Cloudera Manager

Date of Publish:

https://docs.hortonworks.com/

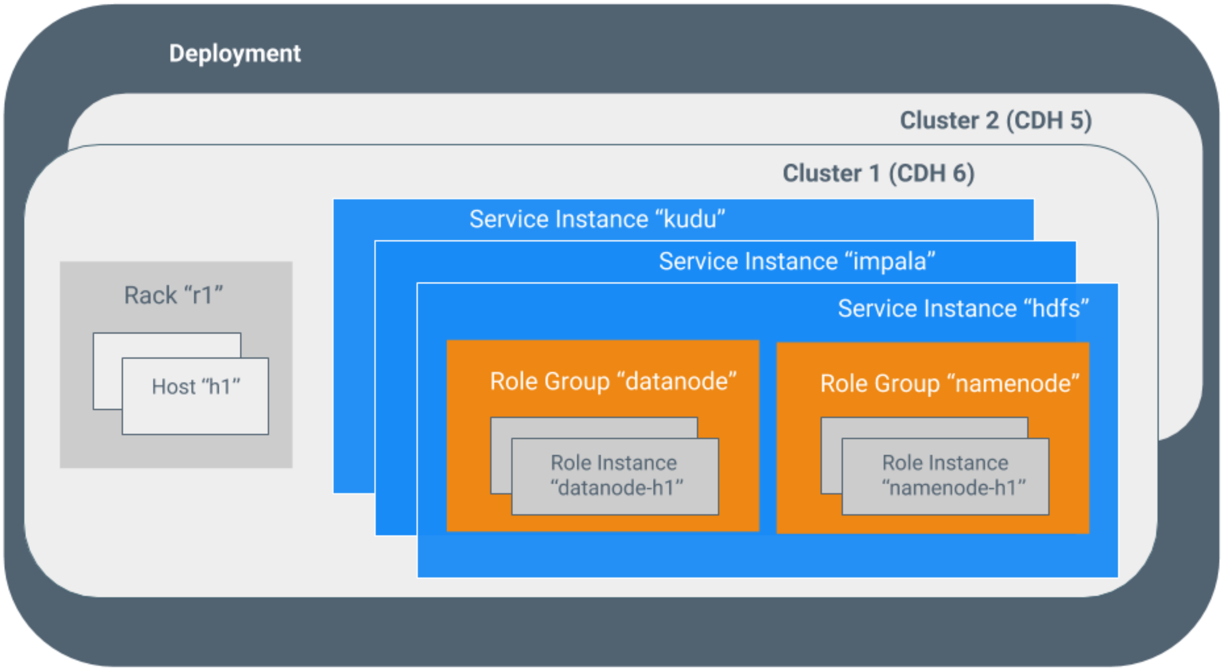
Contents

Cloudera Manager Overview

Cloudera Manager is an end-to-end application for managing CDH clusters. Cloudera Manager sets the standard for enterprise deployment by delivering granular visibility into and control over every part of the CDH cluster—empowering operators to improve performance, enhance quality of service, increase compliance and reduce administrative costs. With Cloudera Manager, you can easily deploy and centrally operate the complete CDH stack and other managed services. The application automates the installation process, reducing deployment time from weeks to minutes; gives you a cluster-wide, real-time view of hosts and services running; provides a single, central console to enact configuration changes across your cluster; and incorporates a full range of reporting and diagnostic tools to help you optimize performance and utilization. This primer introduces the basic concepts, structure, and functions of Cloudera Manager.

Terminology

To effectively use Cloudera Manager, you should first understand its terminology. The relationship between the terms is illustrated below and their definitions follow:



Some of the terms, such as cluster and service, are used without further explanation. Other terms, such as role group, gateway, host template, and parcel are explained in the below sections.

Sometimes the terms *service* and *role* are used to refer to both types and instances, which can be confusing. Cloudera Manager and this section sometimes use the same term for type and instance. For example, the Cloudera Manager Admin Console **Home** > **Status** tab and the **Clusters** > **ClusterName** menu list service instances. This is similar to the practice in programming languages where the term "string" might indicate a type (java.lang.String) or an instance of that type ("hi there"). Where it is necessary to distinguish between types and instances, the word "type" is appended to indicate a type and the word "instance" is appended to explicitly indicate an instance.

deployment

A configuration of Cloudera Manager and all the clusters it manages.

dynamic resource pool

In Cloudera Manager, a named configuration of resources and a policy for scheduling the resources among YARN applications or Impala queries running in the pool.

cluster

- A set of computers or racks of computers that contains an HDFS filesystem and runs MapReduce and other processes on that data. A pseudo-distributed cluster is a CDH installation run on a single machine and useful for demonstrations and individual study.
- In Cloudera Manager, a logical entity that contains a set of hosts, a single version of CDH installed on the hosts, and the service and role instances running on the hosts. A host can belong to only one cluster. Cloudera Manager can manage multiple CDH clusters, however each cluster can only be associated with a single Cloudera Manager Server or Cloudera Manager HA pair.

host

In Cloudera Manager, a physical or virtual machine that runs role instances. A host can belong to only one cluster.

rack

In Cloudera Manager, a physical entity that contains a set of physical hosts typically served by the same switch.

service

- A Linux command that runs a System V init script in /etc/init.d/ in as predictable an environment as possible, removing most environment variables and setting the current working directory to /.
- A category of managed functionality in Cloudera Manager, which may be distributed or not, running in a cluster. Sometimes referred to as a service type. For example: MapReduce, HDFS, YARN, Spark, and Accumulo. In traditional environments, multiple services run on one host; in distributed systems, a service runs on many hosts.

service instance

In Cloudera Manager, an instance of a service running on a cluster. For example: "HDFS-1" and "yarn". A service instance spans many role instances.

role

In Cloudera Manager, a category of functionality within a service. For example, the HDFS service has the following roles: NameNode, SecondaryNameNode, DataNode, and Balancer. Sometimes referred to as a role type. See also user role.

role instance

In Cloudera Manager, an instance of a role running on a host. It typically maps to a Unix process. For example: "NameNode-h1" and "DataNode-h1".

role group

In Cloudera Manager, a set of configuration properties for a set of role instances.

host template

A set of role groups in Cloudera Manager. When a template is applied to a host, a role instance from each role group is created and assigned to that host.

gateway

A type of role that typically provides client access to specific cluster services. For example, HDFS, Hive, Kafka, MapReduce, Solr, and Spark each have gateway roles to provide access for their clients to their respective services. Gateway roles do not always have "gateway" in their names, nor are they exclusively for client access. For example, Hue Kerberos Ticket Renewer is a gateway role that proxies tickets from Kerberos.

The node supporting one or more gateway roles is sometimes referred to as the *gateway node* or *edge node*, with the notion of "edge" common in network or cloud environments. In terms of the Cloudera cluster, the gateway nodes in the cluster receive the appropriate client configuration files when **Deploy Client Configuration** is selected from the Actions menu in Cloudera Manager Admin Console.

parcel

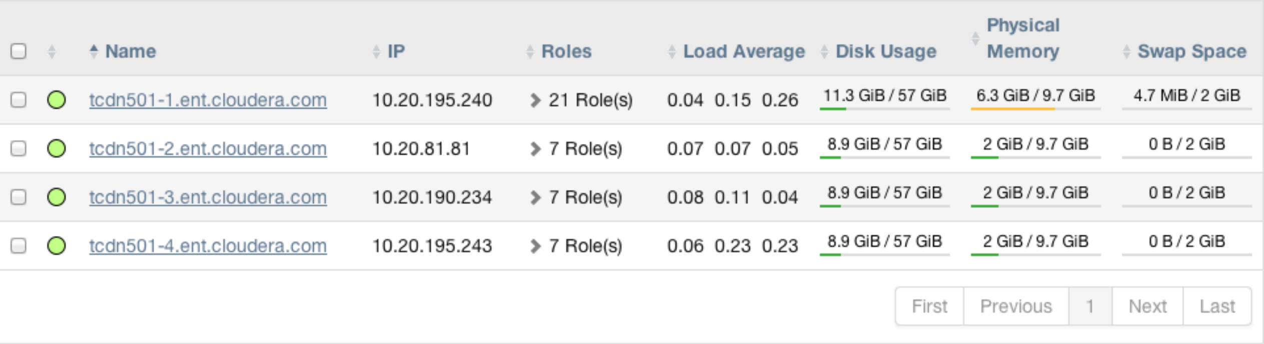
A binary distribution format that contains compiled code and meta-information such as a package description, version, and dependencies.

static service pool

In Cloudera Manager, a static partitioning of total cluster resources—CPU, memory, and I/O weight—across a set of services.

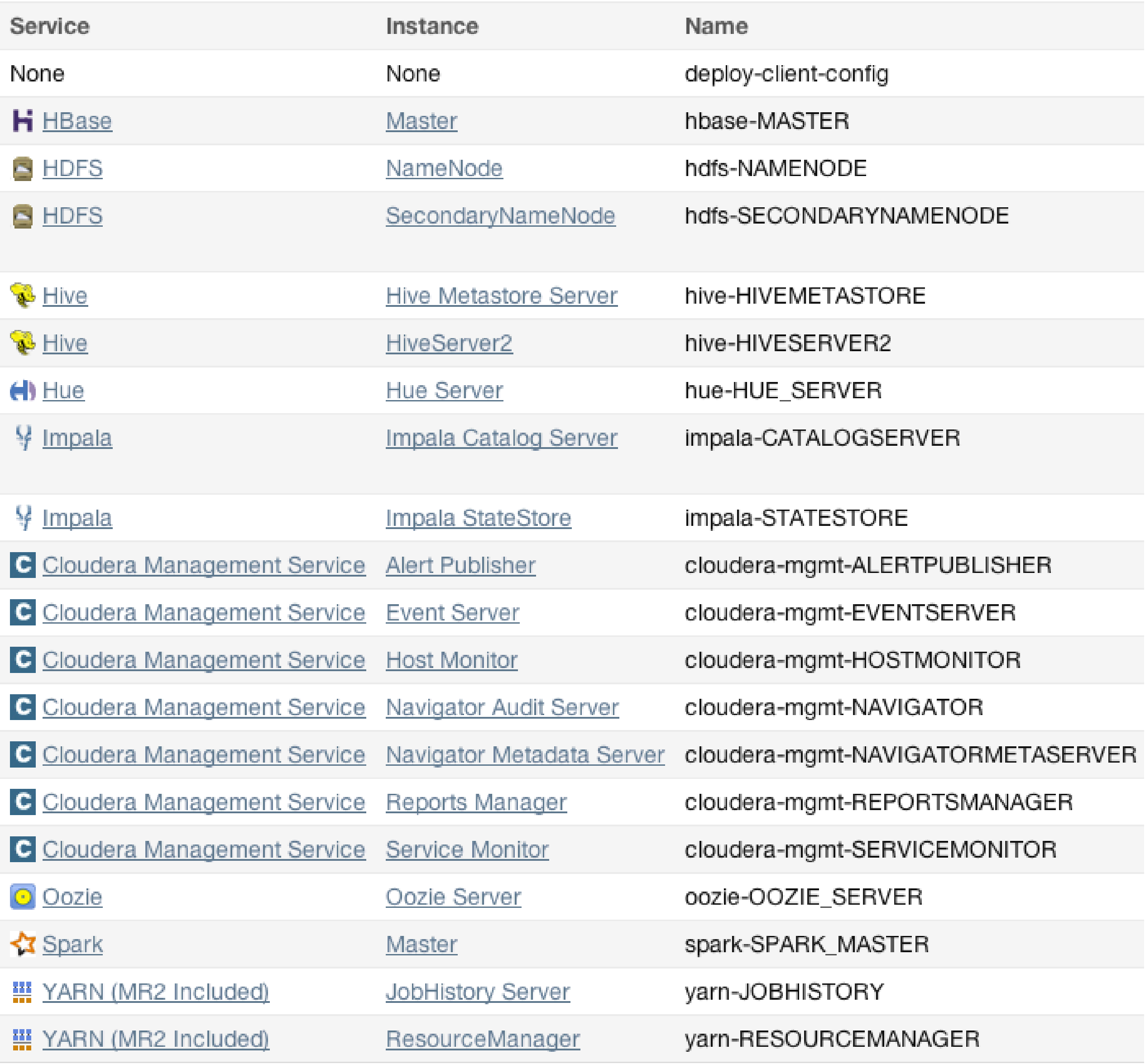
Cluster Example

Consider a cluster Cluster 1 with four hosts as shown in the following listing from Cloudera Manager:



□		Name	IP	Roles	Load Average	Disk Usage	Physical Memory	Swap Space
□	●	tcdn501-1.ent.cloudera.com	10.20.195.240	› 21 Role(s)	0.04 0.15 0.26	11.3 GiB / 57 GiB	6.3 GiB / 9.7 GiB	4.7 MiB / 2 GiB
□	●	tcdn501-2.ent.cloudera.com	10.20.81.81	› 7 Role(s)	0.07 0.07 0.05	8.9 GiB / 57 GiB	2 GiB / 9.7 GiB	0 B / 2 GiB
□	●	tcdn501-3.ent.cloudera.com	10.20.190.234	› 7 Role(s)	0.08 0.11 0.04	8.9 GiB / 57 GiB	2 GiB / 9.7 GiB	0 B / 2 GiB
□	●	tcdn501-4.ent.cloudera.com	10.20.195.243	› 7 Role(s)	0.06 0.23 0.23	8.9 GiB / 57 GiB	2 GiB / 9.7 GiB	0 B / 2 GiB

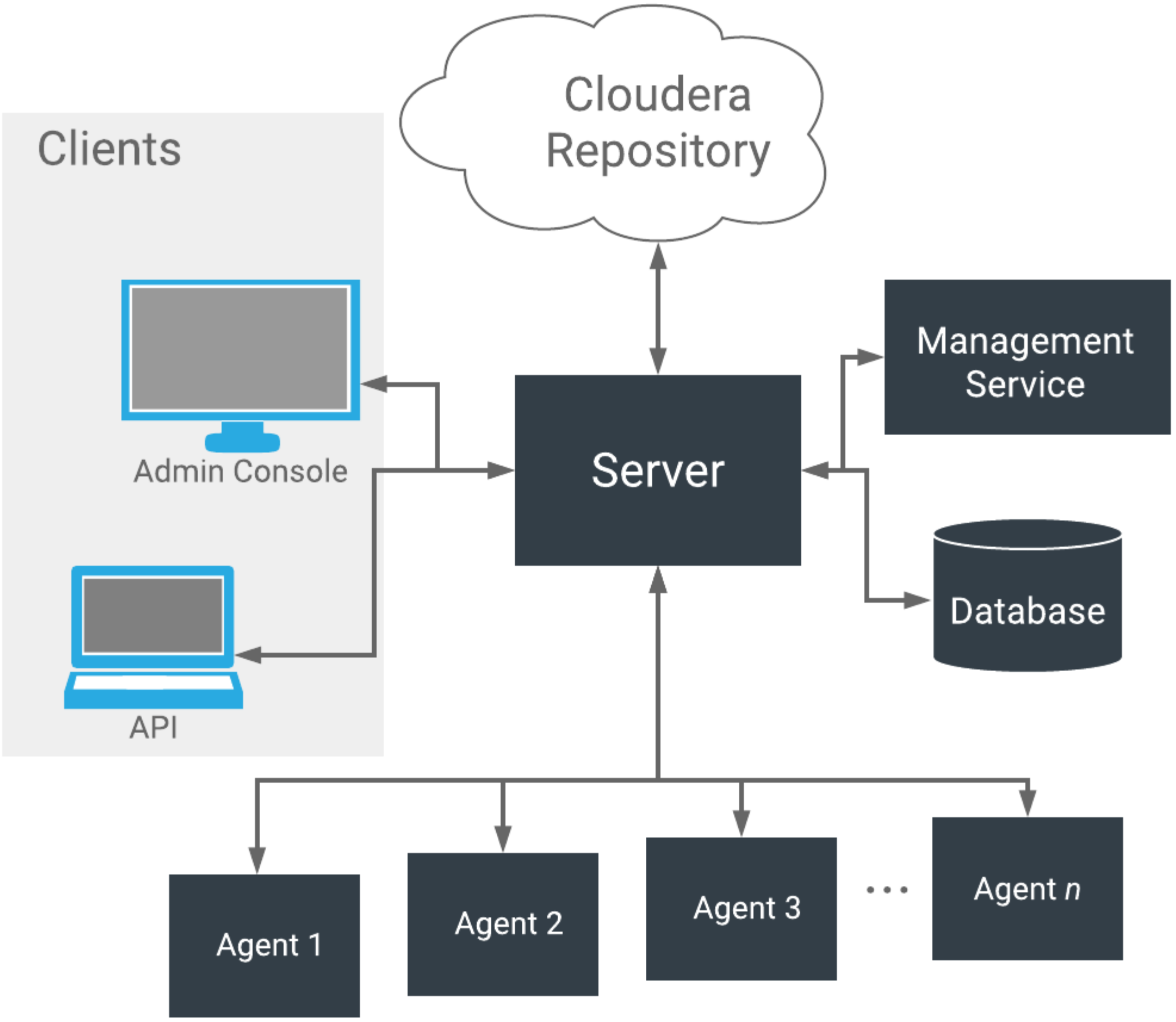
The host tcdn501-1 is the "master" host for the cluster, so it has many more role instances, 21, compared with the 7 role instances running on the other hosts. In addition to the CDH "master" role instances, tcdn501-1 also has Cloudera Management Service roles:



Service	Instance	Name
None	None	deploy-client-config
HBase	Master	hbase-MASTER
HDFS	NameNode	hdfs-NAMENODE
HDFS	SecondaryNameNode	hdfs-SECONDARYNAMENODE
Hive	Hive Metastore Server	hive-HIVEMETASTORE
Hive	HiveServer2	hive-HIVESERVER2
Hue	Hue Server	hue-HUE_SERVER
Impala	Impala Catalog Server	impala-CATALOGSERVER
Impala	Impala StateStore	impala-STATESTORE
Cloudera Management Service	Alert Publisher	cloudera-mgmt-ALERTPUBLISHER
Cloudera Management Service	Event Server	cloudera-mgmt-EVENTSERVER
Cloudera Management Service	Host Monitor	cloudera-mgmt-HOSTMONITOR
Cloudera Management Service	Navigator Audit Server	cloudera-mgmt-NAVIGATOR
Cloudera Management Service	Navigator Metadata Server	cloudera-mgmt-NAVIGATORMETASERVER
Cloudera Management Service	Reports Manager	cloudera-mgmt-REPORTSMANAGER
Cloudera Management Service	Service Monitor	cloudera-mgmt-SERVICEMONITOR
Oozie	Oozie Server	oozie-OOZIE_SERVER
Spark	Master	spark-SPARK_MASTER
YARN (MR2 Included)	JobHistory Server	yarn-JOBHISTORY
YARN (MR2 Included)	ResourceManager	yarn-RESOURCEMANAGER

Architecture

As depicted below, the heart of Cloudera Manager is the Cloudera Manager Server. The Server hosts the Admin Console Web Server and the application logic, and is responsible for installing software, configuring, starting, and stopping services, and managing the cluster on which the services run.



The Cloudera Manager Server works with several other components:

- Agent - installed on every host. The agent is responsible for starting and stopping processes, unpacking configurations, triggering installations, and monitoring the host.
- Management Service - a service consisting of a set of roles that perform various monitoring, alerting, and reporting functions.
- Database - stores configuration and monitoring information. Typically, multiple logical databases run across one or more database servers. For example, the Cloudera Manager Server and the monitoring roles use different logical databases.
- Cloudera Repository - repository of software for distribution by Cloudera Manager.
- Clients - are the interfaces for interacting with the server:
  - Admin Console - Web-based UI with which administrators manage clusters and Cloudera Manager.
  - API - API with which developers create custom Cloudera Manager applications.

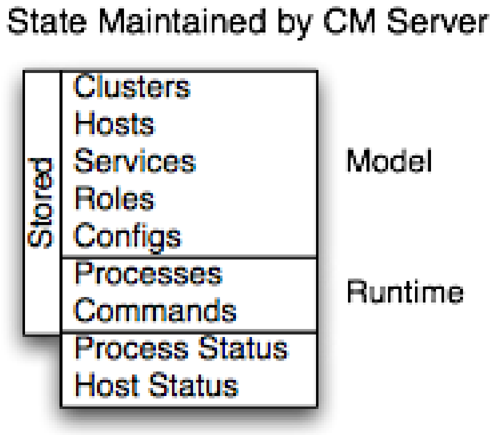
Heartbeating

Heartbeats are a primary communication mechanism in Cloudera Manager. By default Agents send heartbeats every 15 seconds to the Cloudera Manager Server. However, to reduce user latency the frequency is increased when state is changing.

During the heartbeat exchange, the Agent notifies the Cloudera Manager Server of its activities. In turn the Cloudera Manager Server responds with the actions the Agent should be performing. Both the Agent and the Cloudera Manager Server end up doing some reconciliation. For example, if you start a service, the Agent attempts to start the relevant processes; if a process fails to start, the Cloudera Manager Server marks the start command as having failed.

State Management

The Cloudera Manager Server maintains the state of the cluster. This state can be divided into two categories: "model" and "runtime", both of which are stored in the Cloudera Manager Server database.



Cloudera Manager models CDH and managed services: their roles, configurations, and inter-dependencies. Model state captures what is supposed to run where, and with what configurations. For example, model state captures the fact that a cluster contains 17 hosts, each of which is supposed to run a DataNode. You interact with the model through the Cloudera Manager Admin Console configuration screens and API and operations such as "Add Service".

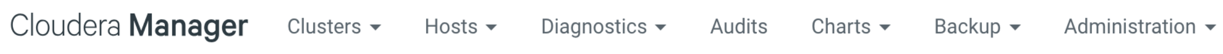
Runtime state is what processes are running where, and what commands (for example, rebalance HDFS or run a Backup/Disaster Recovery schedule or rolling restart or stop) are currently running. The runtime state includes the exact configuration files needed to run a process. When you select Start in the Cloudera Manager Admin Console, the server gathers up all the configuration for the relevant services and roles, validates it, generates the configuration files, and stores them in the database.

When you update a configuration (for example, the Hue Server web port), you have updated the model state. However, if Hue is running while you do this, it is still using the old port. When this kind of mismatch occurs, the role is marked as having an "outdated configuration". To resynchronize, you restart the role (which triggers the configuration re-generation and process restart).

While Cloudera Manager models all of the reasonable configurations, some cases inevitably require special handling. To allow you to workaround, for example, a bug or to explore unsupported options, Cloudera Manager supports an "advanced configuration snippet" mechanism that lets you add properties directly to the configuration files.

Cloudera Manager Admin Console

Cloudera Manager Admin Console is the web-based UI that you use to configure, manage, and monitor CDH.



The Cloudera Manager Admin Console top navigation bar provides the following tabs and menus:

- **Clusters** > **cluster_name**
  - Services - Display individual services, and the Cloudera Management Service. In these pages you can:
    - View the status and other details of a service instance or the role instances associated with the service
    - Make configuration changes to a service instance, a role, or a specific role instance
    - Add and delete a service or role
    - Stop, start, or restart a service or role.
    - View the commands that have been run for a service or a role
    - View an audit event history
    - Deploy and download client configurations
    - Decommission and recommission role instances
    - Enter or exit maintenance mode
    - Perform actions unique to a specific type of service. For example:
      - Enable HDFS high availability or NameNode federation
      - Run the HDFS Balancer
      - Create HBase, Hive, and Sqoop directories
  - **Cloudera Manager Management Service** - Manage and monitor the Cloudera Manager Management Service. This includes the following roles: Activity Monitor, Alert Publisher, Event Server, Host Monitor, Navigator Audit Server, Navigator Metadata Server, Reports Manager, and Service Monitor.
  - **Cloudera Navigator** - Opens the Cloudera Navigator user interface.
  - Hosts - Displays the hosts in the cluster.
  - Reports - Create reports about the HDFS, MapReduce, YARN, and Impala usage and browse HDFS files, and manage quotas for HDFS directories.
  - **Utilization Report** - Opens the **Cluster Utilization Report.** displays aggregated utilization information for YARN and Impala jobs.
  - MapReduce_service_name Jobs - Query information about MapReduce jobs running on your cluster.
  - YARN_service_name Applications - Query information about YARN applications running on your cluster.
  - Impala_service_name Queries - Query information about Impala queries running on your cluster.
  - Dynamic Resource Pools - Manage dynamic allocation of cluster resources to YARN and Impala services by specifying the relative weights of named pools.
  - Static Service Pools - Manage static allocation of cluster resources to HBase, HDFS, Impala, MapReduce, and YARN services.
- Hosts - Display the hosts managed by Cloudera Manager.
  - **All Hosts** - Displays a list of manage hosts in the cluster.
  - **Roles** - Displays the roles deployed on each host.

  - **Host Templates** - Create and manage **Host Templates**, which define sets of role groups that can be used to easily expand a cluster.
  - **Disks Overview** - Displays the status of all disks in the cluster.
  - **Parcels** - Displays parcels available in the cluster and allows you to download, distribute, and activate new parcels.

  In this page you can:

  - View the status and a variety of detail metrics about individual hosts
  - Make configuration changes for host monitoring
  - View all the processes running on a host
  - Run the Host Inspector
  - Add and delete hosts
  - Create and manage host templates
  - Manage parcels
  - Decommission and recommission hosts
  - Make rack assignments
  - Run the host upgrade wizard
- Diagnostics - Review logs, events, and alerts to diagnose problems. The subpages are:
  - Events - Search for and displaying events and alerts that have occurred.
  - Logs - Search logs by service, role, host, and search phrase as well as log level (severity).
  - Server Log -Display the Cloudera Manager Server log.
- Audits - Query and filter audit events across clusters, including logins, across clusters.
- Charts - Query for metrics of interest, display them as charts, and display personalized chart dashboards.
- Backup - Manage replication schedules and snapshot policies.
- Administration - Administer Cloudera Manager. The subpages are:
  - Settings - Configure Cloudera Manager.
  - Alerts - Display when alerts will be generated, configure alert recipients, and send test alert email.
  - Users - Manage Cloudera Manager users and user sessions.
  - Security - Generate Kerberos credentials and inspect hosts.
  - License - Manage Cloudera licenses.
  - Language - Set the language used for the content of activity events, health events, and alert email messages.
  - AWS Credentials - Configure S3 connectivity to Cloudera Manager.
- Parcel Icon

  -
  link to the **Hosts** > **Parcels** page.
- Running Commands Indicator
  -
  .
- Search - Supports searching for services, roles, hosts, configuration properties, and commands. You can enter a partial string and a drop-down list with up to sixteen entities that match will display.
- Support - Displays various support actions. The subcommands are:
  - Send Diagnostic Data - Sends data to Cloudera Support to support troubleshooting.
  - Support Portal () - Displays the Cloudera Support portal.
  - Mailing List () - Displays the Cloudera Manager Users list.

  - Scheduled Diagnostics: Weekly - Configure the frequency of automatically collecting diagnostic data and sending to Cloudera support.
  - The following links open the latest documentation on the Cloudera web site:
    - Help
    - Installation Guide
    - API Documentation
    - Release Notes
  - About - Version number and build details of Cloudera Manager and the current date and time stamp of the Cloudera Manager server.
- Logged-in User Menu - The currently logged-in user. The subcommands are:
  - Change Password - Change the password of the currently logged in user.
  - Logout

Cloudera Manager Admin Console Home Page

When you start the Cloudera Manager Admin Console on page 11, the **Home** > **Status** tab displays. You can also go to the **Home** > **Status** tab by clicking the Cloudera Manager logo in the top navigation bar.

The **Status** tab has two potential views: Table View and Classic View. The Classic View contains a set of charts for the selected cluster, while the Table View separates regular clusters, compute clusters, and other services into summary tables. You can use the **Switch to Table View** and **Switch to Classic View** links on each view to switch between the two views. Cloudera Manager remembers which view you select and remains in that view.

Figure 1: Cloudera Manager Admin Console: Classic View

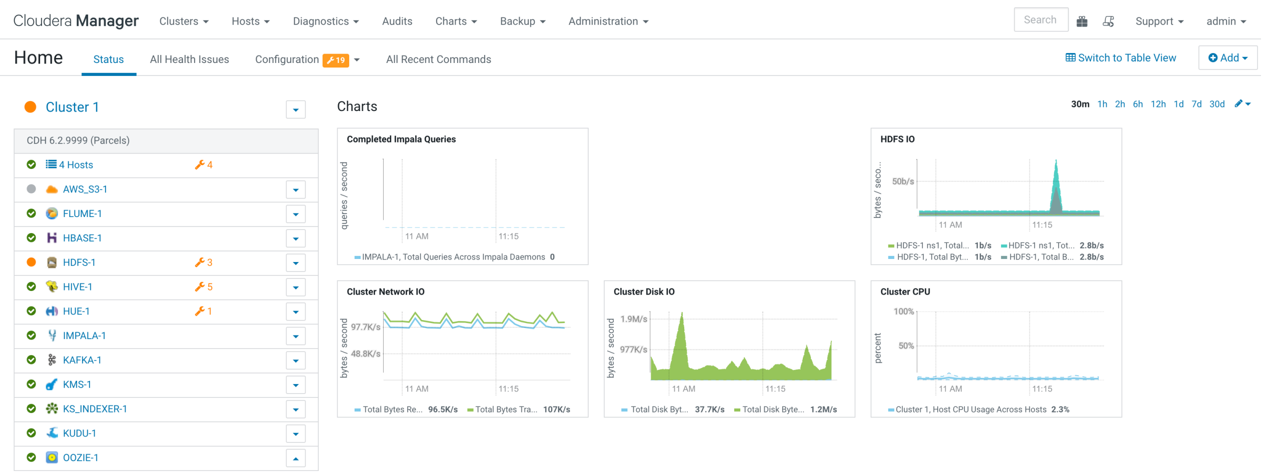
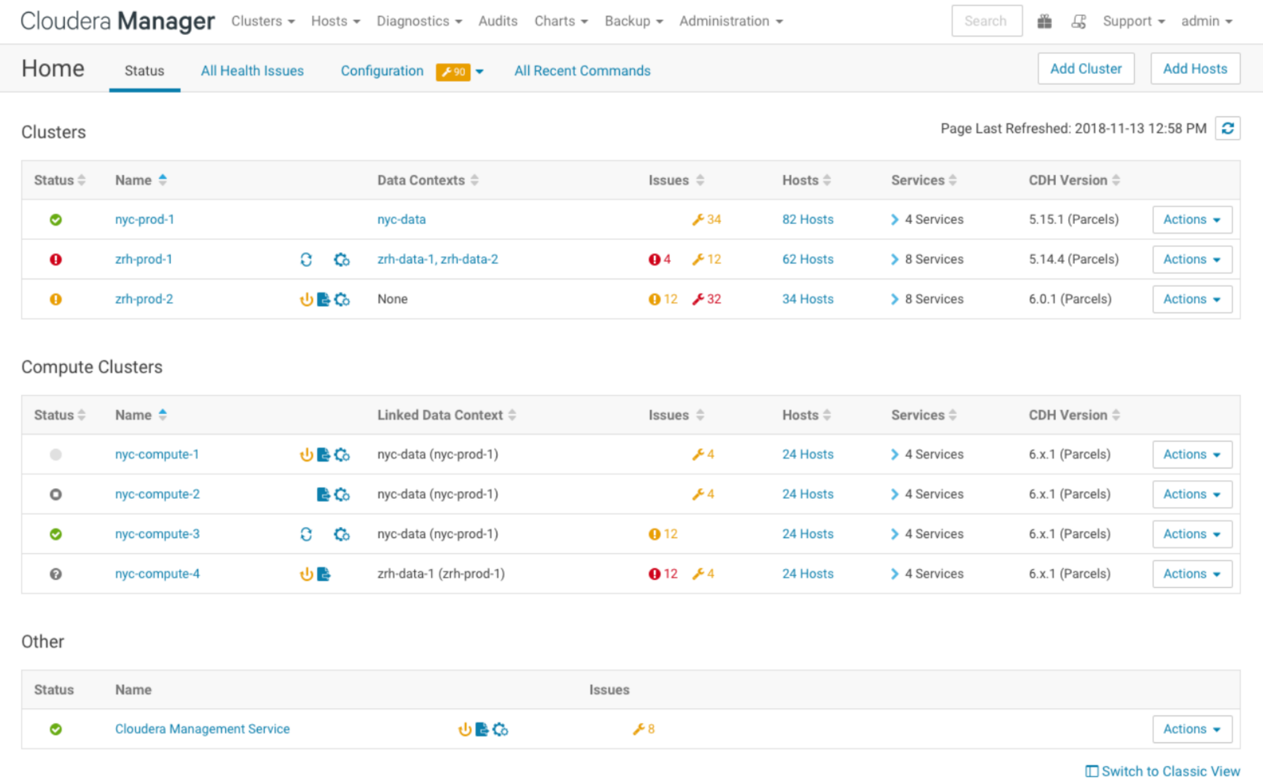


Figure 2: Cloudera Manager Admin Console: Table View



Clusters

Page Last Refreshed: 2018-11-13 12:58 PM

Status	Name	Data Contexts	Issues	Hosts	Services	CDH Version	
✓	nyc-prod-1	nyc-data	🔧34	82 Hosts	› 4 Services	5.15.1 (Parcels)	Actions ▾
❗	zrh-prod-1	zrh-data-1, zrh-data-2	❗4 🔧12	62 Hosts	› 8 Services	5.14.4 (Parcels)	Actions ▾
❗	zrh-prod-2	None	❗12 🔧32	34 Hosts	› 8 Services	6.0.1 (Parcels)	Actions ▾

Compute Clusters

Status	Name	Linked Data Context	Issues	Hosts	Services	CDH Version	
●	nyc-compute-1	nyc-data (nyc-prod-1)	🔧4	24 Hosts	› 4 Services	6.x.1 (Parcels)	Actions ▾
○	nyc-compute-2	nyc-data (nyc-prod-1)	🔧4	24 Hosts	› 4 Services	6.x.1 (Parcels)	Actions ▾
✓	nyc-compute-3	nyc-data (nyc-prod-1)	❗12	24 Hosts	› 4 Services	6.x.1 (Parcels)	Actions ▾
?	nyc-compute-4	zrh-data-1 (zrh-prod-1)	❗12 🔧4	24 Hosts	› 4 Services	6.x.1 (Parcels)	Actions ▾

Other

Status	Name	Issues	
✓	Cloudera Management Service	🔧8	Actions ▾

Switch to Classic View

Status

The Status tab contains:

- Clusters - The clusters being managed by Cloudera Manager. Each cluster is displayed either in summary form or in full form depending on the configuration of the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property. When the number of clusters exceeds the value of the property, only cluster summary information displays.
  - Summary Form - A list of links to cluster status pages. Click Customize to jump to the **Administration** > **Settings** > **Other** > **Maximum Cluster Count Shown In Full** property.
  - Full Form - A separate section for each cluster containing a link to the cluster status page and a table containing links to the Hosts page and the status pages of the services running in the cluster.

    Each service row in the table has a menu of actions that you select by clicking

    ▾

    and can contain one or more of the following indicators:

Indicator	Meaning	Description
❗2	Health issue	Indicates that the service has at least one health issue. The indicator shows the number of health issues at the highest severity level. If there are Bad health test results, the indicator is red. If there are no Bad health test results, but Concerning test results exist, then the indicator is yellow. No indicator is shown if there are no Bad or Concerning health test results. ⚠ **Important:** If there is one Bad health test result and two Concerning health results, there will be three health issues, but the number will be one. Click the indicator to display the **Health Issues** pop-up dialog box. By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show n concerning issue(s)** link.Click the link to display the Status page containing with details about the health test result.

Indicator	Meaning	Description
✕ 4	Configuration issue	Indicates that the service has at least one configuration issue. The indicator shows the number of configuration issues at the highest severity level. If there are configuration errors, the indicator is red. If there are no errors but configuration warnings exist, then the indicator is yellow. No indicator is shown if there are no configuration notifications. ⚠ **Important:** If there is one configuration error and two configuration warnings, there will be three configuration issues, but the number will be one. Click the indicator to display the **Configuration Issues** pop-up dialog box. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show n warning(s)** link.Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.See #unique_12.
⏻ Restart Needed ⟳ Refresh Needed	Configuration modified	Indicates that at least one of a service's roles is running with a configuration that does not match the current configuration settings in Cloudera Manager. Click the indicator to display the Stale Configurations on page 46 page.To bring the cluster up-to-date, click the Refresh or Restart button on the Stale Configurations page or follow the instructions in Refreshing a Cluster, Restarting a Cluster, or #unique_15.
⇨	Client configuration redeployment required	Indicates that the client configuration for a service should be redeployed. Click the indicator to display the Stale Configurations on page 46 page.To bring the cluster up-to-date, click the Deploy Client Configuration button on the Stale Configurations page or follow the instructions in #unique_16.

- Cloudera Management Service - A table containing a link to the Cloudera Manager Service. The Cloudera Manager Service has a menu of actions that you select by clicking ▾ .
- Charts - A set of charts (dashboard) that summarize resource utilization (IO, CPU usage) and processing metrics.

All Health Issues

Displays all health issues by cluster. The number badge has the same semantics as the per service health issues reported on the Status tab.

- By default only Bad health test results are shown in the dialog box. To display Concerning health test results, click the **Also show n concerning issue(s)** link.
- To group the health test results by entity or health test, click the buttons on the **Organize by Entity/ Organize by Health Test** switch.
- Click the link to display the Status page containing with details about the health test result.

All Configuration Issues

Displays all configuration issues by cluster. The number badge has the same semantics as the per service configuration issues reported on the Status tab. By default only notifications at the Error severity level are listed, grouped by service name are shown in the dialog box. To display Warning notifications, click the **Also show n warning(s)** link. Click the message associated with an error or warning to be taken to the configuration property for which the notification has been issued where you can address the issue.

All Recent Commands

Displays all commands run recently across the clusters. A badge

indicates how many recent commands are still running. Click the command link to display details about the command and child commands. See also #unique_18.

Starting and Logging into the Cloudera Manager Admin Console

1. Log into Cloudera Manager Admin Console using the credentials assigned by your administrator. User accounts are assigned roles that constrain the features available to you.

Note: You can configure the Cloudera Manager Admin Console to automatically log out a user after a configurable period of time. See Automatic Logout on page 27.

Displaying the Cloudera Manager Server Version and Server Time

To display the version, build number, and time for the Cloudera Manager Server:

1. Open the Cloudera Manager Admin Console.
2. Select **Support** > **About**.

Process Management

In a non-Cloudera Manager managed cluster, you most likely start a role instance process using an init script, for example, service hadoop-hdfs-datanode start. Cloudera Manager does not use init scripts for the daemons it manages; in a Cloudera Manager managed cluster, starting and stopping services using init scripts will not work.

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.

Stopping the Cloudera Manager Server and the Cloudera Manager Agents will not bring down your services; any running role instances keep running.

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.

These actions reinforce an important point: a Cloudera Manager process never travels alone. In other words, a process is more than just the arguments to exec()—it also includes configuration files, directories that need to be created, and other information.

Host Management

Cloudera Manager provides several features to manage the hosts in your Hadoop clusters. The first time you run Cloudera Manager Admin Console you can search for hosts to add to the cluster and once the hosts are selected you can map the assignment of CDH roles to hosts. Cloudera Manager automatically deploys all software required to participate as a managed host in a cluster: JDK, Cloudera Manager Agent, CDH, Impala, Solr, and so on to the hosts.

Once the services are deployed and running, the Hosts area within the Admin Console shows the overall status of the managed hosts in your cluster. The information provided includes the version of CDH running on the host, the cluster to which the host belongs, and the number of roles running on the host. Cloudera Manager provides operations to manage the lifecycle of the participating hosts and to add and delete hosts. The Cloudera Management Service Host Monitor role performs health tests and collects host metrics to allow you to monitor the health and performance of the hosts.

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.

Resource Management

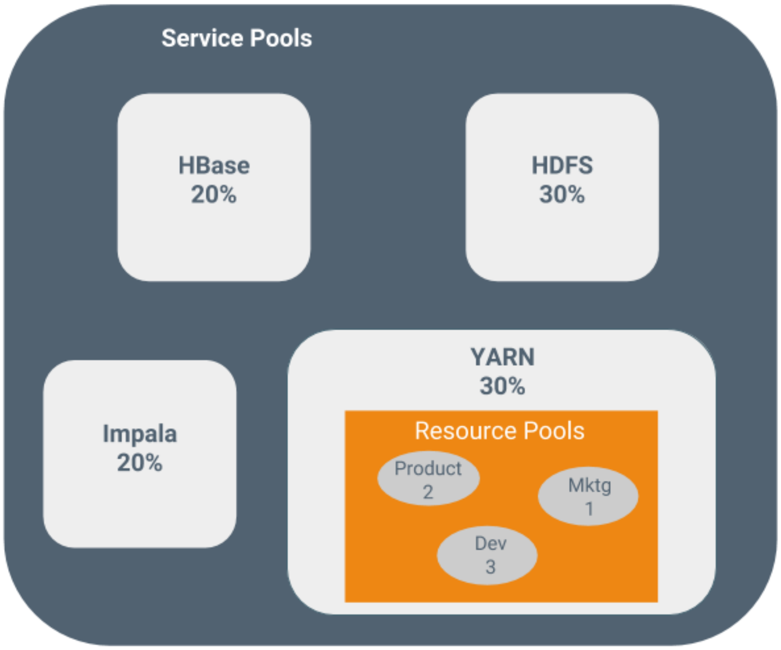
Resource management helps ensure predictable behavior by defining the impact of different services on cluster resources. Use resource management to:

- 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.

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.

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.

User Management

Access to Cloudera Manager features is controlled by user accounts. A user account identifies how a user is authenticated and determines what privileges are granted to the user.

Cloudera Manager provides several mechanisms for authenticating users. You can configure Cloudera Manager to authenticate users against the Cloudera Manager database or against an external authentication service. The external authentication service can be an LDAP server (Active Directory or an OpenLDAP compatible directory), or you can specify another external service. Cloudera Manager also supports using the Security Assertion Markup Language (SAML) to enable single sign-on.

For information about the privileges associated with each of the Cloudera Manager user roles, see #unique_21.

Security Management

Cloudera Manager strives to consolidate security configurations across several projects.

Authentication

Authentication is a process that requires users and services to prove their identity when trying to access a system resource. Organizations typically manage user identity and authentication through various time-tested technologies, including Lightweight Directory Access Protocol (LDAP) for identity, directory, and other services, such as group management, and Kerberos for authentication.

Cloudera clusters support integration with both of these technologies. For example, organizations with existing LDAP directory services such as Active Directory (included in Microsoft Windows Server as part of its suite of Active Directory Services) can leverage the organization's existing user accounts and group listings instead of creating new accounts throughout the cluster. Using an external system such as Active Directory or OpenLDAP is required to support the user role authorization mechanism implemented in Cloudera Navigator.

For authentication, Cloudera supports integration with MIT Kerberos and with Active Directory. Microsoft Active Directory supports Kerberos for authentication in addition to its identity management and directory functionality, that is, LDAP.

These systems are not mutually exclusive. For example, Microsoft Active Directory is an LDAP directory service that also provides Kerberos authentication services, and Kerberos credentials can be stored and managed in an LDAP directory service. Cloudera Manager Server, CDH nodes, and Cloudera Enterprise components, such as Cloudera Navigator, Apache Hive, Hue, and Impala, can all make use of Kerberos authentication.

Authorization

Authorization is concerned with who or what has access or control over a given resource or service. Since Hadoop merges together the capabilities of multiple varied, and previously separate IT systems as an enterprise data hub that stores and works on all data within an organization, it requires multiple authorization controls with varying granularities. In such cases, Hadoop management tools simplify setup and maintenance by:

- Tying all users to groups, which can be specified in existing LDAP or AD directories.
- Providing role-based access control for similar interaction methods, like batch and interactive SQL queries. For example, Apache Sentry permissions apply to Hive (HiveServer2) and Impala.

CDH currently provides the following forms of access control:

- Traditional POSIX-style permissions for directories and files, where each directory and file is assigned a single owner and group. Each assignment has a basic set of permissions available; file permissions are simply read, write, and execute, and directories have an additional permission to determine access to child directories.
- Extended Access Control Lists (ACLs) for HDFS that provide fine-grained control of permissions for HDFS files by allowing you to set different permissions for specific named users or named groups.
- Apache HBase uses ACLs to authorize various operations (READ, WRITE, CREATE, ADMIN) by column, column family, and column family qualifier. HBase ACLs are granted and revoked to both users and groups.
- Role-based access control with Apache Sentry.

Encryption

Data at rest and data in transit encryption function at different technology layers of the cluster:

Cloudera Management Service

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

- Activity Monitor - collects information about activities run by the MapReduce service. This role is not added by default.
- 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 **Home** > **Status** tab, in **Cloudera Management Service** table, click the Cloudera Management Service link.

In addition, for certain editions of the Cloudera Enterprise license, the Cloudera Management Service provides the Navigator Audit Server and Navigator Metadata Server roles for Cloudera Navigator.

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 ❗ Alert 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).

Cloudera Manager API

The Cloudera Manager API provides configuration and service lifecycle management, service health information and metrics, and allows you to configure Cloudera Manager itself. The API is served on the same host and port as the Cloudera Manager Admin Console on page 11, and does not require an extra process or extra configuration. The API supports HTTP Basic Authentication, accepting the same users and credentials as the Cloudera Manager Admin Console.

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

API Documentation Resources

- Quick Start
-
-
- Python Client (deprecated)
- Python Client (Swagger-based)
- Java Client (Swagger-based)
- Java SDK Reference
- Using the Cloudera Manager API for Cluster Automation

Obtaining Configuration Files

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

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

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

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

3. Obtain the content of any particular file:

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

   For example:

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

Retrieving Service and Host Properties

To update a service property using the Cloudera Manager APIs, you'll need to know the name of the property, not just the display name. If you know the property's display name but not the property name itself, retrieve the documentation by requesting any configuration object with the query string view=FULL appended to the URL. For example:

```
http://cm_server_host:7180/api//clusters/Cluster%201/services/service_name/
config?view=FULL
```

Search the results for the display name of the desired property. For example, a search for the display name HDFS Service Environment Advanced Configuration Snippet (Safety Valve) shows that the corresponding property name is hdfs_service_env_safety_valve:

```
{
   "name" : "hdfs_service_env_safety_valve",
   "require" : false,
   "displayName" : "HDFS Service Environment Advanced Configuration Snippet
 (Safety Valve)",
   "description" : "For advanced use onlyu, key/value pairs (one on each
 line) to be inserted into a roles
   environment. Applies to configurations of all roles in this service
 except client configuration.",
   "relatedName" : "",
   "validationState" : "OK"
}
```

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

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

This should return host objects of the form:

```
{
   "hostId" : "2c2e951c-aaf2-4780-a69f-0382181f1821",
   "ipAddress" : "10.30.195.116",
   "hostname" : "cm_server_host",
   "rackId" : "/default",
   "hostUrl" : "http://cm_server_host:7180/cmf/hostRedirect/2c2e951c-
adf2-4780-a69f-0382181f1821",
   "maintenanceMode" : false,
   "maintenanceOwners" : [ ],
   "commissionState" : "COMMISSIONED",
   "numCores" : 4,
   "totalPhysMemBytes" : 10371174400
}
```

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

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

Using the Cloudera Manager API to Obtain Configuration Files

About this task

Procedure

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

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

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

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

3. Obtain the content of any particular file:

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

   For example:

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

Retrieving Service and Host Properties

To update a service property using the Cloudera Manager APIs, you'll need to know the name of the property, not just the display name. If you know the property's display name but not the property name itself, retrieve the documentation by requesting any configuration object with the query string view=FULL appended to the URL. For example:

```
http://cm_server_host:7180/api//clusters/Cluster%201/services/service_name/
config?view=FULL
```

Search the results for the display name of the desired property. For example, a search for the display name HDFS Service Environment Advanced Configuration Snippet (Safety Valve) shows that the corresponding property name is hdfs_service_env_safety_valve:

```
{
   "name" : "hdfs_service_env_safety_valve",
   "require" : false,
   "displayName" : "HDFS Service Environment Advanced Configuration Snippet
 (Safety Valve)",
   "description" : "For advanced use onlyu, key/value pairs (one on each
 line) to be inserted into a roles
   environment. Applies to configurations of all roles in this service
 except client configuration.",
   "relatedName" : "",
   "validationState" : "OK"
}
```

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

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

This should return host objects of the form:

```
{
   "hostId" : "2c2e951c-aaf2-4780-a69f-0382181f1821",
```

```
    "ipAddress" : "10.30.195.116",
    "hostname" : "cm_server_host",
    "rackId" : "/default",
    "hostUrl" : "http://cm_server_host:7180/cmf/hostRedirect/2c2e951c-
adf2-4780-a69f-0382181f1821",
    "maintenanceMode" : false,
    "maintenanceOwners" : [ ],
    "commissionState" : "COMMISSIONED",
    "numCores" : 4,
    "totalPhysMemBytes" : 10371174400
}
```

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

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

Backing Up and Restoring the Cloudera Manager Configuration

About this task

You can use the Cloudera Manager REST API to export and import all of its configuration data. The API exports a JSON document that contains configuration data for the Cloudera Manager instance. You can use this JSON document to back up and restore a Cloudera Manager deployment.

Procedure

Exporting the Cloudera Manager Configuration

1. Export the Cloudera Manager configuration:

   a) Log in to the Cloudera Manager server host as the root user.

   b) Run the following command:

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

   Where:

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

2. Redact Sensitive information from the Exported Configuration

   The exported configuration may contain passwords and other sensitive information. You can configure redaction of the sensitive items by specifying a JVM parameter for Cloudera Manager. When you set this parameter, API calls to Cloudera Manager for configuration data do not include the sensitive information.

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

   To configure redaction for the API:

   a) Log in the Cloudera Manager server host

b) Edit the /etc/default/cloudera-scm-server file by adding the following property (separate each property with a space) to the line that begins with export CMF_JAVA_OPTS.

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

For example:

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

c) Restart Cloudera Manager:

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

3. Restore the Cloudera Manager Configuration

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

   a) Using the Cloudera Manager Administration Console, stop all running services in your cluster:

      1. On the **Home** > **Status** 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.

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

   b) Log in to the Cloudera Manager server host as the root user.

   c) Run the following command:

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

   Where:

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

   d) Restart the Cloudera Manager Server.

RHEL 7, SLES 12, Debian 8, Ubuntu 16.04 and higher	`sudo systemctl restart cloudera-scm-server`
RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04	`sudo service cloudera-scm-server restart`

How To

Logging in to the Cloudera Manager Admin Console

How to log in to the Cloudera Manager Admin Console.

About this task

Before you begin

You must have network access to the Cloudera Manager server host.

Procedure

1. In a web browser, go to http://<server_host>:7180, where <server_host> is the FQDN or IP address of the host where the Cloudera Manager Server is running.
2. Log into Cloudera Manager Admin Console using the credentials assigned by your administrator. User accounts are assigned roles that constrain the features available to you.

Results

The Cloudera Manager Admin Console opens.

What to do next

Note: You can configure the Cloudera Manager Admin Console to automatically log out a user after a configurable period of time. See Automatic Logout on page 27.

Automatic Logout

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

To configure the timeout period:

1. Click **Administration** > **Settings**.
2. Click **Category** > **Security**.
3. Edit the **Session Timeout** property.

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:

Displaying Cloudera Manager Documentation

Procedure

1. Open the Cloudera Manager Admin Console
2. Select **Support** > **Help, Installation Guide, API Documentation, or Release Notes**. By default, the Help and Installation Guide files from the Cloudera web site are opened. This is because local help

files are not updated after installation. You can configure Cloudera Manager to open either the latest Help and Installation Guide from the Cloudera web site (this option requires Internet access from the browser) or locally-installed Help and Installation Guide by configuring the **Administration** > **Settings** > **Support** > **Open latest Help files from the Cloudera website** property.

Starting, Stopping, and Restarting the Cloudera Manager Server

To start the Cloudera Manager Server:

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

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

To stop the Cloudera Manager Server:

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

To restart the Cloudera Manager Server:

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

Managing Cloudera Manager Server Logs

About this task

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

Viewing the Cloudera Manager Server Logs

About this task

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. Select **Diagnostics** > **Logs** on the top navigation bar.
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

For more information about the Logs page, see #unique_50.

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

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
   ```

Managing the Cloudera Manager Agent Logs

About this task

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

About this task

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

Procedure

1. Select **Diagnostics** > **Logs** on the top navigation bar.
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

About this task

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
   ```

Configuring Cloudera Manager

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. See #unique_55/unique_55_Connect_42_section_fvb_4rq_wm.
- Advanced - Enable API debugging and other advanced options.
- Monitoring - Set Agent health status parameters. For configuration instructions, see #unique_55.
- Security - Set TLS encryption settings to enable TLS encryption between the Cloudera Manager Server, Agents, and clients. For configuration instructions, see #unique_58. You can also:
  - Set the realm for Kerberos security and point to a custom keytab retrieval script. For configuration instructions, see #unique_59.
  - Specify session timeout and a "Remember Me" option.
- Ports and Addresses - Set ports for the Cloudera Manager Admin Console and Server. For configuration instructions, see Configuring Cloudera Manager Server Ports.
- 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 files.
- External Authentication - Specify the configuration to use LDAP, Active Directory, or an external program for authentication. See #unique_32 for instructions.
- 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. See #unique_65.
- Custom Service Descriptors - Configure custom service descriptor properties for #unique_30.

You can also configure the following:

- Alerts

  See #unique_66.
- Users

  See Cloudera Manager User Accounts.
- Kerberos

  See #unique_67.
- License

  See #unique_68.
- 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 #unique_69.

Cluster Configuration Overview

When Cloudera Manager configures a service, it allocates *roles* that are required for that service to the hosts in your cluster. The role determines which service daemons run on a host.

For example, for an HDFS service instance, Cloudera Manager configures:

- One host to run the NameNode role.
- One host to run as the secondary NameNode role.
- One host to run the Balancer role.
- Remaining hosts as to run DataNode roles.

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.

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.

Cloudera Manager wizards autoconfigure role group properties based on the resources available on the hosts. For properties that are not dependent on host resources, Cloudera Manager default values typically align with CDH default values for that configuration. Cloudera Manager deviates when the CDH default is not a recommended configuration or when the default values are illegal. For the complete catalog of properties and their default values, see #unique_72.

Server and Client Configuration

Administrators are sometimes surprised that modifying /etc/hadoop/conf and then restarting HDFS has no effect. That is because service instances started by Cloudera Manager do not read configurations from the default locations. To use HDFS as an example, when not managed by Cloudera Manager, there would usually be one HDFS configuration per host, located at /etc/hadoop/conf/hdfs-site.xml. Server-side daemons and clients running on the same host would all use that same configuration.

Cloudera Manager distinguishes between server and client configuration. In the case of HDFS, the file /etc/hadoop/conf/hdfs-site.xml contains only configuration relevant to an HDFS client. That is, by default, if you run a program that needs to communicate with Hadoop, it will get the addresses of the NameNode and JobTracker, and other important configurations, from that directory. A similar approach is taken for /etc/hbase/conf and /etc/hive/conf.

In contrast, the HDFS role instances (for example, NameNode and DataNode) obtain their configurations from a private per-process directory, under /var/run/cloudera-scm-agent/process/unique-process-name. Giving each process its own private execution and configuration environment allows Cloudera Manager to control each process independently. For example, here are the contents of an example 879-hdfs-NAMENODE process directory:

```
$ tree -a /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
  /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
  ### cloudera_manager_Agent_fencer.py
  ### cloudera_manager_Agent_fencer_secret_key.txt
  ### cloudera-monitor.properties
  ### core-site.xml
  ### dfs_hosts_allow.txt
  ### dfs_hosts_exclude.txt
  ### event-filter-rules.json
  ### hadoop-metrics2.properties
  ### hdfs.keytab
  ### hdfs-site.xml
  ### log4j.properties
  ### logs
  #   ### stderr.log
  #   ### stdout.log
  ### topology.map
  ### topology.py
```

Distinguishing between server and client configuration provides several advantages:

- Sensitive information in the server-side configuration, such as the password for the Hive Metastore RDBMS, is not exposed to the clients.
- A service that depends on another service may deploy with customized configuration. For example, to get good HDFS read performance, Impala needs a specialized version of the HDFS client configuration, which may be harmful to a generic client. This is achieved by separating the HDFS configuration for the Impala daemons (stored in the per-process directory mentioned above) from that of the generic client (/etc/hadoop/conf).
- Client configuration files are much smaller and more readable. This also avoids confusing non-administrator Hadoop users with irrelevant server-side properties.

Server and Client Configuration

Administrators are sometimes surprised that modifying /etc/hadoop/conf and then restarting HDFS has no effect. That is because service instances started by Cloudera Manager do not read configurations from the default locations. To use HDFS as an example, when not managed by Cloudera Manager, there would usually be one HDFS con+figuration per host, located at /etc/hadoop/conf/hdfs-site.xml. Server-side daemons and clients running on the same host would all use that same configuration.

Cloudera Manager distinguishes between server and client configuration. In the case of HDFS, the file /etc/hadoop/conf/hdfs-site.xml contains only configuration relevant to an HDFS client. That is, by default, if you run a program that needs to communicate with Hadoop, it will get the addresses of the NameNode and JobTracker, and other important configurations, from that directory. A similar approach is taken for /etc/hbase/conf and /etc/hive/conf.

In contrast, the HDFS role instances (for example, NameNode and DataNode) obtain their configurations from a private per-process directory, under /var/run/cloudera-scm-agent/process/unique-process-name. Giving each process its own private execution and configuration environment allows Cloudera Manager to control each process independently. For example, here are the contents of an example 879-hdfs-NAMENODE process directory:

```
$ tree -a /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
  /var/run/cloudera-scm-Agent/process/879-hdfs-NAMENODE/
  ### cloudera_manager_Agent_fencer.py
  ### cloudera_manager_Agent_fencer_secret_key.txt
  ### cloudera-monitor.properties
  ### core-site.xml
  ### dfs_hosts_allow.txt
  ### dfs_hosts_exclude.txt
  ### event-filter-rules.json
  ### hadoop-metrics2.properties
  ### hdfs.keytab
  ### hdfs-site.xml
  ### log4j.properties
  ### logs
  #   ### stderr.log
  #   ### stdout.log
  ### topology.map
  ### topology.py
```

Distinguishing between server and client configuration provides several advantages:

- Sensitive information in the server-side configuration, such as the password for the Hive Metastore RDBMS, is not exposed to the clients.
- A service that depends on another service may deploy with customized configuration. For example, to get good HDFS read performance, Impala needs a specialized version of the HDFS client configuration, which may be harmful to a generic client. This is achieved by separating the HDFS configuration for the Impala daemons (stored in the per-process directory mentioned above) from that of the generic client (/etc/hadoop/conf).
- Client configuration files are much smaller and more readable. This also avoids confusing non-administrator Hadoop users with irrelevant server-side properties.

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. See Role Groups.

Changing the Configuration of a Service or Role Instance

Procedure

1. Go to the service status page. (**Cluster** > **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

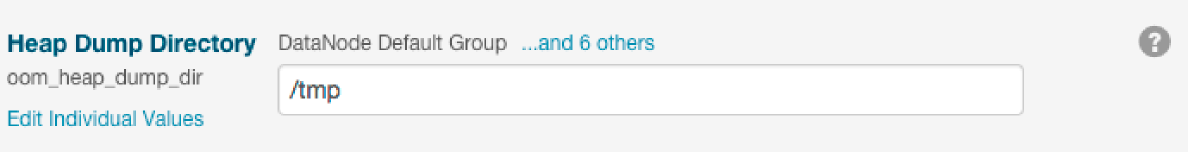
     ➖

     icon to the right of the field you want to remove.

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



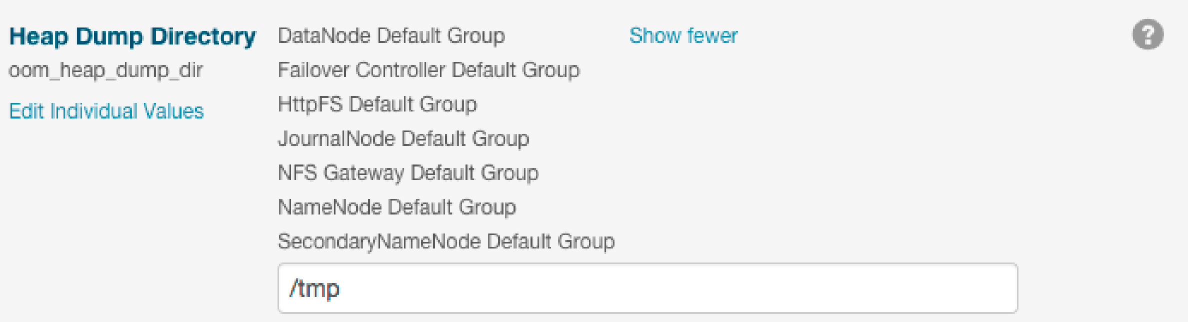
Heap Dump Directory DataNode Default Group ...and 6 others

oom_heap_dump_dir

Edit Individual Values

/tmp

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



Heap Dump Directory DataNode Default Group Show fewer

oom_heap_dump_dir

Edit Individual Values

Failover Controller Default Group

HttpFS Default Group

JournalNode Default Group

NFS Gateway Default Group

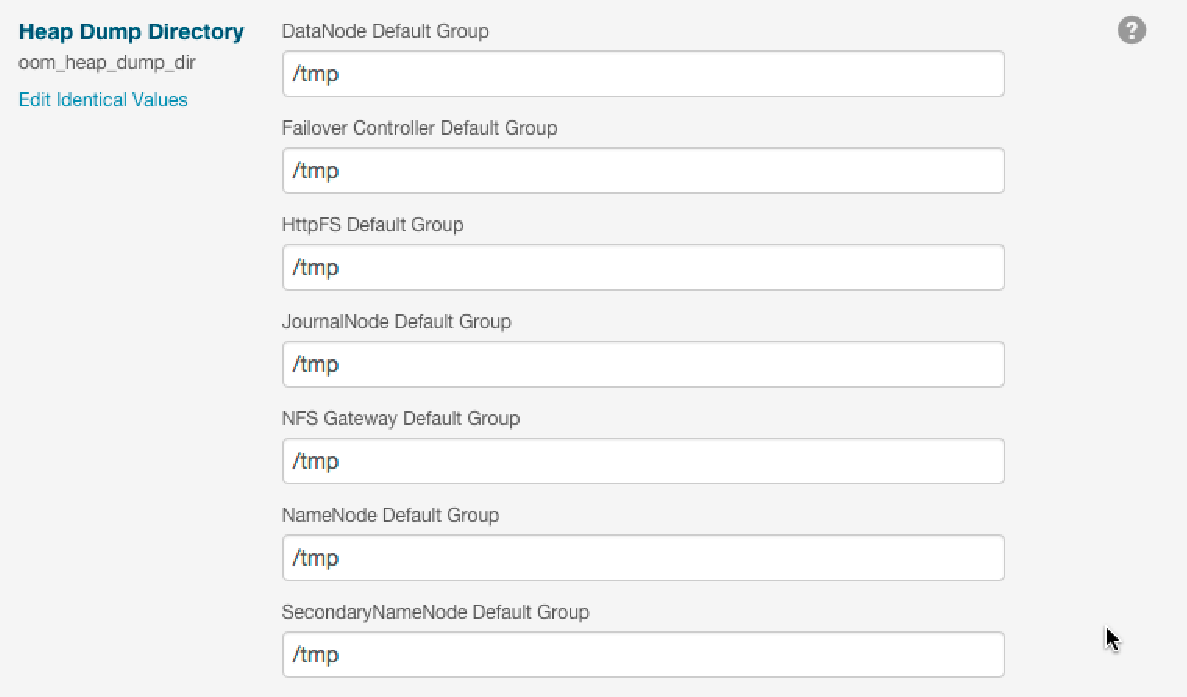
NameNode Default Group

SecondaryNameNode Default Group

/tmp

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:



Heap Dump Directory
oom_heap_dump_dir
Edit Identical Values

DataNode Default Group: /tmp

Failover Controller Default Group: /tmp

HttpFS Default Group: /tmp

JournalNode Default Group: /tmp

NFS Gateway Default Group: /tmp

NameNode Default Group: /tmp

SecondaryNameNode Default Group: /tmp

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), ↻ (Re Needed), or outdated client configuration 🗎 indicator. Click the indicator to display the Stale Configurations on page 46 page.

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 **Status** page for the service whose role you want to change.
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

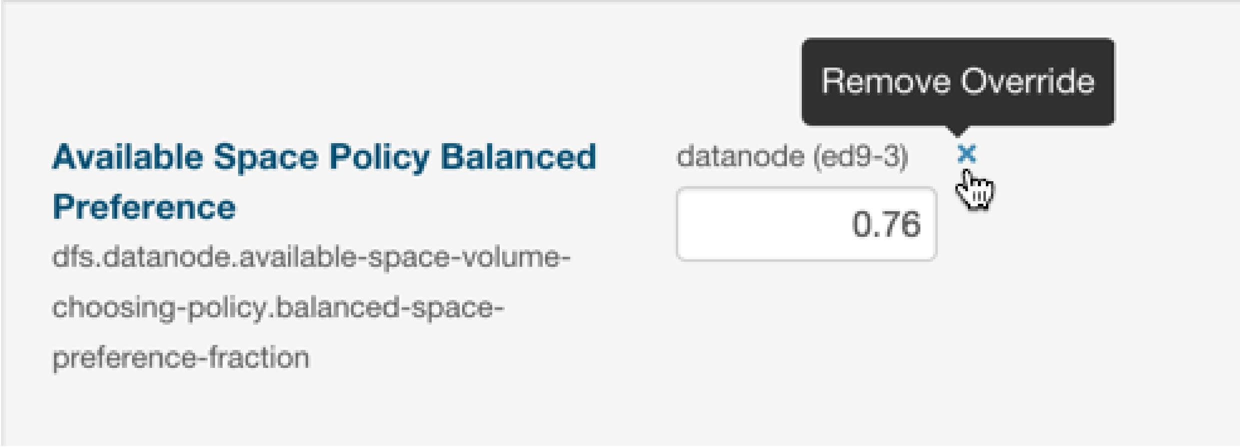
You will most likely need to restart your service or role to have your configuration changes take effect. See Stale Configuration Actions on page 48.

Viewing and Editing Overridden Configuration Properties

To see a list of all role instances that have an override value for a particular configuration setting, go to the Status 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.



Available Space Policy Balanced Preference

dfs.datanode.available-space-volume-choosing-policy.balanced-space-preference-fraction

datanode (ed9-3) ✖

0.76

Resetting Configuration Properties to the Default Value

To reset a property back to its default value, click the

↩

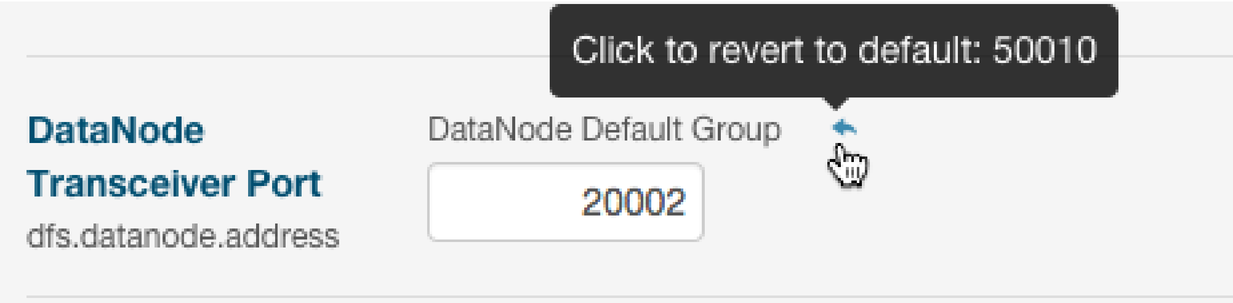
icon. The default value is inserted and the icon turns into an Undo icon

(↺).

Explicitly setting a configuration to the same value as its default (inherited value) has the same effect as using the

↩

icon.



Click to revert to default: 50010

DataNode Transceiver Port

dfs.datanode.address

DataNode Default Group ↩

20002

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.

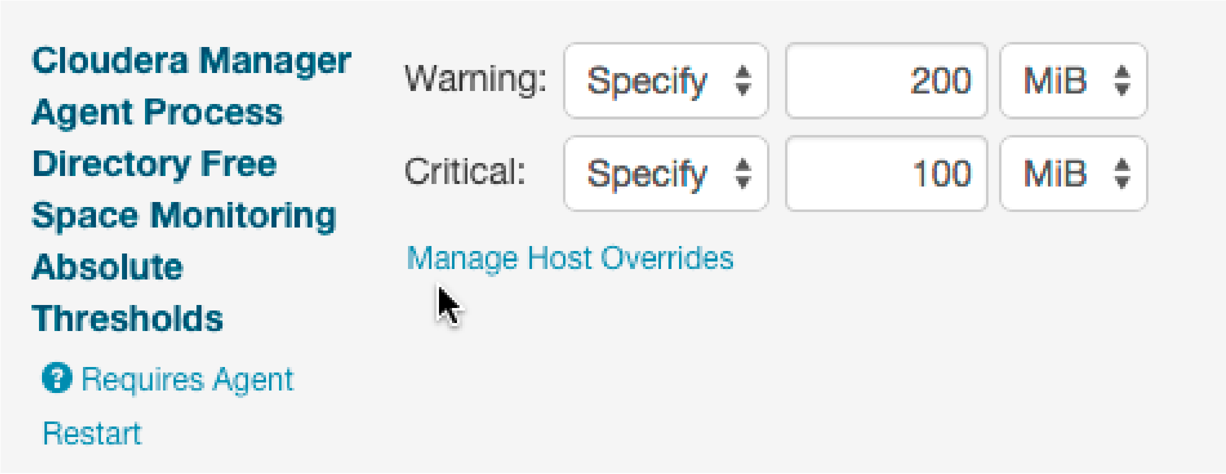
Viewing and Editing Host Overrides

Before you begin

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

Procedure

1. Click the **Hosts** tab.
2. Click the **Configuration** tab.
3. Use the Filters or Search box to locate the property that you want to override.
4. Click the **Manage Host Overrides** link.



Cloudera Manager Agent Process Directory Free Space Monitoring Absolute Thresholds

Requires Agent Restart

Warning: Specify 200 MiB

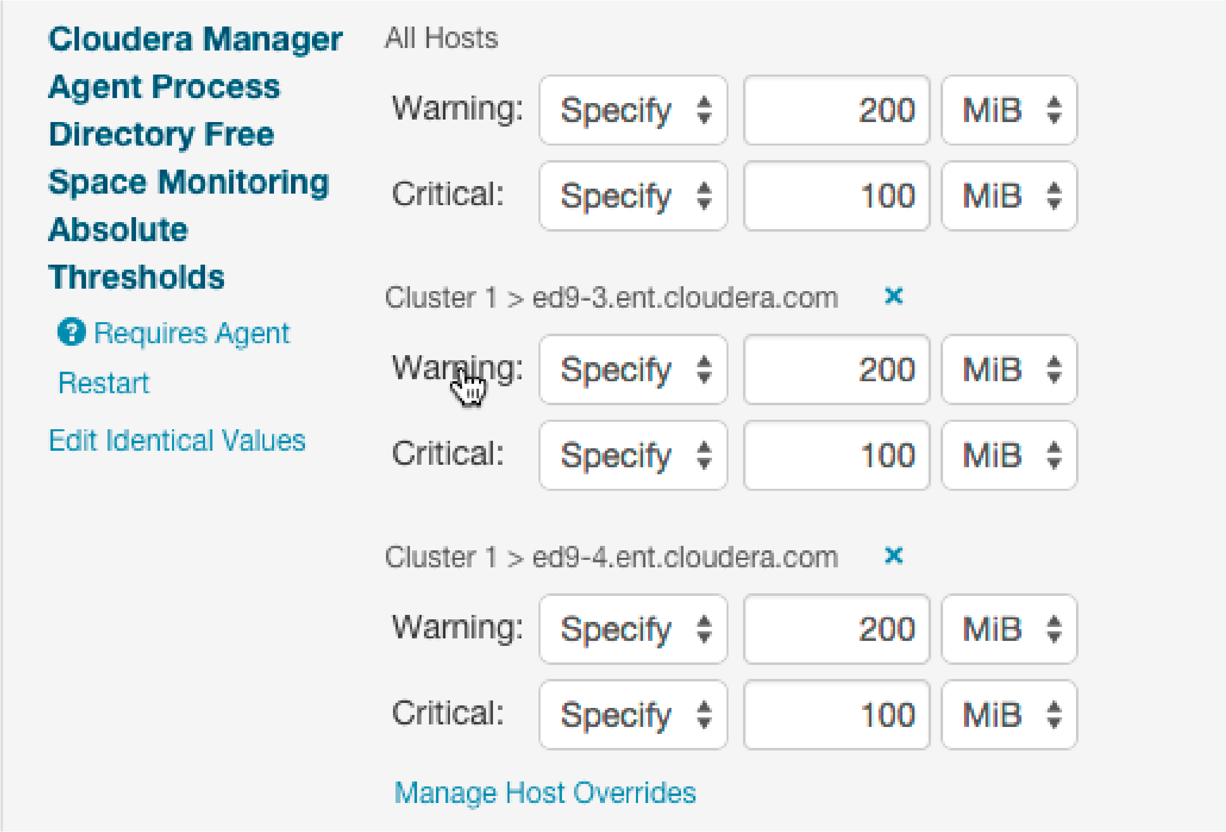
Critical: Specify 100 MiB

Manage Host Overrides

The **Manage Overrides** dialog box displays.

5. Select one or more hosts to override this property.
6. 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.



Cloudera Manager Agent Process Directory Free Space Monitoring Absolute Thresholds

? Requires Agent Restart

Edit Identical Values

All Hosts

Warning: Specify 200 MiB

Critical: Specify 100 MiB

Cluster 1 > ed9-3.ent.cloudera.com ×

Warning: Specify 200 MiB

Critical: Specify 100 MiB

Cluster 1 > ed9-4.ent.cloudera.com ×

Warning: Specify 200 MiB

Critical: Specify 100 MiB

Manage Host Overrides

- 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.

7. If the property indicates **Requires Agent Restart**, restart the agent on the affected hosts.

Restarting Services and Instances after Configuration Changes

About this task

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. 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.For more information, see Stale Configurations on page 46.

Procedure

1. Follow the instructions in #unique_86 or #unique_87/unique_87_Connect_42_section_k3v_fnq_gn.
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.

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

About this task

When viewing the configuration issues, you can suppress each warning. A dialog box opens where you can enter a comment about the suppression. Click **Confirm** when you are done.

Procedure

To suppress warnings from the **All Configuration Issues** screen:

1. Browse to the **Home** screen.
2. Click **Configurations** > **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 **Host** tab, select **Configuration** > **Category** > **Suppressions** and select **Status** > **Non-default**.

Cluster-Wide Configuration

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

- All Clusters

  1. Select **Configuration** and then select one of the following classes of properties:

     - Advanced Configuration Snippets
     - Databases
     - Disk Space Thresholds
     - Local Data Directories
     - Local Data 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
     - Port Configurations
     - Service Dependencies

  You can also select **Configuration Issues** to view a list of configuration issues for all clusters.
- Specific Cluster

  1. On the **Home** page, click a cluster name.
  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

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. See Setting the class path.

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
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:

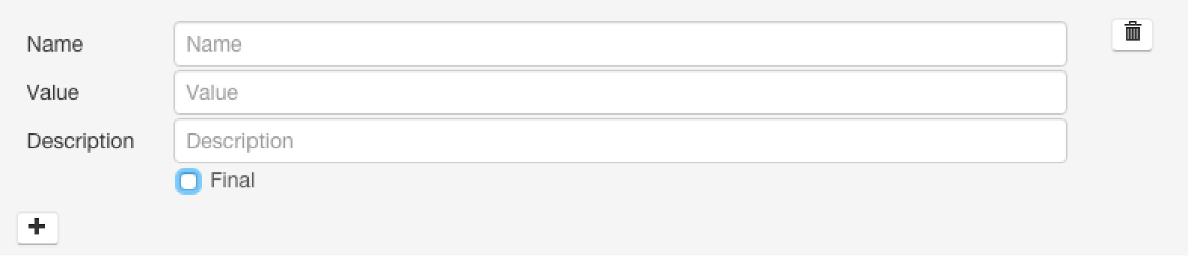
```
host1.domain1 host2.domain2
```

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



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.

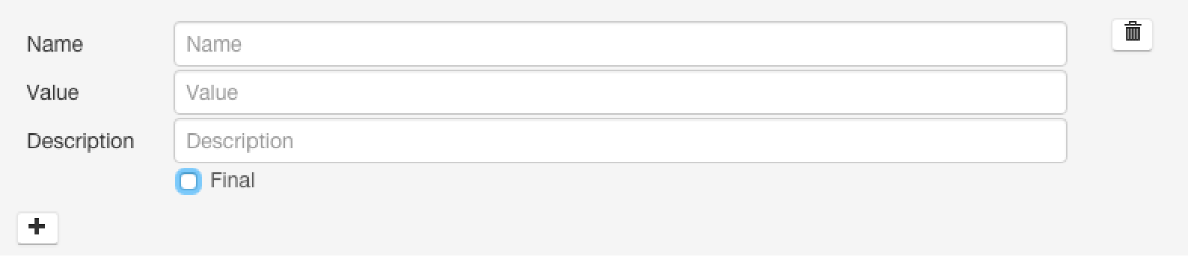
6. Enter a **Reason for change**, and then click **Save Changes** to commit the changes.
7. Restart the service or role or redeploy client configurations as indicated.

Setting an Advanced Configuration Snippet for a Cluster

Procedure

1. To configure a specific cluster, select a cluster from the **Home** > **Status** page, otherwise skip this step to configure all clusters..
2. Select **Configuration** > **Advanced 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>
```

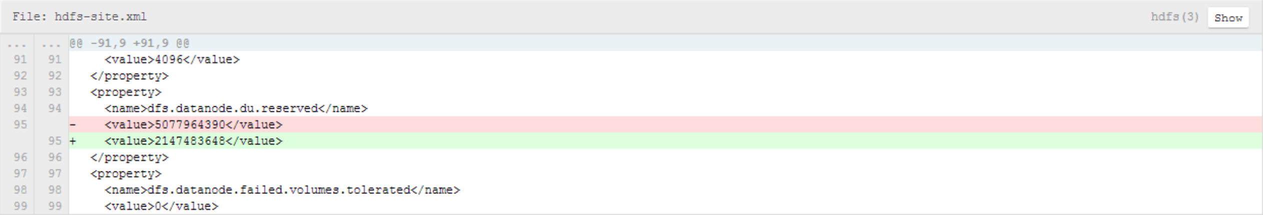
To indicate that the property value cannot be overridden, specify <final>true</final>.

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.

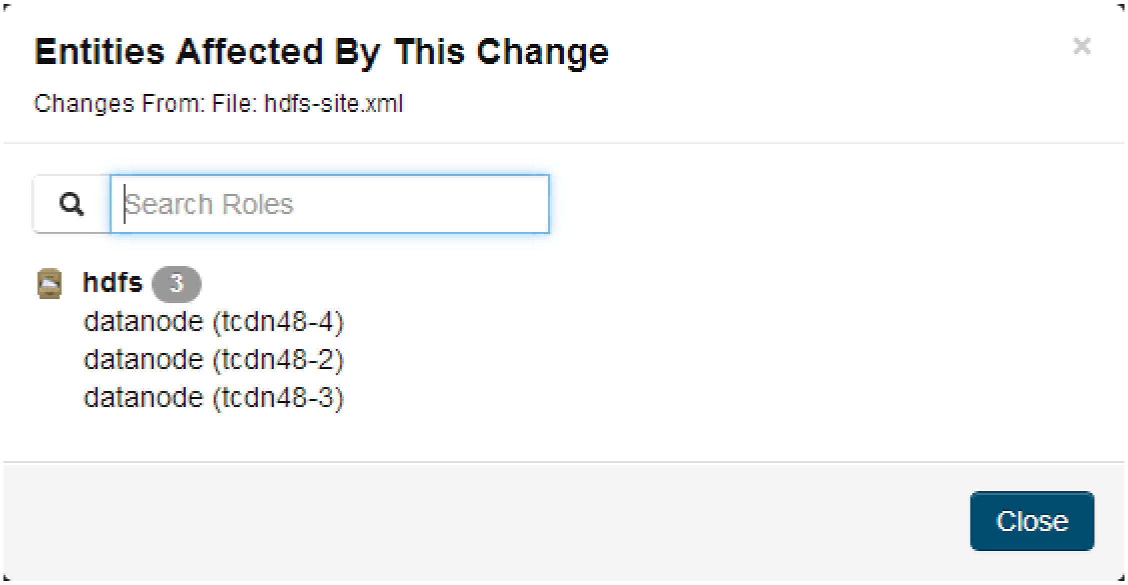
Stale Configurations

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        -    <value>5077964390</value>
      95   +    <value>2147483648</value>
 96   96      </property>
 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)

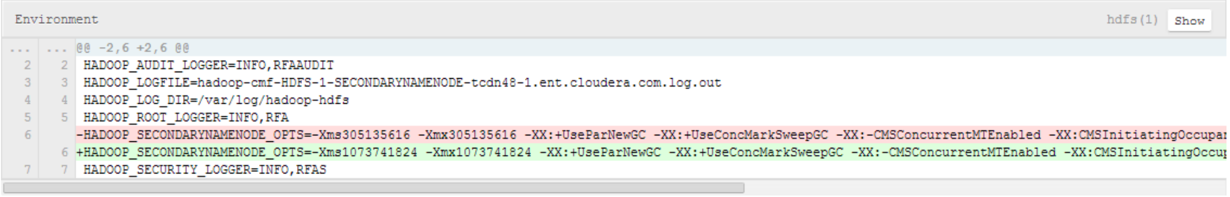
Viewing Stale Configurations

To view stale configurations, click the ⏻ , ↻ , or ⎘ indicator next to a service on the Cloudera Manager Admin Console Home Page on page 13 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.



```
@@ -2,6 +2,6 @@
 HADOOP_AUDIT_LOGGER=INFO,RFAAUDIT
 HADOOP_LOGFILE=hadoop-cmf-HDFS-1-SECONDARYNAMENODE-tcdn48-1.ent.cloudera.com.log.out
 HADOOP_LOG_DIR=/var/log/hadoop-hdfs
 HADOOP_ROOT_LOGGER=INFO,RFA
-HADOOP_SECONDARYNAMENODE_OPTS=-Xms305135616 -Xmx305135616 -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccupa
+HADOOP_SECONDARYNAMENODE_OPTS=-Xms1073741824 -Xmx1073741824 -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:-CMSConcurrentMTEnabled -XX:CMSInitiatingOccu
 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.
- Restart Cloudera Management Service - Runs the restart Cloudera Management Service action.
- Deploy Client Configuration - Runs the cluster deploy client configurations action.

Client Configuration Files

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 on page 46 page.

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 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.

Client Configuration Files

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 on page 46 page.

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

Viewing Configuration Changes

1. For a service, role, or host, click the **Configuration** tab.
2. Click the **History and Rollback** button. The most recent revision, currently in effect, is shown under Current Revision. 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.

Revision Details Dialog

For a service or role instance, shows the following:

- A brief message describing the context of the changes
- The date/time stamp of the change
- The user who performed the change
- The names of any role groups created
- The names of any role groups deleted

For a host instance, shows just a message, date and time stamp, and the user.

The dialog box contains two tabs:

- Configuration Values - displays configuration value changes, where changes are organized under the role group to which they were applied. (For example, if you changed a Service-Wide property, it will affect all role groups for that service). For each modified property, the Value column shows the new value of the property and the previous value.
- Group Membership - displays changes to the changed the group membership of a role instance (moved the instance from one group to another). This tab is only shown for service and role configurations.

Reverting Configuration Changes

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. You cannot use this feature to:

- Revert NameNode high availability. You must perform this action by explicitly disabling high availability.
- Disable Kerberos security.
- Revert role group actions (creating, deleting, or moving membership among groups). You must perform these actions explicitly in the Role Groups feature.

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.

Reverting Configuration Changes

About this task

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.

What to do next

⚠ **Important:** This feature can only be used to revert changes to configuration values. You cannot use this feature to:

- Revert NameNode high availability. You must perform this action by explicitly disabling high availability.
- Disable Kerberos security.
- Revert role group actions (creating, deleting, or moving membership among groups). You must perform these actions explicitly in the Role Groups feature.

Client Configuration Files

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 on page 46 page.

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 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.