

CDP Public Cloud 7.2.2

Managing Clusters

Date published: 2020-07-10

Date modified: 2020-09-17

CLOU_DERA

<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.....	5
Accessing the Cloudera Manager for Data Lake clusters using workload credentials.....	5
Starting, Stopping, Refreshing, and Restarting a Cluster.....	6
Pausing a Cluster in AWS.....	7
Shutting Down and Starting Up the Cluster.....	7
Managing Hosts.....	9
Viewing Host Status.....	9
Viewing Host Role Assignments.....	10
Hosts Disks Overview.....	10
Stopping All the Roles on a Host.....	10
Starting All the Roles on a Host.....	10
Performing Maintenance on a Cluster Host.....	11
Decommissioning Hosts.....	11
Recommissioning Hosts.....	12
Tuning and Troubleshooting Host Decommissioning.....	13
Maintenance Mode.....	16
Viewing the Maintenance Mode Status of a Cluster.....	18
Managing Roles.....	19
Role Instances.....	20
Starting, Stopping, and Restarting Role Instances.....	20
Decommissioning Role Instances.....	20
Recommissioning Role Instances.....	21
Configuring Roles to Use a Custom Garbage Collection Parameter.....	21
Role Groups.....	22
Creating a Role Group.....	22
Managing Role Groups.....	23
Default User Roles.....	23
Managing Cloudera Runtime Services.....	24
Comparing Configurations for a Service Between Clusters.....	24
Starting a Cloudera Runtime Service on All Hosts.....	25
Stopping a Cloudera Runtime Service on All Hosts.....	26
Restarting a Cloudera Runtime Service.....	26
Rolling Restart.....	27

Aborting a Pending Command.....	29
Configuring Maximum File Descriptors.....	30
Core Configuration Service.....	30
Cloudera Manager API.....	31
Exporting and Importing Cloudera Manager Configuration.....	32
Backing Up and Restoring the Cloudera Manager Configuration.....	32
Using Tags in Cloudera Manager.....	33
Performance Management.....	36
Optimizing Performance in Cloudera Runtime.....	36
Disabling Transparent Hugepages (THP).....	36
Setting the vm.swappiness Linux Kernel Parameter.....	37
Improving Performance in Shuffle Handler and IFile Reader.....	38
Tips and Best Practices for Jobs.....	38
Decrease Reserve Space.....	39
Choosing and Configuring Data Compression.....	39
Resource Management.....	39
Static Service Pools.....	40
Data Storage for Monitoring Data.....	41
Configuring Service Monitor Data Storage.....	41
Configuring Host Monitor Data Storage.....	41
Viewing Host and Service Monitor Data Storage.....	41
Data Granularity and Time-Series Metric Data.....	42
Moving Monitoring Data on an Active Cluster.....	42
Host Monitor and Service Monitor Memory Configuration.....	43
Accessing Storage Using Amazon S3.....	44
Referencing S3 Credentials for YARN, MapReduce, or Spark Clients.....	44
Referencing Amazon S3 in URIs.....	46
Using Fast Upload with Amazon S3.....	46
Enabling Fast Upload using Cloudera Manager.....	46
How to Configure a MapReduce Job to Access S3 with an HDFS Credstore.....	47
Importing Data into Amazon S3 Using Sqoop.....	48
Authentication.....	49
Sqoop Import into Amazon S3.....	50
Accessing Storage Using Microsoft ADLS Gen 2.....	53
Configuring OAuth in Data Hub.....	53
Configuring OAuth with core-site.xml.....	53
Configuring OAuth with the Hadoop CredentialProvider.....	54
Configuring Native TLS Acceleration.....	55
Importing Data into Microsoft Azure Data Lake Store (Gen1 and Gen2) Using Sqoop.....	55
Prerequisites.....	56
Authentication.....	56
Sqoop Import into ADLS.....	56

Accessing the Cloudera Manager Admin Console

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 in a Web browser using the following link: `http://<cm-server-url>/`
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.

Starting, Stopping, Refreshing, and Restarting a Cluster

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

Complete the steps below to start, stop, refresh, and restart a cluster.

Starting a Cluster

1. On the HomeStatus tab, click  to the right of the cluster name and select Start.
2. Click Start that appears in the next screen to confirm. The Command Details window shows the progress of starting services.

When All services successfully started appears, the task is complete and you can close the Command Details window.



Note: The cluster-level Start action starts only Cloudera Runtime and other product services (Impala, Cloudera Search). It does not start the Cloudera Management Service. You must start the Cloudera Management Service separately if it is not already running.

Stopping a Cluster

1. On the HomeStatus tab, click  to the right of the cluster name and select Stop.
2. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services.

When All services successfully stopped appears, the task is complete and you can close the Command Details window.



Note: The cluster-level Stop action does not stop the Cloudera Management Service. You must stop the Cloudera Management Service separately.

Refreshing a Cluster

Runs a cluster refresh action to bring the configuration up to date without restarting all services. For example, certain masters (for example NameNode and ResourceManager) have some configuration files (for example, fair-scheduler.xml, mapred_hosts_allow.txt, topology.map) that can be refreshed. If anything changes in those files then a refresh can be used to update them in the master. Here is a summary of the operations performed in a refresh action:

✓ Refresh Cluster Cluster 1 Finished Mar 19, 2014 11:31:55 AM PDT Mar 19, 2014 11:32:09 AM PDT

Successfully refreshed roles in the cluster.

Command Progress

Completed 4 of 4 steps.

Run 1 steps in parallel
Successfully refreshed datanode allow/exclude lists.
[Details](#) ↗

Run 1 steps in parallel
Successfully refreshed ResourceManager.
[Details](#) ↗

Run 3 steps in parallel
Successfully refreshed NodeManager.
[Details](#) ↗

Run 3 steps in parallel
Refreshed Impala Daemon's Pools configuration and ACLs successfully.
[Details](#) ↗

To refresh a cluster, in the HomeStatus tab, click  to the right of the cluster name and select Refresh Cluster.

Restarting a Cluster

1. On the HomeStatus tab, click  to the right of the cluster name and select Restart.
2. Click Restart that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose Rolling Restart instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

When All services successfully started appears, the task is complete and you can close the Command Details window.

Pausing a Cluster in AWS

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

If all data for a cluster is stored on EBS volumes, you can pause the cluster and stop your AWS EC2 instances during periods when the cluster will not be used. The cluster will not be available while paused and can't be used to ingest or process data, but you won't be billed by Amazon for the stopped EC2 instances. Provisioned EBS storage volumes will continue to accrue charges.



Important: Pausing a cluster requires using EBS volumes for all storage, both on management and worker nodes. Data stored on ephemeral disks will be lost after EC2 instances are stopped.

Shutting Down and Starting Up the Cluster

To pause an AWS cluster, follow the shutdown procedure. To restart the cluster after a pause, follow the startup procedure.

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

In the shutdown and startup procedures below, some steps are performed in the AWS console and some are performed in Cloudera Manager:

- For AWS actions, use one of the following interfaces:
 - AWS console
 - AWS CLI
 - AWS API
- For cluster actions, use one of the following interfaces:
 - The Cloudera Manager web UI
 - The Cloudera API start and stop commands

Shutdown procedure

To pause the cluster, complete the following steps:

1. Navigate to the Cloudera Manager web UI.
2. Stop the cluster.
 - a. On the HomeStatus tab, click  to the right of the cluster name and select Stop.
 - b. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services. When All services successfully stopped appears, the task is complete and you can close the Command Details window.
3. Stop the Cloudera Management Service.
 - a. On the HomeStatus tab, click  to the right of the service name and select Stop.
 - b. Click Stop in the next screen to confirm. When you see a Finished status, the service has stopped.
4. In AWS, stop all cluster EC2 instances, including the Cloudera Manager host .

Startup procedure

To restart the cluster after a pause, the steps are reversed:

1. In AWS, start all cluster EC2 instances.
2. Navigate to the Cloudera Manager UI.
3. Start the Cloudera Management Service.
 - a. On the HomeStatus tab, click  to the right of the service name and select Start.
 - b. Click Start that appears in the next screen to confirm. When you see a Finished status, the service has started.
4. Start the cluster.
 - a. On the HomeStatus tab, click  to the right of the cluster name and select Start.
 - b. Click Start that appears in the next screen to confirm. The Command Details window shows the progress of starting services. When All services successfully started appears, the task is complete and you can close the Command Details window.

Considerations after Restart

Since the cluster was completely stopped before stopping the EC2 instances, the cluster should be healthy upon restart and ready for use. You should be aware of the following about the restarted cluster:

- After starting the EC2 instances, Cloudera Manager and its agents will be running but the cluster will be stopped. There will be gaps in Cloudera Manager's time-based metrics and charts.

- EC2 instances retain their internal IP address and hostname for their lifetime, so no reconfiguration of CDH or Runtime is required after restart. The public IP and DNS hostnames, however, will be different. Elastic IPs can be configured to remain associated with a stopped instance at additional cost, but it isn't necessary to maintain proper cluster operation.

Managing Hosts

How to use Cloudera Manager to configure and manage the hosts in your clusters.

Viewing Host 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 HostsAll 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. Click the Filters toggle to show or hide the Filters section.
- 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.

Viewing Host 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.

Hosts Disks Overview

How to view the status of all disks in a cluster.

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.

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:

1. Decommission the hosts.
2. Perform the necessary maintenance on the hosts.
3. Recommission the 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. If the host has a DataNode, and you are planning to replicate data to other hosts (for longer term maintenance operations or to permanently decommission or repurpose the host), perform the steps in [Tuning HDFS Prior to Decommissioning DataNodes](#).
2. In Cloudera Manager, select the cluster where you want to decommission hosts.
3. In the left menu, click HostsAll Hosts.
4. Select the hosts that you want to decommission.

5. 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.

6. 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 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.

7. 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 HostsAll Hosts.
3. Select the hosts that you want to recommission.

4. Select Actions for Selected End 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, and 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.

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 UINameNode 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    DFSClient_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 '/^\s*\/\s*{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.

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

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, go to 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.

About this task

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator) This feature is not available when using Cloudera Manager to manage Data Hub clusters.

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

About this task

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

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

[Performance Considerations](#)

[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

About this task

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

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

[Performance Considerations](#)

[Modifying Configuration Properties Using Cloudera Manager](#)

Managing Role Groups

About this task

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

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](#)

Default User Roles

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

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

Permitted Operations	Auditor	Cluster Administrator	Cluster Creator	Configurator	Dashboard User	Full Administrator	Key Administrator	Limited Cluster Administrator	Limited Operator	Navigator Administrator	Operator	Read-Only	Replication Administrator	User Administrator
Access all functionality that Cloudera Manager offers		Y				Y								
Add and Remove Entity Tags		Y				Y		Y						
Administer Cloudera Navigator		Y				Y				Y				
Apply policies to redact sensitive data		Y				Y								
Configure HDFS Encryption, administer Key Trustee Server, and manage encryption keys						Y	Y							
Create clusters		Y	Y			Y								
Create replication policies and snapshot policies						Y							Y	
Create, modify, and delete your own dashboards					Y	Y								
Create, update, or delete external account configuration						Y								Y
Decommission hosts		Y		Y		Y		Y	Y		Y			
Edit the configuration of services and roles		Y		Y		Y		Y						
Enter and exit Maintenance Mode		Y		Y		Y		Y						
Import Cluster Template		Y				Y		Y						
Inspect Hosts		Y				Y		Y						
Manage Full Administrator accounts						Y								
Manage user accounts and configuration of external authentication						Y								Y
Recommission hosts, and decommission and recommission roles		Y		Y		Y		Y			Y			
See available hosts		Y	Y			Y		Y						
Send Diagnostic Bundles		Y				X		Y						
Start, stop, and restart KMS		Y		Y		Y	Y	Y			Y			
Start, stop, and restart most clusters, services, and roles		Y		Y		Y		Y			Y			
Upgrade Clusters		Y				Y								
View and perform parcels operations		Y	Y			Y		Y						
View audit events	Y					Y				Y				
View data in Cloudera Manager	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Historical Disk Usage By Directory	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Directory Usage		Y				Y								
File Browser		Y				Y								

Managing Cloudera Runtime Services

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

You can add new services and roles if needed, gracefully start, stop and restart services or roles, and decommission and delete roles or services if necessary. Further, you can modify the configuration properties for services or for individual role instances. You can also view past configuration changes and roll back to a previous revision. 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.

Comparing Configurations for a Service Between Clusters

You can compare the configuration settings for a particular service between two different clusters in a Cloudera Manager deployment.

About this task

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

Procedure

1. On the HomeStatus tab, click the name of the service you want to compare, or click the Clusters menu and select the name of the service.

- 2.
3. Click the drop-down menu above the Filters pane, and select from one of the options that begins Diff with...:
 - *service on cluster* - For example, HBASE-1 on Cluster 1. This is the default display setting. All properties are displayed for the selected instance of the service.
 - *service on all clusters* - For example, HBase on all clusters. All properties are displayed for all instances of the service.
 - Diff with *service on cluster* - For example, Diff with HBase on Cluster 2. Properties are displayed only if the values for the instance of the service whose page you are on differ from the values for the instance selected in the drop-down menu.
 - Diff with *service on all clusters* - For example, Diff with HBase on all clusters. Properties are displayed if the values for the instance of the service whose page you are on differ from the values for one or more other instances in the Cloudera Manager deployment.

The service's properties will be displayed showing the values for each property for the selected clusters. The filters on the left side can be used to limit the properties displayed.

You can also view property configuration values that differ between clusters across a deployment by selecting Non-uniform Values on the Configuration tab of the Cloudera Manager HomeStatus page.

Starting a Cloudera Runtime Service on All Hosts

Starting and Stopping Cloudera Runtime services.

About this task

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

It is important to start and stop services that have dependencies in the correct order. For example, because MapReduce and YARN have a dependency on HDFS, you must start HDFS before starting MapReduce or YARN. The Cloudera Management Service and Hue are the only two services on which no other services depend; although you can start and stop them at anytime, their preferred order is shown in the following procedures. The Cloudera Manager cluster actions start and stop services in the correct order. To start or stop all services in a cluster, follow the instructions in Starting, Stopping, Refreshing, and Restarting a Cluster.

Before you begin

The order in which to start services is:

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

About this task

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

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

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 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.

If you have [HDFS High Availability](#) enabled, you can also perform a cluster-level rolling restart. At the cluster level, the rolling restart of worker hosts is performed on a host-by-host basis, rather than per service, to avoid all roles for a service potentially being unavailable at the same time. During a cluster restart, to avoid having your NameNode (and thus the cluster) be unavailable during the restart, Cloudera Manager forces a failover to the standby NameNode.

Job Tracker and Resource Manager High availability are not required for a cluster-level rolling restart. However, if you have JobTracker or ResourceManager high availability enabled, Cloudera Manager will force a failover to the standby JobTracker or ResourceManager.

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.

Performing a Cluster-Level Rolling Restart

You can perform a cluster-level rolling restart on demand from the Cloudera Manager Admin Console. A cluster-level rolling restart is also performed as the last step in a rolling upgrade when the cluster is configured with HDFS high availability enabled.

1. If you have not already done so, enable high availability. See [HDFS High Availability](#) for instructions. You do not need to enable automatic failover for rolling restart to work, though you can enable it if you want. Automatic failover does not affect the rolling restart operation.
2. For the cluster you want to restart select ActionsRolling Restart.
3. In the pop-up dialog box, select the services you want to restart. Please review the caveats in the preceding section for the services you elect to have restarted. The services that do not support rolling restart will simply be restarted, and will be unavailable during their restart.
4. If you select an HDFS, HBase, or MapReduce 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.
5. Click Restart to start the rolling restart. While the restart is in progress, the Command Details page shows the steps for stopping and restarting the services.

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](#)

Configuring Maximum File Descriptors

You can set the maximum file descriptor parameter for all daemon roles. When not specified, the role uses whatever value it inherits from supervisor. When specified, configures soft and hard limits to the configured value.

About this task

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

Procedure

1. Go to a service.
2. Click the **Configuration** tab.
3. In the Search box, type `rlimit_fds`.
4. Set the Maximum Process File Descriptors property for one or more roles.
5. Enter a Reason for change, and then click Save Changes to commit the changes.
6. Restart the affected role instances.

Core Configuration Service

The Core Configuration Service allows you to create clusters without the HDFS service.

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

The Core Configuration service allows you to create more types of clusters without having to include the HDFS service. Previously, the HDFS service was required in many cases even when data was not being stored in HDFS because some services like Sentry and Spark required cluster-wide configuration files that Cloudera Manager deploys within the HDFS service. The Core Configuration service provides this configuration in a standalone fashion and thus eliminates the need for an HDFS service for certain types of clusters where no HDFS storage is required (e.g. Kudu, Kafka, or 'Compute' clusters using exclusively object storage like S3 or ADLS). The Core Configuration service is also useful when creating a Compute cluster that accesses data on an HDFS service located in the Base cluster.

You can add the Core Configuration service in the following ways:

- When installing a Cloudera Runtime cluster using the installation wizard you can select the Core Configuration from the list of services.
- When adding a new cluster using the Add Cluster wizard, you can select the Core Configuration from the list of services.
- You can add the Core Configuration service to an existing cluster that does not include HDFS:
 1. Open the Cloudera Manager Admin Console and navigate to the cluster where you want to add the service.
 2. Click Actions > Add Service
 3. Select Core Configuration from the list of services.
 4. Click Continue.
 5. Follow the prompts on the screen to complete adding the service.



Important: You cannot use both the HDFS service and the Core Configuration service in the same cluster.

Cloudera Manager API

The Cloudera Manager API provides configuration and service lifecycle management, service health information and metrics, and allows you to configure Cloudera Manager itself. The API is served on the same host and port as the Cloudera Manager Admin Console, and does not require an extra process or extra configuration. The API supports HTTP Basic Authentication, accepting the same users and credentials as the Cloudera Manager Admin Console.

You can also access the Cloudera Manager Swagger API user interface from the Cloudera Manager Admin Console. Go to SupportAPI Explorer to open Swagger.

Cloudera Manager API Resources

[Cloudera Manager REST API documentation](#)

API timeseries endpoint: [Cloudera Manager API documentation timeseries Endpoint](#)

API Version v41

Code API Version v41

[/cm/deployment](#)

[Javadoc](#)

[Cloudera Manager API tutorial](#)

Obtaining Configuration Files

1. Obtain the list of a service's roles:

```
http://cm_server_host:7180/api/v41/clusters/clusterName/services/serviceName/roles
```

2. Obtain the list of configuration files a process is using:

```
http://cm_server_host:7180/api/v41/clusters/clusterName/services/serviceName/roles/roleName/process
```

3. Obtain the content of any particular file:

For example:

```
http://cm_server_host:7180/api/v41/clusters/Cluster%201/services/OOZIE-1/roles/OOZIE-1-OOZIE_SERVER-e121641328fcb107999f2b5fd856880d/process/configFiles/oozie-site.xml
```

Similar to finding service properties, you can also find host properties. First, get the host IDs for a cluster with the URL:

```
http://cm_server_host:7180/api/v41/hosts
```

This should return host objects of the form:

```
{
  "hostId" : "2c2e951c-aaf2-4780-a69f-0382181f1821",
  "ipAddress" : "10.30.195.116",
  "hostname" : "cm_server_host",
```

```

    "rackId" : "/default",
    "hostUrl" : "http://cm_server_host:7180/cm/hostRedirect/2c2e951c-adf2-4780-a69f-0382181f1821",
    "maintenanceMode" : false,
    "maintenanceOwners" : [ ],
    "commissionState" : "COMMISSIONED",
    "numCores" : 4,
    "totalPhysMemBytes" : 10371174400
  }

```

Then obtain the host properties by including one of the returned host IDs in the URL:

```

http://cm_server_host:7180/api/v41/hosts/2c2e951c-adf2-4780-a69f-0382181f1821?view=FULL

```

Exporting and Importing Cloudera Manager Configuration

You can use the Cloudera Manager API to programmatically export and import a definition of all the entities in your Cloudera Manager-managed deployment—clusters, service, roles, hosts, users and so on.

See the Cloudera Manager API documentation on how to manage deployments using the [/cm/deployment](#) resource.

Backing Up and Restoring the Cloudera Manager Configuration

You can use the Cloudera Manager REST API to export and import all of its configuration data.

To export configurations, the Cloudera Manager API exports a JSON document that contains configuration data for the Cloudera Manager instance. You can use this JSON document to back up and restore a Cloudera Manager deployment.

Minimum Required Role: [Cluster Administrator](#) (also provided by Full Administrator) This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Exporting the Cloudera Manager Configuration

1. Log in to the Cloudera Manager server host as the root user.
2. Run the following command:

```

# curl -u admin_uname:admin_pass "http://cm_server_host:7180/api/v41/cm/deployment" >
path_to_file/cm-deployment.json

```

Where:

- *admin_uname* is a username with either the Full Administrator or Cluster Administrator role.
- *admin_pass* is the password for the *admin_uname* username.
- *cm_server_host* is the hostname of the Cloudera Manager server.
- *path_to_file* is the path to the file where you want to save the configuration.

Redacting Sensitive Information from the Exported Configuration

The exported configuration may contain passwords and other sensitive information. The Cloudera Manager API automatically redacts the sensitive items returned from API calls.



Important: Due to this redaction, you cannot use an exported configuration to restore the configuration of your cluster due to the redacted information.

To disable redaction for the Cloudera Manager API:

1. Log in the Cloudera Manager server host.
2. Edit the `/etc/default/cloudera-scm-server` file by adding the following property (separate each property with a space) to the line that begins with `export CMF_JAVA_OPTS`:

```
-Dcom.cloudera.api.redaction=true
```

For example:

```
export CMF_JAVA_OPTS="-Xmx2G -Dcom.cloudera.api.redaction=false"
```

3. Restart Cloudera Manager:

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

Restoring the Cloudera Manager Configuration

Using a previously saved JSON document that contains the Cloudera Manager configuration data, you can restore that configuration to a running cluster.

1. Using the Cloudera Manager Administration Console, stop all running services in your cluster:

- a. On the HomeStatus tab, click  to the right of the cluster name and select Stop.
- b. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services. When All services successfully stopped appears, the task is complete and you can close the Command Details window.



Warning: If you do not stop the cluster before making this API call, the API call will stop all cluster services before running the job. Any running jobs and data are lost.

2. Log in to the Cloudera Manager server host as the root user.
3. Run the following command:

```
curl -H "Content-Type: application/json" --upload-file path_to_file/cm-deployment.json -u admin:admin http://cm_server_host:7180/api/v41/cm/deployment?deleteCurrentDeployment=true
```

Where:

- *admin_uname* is a username with either the Full Administrator or Cluster Administrator role.
 - *admin_pass* is the password for the *admin_uname* username.
 - *cm_server_host* is the hostname of the Cloudera Manager server.
 - *path_to_file* is the path to the file containing the JSON configuration file.
4. Restart the Cloudera Manager Server.

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04 and higher

```
sudo systemctl restart cloudera-scm-server
```

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

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

Using Tags in Cloudera Manager

Tags provide a way to add information to clusters, hosts, services, and roles using the Cloudera Manager API.

A tag in the context of Cloudera Manager is a name-value pair that is attached to a cluster, service, role, or host. Tags allow information to be added to an entity in a very open-ended fashion. A single entity may have any number of tags, as long as each tag name is unique, and tags may be added or removed at any time. All tags appear in diagnostic bundles.

Tags may be created by various cluster components, using the following naming convention: `_cldr_{component_specific_prefix}_tagname`. You can also use the Cloudera Manager API to create, update, delete, and retrieve tags.

Setting and Retrieving Tags using the Cloudera Manager API

A `TagsResource` endpoint for retrieving tags globally is available in versions 41 and higher of the Cloudera Manager API. Endpoints to read, delete, and add or modify tags on clusters, services, hosts and roles are also available. For complete documentation, open the Cloudera Manager Admin Console and go to Support API Explorer.

You can manage tags for services, roles, and hosts using the following endpoints:

Clusters

`/clusters/{clusterName}/tags`

Hosts

`/hosts/{hostname}/tags`

Services

`/clusters/{clusterName}/services/<service-name>/tags`

Roles

`/clusters/{clusterName}/services/{serviceName}/roles/{roleName}/tags`

Retrieve all tags:

`/tags`

API Examples

The following examples show how to add, modify, or delete tags for a cluster:

Add or update a tag on a cluster

This example adds a tag with the name `example_name` and value `example_value` to `Cluster1`. If a tag with the name `example_name` is already attached to `Cluster1`, the value of that tag will be updated to `example_value`.

```
curl -X PUT --header 'Content-Type: application/json' --header '
Accept: application/json' -d '[
  {
    "name": "example_name",
    "value": "example_value"
  }
]' 'http://username:password@<Cloudera Manager_URL>:7180/api/v41/
clusters/<cluster-name>/tags'
```

Delete a tag on a cluster

This example deletes the tag named `example_name` from `Cluster1`, if the tag exists.

```
curl -X DELETE --header 'Content-Type: application/json' --header
'Accept: application/json' -d '[
  {
    "name": "example_name"
  }
]' 'http://username:password@<Cloudera Manager_URL>:7180/a
pi/v41/clusters/<cluster-name>/tags'
```

Retrieve all tags

This example returns a serialized JSON list of all tags, including which entities each is attached to, with a maximum of 10 tags and beginning with the most recently created tag.

```
curl -X GET --header 'Accept: application/json' http
://username:password@<Cloudera Manager_URL>:7180/api/v41/tags?li
mit=10&offset=0'
```

Setting Tags using ImportClusterTemplate (Example)

You can set tags on clusters, services, and hosts (but not roles) using a cluster template provided to /cm/importClusterTemplate in the Cloudera Manager API versions 41 and higher.

The following is an example of a template using tags:

```
{
  "cdhVersion" : "7.0.2",
  "displayName" : "Cluster 2",
  "cmVersion" : "7.0.2",
  "tags" : [ {
    "name" : "_cldr_cm_clustertag",
    "value" : "example1"
  } ],
  "services" : [ {
    "refName" : "hdfs",
    "serviceType" : "HDFS",
    "tags" : [ {
      "name" : "_cldr_cm_servicetag",
      "value" : "example2"
    } ],
    "roleConfigGroups" : [ {
      "refName" : "hdfs-NAMENODE-BASE",
      "roleType" : "NAMENODE",
      "base" : true
    },
    {
      "refName" : "hdfs-DATANODE-BASE",
      "roleType" : "DATANODE",
      "base" : true
    } ]
  } ],
  "hostTemplates" : [ {
    "refName" : "HostTemplate1",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "hdfs-NAMENODE-BASE" ]
  }, {
    "refName" : "HostTemplate2",
    "cardinality" : 1,
    "roleConfigGroupsRefNames" : [ "hdfs-DATANODE-BASE" ]
  } ],
  "instantiator" : {
    "clusterName" : "Cluster 2",
    "hosts" : [ {
      "hostName" : "host00003",
      "hostTemplateRefName" : "HostTemplate1"
    },
    {
      "hostName" : "host00004",
```

```

    "hostTemplateRefName" : "HostTemplate2"
  }],
  "lenient" : false
}
}

```

This adds the `_cldr_cm_clustertag` to the cluster that is being created, the `_cldr_cm_servicetag` to the HDFS service on that cluster, and the `_cldr_cm_hosttag` to `host00003`. Note that even if the host templates are persisted (which is optional during cluster template import), the `_cldr_cm_hosttag` will no longer be associated with `HostTemplate1`, only with `host00003` itself.

Performance Management

This section describes mechanisms and best practices for improving performance.

Optimizing Performance in Cloudera Runtime

This section provides solutions to some performance problems, and describes configuration best practices.



Important: Work with your network administrators and hardware vendors to ensure that you have the proper NIC firmware, drivers, and configurations in place and that your network performs properly. Cloudera recognizes that network setup and upgrade are challenging problems, and will do its best to share useful experiences.

Disabling Transparent Hugepages (THP)

Most Linux platforms supported by Cloudera Runtime include a feature called *transparent hugepages*, which interacts poorly with Hadoop workloads and can seriously degrade performance.

About this task

Symptom: `top` and other system monitoring tools show a large percentage of the CPU usage classified as "system CPU". If system CPU usage is 30% or more of the total CPU usage, your system may be experiencing this issue.

To see whether transparent hugepages are enabled, run the following commands and check the output:

```

$ cat defrag_file_pathname
$ cat enabled_file_pathname

```

- `[always]` never means that transparent hugepages is enabled.
- `always [never]` means that transparent hugepages is disabled.

To disable Transparent Hugepages, perform the following steps on all cluster hosts:

Procedure

1. (Required for hosts running RHEL/CentOS 7.x.) To disable transparent hugepages on reboot, add the following commands to the `/etc/rc.d/rc.local` file on all cluster hosts:

- RHEL/CentOS 7.x:

```

echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag

```

- RHEL/CentOS 6.x

```

echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled

```

- Ubuntu/Debian, OL, SLES:

```
echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

Modify the permissions of the rc.local file:

```
chmod +x /etc/rc.d/rc.local
```

2. If your cluster hosts are running RHEL/CentOS 7.x, modify the GRUB configuration to disable THP:
 - a) Add the following line to the GRUB_CMDLINE_LINUX options in the /etc/default/grub file:

```
transparent_hugepage=never
```

- b) Run the following command:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Disable the tuned service, as described above.

You can also disable transparent hugepages interactively (but remember this will not survive a reboot).

To disable transparent hugepages temporarily as root:

```
# echo 'never' > defrag_file_pathname
# echo 'never' > enabled_file_pathname
```

To disable transparent hugepages temporarily using sudo:

```
$ sudo sh -c "echo 'never' > defrag_file_pathname"
$ sudo sh -c "echo 'never' > enabled_file_pathname"
```

Setting the vm.swappiness Linux Kernel Parameter

The Linux kernel parameter, `vm.swappiness`, is a value from 0-100 that controls the swapping of application data (as anonymous pages) from physical memory to virtual memory on disk. You can set the value of the `vm.swappiness` parameter for minimum swapping.

The higher the parameter value, the more aggressively inactive processes are swapped out from physical memory. The lower the value, the less they are swapped, forcing filesystem buffers to be emptied.

On most systems, `vm.swappiness` is set to 60 by default. This is not suitable for Hadoop clusters because processes are sometimes swapped even when enough memory is available. This can cause lengthy garbage collection pauses for important system daemons, affecting stability and performance.

Cloudera recommends that you set `vm.swappiness` to a value between 1 and 10, preferably 1, for minimum swapping on systems where the RHEL kernel is 2.6.32-642.el6 or higher.

To view your current setting for `vm.swappiness`, run:

```
cat /proc/sys/vm/swappiness
```

To set `vm.swappiness` to 1, run:

```
sudo sysctl -w vm.swappiness=1
```

Swap space allocation

Cloudera recommends following the guidelines provided by your operating system vendor to configure the swap space on each host. If your vendor recommends a swap space range, then use the lowest recommended value.



Note: If your operating system vendor does not have a recommendation for swap space, then search online for "*Linux default swap space*" and see the recommendation provided by [Red Hat](#).

Improving Performance in Shuffle Handler and IFile Reader

The MapReduce shuffle handler and IFile reader use native Linux calls, (`posix_fadvise(2)` and `sync_data_range`), on Linux systems with Hadoop native libraries installed.

Shuffle Handler

You can improve MapReduce shuffle handler performance by enabling shuffle readahead. This causes the TaskTracker or Node Manager to pre-fetch map output before sending it over the socket to the reducer.

- To enable this feature for YARN, set `mapreduce.shuffle.manage.os.cache`, to true (default). To further tune performance, adjust the value of `mapreduce.shuffle.readahead.bytes`. The default value is 4 MB.
- To enable this feature for MapReduce, set the `mapred.tasktracker.shuffle.fadvise` to true (default). To further tune performance, adjust the value of `mapred.tasktracker.shuffle.readahead.bytes`. The default value is 4 MB.

IFile Reader

Enabling IFile readahead increases the performance of merge operations. To enable this feature for either MRv1 or YARN, set `mapreduce.ifile.readahead` to true (default). To further tune the performance, adjust the value of `mapreduce.ifile.readahead.bytes`. The default value is 4MB.

Tips and Best Practices for Jobs

This section describes changes you can make at the job level.

Use the Distributed Cache to Transfer the Job JAR

Use the distributed cache to transfer the job JAR rather than using the `JobConf(Class)` constructor and the `JobConf.setJar()` and `JobConf.setJarByClass()` methods.

To add JARs to the classpath, use `-libjars jar1.jar2`. This copies the local JAR files to HDFS and uses the distributed cache mechanism to ensure they are available on the task nodes and added to the task classpath.

The advantage of this, over `JobConf.setJar`, is that if the JAR is on a task node, it does not need to be copied again if a second task from the same job runs on that node, though it will still need to be copied from the launch machine to HDFS.



Note: `-libjars` works only if your MapReduce driver uses `ToolRunner`. If it does not, you would need to use the `DistributedCache` APIs (Cloudera does not recommend this).

For more information, see item 1 in the blog post *How to Include Third-Party Libraries in Your MapReduce Job*.

Changing the Logging Level on a Job (MRv1)

You can change the logging level for an individual job. You do this by setting the following properties in the job configuration:

- `mapreduce.map.log.level`
- `mapreduce.reduce.log.level`

Valid values are NONE, INFO, WARN, DEBUG, TRACE, and ALL.

Example:

```
JobConf conf = new JobConf();
...

conf.set("mapreduce.map.log.level", "DEBUG");
conf.set("mapreduce.reduce.log.level", "TRACE");
...
```

Decrease Reserve Space

By default, the ext3 and ext4 filesystems reserve 5% space for use by the root user. This reserved space counts as Non DFS Used.

To view the reserved space use the tune2fs command:

```
# tune2fs -l /dev/sde1 | egrep "Block size:|Reserved block count"
Reserved block count: 36628312
Block size: 4096
```

The Reserved block count is the number of ext3/ext4 filesystem blocks that are reserved. The block size is the size in bytes. In this example, 150 GB (139.72 Gigabytes) are reserved on this filesystem.

Cloudera recommends reducing the root user block reservation from 5% to 1% for the DataNode volumes. To set reserved space to 1% with the tune2fs command:

```
# tune2fs -m 1 /dev/sde1
```

Choosing and Configuring Data Compression

Guidelines for compression types.

Guidelines for Choosing Data Compression

- GZIP compression uses more CPU resources than Snappy or LZO, but provides a higher compression ratio. GZip is often a good choice for cold data, which is accessed infrequently. Snappy or LZO are a better choice for hot data, which is accessed frequently.
- BZip2 can also produce more compression than GZip for some types of files, at the cost of some speed when compressing and decompressing. HBase does not support BZip2 compression.
- Snappy often performs better than LZO. It is worth running tests to see if you detect a significant difference.

Related Information

[Hadoop File Formats Support](#)

Resource Management

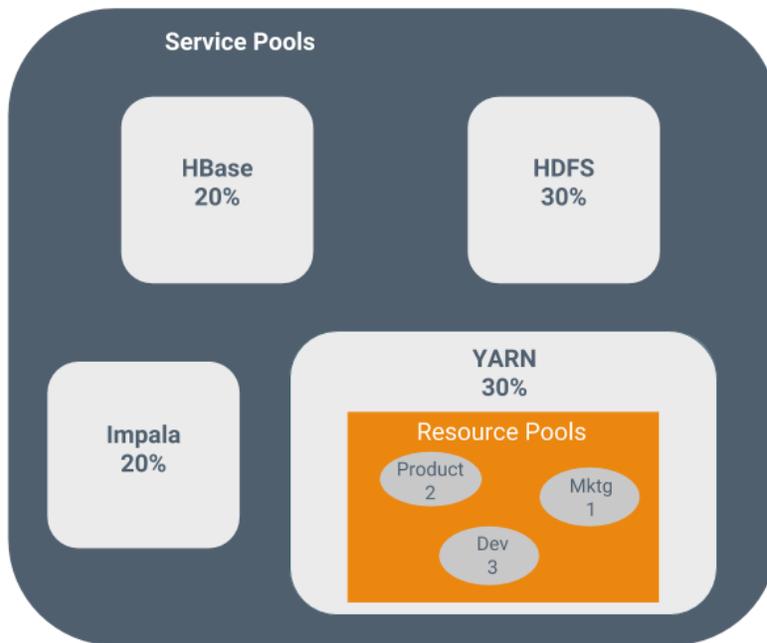
Resource management helps ensure predictable behavior by defining the impact of different services on cluster resources.

Use resource management to:

- Guarantee completion in a reasonable time frame for critical workloads.
- Support reasonable cluster scheduling between groups of users based on fair allocation of resources per group.
- Prevent users from depriving other users access to the cluster.

Statically allocating resources using cgroups is configurable through a single static service pool wizard. You allocate services as a percentage of total resources, and the wizard configures the cgroups.

For example, the following figure illustrates static pools for HBase, HDFS, Impala, and YARN services that are respectively assigned 20%, 30%, 20%, and 30% of cluster resources.



You can dynamically apportion resources that are statically allocated to YARN and Impala by using dynamic resource pools.

Depending on the version of CDH you are using, dynamic resource pools in Cloudera Manager support the following scenarios:

- **YARN** - YARN manages the virtual cores, memory, running applications, maximum resources for undeclared children (for parent pools), and scheduling policy for each pool. In the preceding diagram, three dynamic resource pools—Dev, Product, and Mktg with weights 3, 2, and 1 respectively—are defined for YARN. If an application starts and is assigned to the Product pool, and other applications are using the Dev and Mktg pools, the Product resource pool receives $30\% \times \frac{2}{6}$ (or 10%) of the total cluster resources. If no applications are using the Dev and Mktg pools, the YARN Product pool is allocated 30% of the cluster resources.
- **Impala** - Impala manages memory for pools running queries and limits the number of running and queued queries in each pool.

Static Service Pools

Static service pools isolate the services in your cluster from one another, so that load on one service has a bounded impact on other services.

Services are allocated a static percentage of total resources—CPU, memory, and I/O weight—which are not shared with other services. When you configure static service pools, Cloudera Manager computes recommended memory, CPU, and I/O configurations for the worker roles of the services that correspond to the percentage assigned to each service. Static service pools are implemented per role group within a cluster, using Linux control groups (cgroups) and cooperative memory limits (for example, Java maximum heap sizes). Static service pools can be used to control access to resources by HBase, HDFS, Impala, MapReduce, Solr, Spark, YARN, and add-on services. Static service pools are not enabled by default.



Note:

- I/O allocation only works when short-circuit reads are enabled.
- I/O allocation does not handle write side I/O because cgroups in the Linux kernel do not currently support buffered writes.

Viewing Static Service Pool Status

Select Clusters *Cluster name* Static Service Pools. If the cluster has a YARN service, the Static Service Pools Status tab displays and shows whether resource management is enabled for the cluster, and the currently configured service pools.

Data Storage for Monitoring Data

The Service Monitor and Host Monitor roles in the Cloudera Management Service store time series data, health data, and Impala query and YARN application metadata.

Configuring Service Monitor Data Storage

The Service Monitor stores time series data and health data, Impala query metadata, and YARN application metadata.

By default, the data is stored in `/var/lib/cloudera-service-monitor/` on the Service Monitor host. You can change this by modifying the Service Monitor Storage Directory configuration (`firehose.storage.base.directory`). To change this configuration on an active system, see *Moving Monitoring Data on an Active Cluster*.

You can control how much disk space to reserve for the different classes of data the Service Monitor stores by changing the following configuration options:

- Time-series metrics and health data - Time-Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)
- Impala query metadata - Impala Storage (`firehose_impala_storage_bytes` - 1 GB default)
- YARN application metadata - YARN Storage (`firehose_yarn_storage_bytes` - 1 GB default)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see [Data Granularity and Time-Series Metric Data](#).

The default values are small, so you should examine disk usage after several days of activity to determine how much space is needed.

Configuring Host Monitor Data Storage

The Host Monitor stores time series data and health data.

By default, the data is stored in `/var/lib/cloudera-host-monitor/` on the Host Monitor host. You can change this by modifying the Host Monitor Storage Directory configuration. To change this configuration on an active system, follow the procedure in *Moving Monitoring Data on an Active Cluster*.

You can control how much disk space to reserve for Host Monitor data by changing the following configuration option:

- Time-series metrics and health data: Time Series Storage (`firehose_time_series_storage_bytes` - 10 GB default, 10 GB minimum)

For information about how metric data is stored in Cloudera Manager and how storage limits impact data retention, see [Data Granularity and Time-Series Metric Data](#).

The default value is small, so you should examine disk usage after several days of activity to determine how much space they need. The Charts Library tab on the Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data.

Viewing Host and Service Monitor Data Storage

The Cloudera Management Service page shows the current disk space consumed and its rate of growth, categorized by the type of data stored. For example, you can compare the space consumed by raw metric data to daily summaries of that data.

Procedure

1. Select ClustersCloudera Management Service.
2. Click the Charts Library tab.

Data Granularity and Time-Series Metric Data

The Service Monitor and Host Monitor store time-series metric data in a variety of ways.

When the data is received, it is written as-is to the metric store. Over time, the raw data is summarized to and stored at various data granularities. For example, after ten minutes, a summary point is written containing the average of the metric over the period as well as the minimum, the maximum, the standard deviation, and a variety of other statistics. This process is summarized to produce hourly, six-hourly, daily, and weekly summaries. This data summarization procedure applies only to metric data. When the Impala query and YARN application monitoring storage limit is reached, the oldest stored records are deleted.

The Service Monitor and Host Monitor internally manage the amount of overall storage space dedicated to each data granularity level. When the limit for a level is reached, the oldest data points at that level are deleted. Metric data for that time period remains available at the lower granularity levels. For example, when an hourly point for a particular time is deleted to free up space, a daily point still exists covering that hour. Because each of these data granularities consumes significantly less storage than the previous summary level, lower granularity levels can be retained for longer periods of time. With the recommended amount of storage, weekly points can often be retained indefinitely.

Some features, such as detailed display of health results, depend on the presence of raw data. Cluster utilization reports depend on hourly data being available for the selected time range. Charts built from weekly data might show a large gap between the last data point and the current time, as the next data point is not available yet. Other granularity levels may exhibit a similar gap, due in part to delays in the summarization process. These gaps may coincide with another one on the other side of the chart if metric history is too short to cover the selected time range. Health history is maintained by the event store dictated by its retention policies.

Moving Monitoring Data on an Active Cluster

You can change where monitoring data is stored on a cluster.

Basic: Changing the Configured Directory

1. Stop the Service Monitor or Host Monitor.
2. Save your old monitoring data and then copy the current directory to the new directory (optional).
3. Update the Storage Directory configuration option (`firehose.storage.base.directory`) on the corresponding role configuration page.
4. Start the Service Monitor or Host Monitor.

Advanced: High Performance

For the best performance, and especially for a large cluster, Host Monitor and Service Monitor storage directories should have their own dedicated spindles. In most cases, that provides sufficient performance, but you can divide your data further if needed. You cannot configure this directly with Cloudera Manager; instead, you must use symbolic links.

For example, if all your Service Monitor data is located in `/data/1/service_monitor`, and you want to separate your Impala data from your time series data, you could do the following:

1. Stop the Service Monitor.
2. Move the original Impala data in `/data/1/service_monitor/impala` to the new directory, for example `/data/2/impala_data`.
3. Create a symbolic link from `/data/1/service_monitor/impala` to `/data/2/impala_data` with the following command:

```
ln -s /data/2/impala_data /data/1/service_monitor/impala
```

4. Start the Service Monitor.

Host Monitor and Service Monitor Memory Configuration

You can configure Java heap size and non-Java memory size. The memory recommended for these configuration options depends on the number of hosts in the cluster, the services running on the cluster, and the number of monitored entities.

Monitored entities are the objects monitored by the Service Monitor or Host Monitor. As the number of hosts and services increases, the number of monitored entities also increases.

In addition to the memory configured, the Host Monitor and Service Monitor use the Linux page cache. Memory available for page caching on the Host Monitor and Service Monitor hosts improves performance.

Configuring Memory Allocations

To configure memory allocations, determine how many entities are being monitored and then consult the tables below for required and recommended memory configurations.

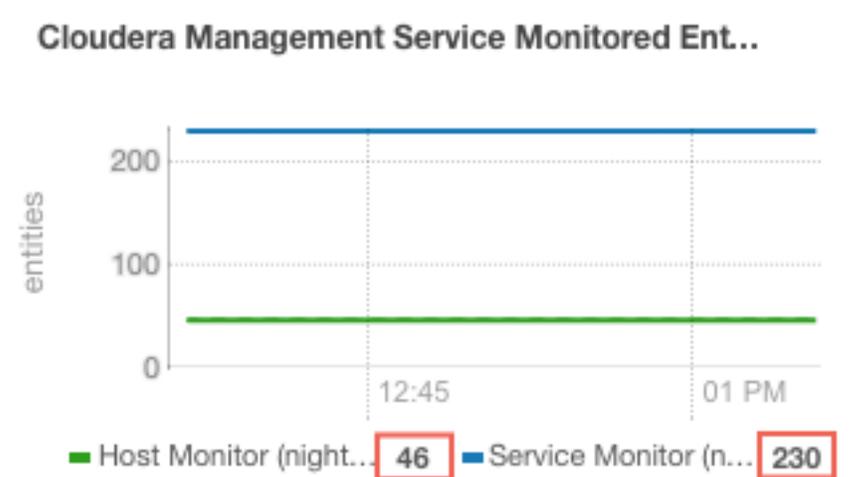
About this task

To determine the number of entities being monitored:

Procedure

1. Go to ClustersCloudera Management Service.
2. Locate the chart with the title Cloudera Management Service Monitored Entities.

The number of monitored entities for the Host Monitor and Service Monitor displays at the bottom of the chart. In the following example, the Host Monitor has 46 monitored entities and the Service Monitor has 230 monitored entities.



3. Use the number of monitored entities for the Host Monitor to determine its memory requirements and recommendations in the tables below.
4. Use the number of monitored entities for the Service Monitor to determine its memory requirements and recommendations in the tables below.

Clusters with HDFS, YARN, or Impala

Use the recommendations in this table for clusters where the only services having worker roles are HDFS, YARN, or Impala.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-2,000	0-100	1 GB	6 GB
2,000-4,000	100-200	1.5 GB	6 GB
4,000-8,000	200-400	1.5 GB	12 GB

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
8,000-16,000	400-800	2.5 GB	12 GB
16,000-20,000	800-1,000	3.5 GB	12 GB

Clusters with HBase, Solr, Kafka, or Kudu

Use the recommendations when services such as HBase, Solr, Kafka, or Kudu are deployed in the cluster. These services typically have larger quantities of monitored entities.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-30,000	0-100	2 GB	12 GB
30,000-60,000	100-200	3 GB	12 GB
60,000-120,000	200-400	3.5 GB	12 GB
120,000-240,000	400-800	8 GB	20 GB

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>
```

- 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:

- Create a local directory under your home directory.
- Copy the configuration files from `/etc/hadoop/conf` to the new directory.
- Set the permissions for the configuration files appropriately for your environment.
- Edit each configuration file:
 - Remove all elements within the `<configuration>` element.
 - 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>
```

- 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>
```

- 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-mapreduce-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 and 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
```

Accessing Storage Using Microsoft ADLS Gen 2

These topics focused on Microsoft ADLS from the core Cloudera Enterprise documentation library can help you deploy, configure, manage, and secure clusters in the cloud. They are listed by broad category:

Note the following limitations:

- ADLS is not supported as the default filesystem. Do not set the default file system property (fs.defaultFS) to an abfss:// URI. You can use ADLS as secondary filesystem while HDFS remains the primary filesystem.
- Hadoop Kerberos authentication is supported, but it is separate from the Azure user used for ADLS authentication.
- Directory and file names should not end with a period. Paths that end in periods can cause inconsistent behavior, including the period disappearing. For more information, see [HADOOP-15860](#).

Configuring OAuth in Data Hub

To connect a DataHub cluster to ADLS Gen2 with OAuth, you must configure the Hadoop CredentialProvider or core-site.xml directly. Although configuring core-site.xml is convenient, it is insecure since the contents of core-site.xml are not encrypted. For this reason, Cloudera recommends using a credential provider.

Before you start, ensure that you have configured OAuth for Azure.

Configuring OAuth with core-site.xml

Before you begin

Configuring your OAuth credentials in core-site.xml is insecure. Cloudera recommends that you only use this method for development environments or other environments where security is not a concern.

Perform the following steps to connect your cluster to ADLS Gen2:

Procedure

1. In the Cloudera Manager Admin Console, search for the following property: Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml .
2. Add the following properties and values:

Table 1: OAuth Properties

Name	Value
fs.azure.account.auth.type	OAuth

Name	Value
fs.azure.account.oauth.provider.type	org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider
fs.azure.account.oauth2.client.endpoint	Provide your tenant ID: https://login.microsoftonline.com/<Tenant_ID>/oauth2/token
fs.azure.account.oauth2.client.id	Provide your <Client_ID>
fs.azure.account.oauth2.client.secret	Provide your <Client_Secret>

What to do next

In addition, you can also provide account-specific keys. To do this, you need to add the following suffix to the key:

```
.<Account>.dfs.core.windows.net
```

Configuring OAuth with the Hadoop CredentialProvider

Before you begin

A more secure way to store your OAuth credentials is with the Hadoop CredentialProvider. When you submit a job, reference the CredentialProvider, which then supplies the OAuth information. Unlike the core-site.xml, the credentials are not stored in plain text.

The following steps describe how to create a credential provider and how to reference it when submitting jobs:

Procedure

1. Create a password for the Hadoop Credential Provider and export it to the environment:

```
export HADOOP_CREDSTORE_PASSWORD=password
```

2. Provision the credentials by running the following commands:

```
hadoop credential create dfs.adls.oauth2.client.id -provider jceks://hdfs/
user/USER_NAME/adls2keyfile.jceks -value client ID
hadoop credential create dfs.adls.oauth2.credential -provider jceks://h
dfs/user/USER_NAME/adls2keyfile.jceks -value client secret
hadoop credential create dfs.adls.oauth2.refresh.url -provider jceks://
hdfs/user/USER_NAME/adls2keyfile.jceks -value refresh URL
```

You can omit the -value option and its value and the command will prompt the user to enter the value.

For more details on the hadoop credential command, see [Credential Management \(Apache Software Foundation\)](#).

3. Export the password to the environment:

```
export HADOOP_CREDSTORE_PASSWORD=password
```

4. Reference the credential provider on the command line when you submit a job:

```
hadoop <command>
  -Ddfs.adls.oauth2.access.token.provider.type=ClientCredential \
  -Dhadoop.security.credential.provider.path=jceks://hdfs/user/USER_NAM
E/adls-cred.jceks \
  abfs[s]://<file_system>@<account_name>.dfs.core.windows.net/<path>/
<file_name>
```

Configuring Native TLS Acceleration

For ADLS Gen2, TLS is enabled by default using the Java implementation of TLS. For better performance, you can use the native OpenSSL implementation of TLS.

Perform the following steps to use the native OpenSSL implementation of TLS:

1. Verify the location of the OpenSSL libraries on the hosts with the following command:

```
whereis libssl
```

2. In the Cloudera Manager Admin Console, search for the following property: Gateway Client Environment Advanced Configuration Snippet (Safety Valve) for `hadoop-env.sh`.
3. Add the following parameter to the property:

```
HADOOP_OPTS="-Dorg.wildfly.openssl.path=<path to OpenSSL libraries> ${HADOOP_OPTS}"
```

For example, if the OpenSSL libraries are in `/usr/lib64`, add the following parameter:

```
HADOOP_OPTS="-Dorg.wildfly.openssl.path=/usr/lib64 ${HADOOP_OPTS}"
```

4. Save the change.
5. Search for the following property: HDFS Client Environment Advanced Configuration Snippet (Safety Valve) for `hadoop-env.sh`
6. Add the following parameter to the property:

```
HADOOP_OPTS="-Dorg.wildfly.openssl.path=<path to OpenSSL libraries> ${HADOOP_OPTS}"
```

For example, if the OpenSSL libraries are in `/usr/lib64`, add the following parameter:

```
HADOOP_OPTS="-Dorg.wildfly.openssl.path=/usr/lib64 ${HADOOP_OPTS}"
```

7. Save the change.
8. [Restart the stale services.](#)
9. [Deploy the client configurations.](#)
10. Verify that you configured native TLS acceleration successfully by running the following command from any host in the cluster:

```
hadoop fs -ls abfss://<container>@<account>.dfs.core.windows.net/
```

A message similar to the following should appear:

```
org.wildfly.openssl.SSL init
INFO: WFOpenssl0002 OpenSSL Version OpenSSL 1.0.1e-fips 11 Feb 2013
```

The message may differ slightly depending on your operating system and OpenSSL version.

Importing Data into Microsoft Azure Data Lake Store (Gen1 and Gen2) Using Sqoop

Microsoft Azure Data Lake Store (ADLS) is a cloud object store designed for use as a hyper-scale repository for big data analytic workloads. ADLS acts as a persistent storage layer for CDH clusters running on Azure.

There are two generations of ADLS, Gen1 and Gen2. You can use Apache Sqoop with both generations of ADLS to efficiently transfer bulk data between these file systems and structured datastores such as relational databases. For more information on ADLS Gen 1 and Gen 2, see:

- [Microsoft ADLS Gen1 documentation](#)
- [Microsoft ADLS Gen2 documentation](#)

You can use Sqoop to import data from any relational database that has a JDBC adaptor such as SQL Server, MySQL, and others, to the ADLS file system.



Note: Sqoop export from the Azure files systems is not supported.

The major benefits of using Sqoop to move data are:

- It leverages RDBMS metadata to get the column data types
- It ensures fault-tolerant and type-safe data handling
- It enables parallel and efficient data movement

Prerequisites

The configuration procedure presumes that you have already set up an Azure account, and have configured an ADLS Gen1 store or ADLS Gen2 storage account and container. See the following resources for information:

- [Microsoft ADLS Gen1 documentation](#)
- [Microsoft ADLS Gen2 documentation](#)
- [Hadoop Azure Data Lake Support](#)

Authentication

To connect a cluster to ADLS with OAuth, you must configure the Hadoop CredentialProvider or core-site.xml directly. Although configuring the core-site.xml is convenient, it is insecure, because the contents of core-site.xml configuration file are not encrypted. For this reason, Cloudera recommends using a credential provider. For more information, see [Configuring OAuth in CDH](#).

You can also pass the credentials by providing them on the Sqoop command line as part of the import command.

```
sqoop import
-Dfs.azure.account.auth.type=...
-Dfs.azure.account.oauth.provider.type=...
-Dfs.azure.account.oauth2.client.endpoint=...
-Dfs.azure.account.oauth2.client.id=...
-Dfs.azure.account.oauth2.client.secret=...
```

For example:

```
sqoop import
-Dfs.azure.account.oauth2.client.endpoint=https://login.microsoftonline.com/$TENANT_ID/oauth2/token
-Dfs.azure.account.oauth2.client.id=$CLIENT_ID
-Dfs.azure.account.oauth2.client.secret=$CLIENT_SECRET
-Dfs.azure.account.auth.type=OAuth
-Dfs.azure.account.oauth.provider.type=org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider
```

Sqoop Import into ADLS

To import data into ADLS from diverse data sources, such as a relational database, enter the Sqoop import command on the command line of your cluster. Make sure that you specify the Sqoop connection to the data source you want to import.

If you want to enter a password for the data source, use the `-P` option in the connection string. If you want to specify a file where the password is stored, use the `--password-file` option.

```
sqoop import
-Dfs.azure.account.auth.type=...
-Dfs.azure.account.oauth.provider.type=...
-Dfs.azure.account.oauth2.client.endpoint=...
-Dfs.azure.account.oauth2.client.id=...
-Dfs.azure.account.oauth2.client.secret=...
--connect... --username... --password... --table... --target-dir... --split-
by...
```

ABFS example:

```
sqoop import
-Dfs.azure.account.oauth2.client.endpoint=https://login.microsoftonline
.com/$TENANT_ID/oauth2/token
-Dfs.azure.account.oauth2.client.id=$CLIENT_ID
-Dfs.azure.account.oauth2.client.secret=$CLIENT_SECRET
-Dfs.azure.account.auth.type=OAuth
-Dfs.azure.account.oauth.provider.type=org.apache.hadoop.fs.azurebfs.oauth2.
ClientCredsTokenProvider
--connect $CONN --username $USER --password $PWD --table $TABLENAME --targe
t-dir abfs://$CONTAINER$ACCOUNT.dfs.core.windows.net/$TARGET-DIRECTORY --s
plit-by $COLUMN_NAME
```

ADLS example:

```
sqoop import
-Dfs.adl.oauth2.refresh.url=https://login.windows.net/$TENANT_ID/oauth2/toke
n
-Dfs.adl.oauth2.client.id=$CLIENT_ID
-Dfs.adl.oauth2.credential=$CLIENT_SECRET
-Dfs.adl.oauth2.access.token.provider.type=ClientCredential
--connect $CONN --username $USER --password $PWD --table $TABLENAME --targe
t-dir adl://$TARGET-ADDRESS/$TARGET-DIRECTORY --split-by $COLUMN_NAME
```