

## Creating a Cluster on Azure

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

<b>Cluster deployment options.....</b>	<b>4</b>
<b>Default cluster configurations.....</b>	<b>4</b>
Data Engineering clusters.....	5
Data Mart clusters.....	9
Operational Database clusters.....	11
Streams Messaging clusters.....	12
Flow Management clusters.....	13
Streaming Analytics clusters.....	14
Data Discovery and Exploration clusters.....	15
<b>Creating a cluster from a definition on Azure.....</b>	<b>16</b>
<b>Creating a custom cluster on Azure.....</b>	<b>18</b>
<b>Create a cluster with an external database.....</b>	<b>19</b>
<b>Creating a cluster from an existing cluster.....</b>	<b>20</b>
<b>Creating a cluster with a specific Java version.....</b>	<b>21</b>
<b>Advanced cluster options.....</b>	<b>21</b>
Tags.....	22
Image catalog.....	22
Custom images and image catalogs.....	23
Creating a custom image and image catalog with the CDP CLI.....	24
Switching image catalogs.....	26
Network and availability.....	27
Hardware and storage.....	27
Cloud storage.....	28
Recipes.....	29
Cluster template overrides.....	29
Creating a cluster template override.....	31
Creating a Cloudera Data Hub cluster with a cluster template override.....	34
Cluster template override CLI commands.....	34
Custom properties.....	35
Cluster deployment logs.....	36
Enabling/disabling cluster deployment log collection.....	36

## 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.

### Related Information

[Default cluster configurations](#)

[Cluster Planning](#)

## 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 Azure
- Data Engineering HA - Spark3 for Azure
- Data Engineering Spark3 for Azure

#### 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."

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.

- 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."

#### 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 Azure

**Table 1: 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.	For clusters created with Cloudera Runtime versions lower than 7.2.14: <ul style="list-style-type: none"> <li>• Standard_D16_v3; StandardSSD_LRS - 100 GB</li> </ul> For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> <li>• Standard_D16_v3</li> </ul>
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"> <li>• Standard_D8_v3; StandardSSD_LRS - 100 GB</li> </ul> For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> <li>• DE and DE Spark3: <ul style="list-style-type: none"> <li>• Standard_D5_v2</li> </ul> </li> <li>• DE HA: <ul style="list-style-type: none"> <li>• Standard_D5_v2</li> </ul> </li> </ul>
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"> <li>• Standard_D8_v3; StandardSSD_LRS - 100 GB</li> </ul> For clusters created with Cloudera Runtime versions 7.2.14 or higher versions: <ul style="list-style-type: none"> <li>• DE and DE Spark3: <ul style="list-style-type: none"> <li>• Standard_D5_v2</li> </ul> </li> <li>• DE HA: <ul style="list-style-type: none"> <li>• Standard_D5_v2</li> </ul> </li> </ul> For Azure, the attached volume count for the compute host group is changed to 0. Only ephemeral/local volumes are used by default.   <b>Note:</b> 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.

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

## Configurations

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

**Table 2:**

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

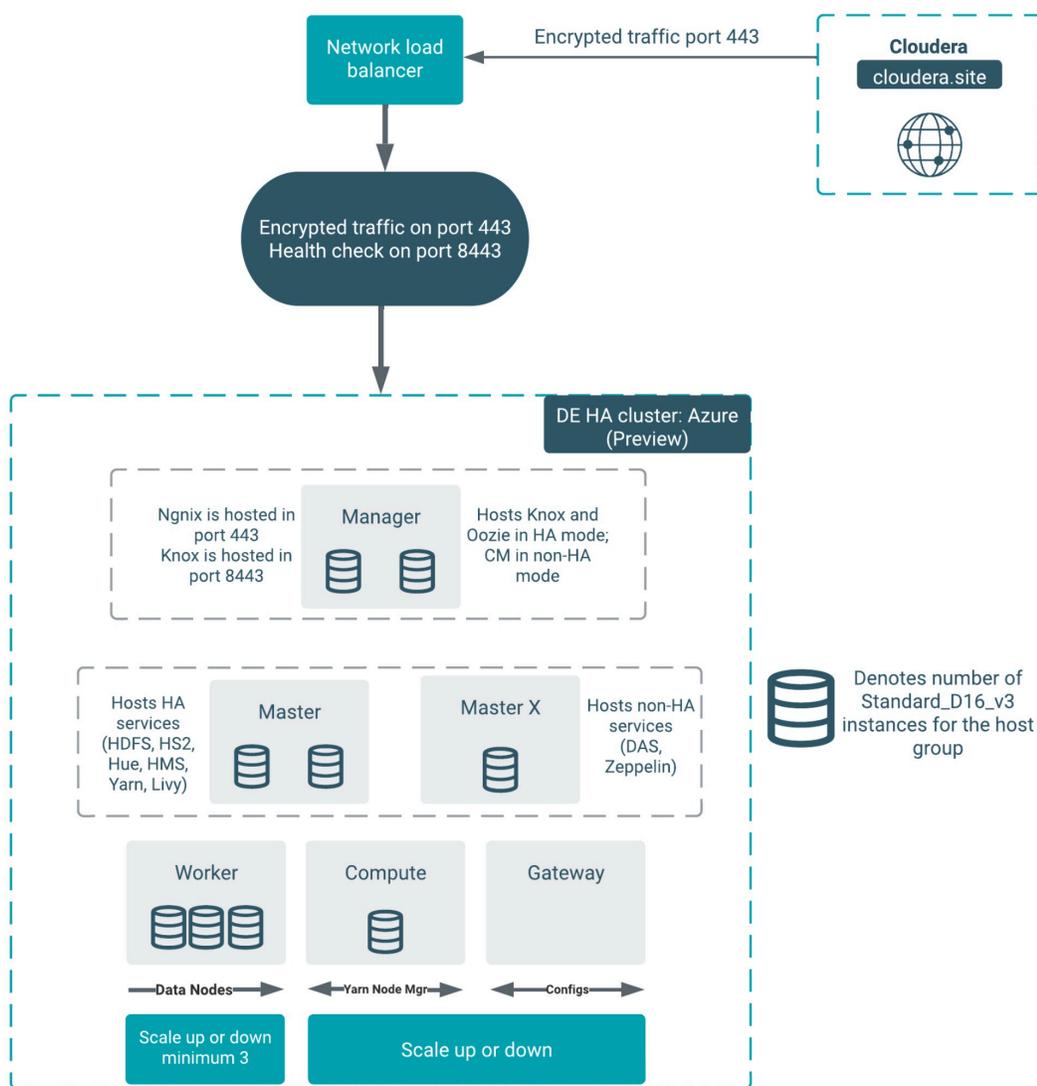
Note the following:

- 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 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](#).

### 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 Azure

#### 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 Azure

#### 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 Azure

**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

[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 Azure
- Streams Messaging Light Duty for Azure
- Streams Messaging HA for Azure (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](#)

[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 Azure
- Flow Management Heavy Duty for Azure

### 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 Azure
- Streaming Analytics Heavy Duty for Azure

### 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 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

#### Related Information

[Solr](#)

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

## Creating a cluster from a definition on Azure

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

### Before you begin

To create a Cloudera Data Hub cluster on Azure, you must have an existing Azure 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:

Selected Environment: aws demo1

Cluster Definition: CDP 1.0 - Data Mart template

Services: Hdfs 3.0.0, Hue 4.3.0, Impala 3.2.0

General Settings: Cluster Name: docs-test

Advanced Options: [Toggle]

Buttons: PROVISION CLUSTER, CREATE AS NEW DEFINITION, SHOW CLI COMMAND, SHOW GENERATED BLUEPRINT

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 Azure 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 <a href="#">Tags</a> .

- Optionally, click on Advanced Options to modify advanced cluster settings. For more information on these options, refer to *Advanced cluster options*.
- 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.
- 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.
- To proceed with cluster provisioning immediately, click on Provision Cluster.
- 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
```

- 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.

### Related Information

[Advanced cluster options](#)

[Accessing a Cluster](#)

## Creating a custom cluster on Azure

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

### Before you begin

To create a Cloudera Data Hub cluster on Azure, you must have an existing Azure 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.

<https://www.youtube.com/embed/pGHHW9nOrPU>

### 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:

**Services**

The following services will be installed as part of this cluster. This list is determined by the selected Cluster Template.

Platform Version  
 ⓘ  
 Platform Version should be the same as the Data Lake cluster's.

Cluster Template  
 ⓘ

 Hdfs 3.0.0     Hue 4.3.0     Impala 3.2.0

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 Azure 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 <a href="#">Tags</a> .

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.

### Related Information

[Advanced cluster options](#)

[Accessing a Cluster](#)

## Create a cluster with an external database

Through the CDP CLI, you can configure an external, durable database when you create a Data Hub cluster. An external database keeps a persistent state for cluster services like Hue, DAS and Zeppelin in case of instance failures. You also have the option to configure an external database as a custom property within a cluster definition.

A top-level custom property for the external managed database is available to be set within a cluster definition:

```
"externalDatabase": {
  "availabilityType": "NONE" | "NON_HA" | "HA"
```

For more information on using custom cluster definitions to create clusters, see *Cluster definitions*.

A new flag, called `--datahub-database`, is available in the CDP CLI `create-azure-cluster` command. The supported values are `NONE`, `NON_HA`, and `HA`. To create a Data Hub cluster with an external managed database, use the following CLI options:

```
create-azure-cluster
  [--cluster-name <value>]
  [--cluster-definition-name <value>]
  [--environment-name <value>]
  [--cluster-template-name <value>]
  [--instance-groups <value>]
  [--subnet-id <value>]
  [--image <value>]
  [--tags <value>]
  [--request-template <value>]
  [--datahub-database <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton]
```

- `--cluster-name`: String. The name of the cluster. This name must be unique, must have between 5 and 40 characters, and must contain only lowercase letters, numbers and hyphens. Names are case-sensitive.
- `--cluster-definition-name`: String. The name or CRN of the cluster definition to use for cluster creation.
- `--environment-name`: Name or CRN of the environment to use when creating the cluster. The environment must be an Azure environment.
- `--cluster-template-name`: String. Name or CRN of the cluster template to use for cluster creation.
- `--instance-groups`: Array. Instance group details.
- `--subnet-id`: String. The subnet ID.
- `--image`: Object. The details of the image used for cluster instances.
- `--tags`: Array. Tags to be added to Data Hub-related resources.
- `--request-template`: String. JSON template to use for cluster creation. This is different from the cluster template and would be removed in the future.
- `--datahub-database`: String. Database type for the Data Hub cluster. Currently supported values: `NONE`, `NON_HA`, `HA`
- `--cli-input-json`: String. Performs service operation based on the JSON string provided. The JSON string follows the format provided by `--generate-cli-skeleton`. If other arguments are provided on the command line, the CLI values will override the JSON-provided values.
- `--generate-cli-skeleton`: Prints a sample input JSON to standard output. Note the specified operation is not run if this argument is specified. The sample input can be used as an argument for `--cli-input-json`.

For information on shorthand and JSON syntax of these options, and the output of the command, invoke the CDP CLI `help`.



**Note:** The external managed database is destroyed along with the Data Hub cluster.

### Related Information

[Cluster definitions](#)

[Default cluster configurations](#)

[Cloudera DataFlow for Data Hub](#)

[Cloudera Runtime](#)

## Creating a cluster from an existing cluster

You can copy the CDP CLI command used to create a Cloudera Data Hub cluster on AWS or Azure and then use the command to create a new cluster with the same or similar settings.

### Procedure

1. Click Data Hub Clusters in the left navigation menu and then click the tile of the Cloudera Data Hub cluster that you want to use to create the new cluster.
2. From the Data Hub details page, click the Actions menu in the top right of the page and then Show CLI Command.
3. Copy the CDP CLI command used for the cluster creation.
4. Before you use the command in the CDP CLI to create a new cluster, ensure that you change the `--cluster-name` parameter as this must be a unique name.

### Results

You can now use the CDP CLI to run the command to create a new cluster.

### Related Information

[CDP CLI client setup](#)

## Creating a cluster with a specific Java version

When you create a new Cloudera Data Hub or Data Lake cluster, you can specify Java 8 or Java 11 as the default JDK.

### Java 11

When you launch a new Cloudera Data Hub or Data Lake cluster, the default JDK version is Java 8 unless you specify Java 11. If you specify Java 11 as the default JDK version, Cloudera Manager and all Cloudera Runtime services (as well as all workloads launched) will use pre-installed Java 11 dependencies rather than Java 8. This may be useful, for instance, when you have workloads such as Spark 3 jobs that are written in Java 11, when your workloads have Java 11 dependencies, or if you are running third-party components on a cluster that require Java 11.

The ability to launch new Cloudera Data Hub and Data Lake clusters with Java 11 is currently available only through the CDP CLI. Include the parameter `--java-version` in the `create-<cloud_provider>-datalake` and `create-<cloud_provider>-datahub` CLI commands. Valid inputs for the `--java-version` parameter are 8 and 11.

Launching clusters with Java 11 as the default is supported from Cloudera Runtime versions 7.2.15.11 and 7.2.16.200 and above. Creating clusters with the Java version property set to 11 on older Cloudera Runtime versions will be blocked by an internal validation. Currently, the Java version selection is only available when creating a new cluster.

### Java 8 and 11 compatibility

Java 11 introduces breaking changes, which could lead to incompatibility with your current workloads or custom/third-party Java-based applications deployed on Cloudera Data Hub clusters.

Cloudera recommends that all existing production workloads are tested before moving them over to a Cloudera Data Hub cluster that is set to use Java 11 as the default JDK.

Note that Data Lake and Cloudera Data Hub hosts will continue to have both Java 8 and 11 pre-installed. The `--java-version` switch controls which Java version should be set as the OS default during bootstrapping of the hosts. Therefore, it is possible to continue to run jobs written for Java 8 even if the OS default is Java 11. Some services (for example, Spark) make this possible by setting the `JAVA_PATH` environment variable to point to the Java 8 path `/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.[...].x86_64/jre/bin/java`.

## Advanced cluster options

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

<https://www.youtube.com/embed/Q8Xjb3-8KFk>

While some of these advanced options can be configured 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 Azure documentation.

### Related Information

[Use tags to organize your Azure resources \(Azure\)](#)

## 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.

### Related Information

[Image catalog updates](#)

## 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](https://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 name> --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-7torotmqi6q7438y --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: <ul style="list-style-type: none"> <li>• --provider</li> <li>• --image-type</li> <li>• --runtime-version</li> </ul> You can provide any combination of these parameters.
cdp imagecatalog create-custom-catalog	Creates a custom catalog.  Parameters: <ul style="list-style-type: none"> <li>• --catalog-name</li> <li>• --description</li> </ul>

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: <ul style="list-style-type: none"> <li>• --catalog-name</li> <li>• --vm-images [region,imageReference]</li> <li>• --source-image-id</li> <li>• --image-id</li> <li>• --base-parcel-url</li> </ul> --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.

Before you begin

- Ensure that both the image catalog that you are currently using and the image catalog that you want to switch to is JSON-based.
- Ensure that the currently used image is present in both image catalogs: the current one and the one that you want to change to.

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.
--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.
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 <a href="#">Using Azure Database for PostgreSQL Flexible Server</a> .

## 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 Azure refer to <a href="#">General Purpose Virtual Machine Sizes</a> in Azure documentation.
Instance Count	Enter the number of instances of a given type. Default is 1.
Storage Type	Select the volume type. The options are: (1) Locally-redundant storage, (2) Geo-redundant storage, (3) Premium locally-redundant storage. For more information about these options refer to <a href="#">Introduction to Azure Storage</a> in Azure documentation.
Attached Volumes Per Instance	Enter the number of volumes attached per instance. Default is 1.
Volume Size	Enter the size in GB for each volume. Default is 100.

Parameter	Description
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 change the value of the root volume size, an osDisk with the given rootVolumeSize will be created for the instance automatically; however, you will have to manually resize the osDisk partition by using the steps provided in <a href="#">How to Resize Linux osDisk Partition on Azure</a> in Azure documentation. If you use a custom Cloudera Data Hub template specifying a root volume size smaller than 200GB, you may encounter an error.
Availability Sets	<p>To support fault tolerance for VMs, Azure uses availability sets. This allows two or more VMs to be mapped to multiple fault domains, each of which defines a group of virtual machines that share a common power source and a network switch. When adding VMs to an availability set, Azure automatically assigns each VM a fault domain. The SLA includes guarantees that during OS Patching in Azure or during maintenance operations, at least one VM belonging to a given fault domain will be available.</p> <p>In Cloudera, no availability sets are configured by default. To have availability sets configured, make sure to navigate to the "Hardware and Storage" tab in the Advanced Options during Cloudera Data Hub cluster creation. In this case, one availability set is configured during cluster creation for each host group with "Instance Count" that is set to 1 or larger. The assignment of fault domains is automated by Azure, so there is no option for this in Cloudera web UI.</p> <p> <b>Note:</b> Although there is no specific option in the "Hardware and Storage" tab to enable availability sets, just navigating to this page will cause the availability sets to be configured. On that page, one can observe the availability set names present for each group. This and other availability set parameters for the availability sets cannot be configured.</p> <p>After the deployment is finished, you can check the layout of the VMs inside an availability set on Azure Portal. You will find the "Availability set" resources corresponding to the host groups inside the deployment's resource group.</p>
Availability Set Name	Choose a name for the availability set that will be created for the selected host group. The following convention is used: "clustername-hostgroupname-as".
Fault Domain Count	The number of fault domains. Default is 2 or 3, depending on the setting supported by Azure.
Update Domain Count	This number of update domains. This can be set to a number in range of 2-20. Default is 20.

## 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 ADLS Gen2 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 ADLS Gen2 location. Note that the specified ADLS Gen2 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.

### Related Information

[Minimal setup for cloud storage](#)

## Recipes

The "Recipes" option allows you to select previously uploaded recipes (scripts that can be run pre or post cluster deployment) for each host group.

This option is available from the Cluster Extensions page of the advanced create cluster wizard. For more information, refer to the documentation linked below.

### Related Information

[Recipes](#)

## Cluster template overrides

You can specify custom configurations that override or append the properties in a built-in Cloudera Data Hub template or a custom template.

### Overview

You can launch Cloudera Data Hub clusters from a set of pre-defined cluster templates created for prescriptive use cases. These cluster templates are a “shared resource” that define the list of services that will be installed on the Cloudera Data Hub, including their configurations.

For example, this is a portion of a default template:

```
{
  "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_namenode_java_heapsize_minimum_validator",
                "value": "true"
            },
            {
                "name": "role_config_suppression_fs_trash_interval_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
            }
        ]
    }
}

```

]

In this section of an example cluster template, two primary types of service configurations are visible: `serviceConfigs` and configurations for various `roleConfigGroups`, in addition to a special type of service configuration called `safety_valves`. You can read about the details of these configs in the *Cloudera Manager Configuration Properties* reference.

Oftentimes you may want to modify the `serviceConfigs`, `roleConfigGroups`, and `safety_valves` present in a default Cloudera Data Hub cluster template. Although you can create a custom cluster template by modifying the JSON of a default template, this process can be cumbersome and error-prone.

As an alternative to creating a custom template, you can specify custom configurations that override or append the properties in a default template. These custom configurations are saved as a shared resource called “cluster template overrides,” and can be used and re-used across Cloudera Data Hub clusters in different environments. As a shared resource, they do not need to be attached to a specific Cloudera Runtime version.

By using a default cluster template along with a cluster template override, you can create a customized Cloudera Data Hub cluster, along with receiving improvements to the default templates that are present in newer Cloudera Runtime releases. Cluster template overrides can be used to override or append properties present in both the identified types of service configurations, as well as safety valves.



**Important:** If you customize safety valves, the provided value will be appended to the already existing value for that safety valve. Custom configs will not override the existing value.

You are not limited to customizing the configs present in a default cluster template; you can add any valid configuration recognized by Cloudera Manager for an included service. New configurations are appended to the cluster template when the cluster is launched.

Cluster template overrides can be applied to custom templates as well. If you want to apply a cluster template override to a custom template that contains properties that are dynamically replaced during cluster creation, the cluster template override will override any dynamically-replaced properties when the two conflict.

### Limitations

At present, there is no way to validate any individual property to ensure that it is valid and recognised by Cloudera Manager, so that it can be overridden to the desired value. Adding an incorrect config name can lead to errors while installing the Cloudera Manager template on the Cloudera Data Hub cluster, or the property could be ignored by Cloudera Manager entirely.

An invalid value for a particular property can also cause errors during the cluster creation process. Carefully review the *Cloudera Manager Configuration Properties* reference in regards to configurations that you want to customize.

Cluster template overrides are only for overriding/appending the `serviceConfigs`, `roleConfigGroups`, and `safety_valves` in a Cloudera Data Hub template. Unlike creating a custom template, you can not add a service to the template.

### Related Information

[Cloudera Manager Configuration Properties](#)

[Building a custom Cloudera Data Hub cluster template](#)

[Custom Properties](#)

## Creating a cluster template override

From the Shared Resources menu, you can create a cluster template override to overwrite the default values of service and role configs in a Cloudera Data Hub template.

### Procedure

1. In the Cloudera Management Console UI, Shared Resources Cluster Template Overrides .

- 2. Click Create Custom Override.

## Cluster Template Overrides / Create

**Name \***

 ?

Cloudera Runtime Version

Please select runtime version ▼

---

Service Config     Role Config

**Config name/path \***

 ?

**Config value \***

 ?

**Service type \***

 ?

---

Service Type	Role Type
No cluster template value override added.	

## 3. Provide the following information:

Property	Description
Name	Provide a unique name for the override.
Cloudera Runtime Version	Select a specific Cloudera Runtime version to which this override applies, or select "Not Runtime specific."
Service Config / Role Config	Use the radio buttons to select whether you are customizing a service config or a role config.
Config Name/Path	Exact name/path of the config are you customizing.
Config Value	New value for the config. Ensure the value is valid.
Service Type	From the drop-down menu, select the service type for the service or role config that you are customizing.
Role Type	For role configurations: the role type as it appears in the cluster template that you will apply the override to (or any valid role configuration that is recognized by Cloudera Manager).

For example:

Service Config  Role Config

**Config name/path \***

oom\_heap\_dump\_enabled ?

**Config value \***

false ?

**Service type \***

HIVE ▼

**Role type \***

HIVEMETASTORE ▼

4. Click Add and then repeat the previous step (excluding the Name and Cloudera Runtime Version) for each service or role config that you want to customize in this template override.

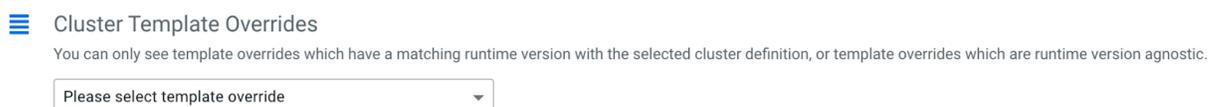
5. When you are finished adding service and role configs to the template override, click Save. The override appears in the list of saved overrides, accessible from the Shared Resources menu.

## Creating a Cloudera Data Hub cluster with a cluster template override

You can use a cluster template override that you previously created to provision a Cloudera Data Hub cluster.

### Procedure

1. Click Data Hub Clusters Create Data Hub Cluster .
2. Enter the information required to create your cluster, then at the bottom of the cluster creation wizard, select Advanced OptionsCluster Extensions.
3. From the Cluster Templates Overrides drop-down box, select the template override that you previously created and want to apply when the cluster is provisioned. The only visible overrides are those with a Cloudera Runtime version matching the selected cluster definition version, or those that are Runtime-agnostic.



4. Click Provision Cluster.

### Results

The Cloudera Data Hub cluster is provisioned using the service and role configs that you specified in the template override.

## Cluster template override CLI commands

You can use the CDP CLI to create cluster template overrides, manage the cluster template overrides, and provision a Data Hub cluster using a cluster template override.

**Table 3: Cluster Template Override CLI**

Command	Description	Example
<code>cdp datahub create-custom-configurations</code>	Creates a new cluster template override shared resource with an indefinite amount of properties.	<pre>cdp datahub create-c ustom-configurations --custom-configuration s-name "test-custom- configs" --configurations configName="name",co nfigValue="value",se rviceType="hdfs" configName="name2",conf igValue="value2",rol eType="role2",servic eType="hive_on_tez" ... --runtime-version "7.2. 8"</pre>
<code>cdp datahub list-custom-configurations</code>	Lists all the cluster template overrides for the account.	
<code>cdp datahub describe-custom-configurations</code>	Returns the name, CRN, the properties, and the platform version associated with the cluster template override resource.	<pre>cdp datahub describe- custom-configurations --custom-configurations "test-overrides"</pre>
<code>cdp datahub delete-custom-configurations</code>	Deletes multiple custom configs given their names or CRNs.	<pre>cdp datahub delete-c ustom-configurations -custom-configurations "test-override" "test- override-2" ...</pre>
<code>cdp datahub create-aws-cluster</code> <code>cdp datahub create-azure-cluster</code> <code>cdp datahub create-gcp-cluster</code>	Launches a Cloudera Data Hub cluster with custom configs from a cluster template override. The parameter for specifying a cluster template override is: <code>--custom-configurations-name</code>	<pre>cdp datahub create-aws- cluster --cluster-name "de-dh1" --environment-name "l ocal-aws-env" --cluster-template-nam e "7.2.8 - Data Engi neering: Apache Spark, Apache Hive, Apache O ozie" --cluster-definition- name "7.2.8 - Data En gineering for AWS" --custom-configuratio ns-name "test-override"</pre>

## Custom properties

The "Custom Properties" option allows you to set properties on a per-cluster basis.

This option is available from the Cluster Extensions page of the advanced create cluster wizard. For more information, refer to the documentation linked below.

## Related Information

[Custom Properties](#)

## Cluster deployment logs

Cluster deployment logs gather the same diagnostic (system and service) logs that are collected into your cloud storage location, but the Cloudera platform sends the logs to Cloudera engineering and support for troubleshooting purposes. You can apply configurable redaction rules on any sensitive data.

Cluster deployment logs are disabled by default. You can change this setting during environment creation or after environment creation, though the Cloudera platform will only collect logs for new deployments. You can also set a default behavior at the Cloudera account level, so that you won't have to enable the setting every time in the wizard. The logs are only gathered during deployment, which is approximately the first 20 minutes of cluster creation.

You can configure anonymization rules for log collection at the Cloudera account level. The rules are a list of rule objects with two fields: a regex pattern (PCRE) as the value, and a replacement string. Configure these rules to hide any sensitive data. By default, the rules hide email addresses and card numbers. This default behavior is the same as Cloudera Manager diagnostics collection.

The anonymization options are available under Global Resources Telemetry:

### Deployments Cluster Logs Collection

Collect diagnostic logs during the initial phase of deployments for troubleshooting purposes.

Enable Cluster Logs Collection

---

#### Anonymization rules

Pattern	Replacement	
<code>\d{4}[\w]\d{4}[\w]\d{4}[\w]\d{4}</code>	XXXX-XXXX-XXXX-XXXX	<a href="#">Remove</a>
<code>\d{3}[\w]\d{2}[\w]\d{4}</code>	XXX-XX-XXXX	<a href="#">Remove</a>
<code>\b([A-Za-z0-9][A-Za-z0-9][A-Za-z0-9\-\.\_]*[A-Za-z0-9])@((([A-Za-z0-9][A-</code>	email@redacted.host	<a href="#">Remove</a>

[New rule](#) [Set defaults](#)

---

#### Test rules

Input text

1234-1234-1234-1234 info@cloudera.com 123-12-1234

Anonymized result

[Test](#) [Test with](#)

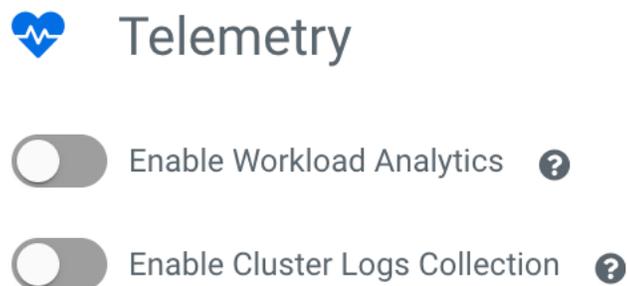
Currently, cluster deployment logs are collected only for Data Lake and Cloudera Data Hub cluster shapes.

## Enabling/disabling cluster deployment log collection

You can enable cluster deployment log collection during environment creation. You can also enable or disable it after environment creation.

To enable cluster deployment log collection during environment creation:

1. Click the slider button under **Telemetry** in the **Region, Networking, Security, and Storage** page of environment registration.



2. Select the checkbox to consent to data collection and click Enable.

To enable or disable cluster log deployment collection after environment creation:

1. Navigate to the **Environments** interface.
2. Select the desired environment, then click the Summary tab.
3. Scroll down to **Telemetry** and click the Edit button.
4. Click the slider button under Enable Deployment Cluster Logs Collection.
5. Select the checkbox to consent to data collection and click Enable.

To enable or disable cluster log deployment collection at the Cloudera account level (globally):

1. From the left navigation menu, click Global Settings and then Telemetry.
2. Click the slider button under Deployment Cluster Logs Collection.
3. Select the checkbox to consent to data collection and click Enable.