

Cloudera Manager 7.0.1

## Configuring Data Hub Clusters

Date published: 2020-02-11

Date modified:

# CLOUDERA

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

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

<b>Accessing the Cloudera Manager Admin Console from Data Hub clusters.....</b>	<b>5</b>
<b>Modifying Configuration Properties Using Cloudera Manager.....</b>	<b>5</b>
Changing the Configuration of a Service or Role Instance.....	5
Searching for Properties.....	8
Validation of Configuration Properties.....	8
Overriding Configuration Properties.....	8
Viewing and Editing Overridden Configuration Properties.....	8
Resetting Configuration Properties to the Default Value.....	9
Viewing and Editing Host Overrides.....	9
Restarting Services and Instances after Configuration Changes.....	11
Suppressing Configuration and Parameter Validation Warnings.....	12
Suppressing a Configuration Validation in Cloudera Manager.....	12
Managing Suppressed Validations.....	12
Suppressing Configuration Validations Before They Trigger Warnings.....	12
Viewing a List of All Suppressed Validations.....	13
<b>Cluster-Wide Configuration.....</b>	<b>13</b>
<b>Custom Configuration.....</b>	<b>14</b>
<b>Setting an Advanced Configuration Snippet for a Runtime Component.....</b>	<b>16</b>
<b>Setting an Advanced Configuration Snippet for a Cluster.....</b>	<b>17</b>
<b>Stale Configurations.....</b>	<b>17</b>
<b>Client Configuration Files.....</b>	<b>19</b>
How Client Configurations are Deployed.....	20
Downloading Client Configuration Files.....	20
Manually Redeploying Client Configuration Files.....	20
<b>Viewing and Reverting Configuration Changes.....</b>	<b>21</b>
Viewing Configuration Changes.....	21
Reverting Configuration Changes.....	21
<b>Autoconfiguration.....</b>	<b>22</b>
Autoconfiguration.....	22

Configuration Scope.....	22
Data Directories.....	23
Memory.....	23
General Rules.....	27
Role-Host Placement.....	28

## 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.

## Modifying Configuration Properties Using Cloudera Manager

When a service is added to Cloudera Manager, either through the installation or upgrade wizard or with the Add Services workflow, Cloudera Manager automatically sets the configuration properties, based on the needs of the service and characteristics of the cluster in which it will run. These configuration properties include both service-wide configuration properties, as well as specific properties for each role type associated with the service, managed through role groups. 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.

### Related Information

[Role Groups](#)

## Changing the Configuration of a Service or Role Instance

### Procedure

1. In the left menu, click `Clusters` *service name* .
2. Click the Configuration tab.

3. Locate the property you want to edit. You can type all or part of the property name in the search box, or use the filters on the left side of the screen.

- Status

The Status section limits the displayed properties by their status. Possible statuses include:

- Error
- Warning
- Edited
- Non-default
- Has Overrides

- Scope

The Scope section of the left hand panel organizes the configuration properties by role types; first those that are Service-Wide, followed by various role types within the service. When you select one of these roles, a set of properties whose values are managed by the default role group for the role display. Any additional role groups that apply to the property also appear in this panel and you can modify values for each role group just as you can the default role group.

- Category

The Category section of the left hand panel allows you to limit the displayed properties by category.

4. Edit the property value.

- To facilitate entering some types of values, you can specify not only the value, but also the units that apply to the value. For example, to enter a setting that specifies bytes per second, you can choose to enter the value in bytes (B), KiBs, MiBs, or GiBs—selected from a drop-down menu that appears when you edit the value.
- If the property allows a list of values, click the **+** icon to the right of the edit field to add an additional field. An example of this is the HDFS DataNode Data Directory property, which can have a comma-delimited list of

directories as its value. To remove an item from such a list, click the  icon to the right of the field you want to remove.

Many configuration properties have different values that are configured by multiple role groups.

To edit configuration values for multiple role groups:

- a) Go to the property. For example, the configuration panel for the Heap Dump Directory property displays the DataNode Default Group (a role group), and a link that says ... and 6 others.

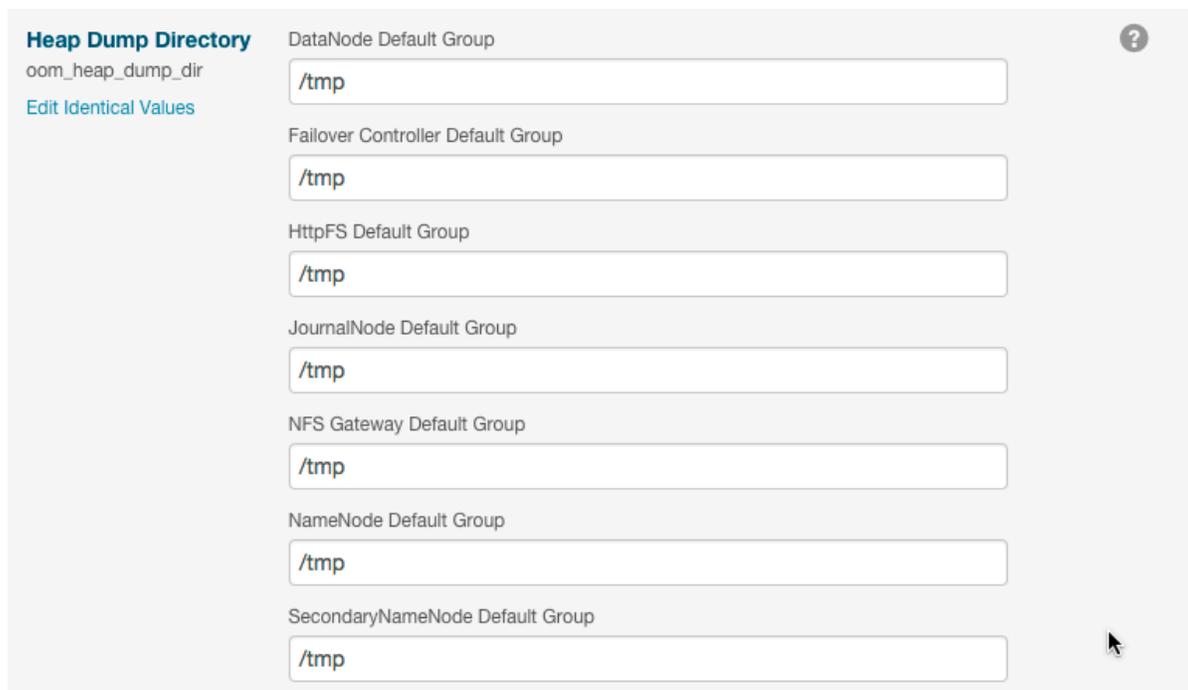


- b) Click the ... and 6 others link to display all of the role groups:



- c) Click the Show fewer link to collapse the list of role groups.

If you edit the single value for this property, Cloudera Manager applies the value to all role groups. To edit the values for one or more of these role groups individually, click Edit Individual Values. Individual fields display where you can edit the values for each role group. For example:



5. Click Save Changes to commit the changes.

You can add a note that is included with the change in the Configuration History. This changes the setting for the role group, and applies to all role instances associated with that role group. Depending on the change you made,

you may need to restart the service or roles associated with the configuration you just changed. Or, you may need to redeploy your client configuration for the service. You should see a message to that effect at the top of the Configuration page, and services will display an outdated configuration  (Restart Needed),  (Refresh Needed), or outdated client configuration  indicator. Click the indicator to display the **Stale Configurations** page.

### Related Information

[Role Groups](#)

[Stale Configurations](#)

## Searching for Properties

You can use the Search box to search for properties by name or label. The search also returns properties whose description matches your search term.

## Validation of Configuration Properties

Cloudera Manager validates the values you specify for configuration properties. If you specify a value that is outside the recommended range of values or is invalid, Cloudera Manager displays a warning at the top of the Configuration tab and in the text box after you click Save Changes. The warning is yellow if the value is outside the recommended range of values and red if the value is invalid.

## Overriding Configuration Properties

### About this task

For role types that allow multiple instances, each role instance inherits its configuration properties from its associated role group. While role groups provide a convenient way to provide alternate configuration properties for selected groups of role instances, there may be situations where you want to make a one-off configuration change—for example when a host has malfunctioned and you want to temporarily reconfigure it. In this case, you can override configuration properties for a specific role instance:

### Procedure

1. Go to the page of the service with the role you want to change. Click Clusters in the left menu and select the service, or click the Status tab on the Cloudera Manager Home page and select the service.
2. Click the Instances tab.
3. Click the role instance you want to change.
4. Click the Configuration tab.
5. Change the configuration values as appropriate.
6. Save your changes.

### What to do next

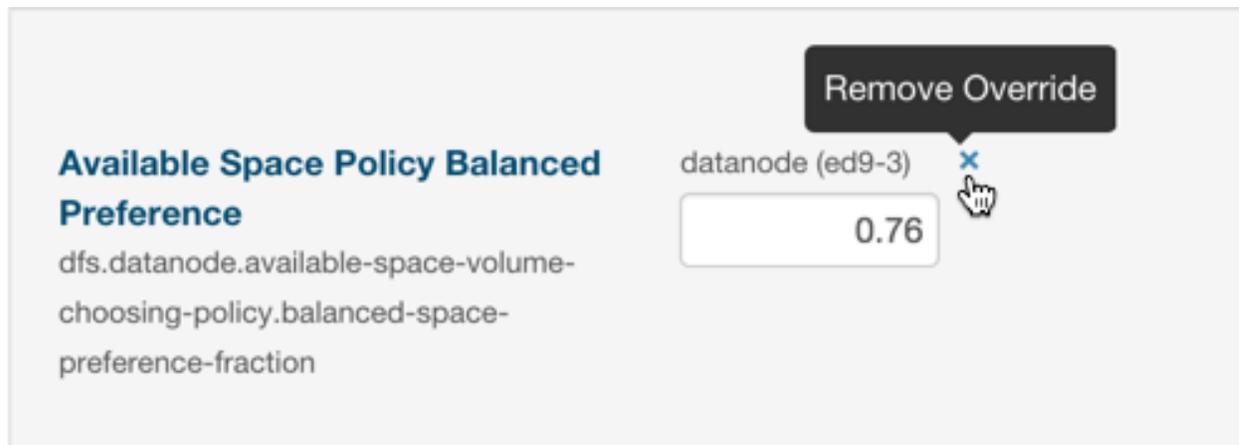
You will most likely need to restart your service or role to have your configuration changes take effect.

### Related Information

[Stale Configurations](#)

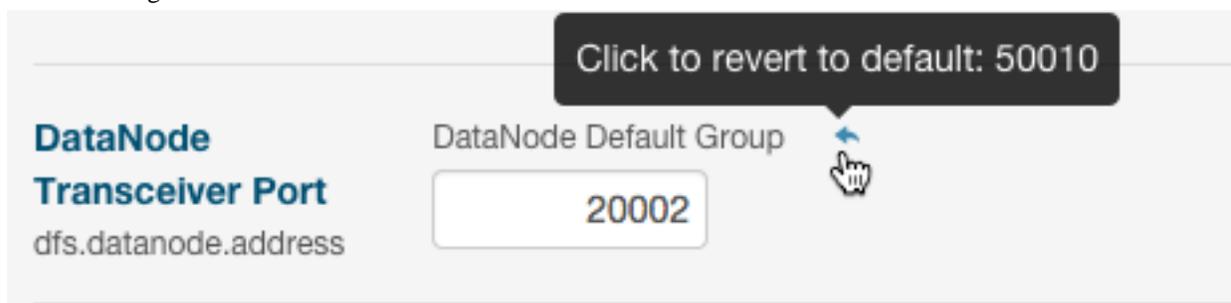
## Viewing and Editing Overridden Configuration Properties

To see a list of all role instances that have an override value for a particular configuration setting, click the Configuration tab on the page for the service and select Status Has overrides . A list of configuration properties where values have been overridden displays. The panel for each configuration property displays the values for each role group or instance. You can edit the value of this property for this instance, or, you can click the  icon next to an instance name to remove the overridden value.



## Resetting Configuration Properties to the Default Value

To reset a property back to its default value, click the  icon. The default value is inserted and the icon turns into an Undo icon (). Explicitly setting a configuration to the same value as its default (inherited value) has the same effect as using the  icon.



There is no mechanism for resetting to an autoconfigured value. However, you can use the configuration history and rollback feature to revert any configuration changes.

### Related Information

[Autoconfiguration](#)

[Viewing and Reverting Configuration Changes](#)

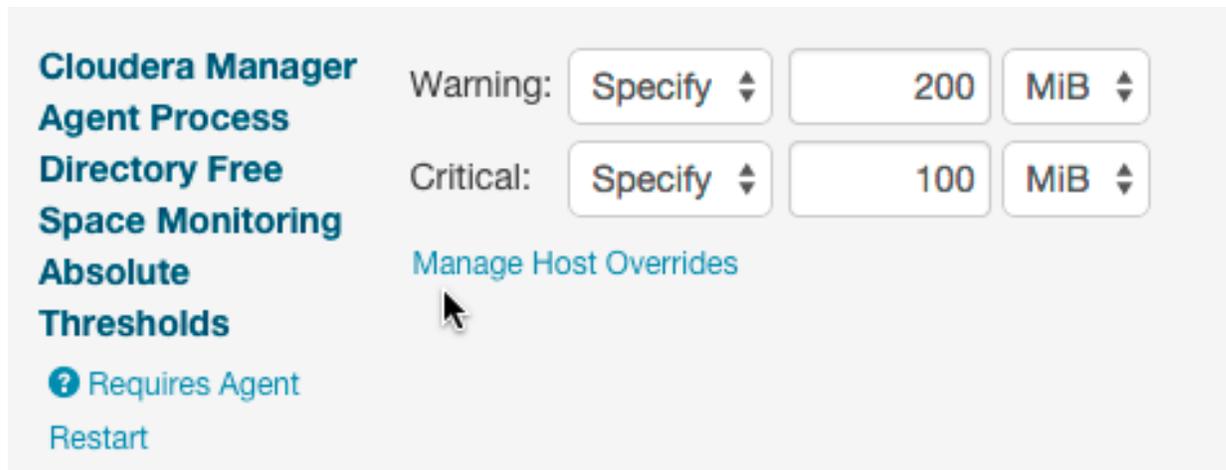
## Viewing and Editing Host Overrides

You can override the properties of individual hosts in your cluster.

### Procedure

1. In the left menu, click HostsConfiguration.
2. Use the Filters or Search box to locate the property that you want to override.

3. Click the Manage Host Overrides link.



**Cloudera Manager**  
**Agent Process**  
**Directory Free**  
**Space Monitoring**  
**Absolute**  
**Thresholds**

Warning: Specify ▾ 200 MiB ▾

Critical: Specify ▾ 100 MiB ▾

[Manage Host Overrides](#)

[? Requires Agent](#)

[Restart](#)

The Manage Overrides dialog box displays.

4. Select one or more hosts to override this property.

5. Click Update.

A new entry area displays where you can enter the override values. In the example below, servers ed9-e.ent.cloudera.com and ed9-r.cloudera.com were selected for overrides. Note that the first set of fields displays the value set for all hosts and the two sets of fields that follow allow you to edit the override values for each specified host.

The screenshot shows the Cloudera Manager configuration page for 'Agent Process Directory Free Space Monitoring Absolute Thresholds'. The page is organized into sections for 'All Hosts' and individual host overrides. For 'All Hosts', the Warning threshold is 200 MiB and the Critical threshold is 100 MiB. Two host overrides are shown: 'Cluster 1 > ed9-3.ent.cloudera.com' and 'Cluster 1 > ed9-4.ent.cloudera.com'. Both host overrides have the same Warning (200 MiB) and Critical (100 MiB) thresholds. A 'Manage Host Overrides' link is located at the bottom of the host override section.

- To remove the override, click the  icon next to the hostname.
  - To apply the same value to all hosts, click Edit Identical Values. Click Edit Individual Values to apply different values to selected hosts.
6. If the property indicates Requires Agent Restart, restart the agent on the affected hosts.

## Restarting Services and Instances after Configuration Changes

If you change the configuration properties after you start a service or instance, you may need to restart the service or instance to have the configuration properties become active.

### About this task

If you change configuration properties at the service level that affect a particular role only (such as all DataNodes but not the NameNodes), you can restart only that role; you do not need to restart the entire service. If you changed the configuration for a particular role instance (such as one of four DataNodes), you may need to restart only that instance.

### Procedure

1. Follow the instructions in the topics *Restarting a Service* or *Starting, Stopping, and Restarting Role Instances*.
2. If you see a Finished status, the service or role instances have restarted.

3. Go to the Home Status tab.

### Results

The service should show a Status of Started for all instances and a health status of Good.

### Related Information

[Restarting a Cloudera Runtime Service](#)

[Starting, Stopping, and Restarting Role Instances](#)

[Stale Configurations](#)

## Suppressing Configuration and Parameter Validation Warnings

You can suppress the warnings that Cloudera Manager issues when a configuration value is outside the recommended range or is invalid. If a warning does not apply to your deployment, you might want to suppress it. Suppressed validation warnings are still retained by Cloudera Manager, and you can unsuppress the warnings at any time. You can suppress each warning when you view it, or you can configure suppression for a specific validation before warnings occur.

### Suppressing a Configuration Validation in Cloudera Manager

When viewing the configuration issues, you can suppress each warning. A dialog box opens where you can enter a comment about the suppression.

#### Procedure

To suppress warnings from the All Configuration Issues screen:

1. Click the Cloudera Manager logo to return to the Home screen.
2. Click Configuration Configuration Issues .
3. Locate the validation message in the list and click the Suppress... link.

A dialog box opens where you can enter a comment about the suppression.

4. Click Confirm.

## Managing Suppressed Validations

### About this task

On pages where you have suppressed validations, you see a link that says Show # Suppressed Warning(s). On this screen, you can:

- Click the Show # Suppressed Warning(s) link to show the warnings.

Each suppressed warning displays an icon:  .

- Click the Unsuppress... link to unsuppress the configuration validation.
- Click the Hide Suppressed Warnings link to re-hide the suppressed warnings.

## Suppressing Configuration Validations Before They Trigger Warnings

### Procedure

1. Go to the service or host with the configuration validation warnings you want to suppress.
2. Click Configuration .

3. In the filters on the left, select `Category Suppressions`.

A list of suppression properties displays. The names of the properties begin with `Suppress Parameter Validation` or `Suppress Configuration Validator`. You can also use the `Search` function to limit the number of properties that display.

4. Select a suppression property to suppress the validation warning.
5. Enter a Reason for change, and then click `Save Changes` to commit the changes.

## Viewing a List of All Suppressed Validations

### About this task

Do one of the following:

- From the Home page or the Status page of a cluster, select `Configuration Suppressed Health and Configuration Issues`.
- From the Status page of a service, select `Configuration Category Suppressions` and select `Status Non-default`.
- From the left menu, select `Hosts``Hosts Configuration`, then `Category Suppressions` and select `Status Non-default`.

## Cluster-Wide Configuration

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

To make configuration changes that apply to an entire cluster, do one of the following to open the configuration page:

- All Clusters
  1. From the Cloudera Manager Home page, click `Configuration` and then select one of the following classes of properties:
    - Advanced Configuration Snippets
    - Database Settings
    - Disk Space Thresholds
    - Local Data Directories and Files
    - Log Directories
    - Navigator Settings
    - Non-default Values - properties whose value differs from the default value
    - Non-uniform Values - properties whose values are not uniform across the cluster or clusters
    - Ports
    - Service Dependencies

You can also select `Configuration Issues` to view a list of configuration issues for all clusters.

- Specific Cluster
  1. From the Home page, click a cluster name, or click `Clusters``Cluster name` from the left menu.
  2. Select `Configuration` and then select one of the classes of properties listed above.

You can also apply the following filters to limit the displayed properties:

- Enter a search term in the `Search` box to search for properties by name or description.
- Expand the `Status` filter to select options that limit the displayed properties to those with errors or warnings, properties that have been edited, properties with non-default values, or properties with overrides. Select `All` to remove any filtering by `Status`.
- Expand the `Scope` filter to display a list of service types. Expand a service type heading to filter on `Service-Wide` configurations for a specific service instance or select one of the default role groups listed under each service type. Select `All` to remove any filtering by `Scope`.

- Expand the Category filter to filter using a sub-grouping of properties. Select All to remove any filtering by Category.

## Custom Configuration

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

Cloudera Manager exposes properties that allow you to insert custom configuration text into XML configuration, property, and text files, or into an environment. The naming convention for these properties is: *XXX* Advanced Configuration Snippet (Safety Valve) for *YYY* or *XXX YYY* Advanced Configuration Snippet (Safety Valve), where *XXX* is a service or role and *YYY* is the target.

The values you enter into a configuration snippet must conform to the syntax of the target. For an XML configuration file, the configuration snippet must contain valid XML property definitions. For a properties file, the configuration snippet must contain valid property definitions. Some files simply require a list of host addresses.

The configuration snippet mechanism is intended for use in cases where there is configuration setting that is not exposed as a configuration property in Cloudera Manager. Configuration snippets generally override normal configuration. Contact Cloudera Support if you are required to use a configuration snippet that is not explicitly documented.

Service-wide configuration snippets apply to all roles in the service; a configuration snippet for a role group applies to all instances of the role associated with that role group.

Server and client configurations have separate configuration snippets. In general after changing a server configuration snippet you must restart the server, and after changing a client configuration snippet you must redeploy the client configuration. Sometimes you can refresh instead of restart. In some cases however, you must restart a dependent server after changing a client configuration. For example, changing a MapReduce client configuration marks the dependent Hive server as stale, which must be restarted. The Admin Console displays an indicator when a server must be restarted. In addition, the All Configuration Issues tab on the **Home** page indicates the actions you must perform to resolve stale configurations.

### Configuration Snippet Types and Syntax

#### Configuration

Set configuration properties in various configuration files; the property name indicates into which configuration file the configuration will be placed. Configuration files have the extension `.xml` or `.conf`.

For example, there are several configuration snippets for the Hive service. One Hive configuration snippet property is called the HiveServer2 Advanced Configuration Snippet for `hive-site.xml`; configurations you enter here are inserted verbatim into the `hive-site.xml` file associated with the HiveServer2 role group.

To see a list of configuration snippets that apply to a specific configuration file, enter the configuration file name in the Search field in the top navigation bar. For example, searching for `mapred-site.xml` shows the configuration snippets that have `mapred-site.xml` in their name.

Some configuration snippet descriptions include the phrase for this role only. These configurations are stored in memory, and only inserted to the configuration when running an application from Cloudera Manager. Otherwise, the configuration changes are added to the configuration file on disk, and are used when running the application both from Cloudera Manager and from the command line.

Syntax:

```
<property>
  <name>property_name</name>
  <value>property_value</value>
```

```
</property>
```

For example, to specify a MySQL connector library, put this property definition in that configuration snippet:

```
<property>
  <name>hive.aux.jars.path</name>
  <value>file:///usr/share/java/mysql-connector-java.jar</value>
</property>
```

## Environment

Specify key-value pairs for a service, role, or client that are inserted into the respective environment.

One example of using an environment configuration snippet is to add a JAR to a classpath. Place JARs in a custom location such as `/opt/myjars` and extend the classpath using the appropriate service environment configuration snippet. The value of a JAR property must conform to the syntax supported by its environment.

Do not place JARs inside locations such as `/opt/cloudera` or `/usr/lib/{hadoop*,hbase*,hive*}` that are managed by Cloudera because they are overwritten at upgrades.

Syntax:

```
key=value
```

For example, to add JDBC connectors to a Hive gateway classpath, add

```
AUX_CLASSPATH=/usr/share/java/mysql-connector-java.jar:\
/usr/share/java/oracle-connector-java.jar
```

or

```
AUX_CLASSPATH=/usr/share/java/*
```

to Gateway Client Advanced Configuration Snippet for `hive-env.sh`.

## Logging

Set log4j properties in a `log4j.properties` file.

Syntax:

```
key1=value1
key2=value2
```

For example:

```
log4j.rootCategory=INFO, console max.log.file.size=200MB
max.log.file.backup.index=10
```

## Metrics

Set properties to configure Hadoop metrics in a `hadoop-metrics.properties` or `hadoop-metrics2.properties` file.

Syntax:

```
key1=value1
key2=value2
```

For example:

```
*.sink.foo.class=org.apache.hadoop.metrics2.sink.FileSink
namenode.sink.foo.filename=/tmp/namenode-metrics.out
secondarynamenode.sink.foo.filename=/tmp/secondarynamenode-metrics.out
```

### Whitelists and blacklists

Specify a list of host addresses that are allowed or disallowed from accessing a service.

Syntax:

```
host1.domain1 host2.domain2
```

### Related Information

[Restarting a Cloudera Runtime Service](#)

[Manually Redeploying Client Configuration Files](#)

[Stale Configurations](#)

[Cloudera Manager Admin Console](#)

[Setting the Class Path](#)

[log4j](#)

## Setting an Advanced Configuration Snippet for a Runtime Component

### About this task

### Procedure

1. Select a runtime component.
2. Click the Configuration tab.
3. In the Search box, type Advanced Configuration Snippet.
4. Choose a property that contains the string Advanced Configuration Snippet (Safety Valve).
5. Specify the snippet properties. If the snippet is an XML file, you have the option to use a snippet editor (the default) or an XML text field:
  - Snippet editor

The screenshot shows a configuration editor interface. It has three input fields: 'Name' with the placeholder text 'Name', 'Value' with the placeholder text 'Value', and 'Description' with the placeholder text 'Description'. Below the 'Description' field is a checkbox labeled 'Final' which is currently unchecked. To the right of the 'Name' field is a trash icon. At the bottom left of the form is a plus sign icon (+).

Click **+** to add a property. Enter the property name, value, and optional description. To indicate that the property value cannot be overridden by another, select the Final checkbox.

- XML text field - Enter the property name, value, and optional description in as XML elements.

```
<property>
  <name>name</name>
```

```
<value>property_value</value>
<final>final_value</final>
</property>
```

To indicate that the property value cannot be overridden, specify `<final>>true</final>`.

To switch between the editor and text field, click the View Editor and View XML links at the top right of the snippet row.

6. Enter a Reason for change, and then click Save Changes to commit the changes.
7. Restart the service or role or redeploy client configurations as indicated.

## Setting an Advanced Configuration Snippet for a Cluster

### Procedure

1. To configure a specific cluster, select a cluster from the HomeStatus page or Clusters *Cluster name* in the left menu. To configure all clusters, start on the Cloudera Manager Home page.
2. Select ConfigurationAdvanced Configuration Snippets.
3. Specify the snippet properties. If the snippet is an XML file, you have the option to use a snippet editor (the default) or an XML text field:

- Snippet editor

Click **+** to add a property. Enter the property name, value, and optional description. To indicate that the property value cannot be overridden by another, select the Final checkbox.

- XML text field - Enter the property name, value, and optional description in as XML elements.

```
<property>
  <name>name</name>
  <value>property_value</value>
  <final>final_value</final>
</property>
```

To indicate that the property value cannot be overridden, specify `<final>>true</final>`.

To switch between the editor and text field, click the View Editor and View XML links at the top right of the snippet row.

4. Enter a Reason for change, and then click Save Changes to commit the changes.
5. Restart the service or role or redeploy client configurations as indicated.

## Stale Configurations

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

The Stale Configurations page provides differential views of changes made in a cluster. For any configuration change, the page contains entries of all affected attributes. For example, the following File entry shows the change to the file `hdfs-site.xml` when you update the property controlling how much disk space is reserved for non-HDFS use on each DataNode:

```
File: hdfs-site.xml hdfs (3) Show
... .. @@ -91,9 +91,9 @@
91 91 <value>4096</value>
92 92 </property>
93 93 <property>
94 94 <name>dfs.datanode.du.reserved</name>
95 95 - <value>5077964390</value>
96 96 + <value>2147483648</value>
97 97 </property>
98 98 <name>dfs.datanode.failed.volumes.tolerated</name>
99 99 <value>0</value>
```

To display the entities affected by a change, click the Show button at the right of the entry. The following dialog box shows that three DataNodes were affected by the disk space change:

### Entities Affected By This Change ×

Changes From: File: hdfs-site.xml

---

**hdfs** 3

- datanode (tcdn48-4)
- datanode (tcdn48-2)
- datanode (tcdn48-3)

Close

## Viewing Stale Configurations

To view stale configurations, click the , , or indicator next to a service on the **Cloudera Manager Admin Console Home** page or on a service status page.

## Attribute Categories

The categories of attributes include:

- Environment - represents environment variables set for the role. For example, the following entry shows the change to the environment that occurs when you update the heap memory configuration of the SecondaryNameNode.

```
Environment hdfs (1) Show
... .. @@ -2,6 +2,6 @@
2 2 HADOOP_AUDIT_LOGGER=INFO,RFAUDIT
3 3 HADOOP_LOGFILE=hadoop-cmf-HDFS-1-SECONDARYNAMENODE-tcdn48-1.ent.cloudera.com.log.out
4 4 HADOOP_LOG_DIR=/var/log/hadoop-hdfs
5 5 HADOOP_ROOT_LOGGER=INFO,RFA
6 6 -HADOOP_SECONDARYNAMENODE_OPTS=-Xmx305135616 -Xmx305135616 -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccupancy
7 7 +HADOOP_SECONDARYNAMENODE_OPTS=-Xmx1073741824 -Xmx1073741824 -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccupancy
7 7 HADOOP_SECURITY_LOGGER=INFO,RFAS
```

- Files - represents configuration files used by the role.
- Process User & Group - represents the user and group for the role. Every role type has a configuration to specify the user/group for the process. If you change a value for a user or group on any service's configuration page it will appear in the Stale Configurations page.

- System Resources - represents system resources allocated for the role, including ports, directories, and cgroup limits. For example, a change to the port of role instance will appear in the System Resources category.
- Client Configs Metadata - represents client configurations.

### Filtering Stale Configurations

You filter the entries on the Stale Configurations page by selecting from one of the drop-down lists:

- Attribute - you can filter by an attribute category such as All Files or by a specific file such as topology.map or yarn-site.xml.
- Service
- Role

After you make a selection, both the page and the drop-down show only entries that match that selection.

To reset the view, click Remove Filter or select All XXX, where XXX is Files, Services, or Roles, from the drop-down. For example, to see all the files, select All Files.

### Stale Configuration Actions

The Stale Configurations page displays action buttons. The action depends on what is required to bring the entire cluster's configuration up to date. If you go to the page by clicking a  (Refresh Needed) indicator, the action button will say Restart Stale Services if one of the roles listed on the page need to be restarted.

- Refresh Stale Services - Refreshes stale services.
- Restart Stale Services - Restarts stale services.
- Deploy Client Configuration - Runs the cluster deploy client configurations action.

### Related Information

[Cloudera Manager Admin Console](#)

[Client Configuration Files](#)

## Client Configuration Files

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

To allow clients to use the HBase, HDFS, Hive, MapReduce, and YARN services, Cloudera Manager creates zip archives of the configuration files containing the service properties. The zip archive is referred to as a *client configuration file*. Each archive contains the set of configuration files needed to access the service: for example, the MapReduce client configuration file contains copies of core-site.xml, hadoop-env.sh, hdfs-site.xml, log4j.properties, and mapred-site.xml.

Client configuration files are generated automatically by Cloudera Manager based on the services and roles you have installed and Cloudera Manager deploys these configurations automatically when you install your cluster, add a service on a host, or add a gateway role on a host. Specifically, for each host that has a service role instance installed, and for each host that is configured as a gateway role for that service, the deploy function downloads the configuration zip file, unzips it into the appropriate configuration directory, and uses the Linux alternatives mechanism to set a given, configurable priority level. If you are installing on a system that happens to have pre-existing alternatives, then it is possible another alternative may have higher priority and will continue to be used. The alternatives priority of the Cloudera Manager client configuration is configurable under the Gateway scope of the Configuration tab for the appropriate service.

You can also manually distribute client configuration files to the clients of a service.

The main circumstance that may require a redeployment of the client configuration files is when you have modified a configuration. In this case you will typically see a message instructing you to redeploy your client configurations. The affected service(s) will also display a  icon. Click the indicator to display the **Stale Configurations** page.

### Related Information

[Gateway Roles](#)

[Alternatives Command](#)

[Stale Configurations](#)

## How Client Configurations are Deployed

Client configuration files are deployed on any host that is a client for a service—that is, that has a role for the service on that host. This includes roles such as DataNodes, TaskTrackers, RegionServers and so on as well as gateway roles for the service.

If roles for multiple services are running on the same host (for example, a DataNode role and a TaskTracker role on the same host) then the client configurations for both roles are deployed on that host, with the alternatives priority determining which configuration takes precedence.

For example, suppose we have six hosts running roles as follows: host H1: HDFS-NameNode; host H2: MR-JobTracker; host H3: HBase-Master; host H4: MR-TaskTracker, HDFS-DataNode, HBase-RegionServer; host H5: MR-Gateway; host H6: HBase-Gateway. Client configuration files will be deployed on these hosts as follows: host H1: hdfs-clientconfig (only); host H2: mapreduce-clientconfig, host H3: hbase-clientconfig; host H4: hdfs-clientconfig, mapreduce-clientconfig, hbase-clientconfig; host H5: mapreduce-clientconfig; host H6: hbase-clientconfig

If the HDFS NameNode and MapReduce JobTracker were on the same host, then that host would have both hdfs-clientconfig and mapreduce-clientconfig installed.

## Downloading Client Configuration Files

### Procedure

1. Go to the Cloudera Manager Admin Console Home page.
2. Click the Options menu (3 vertical dots) to the right of the cluster name and select View Client Configuration URLs.
3. Click a link or save the link URL and download the file using wget or curl.

## Manually Redeploying Client Configuration Files

### About this task

Although Cloudera Manager will deploy client configuration files automatically in many cases, if you have modified the configurations for a service, you may need to redeploy those configuration files.

If your client configurations were deployed automatically, the command described in this section will attempt to redeploy them as appropriate.



**Note:** If you are deploying client configurations on a host that has multiple services installed, some of the same configuration files, though with different configurations, will be installed in the conf directories for each service. Cloudera Manager uses the priority parameter in the alternatives `--install` command to ensure that the correct configuration directory is made active based on the combination of services on that host. The priority order is YARN > MapReduce > HDFS. The priority can be configured under the Gateway sections of the Configuration tab for the appropriate service.

### Procedure

1. On the Home Status tab, click  to the right of the cluster name and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

## Viewing and Reverting Configuration Changes

Whenever you change and save a set of configuration settings for a service or role instance or a host, Cloudera Manager saves a revision of the previous settings and the name of the user who made the changes.

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

You can then view past revisions of the configuration settings, and, if desired, roll back the settings to a previous state.

## Viewing Configuration Changes

### About this task

### Procedure

1. For a service, role, or host, click the Configuration tab.
2. Click the History and Rollback button.  
Prior revisions are shown under Past Revisions
  - By default, or if you click Show All, a list of all revisions is shown. If you are viewing a service or role instance, all service/role group related revisions are shown. If you are viewing a host or all hosts, all host/all hosts related revisions are shown.
  - To list only the configuration revisions that were done in a particular time period, use the Time Range Selector to select a time range. Then, click Show within the Selected Time Range.
3. Click the Details... link.  
The Revision Details dialog box displays.

### Related Information

[Time Range Selector](#)

## Reverting Configuration Changes

### Procedure

1. Select the current or past revision to which to roll back.
2. Click the Details... link. The **Revision Details** dialog box displays.

3. Click the Configuration Values tab.
4. Click the Revert Configuration Changes button. The revert action occurs immediately. You may need to restart the service or the affected roles for the change to take effect.



**Important:** This feature can only be used to revert changes to configuration values.

## Autoconfiguration

Cloudera Manager provides several interactive wizards to automate common workflows:

- Add Service - used when adding a new service
- Import MapReduce - used when migrating from MapReduce to YARN

In some of these wizards, Cloudera Manager uses a set of rules to automatically configure certain settings to best suit the characteristics of the deployment. For example, the number of hosts in the deployment drives the memory requirements for certain monitoring daemons: the more hosts, the more memory is needed. Additionally, wizards that are tasked with creating new roles will use a similar set of rules to determine an ideal host placement for those roles.

### Scope

The following table shows, for each wizard, the scope of entities it affects during autoconfiguration and role-host placement.

Wizard	Autoconfiguration Scope	Role-Host Placement Scope
Import MapReduce	Existing YARN service	N/A

Certain autoconfiguration rules are unscoped, that is, they configure settings belonging to entities that aren't necessarily the entities under the wizard's scope. These exceptions are explicitly listed.

### Related Information

[Role Groups](#)

## Autoconfiguration

Cloudera Manager employs several different rules to drive automatic configuration, with some variation from wizard to wizard. These rules range from the simple to the complex.

### Configuration Scope

One of the points of complexity in autoconfiguration is configuration scope. The configuration hierarchy as it applies to services is as follows: configurations may be modified at the service level (affecting every role in the service), role group level (affecting every role instance in the group), or role level (affecting one role instance). A configuration found in a lower level takes precedence over a configuration found in a higher level.

With the exception of the Static Service Pools, and the Import MapReduce wizard, all Cloudera Manager wizards follow a basic pattern:

1. Every role in scope is moved into its own, new, role group.
2. This role group is the receptacle for the role's "idealized" configuration. Much of this configuration is driven by properties of the role's host, which can vary from role to role.
3. Once autoconfiguration is complete, new role groups with common configurations are merged.
4. The end result is a smaller set of role groups, each with an "idealized" configuration for some subset of the roles in scope. A subset can have any number of roles; perhaps all of them, perhaps just one, and so on.

The Static Service Pools and Import MapReduce wizards configure role groups directly and do not perform any merging.

## Data Directories

Several autoconfiguration rules work with data directories, and there's a common sub-rule used by all such rules to determine, out of all the mountpoints present on a host, which are appropriate for data. The subrule works as follows:

- The initial set of mountpoints for a host includes all those that are disk-backed. Network-backed mountpoints are excluded.
- Mountpoints beginning with /boot, /cdrom, /usr, /tmp, /home, or /dev are excluded.
- Mountpoints beginning with /media are excluded, unless the backing device's name contains /xvd somewhere in it.
- Mountpoints beginning with /var are excluded, unless they are /var or /var/lib.
- The largest mount point (in terms of total space, not available space) is determined.
- Other mountpoints with less than 1% total space of the largest are excluded.
- Mountpoints beginning with /var or equal to / are excluded unless they're the largest mount point.
- Remaining mountpoints are sorted lexicographically and retained for future use.

## Memory

The rules used to autoconfigure memory reservations are perhaps the most complicated rules employed by Cloudera Manager. When configuring memory, Cloudera Manager must take into consideration which roles are likely to enjoy more memory, and must not over commit hosts if at all possible. To that end, it needs to consider each host as an entire unit, partitioning its available RAM into segments, one segment for each role. To make matters worse, some roles have more than one memory segment. For example, a Solr server has two memory segments: a JVM heap used for most memory allocation, and a JVM direct memory pool used for HDFS block caching. Here is the overall flow during memory autoconfiguration:

1. The set of participants includes every host under scope as well as every {role, memory segment} pair on those hosts. Some roles are under scope while others are not.
2. For each {role, segment} pair where the role is under scope, a rule is run to determine four different values for that pair:
  - Minimum memory configuration. Cloudera Manager must satisfy this minimum, possibly over-committing the host if necessary.
  - Minimum memory consumption. Like the above, but possibly scaled to account for inherent overhead. For example, JVM memory values are multiplied by 1.3 to arrive at their consumption value.
  - Ideal memory configuration. If RAM permits, Cloudera Manager will provide the pair with all of this memory.
  - Ideal memory consumption. Like the above, but scaled if necessary.
3. For each {role, segment} pair where the role is not under scope, a rule is run to determine that pair's existing memory consumption. Cloudera Manager will not configure this segment but will take it into consideration by setting the pair's "minimum" and "ideal" to the memory consumption value.
4. For each host, the following steps are taken:
  - a. 20% of the host's available RAM is subtracted and reserved for the OS.
  - b.  $\text{sum}(\text{minimum\_consumption})$  and  $\text{sum}(\text{ideal\_consumption})$  are calculated.
  - c. An "availability ratio" is built by comparing the two sums against the host's available RAM.
    1. If  $\text{RAM} < \text{sum}(\text{minimum})$  ratio = 0
    2. If  $\text{RAM} \geq \text{sum}(\text{ideal})$  ratio = 1
  - d. If the host has more available memory than the total of the ideal memory for all roles assigned to the host, each role is assigned its ideal memory and autoconfiguration is finished.
  - e. Cloudera Manager assigns all available host memory by setting each {role, segment} pair to the same consumption value, except in cases where that value is below the minimum memory or above the ideal memory for that pair. In that case, it is set to the minimum memory or the ideal memory as appropriate. This ensures that pairs with low ideal memory requirements are completely satisfied before pairs with higher ideal memory requirements.

5. The {role, segment} pair is set with the value from the previous step. In the Static Service Pools wizard, the role group is set just once (as opposed to each role).
6. Custom post-configuration rules are run.

Customization rules are applied in steps 2, 3 and 7. In step 2, there's a generic rule for most cases, as well as a series of custom rules for certain {role, segment} pairs. Likewise, there's a generic rule to calculate memory consumption in step 3 as well as some custom consumption functions for certain {role, segment} pairs.

### Step 2 Generic Rule

For every {role, segment} pair where the segment defines a default value, the pair's minimum is set to the segment's minimum value (or 0 if undefined), and the ideal is set to the segment's default value.

### Step 2 Custom Rules

#### HDFS

For the NameNode and Secondary NameNode JVM heaps, the minimum is 50 MB and the ideal is  $\max(4 \text{ GB}, \text{sum\_over\_all}(\text{DataNode mountpoints' available space}) / 0.000008)$ .

#### MapReduce

For the JobTracker JVM heap, the minimum is 50 MB and the ideal is  $\max(1 \text{ GB}, \text{round}((1 \text{ GB} * 2.3717181092 * \ln(\text{number of TaskTrackers in MapReduce service})) - 2.6019933306))$ . If the number of TaskTrackers  $\leq 5$ , the ideal is 1 GB.

For the mapper JVM heaps, the minimum is 1 and the ideal is the number of cores, including hyperthreads, on the TaskTracker host. Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a task's heap).

For the reducer JVM heaps, the minimum is 1 and the ideal is  $(\text{number of cores, including hyperthreads, on the TaskTracker host}) / 2$ . Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a task's heap).

#### HBase

For the memory total allowed for HBase RegionServer JVM heap, the minimum is 50 MB and the ideal is  $\min(31 \text{ GB}, (\text{total RAM on region server host}) * 0.64)$

#### YARN

For the memory total allowed for containers, the minimum is 1 GB and the ideal is  $(\text{total RAM on NodeManager host}) * 0.64$ .

#### Hue

With the exception of the Beeswax Server (only in CDH 4), Hue roles do not have memory limits. Therefore, Cloudera Manager treats them as roles that consume a fixed amount of memory by setting their minimum and ideal consumption values, but not their configuration values. The two consumption values are set to 256 MB.

#### Impala

With the exception of the Impala daemon, Impala roles do not have memory limits. Therefore, Cloudera Manager treats them as roles that consume a fixed amount of memory by setting their minimum/ideal consumption values, but not their configuration values. The two consumption values are set to 150 MB for the Catalog Server and 64 MB for the StateStore.

For the Impala Daemon memory limit, the minimum is 256 MB and the ideal is  $(\text{total RAM on daemon host}) * 0.64$ .

## Solr

For the Solr Server JVM heap, the minimum is 50 MB and the ideal is  $\min(64 \text{ GB}, (\text{total RAM on Solr Server host}) * 0.64) / 2.6$ . For the Solr Server JVM direct memory segment, the minimum is 256 MB and the ideal is  $\min(64 \text{ GB}, (\text{total RAM on Solr Server host}) * 0.64) / 2$ .

## Cloudera Management Service

- Alert Publisher JVM heap - Treated as if it consumed a fixed amount of memory by setting the minimum/ideal consumption values, but not the configuration values. The two consumption values are set to 256 MB.
- Service and Host Monitor JVM heaps - The minimum is 50 MB and the ideal is either 256 MB (10 or fewer managed hosts), 1 GB (100 or fewer managed hosts), or 2 GB (over 100 managed hosts).
- Event Server, Reports Manager, and Navigator Audit Server JVM heaps - The minimum is 50 MB and the ideal is 1 GB.
- Navigator Metadata Server JVM heap - The minimum is 512 MB and the ideal is 2 GB.
- Service and Host Monitor off-heap memory segments - The minimum is either 768 MB (10 or fewer managed hosts), 2 GB (100 or fewer managed hosts), or 6 GB (over 100 managed hosts). The ideal is always twice the minimum.

## Step 2 Generic Rule for Static Service Pools Wizard

For every {role, segment} pair where the segment defines a default value and an autoconfiguration share, the pair's minimum is set to the segment's default value, and the ideal is set to  $\min((\text{segment soft max (if exists) or segment max (if exists) or } 2^{63}-1), (\text{total RAM on role's host} * 0.8 / \text{segment scale factor} * \text{service percentage chosen in wizard} * \text{segment autoconfiguration share}))$ .

Autoconfiguration shares are defined as follows:

- HBase RegionServer JVM heap: 1
- HDFS DataNode JVM heap: 1 in CDH 4, 0.2 in CDH 5
- HDFS DataNode maximum locked memory: 0.8 (CDH 5 only)
- Solr Server JVM heap: 0.5
- Solr Server JVM direct memory: 0.5
- Spark Standalone Worker JVM heap: 1
- Accumulo Tablet Server JVM heap: 1
- Add-on services: any

Roles not mentioned here do not define autoconfiguration shares and thus aren't affected by this rule.

Additionally, there's a generic rule to handle `cgroup.memory_limit_in_bytes`, which is unused by Cloudera services but is available for add-on services. Its behavior varies depending on whether the role in question has segments or not.

## With Segments

The minimum is the  $\min(\text{cgroup.memory\_limit\_in\_bytes\_min (if exists) or } 0, \text{sum\_over\_all}(\text{segment minimum consumption}))$ , and the ideal is the sum of all segment ideal consumptions.

## Without Segments

The minimum is `cgroup.memory_limit_in_bytes_min` (if exists) or 0, and the ideal is  $(\text{total RAM on role's host} * 0.8 * \text{service percentage chosen in wizard})$ .

## Step 3 Custom Rules for Static Service Pools Wizard

### YARN

For the memory total allowed for containers, the minimum is 1 GB and the ideal is  $\min(8 \text{ GB}, (\text{total RAM on Node Manager host}) * 0.8 * \text{service percentage chosen in wizard})$ .

## Impala

For the Impala Daemon memory limit, the minimum is 256 MB and the ideal is  $((\text{total RAM on Daemon host}) * 0.8 * \text{service percentage chosen in wizard})$ .

## MapReduce

- Mapper JVM heaps - the minimum is 1 and the ideal is  $(\text{number of cores, including hyperthreads, on the TaskTracker host} * \text{service percentage chosen in wizard})$ . Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a given task's heap).
- Reducer JVM heaps - the minimum is 1 and the ideal is  $(\text{number of cores, including hyperthreads on the TaskTracker host} * \text{service percentage chosen in wizard}) / 2$ . Memory consumption is scaled by `mapred_child_java_opts_max_heap` (the size of a given task's heap).

## Step 3 Generic Rule

For every {role, segment} pair, the segment's current value is converted into bytes, and then multiplied by the scale factor (1.0 by default, 1.3 for JVM heaps, and freely defined for Custom Service Descriptor services).

## Step 3 Custom Rules

### Impala

For the Impala Daemon, the memory consumption is 0 if YARN Service for Resource Management is set. If the memory limit is defined but not -1, its value is used verbatim. If it's defined but -1, the consumption is equal to the total RAM on the Daemon host. If it is undefined, the consumption is  $(\text{total RAM} * 0.8)$ .

### Solr

For the Solr Server JVM direct memory segment, the consumption is equal to the value verbatim provided `solr.hdfs.blockcache.enable` and `solr.hdfs.blockcache.direct.memory.allocation` are both true. Otherwise, the consumption is 0.

## Step 7 Custom Rules

### HDFS

- NameNode JVM heaps are equalized. For every pair of NameNodes in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.
- JournalNode JVM heaps are equalized. For every pair of JournalNodes in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.
- NameNode and Secondary NameNode JVM heaps are equalized. For every {NameNode, Secondary NameNode} pair in an HDFS service with different heap sizes, the larger heap size is reset to the smaller one.

### HBase

Master JVM heaps are equalized. For every pair of Masters in an HBase service with different heap sizes, the larger heap size is reset to the smaller one.

### Impala

If an Impala service has YARN Service for Resource Management set, every Impala Daemon memory limit is set to the value of  $(\text{yarn.nodemanager.resource.memory-mb} * 1 \text{ GB})$  if there's a YARN NodeManager co-located with the Impala Daemon.

### MapReduce

JobTracker JVM heaps are equalized. For every pair of JobTrackers in an MapReduce service with different heap sizes, the larger heap size is reset to the smaller one.

### Oozie

Oozie Server JVM heaps are equalized. For every pair of Oozie Servers in an Oozie service with different heap sizes, the larger heap size is reset to the smaller one.

### YARN

ResourceManager JVM heaps are equalized. For every pair of ResourceManagers in a YARN service with different heap sizes, the larger heap size is reset to the smaller one.

### ZooKeeper

ZooKeeper Server JVM heaps are equalized. For every pair of servers in a ZooKeeper service with different heap sizes, the larger heap size is reset to the smaller one.

## General Rules

### HBase

- `hbase.replication` - For each HBase service, set to true if there's a Key-Value Store Indexer service in the cluster. This rule is unscoped; it can fire even if the HBase service is not under scope.
- `replication.replicationsource.implementation` - For each HBase service, set to `com.ngdata.sep.impl.SepReplicationSource` if there's a Keystore Indexer service in the cluster. This rule is unscoped; it can fire even if the HBase service is not under scope.

### HDFS

- `dfs.datanode.du.reserved` - For each DataNode, set to  $\min((\text{total space of DataNode host largest mountpoint}) / 10, 10 \text{ GB})$ .
- `dfs.namenode.name.dir` - For each NameNode, set to the first two mountpoints on the NameNode host with `/dfs/nm` appended.
- `dfs.namenode.checkpoint.dir` - For each Secondary NameNode, set to the first mountpoint on the Secondary NameNode host with `/dfs/snn` appended.
- `dfs.datanode.data.dir` - For each DataNode, set to all the mountpoints on the host with `/dfs/dn` appended.
- `dfs.journalnode.edits.dir` - For each JournalNode, set to the first mountpoint on the JournalNode host with `/dfs/jn` appended.
- `dfs.datanode.failed.volumes.tolerated` - For each DataNode, set to  $(\text{number of mountpoints on DataNode host}) / 2$ .
- `dfs.namenode.service.handler.count` and `dfs.namenode.handler.count` - For each NameNode, set to  $\ln(\text{number of DataNodes in this HDFS service}) * 20$ .
- `dfs.datanode.hdfs-blocks-metadata.enabled` - For each HDFS service, set to true if there's an Impala service in the cluster. This rule is unscoped; it can fire even if the HDFS service is not under scope.
- `dfs.client.read.shortcircuit` - For each HDFS service, set to true if there's an Impala service in the cluster. This rule is unscoped; it can fire even if the HDFS service is not under scope.
- `dfs.datanode.data.dir.perm` - For each DataNode, set to 755 if there's an Impala service in the cluster and the cluster isn't Kerberized. This rule is unscoped; it can fire even if the HDFS service is not under scope.
- `fs.trash.interval` - For each HDFS service, set to 1.

### Hue

- WebHDFS dependency - For each Hue service, set to either the first HttpFS role in the cluster, or, if there are none, the first NameNode in the cluster.
- HBase Thrift Server dependency - For each Hue service in a CDH 4.4 or higher cluster, set to the first HBase Thrift Server in the cluster.

## Impala

For each Impala service, set Enable Audit Collection and Enable Lineage Collection to true if there's a Cloudera Management Service with a Navigator Audit Server and Navigator Metadata Server roles. This rule is unscoped; it can fire even if the Impala service is not under scope.

## MapReduce

- `mapred.local.dir` - For each JobTracker, set to the first mountpoint on the JobTracker host with `/mapred/jt` appended.
- `mapred.local.dir` - For each TaskTracker, set to all the mountpoints on the host with `/mapred/local` appended.
- `mapred.reduce.tasks` - For each MapReduce service, set to  $\max(1, \text{sum\_over\_all}(\text{TaskTracker number of reduce tasks (determined via } \text{mapred.tasktracker.reduce.tasks.maximum} \text{ for that TaskTracker, which is configured separately)}) / 2)$ .
- `mapred.job.tracker.handler.count` - For each JobTracker, set to  $\max(10, \ln(\text{number of TaskTrackers in this MapReduce service}) * 20)$ .
- `mapred.submit.replication` - If there's an HDFS service in the cluster, for each MapReduce service, set to  $\max(\min(\text{number of DataNodes in the HDFS service, value of HDFS Replication Factor}), \sqrt{\text{number of DataNodes in the HDFS service}})$ .
- `mapred.tasktracker.instrumentation` - If there's a management service, for each MapReduce service, set to `org.apache.hadoop.mapred.TaskTrackerCmonInst`. This rule is unscoped; it can fire even if the MapReduce service is not under scope.

## YARN

- `yarn.nodemanager.local-dirs` - For each NodeManager, set to all the mountpoints on the NodeManager host with `/yarn/nm` appended.
- `yarn.nodemanager.resource.cpu-vcores` - For each NodeManager, set to the number of cores (including hyperthreads) on the NodeManager host.
- `mapred.reduce.tasks` - For each YARN service, set to  $\max(1, \text{sum\_over\_all}(\text{NodeManager number of cores, including hyperthreads}) / 2)$ .
- `yarn.resourcemanager.nodemangers.heartbeat-interval-ms` - For each NodeManager, set to  $\max(100, 10 * (\text{number of NodeManagers in this YARN service}))$ .
- `yarn.scheduler.maximum-allocation-vcores` - For each ResourceManager, set to  $\max\_over\_all(\text{NodeManager number of vcores (determined via } \text{yarn.nodemanager.resource.cpu-vcores} \text{ for that NodeManager, which is configured separately)})$ .
- `yarn.scheduler.maximum-allocation-mb` - For each ResourceManager, set to  $\max\_over\_all(\text{NodeManager amount of RAM (determined via } \text{yarn.nodemanager.resource.memory-mb} \text{ for that NodeManager, which is configured separately)})$ .
- `mapreduce.client.submit.file.replication` - If there's an HDFS service in the cluster, for each YARN service, set to  $\max(\min(\text{number of DataNodes in the HDFS service, value of HDFS Replication Factor}), \sqrt{\text{number of DataNodes in the HDFS service}})$ .

## All Services

If a service dependency is unset, and a service with the desired type exists in the cluster, set the service dependency to the first such target service. Applies to all service dependencies except YARN Service for Resource Management. Applies only to the Installation and Add Cluster wizards.

## Role-Host Placement

Cloudera Manager employs the same role-host placement rule regardless of wizard. The set of hosts considered depends on the scope. If the scope is a cluster, all hosts in the cluster are included. If a service, all hosts in the service's cluster are included. If the Cloudera Management Service, all hosts in the deployment are included. The rules are as follows:

1. The hosts are sorted from most to least physical RAM. Ties are broken by sorting on hostname (ascending) followed by host identifier (ascending).
2. The overall number of hosts is used to determine which arrangement to use. These arrangements are hard-coded, each dictating for a given "master" role type, what index (or indexes) into the sorted host list in step 1 to use.
3. Master role types are included based on several factors:
  - Is this role type part of the service (or services) under scope?
  - Does the service already have the right number of instances of this role type?
  - Does the cluster's CDH version support this role type?
  - Does the installed license allow for this role type to exist?
4. Master roles are placed on each host using the indexes and the sorted host list. If a host already has a given master role, it is skipped.
5. An HDFS DataNode is placed on every host outside of the arrangement described in step 2, provided HDFS is one of the services under scope.
6. Certain "worker" roles are placed on every host where an HDFS DataNode exists, either because it existed there prior to the wizard, or because it was added in the previous step. The supported worker role types are:
  - MapReduce TaskTrackers
  - YARN NodeManagers
  - HBase RegionServers
  - Impala Daemons
  - Spark Workers
7. Hive gateways are placed on every host, provided a Hive service is under scope and a gateway didn't already exist on a given host.
8. Spark on YARN gateways are placed on every host, provided a Spark on YARN service is under scope and a gateway didn't already exist on a given host.

This rule merely dictates the default placement of roles; you are free to modify it before it is applied by the wizard.