

Cloudera Manager 7.0.1

Managing Data Hub Clusters

Date published: 2020-02-11

Date modified:

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. 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

Accessing the Cloudera Manager Admin Console from Data Hub clusters.....	5
Accessing the Cloudera Manager for Data Lake clusters using workload credentials.....	5
Managing Hosts.....	6
Status.....	6
Configuration.....	7
Roles.....	7
Host Templates.....	7
Stopping All the Roles on a Host.....	7
Starting All the Roles on a Host.....	8
Performing Maintenance on a Cluster Host.....	8
Decommissioning Hosts.....	8
Recommissioning Hosts.....	9
Tuning and Troubleshooting Host Decommissioning.....	10
Tuning HDFS Prior to Decommissioning DataNodes.....	10
Tuning HBase Prior to Decommissioning DataNodes.....	11
Performance Considerations.....	12
Troubleshooting Performance of Decommissioning.....	12
Maintenance Mode.....	13
Entering Maintenance Mode.....	14
Exiting Maintenance Mode.....	15
Viewing the Maintenance Mode Status of a Cluster.....	16
Managing Roles.....	16
Role Instances.....	17
Starting, Stopping, and Restarting Role Instances.....	17
Decommissioning Role Instances.....	17
Recommissioning Role Instances.....	18
Configuring Roles to Use a Custom Garbage Collection Parameter.....	18
Role Groups.....	19
Creating a Role Group.....	19
Managing Role Groups.....	20
Managing Cloudera Runtime Services.....	20
Starting a Cloudera Runtime Service on All Hosts.....	21
Stopping a Cloudera Runtime Service on All Hosts.....	21
Restarting a Cloudera Runtime Service.....	22
Rolling Restart.....	22
Aborting a Pending Command.....	24

Managing Cloudera Manager.....	25
Automatic Logout.....	25
Starting, Stopping, and Restarting the Cloudera Manager Server.....	26
Managing Cloudera Manager Server Logs.....	26
Viewing the Cloudera Manager Server Logs.....	27
Setting the Cloudera Manager Server Log Location.....	27
Configuring Cloudera Manager.....	27
Managing Anonymous Usage Data Collection.....	28
Diagnostic Data Collection.....	28
Manually Triggering Collection and Transfer of Diagnostic Data.....	29
Cloudera Manager Agents.....	30
Starting, Stopping, and Restarting Cloudera Manager Agents.....	30
Managing the Cloudera Manager Agent Logs.....	31
Viewing the Cloudera Manager Agent Logs.....	32
Setting the Cloudera Manager Agent Log Location.....	32
Default User Roles.....	32
Accessing Storage Using Amazon S3.....	35
Referencing S3 Credentials for YARN, MapReduce, or Spark Clients.....	35
Referencing Amazon S3 in URIs.....	37
Using Fast Upload with Amazon S3.....	37
Enabling Fast Upload using Cloudera Manager.....	38
How to Configure a MapReduce Job to Access S3 with an HDFS Credstore.....	38
Importing Data into Amazon S3 Using Sqoop.....	39
Authentication.....	40
Sqoop Import into Amazon S3.....	41

Accessing the Cloudera Manager Admin Console from Data Hub clusters

After you create a Data Hub cluster using the Cloudera Management Console, you can access the Cloudera Manager Admin Console to manage, configure, and monitor the cluster and its Cloudera Runtime services.

About this task

To access the Cloudera Manager Admin Console:

Procedure

1. Open the Cloudera Management Console.
2. Click the Data Hub Clusters service.
3. Click the name of the Data Hub cluster you want to manage.
The cluster details page displays.
4. Click the URL for Cloudera Manager.

Results

The Cloudera Manager Admin Console opens in a new browser tab. You do not need to login to the Cloudera Manager Admin Console.

Accessing the Cloudera Manager for Data Lake clusters using workload credentials

When you access Cloudera Manager from a Data Lake cluster through the Management Console, the system authenticates you using SSO. If you want to use credentials instead of SSO login to log in to the Cloudera manager then you can use workload credentials. To log in to the Cloudera Manager using workload credentials, you must use a different URL.

About this task

To log in to the Cloudera Manager using workload credentials:

Before you begin

You must have the IP address of the host on which Cloudera Manager is running.

Procedure

1. Open a web browser and specify the following URL in the address bar:

```
https://[***CM-HOST-IP-ADDRESS**]/clouderamanager/
```



Important: You must use the Cloudera Manager server IP address in the above URL.

2. Enter your workload user name and password.
3. Click Sign In.

Results

The Cloudera Manager Admin Console opens for the Data Lake cluster.

Managing Hosts

Cloudera Manager provides a number of features that let you configure and manage the hosts in your clusters.

The **Hosts** page has the following sections:

Status

You can view summary information about the hosts managed by Cloudera Manager. You can view information for all hosts, the hosts in a cluster, or individual hosts.

Viewing All Hosts

To display summary information about all the hosts managed by Cloudera Manager, click Hosts>All Hosts in the left menu. The **All Hosts** page displays with a list of all the hosts managed by Cloudera Manager.

The list of hosts shows the overall status of the Cloudera Manager-managed hosts in your cluster.

- The information provided varies depending on which columns are selected. To change the columns, click the Columns: *n* Selected drop-down and select the checkboxes next to the columns to display.
- Click ▶ to the left of the number of roles to list all the role instances running on that host.
- Filter the hosts list by entering search terms (hostname, IP address, or role) in the search box separated by commas or spaces. Use quotes for exact matches (for example, strings that contain spaces, such as a role name) and brackets to search for ranges. Hosts that match any of the search terms are displayed. For example:

```
hostname[1-3], hostname8 hostname9, "hostname.example.com"
hostname.example.com "HDFS DataNode"
```

- You can also search for hosts by selecting a value from the facets in the Filters section at the left of the page.
- If the agent heartbeat and health status properties are configured as follows:
 - Send Agent heartbeat every *x*
 - Set health status to Concerning if the Agent heartbeats fail *y*
 - Set health status to Bad if the Agent heartbeats fail *z*

The value *v* for a host's Last Heartbeat facet is computed as follows:

- $v < x * y = \text{Good}$
- $v \geq x * y$ and $\leq x * z = \text{Concerning}$
- $v \geq x * z = \text{Bad}$

Viewing the Hosts in a Cluster

Do one of the following:

- Select Clusters *Cluster name* Hosts .
- In the Home screen, click  Hosts in a full form cluster table.

The **All Hosts** page displays with a list of the hosts filtered by the cluster name.

Viewing Individual Hosts

You can view detailed information about an individual host—resources (CPU/memory/storage) used and available, which processes it is running, details about the host agent, and much more—by clicking a host link on the **All Hosts** page.

Configuration

The Configuration tab lets you set properties related to parcels and to resource management, and also monitoring properties for the hosts under management.

The configuration settings you make here will affect all your managed hosts. You can also configure properties for individual hosts by clicking on the host in the **All Hosts** page, which will override the global properties set here).

To edit the default configuration properties for hosts, click the Configuration tab.

Related Information

[Modifying Configuration Properties Using Cloudera Manager](#)

Roles

Role Assignments

You can view the assignment of roles to hosts as follows:

1. In the left menu, click HostsRoles.
2. Click a cluster name or All Clusters.

Disks Overview

In the left menu, click HostsDisks Overview to display an overview of the status of all disks in the deployment. The statistics exposed match or build on those in iostat, and are shown in a series of histograms that by default cover every physical disk in the system.

Adjust the endpoints of the time line to see the statistics for different time periods. Specify a filter in the box to limit the displayed data. For example, to see the disks for a single rack rack1, set the filter to: logicalPartition = false and rackId = "rack1" and click Filter. Click a histogram to drill down and identify outliers. Mouse over the graph and click  to display additional information about the chart.

Host Templates

The **Host Templates** page lets you create and manage host templates, which provide a way to specify a set of role configurations that should be applied to a host.

This greatly simplifies the process of adding new hosts, because it lets you specify the configuration for multiple roles on a host in a single step, and then (optionally) start all those roles.

To access the **Host Templates** page, click HostsHost Templates in the left menu.

Stopping All the Roles on a Host

You can stop all of the roles on a host from the **Hosts** page.

About this task

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Procedure

1. In the left menu, click ClustersHosts or HostsAll Hosts.
2. Select one or more hosts on which to stop all roles.

3. Select Actions for SelectedStop Roles on Hosts.

Starting All the Roles on a Host

You can start all the roles on a host from the **Hosts** page.

About this task

Minimum Required Role: **Operator** (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Procedure

1. Click the Hosts tab.
2. Select one or more hosts on which to start all roles.
3. Select Actions for SelectedStart Roles on Hosts.

Performing Maintenance on a Cluster Host

You can perform minor maintenance on cluster hosts by using Cloudera Manager to manage the host decommission and recommission process.

In this process, you can specify whether to suppress alerts from the decommissioned host and, for hosts running the DataNode role, you can specify whether or not to replicate under-replicated data blocks to other DataNodes to maintain the cluster's replication factor. This feature is useful when performing minor maintenance on cluster hosts, such as adding memory or changing network cards or cables where the maintenance window is expected to be short and the extra cluster resources consumed by replicating missing blocks is undesirable.

You can also place hosts into Maintenance Mode, which suppresses unneeded alerts during a maintenance window but does not decommission the hosts.

To perform host maintenance on cluster hosts:

Decommissioning Hosts

Cloudera Manager manages the host decommission and recommission process and allows you the option to specify whether to replicate the data to other DataNodes, and whether or not to suppress alerts.

About this task

Decommissioning a host decommissions and stops all roles on the host without requiring you to individually decommission the roles on each service. Decommissioning applies to only to HDFS DataNode, MapReduce TaskTracker, YARN NodeManager, and HBase RegionServer roles. If the host has other roles running on it, those roles are stopped.



Note: Hosts with DataNodes and DataNode roles themselves can only be decommissioned if the resulting action leaves enough DataNodes commissioned to maintain the configured HDFS replication factor (by default 3). If you attempt to decommission a DataNode or a host with a DataNode in such situations, the decommission process will not complete and must be aborted.

Before you begin

Minimum Required Role: **Limited Operator** (also provided by Operator, Configurator, Cluster Administrator, Limited Cluster Administrator, or Full Administrator).

Procedure

To decommission one or more hosts:

1. In Cloudera Manager, select the cluster where you want to decommission hosts.
2. In the left menu, click Hosts>All Hosts.
3. Select the hosts that you want to decommission.
4. Select Actions for Selected>Begin Maintenance (Suppress Alerts/Decommission).

(If you are logged in as a user with the Limited Operator or Operator role, the menu item is labeled Decommission Host(s) and you will not see the option to suppress alerts.)

The Begin Maintenance (Suppress Alerts/Decommission) dialog box opens. The role instances running on the hosts display at the top.

5. To decommission the hosts and suppress alerts, select Decommission Host(s). When you select this option for hosts running a DataNode role, choose one of the following (if the host is not running a DataNode role, you will only see the Decommission Host(s) option:):

- Decommission DataNodes

This option re-replicates data to other DataNodes in the cluster according to the configured replication factor. Depending on the amount of data and other factors, this can take a significant amount of time and uses a great deal of network bandwidth. This option is appropriate when replacing disks, repurposing hosts for non-HDFS use, or permanently retiring hardware.

- Take DataNode Offline

This option does not re-replicate HDFS data to other DataNodes until the amount of time you specify has passed, making it less disruptive to active workloads. After this time has passed, the DataNode is automatically recommissioned, but the DataNode role is not started. This option is appropriate for short-term maintenance tasks such as not involving disks, such as rebooting, CPU/RAM upgrades, or switching network cables.



Caution: Taking multiple DataNodes offline simultaneously increases the chances that some HDFS data may become unavailable during maintenance. Configuring the proper value for the Maintenance State Minimal Block Replication HDFS configuration property will avoid risking data availability.

6. Click Begin Maintenance.

The Host Decommission Command dialog box opens and displays the progress of the command.

Results



Note:

- You cannot start roles on a decommissioned host.
- When a DataNode is decommissioned, although HDFS data is replicated to other DataNodes, local files containing the original data blocks are not automatically removed from the storage directories on the host. If you want to permanently remove these files from the host to reclaim disk space, you must do so manually.

What to do next

Perform the necessary maintenance on the hosts.

Recommissioning Hosts

About this task

Only hosts that are decommissioned using Cloudera Manager can be recommissioned.

Before you begin

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Procedure

1. In Cloudera Manager, select the cluster where you want to recommission hosts.
2. In the left menu, click Hosts>All Hosts.
3. Select the hosts that you want to recommission.
4. Select Actions for SelectedEnd Maintenance (Suppress Alerts/Decommission). The End Maintenance (Suppress Alerts/Decommission dialog box opens. The role instances running on the hosts display at the top.
5. To recommission the hosts, select Recommission Host(s).
6. Choose one of the following:
 - Bring hosts online and start all roles
All decommissioned roles will be recommissioned and started. HDFS DataNodes will be started first and brought online before decommissioning to avoid excess replication.
 - Bring hosts online
All decommissioned roles will be recommissioned but remain stopped. You can [restart the roles](#) later.
7. Click End Maintenance.

Results

The Recommission Hosts and Start Roles Command dialog box opens and displays the progress of recommissioning the hosts and restarting the roles

Tuning and Troubleshooting Host Decommissioning

Decommissioning a host decommissions and stops all roles on the host without requiring you to individually decommission the roles on each service. The decommissioning process can take a long time and uses a great deal of cluster resources, including network bandwidth. You can tune the decommissioning process to improve performance and mitigate the performance impact on the cluster.

You can use the Decommission and Recommission features to perform minor maintenance on cluster hosts using Cloudera Manager to manage the process.

Tuning HDFS Prior to Decommissioning DataNodes

When a DataNode is decommissioned, the NameNode ensures that every block from the DataNode will still be available across the cluster as dictated by the replication factor. This procedure involves copying blocks from the DataNode in small batches. If a DataNode has thousands of blocks, decommissioning can take several hours. Before decommissioning hosts with DataNodes, you should first tune HDFS:

About this task

Minimum Required Role: [Configurator](#) (also provided by Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Procedure

1. Run the following command to identify any problems in the HDFS file system:

```
hdfs fsck / -list-corruptfileblocks -openforwrite -files -blocks -locations 2>&1 > /tmp/hdfs-fsck.txt
```

2. Fix any issues reported by the fsck command. If the command output lists corrupted files, use the fsck command to move them to the lost+found directory or delete them:

```
hdfs fsck file_name -move
```

or

```
hdfs fsck file_name -delete
```

3. Raise the heap size of the DataNodes. DataNodes should be configured with at least 4 GB heap size to allow for the increase in iterations and max streams.

- a) Go to the HDFS service page.
- b) Click the Configuration tab.
- c) Select ScopeDataNode.
- d) Select CategoryResource Management.
- e) Set the Java Heap Size of DataNode in Bytes property as recommended.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group.

4. Increase the replication work multiplier per iteration to a larger number (the default is 2, however 10 is recommended).

- a) Select ScopeNameNode.
- b) Expand the CategoryAdvanced category.
- c) Configure the Replication Work Multiplier Per Iteration property to a value such as 10.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group.

d)

5. Increase the replication maximum threads and maximum replication thread hard limits.

- a) Select ScopeNameNode.
- b) Expand the CategoryAdvanced category.
- c) Configure the Maximum number of replication threads on a DataNode and Hard limit on the number of replication threads on a DataNode properties to 50 and 100 respectively. You can decrease the number of threads (or use the default values) to minimize the impact of decommissioning on the cluster, but the trade off is that decommissioning will take longer.

To apply this configuration property to other role groups as needed, edit the value for the appropriate role group.

d)

6. Restart the HDFS service.

Related Information

[Performance Considerations](#)

[Modifying Configuration Properties Using Cloudera Manager](#)

Tuning HBase Prior to Decommissioning DataNodes

To increase the speed of a rolling restart of the HBase service, set the Region Mover Threads property to a higher value.

Minimum Required Role: [Configurator](#) (also provided by Cluster Administrator, Limited Cluster Administrator, and Full Administrator)

This increases the number of regions that can be moved in parallel, but places additional strain on the HMaster. In most cases, Region Mover Threads should be set to 5 or lower.

Performance Considerations

Decommissioning a DataNode does not happen instantly because the process requires replication of a potentially large number of blocks. During decommissioning, the performance of your cluster may be impacted.

This section describes the decommissioning process and suggests solutions for several common performance issues.

Decommissioning occurs in two steps:

1. The Commission State of the DataNode is marked as Decommissioning and the data is replicated from this node to other available nodes. Until all blocks are replicated, the node remains in a Decommissioning state. You can view this state from the NameNode Web UI. (Go to the HDFS service and select Web UI/NameNode Web UI.)
2. When all data blocks are replicated to other nodes, the node is marked as Decommissioned.

Decommissioning can impact performance in the following ways:

- There must be enough disk space on the other active DataNodes for the data to be replicated. After decommissioning, the remaining active DataNodes have more blocks and therefore decommissioning these DataNodes in the future may take more time.
- There will be increased network traffic and disk I/O while the data blocks are replicated.
- Data balance and data locality can be affected, which can lead to a decrease in performance of any running or submitted jobs.
- Decommissioning a large numbers of DataNodes at the same time can decrease performance.
- If you are decommissioning a minority of the DataNodes, the speed of data reads from these nodes limits the performance of decommissioning because decommissioning maxes out network bandwidth when reading data blocks from the DataNode and spreads the bandwidth used to replicate the blocks among other DataNodes in the cluster. To avoid performance impacts in the cluster, Cloudera recommends that you only decommission a minority of the DataNodes at the same time.
- You can decrease the number of replication threads to decrease the performance impact of the replications, but this will cause the decommissioning process to take longer to complete.

Cloudera recommends that you add DataNodes and decommission DataNodes in parallel, in smaller groups. For example, if the replication factor is 3, then you should add two DataNodes and decommission two DataNodes at the same time.

Related Information

[Tuning HDFS Prior to Decommissioning DataNodes](#)

Troubleshooting Performance of Decommissioning

Several conditions can impact performance when you decommission DataNodes.

Open Files

Write operations on the DataNode do not involve the NameNode. If there are blocks associated with open files located on a DataNode, they are not relocated until the file is closed. This commonly occurs with:

- Clusters using HBase
- Open Flume files
- Long running tasks

To find open files, run the following command:

```
hdfs dfsadmin -listOpenFiles -blockingDecommission
```

The command returns output similar to the following example:

```
Client Host      Client Name      Open File Path
172.26.12.77    DFSCClient_NONMAPREDUCE_-698274460_1 /hbase/ol
dWALs/dn3.cloudera.com%2C22101%2C1540973344249.dn3.cloudera.com%
2C22101%2C1540973344249.regiongroup-0.154099857098
```

After you find the open files, perform the appropriate action to restart process to close the file. For example, major compaction closes all files in a region for HBase.

Alternatively, you may evict writers to those decommissioning DataNodes with the following command:

```
hdfs dfsadmin -evictWriters <datanode_host:ipc_port>
```

For example:

```
hdfs dfsadmin -evictWriters datanode1:20001
```

A block cannot be relocated because there are not enough DataNodes to satisfy the block placement policy.

For example, for a 10 node cluster, if the `mapred.submit.replication` is set to the default of 10 while attempting to decommission one DataNode, there will be difficulties relocating blocks that are associated with map/reduce jobs. This condition will lead to errors in the NameNode logs similar to the following:

```
org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyDefault: Not able to place enough replicas, still in need of 3 to reach 3
```

Use the following steps to find the number of files where the block replication policy is equal to or above your current cluster size:

1. Provide a listing of open files, their blocks, the locations of those blocks by running the following command:

```
hadoop fsck / -files -blocks -locations -openforwrite 2>&1 >
openfiles.out
```

2. Run the following command to return a list of how many files have a given replication factor:

```
grep repl= openfiles.out | awk '{print $NF}' | sort | uniq -c
```

For example, when the replication factor is 10, and decommissioning one:

```
egrep -B4 "repl=10" openfiles.out | grep -v '<dir>' | awk '/^
\\/{print $1}'
```

3. Examine the paths, and decide whether to reduce the replication factor of the files, or remove them from the cluster.

Maintenance Mode

Maintenance mode allows you to suppress alerts for a host, service, role, or an entire cluster. This can be useful when you need to take actions in your cluster (make configuration changes and restart various elements) and do not want to see the alerts that will be generated due to those actions.

Putting an entity into maintenance mode does not prevent events from being logged; it only suppresses the alerts that those events would otherwise generate. You can see a history of all the events that were recorded for entities during the period that those entities were in maintenance mode.

Explicit and Effective Maintenance Mode

When you enter maintenance mode on an entity (cluster, service, or host) that has subordinate entities (for example, the roles for a service) the subordinate entities are also put into maintenance mode. These are considered to be in *effective maintenance mode*, as they have inherited the setting from the higher-level entity.

For example:

- If you set the HBase service into maintenance mode, then its roles (HBase Master and all RegionServers) are put into effective maintenance mode.
- If you set a host into maintenance mode, then any roles running on that host are put into effective maintenance mode.

Entities that have been explicitly put into maintenance mode show the icon . Entities that have entered effective maintenance mode as a result of inheritance from a higher-level entity show the icon .

When an entity (role, host or service) is in effective maintenance mode, it can only be removed from maintenance mode when the higher-level entity exits maintenance mode. For example, if you put a service into maintenance mode, the roles associated with that service are entered into effective maintenance mode, and remain in effective maintenance mode until the service exits maintenance mode. You cannot remove them from maintenance mode individually.

Alternatively, an entity that is in effective maintenance mode can be put into explicit maintenance mode. In this case, the entity remains in maintenance mode even when the higher-level entity exits maintenance mode. For example, suppose you put a host into maintenance mode, (which puts all the roles on that host into effective maintenance mode). You then select one of the roles on that host and put it explicitly into maintenance mode. When you have the host exit maintenance mode, that one role remains in maintenance mode. You need to select it individually and specifically have it exit maintenance mode.

Entering Maintenance Mode

You can enable maintenance mode for a cluster, service, role, or host.

Putting a Cluster into Maintenance Mode

1. In the left menu, click Clusters<cluster name>.
2. Click the Actions menu () to the right of the cluster name and select Enter Maintenance Mode.
3. Confirm that you want to do this.

The cluster is put into explicit maintenance mode, as indicated by the  icon. All services and roles in the cluster are entered into effective maintenance mode, as indicated by the  icon.

Putting a Service into Maintenance Mode

1. In the left menu, click Clusters and select the service.
2. Click ActionsEnter Maintenance Mode.
3. Confirm that you want to do this.

The service is put into explicit maintenance mode, as indicated by the  icon. All roles for the service are entered into effective maintenance mode, as indicated by the  icon.

Putting Roles into Maintenance Mode

1. In the left menu, click Clusters and select the service.
2. Click the Instances tab.
3. Select the role(s) you want to put into maintenance mode.
4. From the Actions for Selected menu, select Enter Maintenance Mode.
5. Confirm that you want to do this.

The roles will be put in explicit maintenance mode. If the roles were already in effective maintenance mode (because its service or host was put into maintenance mode) the roles will now be in explicit maintenance mode. This means that they will not exit maintenance mode automatically if their host or service exits maintenance mode; they must be explicitly removed from maintenance mode.

Putting Hosts into Maintenance Mode

1. In Cloudera Manager, select the cluster where you want to decommission hosts.
2. Click HostsAll Hosts.
3. Select the hosts that you want to put into Maintenance Mode.
4. Select Actions for SelectedBegin Maintenance (Suppress Alerts/Decommission).

The Begin Maintenance (Suppress Alerts/Decommission) dialog box opens. The role instances running on the hosts display at the top. You can also use this dialog box to decommission the host.

5. Deselect the Decommission Host(s) option to put the host into Maintenance Mode. In this mode, alerts from the hosts are suppressed until the host exits Maintenance Mode. The events, however, are still logged. Hosts that are

currently in Maintenance Mode display the  icon.

6. Click Begin Maintenance.

The Host Decommission Command dialog box opens and displays the progress of the command.

Exiting Maintenance Mode

When you exit maintenance mode, the maintenance mode icons are removed and alert notification resumes.

Exiting a Cluster from Maintenance Mode

1. Click  to the right of the cluster name and select Exit Maintenance Mode.
2. Confirm that you want to do this.

Exiting a Service from Maintenance Mode

1. Click  to the right of the service name and select Exit Maintenance Mode.
2. Confirm that you want to do this.

Exiting Roles from Maintenance Mode

1. Go to the services page that includes the role.
2. Go to the Instances tab.
3. Select the role(s) you want to exit from maintenance mode.
4. From the Actions for Selected menu, select Exit Maintenance Mode.
5. Confirm that you want to do this.

Taking Hosts out of Maintenance Mode

1. In Cloudera Manager, to go the cluster with the hosts you want to take out of Maintenance Mode.
2. Click HostsAll Hosts.

3. Select the hosts that are ready to exit Maintenance Mode.
4. Select Actions for SelectedEnd Maintenance (Suppress Alerts/Decommission).

The End Maintenance (Suppress Alerts/Decommission) dialog box opens. The role instances running on the hosts display at the top.

5. Deselect the Recommission Host(s) option to take the host out of Maintenance Mode and re-enable alerts from the hosts. Hosts that are currently in Maintenance Mode display the  icon on the All Hosts page.
6. Click End Maintenance.

Viewing the Maintenance Mode Status of a Cluster

For any cluster, you can view the components (service, roles, or hosts) that are in maintenance mode.

Procedure

1. From the Cloudera Manager Home page, select the cluster that you want to view the maintenance mode status for.
2. Click Actions View Maintenance Mode Status... .

This pops up a dialog box that shows the components in your cluster that are in maintenance mode, and indicates which are in effective maintenance mode as well as those that have been explicitly placed into maintenance mode.

From this dialog box you can select any of the components shown there and remove them from maintenance mode.

If individual services are in maintenance mode, you will see the maintenance mode icon  next to the Actions button for that service.



Note: The Actions button is not enabled if you are viewing status for a point of time in the past.

Managing Roles

When Cloudera Manager configures a service, it configures hosts in your cluster with one or more functions (called roles in Cloudera Manager) that are required for that service. The role determines which Hadoop daemons run on a given host. For example, when Cloudera Manager configures an HDFS service instance it configures one host to run the NameNode role, another host to run as the Secondary NameNode role, another host to run the Balancer role, and some or all of the remaining hosts to run DataNode roles.

Configuration settings are organized in role groups. A *role group* includes a set of configuration properties for a specific group, as well as a list of role instances associated with that role group. Cloudera Manager automatically creates default role groups.

For role types that allow multiple instances on multiple hosts, such as DataNodes, TaskTrackers, RegionServers (and many others), you can create multiple role groups to allow one set of role instances to use different configuration settings than another set of instances of the same role type. In fact, upon initial cluster setup, if you are installing on identical hosts with limited memory, Cloudera Manager will (typically) automatically create two role groups for each worker role — one group for the role instances on hosts with only other worker roles, and a separate group for the instance running on the host that is also hosting master roles.

The HDFS service is an example of this: Cloudera Manager typically creates one role group (DataNode Default Group) for the DataNode role instances running on the worker hosts, and another group (HDFS-1-DATANODE-1) for the DataNode instance running on the host that is also running the master roles such as the NameNode, JobTracker, HBase Master and so on. Typically the configurations for those two classes of hosts will differ in terms of settings such as memory for JVMs.

Cloudera Manager configuration screens offer two layout options: classic and new. The new layout is the default; however, on each configuration page you can easily switch between layouts using the Switch to XXX layout link at the top right of the page.

Gateway Roles

A *gateway* is a special type of role whose sole purpose is to designate a host that should receive a client configuration for a specific service, when the host does not have any roles running on it. Gateway roles enable Cloudera Manager to install and manage client configurations on that host. There is no process associated with a gateway role, and its status will always be Stopped. You can configure gateway roles for HBase, HDFS, Hive, Kafka, MapReduce, Solr, Spark, Sqoop 1 Client, and YARN.

Related Information

[Cluster Configuration Overview](#)

Role Instances

Starting, Stopping, and Restarting Role Instances

About this task

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

If the host for the role instance is currently decommissioned, you will not be able to start the role until the host has been recommissioned.



Important: Use Cloudera Manager to stop the Node Manager service. If it is stopped manually, it can cause jobs to fail.

Procedure

1. Go to the service that contains the role instances to start, stop, or restart.
2. Click the Instances tab.
3. Check the checkboxes next to the role instances to start, stop, or restart (such as a DataNode instance).
4. Select Actions for SelectedStart, Stop, or Restart, and then click Start, Stop, or Restart again to start the process. When you see a Finished status, the process has finished.

Related Information

[Rolling Restart](#)

Decommissioning Role Instances

You can remove a role instance such as a DataNode from a cluster while the cluster is running by decommissioning the role instance.

About this task

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

When you decommission a role instance, Cloudera Manager performs a procedure so that you can safely retire a host without losing data. Role decommissioning applies to HDFS DataNode, MapReduce TaskTracker, YARN NodeManager, and HBase RegionServer roles.

Hosts with DataNodes and DataNode roles themselves can only be decommissioned if the resulting action leaves enough DataNodes commissioned to maintain the configured HDFS replication factor (by default 3). If you attempt to

decommission a DataNode or a host with a DataNode in such situations, the decommission process will not complete and must be aborted.

A role will be decommissioned if its host is decommissioned.

To remove a DataNode from the cluster, you decommission the DataNode role as described here and then perform a few additional steps to remove the role. See the topic [Delete a DataNode](#).

Procedure

To decommission role instances:

1. If you are decommissioning DataNodes, perform the steps in the topic *Tuning HDFS Prior to Decommissioning DataNodes*.
2. Click the service instance that contains the role instance you want to decommission.
3. Click the Instances tab.
4. Check the checkboxes next to the role instances to decommission.
5. Select Actions for SelectedDecommission, and then click Decommission again to start the process.

Results

A Decommission Command pop-up displays that shows each step or decommission command as it is run. In the Details area, click  to see the subcommands that are run. Depending on the role, the steps may include adding the host to an "exclusions list" and refreshing the NameNode, JobTracker, or NodeManager; stopping the Balancer (if it is running); and moving data blocks or regions. Roles that do not have specific decommission actions are stopped.

You can abort the decommission process by clicking the Abort button, but you must recommission and restart the role.

The Commission State facet in the Filters list displays  Decommissioning while decommissioning is in progress, and  Decommissioned when the decommissioning process has finished. When the process is complete, a  is added in front of Decommission Command.

Related Information

[Tuning HDFS Prior to Decommissioning DataNodes](#)

Recommissioning Role Instances

Procedure

1. Click the service that contains the role instance you want to recommission.
2. Click the Instances tab.
3. Check the checkboxes next to the decommissioned role instances to recommission.
4. Select Actions for SelectedRecommission, and then click Recommission to start the process. A Recommission Command pop-up displays that shows each step or recommission command as it is run. When the process is complete, a  is added in front of Recommission Command.
5. Restart the role instance.

Configuring Roles to Use a Custom Garbage Collection Parameter

You can use Java configuration options to configure roles to use a custom garbage collection parameter.

Every Java-based role in Cloudera Manager has a configuration setting called Java Configuration Options for *role* where you can enter command line options. Commonly, garbage collection flags or extra debugging flags would be passed here. To find the appropriate configuration setting, select the service you want to modify in the Cloudera Manager Admin Console, then use the Search box to search for Java Configuration Options.

You can add configuration options for all instances of a given role by making this configuration change at the service level. For example, to modify the setting for all DataNodes, select the HDFS service, then modify the Java Configuration Options for DataNode setting.

To modify a configuration option for a given instance of a role, select the service, then select the particular role instance (for example, a specific DataNode). The configuration settings you modify will apply to the selected role instance only.

Related Information

[Modifying Configuration Properties Using Cloudera Manager](#)

Role Groups

Minimum Required Role: [Configurator](#) (also provided by Cluster Administrator, Limited Cluster Administrator, and Full Administrator)

A *role group* is a set of configuration properties for a role type, as well as a list of role instances associated with that group. Cloudera Manager automatically creates a default role group named *Role Type Default Group* for each role type. Each role instance can be associated with only a single role group.

Role groups provide two types of properties: those that affect the configuration of the service itself and those that affect monitoring of the service, if applicable (the Monitoring subcategory). Not all services have monitoring properties.

When you run the installation or upgrade wizard, Cloudera Manager configures the default role groups it adds, and adds any other required role groups for a given role type. For example, a DataNode role on the same host as the NameNode might require a different configuration than DataNode roles running on other hosts. Cloudera Manager creates a separate role group for the DataNode role running on the NameNode host and uses the default configuration for DataNode roles running on other hosts.

You can modify the settings of the default role group, or you can create new role groups and associate role instances to whichever role group is most appropriate. This simplifies the management of role configurations when one group of role instances may require different settings than another group of instances of the same role type—for example, due to differences in the hardware the roles run on. You modify the configuration for any of the service's role groups through the Configuration tab for the service. You can also override the settings inherited from a role group for a role instance.

If there are multiple role groups for a role type, you can move role instances from one group to another. When you move a role instance to a different group, it inherits the configuration settings for its new group.

Related Information

[Configuring Monitoring Settings](#)

[Overriding Configuration Properties](#)

Creating a Role Group

Procedure

1. Go to a service status page.
2. Click the Instances or Configuration tab.
3. Click Role Groups.
4. Click Create new group....
5. Provide a name for the group.
6. Select the role type for the group. You can select role types that allow multiple instances and that exist for the service you have selected.

7. In the Copy From field, select the source of the basic configuration information for the role group:
 - An existing role group of the appropriate type.
 - None.... The role group is set up with generic default values that are not the same as the values Cloudera Manager sets in the default role group, as Cloudera Manager specifically sets the appropriate configuration properties for the services and roles it installs. After you create the group you must edit the configuration to set missing properties (for example the TaskTracker Local Data Directory List property, which is not populated if you select None) and clear other validation warnings and errors.

Related Information

[Modifying Configuration Properties Using Cloudera Manager](#)

Managing Role Groups

Procedure

1. Go to a service status page.
2. Click the Instances or Configuration tab.
3. Click Role Groups.
4. Click the group you want to manage. Role instances assigned to the role group are listed.
5. Perform the appropriate procedure for the action:
 - Rename
 - a. Click the role group name, and click Rename.
 - b. Specify the new name and click Rename.
 - Delete

You cannot delete any of the default groups. The group must first be empty; if you want to delete a group you've created, you must move any role instances to a different role group.

 - a. Click the role group name.
 - b. Click Delete, and confirm by clicking Delete. Deleting a role group removes it from host templates.
 - Move
 - a. Select the role instance(s) to move.
 - b. Select Actions for Selected Move To Different Role Group....
 - c. In the pop-up that appears, select the target role group and click Move.

Related Information

[Managing Hosts](#)

Managing Cloudera Runtime Services

Cloudera Manager service configuration features let you manage the deployment and configuration of Cloudera Runtime and managed services.

Using Cloudera Manager, you can gracefully start, stop and restart services or roles. Further, you can modify the configuration properties for services or for individual role instances. . You can also generate client configuration files, enabling you to easily distribute them to the users of a service.

The topics in this chapter describe how to configure and use the services on your cluster. Some services have unique configuration requirements or provide unique features. See the documentation for an individual service for more information.

Starting a Cloudera Runtime Service on All Hosts

Before you begin

The order in which to start services is:

1. Cloudera Management Service
2. ZooKeeper
3. HDFS
4. Solr
5. Flume
6. HBase
7. Key-Value Store Indexer
8. MapReduce or YARN
9. Hive
10. Impala
11. Oozie
12. Sqoop
13. Hue

Procedure

1. In the left menu, click Clusters and select a service.
2. Click  to the right of the service name and select Start.
3. Click Start in the next screen to confirm.
When you see a Finished status, the service has started.

Results



Note: If you are unable to start the HDFS service, it's possible that one of the roles instances, such as a DataNode, was running on a host that is no longer connected to the Cloudera Manager Server host, perhaps because of a hardware or network failure. If this is the case, the Cloudera Manager Server will be unable to connect to the Cloudera Manager Agent on that disconnected host to start the role instance, which will prevent the HDFS service from starting. To work around this, you can stop all services, abort the pending command to start the role instance on the disconnected host, and then restart all services again without that role instance.

Related Information

[Aborting a Pending Command](#)

Stopping a Cloudera Runtime Service on All Hosts

Before you begin

The order in which to stop services is:

1. Hue
2. Sqoop
3. Oozie
4. Impala
5. Hive

6. MapReduce or YARN
7. Key-Value Store Indexer
8. HBase
9. Flume
10. Solr
11. HDFS
12. ZooKeeper
13. Cloudera Management Service

Procedure

1. In the left menu, click Clusters and select a service.
2. Click  to the right of the service name and select Stop.
3. Click Stop in the next screen to confirm.
When you see a Finished status, the service has stopped.

Restarting a Cloudera Runtime Service

Procedure

1. In the left menu, click Clusters and select a service.
2. Click  to the right of the service name and select Restart.
3. Click Start on the next screen to confirm.

Results

When you see a Finished status, the service has restarted.

What to do next

To restart all services, restart the cluster.

Rolling Restart

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Rolling restart allows you to conditionally restart the role instances of the following services to update software or use a new configuration:

- Flume
- HBase
- HDFS
- Kafka
- Key Trustee KMS
- Key Trustee Server
- MapReduce
- Oozie
- YARN
- ZooKeeper

If the service is not running, rolling restart is not available for that service. You can specify a rolling restart of each service individually.

Performing a Service or Role Rolling Restart

You can initiate a rolling restart from either the Status page for one of the eligible services, or from the service's Instances page, where you can select individual roles to be restarted.

1. Go to the service you want to restart.
2. Do one of the following:
 - service - Select ActionsRolling Restart.
 - role -
 - a. Click the Instances tab.
 - b. Select the roles to restart.
 - c. Select Actions for SelectedRolling Restart.
3. In the pop-up dialog box, select the options you want:
 - Restart only roles whose configurations are stale
 - Restart only roles that are running outdated software versions
 - Which role types to restart
4. If you select an HDFS, HBase, MapReduce, or YARN service, you can have their worker roles restarted in batches. You can configure:
 - How many roles should be included in a batch - Cloudera Manager restarts the worker roles rack-by-rack in alphabetical order, and within each rack, hosts are restarted in alphabetical order. If you are using the default replication factor of 3, Hadoop tries to keep the replicas on at least 2 different racks. So if you have multiple racks, you can use a higher batch size than the default 1. But you should be aware that using too high batch size also means that fewer worker roles are active at any time during the upgrade, so it can cause temporary performance degradation. If you are using a single rack only, you should only restart one worker node at a time to ensure data availability during upgrade.
 - How long should Cloudera Manager wait before starting the next batch.

- The number of batch failures that will cause the entire rolling restart to fail (this is an advanced feature). For example if you have a very large cluster you can use this option to allow failures because if you know that your cluster will be functional even if some worker roles are down.



Note:

- HDFS - If you do not have HDFS high availability configured, a warning appears reminding you that the service will become unavailable during the restart while the NameNode is restarted. Services that depend on that HDFS service will also be disrupted. Cloudera recommends that you restart the DataNodes one at a time—one host per batch, which is the default.
- HBase
 - Administration operations such as any of the following should not be performed during the rolling restart, to avoid leaving the cluster in an inconsistent state:
 - Split
 - Create, disable, enable, or drop table
 - Metadata changes
 - Create, clone, or restore a snapshot. Snapshots rely on the RegionServers being up; otherwise the snapshot will fail.
 - To increase the speed of a rolling restart of the HBase service, set the Region Mover Threads property to a higher value. This increases the number of regions that can be moved in parallel, but places additional strain on the HMaster. In most cases, Region Mover Threads should be set to 5 or lower.
 - Another option to increase the speed of a rolling restart of the HBase service is to set the Skip Region Reload During Rolling Restart property to true. This setting can cause regions to be moved around multiple times, which can degrade HBase client performance.
- MapReduce - If you restart the JobTracker, all current jobs will fail.
- YARN - If you restart ResourceManager and ResourceManager HA is enabled, current jobs continue running: they do not restart or fail.
- ZooKeeper and Flume - For both ZooKeeper and Flume, the option to restart roles in batches is not available. They are always restarted one by one.

5. Click Confirm to start the rolling restart.

Aborting a Pending Command

Minimum Required Role: [Operator](#) (also provided by Configurator, Cluster Administrator, Limited Cluster Administrator, Full Administrator)

Commands will time out if they are unable to complete after a period of time.

If necessary, you can abort a pending command. For example, this may become necessary because of a hardware or network failure where a host running a role instance becomes disconnected from the Cloudera Manager Server host. In this case, the Cloudera Manager Server will be unable to connect to the Cloudera Manager Agent on that disconnected host to start or stop the role instance which will prevent the corresponding service from starting or stopping. To work around this, you can abort the command to start or stop the role instance on the disconnected host, and then you can start or stop the service again.

To abort any pending command:



You can click the Recent Commands indicator (), which shows the number of commands that are currently running in your cluster (if any). This indicator is positioned above the Support link at the bottom of the left menu. Unlike the Commands tab for a role or service, this indicator includes all commands running for all services or roles in the cluster. In the **Running Commands** window, click Abort to abort the pending command.

To abort a pending command for a service or role:

1. In the left menu, click Clusters and select the service where the role instance you want to stop is located. For example, click ClustersHDFS Service if you want to abort a pending command for a DataNode.
2. Click the Instances tab.
3. In the list of instances, click the link for role instance where the command is running (for example, the instance that is located on the disconnected host).
4. Go to the Commands tab.
5. Find the command in the list of Running Commands and click Abort Command to abort the running command.

Related Information

[Viewing Running and Recent Commands](#)

Managing Cloudera Manager

Automatic Logout

For security purposes, Cloudera Manager automatically logs out a user session after 30 minutes. You can change this session logout period.

To configure the timeout period:

1. Click AdministrationSettings.
2. Click CategorySecurity.
3. Edit the Session Timeout property.
4. Enter a Reason for change, and then click Save Changes to commit the changes.

When the timeout is one minute from triggering, the user sees the following message:

Automatic Logout for Your Protection ✕

Due to inactivity, your current work session is about to expire. For your security, Cloudera Manager sessions automatically end after 30 minutes of inactivity.

Your current session will expire in **1 minute**.
Press any key or click anywhere to continue.

If the user does not click the mouse or press a key, the user is logged out of the session and the following message appears:

Automatic Log Out Due to Inactivity

You are now logged out of your account.

We hadn't heard from you for about 30 minute(s), so for your security Cloudera Manager automatically logged you out of your account. Log back in below to continue.

 Remember me

Starting, Stopping, and Restarting the Cloudera Manager Server

To start the Cloudera Manager Server:

```
sudo service cloudera-scm-server start
```

You can stop (for example, to perform maintenance on its host) or restart the Cloudera Manager Server without affecting the other services running on your cluster. Statistics data used by activity monitoring and service monitoring will continue to be collected during the time the server is down.

To stop the Cloudera Manager Server:

```
sudo service cloudera-scm-server stop
```

To restart the Cloudera Manager Server:

```
sudo service cloudera-scm-server restart
```

Managing Cloudera Manager Server Logs

You can use the Cloudera Manager Server logs to troubleshoot problems with Cloudera Manager .

Related Information

[Logs](#)

Viewing the Cloudera Manager Server Logs

To help you troubleshoot problems, you can view the Cloudera Manager Server log. You can view the logs in the **Logs** page or in specific pages for the log.

Procedure

1. In the left menu, click **Diagnostics**Logs.
2. Next to **Sources**, select the **Cloudera Manager Server** checkbox and deselect the other options.
3. Adjust the search criteria and click **Search**.

What to do next

You can also view the raw Cloudera Manager Server log by logging in to the Cloudera Manager Server host and view the `/var/log/cloudera-scm-server/cloudera-scm-server.log` file.

Setting the Cloudera Manager Server Log Location

You can set the location of the Cloudera Manager Server log.

Procedure

1. Stop the Cloudera Manager Server:

```
sudo service cloudera-scm-server stop
```

2. Set the `CMF_VAR` environment variable in `/etc/default/cloudera-scm-server` to the new parent directory:

```
export CMF_VAR=/opt
```

3. Create `log/cloudera-scm_server` and run directories in the new parent directory and set the owner and group of all directories to `cloudera-scm`. For example, if the new parent directory is `/opt/`, do the following:

```
sudo su
cd /opt
mkdir log
chown cloudera-scm:cloudera-scm log
mkdir /opt/log/cloudera-scm-server
chown cloudera-scm:cloudera-scm log/cloudera-scm-server
mkdir run
chown cloudera-scm:cloudera-scm run
```

4. Restart the Cloudera Manager Server:

```
sudo service cloudera-scm-server start
```

Configuring Cloudera Manager

From the Administration menu you can select options for configuring settings that affect how Cloudera Manager interacts with your clusters.

Settings

The Settings page provides a number of categories as follows:

- Performance - Set the Cloudera Manager Agent heartbeat interval.

- Advanced - Enable API debugging and other advanced options.
- Monitoring - Set Agent health status parameters. For configuration instructions, see the topic *Configuring Cloudera Manager Agents*.
- Other
 - Enable Cloudera usage data collection For configuration instructions, see *Managing Anonymous Usage Data Collection*.
 - Set a custom header color and banner text for the Admin console.
 - Set an "Information Assurance Policy" statement – this statement will be presented to every user before they are allowed to access the login dialog box. The user must click "I Agree" in order to proceed to the login dialog box.
 - Disable/enable the auto-search for the Events panel at the bottom of a page.
- Support
 - Configure diagnostic data collection properties. See *Diagnostic Data Collection*.
 - Configure how to access Cloudera Manager help documentation.

You can also configure the following:

- Alerts
- Users
- Language

You can change the language of the Cloudera Manager Admin Console User Interface through the language preference in your browser. Information on how to do this for the browsers supported by Cloudera Manager is shown under the Administration page. You can also change the language for the information provided with activity and health events, and for alert email messages by selecting Language, selecting the language you want from the drop-down list on this page, then clicking Save Changes.

Related Information

[Managing Anonymous Usage Data Collection](#)

[Diagnostic Data Collection](#)

Managing Anonymous Usage Data Collection

Cloudera Manager sends anonymous usage information using Google Analytics to Cloudera. The information helps Cloudera improve Cloudera Manager. By default, anonymous usage data collection is enabled.

Procedure

1. In the left menu, select AdministrationSettings.
2. Under the Other category, set the Allow Usage Data Collection property.
3. Enter a Reason for change, and then click Save Changes to commit the changes.

Diagnostic Data Collection

To help with solving problems when using Cloudera Manager on your cluster, Cloudera Manager collects diagnostic data on a regular schedule, and automatically sends it to Cloudera.

By default Cloudera Manager is configured to collect this data weekly and to send it automatically. Cloudera analyzes this data and uses it to improve the software. If Cloudera discovers a serious issue, Cloudera searches this diagnostic data and notifies customers who might encounter problems due to the issue. You can schedule the frequency of data collection on a daily, weekly, or monthly schedule, or disable the scheduled collection of data entirely. You can also send a collected data set manually.

Automatically sending diagnostic data requires the Cloudera Manager Server host to have Internet access, and be configured for sending data automatically. If your Cloudera Manager Server does not have Internet access, you can manually send the diagnostic data.

Automatically sending diagnostic data might fail sometimes and return an error message of "Could not send data to Cloudera." To work around this issue, you can manually send the data to Cloudera Support.

Related Information

[Manually Triggering Collection and Transfer of Diagnostic Data](#)

Manually Triggering Collection and Transfer of Diagnostic Data

To troubleshoot specific problems, or to re-send an automatic bundle that failed to send, you can manually send diagnostic data to Cloudera.

Procedure

1. Optionally, change the System Identifier property:
 - a) In the left menu, select AdministrationSettings.
 - b) Under the Other category, set the System Identifier property and click Save Changes.
2. Click Support at the bottom of the left menu and choose Send Diagnostic Data. The **Send Diagnostic Data** form displays.
3. Fill in or change the information here as appropriate.

Optionally, you can improve performance by reducing the size of the data bundle that is sent. Click Restrict log and metrics collection to expand this section of the form. The three filters, Host, Service, and Role Type, allow you to restrict the data that will be sent. Cloudera Manager will only collect logs and metrics for roles that match all three filters.

Select one of the following under Data Selection:

- Select By Target Size to manually set the maximum size of the bundle. Cloudera Manager populates the End Time based on the setting of the Time Range selector. You should change this to be a few minutes after you observed the problem or condition that you are trying to capture. The time range is based on the timezone of the host where Cloudera Manager Server is running.
- Select By Date Range to manually set the start time and end time to collect the diagnostic data. Click the Estimate button to calculate the size of the bundle based on the start and end times. If the bundle is too large, narrow the selection using the start and end times or by selecting additional filters.

If you have a support ticket open with Cloudera Support, include the support ticket number in the field provided.

4. Depending on whether you have disabled automatic sending of data, do one of the following:
 - Click Collect and Upload Diagnostic Data to Cloudera Support. A **Running Commands** window shows you the progress of the data collection steps. When these steps are complete, the collected data is sent to Cloudera.
 - Click Collect Diagnostic Data only. A **Command Details** window shows you the progress of the data collection steps.
 - a) In the Command Details window, click Download Result Data to download and save a zip file of the information.
 - b) Send the data to Cloudera Support by doing one of the following:
 - Send the bundle using a Python script. Download the phone_home script and copy the script and downloaded data file to a host that has internet access. Run the following command on that host: `python phone_home.py --file downloaded data file`
 - Attach the bundle to the SFDC case. Do not rename the bundle as this can cause a delay in processing the bundle.
 - Contact Cloudera Support and arrange to send the data file.

Related Information

[phone_home script](#)

Cloudera Manager Agents

The Cloudera Manager Agent is a Cloudera Manager component that works with the Cloudera Manager Server to manage the processes that map to role instances.

In a Cloudera Manager managed cluster, you can only start or stop role instance processes using Cloudera Manager. Cloudera Manager uses an open source process management tool called `supervisord`, that starts processes, takes care of redirecting log files, notifying of process failure, setting the effective user ID of the calling process to the right user, and so on. Cloudera Manager supports automatically restarting a crashed process. It will also flag a role instance with a bad health flag if its process crashes repeatedly right after start up.

The Agent is started by `init.d` at start-up. It, in turn, contacts the Cloudera Manager Server and determines which processes should be running. The Agent is monitored as part of Cloudera Manager's host monitoring. If the Agent stops heartbeating, the host is marked as having bad health.

One of the Agent's main responsibilities is to start and stop processes. When the Agent detects a new process from the Server heartbeat, the Agent creates a directory for it in `/var/run/cloudera-scm-agent` and unpacks the configuration. It then contacts `supervisord`, which starts the process.

cm_processes

To enable Cloudera Manager to run scripts in subdirectories of `/var/run/cloudera-scm-agent`, (because `/var/run` is mounted `noexec` in many Linux distributions), Cloudera Manager mounts a `tmpfs`, named `cm_processes`, for process subdirectories.

A `tmpfs` defaults to a max size of 50% of physical RAM but this space is not allocated until its used, and `tmpfs` is paged out to swap if there is memory pressure.

The lifecycle actions of `cm_processes` can be described by the following statements:

- Created when the Agent starts up for the first time with a new `supervisord` process.
- If it already exists without `noexec`, reused when the Agent is started using `start` and not recreated.
- Remounted if Agent is started using `clean_restart`.
- Unmounting and remounting cleans out the contents (since it is mounted as a `tmpfs`).
- Unmounted when the host is rebooted.
- Not unmounted when the Agent is stopped.

Related Information

[supervisord](#)

[tmpfs](#)

Starting, Stopping, and Restarting Cloudera Manager Agents

Starting Agents

To start Agents, the `supervisord` process, and all managed service processes, use the following command:

- Start

```
sudo service cloudera-scm-agent start
```

Stopping and Restarting Agents

To stop or restart Agents while leaving the managed processes running, use one of the following commands:

- Stop

```
sudo service cloudera-scm-agent stop
```

- Restart

```
sudo service cloudera-scm-agent restart
```

Hard Stopping and Restarting Agents



Warning: The `hard_stop` and `hard_restart` commands kill all running managed service processes on the host(s) where the command is run.

To stop or restart Agents, the `supervisord` process, and all managed service processes, use one of the following commands:

- Hard Stop

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo /etc/init.d/cloudera-scm-agent next_stop_hard
sudo systemctl stop cloudera-scm-agent
```

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

```
sudo service cloudera-scm-agent hard_stop
```

- Hard Restart

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

```
sudo /etc/init.d/cloudera-scm-agent next_stop_hard
sudo systemctl restart cloudera-scm-agent
```

RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04, 14.04

```
sudo service cloudera-scm-agent hard_restart
```

Hard restart is useful for the following situations:

- You are upgrading Cloudera Manager and the `supervisord` code has changed between your current version and the new one. To properly do this upgrade you need to restart supervisor too.
- `supervisord` freezes and needs to be restarted.
- You want to clear out all running state pertaining to Cloudera Manager and managed services.

Checking Agent Status

To check the status of the Agent process, use the command:

```
sudo service cloudera-scm-agent status
```

Managing the Cloudera Manager Agent Logs

To help you troubleshoot problems, you can view the Cloudera Manager Agent logs. You can view the logs in the Logs page or in specific pages for the logs.

Viewing the Cloudera Manager Agent Logs

Use the procedure to view and search the logs from all Cloudera Manager agents managed by this instance of Cloudera Manager.

Procedure

1. In the left menu, click DiagnosticsLogs.
2. Click Select Sources to display the log source list.
3. Uncheck the All Sources checkbox.
4. Click ► to the left of Cloudera Manager and select the Agent checkbox.
5. Click Search.

What to do next

You can also view the Cloudera Manager Agent log at `/var/log/cloudera-scm-agent/cloudera-scm-agent.log` on the Agent hosts.

Setting the Cloudera Manager Agent Log Location

By default, the Cloudera Manager Agent log is stored in `/var/log/cloudera-scm-agent/`. If there is not enough space in that directory, you can change the location of the log file.

Procedure

1. Set the `log_file` property in the Cloudera Manager Agent configuration file:

```
log_file=/opt/log/cloudera-scm-agent/cloudera-scm-agent.log
```

2. Create `log/cloudera-scm_agent` directories and set the owner and group to `cloudera-scm`. For example, if the log is stored in `/opt/log/cloudera-scm-agent`, do the following:

```
sudo su
cd /opt
mkdir log
chown cloudera-scm:cloudera-scm log
mkdir /opt/log/cloudera-scm-agent
chown cloudera-scm:cloudera-scm log/cloudera-scm-agent
```

3. Restart the Agent:

```
sudo service cloudera-scm-agent restart
```

Default User Roles

By default, Cloudera Manager ships with user roles that have privileges for all clusters managed by Cloudera Manager.

The following list describes the actions each user role can perform:

- Auditor
 - View configuration and monitoring information in Cloudera Manager.
 - View audit events.

- Read-Only
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - View events and logs.
 - View replication jobs and snapshot policies.
 - View YARN applications and Impala queries.

The Read-Only role does not allow the user to:

- Add services or take any actions that affect the state of the cluster.
- Use the HDFS file browser.
- Use the HBase table browser.
- Use the Solr Collection Statistics browser.
- Dashboard
 - Create, edit, or remove dashboards that belong to the user.
 - Add an existing chart or create a new chart to add to a dashboard that belongs to the user.
 - Perform the same actions as the [Read-Only role](#).
- Limited Operator
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - Decommission hosts (except hosts running Cloudera Management Service roles).
 - Perform the same actions as the [Read-Only role](#).

The Limited Operator role does not allow the user to add services or take any other actions that affect the state of the cluster.

- Operator
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - Stop, start, and restart clusters, services (except the Cloudera Management Service), and roles.
 - Decommission and recommission hosts (except hosts running Cloudera Management Service roles).
 - Decommission and recommission roles (except Cloudera Management Service roles).
 - Start, stop, and restart KMS.
 - Perform the same actions as the [Read-Only role](#).

The Operator role does not allow the user to add services, roles, or hosts, or take any other actions that affect the state of the cluster.

- Configurator
 - View configuration and monitoring information in Cloudera Manager.
 - Perform all Operator operations.
 - Configure roles and services (except the Cloudera Management Service).
 - Enter and exit maintenance mode.
 - Manage dashboards (including Cloudera Management Service dashboards).
 - Start, stop, and restart KMS
 - Perform the same actions as the [Read-Only role](#).

- Cluster Administrator
 - Apply policies to redact sensitive data.
 - Recommission hosts, and decommission and recommission roles.
 - Enter and exit Maintenance Mode.
 - Edit the configuration of services and roles.
 - Access all functionality that Cloudera Manager offers.
 - Start, stop, and restart most clusters, services, and roles.
 - View data in Cloudera Manager.
 - Start, stop, and restart KMS.
 - Decommission hosts.
- BDR Administrator
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - Perform replication and define snapshot operations.
 - Perform the same actions as the [Read-Only role](#).
- Navigator Administrator
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - Administer Cloudera Navigator.
 - View audit events.
 - Perform the same actions as the [Read-Only role](#).
- User Administrator
 - View configuration and monitoring information in Cloudera Manager.
 - View service and monitoring information.
 - Manage user accounts and configuration of external authentication.
 - Create, update, or delete external account configuration.
 - Perform the same actions as the [Read-Only role](#).
- Key Administrator
 - View configuration and monitoring information in Cloudera Manager.
 - Configure HDFS encryption, administer Key Trustee Server, and manage encryption keys.
 - Start, stop, and restart KMS
 - Configure KMS ACLs
 - Perform the same actions as the [Read-Only role](#).

- Full Administrator
 - Apply policies to redact sensitive data.
 - Administer Cloudera Navigator.
 - Create, modify, and delete your own dashboards.
 - Manage user accounts and configuration of external authentication.
 - Enter and exit Maintenance Mode.
 - Edit the configuration of services and roles.
 - View data in Cloudera Manager.
 - Start, stop, and restart KMS.
 - Manage Full Administrator accounts.
 - Decommission hosts.
 - View audit events.
 - Create, update, or delete external account configuration.
 - Configure HDFS Encryption, administer Key Trustee Server, and manage encryption keys.
 - Recommission hosts, and decommission and recommission roles.
 - Access all functionality that Cloudera Manager offers.
 - Create replication schedules and snapshot policies.
 - Start, stop, and restart most clusters, services, and roles.

Accessing Storage Using Amazon S3

Referencing S3 Credentials for YARN, MapReduce, or Spark Clients

If you have selected IAM authentication, no additional steps are needed. If you are not using IAM authentication, use one of the following three options to provide Amazon S3 credentials to clients.



Note: This method of specifying AWS credentials to clients does not completely distribute secrets securely because the credentials are not encrypted. Use caution when operating in a multi-tenant environment.

Programmatic

Specify the credentials in the configuration for the job. This option is most useful for Spark jobs.

Make a modified copy of the configuration files

Make a copy of the configuration files and add the S3 credentials:

1. For YARN and MapReduce jobs, copy the contents of the `/etc/hadoop/conf` directory to a local working directory under the home directory of the host where you will submit the job. For Spark jobs, copy `/etc/spark/conf` to a local directory under the home directory of the host where you will submit the job.
2. Set the permissions for the configuration files appropriately for your environment and ensure that unauthorized users cannot access sensitive configurations in these files.
3. Add the following to the `core-site.xml` file within the `<configuration>` element:

```
<property>
  <name>fs.s3a.access.key</name>
  <value>Amazon S3 Access Key</value>
</property>

<property>
  <name>fs.s3a.secret.key</name>
  <value>Amazon S3 Secret Key</value>
```

```
</property>
```

4. Reference these versions of the configuration files when submitting jobs by running the following command:

- YARN or MapReduce:

```
export HADOOP_CONF_DIR=path to local configuration directory
```

- Spark:

```
export SPARK_CONF_DIR=path to local configuration directory
```



Note: If you update the client configuration files from Cloudera Manager, you must repeat these steps to use the new configurations.

Reference the managed configuration files and add AWS credentials

This option allows you to continue to use the configuration files managed by Cloudera Manager. If you deploy new configuration files, the new values are included by reference in your copy of the configuration files while also maintaining a version of the configuration that contains the Amazon S3 credentials:

1. Create a local directory under your home directory.
2. Copy the configuration files from `/etc/hadoop/conf` to the new directory.
3. Set the permissions for the configuration files appropriately for your environment.
4. Edit each configuration file:
 - a. Remove all elements within the `<configuration>` element.
 - b. Add an XML `<include>` element within the `<configuration>` element to reference the configuration files managed by Cloudera Manager. For example:

```
<include xmlns="http://www.w3.org/2001/XInclude"
  href="/etc/hadoop/conf/hdfs-site.xml">
  <fallback />
</include>
```

5. Add the following to the `core-site.xml` file within the `<configuration>` element:

```
<property>
  <name>fs.s3a.access.key</name>
  <value>Amazon S3 Access Key</value>
</property>

<property>
  <name>fs.s3a.secret.key</name>
  <value>Amazon S3 Secret Key</value>
</property>
```

6. Reference these versions of the configuration files when submitting jobs by running the following command:

- YARN or MapReduce:

```
export HADOOP_CONF_DIR=path to local configuration directory
```

- Spark:

```
export SPARK_CONF_DIR=path to local configuration directory
```

Example core-site.xml file:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <include xmlns="http://www.w3.org/2001/XInclude"
    href="/etc/hadoop/conf/core-site.xml">
    <fallback />
  </include>

  <property>
    <name>fs.s3a.access.key</name>
    <value>Amazon S3 Access Key</value>
  </property>

  <property>
    <name>fs.s3a.secret.key</name>
    <value>Amazon S3 Secret Key</value>
  </property>
</configuration>
```

Referencing Amazon S3 in URIs

By default, files are still placed on the local HDFS and not on S3 if the protocol is not specified in the URI. When you have added the Amazon S3 service, use one of the following options to construct the URIs to reference when submitting jobs:

- Amazon S3:

```
s3a://bucket_name/path
```

- HDFS:

```
hdfs://path
```

or

```
/path
```

Related Information

[Accessing Data Stored in Amazon S3 through Spark](#)

[Impala with Amazon S3](#)

Using Fast Upload with Amazon S3

Writing data to Amazon S3 is subject to limitations of the s3a OutputStream implementation, which buffers the entire file to disk before uploading it to S3. This can cause the upload to proceed very slowly and can require a large amount of temporary disk space on local disks.

You can configure a cluster to use the Fast Upload feature. This feature implements several performance improvements and has tunable parameters for buffering to disk (the default) or to memory, tuning the number of threads, and for specifying the disk directories used for buffering.

Related Information

[Hadoop-AWS module: Integration with Amazon Web Services](#)

Enabling Fast Upload using Cloudera Manager

Procedure

To enable Fast Upload for clusters managed by Cloudera Manager:

1. Go to the HDFS service.
2. Click the Configuration tab.
3. Search for "core-site.xml" and locate the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml property.
4. Add the fs.s3a.fast.upload property and set it to true.
5. Set any additional tuning properties in the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml configuration properties.
6. Click Save Changes.

Results

Cloudera Manager will indicate that there are stale services and which services need to be restarted.

Related Information

[Setting an Advanced Configuration Snippet for a Cluster](#)

[Restarting a Cloudera Runtime Service](#)

How to Configure a MapReduce Job to Access S3 with an HDFS Credstore

Configure your MapReduce jobs to read and write to Amazon S3 using a custom password for an HDFS Credstore.

Procedure

1. Copy the contents of the /etc/hadoop/conf directory to a local working directory on the host where you will submit the MapReduce job. Use the --dereference option when copying the file so that symlinks are correctly resolved. For example:

```
cp -r --dereference /etc/hadoop/conf ~/my_custom_config_directory
```

2. Change the permissions of the directory so that only you have access:

```
chmod go-wrx -R my_custom_config_directory/
```

If you see the following message, you can ignore it:

```
cp: cannot open '/etc/hadoop/conf/container-executor.cfg' for reading: Permission denied
```

3. Add the following to the copy of the core-site.xml file in the working directory:

```
<property>
  <name>hadoop.security.credential.provider.path</name>
  <value>jceks://hdfs/user/username/awscreds.jceks</value>
</property>
```

4. Specify a custom Credstore by running the following command on the client host:

```
export HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password
```

5. In the working directory, edit the `mapred-site.xml` file:

a) Add the following properties:

```
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password</value>
</property>

<property>
  <name>mapred.child.env</name>
  <value>HADOOP_CREDSTORE_PASSWORD=your_custom_keystore_password</value>
</property>
```

b) Add `yarn.app.mapreduce.am.env` and `mapred.child.env` to the comma-separated list of values of the `mapreduce.job.redacted-properties` property. For example (new values shown bold):

```
<property>
  <name>mapreduce.job.redacted-properties</name>
  <value>fs.s3a.access.key,fs.s3a.secret
  .key,yarn.app.mapreduce.am.env,mapred.child.env</value>
</property>
```

6. Set the environment variable to point to your working directory:

```
export HADOOP_CONF_DIR=~/path_to_working_directory
```

7. Create the Credstore by running the following commands:

```
hadoop credential create fs.s3a.access.key
hadoop credential create fs.s3a.secret.key
```

You will be prompted to enter the access key and secret key.

8. List the credentials to make sure they were created correctly by running the following command:

```
hadoop credential list
```

9. Submit your job. For example:

- `ls`

```
hdfs dfs -ls s3a://S3_Bucket/
```

- `distcp`

```
hadoop distcp hdfs_path s3a://S3_Bucket/S3_path
```

- `teragen` (package-based installations)

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar teragen
100 s3a://S3_Bucket/teragen_test
```

- `teragen` (parcel-based installations)

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapredu
ce-examples.jar teragen 100 s3a://S3_Bucket/teragen_test
```

Importing Data into Amazon S3 Using Sqoop

Sqoop supports data import from RDBMS into Amazon S3.



Note: Sqoop import is supported only into the S3A (s3a:// protocol) filesystem.

Related Information

[Hadoop-AWS module: Integration with Amazon Web Services](#)

Authentication

You must authenticate to an S3 bucket using Amazon Web Service credentials. There are three ways to pass these credentials:

- Provide them in the configuration file or files manually.
- Provide them on the sqoop command line.
- Reference a credential store to "hide" sensitive data, so that they do not appear in the console output, configuration file, or log files.

Amazon S3 Block Filesystem URI example:

```
s3a://bucket_name/path/to/file
```

S3 credentials can be provided in a configuration file (for example, core-site.xml):

```
<property>
  <name>fs.s3a.access.key</name>
  <value>...</value>
</property>
<property>
  <name>fs.s3a.secret.key</name>
  <value>...</value>
</property>
```

You can also set up the configurations through Cloudera Manager by adding the configurations to the appropriate Advanced Configuration Snippet property.

Credentials can be provided through the command line:

```
sqoop import -Dfs.s3a.access.key=... -Dfs.s3a.secret.key=... --target-dir s3a://
```

For example:

```
sqoop import -Dfs.s3a.access.key=$ACCES_KEY -Dfs.s3a.secret.key=$SECRET_KEY
--connect $CONN --username $USER --password $PWD --table $TABLENAME --target
-dir s3a://example-bucket/target-directory
```



Note: Entering sensitive data on the command line is inherently insecure. The data entered can be accessed in log files and other artifacts. Cloudera recommends that you use a credential provider to store credentials.

Using a Credential Provider to Secure S3 Credentials

You can run the sqoop command without entering the access key and secret key on the command line. This prevents these credentials from being exposed in the console output, log files, configuration files, and other artifacts. Running the command this way requires that you provision a credential store to securely store the access key and secret key. The credential store file is saved in HDFS.



Note: Using a Credential Provider does not work with MapReduce v1 (MRV1).

To provision credentials in a credential store:

1. Provision the credentials by running the following commands:

```
hadoop credential create fs.s3a.access.key -value access_key -provider jceks://hdfs/path_to_credential_store_file
hadoop credential create fs.s3a.secret.key -value secret_key -provider jceks://hdfs/path_to_credential_store_file
```

For example:

```
hadoop credential create fs.s3a.access.key -value foobar -provider jceks://hdfs/user/alice/home/keystores/aws.jceks
hadoop credential create fs.s3a.secret.key -value barfoo -provider jceks://hdfs/user/alice/home/keystores/aws.jceks
```

You can omit the `-value` option and its value. When the option is omitted, the command will prompt the user to enter the value.

2. Copy the contents of the `/etc/hadoop/conf` directory to a working directory.
3. Add the following to the `core-site.xml` file in the working directory:

```
<property>
<name>hadoop.security.credential.provider.path</name>
<value>jceks://hdfs/path_to_credential_store_file</value>
</property>
```

4. Set the `HADOOP_CONF_DIR` environment variable to the location of the working directory:

```
export HADOOP_CONF_DIR=path_to_working_directory
```

After completing these steps, you can run the `sqoop` command using the following syntax:

Import into a target directory in an Amazon S3 bucket while credentials are stored in a credential store file and its path is set in the `core-site.xml`.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLENAME --target-dir s3a://example-bucket/target-directory
```

You can also reference the credential store on the command line, without having to enter it in a copy of the `core-site.xml` file. You also do not have to set a value for `HADOOP_CONF_DIR`. Use the following syntax:

Import into a target directory in an Amazon S3 bucket while credentials are stored in a credential store file and its path is passed on the command line.

```
sqoop import -Dhadoop.security.credential.provider.path=jceks://hdfspath-to-credential-store-file --connect $CONN --username $USER --password $PWD --table $TABLENAME --target-dir s3a://example-bucket/target-directory
```

Related Information

[Credential Management \(Apache Software Foundation\)](#)

Sqoop Import into Amazon S3

Import Data from RDBMS into an S3 Bucket

The `--target-dir` option must be set to the target location in the S3 bucket to import data from RDBMS into an S3 bucket.

Example command: Import data into a target directory in an Amazon S3 bucket.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLENAME --target-dir s3a://example-bucket/target-directory
```

Data from RDBMS can be imported into S3 as Sequence or Avro file format too.

Parquet import into S3 is also supported if the Parquet Hadoop API based implementation is used, meaning that the `--parquet-configurator-implementation` option is set to `hadoop`.

Example command: Import data into a target directory in an Amazon S3 bucket as Parquet file.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLENAME --target-dir s3a://example-bucket/target-directory --as-parquetfile --parquet-configurator-implementation hadoop
```

Import Data into S3 Bucket in Incremental Mode

The `--temporary-rootdir` option must be set to point to a location in the S3 bucket to import data into an S3 bucket in incremental mode.

Append Mode

When importing data into a target directory in an Amazon S3 bucket in incremental append mode, the location of the temporary root directory must be in the same bucket as the directory. For example: `s3a://example-bucket/temporary-rootdir` or `s3a://example-bucket/target-directory/temporary-rootdir`.

Example command: Import data into a target directory in an Amazon S3 bucket in incremental append mode.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incremental append --check-column $CHECK_COLUMN --last-value $LAST_VALUE --temporary-rootdir s3a://example-bucket/temporary-rootdir
```

Data from RDBMS can be imported into S3 in incremental append mode as Sequence or Avro file format. too

Parquet import into S3 in incremental append mode is also supported if the Parquet Hadoop API based implementation is used, meaning that the `--parquet-configurator-implementation` option is set to `hadoop`.

Example command: Import data into a target directory in an Amazon S3 bucket in incremental append mode as Parquet file.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incremental append --check-column $CHECK_COLUMN --last-value $LAST_VALUE --temporary-rootdir s3a://example-bucket/temporary-rootdir --as-parquetfile --parquet-configurator-implementation hadoop
```

Lastmodified Mode

When importing data into a target directory in an Amazon S3 bucket in incremental lastmodified mode, the location of the temporary root directory must be in the same bucket and in the same directory as the target directory. For example: `s3a://example-bucket/temporary-rootdir` in case of `s3a://example-bucket/target-directory`.

Example command: Import data into a target directory in an Amazon S3 bucket in incremental lastmodified mode.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incremental lastmodified --check-column $CHECK_COLUMN --merge-key $MERGE_KEY --last-value $LAST_VALUE --temporary-rootdir s3a://example-bucket/temporary-rootdir
```

Parquet import into S3 in incremental lastmodified mode is supported if the Parquet Hadoop API based implementation is used, meaning that the `--parquet-configurator-implementation` option is set to `hadoop`.

Example command: Import data into a target directory in an Amazon S3 bucket in incremental lastmodified mode as Parquet file.

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incremental lastmodified --check-column $CHECK_COLUMN --merge-key $MERGE_KEY --last-value $LAST_VALUE --temporary-rootdir s3a://example-bucket/temporary-rootdir --as-parquetfile --parquet-configurator-implementation hadoop
```

Import Data into an External Hive Table Backed by S3

The AWS credentials must be set in the Hive configuration file (hive-site.xml) to import data from RDBMS into an external Hive table backed by S3. The configuration file can be edited manually or by using the advanced configuration snippets.

Both `--target-dir` and `--external-table-dir` options have to be set. The `--external-table-dir` has to point to the Hive table location in the S3 bucket.

Parquet import into an external Hive table backed by S3 is supported if the Parquet Hadoop API based implementation is used, meaning that the `--parquet-configurator-implementation` option is set to `hadoop`.

Example Commands: Create an External Hive Table Backed by S3

Create an external Hive table backed by S3 using HiveServer2:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --create-hive-table --hs2-url $HS2_URL --hs2-user $HS2_USER --hs2-keytab $HS2_KEYTAB --hive-table $HIVE_TABLE_NAME --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory
```

Create an external Hive table backed by S3 using Hive CLI:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --create-hive-table --hive-table $HIVE_TABLE_NAME --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory
```

Create an external Hive table backed by S3 as Parquet file using Hive CLI:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --create-hive-table --hive-table $HIVE_TABLE_NAME --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory --as-parquetfile --parquet-configurator-implementation hadoop
```

Example Commands: Import Data into an External Hive Table Backed by S3

Import data into an external Hive table backed by S3 using HiveServer2:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --hs2-url $HS2_URL --hs2-user $HS2_USER --hs2-keytab $HS2_KEYTAB --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory
```

Import data into an external Hive table backed by S3 using Hive CLI:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory
```

Import data into an external Hive table backed by S3 as Parquet file using Hive CLI:

```
sqoop import --connect $CONN --username $USER --password $PWD --table $TABLE_NAME --hive-import --target-dir s3a://example-bucket/target-directory --external-table-dir s3a://example-bucket/external-directory --as-parquetfile --parquet-configurator-implementation hadoop
```