No impact for AWS and Azure as of Cloudera Runtime version 7.2.11. If you create a custom template for DE HA, follow these two rules: 1. Oozie must be in single hostgroup. 2. Oozie and Hue must not be in the same hostgroup.

Important: If you are creating a DE HA cluster through the CDP CLI using the create-aws-cluster command, note that there is a CLI parameter to provision the network load-balancer in HA cluster shapes. Make sure to use the [--enable-load-balancer | --no-enable-load-balancer] parameter when provisioning a DE HA cluster via the CLI. For more information see the CDP CLI reference.

Custom templates

Any custom DE HA template that you create must be forked from the default templates of the corresponding version. You must create a custom cluster definition for this with the JSON parameter "enableLoadBalancers": true , using the create-aws/azure/gcp-cluster CLI command parameter --request-template. Support for pre-existing custom cluster definitions will be added in a future release. As with the template, the custom cluster definition must be forked from the default cluster definition. You are allowed to modify the instance types and disks in the custom cluster definition. You must not change the placement of the services like Cloudera Manager, Oozie, and Hue. Currently the custom template is fully supported only via CLI.

The simplest way to change the DE HA definition is to create a custom cluster definition. In the Create Data Hub UI when you click Advanced Options, the default definition is not used fully, which will cause issues in the HA setup.

Related Information
HDFS
Hive
Hue
Oozie
Spark
YARN
Zookeeper

Data Mart clusters

Learn about the default Data Mart and Real Time Data Mart clusters, including cluster definition and template names, included services, and compatible Runtime version.

Data Mart is an MPP SQL database powered by Apache Impala designed to support custom Data Mart applications at big data scale. Impala easily scales to petabytes of data, processes tables with trillions of rows, and allows users to store, browse, query, and explore their data in an interactive way.

Data Mart clusters

The Data Mart template provides a ready to use, fully capable, standalone deployment of Impala. Upon deployment, it can be used as a standalone Data Mart to which users point their BI dashboards using JDBC/ODBC end points. Users can also choose to author SQL queries in Cloudera's web-based SQL query editor, Hue, and run them with Impala providing a delightful end-user focused and interactive SQL/BI experience.

Cluster definition names

- Data Mart for AWS

Cluster template name

Cloudera - Data Mart: Apache Impala, Hue

Included services

- HDFS
- Hue
- Impala

Compatible Cloudera Runtime versions

7.1.0, 7.2.0, 7.2.1, 7.2.2, 7.2.6, 7.2.7, 7.2.8, 7.2.9, 7.2.10, 7.2.11, 7.2.12, 7.2.14, 7.2.15, 7.2.16, 7.2.17, 7.2.18

Real Time Data Mart clusters

The Real-Time Data Mart template provides a ready-to-use, fully capable, standalone deployment of Impala and Kudu. You can use a Real Time Data Mart cluster as a standalone Data Mart which allows high throughput streaming ingest, supporting updates and deletes as well as inserts. You can immediately query data through BI dashboards using JDBC/ODBC end points. You can choose to author SQL queries in Cloudera's web-based SQL query editor, Hue. Executing queries with Impala, you will enjoy an end-user focused and interactive SQL/BI experience. This template is commonly used for Operational Reporting, Time Series, and other real time analytics use cases.

Cluster definition names

- Real-time Data Mart for AWS

Cluster template name

Cloudera - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark

Included services

- HDFS
- Hue
- Impala
- Kudu
- Spark 2
- Yarn

Compatible Cloudera Runtime versions

7.1.0, 7.2.0, 7.2.1, 7.2.2, 7.2.6, 7.2.7, 7.2.8, 7.2.9, 7.2.10, 7.2.11, 7.2.12, 7.2.14, 7.2.15, 7.2.16, 7.2.17

Cluster definition names

- Real-time Data Mart - Spark3 for AWS

Cluster template name

Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark3

Included services

- HDFS

- Hue
- Impala
- Kudu
- Spark 3
- Yarn

Compatible Cloudera Runtime versions

7.2.16, 7.2.17, 7.2.18

High availability

Cloudera recommends that you use high availability (HA), and track any services that are not capable of restarting or performing failover in some way.

Impala HA

The Impala nodes offer high availability. The following Impala services are not HA.

- Catalog service
- Statestore service

Kudu HA

Both Kudu Masters and TabletServers offer high availability.

Related Information
HDFS
Hue
Impala
Kudu
Spark
YARN

Operational Database clusters

The Operational Database (OpDB) template is removed from the Cloudera Data Hub. You can access the Cloudera Operational Database instead as a superior product.

The Cloudera Operational Database is a NoSQL database powered by Apache HBase designed to support custom OLTP applications that want to leverage the power of BigData. Apache HBase is a NoSQL, scale-out database that can easily scale to petabytes and stores tables with millions of columns and billions of rows.

Cloudera Operational Database also contains Apache Phoenix which provides a way to use HBase through an SQL interface.

Cloudera recommends you to use the Cloudera Operational Database to create Operational Database clusters.

Related Information
Cloudera Operational Database
Getting started with Operational Database
Before you create an Operational Database cluster
Creating an Operational Database cluster
HDFS
HBase
Knox
Zookeeper
Phoenix

Streams Messaging clusters

Learn about the default Streams Messaging clusters, including cluster definition and template names, included services, and compatible Cloudera Runtime version.

Streams Messaging provides the following features:

- Advanced messaging and real-time processing on streaming data using Apache Kafka
- Centralized schema management using Schema Registry
- Management and monitoring capabilities powered by Streams Messaging Manager
- Cross-cluster Kafka topic replication using Streams Replication Manger
- Kafka partition rebalancing with Cruise Control

This template sets up a fault-tolerant standalone deployment of Apache Kafka and supporting Cloudera components (Schema Registry, Streams Messaging Manager, Streams Replication Manager and Cruise Control), which can be used for production Kafka workloads in the cloud or as a disaster recovery instance for on-premises. Kafka clusters.

> **Note:**
>
> Streams Messaging clusters have distinct planning considerations and how-to information. See the Cloudera DataFlow for Data Hub documentation for information about:
>
> - Planning your Streams Messaging cluster deployment
> - Creating your first Streams Messaging cluster
> - Connecting Kafka clients to Cloudera Public Cloud clusters

Cluster definition names

- Streams Messaging Heavy Duty for AWS
- Streams Messaging Light Duty for AWS
- Streams Messaging HA for AWS

Cluster template name

- Cloudera - Streams Messaging Heavy Duty
- Cloudera - Streams Messaging Light Duty
- Cloudera - Streams Messaging High Availability

Included services

- Kafka
- Schema Registry
- Streams Messaging Manager
- Streams Replication Manager
- Cruise Control
- Kafka Connect

Compatible Cloudera Runtime version

- 7.1.0 (Preview)
- 7.2.0
- 7.2.1
- 7.2.2
- 7.2.6
- 7.2.7
- 7.2.8
- 7.2.9
- 7.2.10
- 7.2.11

- 7.2.12
- 7.2.14
- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18

Related Information

Setting up your Streams Messaging cluster

Ingesting Data into Cloudera Public Cloud

Kafka

Schema Registry

Streams Messaging Manager

Streams Replication Manager

Flow Management clusters

Learn about the default Flow Management clusters, including cluster definition and template names, included services, and compatible Cloudera Runtime versions.

Flow Management delivers high-scale data ingestion, transformation, and management to enterprises from any-to-any environment. It addresses key enterprise use cases such as data movement, continuous data ingestion, log data ingestion, and acquisition of all types of streaming data including social, mobile, clickstream, and IoT data.

The Flow Management template includes a no-code data ingestion and management solution powered by Apache NiFi. With NiFi's intuitive graphical interface and 300+ processors, Flow Management enables easy data ingestion and movement between CDP services as well as 3rd party cloud services. NiFi Registry is automatically set up and provides a central place to manage versioned Data Flows.

> **Note:**
>
> Flow Management clusters have distinct planning considerations and how-to information. See the Cloudera DataFlow for Data Hub documentation for information about:
>
> - Planning your Flow Management cluster deployment
> - Creating your first Flow Management cluster
> - Security considerations for Flow Management clusters
> - Using Apache NiFi to ingest data into Cloudera Public Cloud
> - Using NiFi and NiFi Registry

Cluster definition names

- Flow Management Light Duty for AWS
- Flow Management Heavy Duty for AWS

Cluster template name

- CDP - Flow Management: Light Duty
- CDP - Flow Management: Heavy Duty

Included services

- NiFi
- NiFI Registry

Compatible Cloudera Runtime versions

- 7.1.0
- 7.2.0

- 7.2.1
- 7.2.2
- 7.2.6
- 7.2.7
- 7.2.8
- 7.2.9
- 7.2.10
- 7.2.11
- 7.2.12
- 7.2.14
- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18

Related Information

Setting up your Flow Management cluster

Apache NiFi documentation

Apache NiFi Registry documentation

Streaming Analytics clusters

Learn about the default Streaming Analytics clusters, including cluster definition and template names, included services, and compatible Cloudera Runtime version.

Streaming Analytics offers real-time stream processing and stream analytics with low-latency and high scaling capabilities powered by Apache Flink.

Streaming Analytics templates include Apache Flink that works out of the box in stateless or heavy state environments. Beside Flink, the template includes its supporting services namely YARN, Zookeeper and HDFS. The Heavy Duty template comes preconfigured with RocksDB as state backend, while Light Duty clusters use the default Heap state backend. You can create your streaming application by choosing between Kafka, Kudu, and HBase as datastream connectors.

You can also use SQL to query real-time data with SQL Stream Builder (SSB) in the Streaming Analytics template. By supporting the SSB service in Cloudera Public Cloud, you can simply and easily declare expressions that filter, aggregate, route, and otherwise mutate streams of data. SSB is a job management interface that you can use to compose and run SQL on streams, as well as to create durable data APIs for the results.

> **Note:**
>
> Streaming Analytics clusters have distinct planning considerations and how-to information. See the Cloudera DataFlow for Data Hub documentation for information about:
>
> - Planning your Streaming Analytics cluster deployment
> - Creating your first Streaming Analytics cluster
> - Analyzing data using Apache Flink
> - Querying data using SQL Stream Builder

Cluster definition names

- Streaming Analytics Light Duty for AWS
- Streaming Analytics Heavy Duty for AWS

Cluster template name

- 7.2.17 - Streaming Analytics Light Duty
- 7.2.17 - Streaming Analytics Heavy Duty

Included services

- Flink
- SQL Stream Builder
- YARN
- Zookeeper
- HDFS
- Kafka

⚠️ **Important:** In the Streaming Analytics cluster templates, Kafka service is included by default to serve as a background service only for the websocket ouput and sampling feature of SQL Stream Builder. The Kafka service in the Streaming Analytics cluster template cannot be used for production, you need to use the Streams Messaging cluster template when Kafka is needed for your deployment.

Compatible Cloudera Runtime version

- 7.2.2
- 7.2.6
- 7.2.7
- 7.2.8
- 7.2.9
- 7.2.10
- 7.2.11
- 7.2.12
- 7.2.14
- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18

Related Information

Setting up your Streaming Analytics cluster

Flink

YARN

Zookeeper

HDFS

Data Discovery and Exploration clusters

Learn about the default Data Discovery and Exploration clusters, including cluster definition and template names, included services, and compatible Cloudera Runtime version.

Data Discovery and Exploration

Explore and discover data sets ad-hoc. Do relevance-based analytics over unstructured data (logs, images, text, PDFs, etc). Get started with search or log analytics. Make data more accessible to everyone with Data Discovery and Exploration.

Cluster Definition Names

- Data Discovery and Exploration for AWS

Cluster Template Name

- Data Discovery and Exploration

Included Services

- Solr
- Spark 2
- HDFS
- Hue
- YARN
- ZooKeeper

Compatible Cloudera Runtime Versions

7.2.0, 7.2.1, 7.2.2, 7.2.6, 7.2.7, 7.2.8, 7.2.9, 7.2.10, 7.2.11, 7.2.12, 7.2.14, 7.2.15, 7.2.16, 7.2.17

Cluster Definition Names

- Data Discovery and Exploration - Spark3 for AWS
- Data Discovery and Exploration - Spark3 for Azure
- Data Discovery and Exploration - Spark3 for Google Cloud

Cluster Template Name

- Data Discovery and Exploration for Spark3

Included Services

- Solr
- Spark 3
- HDFS
- Hue
- YARN
- ZooKeeper

Compatible Cloudera Runtime Version

7.2.18

Related Information

Solr

Spark

HDFS

Hue

YARN

Zookeeper

Create a cluster from a definition on AWS

You can quickly create clusters from default or custom cluster definitions within an existing AWS environment.

Before you begin

To create a Cloudera Data Hub cluster on AWS, you must have an existing AWS environment. Also, you should make sure that the Runtime version of the Data Lake cluster matches the Cloudera Runtime version of the Cloudera Data Hub cluster that you are about to create; If these versions don't match, you may encounter warnings and/or errors.

About this task

Required role: DataHubCreator, EnvironmentAdmin at the scope of the environment where the Cloudera Data Hub cluster is running, or Owner of the environment

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the Cloudera Management Console > Environments > click on an environment where you would like to create a cluster > click Create Data Hub. The following page is displayed:

3. Select Cluster Definition.
4. From the Cluster Definition dropdown, select the cluster definition that you would like to use for your cluster.

   The cluster template referenced in the selected cluster definition determines which services are included in the cluster. The list of services is automatically shown below the selected cluster definition name:

5. Specify General Settings for your cluster:

Parameter	Description
Cluster Name	Enter a name for your cluster. The name must be between 5 and 40 characters, must start with a letter, and must only include lowercase letters, numbers, and hyphens.
Tags	(Optional) Add tags that Data Hub should use to tag your AWS resources. Click Add to add a tag, and then enter a key and value for each tag. Repeat the steps if you would like to add more tags. For more information about tags, refer to Tags.

6. Optionally, click on Advanced Options to modify advanced cluster settings. For more information on these options, refer to *Advanced cluster options*.
7. On AWS and Azure only: Optionally, when you have finished providing the cluster settings, you can click the Show CLI Command button at the bottom of the page to review or copy the CDP CLI command used to create the cluster. You can copy the command from the pop-up window that appears, either to provision the cluster later or for use in scripts.
8. You also have the option to review or copy the cluster template that is generated and will be used in cluster creation. Click the Show Generated Cluster Template button at the bottom of the page.
9. To proceed with cluster provisioning immediately, click on Provision Cluster.
10. You will be redirected to the Cloudera Data Hub cluster dashboard, and a new tile representing your cluster will appear at the top of the page.

    The following messages are written to the event history as your cluster is being created:

```
Cluster built; Cluster manager ip:10.97.82.237
8/2/2019, 6:10:44 PM
```

```
Updating Cluster Proxy service with gateway configuration
8/2/2019, 6:10:43 PM
Building cluster; Cluster manager ip:10.97.82.237
8/2/2019, 5:45:12 PM
Starting cluster services
8/2/2019, 5:42:33 PM
Mounting attached disks
8/2/2019, 5:42:18 PM
Setting up infrastructure metadata
8/2/2019, 5:42:14 PM
Bootstrapping infrastructure cluster
8/2/2019, 5:41:59 PM
Registering cluster with Cluster Proxy service
8/2/2019, 5:41:57 PM
Infrastructure successfully provisioned
8/2/2019, 5:41:56 PM
Billing started, Infrastructure successfully provisioned
8/2/2019, 5:41:56 PM
Infrastructure metadata collection finished
8/2/2019, 5:41:55 PM
Infrastructure creation took 96 seconds
8/2/2019, 5:41:52 PM
Creating infrastructure
8/2/2019, 5:40:16 PM
Setting up CDP image
8/2/2019, 5:40:15 PM
```

11. When your cluster is ready, its status will change to Running.

What to do next

You can access links to Cloudera Manager, and other cluster UIs and endpoints from cluster details.

Create a custom cluster on AWS

Create a custom Cloudera Data Hub cluster within an existing AWS environment.

Before you begin

To create a Cloudera Data Hub cluster on AWS, you must have an existing AWS environment. Also, you should make sure that the Cloudera Runtime version of the Data Lake cluster matches the Runtime version of the Cloudera Data Hub cluster that you are about to create; If these versions don't match, you may encounter warnings and/or errors.

About this task

Required role: DataHubCreator, EnvironmentAdmin at the scope of the environment where the Data Hub is running, or Owner of the environment

Procedure

1. Log in to the Cloudera web interface.

2. Navigate to the Cloudera Management Console > Environments > click on an environment > click Create Data Hub.

3. Under Selected Environment, confirm that the selected environment is the one where you would like to create your cluster.

4. Select Custom.

5. Under Platform Version, current Cloudera Runtime version is pre-selected.

6. Under Cluster template, select the cluster template that you would like to use.

The selected cluster template determines which services are included in the cluster. Select the cluster template and the list of services is automatically shown below it:

For more information about cluster templates, refer to Cluster templates.

7. Specify General Settings for your cluster:

Parameter	Description
Cluster Name	Enter a name for your cluster. The name must be between 5 and 40 characters, must start with a letter, and must only include lowercase letters, numbers, and hyphens.
Tags	(Optional) Add tags that Data Hub should use to tag your AWS resources. Click Add to add a tag, and then enter a key and value for each tag. Repeat the steps if you would like to add more tags. For more information about tags, refer to Tags.

8. Optionally, click on Advanced Options to modify advanced cluster settings. For more information on these options, refer to LINK.

9. Once done, click on Provision Cluster.

10. You will be redirected to the Cloudera Data Hub cluster dashboard, and a new tile representing your cluster will appear at the top of the page.

The following messages are written to the event history as your cluster is being created:

```
Cluster built; Cluster manager ip:10.97.82.237
8/2/2019, 6:10:44 PM
Updating Cluster Proxy service with gateway configuration
8/2/2019, 6:10:43 PM
Building cluster; Cluster manager ip:10.97.82.237
8/2/2019, 5:45:12 PM
Starting cluster services
8/2/2019, 5:42:33 PM
Mounting attached disks
8/2/2019, 5:42:18 PM
Setting up infrastructure metadata
8/2/2019, 5:42:14 PM
Bootstrapping infrastructure cluster
8/2/2019, 5:41:59 PM
Registering cluster with Cluster Proxy service
8/2/2019, 5:41:57 PM
Infrastructure successfully provisioned
8/2/2019, 5:41:56 PM
Billing started, Infrastructure successfully provisioned
8/2/2019, 5:41:56 PM
Infrastructure metadata collection finished
8/2/2019, 5:41:55 PM
Infrastructure creation took 96 seconds
8/2/2019, 5:41:52 PM
Creating infrastructure
8/2/2019, 5:40:16 PM
Setting up CDP image
```

```
8/2/2019, 5:40:15 PM
```

11. When your cluster is ready, its status will change to Running.

What to do next
You can access links to Cloudera Manager, and other cluster UIs and endpoints from cluster details.

Advanced cluster options

In the create cluster wizard, click on Advanced Options to view the advanced cluster configuration options.

While some of these advanced options can be configure in the wizard, others require prior setup.

The following options are available:

Tags

You can define tags that will be applied to your cluster-related resources (such as VMs) on your cloud provider account.

The tags added during cluster creation are displayed in your cloud account on the resources that Cloudera Data Hub provisioned for your clusters. You can use tags to categorize your cloud resources by purpose, owner, and so on. Tags come in especially handy when you are using a corporate cloud provider account and you want to quickly identify which resources belong to your cluster(s). In fact, your corporate cloud account admin may require you to tag all the resources that you create, in particular resources, such as VMs, which incur charges.

By default, the following tags are created:

Tag	Description
Cloudera-Resource-Name	The workload-appropriate Cloudera resource name. This CRN serves as a unique identifier for the resource over time.
Cloudera-Creator-Resource-Name	Cloudera resource name of the Cloudera user that created the resource.
Cloudera-Environment-Resource-Name	The name of the environment with which the resource is associated.

You can optionally add additional tags. To add custom tags:

1. In the create cluster wizard, navigate to the General Configuration page.
2. Specify your tags in the Tags section by providing a key and value for each tag.

> **Note:**
>
> It is not possible to add tags via Cloudera Data Hub after your cluster has been created. In this case, you can only add the tags manually via your cloud provider's interface.

To learn more about tags and their restrictions, refer to AWS documentation.

Image catalog

The options on the "Image Settings" page of the advanced create cluster wizard allow you to select custom image settings.

By default, Cloudera Data Hub uses the prewarmed image from the image catalog provided in Cloudera Data Hub. If necessary, you can also customize a default image.

Choose image catalog

Cloudera Data Hub uses the image catalog provided by default. If you would like to use a custom image catalog instead of the default image catalog, you must first prepare your custom images, then create and register an image catalog.

Choose image type

By default, Cloudera Data Hub uses the included prewarmed images with the default Cloudera Manager and Cloudera Runtime version, but you can select a different prewarmed image or customized prewarmed image to use for your cluster. Cloudera Data Hub currently supports the following types of images for launching clusters:

Image type	Description	Default images provided
Prewarmed Image	By default, Cloudera Data Hub launches clusters from prewarmed images. Prewarmed images include the operating system as well as Cloudera Manager and Cloudera Runtime. The Cloudera Manager and Cloudera Runtime version used on prewarmed images cannot be customized.	Yes
Custom Image	You can customize a default image for compliance or security reasons. You can then use the CDP CLI to register a custom image catalog and set the custom image within the custom image catalog. Later, you can use this custom image to create a Cloudera Data Hub cluster.	

Choose image

This option allows you to select a different image.

Custom images and image catalogs

If necessary, you can use a custom Cloudera Runtime or FreeIPA image for compliance or security reasons. You can then use the Cloudera CLI to register a custom image catalog and set the custom image within the custom image catalog. Later, you can use this custom image to create a Data Lake, Cloudera Data Hub cluster or environment with a custom FreeIPA image.

Overview

A custom image should inherit most of its attributes from its source image, which is a default image that you select from the cdp-default image catalog.

The typical method of creating a Data Lake or Cloudera Data Hub picks up the latest pre-warmed image from the cdp-default image catalog for the specified version of Cloudera Runtime. These default images are pre-warmed VM images that contain a base URL to the default parcels in the Cloudera archive, amongst other configurations. If the default pre-warmed images do not suit your business needs, you can specify that the Data Lake, Cloudera Data Hub cluster or the environment (in the case of FreeIPA) uses a custom image instead.

What is a custom image?

A custom image is an entry in a custom image catalog that inherits most of its attributes from a source (default) image.

Custom image entries have:

- An image type: Cloudera Runtime [which includes Cloudera Data Hub and Data Lake images] or FreeIPA
- A source image ID that points to an image in the cdp-default image catalog

- A timestamp of creation
- An option to specify a VM region and image reference (such as an AMI ID) if you are overriding the source image with a custom VM image
- An option to override the parcel base URL

Why use a custom image?

You might require a custom image for compliance or security reasons (a "hardened" image), or to have your own packages pre-installed on the image, for example monitoring tools or software. You might also want to specify a custom image if you need to use a default image with a specific Runtime maintenance version applied, rather than simply specifying the latest major Runtime version.

Note: When customizing VM images, note that certain customizations (for example, CIS hardening rules) may not be compatible with Cloudera.

What can you customize?

In a custom image entry, you can override the VM images themselves with your own custom images that are sufficiently hardened. Importantly, you should only customize a default image from the cdp-default catalog as opposed to creating one from scratch. You can also override the default parcel base URL (at archive.cloudera.com) with your own host site.

What is a custom image catalog?

A custom image catalog is simply a catalog that holds custom images. A custom image catalog can contain a single or multiple custom image entries.

Custom image catalogs have:

- A name. The name is a unique identifier and is used to refer to the catalog during environment, Data Lake, and Cloudera Data Hub cluster creation; as well as during catalog operations like creating an image.
- A description.
- An owner. The owner is the user who runs the command to create the catalog.

What is the process for creating a custom image and catalog?

- If you are replacing the VM images in a custom image entry with a customized version, you should first prepare the image by modifying an official Cloudera default image, which you can find under Shared Resources > Image Catalogs > cdp-default.
- Select a source image from the cdp-default image catalog to be the source of customization. When you run the CLI command to find a default image, you specify the Runtime version, provider, image type, or a combination of the three.
- Create a custom image catalog, or identify an existing catalog where you want to save the custom image entry.
- Apply the necessary changes to the custom image entry, like the override AMI IDs with the new, customized AMIs; or add a new parcel base URL using the `--base-parcel-url` command when you set the custom image.
- You can then create an environment, Data Lake, or Cloudera Data Hub cluster, based on custom catalogs via the CDP CLI.

Creating a custom image and image catalog with the CDP CLI

You can create a custom Cloudera Runtime or FreeIPA image and image catalog through the CDP CLI.

Before you begin
If you are replacing the VM images in a custom image entry with a customized version, you must first prepare the image by modifying an official Cloudera default image, which you can find under Shared Resources Image Catalogs cdp-default . Take note of the image reference, such as the AMI ID.

Procedure

1. To find a source image from the cdp-default catalog that you want to use as the source of your custom image entry, run the following command:

```
cdp imagecatalog find-default-image --provider <cloud provider> --image-
type <image type> --runtime-version <Cloudera Runtime version>
```

For example:

```
cdp imagecatalog find-default-image --provider AWS --image-type runtime --
runtime-version 7.2.12
```

2. A custom image requires a custom catalog. If you haven't yet created a custom catalog, or if you want to create a new one for a new custom image, run the following command:

```
cdp imagecatalog create-custom-catalog --catalog-name <unique catalog na
me> --description <catalog description>
```

For example:

```
cdp imagecatalog create-custom-catalog --catalog-name my custom catalog --
description test catalog
```

3. Within the custom image catalog that you created (or an older custom catalog that you want to use), create a custom image entry with the selected source image marked as its source. Providing your own VM images is optional. Run the following command:

```
cdp imagecatalog set-<image-type>-image --catalog-name <name of the custom
 catalog> --vm-images region=<region of customized image>,imageReference=<
cloud provider specific ID of a customized image> --source-image-id <cdp
 image ID of source image>
```

For example:

```
cdp imagecatalog set-runtime-image --catalog-name my custom catalog --vm-
images region=eu-central-1,imageReference=ami-7torotmhqi6q7438y --source-
image-id 8t4y9853-12b6-3n6z-75dh-tx775k4c793w
```

4. You can then apply the necessary changes to the custom image entry, like overriding AMI IDs with new, customized ones, or adding a new parcel base URL.

Command	Description
cdp imagecatalog find-default-image	Finds the default images in the cdp-default image catalog for a specified version of Cloudera Runtime. Parameters: • --provider • --image-type • --runtime-version You can provide any combination of these parameters.
cdp imagecatalog create-custom-catalog	Creates a custom catalog. Parameters: • --catalog-name • --description

Command	Description
	–catalog-name is required.
cdp imagecatalog set-runtime-image or cdp imagecatalog set-freeipa-image	Creates a custom image entry (either Cloudera Runtime [Cloudera Data Hub/Data Lake] or FreeIPA) within the specified catalog. Parameters: • --catalog-name • --vm-images [region,imageReference] • --source-image-id • --image-id • --base-parcel-url --catalog-name and --source-image-id are required.

Switching image catalogs

You can switch the image catalog of an already existing Cloudera Data Hub, Data Lake, or FreeIPA cluster. You may want to switch the image catalog for a cluster in order to restrict which Cloudera Runtime version can be upgraded to, or in order to move to custom images for an existing cluster.

Use the following CDP CLI commands to switch the image catalog for an existing cluster:

• FreeIPA:

```
cdp environments set-catalog --environment $ENVIRONMENT_NAME --catalog $
CATALOG
```

Parameter	Description
--environment	Name or CRN of the environment that holds the FreeIPA installation.
--catalog	URL of the FreeIPA catalog to be used.

• Cloudera Data Hub:

```
cdp datahub set-catalog --cluster $DATAHUB_NAME --catalog-name $CATALOG
```

Parameter	Description
--cluster	Name or CRN of the Cloudera Data Hub cluster for which you want to use the new image catalog.
--catalog-name	Name of the image catalog to be used. The image catalog must be a JSON based catalog, and switching is only possible from catalogs that are JSON based.

• Data Lake:

```
cdp datalake set-catalog --datalake $DATALAKE_NAME --catalog-name $CATALOG
```

Parameter	Description
--datalake	Name of the Data Lake for which you want to use the new image catalog.

Parameter	Description
--catalog-name	Name of the image catalog to be used. The image catalog must be a JSON based catalog, and switching is only possible from catalogs that are JSON based.

Network and availability

The "Network and availability" options allow you to customize the networking settings of your cluster.

On the Network and Availability page, provide the following to specify the networking resources that will be used for your cluster:

Parameter	Description
Select Subnet	If your environment includes a single subnet, that subnet is preselected and it cannot be changed. If your environment includes more than one subnet, you can select the subnet(s) in which your cluster will be provisioned. If you would like to deploy your Cloudera Data Hub cluster in multiple availability zones, you should select multiple subnets.
Select Azure Database Server	A Cloudera Data Hub cluster uses the same Flexible Server or Single Server settings as the environment in which it runs but you can choose to enable a Flexible server on Cloudera Data Hub clusters running in an environment that uses a Single server. For more information, see Using Azure Database for PostgreSQL Flexible Server.

Hardware and storage

The "Hardware and storage" options allow you to customize the cloud provider specific cluster hardware and storage options.

The Hardware and Storage options can be selected for each host group. To edit this section for a specific host group, click on the ✏️. When done editing, click on the 📋 to save the changes. Repeat for these steps for all host groups that you would like to edit.

The following hardware and storage settings are available:

Parameter	Description
Cloudera Manager Server	You must select one node for Cloudera Manager Server by clicking the ⬤ button. The "Instance Count" for that host group must be set to "1". If you are using one of the default cluster templates, this is set by default.
Instance Type	Select an instance type. For information about instance types on AWS refer to Amazon EC2 Instance Types in AWS documentation.
Instance Count	Enter the number of instances of a given type. Default is 1.
Storage Type	Select the volume type. The options vary by instance type and include: (1) Ephemeral (2) Magnetic (3) General Purpose SSD, (4) Throughput Optimized HDD. For more information about these options refer to Amazon EC2 Instance Store in AWS documentation. **Note:** Stopping and restarting Cloudera Data Hub clusters using ephemeral storage is not supported.
Encryption	Under Encryption Key, you can select an existing encryption key. For more information, refer to EBS Encryption on AWS.
Attached Volumes Per Instance	Enter the number of volumes attached per instance. Default is 1.

Parameter	Description
Volume Size	Enter the size in GB for each volume. Default is 100.
Root Volume Size	This option allows you to increase or decrease the root volume size. Default is 200 GB. This option is useful if your custom image requires more space than the default 200 GB. If you use a custom Cloudera Data Hub template specifying a root volume size smaller than 200GB, you may encounter an error.

Cloud storage

The options on the "Cloud Storage" page allow you to optionally specify the base storage location used for YARN and Zeppelin.

During environment creation under Data Access > Storage Location Base you configure a default S3 base storage location for the environment, and all Cloudera Data Hub clusters created within that environment use this location. The Cloud Storage options in the Cloudera Data Hub cluster wizard allow you to additionally specify a different location for YARN application logs and Zeppelin Notebook's root directory:

- Existing Base Storage Location - By default, this is set to the Storage Location Base configured on environment level. If you do not want to make any changes, simply leave this blank. If you would like to use a different location for YARN application logs and Zeppelin Notebook's root directory, you can specify a different S3 location. Note that the specified S3 location must exist prior to Cloudera Data Hub cluster creation and that you must adjust the IAM policies created during environment's cloud storage setup to make sure that IDBroker has write access to this location.
- Path for YARN Application Logs property - This directory structure gets created automatically during cluster creation. You can customize it if you would like it be different than what is suggested by default.
- Path for Zeppelin Notebooks Root Directory property - This directory structure gets created automatically during cluster creation. You can customize it if you would like it to be different than what is suggested by default.

Any S3 bucket that you designate for Cloudera Data Hub cloud storage on AWS must be in the same region as the environment.

Building a custom cluster template

You can build a custom cluster template to modify the cluster Cloudera Runtime services, including the Cloudera Runtime configuration properties and the distribution of Runtime services across host groups. To create a custom template, modify an existing default cluster template and then upload and register the custom template.

About this task
If the default cluster templates are insufficient for the cluster that you want to create, you can build a custom cluster template. The recommended method for building a cluster template is to modify an existing default template, so that the structure of the template is mostly preserved.

Required role: EnvironmentCreator can create a shared resource and then assign users to it. SharedResourceUser or Owner of the shared resource can use the resource. The Owner of the shared resource can delete it.

Procedure

1. Review information on the default cluster configurations to find one that includes services suitable for the type of cluster that you want to create. In general, it is best to use the templates for the current release.
2. To access the existing default cluster templates, click Shared ResourcesCluster Templates.
3. To find the newest template versions, click Platform at the top of the Platform column to sort the templates in descending order.

4. Under the Name column, click the desired default ("Built In") template.

Environments	Shared Resources

Cluster Definitions	**67** Cluster Templates ⟳
Cluster Templates	
Proxies	🔍 Search
Credentials	
Recipes	
Image Catalogs	
Tags	
Telemetry	

Name	Platform ↓	Group Count	Description	Tags	Time Created
7.2.2 - Data Discovery and Exploration	Cloudera Runtime 7.2.2	5	7.2.2 - Data Discovery and Exploration	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Data Engineering: Apache Spark, Apache Hive, Apache Oozie	Cloudera Runtime 7.2.2	4	7.2.2 - Data Engineering	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Data Engineering: Apache Spark3	Cloudera Runtime 7.2.2	3	7.2.2 - Data Engineering Spark3	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Data Engineering: HA: Apache Spark, Apache Hive, Apache Oozie	Cloudera Runtime 7.2.2	5	7.2.2 - Data Engineering HA	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Data Mart: Apache Impala, Hue	Cloudera Runtime 7.2.2	3	7.2.2 - Data Mart with Apache Impala and Hue	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Flow Management Heavy Duty with Apache NiFi, Apache NiFi Registry	Cloudera Runtime 7.2.2	3	7.2.2 - Flow Management Heavy Duty with Apache NiFi, Apache NiFi Registry	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Flow Management Light Duty with Apache NiFi, Apache NiFi Registry	Cloudera Runtime 7.2.2	2	7.2.2 - Flow Management Light Duty with Apache NiFi, Apache NiFi Registry	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Operational Database: Apache HBase, Phoenix	Cloudera Runtime 7.2.2	4	7.2.2 - Operational Database: Apache HBase, Phoenix	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark	Cloudera Runtime 7.2.2	5	7.2.2 - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark	Built-In	8/17/2020, 1:35:51 PM CDT
7.2.2 - Streaming Analytics Heavy Duty with Apache Flink	Cloudera Runtime 7.2.2	3	7.2.2 - Streaming Analytics Heavy Duty with Apache Flink	Built-In	8/17/2020, 1:35:51 PM CDT

The template opens in LIST view, which shows how the template is structured across host groups.

5. Click RAW VIEW to view the JSON structure.

Environments   Shared Resources

Cluster Templates
Proxies
Credentials
Recipes
Image Catalogs

7.2.2 - Data Engineering: Apache Spark, Apache Hive, Apache Oozie
7.2.2 - Data Engineering

Delete

LIST VIEW   RAW VIEW

```json
{
  "cdhVersion": "7.2.2",
  "displayName": "dataengineering",
  "services": [
    {
      "refName": "zookeeper",
      "serviceType": "ZOOKEEPER",
      "serviceConfigs": [
        {
          "name": "service_config_suppression_server_count_validator",
          "value": "true"
        }
      ],
      "roleConfigGroups": [
        {
          "refName": "zookeeper-SERVER-BASE",
          "roleType": "SERVER",
          "base": true
        }
      ]
    },
    {
      "refName": "hdfs",
      "serviceType": "HDFS",
      "serviceConfigs": [
        {
          "name": "hdfs_verify_ec_with_topology_enabled",
          "value": false
        },
        {
          "name": "core_site_safety_valve",
          "value": "<property><name>fs.s3a.buffer.dir</name><value>${env.LOCAL_DIRS:-${hadoop.tmp.dir}}/s3a</value></property><property>
<name>fs.s3a.committer.name</name><value>directory</value></property>"
        }
      ],
      "roleConfigGroups": [
        {
          "refName": "hdfs-NAMENODE-BASE",
          "roleType": "NAMENODE",
          "base": true,
          "configs": [
            {
              "name": "role_config_suppression_fs_trash_interval_minimum_validator",
              "value": "true"
            },
            {
              "name": "role_config_suppression_namenode_java_heapsize_minimum_validator",
              "value": "true"
            },
            {
              "name": "fs_trash_interval",
              "value": "0"
            },
            {
              "name": "fs_trash_checkpoint_interval",
              "value": "0"
            },
            {
              "name": "erasure_coding_default_policy",
              "value": " "
            }
          ]
        },
        {
          "refName": "hdfs-SECONDARYNAMENODE-BASE",
          "roleType": "SECONDARYNAMENODE",
          "base": true
        },
        {
          "refName": "hdfs-DATANODE-BASE",
          "roleType": "DATANODE",
          "base": true
        },
        {
          "refName": "hdfs-BALANCER-BASE",
          "roleType": "BALANCER",
          "base": true
        },
        {
          "refName": "hdfs-GATEWAY-BASE",
          "roleType": "GATEWAY",
          "base": true,
          "configs": [
            {
              "name": "dfs_client_use_trash",
              "value": false
            },
            {
              "name": "role_config_suppression_hdfs_trash_disabled_validator",
              "value": "true"
            },
            {
              "name": "hdfs_client_env_safety_valve",
              "value": "HADOOP_OPTS=\"-Dorg.wildfly.openssl.path=/usr/lib64 ${HADOOP_OPTS}\""
            }
          ]
        }
      ]
    },
    {
      "refName": "yarn",
      "serviceType": "YARN",
      "serviceConfigs": [
        {
          "name": "yarn_admin_acl",
          "value": "yarn,hive,hdfs,mapred"
        },
        {
          "name": "yarn_service_mapred_safety_valve",
          "value": "<property><name>mapreduce.fileoutputcommitter.algorithm.version</name><value>1</value></property><property>
<name>mapreduce.input.fileinputformat.list-status.num-threads</name><value>100</value></property>"
        }
      ],
      "roleConfigGroups": [
        {
          "refName": "yarn-RESOURCEMANAGER-BASE",
          "roleType": "RESOURCEMANAGER",
          "base": true,
          "configs": [
            {
              "name": "resourcemanager_config_safety_valve",
              "value": "<property><name>yarn.scheduler.configuration.store.class</name><value>zk</value></property>"
            },
            {
              "name": "yarn_resourcemanager_scheduler_class",
              "value": "org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler"
            },
            {
              "name": "yarn_scheduler_capacity_resource_calculator",
              "value": "org.apache.hadoop.yarn.util.resource.DefaultResourceCalculator"
            },
            {
              "name": "resourcemanager_capacity_scheduler_configuration",
              "value": "<configuration><property><name>yarn.scheduler.capacity.root.queues</name><value>default</value></property><property>
<name>yarn.scheduler.capacity.root.capacity</name><value>100</value></property><property><name>yarn.scheduler.capacity.root.default.capacity</name>
<value>100</value></property><property><name>yarn.scheduler.capacity.root.acl_submit_applications</name><value> </value></property><property>
<name>yarn.scheduler.capacity.root.acl_administer_queue</name><value> </value></property><property>
<name>yarn.scheduler.capacity.root.default.acl_submit_applications</name><value>*</value></property><property>
<name>yarn.scheduler.capacity.root.default.minimum-user-limit-percent</name><value>100</value></property><property>
<name>yarn.scheduler.capacity.maximum-am-resource-percent</name><value>0.33</value></property><property><name>yarn.scheduler.capacity.node-
locality-delay</name><value>0</value></property><property><name>yarn.scheduler.capacity.schedule-asynchronously.maximum-threads</name>
<value>1</value></property><property><name>yarn.scheduler.capacity.schedule-asynchronously.scheduling-interval-ms</name><value>10</value>
</property></configuration>"
            }
          ]
        },
        {
          "refName": "yarn-NODEMANAGER-WORKER",
          "roleType": "NODEMANAGER",
          "base": false
        },
        {
          "refName": "yarn-NODEMANAGER-COMPUTE",
          "roleType": "NODEMANAGER",
          "base": false
        },
        {
          "refName": "yarn-JOBHISTORY-BASE",
          "roleType": "JOBHISTORY",
          "base": true
        },
        {
          "refName": "yarn-GATEWAY-BASE"
```

29

6. Select all of the JSON code, copy it, and paste it to a suitable code editor. Standard text editors are not recommended.

7. If you are using a code editor such as Microsoft Visual Studio Code, you can use built-in tools to validate the JSON before you proceed. Optionally, you can save the file as-is (without having made any changes), and upload it using the instructions in the topic *Upload a cluster template*. To verify that the template JSON is functional, you can create a test cluster by selecting the template that you just registered, and see if the cluster successfully deploys.

8. When you are satisfied that you are working with a clean template, you can begin to modify the template. Return to the JSON template file in your code editor.

Each default template consists of two main sections: the services section and the hostTemplates section. The serv ices section includes the components that make up the cluster. This is where you can add or remove services, as well as modify service configuration properties. If you want to modify a service's configuration, for example to tune Yarn or Hive, refer to the Cloudera Manager configuration properties for the desired service. You can search these properties by their API Name, which is how they appear in a Cloudera Data Hub template.

For example, in a Data Engineering template you might want to adjust the amount of physical memory allocated for containers by configuring the yarn.nodemanager.resource.memory-mb property. If you want to configure this property to 80% of the total system RAM, for a 256 GB machine this would look like:

```
{
      "refName": "yarn",
      "serviceType": "YARN",
      "serviceConfigs": [
        {
          "name": "yarn_admin_acl",
          "value": "yarn,hive,hdfs,mapred"
        }
      ],
      "roleConfigGroups": [
        {
          "refName": "yarn-RESOURCEMANAGER-BASE",
          "roleType": "RESOURCEMANAGER",
          "base": true,
          "configs": [
            {
              "name": "yarn.nodemanager.resource.memory-mb",
              "value": 2052096
            },
```

If you want to add a service in a template, the simplest method is to find the service in the RAW VIEW of another default template and copy it into your JSON.

For example, if you want to add Sqoop to the services in a template, copy it from the Data Engineering or Data Engineering HA template into the services section of another template:

```
{
      "refName": "sqoop",
      "serviceType": "SQOOP_CLIENT",
      "roleConfigGroups": [
        {
          "refName": "sqoop-SQOOP_CLIENT-GATEWAY-BASE",
          "roleType": "GATEWAY",
          "configs": [],
          "base": true
        }
      ]
```

```
        },
```

The hostTemplates section of the JSON file describes the nodes by their type and the services on the node. This section also includes a cardinality parameter, which you can set to increase or decrease the quantity of that specific node type.

For example, say that you want to create a new node type called "ZKserver" that runs a single service, Zookeeper. Assuming that Zookeeper is already a service defined in the services section of the template, you can move down to the hostTemplate section. In the master node section of this Data Engineering template, you can see that Zookeeper is already defined in the "master" node section with the string "zookeeper-SERVER-BASE":

```
 "hostTemplates": [
     {
       "refName": "master",
       "cardinality": 1,
       "roleConfigGroupsRefNames": [
         "hdfs-BALANCER-BASE",
         "hdfs-NAMENODE-BASE",
         "hdfs-SECONDARYNAMENODE-BASE",
         "hdfs-GATEWAY-BASE",
         "hms-GATEWAY-BASE",
         "hms-HIVEMETASTORE-BASE",
         "hive_on_tez-HIVESERVER2-BASE",
         "hive_on_tez-GATEWAY-BASE",
         "hue-HUE_LOAD_BALANCER-BASE",
         "hue-HUE_SERVER-BASE",
         "tez-GATEWAY-BASE",
         "spark_on_yarn-GATEWAY-BASE",
         "spark_on_yarn-SPARK_YARN_HISTORY_SERVER-BASE",
         "livy-LIVY_SERVER-BASE",
         "zeppelin-ZEPPELIN_SERVER-BASE",
         "oozie-OOZIE_SERVER-BASE",
         "sqoop-SQOOP_CLIENT-GATEWAY-BASE",
         "yarn-JOBHISTORY-BASE",
         "yarn-RESOURCEMANAGER-BASE",
         "zookeeper-SERVER-BASE",
         "das-DAS_WEBAPP",
         "das-DAS_EVENT_PROCESSOR",
         "yarn-QUEUEMANAGER_WEBAPP-BASE",
         "yarn-QUEUEMANAGER_STORE-BASE",
         "yarn-GATEWAY-BASE"
       ]
```

To create our new ZKserver node, you can copy the standard node format and modify it for your purpose:

```
 {
         "refName": "ZKserver",
         "cardinality": 1,
         "roleConfigGroupsRefNames": [
           "zookeeper-SERVER-BASE"
         ]
       }
```

If you also want to include dynamic parameters in your custom template, see the documentation for Dynamic cluster templates. During the cluster creation phase, dynamic parameters pick up the parameter values that you provided in the Cloudera Data Hub cluster wizard. See the *Custom Properties* documentation for a list of properties that can be dynamically replaced. You might want to use dynamic parameters when you regularly provision clusters using a specific template, but want to change a few of the property values each time you provision a new cluster.

9. When you have finished modifying the template, validate the JSON in your code editor and save the template.

What to do next

Upload the JSON file and register the template following the instructions in the *Upload a cluster template* topic. Then, when you navigate to the Data Hubs page and select Create Data Hub, be sure to select the Custom radio button underneath environment selection. Here you can provision a Cloudera Data Hub cluster using the custom template that you registered. Select your custom template from the drop-down menu before you configure any advanced options.

Related Information

Upload a cluster template

CDP Public Cloud upgrade advisor

Compute resources deployed via cloud services are generally considered transient in nature. With the separation of compute and storage, compute clusters in CDP can be stopped or decommissioned, while the data used or created by workloads running on these clusters generally remains accessible on persistent cloud storage.

There are some exceptions to the above, most importantly SDX metadata and cloud service-specific metadata. SDX metadata is stored in the Data Lake, while metadata specific to a cloud service may be stored in databases, local configurations, or even locally attached disks. This local storage can also be persistent (block storage volumes) or transient (ephemeral disks).

In cloud services where compute is elastic and state is transient, we need safeguards to protect all data that is not persistent, especially when changes are performed on the services themselves. In CDP Public Cloud, there can be several changes, most notably Runtime maintenance upgrades, Runtime minor/major software version upgrades, and OS upgrades.

In general, there are two main approaches to upgrading cloud services:

1. Backup, re-create and restore
2. In-place upgrade

These two approaches have similarities: a prior backup should be performed, service endpoints should remain stable after the operation, and they should result in the same outcome. CDP Public Cloud supports both approaches. While the first approach may be convenient for simple data workloads, complex data applications and their custom data pipelines spanning multiple clusters may require an in-place upgrade path.

In this guide we will describe the high-level steps of performing in-place upgrade of Data Lake and Data Hub clusters. For steps required for the backup and restore approach, refer to the respective documentation on backing up Data Lakes and Data Hubs and performing metadata restore (automated for Data Lake clusters only).

Upgrade Checklist FAQ

During the preparation for an upgrade, Cloudera recommends carefully reviewing the questions and answers below.

What is the length of the available maintenance window?

> Currently, Data Lake backup and restore requires a maintenance window, where no metadata changes occur. Furthermore, Data Hubs need to be stopped during an upgrade.

> The CDP Public Cloud environment does not need to be upgraded in one go: you may opt to upgrade the Data Lake and all attached Data Hubs together, or start with the Data Lake upgrade only and perform individual Data Hub upgrades in consecutive steps. However, after you perform a major/minor Data Lake upgrade, you must upgrade all attached Data Hub clusters, as the Data Hubs must run the same major/minor version of Runtime as the Data Lake.

What type of upgrade is required?

> Currently, there are three types of upgrades available to Data Lake and Data Hub clusters: maintenance upgrades; minor/major version upgrades; and OS upgrades. Maintenance and minor/

major version upgrades install a newer version of Cloudera Manager and/or Cloudera Runtime. OS upgrades for Data Lakes and Data Hubs are complementary and will bring the image of the cluster hosts to a newer version. If you plan to also perform an OS upgrade, plan the maintenance window accordingly.

Are ephemeral disks used for user or workload-related persistent data?

Major/minor version upgrades as well as maintenance upgrades will bring Cloudera Manager and Cloudera Runtime to the selected version without impacting the underlying VM. However, OS upgrades will recreate the underlying VM with a fresh image, which results in the loss of any data stored on ephemeral disks.

If you are currently storing user or workload-related data on volumes using ephemeral disks, please reach out to Cloudera support while planning for the upgrade.

What Data Hub cluster templates are in use? Are you using custom templates?

Check whether in-place upgrade is supported for your built-in or custom data hub template. Depending on the type and version of the Data Hub, additional backup steps, manual configuration changes or post-upgrade steps may be required. Check specific steps for upgrading the OS if you use Flow Management. Operational Database clusters have a different upgrade process.

What is the size of the SDX / Data Lake metadata?

SDX metadata includes the Hive Metastore database, Ranger audit log index, as well as Atlas metadata. If you are planning to perform a Data Lake backup before an upgrade (which is recommended), prepare your maintenance window accordingly. CDP supports skipping the backup of certain metadata to reduce the time required for backup and restore operations.

Are you using Data Services?

If you have deployed Cloudera Data Engineering, Data Warehouse, Data Flow, or Machine Learning in your environment, it is recommended that you upgrade them to the latest version before upgrading your Data Lake to a more recent minor/major version.

Preparing for an upgrade

Upgrading CDP Public Cloud consists of two major steps: upgrading the Data Lake within an environment and then upgrading the attached Data Hubs. Currently the Data Hubs must run the same version of Cloudera Runtime as the Data Lake.

You should also periodically upgrade the environment (FreeIPA cluster) to ensure that you are running the latest security patches, but this is not required at the same time as upgrading the Data Lake and Data Hubs.

Data Lake upgrade workflow

Pre-upgrade tasks

1. Review the Data Lake upgrade requirements and limitations.
2. Carefully review the differences between the types of upgrades and what they entail.
3. Check the Data Lake support matrix for upgrade to verify which Runtime versions you can upgrade to and from.
4. If you have not configured your Data Lake for backup and restore, you will need to do so. The backup and restore process is integrated into the upgrade flow automatically, but successful upgrade requires that the correct IAM policies exist on the DATALAKE_ADMIN_ROLE and RANGER_AUDIT_ROLE (for backup), and the DATALAKE_ADMIN_ROLE, RANGER_AUDIT_ROLE, and LOG_ROLE (for restore).

   If your roles are not configured correctly, CDP will not be able to write the backup to the BACKUP_LOCATION_BASE path of your cloud storage.
5. If you are performing the backup manually (as opposed to the integrated backup available during the upgrade process), you can launch the Data Lake backup from the UI or CLI. When using the CLI, you can specify to skip

certain backup actions (skip HMS, Atlas metadata or Ranger audit log index backup). You can monitor the backup process using the CLI.

6. From the Data Lake UI, run the Validate and Prepare option to check for any configuration issues and begin the Cloudera Runtime parcel download and distribution. Using the validate and prepare option does not require downtime and makes the maintenance window for an upgrade shorter. Validate and prepare also does not make any changes to your cluster and can be run independently of the upgrade itself. Although you can begin the upgrade without first running the validate and prepare option, using it will make the process smoother and the downtime shorter. (The parcels that are downloaded and distributed by the validate and prepare option are specific to the Runtime version that you have selected, so if you use validate and prepare and then decide to upgrade to a different Runtime version instead, you will need to re-run validate and prepare. Be aware that if you use validate and prepare for multiple major/minor Runtime versions, the parcels for different versions are not cleaned up and may saturate the disk. These parcels are cleaned up only once the upgrade is complete.)

Data Lake upgrade tasks

1. Perform the upgrade, either through the UI or CLI. The type of upgrade that you perform will depend on whether a new version of Runtime or only a hotfix is available. A new OS image may also be available to upgrade to.
2. If the upgrade fails, check the logs for manual troubleshooting info. You can also recover from failed upgrades.
3. When the upgrade succeeds, proceed to upgrading the attached Data Hubs.

Data Hub upgrade workflow

Pre-upgrade tasks

1. Carefully review the differences between the types of upgrades and what they entail.
2. Check that the cluster that you want to upgrade is supported.
3. From the Data Hub UI, run the Validate and Prepare option to check for any configuration issues and begin the Cloudera Runtime parcel download and distribution. Using the validate and prepare option does not require downtime and makes the maintenance window for an upgrade shorter. Validate and prepare also does not make any changes to your cluster and can be run independently of the upgrade itself. Although you can begin the upgrade without first running the validate and prepare option, using it will make the process smoother and the downtime shorter. (The parcels that are downloaded and distributed by the validate and prepare option are specific to the Runtime version that you have selected, so if you use validate and prepare and then decide to upgrade to a different Runtime version instead, you will need to re-run validate and prepare. Be aware that if you use validate and prepare for multiple major/minor Runtime versions, the parcels for different versions are not cleaned up and may saturate the disk. These parcels are cleaned up only once the upgrade is complete.)

Major/minor Runtime version upgrade tasks

1. Backup the cluster data and CM configurations.
2. Perform the upgrade, either through the UI or CLI. (Operational database clusters have a different process.)
3. Complete any post-upgrade tasks for the type of cluster that you upgraded.
4. For DE clusters, use the CM UI to add any configs that were not added during the upgrade.
5. If the upgrade fails, check the Event log and the troubleshooting section.
6. Complete any post-upgrade tasks for the type of cluster that you upgraded.
7. For DE clusters, use the CM UI to add any configs that were not added during the upgrade.
8. If the upgrade fails, check the Event log and the troubleshooting section.

Maintenance (hotfix) upgrade tasks

1. Backup the cluster data and CM configurations.
2. Perform the upgrade, either through the UI or CLI.
3. If the upgrade fails, check the Event log and the troubleshooting section.

OS upgrade tasks

1. Review the Before you begin section to verify that there is no data belonging to NiFi or NiFi Registry on the root disk of the VM. Note that during an OS upgrade, any data on the root volume (parcels, service logs, custom software) will be lost.
2. Unlike the Data Lake upgrade, OS upgrades are not integrated in the larger upgrade flow and must be performed separately, either through the UI or CLI.

FreeIPA upgrades

When a FreeIPA upgrade is available, upgrade the FreeIPA cluster to ensure that the nodes are running the latest OS-level security patches.