Cloudera Manager 7.0.2

Managing Data Hub Clusters

Date published: 2020-02-11
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Accessing the Cloudera Manager Admin Console from Data Hub clusters

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

About this task
To access the Cloudera Manager Admin Console:

Procedure

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

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

Accessing the Cloudera Manager for Data Lake clusters using workload credentials

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

About this task
To log in to the Cloudera Manager using workload credentials:

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

Procedure

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

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

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

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

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

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

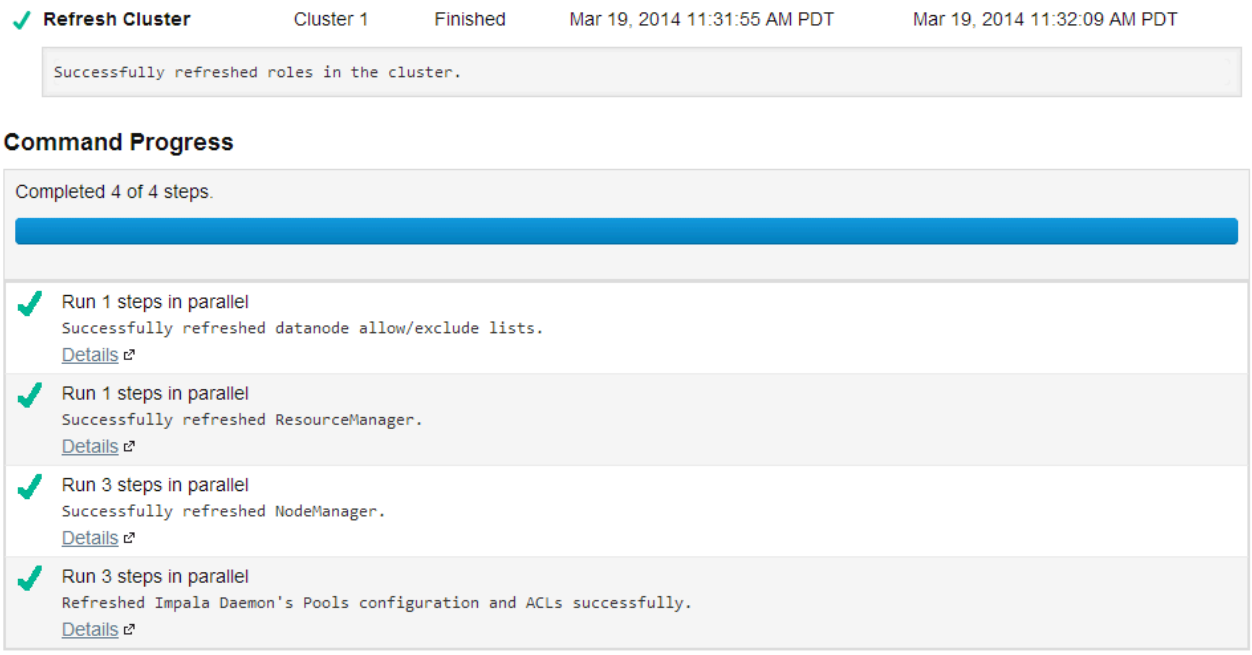
1. On the HomeStatus tab, click  to the right of the cluster name and select Stop.
2. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services.

   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:



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

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.

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.

Renaming a Cluster

About this task

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Managing Hosts

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

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

Status

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

Viewing All Hosts

To display summary information about all the hosts managed by Cloudera Manager, click 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.
- 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* = Good
  - *v* >= *x* * *y* and <= *x* * *z* = Concerning
  - *v* >= *x* * *z* = Bad

Viewing the Hosts in a Cluster

Do one of the following:

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

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

Viewing Individual Hosts

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

Configuration

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

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

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

Related Information

Modifying Configuration Properties Using Cloudera Manager

Roles

Role Assignments

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

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

Disks Overview

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

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

Host Templates

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

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

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

Stopping All the Roles on a Host

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

About this task

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

Procedure

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

Starting All the Roles on a Host

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

About this task

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

Procedure

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

Specifying Racks for Hosts

To get maximum performance, it is important to configure CDH so that it knows the topology of your network. Network locations such as hosts and racks are represented in a tree, which reflects the network “distance” between locations. HDFS will use the network location to be able to place block replicas more intelligently to trade off performance and resilience.

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.

When placing jobs on hosts, CDH prefers within-rack transfers (where there is more bandwidth available) to off-rack transfers; the MapReduce and YARN schedulers use network location to determine where the closest replica is as input to a map task. These computations are performed with the assistance of rack awareness scripts.

Cloudera Manager includes internal rack awareness scripts, but you must specify the racks where the hosts in your cluster are located. If your cluster contains more than 10 hosts, Cloudera recommends that you specify the rack for each host. HDFS, MapReduce, and YARN will automatically use the racks you specify.

Cloudera Manager supports nested rack specifications. For example, you could specify the rack /rack3, or /gro up5/rack3 to indicate the third rack in the fifth group. All hosts in a cluster must have the same number of path components in their rack specifications.

Procedure

1. Click the Hosts tab.
2. Check the checkboxes next to the host(s) for a particular rack, such as all hosts for /rack123.
3. Click Actions for Selected (*n*)Assign Rack, where *n* is the number of selected hosts.
4. Enter a rack name or ID that starts with a slash /, such as /rack123 or /aisle1/rack123, and then click Confirm.
5. Optionally restart affected services. Rack assignments are not automatically updated for running services.

Host Templates

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Host templates let you designate a set of role groups that can be applied in a single operation to a host or a set of hosts. This significantly simplifies the process of configuring new hosts when you need to expand your cluster. Host templates are supported for both CDH 4 and CDH 5 cluster hosts.

Important: A host template can only be applied on a host with a version of CDH that matches the CDH version running on the cluster to which the host template belongs.

You can create and manage host templates by clicking HostsHost Templates.

Templates are not required; Cloudera Manager assigns roles and role groups to the hosts of your cluster when you perform the initial cluster installation. However, if you want to add new hosts to your cluster, a host template can make this much easier.

If there are existing host templates, they are listed on the page, along with links to each role group included in the template.

If you are managing multiple clusters, you must create separate host templates for each cluster, as the templates specify role configurations specific to the roles in a single cluster. Existing host templates are listed under the cluster to which they apply.

- You can click a role group name to be taken to the Edit configuration page for that role group, where you can modify the role group settings.
- From the Actions menu associated with the template you can edit the template, clone it, or delete it.

Creating a Host Template

When you create a host template, you choose a name for the template and select appropriate role groups for each role.

Procedure

1. Click HostsHost Templates.
2. From the **Host Templates** page, click Create.
   The **Create New Host Template** pop-up window appears.
3. Type a name for the template.
4. For each role, select the appropriate role group. There may be multiple role groups for a given role type — you want to select the one with the configuration that meets your needs.
5. Click Create to create the host template.

Editing a Host Template

You can edit the name of a host template, in addition to any of the role group selections.

Procedure

1. Click HostsHost Templates.
2. Pull down the Actions menu for the template you want to modify, and click Edit.
   The **Edit Host Template** window appears. This page is identical to the Create New Host Template page. You can modify the template name or any of the role group selections.
3. Click OK when you have finished.

Applying a Host Template to a Host

You can use a host template to apply configurations for multiple roles in a single operation.

About this task

You can apply a template to a host that has no roles on it, or that has roles from the same services as those included in the host template. New roles specified in the template that do not already exist on the host will be added. A role on the host that is already a member of the role group specified in the template will be left unchanged. If a role on the host matches a role in the template, but is a member of a different role group, it will be moved to the role group specified by the template.

For example, suppose you have two role groups for a DataNode (DataNode Default Group and DataNode (1)). The host has a DataNode role that belongs to DataNode Default Group. If you apply a host template that specifies the DataNode (1) group, the role on the host will be moved from DataNode Default Group to DataNode (1).

However, if you have two instances of a service, such as MapReduce (for example, mr1 and mr2) and the host has a TaskTracker role from service mr2, you cannot apply a TaskTracker role from service mr1.

A host may have no roles on it if you have just added the host to your cluster, or if you decommissioned a managed host and removed its existing roles.

Also, the host must have the same version of CDH installed as is running on the cluster whose host templates you are applying.

If a host belongs to a different cluster than the one for which you created the host template, you can apply the host template if the "foreign" host either has no roles on it, or has only management roles on it. When you apply the host template, the host will then become a member of the cluster whose host template you applied. The following instructions assume you have already created the appropriate host template.

Procedure

1. Click HostsAll Hosts.
2. Select the host(s) to which you want to apply your host template.
3. From the Actions for Selected menu, select Apply Host Template.
4. In the pop-up window that appears, select the host template you want to apply.
5. Optionally you can have Cloudera Manager start the roles created per the host template. To enable this, check the box.
6. Click Confirm to initiate the action.

Performing Maintenance on a Cluster Host

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

In this process, you can specify whether to suppress alerts from the decommissioned host and, for hosts running the DataNode role, you can specify whether or not to replicate under-replicated data blocks to other DataNodes to maintain the cluster's replication factor. This feature is useful when performing minor maintenance on cluster hosts,

such as adding memory or changing network cards or cables where the maintenance window is expected to be short and the extra cluster resources consumed by replicating missing blocks is undesirable.

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

To perform host maintenance on cluster hosts:

Decommissioning Hosts

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

About this task

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

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

Before you begin

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

Procedure

To decommission one or more hosts:

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

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

   The Begin Maintenance (Suppress Alerts/Decommission) dialog box opens. The role instances running on the hosts display at the top.
5. To decommission the hosts and suppress alerts, select Decommission Host(s). When you select this option for hosts running a DataNode role, choose one of the following (if the host is not running a DataNode role, you will only see the Decommission Host(s) option:):

   - Decommission DataNodes

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

     This option does not re-replicate HDFS data to other DataNodes until the amount of time you specify has passed, making it less disruptive to active workloads. After this time has passed, the DataNode is automatically

recommissioned, but the DataNode role is not started. This option is appropriate for short-term maintenance tasks such not involving disks, such as rebooting, CPU/RAM upgrades, or switching network cables.

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

6. Click Begin Maintenance.
   The Host Decommission Command dialog box opens and displays the progress of the command.

Results

Note:

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

What to do next

Perform the necessary maintenance on the hosts.

Recommissioning Hosts

About this task

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

Before you begin

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

Procedure

1. In Cloudera Manager, select the cluster where you want to recommission hosts.
2. In the left menu, click HostsAll Hosts.
3. Select the hosts that you want to recommission.
4. Select Actions for SelectedEnd Maintenance (Suppress Alerts/Decommission.
   The End Maintenance (Suppress Alerts/Decommission dialog box opens. The role instances running on the hosts display at the top.
5. To recommission the hosts, select Recommission Host(s).
6. Choose one of the following:
   - Bring hosts online and start all roles

     All decommissioned roles will be recommissioned and started. HDFS DataNodes will be started first and brought online before decommissioning to avoid excess replication.
   - Bring hosts online

     All decommissioned roles will be recommissioned but remain stopped. You can restart the roles later.
7. Click End Maintenance.

Results

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

Tuning and Troubleshooting Host Decommissioning

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

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

Tuning HDFS Prior to Decommissioning DataNodes

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

About this task

Minimum Required Role: Configurator (also provided by Cluster Administrator, Limited Cluster Administrator , 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 -locati
ons 2>&1 > /tmp/hdfs-fsck.txt
```

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

```
hdfs fsck file_name -move
```

or

```
hdfs fsck file_name -delete
```

3. Raise the heap size of the DataNodes. DataNodes should be configured with at least 4 GB heap size to allow for the increase in iterations and max streams.
   a) Go to the HDFS service page.
   b) Click the Configuration tab.
   c) Select ScopeDataNode.
   d) Select CategoryResource Management.
   e) Set the Java Heap Size of DataNode in Bytes property as recommended.

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

4. Increase the replication work multiplier per iteration to a larger number (the default is 2, however 10 is recommended).
   a) Select ScopeNameNode.
   b) Expand the CategoryAdvanced category.
   c) Configure the Replication Work Multiplier Per Iteration property to a value such as 10.

      To apply this configuration property to other role groups as needed, edit the value for the appropriate role group.
   d)
5. Increase the replication maximum threads and maximum replication thread hard limits.
   a) Select ScopeNameNode.
   b) Expand the CategoryAdvanced category.
   c) Configure the Maximum number of replication threads on a DataNode and Hard limit on the number of replication threads on a DataNode properties to 50 and 100 respectively. You can decrease the number of threads (or use the default values) to minimize the impact of decommissioning on the cluster, but the trade off is that decommissioning will take longer.

      To apply this configuration property to other role groups as needed, edit the value for the appropriate role group.
   d)
6. Restart the HDFS service.

Related Information

Performance Considerations

Modifying Configuration Properties Using Cloudera Manager

Tuning HBase Prior to Decommissioning DataNodes

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

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

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

Performance Considerations

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

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

Decommissioning occurs in two steps:

1. The Commission State of the DataNode is marked as Decommissioning and the data is replicated from this node to other available nodes. Until all blocks are replicated, the node remains in a Decommissioning state. You can view this state from the NameNode Web UI. (Go to the HDFS service and select Web 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.BlockPlacementPoli
cyDefault: Not able to place enough replicas, still in need of 3
 to reach 3
```

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

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

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

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

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

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

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

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

Maintenance Mode

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

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

Explicit and Effective Maintenance Mode

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

For example:

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

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

When an entity (role, host or service) is in effective maintenance mode, it can only be removed from maintenance mode when the higher-level entity exits maintenance mode. For example, if you put a service into maintenance mode, the roles associated with that service are entered into effective maintenance mode, and remain in effective

maintenance mode until the service exits maintenance mode. You cannot remove them from maintenance mode individually.

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

Entering Maintenance Mode

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

Putting a Cluster into Maintenance Mode

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

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

Putting a Service into Maintenance Mode

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

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

Putting Roles into Maintenance Mode

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

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

Putting Hosts into Maintenance Mode

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

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

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

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

Exiting Maintenance Mode

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

Exiting a Cluster from Maintenance Mode

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

Exiting a Service from Maintenance Mode

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

Exiting Roles from Maintenance Mode

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

Taking Hosts out of Maintenance Mode

1. In Cloudera Manager, to go the cluster with the hosts you want to take out of Maintenance Mode.
2. Click HostsAll Hosts.
3. Select the hosts that are ready to exit Maintenance Mode.
4. Select Actions for SelectedEnd Maintenance (Suppress Alerts/Decommission.

   The End Maintenance (Suppress Alerts/Decommission) dialog box opens. The role instances running on the hosts display at the top.
5. Deselect the Recommission Host(s) option to take the host out of Maintenance Mode and re-enable alerts from the hosts. Hosts that are currently in Maintenance Mode display the  icon on the All Hosts page.
6. Click End Maintenance.

Viewing the Maintenance Mode Status of a Cluster

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

Procedure

1. From the Cloudera Manager Home page, select the cluster that you want to view the maintenance mode status for.

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

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

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

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

Changing Hostnames

After you have installed Cloudera Manager and created a cluster, you may need to update the names of the hosts running the Cloudera Manager Server or cluster services.

About this task

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Important:

- The process described here requires Cloudera Manager and cluster downtime.
- If any user-created scripts reference specific hostnames, those must also be updated.
- Due to the length and complexity of the following procedure, changing cluster hostnames is not recommended by Cloudera.

To update a deployment with new hostnames, follow these steps:

Procedure

1. Verify if TLS/SSL certificates have been issued for any of the services and make sure to create new TLS/SSL certificates in advance for services protected by TLS/SSL.
2. Export the Cloudera Manager configuration using one of the following methods:
   - Open a browser and go to this URL http://*cm_hostname*:7180/api/*api_version*/cm/deployment. Save the displayed configuration.
   - From terminal type:

     $ curl -u admin:admin http://*cm_hostname*:7180/api/*api_version*/cm/deployment > cme-cm-export.json

     If Cloudera Manager SSL is in use, specify the -k switch:

     $ curl -k -u admin:admin http://*cm_hostname*:7180/api/*api_version*/cm/deployment > cme-cm-export.json

     where *cm_hostname* is the name of the Cloudera Manager host and *api_version* is the correct version of the API for the version of Cloudera Manager you are using. For example, http://tcdn5-1.ent.cloudera.com:7180/api/v40/cm/deployment.
3. Stop all services on the cluster.
4. Stop the Cloudera Management Service.
5. Stop the Cloudera Manager Server.
6. Stop the Cloudera Manager Agents on the hosts that you want to change the hostname of.
7. Back up the Cloudera Manager Server database using mysqldump, pg_dump, or another preferred backup utility. Store the backup in a safe location.

8. Update names and principals:
   a) Update the target hosts using standard per-OS/name service methods (/etc/hosts, dns, /etc/sysconfig/network, hostname, and so on). Ensure that you remove the old hostname.
   b) If you are changing the hostname of the host running Cloudera Manager Server do the following:
      1. Change the hostname per Step 8.a.
      2. Update the Cloudera Manager hostname in /etc/cloudera-scm-agent/config.ini on all Agents.
   c) If the cluster is configured for Kerberos security, do the following:
      1. Remove the old hostname cluster principals.
         - If you are using an MIT KDC, remove old hostname cluster service principals from the KDC database using one of the following:
           - Use the delprinc command within kadmin.local interactive shell.

             OR
           - From the command line:

             ```
             kadmin.local -q "listprincs" | grep -E "(HTTP|hbase|hdfs|hive|ht
             tpfs|hue|impala|mapred|solr|oozie|yarn|zookeeper)[^/]*/[^/]*@" >
              cluster-princ.txt
             ```

             Open cluster-princ.txt and remove any noncluster service principal entries. Make sure that the default krbtgt and other principals you created, or that were created by Kerberos by default, are not removed by running the following: for i in `cat cluster-princ.txt`; do yes yes | kadmin.local -q "delprinc $i"; done.
         - For an Active Directory KDC, an AD administrator must manually delete the principals for the old hostname from Active Directory.
      2. Start the Cloudera Manager database and Cloudera Manager Server.
      3. Start the Cloudera Manager Agents on the newly renamed hosts. The Agents should show a current heartbeat in Cloudera Manager.
      4. Within the Cloudera Manager Admin Console click the Hosts tab.
      5. Select the checkbox next to the host with the new name.
      6. Select ActionsRegenerate Keytab.
9. If one of the hosts that was renamed has a NameNode configured with high availability and automatic failover enabled, reconfigure the ZooKeeper Failover Controller znodes to reflect the new hostname.
   a) Start ZooKeeper Servers.

      Warning: All other services, and most importantly HDFS, and the ZooKeeper Failover Controller (FC) role within the HDFS, should not be running.
   b) On one of the hosts that has a ZooKeeper Server role, run zookeeper-client.
      1. If the cluster is configured for Kerberos security, configure ZooKeeper authorization as follows:
         - Go to the HDFS service.
         - Click the Instances tab.
         - Click the Failover Controller role.
         - Click the Process tab.
         - In the Configuration Files column of the hdfs/hdfs.sh ["zkfc"] program, expand Show.
         - Inspect core-site.xml in the displayed list of files and determine the value of the ha.zookeeper.auth property, which will be something like: digest:hdfs-fcs:TEbW2bgoODa96rO3ZTn7ND5fSOGx0h. The

part after digest:hdfs-fcs: is the password (in the example it is TEbW2bgoODa96rO3ZTn7ND5fSOGx0h)

- Run the addauth command with the password:

```
addauth digest hdfs-fcs:TEbW2bgoODa96rO3ZTn7ND5fSOGx0h
```

2. Verify that the HA znode exists: ls /hadoop-ha.
3. Delete the HDFS znode: rmr /hadoop-ha/nameservice1.
4. If you are not running JobTracker in a high availability configuration, delete the HA znode: rmr /hadoop-ha.

c) In the Cloudera Manager Admin Console, go to the HDFS service.
d) Click the Instances tab.
e) Select ActionsInitialize High Availability State in ZooKeeper....

10. Update the Hive metastore:
    a) Back up the Hive metastore database.
    b) In the Cloudera Manager Admin Console, go to the Hive service.
    c) Select ActionsUpdate Hive Metastore NameNodes and confirm the command.
11. Update the Database Hostname property for each of the cluster roles for which a database is located on the host being renamed. This is required for both Cloudera Management Service roles (Reports Manager, Activity Monitor, Navigator Audit and Metadata Server) and for cluster services such as Hue, Hive, and so on.
12. Start all cluster services.
13. Start the Cloudera Management Service.
14. Deploy client configurations.

Deleting Hosts

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

You can remove a host from a cluster in two ways:

- Delete the host entirely from Cloudera Manager.
- Remove a host from a cluster, but leave it available to other clusters managed by Cloudera Manager.

Both methods decommission the hosts, delete roles, and remove managed service software, but preserve data directories.

Deleting a Host from Cloudera Manager

To delete a host from Cloudera Manager, first decommission the host and then remove it.

Procedure

1. In the Cloudera Manager Admin Console, go to Hosts All Hosts.
2. Select the hosts to delete.
3. Select Actions for SelectedHosts Decommission.
4. Stop the Agent on the host.
5. In the Cloudera Manager Admin Console, go to Hosts All Hosts.
6. Reselect the hosts you selected in Step 2.
7. Select Actions for SelectedRemove from Cloudera Manager.

Removing a Host From a Cluster

Removing a host from a cluster leaves the host managed by Cloudera Manager and preserves the Cloudera Management Service roles (such as the Events Server, Activity Monitor, and so on).

Procedure

1. In the Cloudera Manager Admin Console, click the Hosts tab.
2. Select the hosts to delete.
3. Select Actions for SelectedRemove From Cluster. The **Remove Hosts From Cluster** dialog box displays.
4. Leave the selections to decommission roles and skip removing the Cloudera Management Service roles. Click Confirm to proceed with removing the selected hosts.

Moving a Host Between Clusters

To move a host between clusters, you must first decommission the host, remove roles from the host, and complete other tasks.

About this task

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Procedure

1. Decommission the host.
2. Remove all roles from the host (except for the Cloudera Manager management roles).
3. Remove the host from the cluster but leave it available to Cloudera Manager.
4. Add the host to the new cluster.
5. Add roles to the host (optionally using one of the host templates associated with the new cluster).

Managing Roles

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

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

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

The HDFS service is an example of this: Cloudera Manager typically creates one role group (DataNode Default Group) for the DataNode role instances running on the worker hosts, and another group (HDFS-1-DATANODE-1)

for the DataNode instance running on the host that is also running the master roles such as the NameNode, JobTracker, HBase Master and so on. Typically the configurations for those two classes of hosts will differ in terms of settings such as memory for JVMs.

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

Gateway Roles

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

Related Information

Cluster Configuration Overview

Role Instances

Starting, Stopping, and Restarting Role Instances

About this task

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

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

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

Procedure

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

Related Information

Rolling Restart

Decommissioning Role Instances

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

About this task

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

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

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

A role will be decommissioned if its host is decommissioned.

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

Procedure

To decommission role instances:

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

Results

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

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

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

Related Information

Tuning HDFS Prior to Decommissioning DataNodes

Recommissioning Role Instances

Procedure

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

Configuring Roles to Use a Custom Garbage Collection Parameter

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

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

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

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

Related Information

Modifying Configuration Properties Using Cloudera Manager

Role Groups

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

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

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

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

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

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

Related Information

Configuring Monitoring Settings
Overriding Configuration Properties

Creating a Role Group

Procedure

1. Go to a service status page.
2. Click the Instances or Configuration tab.
3. Click Role Groups.
4. Click Create new group....
5. Provide a name for the group.

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

Related Information

Modifying Configuration Properties Using Cloudera Manager

Managing Role Groups

Procedure

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

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

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

Related Information

Managing Hosts

Managing Cloudera Runtime Services

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

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

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

Starting a Cloudera Runtime Service on All Hosts

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

Before you begin

The order in which to stop services is:

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

Procedure

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

Restarting a Cloudera Runtime Service

Procedure

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

Results

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

What to do next

To restart all services, restart the cluster.

Rolling Restart

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

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

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

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

Performing a Service or Role Rolling Restart

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

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

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

Note:

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

5. Click Confirm to start the rolling restart.

Aborting a Pending Command

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

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

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

To abort any pending command:

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

To abort a pending command for a service or role:

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

Related Information

Viewing Running and Recent Commands

Managing Cloudera Manager

Automatic Logout

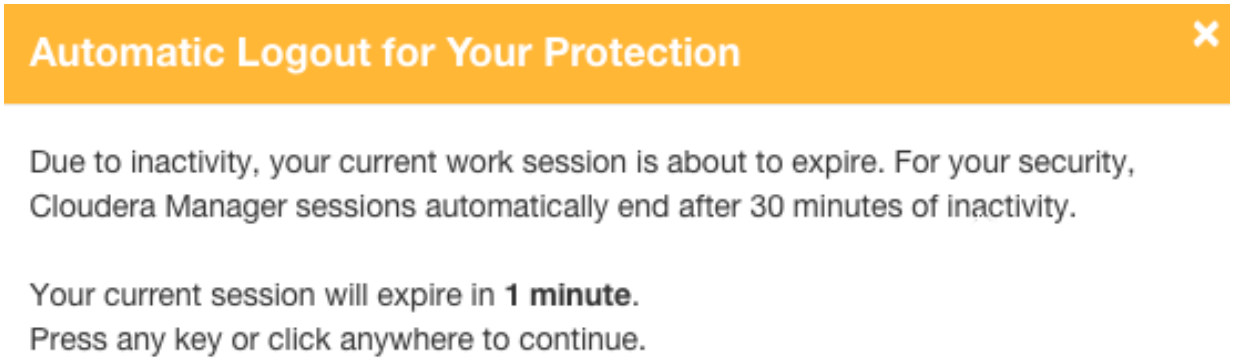
For security purposes, Cloudera Manager automatically logs out a user session after 30 minutes. You can change this session logout period.

Procedure

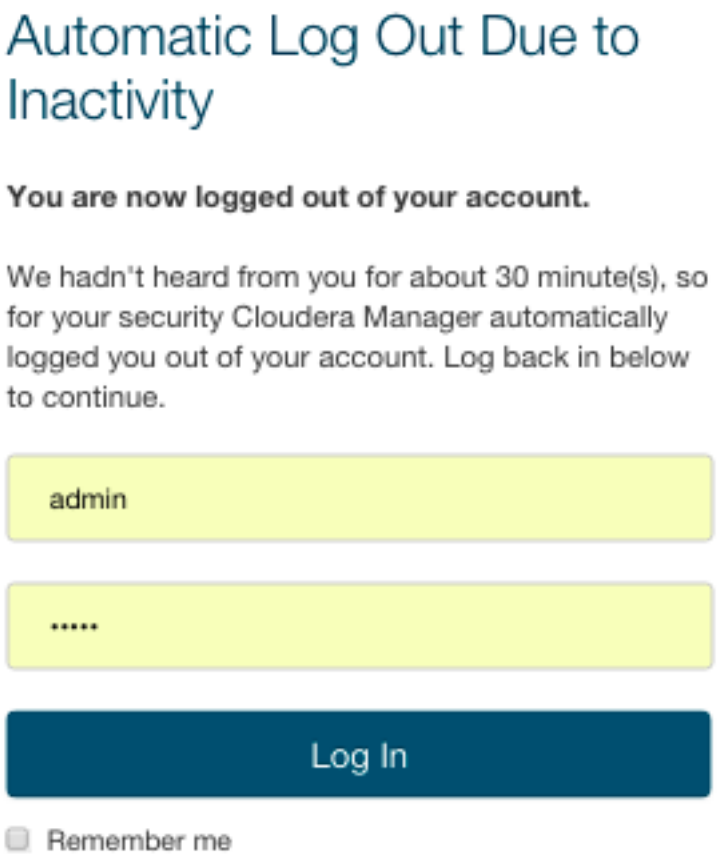
1. Click AdministrationSettings.
2. Click CategorySecurity.
3. Edit the Session Timeout property.

4. Enter a Reason for change, and then click Save Changes to commit the changes.

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



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



Starting, Stopping, and Restarting the Cloudera Manager Server

To start the Cloudera Manager Server:

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

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

To stop the Cloudera Manager Server:

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

To restart the Cloudera Manager Server:

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

Configuring Cloudera Manager Server Ports

You can specify the ports used to access the Cloudera Manager Server using the Admin Console. You can also specify the port used by agents to connect to the Server.

About this task

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Procedure

1. Select AdministrationSettings.
2. Under the Ports and Addresses category, set the following options as described below:

Setting	Description
HTTP Port for Admin Console	Specify the HTTP port to use to access the Server using the Admin Console.
HTTPS Port for Admin Console	Specify the HTTPS port to use to access the Server using the Admin Console.
Agent Port to connect to Server	Specify the port for Agents to use to connect to the Server.

3. Click Save Changes.
4. Restart the Cloudera Manager Server.

Moving the Cloudera Manager Server to a New Host

You can move the Cloudera Manager Server if either the Cloudera Manager database server or a current backup of the Cloudera Manager database is available.

Procedure

1. Identify a new host on which to install Cloudera Manager.

2. Install Cloudera Manager on a new host, using the method described in the topic *Install the Cloudera Manager Server Packages*.

   Important:

   - The Cloudera Manager version on the destination host must match the version on the source host.
   - Do not install the other components, such as CDH and databases.

3. Copy the entire content of /var/lib/cloudera-scm-server/ on the old host to that same path on the new host. Ensure you preserve permissions and all file content.
4. If the database server is not available:
   a) Install the database packages on the host that will host the restored database. This could be the same host on which you have just installed Cloudera Manager or it could be a different host. If you used the embedded PostgreSQL database, install the PostgreSQL package as described in the topic *Managing the Embedded PostgreSQL Database*. If you used an external MySQL, PostgreSQL, or Oracle database, reinstall the database following the instructions in *Step 4: Install and Configure Databases*.
   b) Restore the backed up databases to the new database installation.
5. Update /etc/cloudera-scm-server/db.properties with the database name, database instance name, username, and password.
6. Do the following on all cluster hosts:
   a) In /etc/cloudera-scm-agent/config.ini, update the server_host property to the new hostname.
   b) If you are replacing the Cloudera Manager database with a new database, and you are not using a backup of the original Cloudera Manager database, delete the /var/lib/cloudera-scm-agent/cm_guid file.
   c) Restart the agent using the following command:

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

7. Stop the Cloudera Manager server on the source host by running the following command:

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

8. Start the Cloudera Manager Server on the new (destination) host. Cloudera Manager should resume functioning as it did before the failure. Because you restored the database from the backup, the server should accept the running state of the Agents, meaning it will not terminate any running processes.

   The process is similar with secure clusters, though files in /etc/cloudera-scm-server must be restored in addition to the database. See the *Security* documentation.

Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database

Cloudera Manager provides an embedded PostgreSQL database server for demonstration and proof of concept deployments when creating a cluster. To remind users that this embedded database is not suitable for production, Cloudera Manager displays the banner text: "You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production."

If, however, you have already used the embedded database, and you are unable to redeploy a fresh cluster, then you must migrate to an external PostgreSQL database.

Note: This procedure does not describe how to migrate to a database server other than PostgreSQL. Moving databases from one database server to a different type of database server is a complex process that requires modification of the schema and matching the data in the database tables to the new schema. It is strongly recommended that you engage with Cloudera Professional Services if you wish to perform a migration to an external database server other than PostgreSQL.

Prerequisites

Before migrating the Cloudera Manager embedded PostgreSQL database to an external PostgreSQL database, ensure that your setup meets the following conditions:

- The external PostgreSQL database server is running.
- The database server is configured to accept remote connections.
- The database server is configured to accept user logins using md5.
- No one has manually created any databases in the external database server for roles that will be migrated.

Note: To view a list of databases in the external database server (requires default superuser permission):

```
sudo -u postgres psql -l
```

- All health issues with your cluster have been resolved.

For details about configuring the database server, see the topic *Configuring and Starting the PostgreSQL Server*.

Important: Only perform the steps in *Configuring and Starting the PostgreSQL Server*. Do not proceed with the creation of databases as described in the subsequent section.

For large clusters, Cloudera recommends running your database server on a dedicated host. Engage Cloudera Professional Services or a certified database administrator to correctly tune your external database server.

Step 1: Identify Roles that Use the Embedded Database Server

Before you can migrate to another database server, you must first identify the databases using the embedded database server. When the Cloudera Manager Embedded Database server is initialized, it creates the Cloudera Manager database and databases for roles in the Management Services. The Installation Wizard (which runs automatically the first time you log in to Cloudera Manager) or Add Service action for a cluster creates additional databases for roles when run. It is in this context that you identify which roles are used in the embedded database server.

Procedure

1. Obtain and save the cloudera-scm superuser password from the embedded database server. You will need this password in subsequent steps:

   ```
   head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
   ```

2. Make a list of all services that are using the embedded database server. Then, after determining which services are not using the embedded database server, remove those services from the list. The scm database must remain in your list. Use the following table as a guide:

Table 1: Cloudera Manager Embedded Database Server Databases

Service	Role	Default Database Name	Default Username
Cloudera Manager Server		scm	scm
Cloudera Management Service	Activity Monitor	amon	amon
Hive	Hive Metastore Server	hive	hive
Hue	Hue Server	hue	7uu7uu7uhue
Cloudera Management Service	Navigator Audit Server	nav	nav
Cloudera Management Service	Navigator Metadata Server	navms	navms
Oozie	Oozie Server	oozie_oozie_server	oozie_oozie_server
Cloudera Management Service	Reports Manager	rman	rman
Sentry	Sentry Server	sentry	sentry

3. Verify which roles are using the embedded database. Roles using the embedded database server always use port 7432 (the default port for the embedded database) on the Cloudera Manager Server host.

   For Cloudera Management Services:

   a. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
   b. Confirm that the hostname for the services being used is the same hostname used by the Cloudera Manager Server.

   Note:

   If any of the following fields contain the value "7432", then the service is using the embedded database:

   - Activity Monitor
   - Navigator Audit Server
   - Navigator Metadata Server
   - Reports Manager

   For the Oozie Service:

   a. Select Oozie service > Configuration, and type "7432" in the Search field.
   b. Confirm that the hostname is the Cloudera Manager Server.

   For Hive, Hue, and Sentry Services:

   a. Select the specific service > Configuration, and type "database host" in the Search field.
   b. Confirm that the hostname is the Cloudera Manager Server.
   c. In the Search field, type “database port” and confirm that the port is 7432.
   d. Repeat these steps for each of the services (Hive, Hue and Sentry).

4. Verify the database names in the embedded database server match the database names on your list (Step 2). Databases that exist on the database server and not used by their roles do not need to be migrated. This step is to confirm that your list is correct.

   Note: Do not add the postgres, template0, or template1 databases to your list. These are used only by the PostgreSQL server.

```
psql -h localhost -p 7432 -U cloudera-scm -l
```

```
Password for user cloudera-scm: <password>

                                                List of databases
       Name        |       Owner        | Encoding |  Collate   |   Ctype
    |         Access
--------------------+--------------------+----------+------------+-------
-----+---------------
amon                | amon               | UTF8     | en_US.UTF8 | en_US.U
TF8 |
hive                | hive               | UTF8     | en_US.UTF8 | en_US.UT
F8 |
hue                 | hue                | UTF8     | en_US.UTF8 | en_US
.UTF8 |
nav                 | nav                | UTF8     | en_US.UTF8 | en_US.
UTF8 |
navms               | navms              | UTF8     | en_US.UTF8 | en_US.U
TF8 |
oozie_oozie_server | oozie_oozie_server | UTF8     | en_US.UTF8 | en_US.UT
F8 |
postgres            | cloudera-scm       | UTF8     | en_US.UTF8 | en_US
.UTF8 |
rman                | rman               | UTF8     | en_US.UTF8 | en_US.
UTF8 |
```

```
scm                 | scm                 | UTF8     | en_US.UTF8 | en_US.U
TF8 |
sentry              | sentry              | UTF8     | en_US.UTF8 | en_US.UT
F8 |
template0           | cloudera-scm        | UTF8     | en_US.UTF8 | en_US
.UTF8 | =c/"cloudera-scm"
template1           | cloudera-scm        | UTF8     | en_US.UTF8 | en_US.UT
F8 | =c/"cloudera-scm"
(12 rows)
```

Results

You should now have a list of all roles and database names that use the embedded database server, and are ready to proceed with the migration of databases from the embedded database server to the external PostgreSQL database server.

What to do next

Proceed to Step 2: Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server.

Step 2: Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server

After you identify the roles that use the embedded database server, you can migrate from the Cloudera Manager embedded database server to the external PostgreSQL database server. When you migrate, you export the PostgreSQL user roles from the embedded database, import the PostgreSQL user roles into the external database, import the Cloudera Manager database on the external database server, and perform other tasks.

About this task

While performing this procedure, ensure that the Cloudera Manager Agents remain running on all hosts. Unless otherwise specified, when prompted for a password use the cloudera-scm password.

Note: After completing this migration, you cannot delete the cloudera-scm postgres superuser unless you remove the access privileges for the migrated databases. Minimally, you should change the cloudera-scm p ostgres superuser password.

Procedure

1. In Cloudera Manager, stop the cluster services identified in the previous step as using the embedded database server. Be sure to stop the Cloudera Management Service as well. Also be sure to stop any services with dependencies on these services. The remaining CDH services will continue to run without downtime.

   Note: If you do not stop the services from within Cloudera Manager before stopping Cloudera Manager Server from the command line, they will continue to run and maintain a network connection to the embedded database server. If this occurs, then the embedded database server will ignore any command line stop commands (Step 2) and require that you manually kill the process, which in turn causes the services to crash instead of stopping cleanly.
2. Navigate to Hosts > All Hosts, and make note of the number of roles assigned to hosts. Also take note whether or not they are in a commissioned state. You will need this information later to validate that your scm database was migrated correctly.
3. Stop the Cloudera Manager Server. To stop the server:

   ```
   sudo service cloudera-scm-server stop
   ```
4. Obtain and save the embedded database superuser password (you will need this password in subsequent steps) from the generated_password.txt file:

   ```
   head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
   ```

5. Export the PostgreSQL user roles from the embedded database server to ensure the correct users, permissions, and passwords are preserved for database access. Passwords are exported as an md5sum and are not visible in plain text. To export the database user roles (you will need the cloudera-scm user password):

```
pg_dumpall -h localhost -p 7432 -U cloudera-scm -v --roles-only -f "/var/
tmp/cloudera_user_roles.sql"
```

6. Edit /var/tmp/cloudera_user_roles.sql to remove any CREATE ROLE and ALTER ROLE commands for databases not in your list. Leave the entries for cloudera-scm untouched, because this user role is used during the database import.
7. Export the data from each of the databases on your list you created when you identified roles that use the embedded database server:

```
pg_dump -F c -h localhost -p 7432 -U cloudera-scm [database_name]  > /var/
tmp/[database_name]_db_backup-$(date +”%m-%d-%Y”).dump
```

   The following is a sample data export command for the scm database:

```
pg_dump -F c -h localhost -p 7432 -U cloudera-scm scm > /var/tmp/scm_db_
backup-$(date +%m-%d-%Y).dump
```

```
Password:
```

8. Stop and disable the embedded database server:

```
service cloudera-scm-server-db stop
chkconfig cloudera-scm-server-db off
```

   Confirm that the embedded database server is stopped:

```
netstat -at | grep 7432
```

9. Back up the Cloudera Manager Server database configuration file:

```
cp /etc/cloudera-scm-server/db.properties /etc/cloudera-scm-server/db.pr
operties.embedded
```

10. Copy the file /var/tmp/cloudera_user_roles.sql and the database dump files from the embedded database server host to /var/tmp on the external database server host:

```
cd /var/tmp
scp cloudera_user_roles.sql *.dump <user>@<postgres-server>:/var/tmp
```

11. Import the PostgreSQL user roles into the external database server.

    The external PostgreSQL database server superuser password is required to import the user roles. If the superuser role has been changed, you will be prompted for the username and password.

    Note: Only run the command that applies to your context; do not execute both commands.

    - To import users when using the default PostgreSQL superuser role:

      ```
      sudo -u postgres psql -f /var/tmp/cloudera_user_roles.sql
      ```

    - To import users when the superuser role has been changed:

      ```
      psql -h <database-hostname> -p <database-port> -U <superuser> -f /var/tm
      p/cloudera_user_roles.sql
      ```

      For example:

      ```
      psql -h pg-server.example.com -p 5432 -U postgres -f /var/tmp/cloudera_u
      ser_roles.sql
      ```

      ```
      Password for user postgres
      ```

12. Import the Cloudera Manager database on the external server. First copy the database dump files from the Cloudera Manager Server host to your external PostgreSQL database server, and then import the database data:

    Note: To successfully run the pg_restore command, there must be an existing database on the database server to complete the connection; the existing database will not be modified. If the -d <existing-databas e> option is not included, then the pg_restore command will fail.

    ```
    pg_restore -C -h <database-hostname> -p <database-port> -d <existing-dat
    abase> -U cloudera-scm -v <data-file>
    ```

    Repeat this import for each database.

    The following example is for the scm database:

    ```
    pg_restore -C -h pg-server.example.com -p 5432 -d postgres -U cloudera-scm
     -v /var/tmp/scm_server_db_backup-20180312.dump
    ```

    ```
    pg_restore: connecting to database for restore
    Password:
    ```

13. Update the Cloudera Manager Server database configuration file to use the external database server. Edit the /etc/cloudera-scm-server/db.properties file as follows:

    a) Update the com.cloudera.cmf.db.host value with the hostname and port number of the external database server.
    b) Change the com.cloudera.cmf.db.setupType value from "EMBEDDED" to "EXTERNAL".

14. Start the Cloudera Manager Server and confirm it is working:

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

    Note that if you start the Cloudera Manager GUI at this point, it may take up to five minutes after executing the start command before it becomes available.

    In Cloudera Manager Server, navigate to Hosts > All Hosts and confirm the number of roles assigned to hosts (this number should match what you found in Step 2); also confirm that they are in a commissioned state that matches what you observed in Step 2.

15. Update the role configurations to use the external database hostname and port number. Only perform this task for services where the database has been migrated.
    - For Cloudera Management Services:
        a. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
        b. Change any database hostname properties from the embedded database to the external database hostname and port number.
        c. Click Save Changes.
    - For the Oozie Service:
        a. Select Oozie service > Configuration, and type "7432" in the Search field.
        b. Change any database hostname properties from the embedded database to the external database hostname and port number.
        c. Click Save Changes.
    - For Hive, Hue, and Sentry Services:
        a. Select the specific service > Configuration, and type "database host" in the Search field.
        b. Change the hostname from the embedded database name to the external database hostname.
        c. Click Save Changes.
16. Start the Cloudera Management Service and confirm that all management services are up and no health tests are failing.
17. Start all Services via the Cloudera Manager web UI. This should start all services that were stopped for the database migration. Confirm that all services are up and no health tests are failing.
18. On the embedded database server host, remove the embedded PostgreSQL database server:
    a) Make a backup of the /var/lib/cloudera-scm-server-db/data directory:

    ```
    tar czvf /var/tmp/embedded_db_data_backup-$(date +"%m-%d-%Y").tgz /var/l
    ib/cloudera-scm-server-db/data
    ```

    b) Remove the embedded database package:

    For RHEL/SLES:

    ```
    rpm --erase cloudera-manager-server-db-2
    ```

    For Debian/Ubuntu:

    ```
    apt-get remove cloudera-manager-server-db-2
    ```

    c) Delete the /var/lib/cloudera-scm-server-db/data directory.

Migrating from the Cloudera Manager External PostgreSQL Database Server to a MySQL/Oracle Database Server

Cloudera Manager provides an embedded PostgreSQL database server for demonstration and proof of concept deployments when creating a cluster. To remind users that this embedded database is not suitable for production, Cloudera Manager displays the banner text: "You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production."

If you have already used the embedded database, and you are unable to redeploy a fresh cluster, then you must migrate to an external PostgreSQL database.

Note: You can migrate to an external MySQL or Oracle database only after successfully migrating from the embedded PostgreSQL database server to the external PostgreSQL database server.

Prerequisites

Before migrating from the Cloudera Manager external PostgreSQL database to an external MySQL/Oracle database, ensure that your setup meets the following conditions:

- Configuration uses Cloudera Manager 5.15.0 or later on supported platforms.
- You must have a valid Cloudera Manager Enterprise license.
- If Cloudera Manager is secured, then you must import Kerberos account manager credentials and regenerate them.
- You must have a destination host installed with the supported database of choice (MySQL or Oracle). For details about installing and configuring MySQL for Cloudera, see the topic *Install and Configure MySQL for Cloudera Software*. For details about installing and configuring Oracle for Cloudera, see the topic *Install and Configure Oracle Database Software for Cloudera Software*.
- You have made configured target database hosts available.
- You have planned for cluster downtime during the migration process.
- You have a plan to follow service specific database migration instructions for services other than Cloudera Manager. Refer to the appropriate service migration documentation for your cluster setup.
- No one has manually created any databases in the external database server for roles that will be migrated.
- All health issues with your cluster are resolved.

For large clusters, Cloudera recommends running your database server on a dedicated host. Engage Cloudera Professional Services or a certified database administrator to correctly tune your external database server.

Migrate from the Cloudera Manager External PostgreSQL Database Server to a MySQL/Oracle Database Server

When you migrate from the Cloudera Manager External PostgreSQL database server to a MySQL or Oracle database server, you export the Cloudera Manager configuration, prepare the target database for Cloudera Manager, and complete other tasks.

Procedure

1. Migrate from the embedded PostgreSQL database server to an external PostgreSQL database server as described in the topic *Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database*.

   Important: Migrating directly from the Cloudera Manager embedded PostgreSQL to a MySQL/Oracle database is not supported. You must first migrate from the Cloudera Manager embedded PostgreSQL database server to the external PostgreSQL database server. After performing this migration, you can use this procedure to migrate from the external PostgreSQL database server to MySQL/Oracle database servers.

2. Export your Cloudera Manager Configuration. First, get the latest supported API version:

```
curl -u <admin_username>:<admin_password> "http://<cm_server_host>:7180/
api/version"
```

```
curl -u <admin_username>:<admin_password> "http://<cm_server_host>:7180/
api/<api_version> /cm/deployment" > <path_to_file>/cm-deployment.json
```

The following is an example of the API version command:

```
curl -u admin:admin "http://10.17.103.191:7180/api/v19/cm/deployment" > /
root/cm-deployment.json
```

Note:

If you have Cloudera Manager with TLS for the Admin Console enabled, retrieve the certificate file and use curl with the --cacert option:

```
curl --cacert <certificate_file> -u admin:admin "https://<cm_server_
host>:7183/api/version"
```

3. Preserve Cloudera Manager's GUID by running the following command to create a /etc/cloudera-scm-server/uuid file. On a host that has an agent, run:

```
sudo -u postgres psql -qtAX scm -c "select GUID from CM_VERSION" > uuid
```

Note: Check to confirm the name of your Cloudera Manager database in /etc/cloudera-scm-server/db.properties.

Then move the UUID file to Cloudera Manager server's /etc/cloudera-scm-server directory.

4. Stop the cluster and the Cloudera Management Services.
5. Stop the Cloudera Manager Server:

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

Note:

For RHEL/CentOS 7, use the systemctl option instead:

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

6. Prepare the target database for Cloudera Manager. For details, refer to the topics *Install and Configure MySQL for Cloudera Software* or *Install and Configure Oracle Database for Cloudera Software*.
7. The process directory (/var/run/cloudera-scm-agent/process/) must be cleaned out for all of the hosts that have agents running on them. The agent completes this cleanup with a server reboot. However, if a server reboot is not a viable option, use one of the following options to accomplish the same task.

Note: This "hard restart" works for all supported platforms except SLES 12.

- **a.** Stop the agent and supervisor:

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

b. Confirm that the agent and supervisor process are stopped:

```
ps -ef | grep -i cmf-agent; ps -ef | grep -i supervisor
```

c. Perform a clean start:

```
service cloudera-scm-agent next_start_clean
```

Alternatively, run the following command to view the start options available on your platform:

```
service cloudera-scm-agent clean_start
```

d. Ensure that the process is empty:

```
ls -la /var/run/cloudera-scm-agent/process/
```

- Alternatively:

a. Stop the agent and supervisor:

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

b. Confirm that the agent and supervisor process are stopped:

```
ps -ef | grep -i cmf-agent; ps -ef | grep -i supervisor
```

c. Move the existing /var/run/cloudera-scm-agent/ directory:

```
mv /var/run/cloudera-scm-agent /var/run/cloudera-scm-agent-BU
```

The agent will recreate the directory. Delete the backed up copy after confirming that the migration was successful.

8. Start the Cloudera Manager server:

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

9. Log in to Cloudera Manager. Exit the installation wizard by clicking the product log in the upper-left corner to stop the wizard and return to the Cloudera Manager home page.
10. Upgrade the Cloudera Manager Enterprise License by navigating to Administration > Licenses and installing a valid Cloudera Manager license.
11. Restore the Cloudera Manager configuration:

```
curl -H "Content-Type: application/json" --upload-file <path_to_file>/cm
-deployment.json -u <admin_username>:<admin_password> "http://<cm_server
_host>:7180/api/<api_version>/cm/deployment?deleteCurrentDeployment=true"
```

The following example shows how to restore a Cloudera Manager configuration:

```
curl -H "Content-Type: application/json" --upload-file /root/cm-deployme
nt.json -u admin:admin "http://172.31.113.146:7180/api/v19/cm/deployment?
deleteCurrentDeployment=true"
```

12. Start the following: Cloudera Management Service, Host Monitor, and Services Monitor. Verify that all the services in the Cloudera Management Service started and are Healthy.
13. Select the Home > Status tab for the cluster(s) that you previously stopped, and in the Actions dropdown, select Start.

Managing Cloudera Manager Server Logs

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

Related Information

Logs

Viewing the Cloudera Manager Server Logs

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

Procedure

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

What to do next

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

Setting the Cloudera Manager Server Log Location

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

Procedure

1. Stop the Cloudera Manager Server:

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

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

```
export CMF_VAR=/opt
```

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

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

4. Restart the Cloudera Manager Server:

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

Configuring Cloudera Manager

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

Settings

The Settings page provides a number of categories as follows:

- Performance - Set the Cloudera Manager Agent heartbeat interval.
- Advanced - Enable API debugging and other advanced options.
- Monitoring - Set Agent health status parameters. For configuration instructions, see the topic *Configuring Cloudera Manager Agents*.

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

You can also configure the following:

- Alerts
- Users
- Language

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

Related Information

Managing Anonymous Usage Data Collection
Diagnostic Data Collection

Cloudera Manager Agents

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

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

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

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

cm_processes

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

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

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

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

Related Information

supervisord
tmpfs

Starting, Stopping, and Restarting Cloudera Manager Agents

Starting Agents

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

- Start

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

Stopping and Restarting Agents

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

- Stop

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

- Restart

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

Hard Stopping and Restarting Agents

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

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

- Hard Stop
  RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

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

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

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

- Hard Restart
  RHEL 7, SLES 12, Debian 8, Ubuntu 16.04

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

```
sudo systemctl restart cloudera-scm-agent
```

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

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

Hard restart is useful for the following situations:

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

Checking Agent Status

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

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

Managing the Cloudera Manager Agent Logs

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

Viewing the Cloudera Manager Agent Logs

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

Procedure

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

What to do next

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

Setting the Cloudera Manager Agent Log Location

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

Procedure

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

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

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

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

```
mkdir /opt/log/cloudera-scm-agent
chown cloudera-scm:cloudera-scm log/cloudera-scm-agent
```

3. Restart the Agent:

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

Default User Roles

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

The following 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								

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 resource.

Other Cloudera Manager Tasks and Settings

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

Settings

The Settings page provides a number of categories as follows:

- Performance - Set the Cloudera Manager Agent heartbeat interval.
- Advanced - Enable API debugging and other advanced options.
- Monitoring - Set Agent health status parameters.
- Security - Set TLS encryption settings to enable TLS encryption between the Cloudera Manager Server, Agents, and clients. You can also:
  - Set the realm for Kerberos security and point to a custom keytab retrieval script.
  - Specify session timeout and a "Remember Me" option.
- Ports and Addresses - Set ports for the Cloudera Manager Admin Console and Server.
- Other
  - Enable Cloudera usage data collection.
  - Set a custom header color and banner text for the Admin console.
  - Set an "Information Assurance Policy" statement – this statement will be presented to every user before they are allowed to access the login dialog box. The user must click "I Agree" in order to proceed to the login dialog box.
  - Disable/enable the auto-search for the Events panel at the bottom of a page.
- Support
  - Configure diagnostic data collection properties.
  - Configure how to access Cloudera Manager help files.
- External Authentication - Specify the configuration to use LDAP, Active Directory, or an external program for authentication.
- Parcels - Configure settings for parcels, including the location of remote repositories that should be made available for download, and other settings such as the frequency with which Cloudera Manager will check for new parcels, limits on the number of downloads or concurrent distribution uploads. See Parcels for more information.
- Network - Configure proxy server settings.
- Custom Service Descriptors - Configure custom service descriptor properties for Add-on services.

Alerts

See *Managing Alerts*.

Users

See *Cloudera Manager User Accounts*.

Kerberos

See *Enabling Kerberos Authentication for Cloudera Runtime*.

License

See *Managing Licenses*.

User Interface Language

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

Peers

See *Designating a Replication Source*.

Cloudera Management Service

The Cloudera Management Service is a set of roles used by Cloudera Manager to manage and monitor clusters.

The Cloudera Management Service implements various management features as a set of roles:

- Host Monitor - collects health and metric information about hosts
- Service Monitor - collects health and metric information about services and activity information from the YARN and Impala services
- Event Server - aggregates relevant Hadoop events and makes them available for alerting and searching
- Alert Publisher - generates and delivers alerts for certain types of events
- Reports Manager - generates reports that provide an historical view into disk utilization by user, user group, and directory, processing activities by user and YARN pool, and HBase tables and namespaces. This role is not added in Cloudera Express.

You can view the status of the Cloudera Management Service by doing one of the following:

- Select Clusters Cloudera Management Service .
- On the HomeStatus tab, in Cloudera Management Service table, click the Cloudera Management Service link.

Health Tests

Cloudera Manager monitors the health of the services, roles, and hosts that are running in your clusters using *health tests*. The Cloudera Management Service also provides health tests for its roles. Role-based health tests are enabled by default. For example, a simple health test is whether there's enough disk space in every NameNode data directory. A more complicated health test may evaluate when the last checkpoint for HDFS was compared to a threshold or whether a DataNode is connected to a NameNode. Some of these health tests also aggregate other health tests: in a distributed system like HDFS, it's normal to have a few DataNodes down (assuming you've got dozens of hosts), so we allow for setting thresholds on what percentage of hosts should color the entire service down.

Health tests can return one of three values: Good, Concerning, and Bad. A test returns Concerning health if the test falls below a warning threshold. A test returns Bad if the test falls below a critical threshold. The overall health of a service or role instance is a roll-up of its health tests. If any health test is Concerning (but none are Bad) the role's or service's health is Concerning; if any health test is Bad, the service's or role's health is Bad.

In the Cloudera Manager Admin Console, health tests results are indicated with colors: Good  , Concerning  , and Bad  .

One common question is whether monitoring can be separated from configuration. One of the goals for monitoring is to enable it without needing to do additional configuration and installing additional tools (for example, Nagios). By having a deep model of the configuration, Cloudera Manager is able to know which directories to monitor, which ports to use, and what credentials to use for those ports. This tight coupling means that, when you install Cloudera Manager all the monitoring is enabled.

Metric Collection and Display

To perform monitoring, the Service Monitor and Host Monitor collects metrics. A *metric* is a numeric value, associated with a name (for example, "CPU seconds"), an entity it applies to ("host17"), and a timestamp. Most metric collection is performed by the Agent. The Agent communicates with a supervised process, requests the metrics, and forwards them to the Service Monitor. In most cases, this is done once per minute.

A few special metrics are collected by the Service Monitor. For example, the Service Monitor hosts an HDFS canary, which tries to write, read, and delete a file from HDFS at regular intervals, and measure whether it succeeded, and how long it took. Once metrics are received, they're aggregated and stored.

Using the Charts page in the Cloudera Manager Admin Console, you can query and explore the metrics being collected. Charts display *time series*, which are streams of metric data points for a specific entity. Each metric data point contains a timestamp and the value of that metric at that timestamp.

Some metrics (for example, total_cpu_seconds) are counters, and the appropriate way to query them is to take their rate over time, which is why a lot of metrics queries contain the dt0 function. For example, dt0(total_cpu_seconds). (The dt0 syntax is intended to remind you of derivatives. The `0` indicates that the rate of a monotonically increasing counter should never have negative rates.)

Events, Alerts, and Triggers

An *event* is a record that something of interest has occurred – a service's health has changed state, a log message (of the appropriate severity) has been logged, and so on. Many events are enabled and configured by default.

An *alert* is an event that is considered especially noteworthy and is triggered by a selected event. Alerts are shown with an 

badge when they appear in a list of events. You can configure the Alert Publisher to send alert notifications by email or by SNMP trap to a trap receiver.

A *trigger* is a statement that specifies an action to be taken when one or more specified conditions are met for a service, role, role configuration group, or host. The conditions are expressed as a tsquery statement, and the action to be taken is to change the health for the service, role, role configuration group, or host to either Concerning (yellow) or Bad (red).

Starting the Cloudera Management Service

How to start the Cloudera Management Service.

Before you begin

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. Do one of the following:
   - **a.** Select Clusters Cloudera Management Service .
     b. Select ActionsStart.
   - On the HomeStatus tab, click the options menu to the right of Cloudera Management Service and select Start.
2. Click Start to confirm. The Command Details window shows the progress of starting the roles.

Results

When Command completed with *n*/*n* successful subcommands appears, the task is complete. Click Close.

Stopping the Cloudera Management Service

How to stop the Cloudera Management Service.

Before you begin

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. Do one of the following:
   - a. Select Clusters Cloudera Management Service .
     b. Select ActionsStop.
   - On the HomeStatus tab, click the options menu to the right of Cloudera Management Service and select Stop.
2. Click Stop to confirm. The Command Details window shows the progress of stopping the roles.

Results

When Command completed with *n*/*n* successful subcommands appears, the task is complete. Click Close.

Restarting the Cloudera Management Service

How to restart the Cloudera Management Service.

Before you begin

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. Do one of the following:
   - a. Select Clusters Cloudera Management Service .
     b. Select ActionsRestart.
   - On the HomeStatus tab, click the options menu to the right of Cloudera Management Service and select Restart.
2. Click Restart to confirm. The Command Details window shows the progress of restarting the roles.

Results

When Command completed with *n*/*n* successful subcommands appears, the task is complete. Click Close.

Starting and Stopping Cloudera Management Service Roles

Before you begin

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Procedure

1. Do one of the following:
   - Select Clusters Cloudera Management Service .
   - On the HomeStatus tab, in Cloudera Management Service table, click the Cloudera Management Service link.
2. Click the Instances tab.
3. Select a role.
4. Do one of the following:s
   - Start: Select Actions for SelectedStart and click Start to confirm
   - Stop: Select Actions for SelectedStop and click Stop to confirm.

Results
When Command completed with *n*/*n* successful subcommands appears, the task is complete. Click Close.

Configuring Management Service Database Limits

Configuring database service limits lets you control the amount of retained monitoring data.

Before you begin

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

About this task

Each Cloudera Management Service role maintains a database for retaining the data it monitors. These databases (as well as the log files maintained by these services) can grow quite large. Limits on these data sets are configured when you create the management services, but you can modify these parameters through the Configuration settings in the Cloudera Manager Admin Console. For example, the Event Server lets you set a total number of events to store.

There are also settings for the logs that these various services create. You can throttle how big the logs are allowed to get and how many previous logs to retain.

Procedure

1. Do one of the following:
   - Select Clusters Cloudera Management Service .
   - On the HomeStatus tab, in Cloudera Management Service table, click the Cloudera Management Service link.
2. Click the Configuration tab.
3. Select Scope and then one of the following.
   - Host Monitor
   - Service Monitor
4. Select CategoryLog Files to view log file size properties.
5. Edit the appropriate properties.

   To apply this configuration property to other role groups as needed, edit the value for the appropriate role group. See .
6. Click Save Changes.

Related Information
Data Storage for Monitoring Data

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 CDH 5 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 (JobConf):

- 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

For an overview of compression, see *Data Compression*.

Guidelines for Choosing a Compression Type

- 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.
- For MapReduce, if you need your compressed data to be splittable, BZip2 and LZO formats can be split. Snappy and GZip blocks are not splittable, but files with Snappy blocks inside a container file format such as SequenceFile or Avro can be split. Snappy is intended to be used with a container format, like SequenceFiles or Avro data files, rather than being used directly on plain text, for example, since the latter is not splittable and cannot be processed in parallel using MapReduce. Splittability is not relevant to HBase data.
- For MapReduce, you can compress either the intermediate data, the output, or both. Adjust the parameters you provide for the MapReduce job accordingly. The following examples compress both the intermediate data and the output. MR2 is shown first, followed by MR1.
  - MRv2

```
hadoop jar hadoop-examples-.jar sort "-Dmapreduce.compress.map.output=tr
ue"
"-Dmapreduce.map.output.compression.codec=org.apache.hadoop.io.compress.
GzipCodec"
"-Dmapreduce.output.compress=true"
"-Dmapreduce.output.compression.codec=org.apache.hadoop.io.compress.Gz
ipCodec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input out
put
```

  - MRv1

```
hadoop jar hadoop-examples-.jar sort "-Dmapred.compress.map.output=true"
"-Dmapred.map.output.compression.codec=org.apache.hadoop.io.compress.Gz
ipCodec"
"-Dmapred.output.compress=true"
"-Dmapred.output.compression.codec=org.apache.hadoop.io.compress.GzipCo
dec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input out
put
```

Configuring Data Compression

To configure support for LZO using Cloudera Manager, you must install the GPL Extras parcel, then configure services to use it. See *Installing the GBL Extras Parcel* and *Configuring Services to Use the GPL Extras Parcel*.

Resource Management

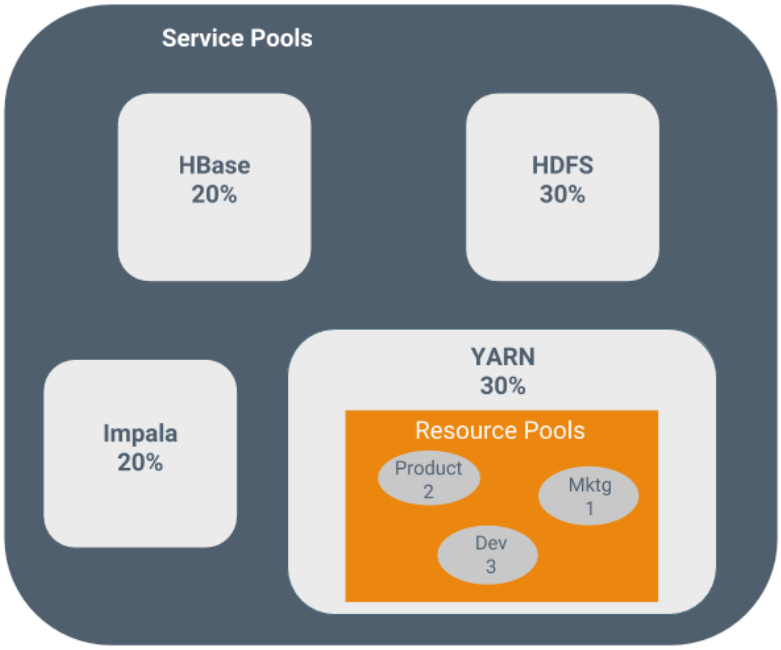
Resource management helps ensure predictable behavior by defining the impact of different services on cluster resources.

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% x 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.

Enabling and Configuring Static Service Pools

To enable and configure static service pools, you enter the percentage of resources to allocate to each service and then restart the cluster.

Before you begin

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. Select Clusters*Cluster name*Static Service Pools.
2. Click the Configuration tab.
   The Step 1 of 4: Basic Allocation Setup page displays. In each field in the basic allocation table, enter the percentage of resources to give to each service. The total must add up to 100%.
3. Click Continue to proceed.
   Step 2: Review Changes - The allocation of resources for each resource type and role displays with the new values as well as the values previously in effect. The values for each role are set by role group; if there is more than one role group for a given role type (for example, for RegionServers or DataNodes) then resources will be allocated separately for the hosts in each role group.
4. Take note of changed settings. If you have previously customized these settings, check these over carefully:
   - Click the › to the right of each percentage to display the allocations for a single service. Click › to the right of the Total (100%) to view all the allocations in a single page.
   - Click the Back button to go to the previous page and change your allocations.
5. When you are satisfied with the allocations, click Continue.
   The Step 3 of 4: Restart Services page displays.
6. To apply the new allocation percentages, click Restart Now to restart the cluster. To skip this step, click Restart Later. If HDFS High Availability is enabled, you will have the option to choose a rolling restart.
7. Step 4 of 4: Progress displays the status of the restart commands. Click Finished after the restart commands complete.

After you enable static service pools, there are three additional tasks:

8. Delete everything under the local directory path on NodeManager hosts. The local directory path is configurable, and can be verified in Cloudera Manager with YARN Configuration NodeManager Local Directories .
9. Enable cgroups for resource management. You can enable cgroups in Cloudera Manager with Yarn Configuration Use CGroups for Resource Management .
10. If you are using the optional Impala scratch directory, delete all files in the Impala scratch directory. The directory path is configurable, and can be verified in Cloudera Manager with Impala Configuration Impala Daemon Scratch Directories .

Disabling Static Service Pools

To disable static service pools, disable cgroup-based resource management for all hosts in all clusters.

Before you begin

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. In the main navigation bar, click Hosts.
2. Click the Configuration tab.
3. Select ScopeResource Management.
4. Clear the Enable Cgroup-based Resource Management property.
5. Click Save Changes.
6. Restart all services.

Results

Static resource management is disabled, but the percentages you set when you configured the pools, and all the changed settings (for example, heap sizes), are retained by the services. The percentages and settings will also be used when you re-enable static service pools. If you want to revert to the settings you had before static service pools were enabled, follow the procedures in *Viewing and Reverting Configuration Changes*.

Linux Control Groups (cgroups)

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Cloudera Manager supports the Linux control groups (cgroups) kernel feature. With cgroups, administrators can impose per-resource restrictions and limits on services and roles. This provides the ability to allocate resources using cgroups to enable isolation of compute frameworks from one another. Resource allocation is implemented by setting properties for the services and roles.

Linux Distribution Support

Cgroups are a feature of the Linux kernel, and as such, support depends on the host's Linux distribution and version as shown in the following tables. If a distribution lacks support for a given parameter, changes to the parameter have no effect.

Table 2: RHEL-compatible

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Red Hat Enterprise Linux, CentOS, and Oracle Enterprise Linux 7	■	■	■	■
Red Hat Enterprise Linux, CentOS, and Oracle Enterprise Linux 6	■	■	■	■

Table 3: SLES

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
SUSE Linux Enterprise Server 12	■	■	■	■
SUSE Linux Enterprise Server 11	■	■	■	■

Table 4: Ubuntu

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Ubuntu 16.04 LTS	■	■	■	■

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Ubuntu 16.04 LTS	■	■	■	■
Ubuntu 14.04 LTS	■	■	■	■
Ubuntu 12.04 LTS	■	■	■	■

Table 5: Debian

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Debian 7.1	■	■	■	■
Debian 7.0	■	■	■	■
Debian 6.0	■	■	■	■

Table 6: Oracle linux (OL)

Distribution	CPU Shares	I/O Weight	Memory Soft Limit	Memory Hard Limit
Oracle linux 7	■	■	■	■
Oracle linux 6	■	■	■	■

The exact level of support can be found in the Cloudera Manager Agent log file, shortly after the Agent has started. In the log file, look for an entry like this:

```
Found cgroups capabilities: {
'has_memory': True,
'default_memory_limit_in_bytes': 9223372036854775807,
'writable_cgroup_dot_procs': True,
'has_cpu': True,
'default_blkio_weight': 1000,
'default_cpu_shares': 1024,
'has_blkio': True}
```

The has_cpu and similar entries correspond directly to support for the CPU, I/O, and memory parameters.

Resource Management with Control Groups

To use cgroups, you must enable cgroup-based resource management under the host resource management configuration properties. However, if you configure static service pools, this property is set as part of that process.

About this task

Cgroups-based resource management can be enabled for all hosts, or on a per-host basis.

Procedure

1. If you have upgraded from a version of Cloudera Manager older than Cloudera Manager 4.5, restart every Cloudera Manager Agent before using cgroups-based resource management:
   a) Stop all services, including the Cloudera Management Service.
   b) On each cluster host, run as root:
      - RHEL-compatible 7 and higher:

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

- All other Linux distributions:

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

c) Start all services.

2. Click the Hosts tab.
3. Optionally click the link of the host where you want to enable cgroups.
4.
5. Select CategoryResource Management.
6. Select Enable Cgroup-based Resource Management.
7. Restart all roles on the host or hosts.

Limitations

- Role group and role instance override cgroup-based resource management parameters must be saved one at a time. Otherwise some of the changes that should be reflected dynamically will be ignored.
- The role group abstraction is an imperfect fit for resource management parameters, where the goal is often to take a numeric value for a host resource and distribute it amongst running roles. The role group represents a "horizontal" slice: the same role across a set of hosts. However, the cluster is often viewed in terms of "vertical" slices, each being a combination of worker roles (such as TaskTracker, DataNode, RegionServer, Impala Daemon, and so on). Nothing in Cloudera Manager guarantees that these disparate horizontal slices are "aligned" (meaning, that the role assignment is identical across hosts). If they are unaligned, some of the role group values will be incorrect on unaligned hosts. For example a host whose role groups have been configured with memory limits but that's missing a role will probably have unassigned memory.

Configuring Resource Parameters

After enabling cgroups, you can restrict and limit the resource consumption of roles (or role groups) on a per-resource basis.

All of these parameters can be found in the Cloudera Manager Admin Console, under the Resource Management category:

- CPU Shares - The more CPU shares given to a role, the larger its share of the CPU when under contention. Until processes on the host (including both roles managed by Cloudera Manager and other system processes) are contending for all of the CPUs, this will have no effect. When there is contention, those processes with higher CPU shares will be given more CPU time. The effect is linear: a process with 4 CPU shares will be given roughly twice as much CPU time as a process with 2 CPU shares.

  Updates to this parameter are dynamically reflected in the running role.
- I/O Weight - The greater the I/O weight, the higher priority will be given to I/O requests made by the role when I/O is under contention (either by roles managed by Cloudera Manager or by other system processes).

  This only affects read requests; write requests remain unprioritized. The Linux I/O scheduler controls when buffered writes are flushed to disk, based on time and quantity thresholds. It continually flushes buffered writes from multiple sources, not certain prioritized processes.

  Updates to this parameter are dynamically reflected in the running role.
- Memory Soft Limit - When the limit is reached, the kernel will reclaim pages charged to the process if and only if the host is facing memory pressure. If reclaiming fails, the kernel may kill the process. Both anonymous as well as page cache pages contribute to the limit.

  After updating this parameter, you must restart the role for changes to take effect.
- Memory Hard Limit - When a role's resident set size (RSS) exceeds the value of this parameter, the kernel will swap out some of the role's memory. If it is unable to do so, it will kill the process. The kernel measures memory consumption in a manner that does not necessarily match what the top or ps report for RSS, so expect that this limit is a rough approximation.

  After updating this parameter, you must restart the role for changes to take effect.

Example: Protecting Production MapReduce Jobs from Impala Queries

Suppose you have MapReduce deployed in production and want to roll out Impala without affecting production MapReduce jobs. For simplicity, we will make the following assumptions:

- The cluster is using homogenous hardware
- Each worker host has two cores
- Each worker host has 8 GB of RAM
- Each worker host is running a DataNode, TaskTracker, and an Impala Daemon
- Each role type is in a single role group
- Cgroups-based resource management has been enabled on all hosts

Action	Procedure
CPU	1. Leave DataNode and TaskTracker role group CPU shares at 1024. 2. Set Impala Daemon role group's CPU shares to 256. 3. The TaskTracker role group should be configured with a Maximum Number of Simultaneous Map Tasks of 2 and a Maximum Number of Simultaneous Reduce Tasks of 1. This yields an upper bound of three MapReduce tasks at any given time; this is an important detail for memory sizing.
Memory	1. Set Impala Daemon role group memory limit to 1024 MB. 2. Leave DataNode maximum Java heap size at 1 GB. 3. Leave TaskTracker maximum Java heap size at 1 GB. 4. Leave MapReduce Child Java Maximum Heap Size for Gateway at 1 GB. 5. Leave cgroups hard memory limits alone. We'll rely on "cooperative" memory limits exclusively, as they yield a nicer user experience than the cgroups-based hard memory limits.
I/O	1. Leave DataNode and TaskTracker role group I/O weight at 500. 2. Impala Daemon role group I/O weight is set to 125.

When you're done with configuration, restart all services for these changes to take effect. The results are:

1. When MapReduce jobs are running, all Impala queries together will consume up to a fifth of the cluster's CPU resources.
2. Individual Impala Daemons will not consume more than 1 GB of RAM. If this figure is exceeded, new queries will be cancelled.
3. DataNodes and TaskTrackers can consume up to 1 GB of RAM each.
4. We expect up to 3 MapReduce tasks at a given time, each with a maximum heap size of 1 GB of RAM. That's up to 3 GB for MapReduce tasks.
5. The remainder of each host's available RAM (6 GB) is reserved for other host processes.
6. When MapReduce jobs are running, read requests issued by Impala queries will receive a fifth of the priority of either HDFS read requests or MapReduce read requests.

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

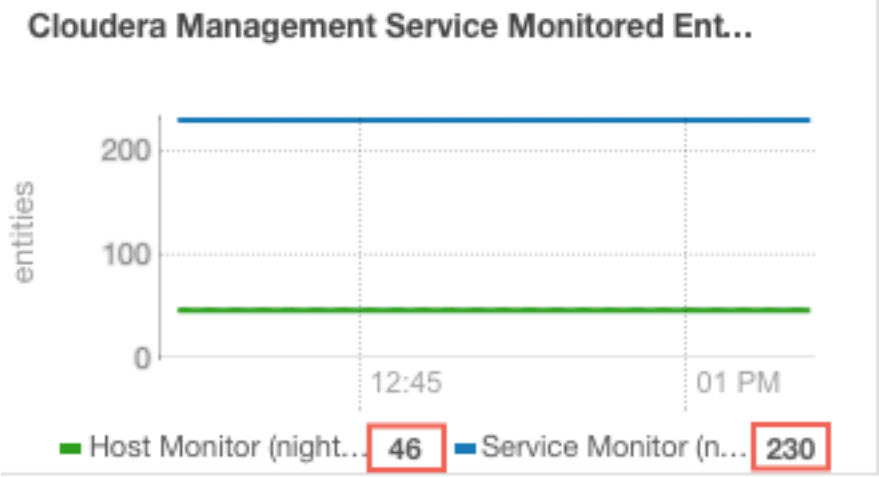
To determine the number of entities being monitored:

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
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>
```

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

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

   - Spark:

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

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

Reference the managed configuration files and add AWS credentials

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

1. Create a local directory under your home directory.
2. Copy the configuration files from /etc/hadoop/conf to the new directory.
3. Set the permissions for the configuration files appropriately for your environment.

4. Edit each configuration file:
   a. Remove all elements within the <configuration> element.
   b. Add an XML <include> element within the <configuration> element to reference the configuration files managed by Cloudera Manager. For example:

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

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

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

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

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

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

   - Spark:

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

Example core-site.xml file:

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

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

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

Referencing Amazon S3 in URIs

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

- Amazon S3:

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

- HDFS:

```
hdfs://path
```

or

```
/path
```

Related Information

Accessing Data Stored in Amazon S3 through Spark
Impala with Amazon S3

Using Fast Upload with Amazon S3

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

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

Related Information

Hadoop-AWS module: Integration with Amazon Web Services

Enabling Fast Upload using Cloudera Manager

Procedure

To enable Fast Upload for clusters managed by Cloudera Manager:

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

Results

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

Related Information

Setting an Advanced Configuration Snippet for a Cluster
Restarting a Cloudera Runtime Service

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

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

Procedure

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

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

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

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

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

   ```
   cp: cannot open `/etc/hadoop/conf/container-executor.cfg' for reading: P
   ermission 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 terag
     en 100 s3a://S3_Bucket/teragen_test
     ```

   - teragen (parcel-based installations)

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

Importing Data into Amazon S3 Using Sqoop

Sqoop supports data import from RDBMS into Amazon S3.

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

Related Information

Hadoop-AWS module: Integration with Amazon Web Services

Authentication

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

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

Amazon S3 Block Filesystem URI example:

s3a://bucket_name/path/to/file

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

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

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

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

Credentials can be provided through the command line:

```
sqoop import -Dfs.s3a.access.key=... -Dfs.s3a.secret.key=... --target-dir s3
a://
```

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 j
   ceks://hdfs/path_to_credential_store_file
   hadoop credential create fs.s3a.secret.key -value secret_key -provider j
   ceks://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 --t
able $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 --t
able $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 --t
able $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 --t
able $TABLENAME --target-dir s3a://example-bucket/target-directory --as-pa
rquetfile --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 --t
able $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incre
mental append --check-column $CHECK_COLUMN --last-value $LAST_VALUE --tempo
rary-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 --t
able $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incre
mental append --check-column $CHECK_COLUMN --last-value $LAST_VALUE --tempo
rary-rootdir s3a://example-bucket/temporary-rootdir --as-parquetfile --parq
uet-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 --t
able $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incre
mental 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 --t
able $TABLE_NAME --target-dir s3a://example-bucket/target-directory --incre
mental 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 --t
able $TABLE_NAME --hive-import --create-hive-table --hs2-url $HS2_URL --h
s2-user $HS2_USER --hs2-keytab $HS2_KEYTAB --hive-table $HIVE_TABLE_NAME --t
arget-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-ta
ble $HIVE_TABLE_NAME --target-dir s3a://example-bucket/target-directory --e
xternal-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-ta
ble $HIVE_TABLE_NAME --target-dir s3a://example-bucket/target-directory --e
xternal-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 --t
able $TABLE_NAME --hive-import --hs2-url $HS2_URL --hs2-user $HS2_USER --h
s2-keytab $HS2_KEYTAB --target-dir s3a://example-bucket/target-directory --e
xternal-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 --t
able $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 --t
able $TABLE_NAME --hive-import --target-dir s3a://example-bucket/target-dire
ctory --external-table-dir s3a://example-bucket/external-directory --as-pa
rquetfile --parquet-configurator-implementation hadoop
```