Cloudera Manager 7.0.2

Concepts

Date published: 2020-02-11
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Overview

Cloudera Manager is an application you use together with the Cloudera Management Admin Console to manage clusters. You use the Cloudera Management Admin Console to create and manage cloud provider environments, users, Data Lakes, clusters, Data Warehouses, Machine Learning workspaces and to register "classic" clusters. You use Cloudera Manager to monitor and configure clusters created by the Cloudera Management Console.

This primer introduces the basic concepts, structure, and functions of Cloudera Manager.

Cloudera Manager Terminology

Terminology used in Cloudera Manager.

To effectively use Cloudera Manager, you should first understand its terminology.

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.
- In Cloudera Manager, a logical entity that contains a set of hosts, a single version of Cloudera Runtime installed on the hosts, and the service and role instances running on the hosts. A host can belong to only one cluster.

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.

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.

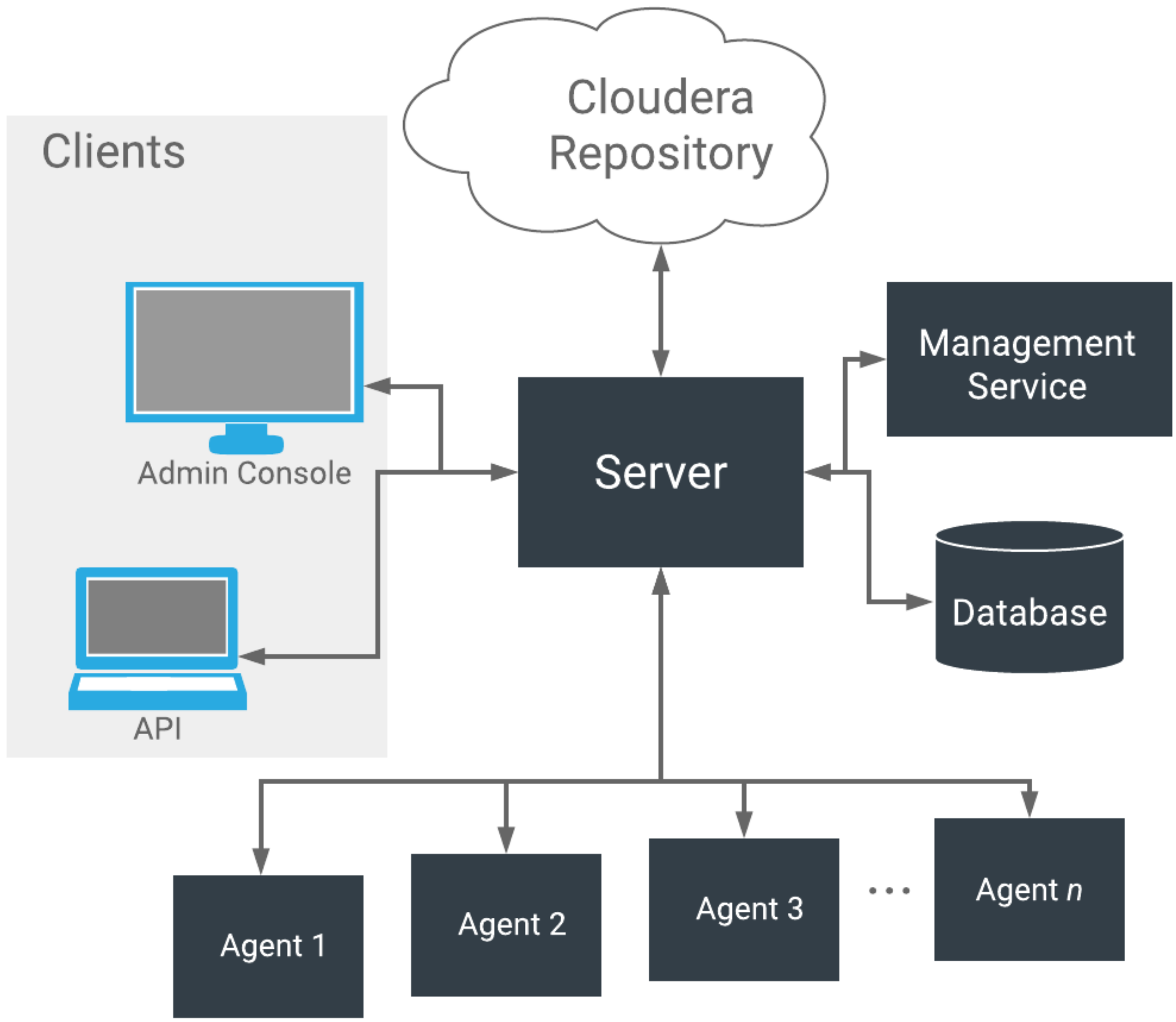
static service pool

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

Cloudera Manager Architecture

Description of the components that comprise Cloudera Manager.

As depicted below, the heart of Cloudera Manager is the Cloudera Manager Server. The Server hosts the Cloudera Manager Admin Console, the Cloudera Manager API, and the application logic, and is responsible for 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:
  - Cloudera Manager Admin Console - Web-based UI with which administrators manage clusters and Cloudera Manager.
  - Cloudera Manager API - API developers use to 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.

Related Information

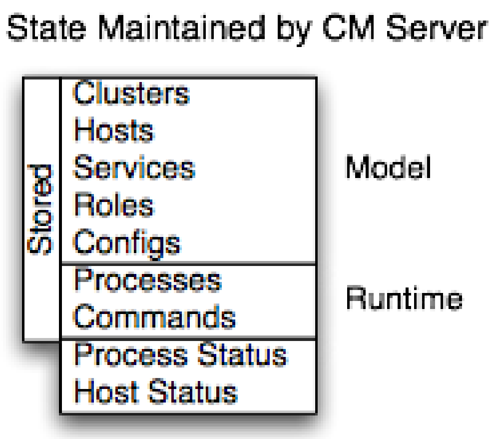
Cloudera Management Service

Cloudera Manager Admin Console

Cloudera Manager API

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 clusters 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.

Related Information

Advanced Configuration Snippets

Cloudera Manager Admin Console

Cloudera Manager Admin Console is the web-based interface that you use to configure, manage, and monitor Cloudera Runtime. After you have created a cluster using the Data Hub service in the Management Console, you can access Cloudera Manager from the Data Hub Clusters page in the Cloudera Management Console.

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

Note: Depending on the user role used to log in, some items may not appear in the Cloudera Manager Admin Console.

- 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.
- 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.
  - 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.
- 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.
- Charts - Query for metrics of interest, display them as charts, and display personalized chart dashboards.
- Running Commands Indicator - displays the number of commands currently running for all services or roles.

- Support - Displays various support actions. The subcommands are:
  - Send Diagnostic Data - Sends data to Cloudera Support to support troubleshooting.
  - Support Portal (Cloudera Enterprise) - Displays the Cloudera Support portal.
  - Mailing List (Cloudera Express) - 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, the HomeStatus tab displays. You can also go to the HomeStatus 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

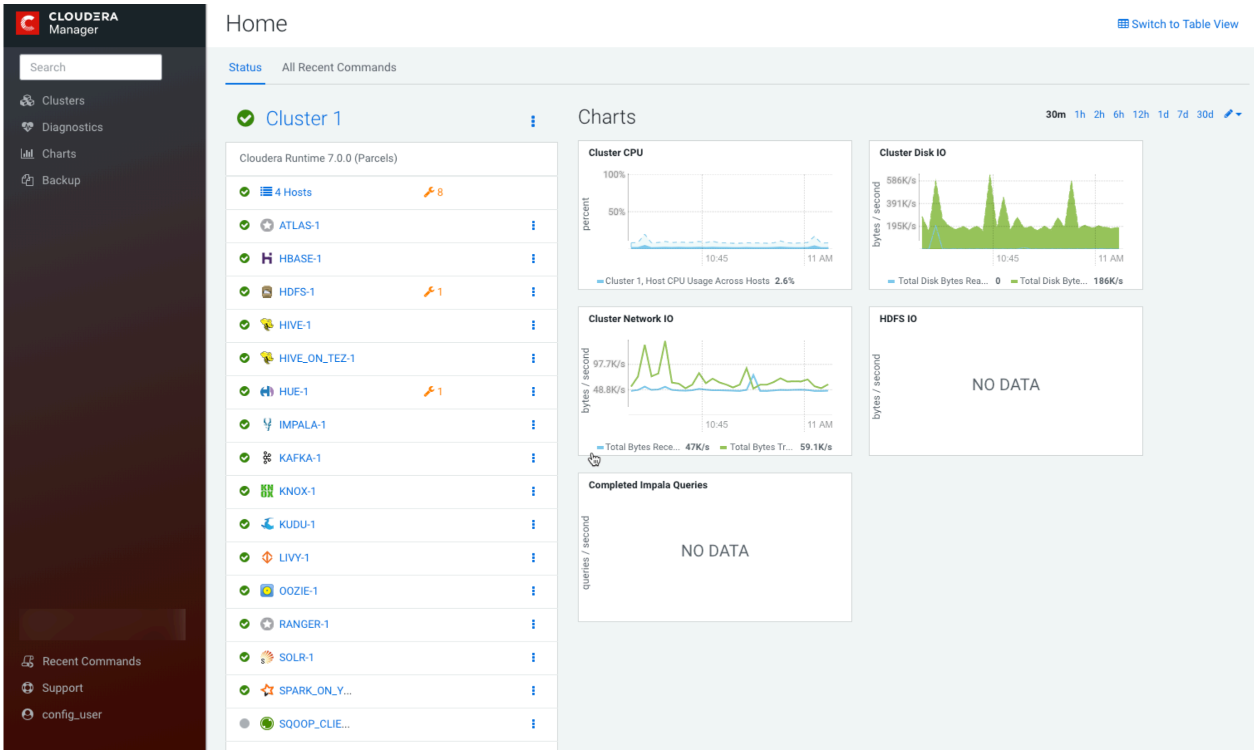
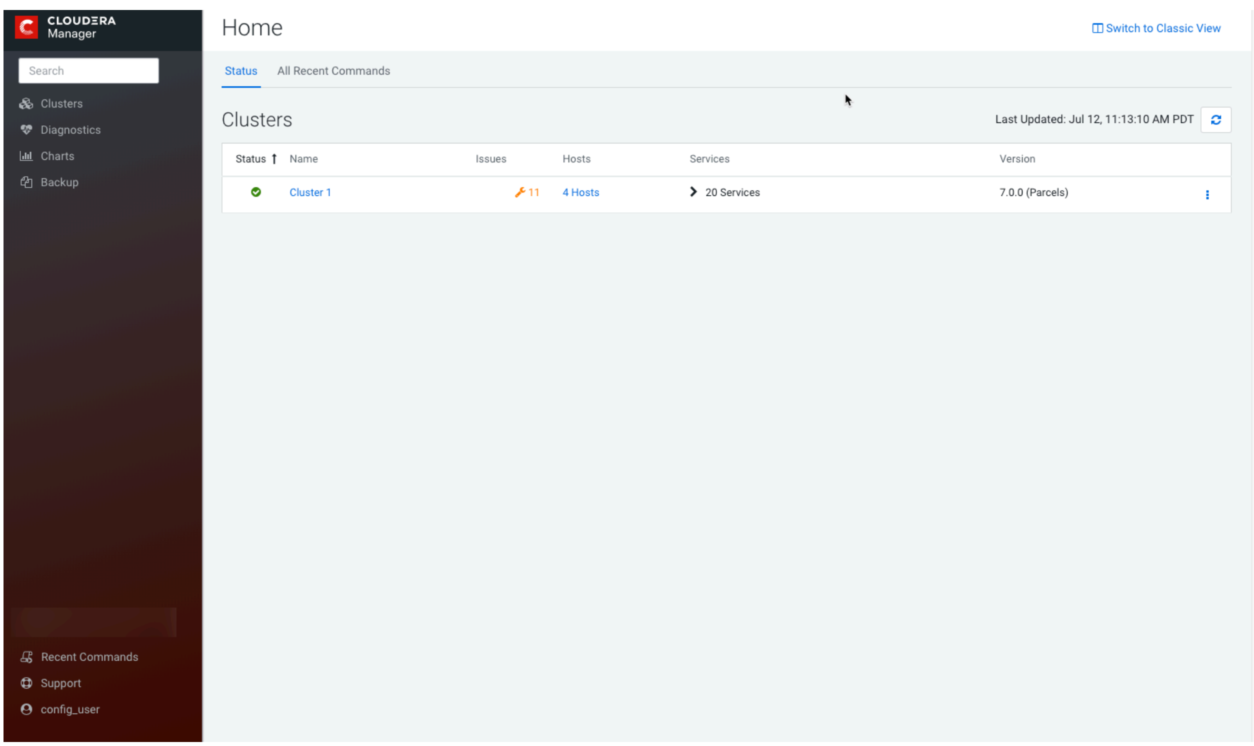


Figure 2: Cloudera Manager Admin Console: Table View



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 AdministrationSettingsOtherMaximum 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 AdministrationSettingsOtherMaximum 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 the Actions Menu ( ) 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.
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.
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 page.To bring the cluster up-to-date, click the Refresh or Restart button on the Stale Configurations page.
	Client configuration redeployment required	Indicates that the client configuration for a service should be redeployed. Click the indicator to display the Stale Configurations page. To bring the cluster up-to-date, click the Deploy Client Configuration button on the Stale Configurations page or manually redeploy the client configuration.

- 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 (dashboards) that summarize resource utilization (IO, CPU usage) and processing metrics.

  Click a line, stack area, scatter, or bar chart to expand it into a full-page view with a legend for the individual charted entities as well more fine-grained axes divisions.

  By default the time scale of a dashboard is 30 minutes. To change the time scale, click a duration link 30m 1h 2h 6h 12h 1d 7d 30d at the top-right of the dashboard.

  To set the dashboard type, click and select one of the following:

  - Custom - displays a custom dashboard.
  - Default - displays a default dashboard.
  - Reset - resets the custom dashboard to the predefined set of charts, discarding any customizations.

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.

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 SupportAbout.

Related Information

Managing Cloudera Runtime Services

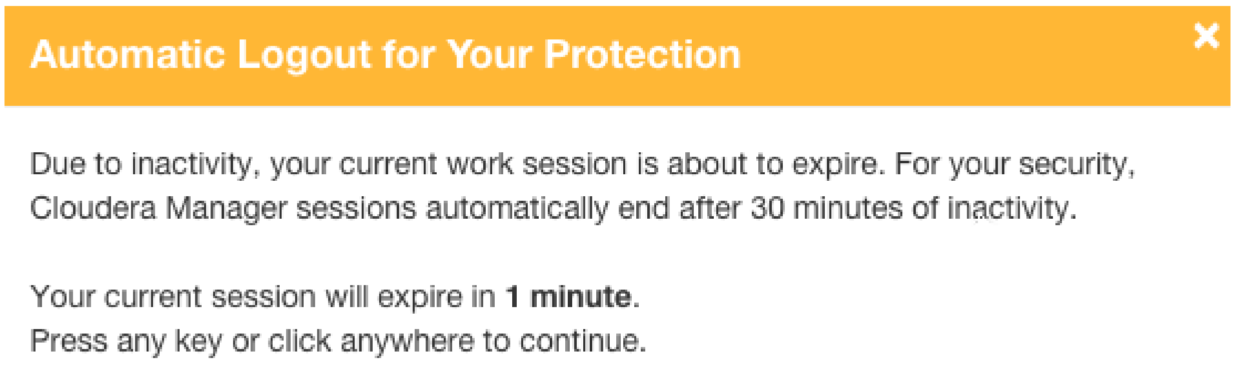
Viewing and Running Recent Commands

Automatic Logout

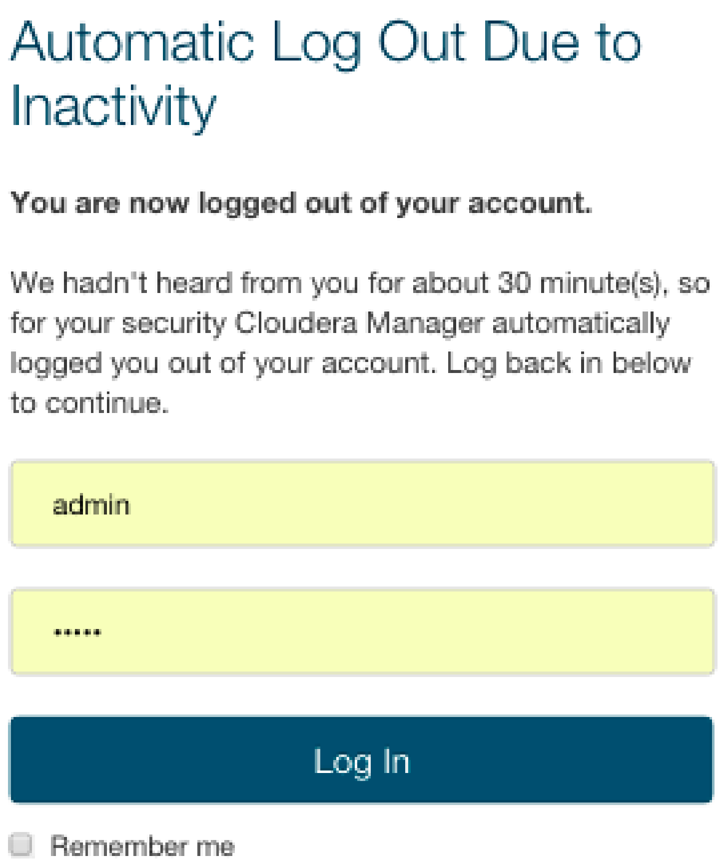
For security purposes, Cloudera Manager automatically logs out a user session after 30 minutes. You can change this session logout period.

Procedure

1. Click AdministrationSettings.
2. Click CategorySecurity.
3. Edit the Session Timeout property.
4. Enter a Reason for change, and then click Save Changes to commit the changes.
   When the timeout is one minute from triggering, the user sees the following message:



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



Process Management

Starting and stopping processes using Cloudera Manager.

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.

Related Information

Supervisor: A Process Control System

Host Management

Cloudera Manager provides several features to manage the hosts in your 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 roles to hosts. Cloudera Manager automatically deploys all software required to participate as a managed host in a cluster: JDK, Cloudera Manager Agent, 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 Cloudera Runtime 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 (temporary file storage), 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.

Important:

Cloudera Manager is designed to operate on OS platforms or containers where the root PID is an INIT process. This INIT process should possess the necessary privileges to effectively terminate any zombie processes. This helps to maintain a clean process table, preventing any unwanted clutter from lingering zombie processes that were earlier managed by Cloudera Manager.

Related Information

Supervisor: A Process Control System

Monitoring a Cluster Using Cloudera Manager

Cloudera Manager provides many features for monitoring the health and performance of the components of your clusters (hosts, service daemons) as well as the performance and resource demands of the jobs running on your clusters.

The following monitoring features are available in Cloudera Manager:

- Monitoring Cloudera Runtime Services - describes how to view the results of health tests at both the service and role instance level. Various types of metrics are displayed in charts that help with problem diagnosis. Health tests include advice about actions you can take if the health of a component becomes concerning or bad. You can also view the history of actions performed on a service or role, and can view an audit log of configuration changes.
- Monitoring Hosts - describes how to view information pertaining to all the hosts on your cluster: which hosts are up or down, current resident and virtual memory consumption for a host, what role instances are running on a host, which hosts are assigned to different racks, and so on. You can look at a summary view for all hosts in your cluster or drill down for extensive details about an individual host, including charts that provide a visual overview of key metrics on your host.
- Activities - describes how to view the activities running on the cluster, both at the current time and through dashboards that show historical activity, and provides many statistics, both in tabular displays and charts, about the resources used by individual jobs. You can compare the performance of similar jobs and view the performance of individual task attempts across a job to help diagnose behavior or performance problems.

- Events - describes how to view events and make them available for alerting and for searching, giving you a view into the history of all relevant events that occur cluster-wide. You can filter events by time range, service, host, keyword, and so on.
- Alerts - describes how to configure Cloudera Manager to generate alerts from certain events. You can configure thresholds for certain types of events, enable and disable them, and configure alert notifications by email or using SNMP trap for critical events. You can also suppress alerts temporarily for individual roles, services, hosts, or even the entire cluster to allow system maintenance/troubleshooting without generating excessive alert traffic.
- Charting Time-Series Data - describes how to search metric data, create charts of the data, group (facet) the data, and save those charts to user-defined dashboards.
- Logs - describes how to access logs in a variety of ways that take into account the current context you are viewing. For example, when monitoring a service, you can easily click a single link to view the log entries related to that specific service, through the same user interface. When viewing information about a user's activity, you can easily view the relevant log entries that occurred on the hosts used by the job while the job was running.
- Reports - describes how to view historical information about disk utilization by user, user group, and by directory and view cluster job activity user, group, or job ID. These reports are aggregated over selected time periods (hourly, daily, weekly, and so on) and can be exported as XLS or CSV files. You can also manage HDFS directories as well, including searching and setting quotas.
- Troubleshooting Cluster Configuration and Operation - contains solutions to some common problems that prevent you from using Cloudera Manager and describes how to use Cloudera Manager log and notification management tools to diagnose problems.

Cloudera Management Service

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

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

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

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

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

Health Tests

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

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

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

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

Metric Collection and Display

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

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

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

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

Events, Alerts, and Triggers

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

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

Related Information

tsquery Language

Cluster Configuration Overview

Cloudera Manager manages the configuration of all roles running in a cluster.

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 Cloudera Runtime default values for that configuration. Cloudera Manager deviates when the Cloudera Runtime default is not a recommended configuration or when the default values are illegal.

Related Information

Cloudera Runtime Configuration Properties Reference

Server and Client Configuration

Cloudera Manager generates server and client configuration files from its database.

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.

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 Cloudera Manager API is served on the same host and port as the Cloudera Manager Admin Console on page 9, and does not require an extra process or extra configuration. The API supports HTTP Basic Authentication, accepting the same users and credentials as the Cloudera Manager Admin Console.

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

API Documentation Resources

- Quick Start
- Python Client (deprecated)
- Python Client (Swagger-based)
- Java Client (Swagger-based)

Using the Cloudera Manager API

Procedures, examples and resources for using the Cloudera Manager API to automate cluster operations.

Using the Cloudera Manager API to Obtain Configuration Files

You can use 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/v40/clusters/clusterName/service
s/serviceName/roles
```

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

```
http://cm_server_host:7180/api/v40/clusters/clusterName/service
s/serviceName/roles/roleName/process
```

3. Obtain the content of any particular file:

```
http://cm_server_host:7180/api/v40/clusters/clusterName/service
s/serviceName/roles/roleName/process/
configFiles/configFileName
```

For example:

```
http://cm_server_host:7180/api/v40/clusters/Cluster%201/services/OOZIE-1/r
oles/
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/v40/clusters/Cluster%201/servic
es/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 l
ine) to be inserted into a roles
   environment. Applies to configurations of all roles in this service exce
pt 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/v40/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/v40/hosts/2c2e951c-adf2-4780-a69f-0382181f1
821?view=FULL
```

Backing Up and Restoring the Cloudera Manager Configuration

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

About this task

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

Procedure

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/v40/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 HomeStatus tab, click to the right of the cluster name and select Stop.
      2. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services.

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

      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/v40/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
```

Using the Cloudera Manager API for Cluster Automation

How to use the Cloudera Manager API to automate cluster management.

One of the complexities of Apache Hadoop is the need to deploy clusters of servers, potentially on a regular basis. If you maintain hundreds of test and development clusters in different configurations, this process can be complex and cumbersome if not automated.

Cluster Automation Use Cases

Cluster automation is useful in various situations. For example, you might work on many versions of CDH, which works on a wide variety of OS distributions (RHEL 6, Ubuntu Precise and Lucid, Debian Wheezy, and SLES 11). You might have complex configuration combinations—highly available HDFS or simple HDFS, Kerberized or non-secure, YARN or MRv1, and so on. With these requirements, you need an easy way to create a new cluster that has the required setup. This cluster can also be used for integration, testing, customer support, demonstrations, and other purposes.

You can install and configure Hadoop according to precise specifications using the Cloudera Manager REST API. Using the API, you can add hosts, install CDH, and define the cluster and its services. You can also tune heap sizes, set up HDFS HA, turn on Kerberos security and generate keytabs, and customize service directories and ports. Every configuration available in Cloudera Manager is exposed in the API.

The API also provides access to management functions:

- Obtaining logs and monitoring the system
- Starting and stopping services
- Polling cluster events
- Creating a disaster recovery replication schedule

For example, you can use the API to retrieve logs from HDFS, HBase, or any other service, without knowing the log locations. You can also stop any service with no additional steps.

Use scenarios for the Cloudera Manager API for cluster automation might include:

- OEM and hardware partners that deliver Hadoop-in-a-box appliances using the API to set up CDH and Cloudera Manager on bare metal in the factory.
- Automated deployment of new clusters, using a combination of Puppet and the Cloudera Manager API. Puppet does the OS-level provisioning and installs the software. The Cloudera Manager API sets up the Hadoop services and configures the cluster.
- Integrating the API with reporting and alerting infrastructure. An external script can poll the API for health and metrics information, as well as the stream of events and alerts, to feed into a custom dashboard.

Java API Example

This example covers the Java API client.

To use the Java client, add this dependency to your project's pom.xml:

```
<project>
  <repositories>
    <repository>
      <id>cdh.repo</id>
      <url>https://repository.cloudera.com/artifactory/cloudera-repos</url>
      <name>Cloudera Repository</name>
    </repository>
    …
  </repositories>
  <dependencies>
    <dependency>
      <groupId>com.cloudera.api</groupId>
      <artifactId>cloudera-manager-api</artifactId>
      <version>4.6.2</version>      <!-- Set to the version of Cloudera Man
ager you use -->
    </dependency>
    …
  </dependencies>
  ...
</project>
```

The Java client works like a proxy. It hides from the caller any details about REST, HTTP, and JSON. The entry point is a handle to the root of the API:

```
RootResourcev40 apiRoot = new ClouderaManagerClientBuilder().withHost("cm.c
loudera.com")
.withUsernamePassword("admin", "admin").build().getRootv40();
```

From the root, you can traverse down to all other resources. (It's called "v40" because that is the current Cloudera Manager API version, but the same builder will also return a root from an earlier version of the API.) The tree view shows some key resources and supported operations:

- RootResourcev40
  - ClustersResourcev40 - host membership, start cluster
    - ServicesResourcev40 - configuration, get metrics, HA, service commands
      - RolesResource - add roles, get metrics, logs
      - RoleConfigGroupsResource - configuration
    - ParcelsResource - parcel management
- HostsResource - host management, get metrics
- UsersResource - user management

For more information, see the Javadoc.

The following example lists and starts a cluster:

```
// List of clusters
ApiClusterList clusters = apiRoot.getClustersResource().readClusters(DataVie
w.SUMMARY);
for (ApiCluster cluster : clusters) {
  LOG.info("{}: {}", cluster.getName(), cluster.getVersion());
}

// Start the first cluster
ApiCommand cmd = apiRoot.getClustersResource().startCommand(clusters.get(
0).getName());
while (cmd.isActive()) {
   Thread.sleep(100);
   cmd = apiRoot.getCommandsResource().readCommand(cmd.getId());
}
LOG.info("Cluster start {}", cmd.getSuccess() ? "succeeded" : "failed " + c
md.getResultMessage());
```

Python Example

You can see an example of automation with Python at the following link: Python example. The example contains information on the requirements and steps to automate a cluster deployment.

Exporting and Importing Cloudera Manager Configuration

How to use the Cloudera Manager API to export and import Cloudera Manager configurations.

You can use the Cloudera Manager API to programmatically export and import a definition of all the entities in your Cloudera Manager-managed deployment—clusters, service, roles, hosts, users and so on. See Using the Cloudera Manager API on page 20 for more information on how to manage deployments using the resource.