

Cluster Definitions

Date published: 2019-12-17

Date modified: 2025-08-18

CLOUDERA

Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Cluster definitions.....	4
Default cluster configurations.....	4
Data Engineering clusters.....	5
Data Mart clusters.....	14
Operational Database clusters.....	16
Streams Messaging clusters.....	16
Flow Management clusters.....	17
Streaming Analytics clusters.....	18
Data Discovery and Exploration clusters.....	20
Saving a custom cluster definition.....	21
Deleting a cluster definition.....	21
Managing cluster definitions from CLI.....	22

Cluster definitions

A cluster definition is a reusable cluster template in JSON format that can be used for creating multiple Cloudera Data Hub clusters with identical cloud provider settings.

Data Hub includes a set of default cluster definitions for common data analytics and data engineering use cases, and allows you to save your own custom cluster definitions. While default cluster definitions can be used across all environments, custom cluster definitions are associated with one or more specific environments. Cluster definitions facilitate repeated cluster reuse and are therefore useful for creating automation around cluster creation.

The easiest way to create a custom cluster definition is to open the create cluster wizard, provide all the parameters, and then save the settings as a cluster definition. The option to save is available on the last page of the create cluster wizard.

**Note:**

A cluster definition is not synonymous with a cluster template. While a cluster template primarily defines Cloudera Runtime services, a cluster definition primarily includes cloud provider settings. Furthermore, a cluster definition always references one specific cluster template.

Prior to creating your own cluster definitions, we recommend that you review the default cluster definitions to check if they meet your requirements. You can access the default cluster definitions by clicking Environments, then selecting an environment and clicking the Cluster Definitions tab. To view details of a cluster definition, click on its name. 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.

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 for Azure
- Data Engineering for Google Cloud
- Data Engineering HA - Spark3 for AWS
- Data Engineering HA - Spark3 for Azure
- Data Engineering HA - Spark3 for Google Cloud
- Data Engineering Spark3 for AWS
- Data Engineering Spark3 for Azure
- Data Engineering Spark3 for Google Cloud
- Data Engineering - Spark3 (ARM) for AWS
- Data Engineering HA - Spark3 (ARM) 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."
- 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 is no longer supported starting with Cloudera Runtime version 7.3.1.
- 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

- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18
- 7.3.1

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:


For AWS**Table 1: Data Engineering in AWS**

Host group	Description	Node configuration
Master Node count: 1	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 as HiveServer2, HMS, Hue etc.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> • m5.4xlarge; gp2 - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> • m5.4xlarge; gp2 - 100 GB
Worker Node count: 3	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.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> • m5.2xlarge; gp2 - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> • DE and DE Spark3: <ul style="list-style-type: none"> • r5d.2xlarge - (gp2/EBS volumes) • DE HA: <ul style="list-style-type: none"> • r5d.4xlarge - (gp2/EBS volumes)
Compute Node count: 0+	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.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> • m5.2xlarge; gp2 - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> • DE and DE Spark3: <ul style="list-style-type: none"> • r5d.2xlarge - (ephemeral volumes) • DE HA: <ul style="list-style-type: none"> • r5d.4xlarge - (ephemeral volumes)  <p>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.</p>

Host group	Description	Node configuration
Gateway Node count: 0+	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.	m5.2xlarge; gp2 - 100 GB

For Azure


Table 2: Data Engineering in Azure

Host group	Description	Node configuration
Master Node count: 1	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 as HiveServer2, HMS, Hue etc.	<p>For clusters created with Cloudera Runtime versions lower than 7.2.14:</p> <ul style="list-style-type: none"> Standard_D16_v3; StandardSSD_LRS - 100 GB <p>For clusters created with Cloudera Runtime versions 7.2.14 or higher versions:</p> <ul style="list-style-type: none"> Standard_D16_v3
Worker Node count: 3	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.	<p>For clusters created with Cloudera Runtime versions lower than 7.2.14:</p> <ul style="list-style-type: none"> Standard_D8_v3; StandardSSD_LRS - 100 GB <p>For clusters created with Cloudera Runtime versions 7.2.14 or higher versions:</p> <ul style="list-style-type: none"> DE and DE Spark3: <ul style="list-style-type: none"> Standard_D5_v2 DE HA: <ul style="list-style-type: none"> Standard_D5_v2
Compute Node count: 0+	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.	<p>For clusters created with Cloudera Runtime versions lower than 7.2.14:</p> <ul style="list-style-type: none"> Standard_D8_v3; StandardSSD_LRS - 100 GB <p>For clusters created with Cloudera Runtime versions 7.2.14 or higher versions:</p> <ul style="list-style-type: none"> DE and DE Spark3: <ul style="list-style-type: none"> Standard_D5_v2 DE HA: <ul style="list-style-type: none"> Standard_D5_v2 <p>For Azure, the attached volume count for the compute host group is changed to 0. Only ephemeral/local volumes are used by default.</p> <p> 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.</p>

Host group	Description	Node configuration
Gateway Node count: 0+	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.	Standard_D8_v3; StandardSSD_LRS - 100 GB

For GCP

Table 3: Data Engineering in GCP

Host group	Description	Node configuration
Master Node count: 1	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 as HiveServer2, HMS, Hue etc.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> e2-standard-16; pd-ssd - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> e2-standard-16; pd-ssd - 100 GB
Worker Node count: 3	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.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> DE and DE Spark3: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB DE HA: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB
Compute Node count: 0+	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.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> DE and DE Spark3: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB DE HA: <ul style="list-style-type: none"> e2-standard-8; pd-ssd - 100 GB  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 Node count: 0+	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.	e2-standard-8; pd-ssd - 100 GB

Configurations

The following table summarizes the service configurations based on the host groups:

Table 4:

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

Note the following:

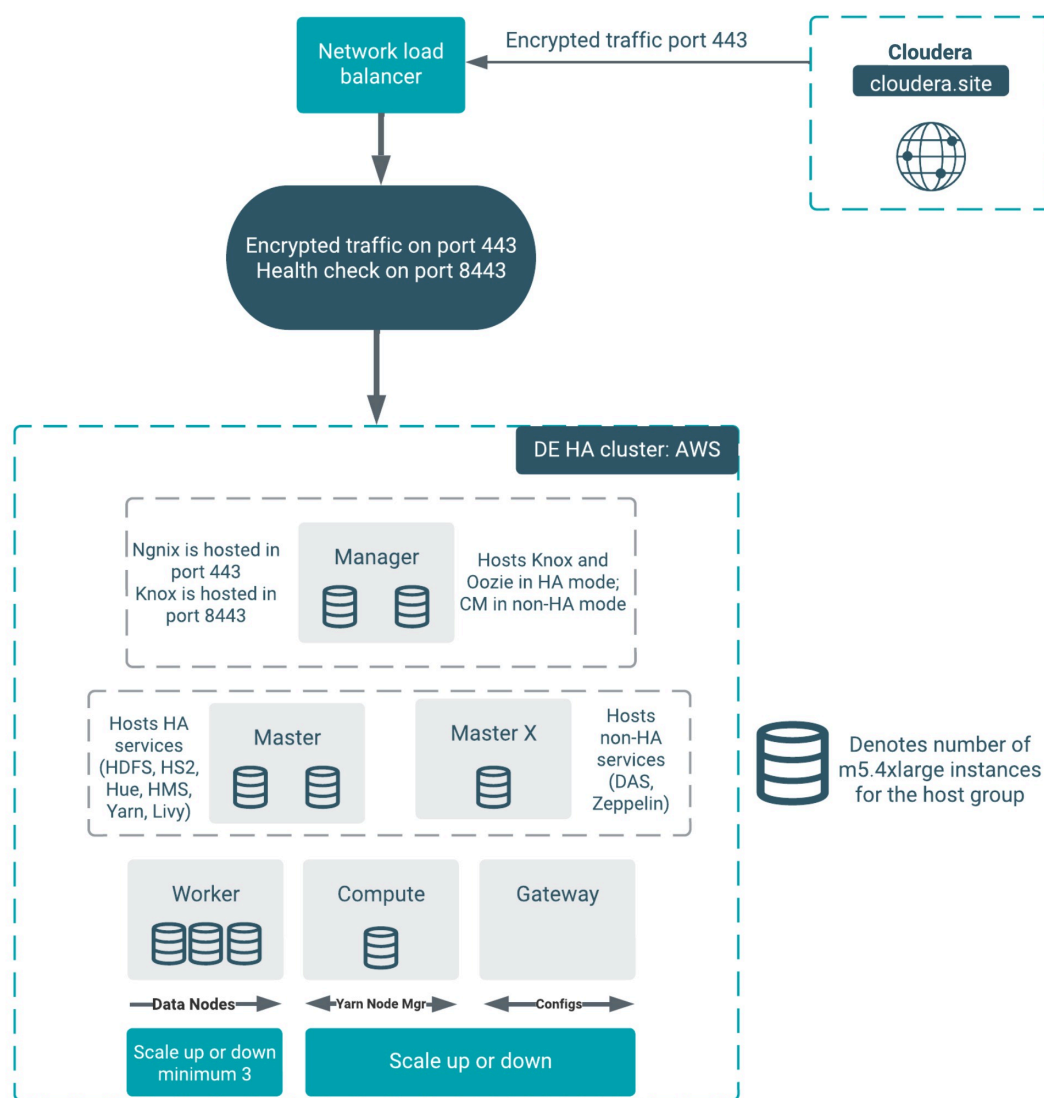
- There is a Hive Metastore Service (HMS) running in the cluster that talks to the same database instance as the Data Lake in the environment.
- If you use CLI to create the cluster, you can optionally pass an argument to create an external database for the cluster use such as Cloudera Manager, Oozie, 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.

For AWS

Architecture of the Data Engineering HA for AWS cluster

The Data Engineering HA for AWS 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 as of Cloudera Runtime version 7.2.11. If you create a custom template for DE HA, follow these two rules: <ol style="list-style-type: none"> 1. Oozie must be in single hostgroup. 2. Oozie and Hue must not be in the same hostgroup.



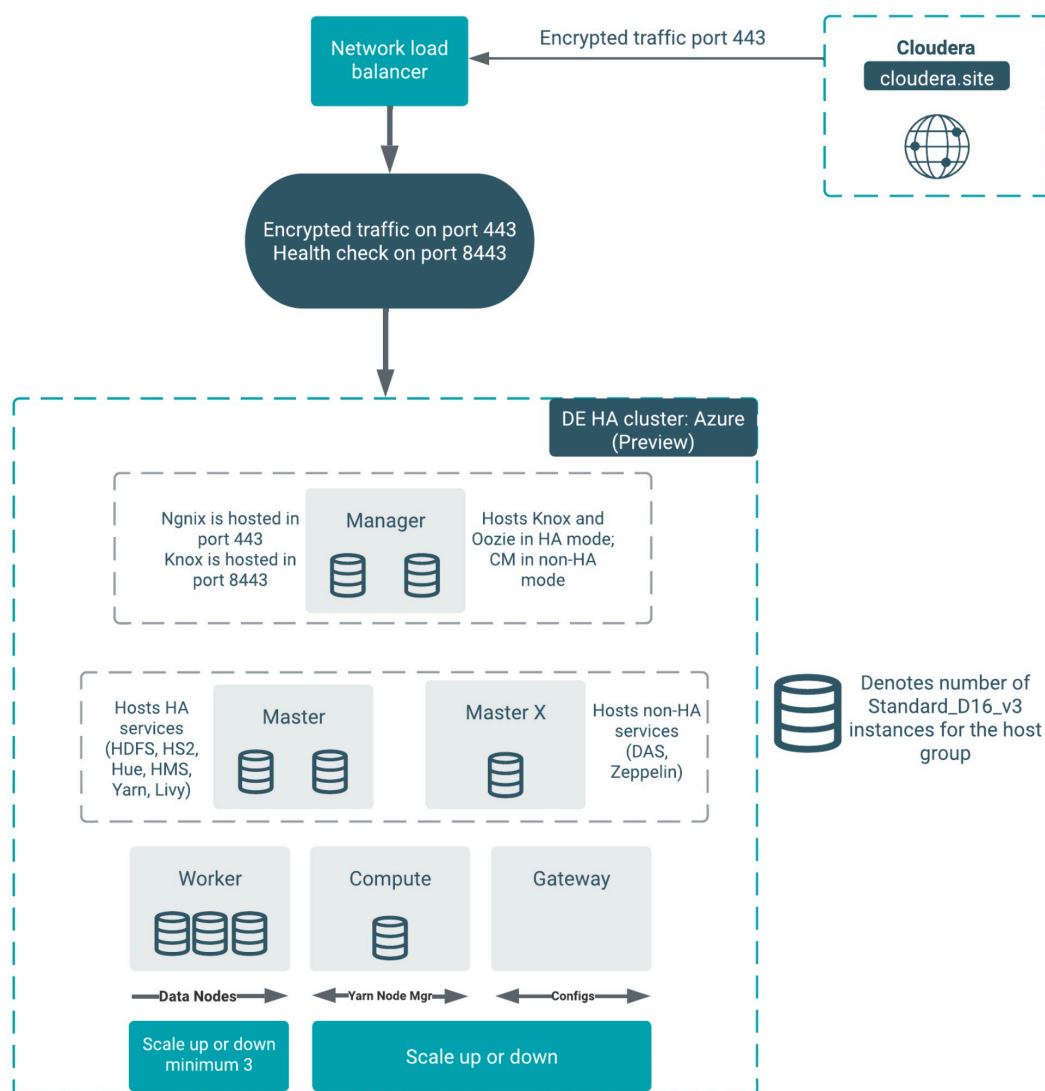
Important: If you are creating a DE HA cluster through the CDP CLI using the `create-aws-cluster` command, note that there is a CLI parameter to provision the network load-balancer in HA cluster shapes. Make sure to use the `--enable-load-balancer | --no-enable-load-balancer` parameter when provisioning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

For Azure

Architecture of the Data Engineering HA for Azure cluster

The Data Engineering HA for 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.



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 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-azure-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](#).

For GCP (preview)



Note: HA for Oozie is not yet available in the GCP template.

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](#)

[Livy](#)

[Oozie](#)

[Spark](#)

[YARN](#)

[Zeppelin](#)

[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
- Data Mart (ARM) for AWS
- Data Mart for Azure
- Data Mart for Google Cloud

Cluster template name

Cloudera - Data Mart: Apache Impala, Hue

Included services

- HDFS
- Hue
- Impala

Compatible Cloudera Runtime versions

- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18
- 7.3.1

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
- Real-time Data Mart for Azure
- Real-time Data Mart for Google Cloud

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

- 7.2.15
- 7.2.16
- 7.2.17

Cluster definition names

- Real-time Data Mart - Spark3 for AWS
- Real-time Data Mart - Spark3 (ARM) for AWS
- Real-time Data Mart - Spark3 for Azure
- Real-time Data Mart - Spark3 for Google Cloud

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
- 7.3.1

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

[Getting started with Operational Database](#)

[Creating an Operational Database cluster](#)

[Cloudera Operational Database](#)

[Creating a database using Cloudera Operational Database](#)

[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 Manager
- 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 on cloud clusters

Cluster definition names

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

- Streams Messaging HA for AWS
- Streams Messaging Heavy Duty (ARM) for AWS
- Streams Messaging Light Duty (ARM) for AWS
- Streams Messaging HA (ARM) for AWS
- Streams Messaging Heavy Duty for Azure
- Streams Messaging Light Duty for Azure
- Streams Messaging HA for Azure (Technical Preview)
- Streams Messaging Heavy Duty for GCP
- Streams Messaging Light Duty for GCP
- Streams Messaging HA for GCP (Technical Preview)

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

- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18
- 7.3.1

Related Information

[Setting up your Streams Messaging cluster](#)

[Authorizing Access to Flow Management Clusters in Cloudera on cloud](#)

[Ingesting Data into Cloudera on cloud](#)

[Ingesting Data into Cloudera on 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 Cloudera 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 on cloud
- Using NiFi and NiFi Registry

Cluster definition names

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

Cluster template name

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

Included services

- NiFi
- NiFi Registry

Compatible Cloudera Runtime versions

- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18
- 7.3.1

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 on 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 Light Duty for Azure
- Streaming Analytics Light Duty for GCP
- Streaming Analytics Heavy Duty for AWS
- Streaming Analytics Heavy Duty for Azure
- Streaming Analytics Heavy Duty for GCP

Cluster template name

- 7.3.1 - Streaming Analytics Light Duty
- 7.3.1 - 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 output 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

- 7.2.15
- 7.2.16
- 7.2.17
- 7.2.18
- 7.3.1

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
- Data Discovery and Exploration for AWS
- Data Discovery and Exploration for Azure

Cluster Template Name

- Data Discovery and Exploration

Included Services

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

Compatible Cloudera Runtime Versions

- 7.2.15
- 7.2.16
- 7.2.17

Cluster Definition Names

- Data Discovery and Exploration - Spark3 for AWS
- Data Discovery and Exploration (ARM) - 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
- 7.3.1

Related Information

[Solr](#)[Spark](#)[HDFS](#)[Hue](#)[YARN](#)[Zookeeper](#)

Saving a custom cluster definition

The easiest way to obtain and save a custom cluster definition is by providing parameters in the create cluster wizard and then generating a cluster definition on the last page of the wizard.

About this task

Required role: Any user who can create a Cloudera Data Hub cluster

Procedure

1. Log in to the Cloudera web interface.
2. In the Cloudera Management Console, navigate to Environments and then navigate to a specific environment.
3. Click on Create Data Hub.
4. In the cluster wizard, provide all parameters that you need.
5. Once done, click on Save as new definition.
6. Provide a name for the template.
7. Click on Save as cluster template.

Once registered, your custom cluster definition can be viewed by clicking Environments, then selecting an environment and clicking the Cluster Definitions tab. Note that the cluster definition is only available for use within the environment for which it was created.

What to do next

Now you can use the registered custom cluster definition to create Cloudera Data Hub clusters.


Deleting a cluster definition

You can delete a custom cluster definition. Default cluster definitions cannot be deleted.

About this task

Required role: Owner of the cluster definition

Procedure

1. Log in to the Cloudera web interface.
2. Navigate to the environment in which you created the definition, then click the Cluster Definitions tab.
3. Select one or more custom cluster definitions.
4. Click .
5. Click Yes to confirm delete.

After performing these steps, the selected cluster definition(s) will be deleted.

Managing cluster definitions from CLI

You can manage cluster definitions from CLI using `cdp datahub` commands.

- Register a cluster definition: `cdp datahub create-cluster-definition --cluster-definition-name <value> --workload-template <value>`
- List all available cluster definitions: `cdp datahub list-cluster-definitions`
- Describe a specific cluster definition: `cdp datahub describe-cluster-definition --cluster-definition-name <value>`
- Delete one or more existing cluster definitions: `cdp datahub delete-cluster-definitions --cluster-definition-names <value>`