What would you do if you knew?™

Hortonworks Data Platform for Teradata

Administrator Guide

Release 2.3, 2.4

B035-6035-075K

June 2016

The information contained in this document is provided on an "as-is" basis, without warranty of any kind, either express or implied, including the implied warranties of merchantability, **fitness** for a particular purpose, or non-infringement. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. In no event will Teradata Corporation be liable for any indirect, direct, special, incidental, or consequential damages, including lost **profits** or lost savings, even if expressly advised of the possibility of such damages.

The information contained in this document may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

Information contained in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. Teradata Corporation may also make improvements or changes in the products or services described in this information at any time without notice.

To maintain the quality of our products and services, we would like your comments on the accuracy, clarity, organization, and value of this document. Please e-mail: teradata-books@lists.teradata.com

Any comments or materials (collectively referred to as "Feedback") sent to Teradata Corporation will be deemed non-confidential. Teradata Corporation will have no obligation of any kind with respect to Feedback and will be free to use, reproduce, disclose, exhibit, display, transform, create derivative works of, and distribute the Feedback and derivative works thereof without limitation on a royalty-free basis. Further, Teradata Corporation will be free to use any ideas, concepts, know-how, or techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, or marketing products or services incorporating Feedback.

Table of Contents

Chapter 3:
##

Chapter 4:
##

Preface

Revision History

Date	Release	Description
June 2016	2.4.2	Added the system setup_kerberos command.
March 2016	2.4	Added the support **manager_config** command.
January 2016	2.3	• Updated *Setting Up External Access* • Added the support tune_coresidence command
October 2015	2.3	Revised hcli commands
July 2015	2.3	Initial release

Audience

This guide is intended for use by:

- System administrators
- Database administrators and relational database developers
- Customers
- Teradata Customer Support

This guide applies to Hortonworks Data Platform on the Teradata Appliance for Hadoop platforms, not to other distributions of Hadoop software or Hadoop software on commodity hardware.

Portions of this document are based upon materials under license from Hortonworks Inc.

Additional Information

Related Links

URL	Description
https://tays.teradata.com	Use Teradata At Your Service to access Orange Books, technical alerts, and knowledge repositories, view and join forums, and download software packages.

URL	Description
www.teradata.com	External site for product, service, resource, support, and other customer information.

Related Documents

Documents are located at http://www.info.teradata.com.

Title	Publication ID
Teradata Appliance for Hadoop 5 Platform Product and Site Preparation Guide Describes specifications and preparation procedures for Teradata Appliance for Hadoop 5.	B035-6030
Teradata Appliance for Hadoop 3 and 4 Platform Product and Site Preparation Guide Describes specifications and preparation procedures for Teradata Appliance for Hadoop 3 and 4.	B035-5523
Hortonworks Data Platform for Teradata Administrator Guide Summarizes new features and fixed issues associated with the release.	B035-6035
Hortonworks Data Platform for Teradata Installation, Configuration, and Upgrade Guide for Customers Describes how to install and configure Hortonworks Data Platform for Teradata components.	B035-6036
Teradata QueryGrid: Teradata Database-to-Hadoop Describes the Teradata QueryGrid: Teradata Database-to-Hadoop SQL interface for transferring data between Teradata Database and remote Hadoop hosts.	B035-1185
Teradata Parallel Transporter Reference Describes Teradata Parallel Transporter (Teradata PT) high-performance data extraction, loading, and updating operations for Teradata Database.	B035-2436
Teradata Studio User Guide Describes the Teradata Studio features.	B035-2041
Teradata Viewpoint User Guide Describes the Teradata Viewpoint portal, portlets, and system administration features.	B035-2206
Teradata Data Mover User Guide Describes how to use the Teradata Data Mover portlets and command-line interface.	B035-4101
Teradata Ecosystem Manager User Guide Describes how to use Teradata Ecosystem Manager portlets.	B035-3201
Parallel Upgrade Tool (PUT) Reference	B035-5716

Title	Publication ID
Describes how to install application software using PUT.	

Customer Education

Teradata Customer Education delivers training for your global workforce, including scheduled public courses, customized on-site training, and web-based training. For information about the classes, schedules, and the Teradata Certification Program, go to www.teradata.com/TEN/.

Product Safety Information

This document may contain information addressing product safety practices related to data or property damage, identified by the word *Notice*. A notice indicates a situation which, if not avoided, could result in damage to property, such as equipment or data, but not related to personal injury.

Example

Notice: Improper use of the Reconfiguration utility can result in data loss.

CHAPTER 1

Hortonworks Data Platform in the Teradata Environment

Hortonworks for Teradata Overview

Hortonworks Data Platform (HDP) is an open source distribution of Apache Hadoop.

The Ambari GUI simplifies provisioning, managing, and monitoring Apache Hadoop clusters. Ambari installs Hadoop services across any number of Hadoop hosts and configures Hadoop services for the cluster. Ambari provides a centralized management solution for stopping, starting, and reconfiguring Hadoop services across the entire cluster. Ambari also provides a dashboard for monitoring the status of the Hadoop cluster. The dashboard leverages the Ambari Metrics System for metrics collection and the Ambari Alert Framework for providing alerts.

HDP Server Nodes

Hortonworks for Teradata is comprised of master, data, and edge nodes.

Master Node for Hadoop

Controls the cluster by storing metadata and running master services, including:

- HCatalog: Describes the structure of data stored in HDFS
- Hive: Queries structured data in HDFS
- JournalNode: Modifies log changes in HDFS from the namenode
- Namenode: Manages HDFS storage; high availability requires an active and standby namenode
- YARN: Schedules application jobs and manages and allocates resources
- Zookeeper: Synchronizes distributed components as well as monitoring the namenode

Data Node for Hadoop

- Stores HDFS blocks
- Answers queries from the namenode for filesystem operations
- Allows client applications to communicate directly with the data node when the namenode determines the data location

Edge Node for Hadoop

The edge node allows client applications to run independently of the master node, reducing both the risk in testing new applications and the impact on Teradata Database throughput by enhancing load performance, which TASM or Teradata Integrated Workload Management ruleset throttles. Located between the Hadoop cluster and the customer network, the edge node runs client services for the cluster:

- Allows access for external applications and user access to the Hadoop environment
- Permits access control
- Enforces policy oversight
- Logs metadata
- Provides fast connections by communicating to the Hadoop cluster over the internal InfiniBand network

Hadoop Interfaces

You can interact with or view information about Hadoop using several methods:

- Administrative
  - Apache Ambari
  - Hadoop Command-Line Interface (hcli)
  - Teradata Viewpoint
  - Teradata Unity Ecosystem Manager
- Data Load and Transfer
  - Teradata Connector for Hadoop (TDCH)
  - Teradata QueryGrid
  - Teradata Parallel Transporter (TPT)
  - Teradata Studio
  - Teradata Unity Data Mover

Apache Ambari

Apache Ambari is a web-based tool that allows the following:

- Monitoring cluster performance and utilization
- Managing configurations of cluster services
- Provisioning services in Hadoop clusters

Hadoop Command-Line Interface (hcli)

The Hadoop Command-Line Interface (hcli) is a user interface providing command-line options for monitoring Hadoop services and cluster management tasks. For information on the hcli commands, see Hadoop Command-Line Interface.

Teradata Connector for Hadoop (TDCH)

Teradata Connector for Hadoop (TDCH) is a set of Application Program Interface (API) and tools that support high performance parallel bi-directional data movement between Teradata Database systems and the Hadoop ecosystem of products.

Teradata Parallel Transporter (TPT)

Parallel and scalable data-loading and unloading utility used for

- High volume batch loads into empty tables
- High volume loads into populated tables
- Continuous loads into existing tables
- Exporting or extracting data from tables

TPT includes interface modules to acquire data from all major sources (e.g., files, ODBC-compliant databases, pipes, and queues) through three control mechanisms:

- A scripting language
- An application programming interface (API)
- Easy Loader, a command-line interface to invoke simple load jobs

Teradata QueryGrid

Teradata QueryGrid: Teradata Database-to-Hadoop

Teradata QueryGrid: Teradata-Hadoop connector provides a SQL interface for transferring data between Teradata Database and remote Hadoop hosts. It allows you to do the following:

- Import Hadoop data into a temporary or permanent Teradata table
- Export data from temporary or permanent Teradata tables into existing Hadoop tables
- Create or drop tables in Hadoop from Teradata Database
- Reference tables on the remote hosts in SELECT and INSERT statements
- Select Hadoop data for use with a business tool
- Select and join Hadoop data with data from independent data warehouses for analytical use

The Teradata QueryGrid: Teradata-Hadoop connector provides the following benefits:

- The ability to export data to Hadoop servers
- A simplified grammar, making the Teradata QueryGrid Teradata-to-Hadoop connector easier to use. After creating a foreign server once, you can then use the server name instead of providing detailed connection information for each SQL query.
- The automatic push down of qualification columns and grammar to execute on a remote host
- The ability to qualify both columns and partitions involved in the query to reduce the amount of data returned
- The ability to create an authorization object to securely store credentials. Foreign servers can be defined to use an authorization object to authenticate with a security system, such as LDAP or Kerberos, protecting Hadoop clusters.

  Note: The Teradata QueryGrid Teradata-to-Hadoop connector does not currently support the use of Kerberos for external security.

- Privileges to control who can read and write to the servers and tables on remote hosts

The syntax used for the connector is a Teradata extension to the ANSI SQL:2011 standard.

Teradata QueryGrid: Teradata Aster Database-to-Hadoop

Teradata QueryGrid Aster-Hadoop is a connector which provides access to Hadoop data tables using standard SQL queries in a low latency, data discovery environment. The table and storage management services for data stored in Hadoop is HCatalog. The connector provides deep metadata layer integration with Apache Hadoop HCatalog.

Some of the benefits of the connector are:

- Querying HCatalog from Aster Database and supporting partitions and partition pruning on HCatalog to improve query performance
- Using Aster Database to access the HCatalog metadata directly, and listing all databases and tables in HCatalog from Aster using the Aster Database cluster Terminal (ACT)
- Using Aster Database to retrieve data from HCatalog, and automatically mapping HCatalog datatype value to the Aster datatype value
- Enabling complex queries which include some data from Hadoop and some data from Aster Database through SQL
- Extending the Aster Analytics library over data stored in Hadoop
- Providing a simple way to load Hadoop data into Aster Database tables
- Creating views of HCatalog data within Aster Database, which enables some business intelligence (BI) tools to generate queries against this HCatalog data

For information on connecting Aster Database to Hadoop nodes, including best-practices and network implementation strategies, and Teradata QueryGrid setup, refer to the *Teradata Aster Big Analytics Appliance Database User Guide*.

Teradata Studio

Client-based graphical interface for performing

- Database administration
- Query development
- Management tasks
    - Teradata Databases
    - Teradata Aster Databases
    - Hadoop systems

Teradata Studio is built on the Eclipse Rich Client Platform (RCP), which takes advantage of the RCP framework for building and deploying native GUI applications. It extends the Eclipse Data Tools Platform (DTP) to provide enhancements for accessing objects and data stored in Teradata-supported databases. Teradata Studio supports the Teradata Unified Data Architecture.

Teradata Unity Data Mover

Data Mover supports moving tables from Teradata Database to Hadoop, and Hive tables from Hadoop to Teradata Database, using the Data Mover portlet or the datamove create command in the command-line interface. Data Mover uses either Teradata Connector for

Hadoop (TDCH) or the Teradata QueryGrid Teradata Database-to-Hadoop command; Data Mover selects which protocol to use automatically.

Data Mover allows bi-directional data movement of objects between databases for specific database versions. See *Teradata Data Mover User Guide*, B035-4101.

Teradata Unity Ecosystem Manager

A monitoring application which

- Determines availability
- Understands events, including every step, process, and inter-relationship within an enterprise data warehouse (EDW) from data acquisition to report delivery
- Knows when one process fails and which users and applications are affected
- Allows UDA workflows to be developed and deployed leveraging the underlying event structure, which provides an end-to-end approach to meeting application SLAs through administration, control and monitoring of single and multi-system environments

Teradata Viewpoint Portlets and Views

Viewpoint TMS monitor multiple clusters. Viewpoint VMS monitors a single cluster at a time.

Viewpoint portlets can be used to monitor Hadoop if you add Hadoop systems and enable data collectors in the Monitored Systems portlet and configure permissions in the Roles Manager portlet. You can send alerts and SNMP traps if you configure them in the Alert Setup portlet. The collected data from Hadoop generates a variety of charts:

- Node-level metrics about various node utilization and the status of services for each node in the system
- Hardware metrics, such as CPU usage, memory usage, and network activity
- Trend reporting metrics, for example, CPU, disk, and memory statistics

The following Viewpoint portlets provide the Hadoop information:

- Alert Viewer
- Hadoop Services
- Metric Heatmap
- Metrics Analysis
- Node Monitor
- Space Usage
- System Health

Viewpoint Dashboard and Viewpoint Mobile show the following Hadoop information:

- Alerts view
- System Health view
- Trends view

Viewpoint Dashboard shows the following Hadoop information:

- Hadoop Services view

For more information, see *Teradata Viewpoint User Guide*, B035-2206.

LAN-based Network Connectivity

The Hadoop cluster has multiple network connections. Each node in the cluster has a minimum of three physical connections:

Server Management

1GbE primary and secondary connections

InfiniBand

40Gb/sec redundant connections used for the Hadoop cluster and connectivity to edge nodes

Public Interfaces

1GbE or 10GbE external interfaces used to access customer network resources and for inbound customer client requests to the cluster

Hadoop communicates on the InfiniBand internal network. The corporate DNS database must include the fully-qualified Hadoop node names and Virtual IP (VIP) names.

Teradata installs and configures networking in the system, but the public interfaces can be customized.

For setting up external connectivity to a customer LAN or external edge node, see Setting Up External Access.

DNS Configuration

Hadoop nodes use a single name. The `/etc/hosts` file is used to resolve hostnames when communicating to nodes within its cluster. The name returned by the hostname -f command is used to resolve IP addresses. The Hadoop hostname must be the leftmost name on the line in `/etc/hosts` in order for Hadoop to recognize it. For example, a hosts file entry is similar to:

```
39.0.8.2     newName1
```

Note: Do not change hostnames for your site configuration after Hadoop installation as unexpected results could occur. For more information, contact your Teradata Customer Support representative.

Internal IP addresses are either manually configured and included in the hosts file or assigned by Server Management via DHCP and referenced via the CMIC configuration.

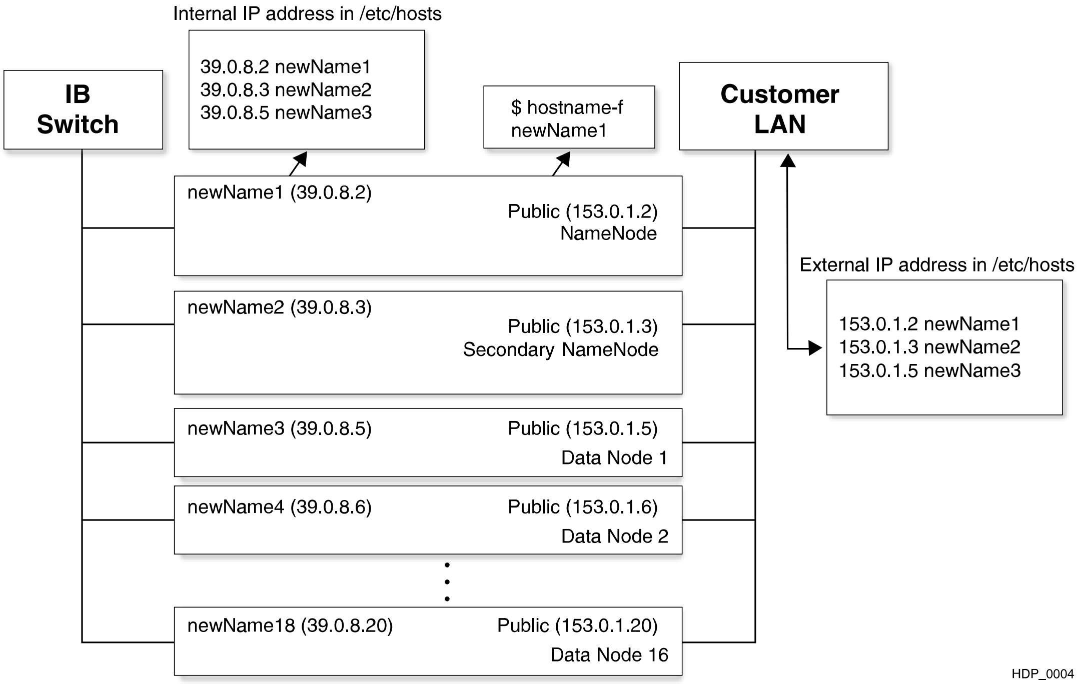
External interfaces and corporate DNS must use the same hostnames that Hadoop uses on the internal BYNET network.

If an application or user requests data from the namenode service, the service returns a location based on the hostname Hadoop is using. When external clients access Hadoop, the

Hadoop hostnames must be included in the corporate DNS or resolve the Hadoop hostname to its external interface via the local `/etc/hosts`.

The diagram illustrates the Hadoop cluster configured for external client access. The `/etc/hosts` file in each Hadoop node has hostnames associated with the internal BYNET network, while the external client's `/etc/hosts` file is associated with the Hadoop node's public IP.

Figure 1: Example of Networking Layout



Setting Up External Access

Prerequisite:

Ethernet cards are required on:

- All master, data, and edge nodes in the Hadoop cluster when the following apply:
  - Jobs are being submitted to the cluster from outside the InfiniBand network
  - Data is being loaded to or from Teradata Database and Aster Database systems that are not on the same InfiniBand network
  - A customer wants to monitor the cluster using a Viewpoint server located outside the InfiniBand network
- All edge nodes when the following apply:
  - Jobs are only being submitted to the cluster from inside the same InfiniBand network or only through the edge node
  - Data is not being loaded to or from Teradata Database and Aster Database systems outside the InfiniBand network

External hosts access Hadoop in one of two ways:

- For site-wide access, include the hostnames and IP addresses in your DNS configuration; DNS records need to match the hostnames returned by `hostname -f`, when run from a node for Hadoop.
- If the hosts are not included in DNS, external hosts can access the Hadoop cluster by including external IP addresses, like 153.64.1.1, in DNS or in a local `/etc/hosts` file.

1. Run `hostname -f` on each node in the Hadoop cluster to determine the Fully Qualified Domain Name (FQDN) including hostname and IP address to include in DNS or local hosts files.
2. Add entries to your DNS configuration for the nodes, or add entries in `/etc/hosts` to enable individual hosts to access the Hadoop cluster.

Virtual IP Addresses

The Oozie VIP is used for high availability of the Oozie server. If the server currently hosting the Oozie VIP fails, the VIP migrates to the other designated server for Oozie. This allows the migration to be transparent to the user or client. The client can continue using the same VIP hostname to access the working Oozie server. The Oozie VIP can be configured in the `/etc/opt/teradata/hadoop-tools/vip.cfg` configuration file.

Note: Oozie VIP works only in non-Kerberos-configured clusters.

CHAPTER 2

Managing HDP in the Teradata Environment

Accessing the Ambari Interface

When you log on to Ambari, you can access the service pages to find the associated interfaces for each service.

1 Go to `http://ambari:8081` where *ambari* is the FQDN of the node where Ambari is installed, for example, *http://hdp001-1.example.com:8081*.

2 Click the Services tab and select a service to see the management console for each service.

3 Click Quick Links and select an option to see the web interface for each related service.

Managing Users

The Ambari administrator can internally define permissions, group membership, and user access with Ambari or externally define permissions by importing the users and groups from a Lightweight Directory Access Protocol (LDAP) directory.

Changing Default Passwords

HDP is installed with default passwords and should be changed for your site security after installation and maintained according to your defined and published security policy.

1 Change the default Ambari password:

   a Log on to Ambari on the first master node, using `admin` as both the username and the password.

   b Click the Admin button and select Manage Ambari.

   c In the Manage Users + Groups window, select Users.

   d In the Users window, select admin.

   e In the Users/admin window, click Change Password.

   f Enter the current password and the new password twice, and click OK.

   g From master node one, run the following: `hcli --nolog support ambaripass <current password>`.

2 Change the Hive database password:

Notice: The Hive password default is `hive`. You must change the default password on the first master node as well as in Ambari.

a Log on as root on the first master node.

b Become the postgres user by typing `su - postgres`.

c Enter the postgres shell by typing `psql`.

d Change the password by entering the SQL statement `ALTER USER hive WITH PASSWORD '<newpassword>'`

3 Change the Hive password in Ambari:

a Log on to Ambari.

b In the Services list, select Hive.

c Select the **Configs** tab.

d Above the configuration metrics, select the Advanced tab.

e At the Database Password prompt, enter the new password twice, and click Save.

4 Change the Oozie password:

a Log on as root on the first master node.

b Become the postgres user by typing `su - postgres`.

c Enter the postgres shell by typing `psql`.

d Change the password by entering the SQL statement `ALTER USER oozie WITH PASSWORD '<newpassword>'`

5 Change the Oozie password in Ambari:

a Log on to Ambari.

b In the Services list, select Oozie.

c Select the **Configs** tab.

d At the Database Password prompt, enter the new password twice, and click Save.

6 Change the postgres password:

a Log on as root on the first master node.

b Become the postgres user by typing `su - postgres`.

c Enter the postgres shell by typing `psql`.

d Change the password by entering the SQL statement `ALTER USER postgres WITH PASSWORD '<newpassword>'`

7 On each node in the cluster, reset the linux user root password by entering `passwd root`.

The default password is `root`.

8 On each node in the cluster reset the linux tdatuser user password by entering `passwd tdatuser`.

The default password is `tdatuser`.

Enabling Kerberos

1 Run hcli system setup_kerberos, which provides values you must enter when running the wizard.

See system for the command syntax.

2 Run the Enable Kerberos wizard.

The output of the hcli command includes the URL for the Kerberos wizard.

Using Maintenance Mode to Disable Unused Services

The Teradata Appliance for Hadoop includes preinstalled services that may not be necessary for all customers. Putting unnecessary services into maintenance mode prevents alerts from starting automatically and sending alerts.

1 Log on to Ambari.

2 In the Services menu, select the service to disable.

3 In the Service Actions drop-down menu, select Turn On Maintenance Mode.

This prevents alerts from being generated.

4 In the Service Actions drop-down menu, select Stop.

5 In the **Confirmation** window, click **Confirm** Stop.

CHAPTER 3

Hadoop Command-Line Interface

Command-Line Interface Overview

Hadoop Command-Line Interface (hcli) is compatible with the Teradata installation of Hortonworks Hadoop. hcli is included in the `teradata-hadoop-tools` package and is installed in the `/opt/teradata/hadoop-tools` directory.

Administrative tasks are performed with hcli. hcli is functional even if the Hadoop cluster is down.

hcli Command Syntax

The following table lists annotations used to describe command syntax.

Object	Description	How used
|	Vertical bars	Separates alternative, mutually exclusive elements
[ ]	Square brackets	Indicates optional elements
bold	Boldface type	Indicates typed exactly as shown
italics	Italics type	Indicates elements for which you supply the values
{ }	Curly braces	Indicates grouped parameters

hcli Global Options

The following global options are available for hcli before each command:

Option	Description
`--yes`	Answers yes to all prompts, useful for running commands from a script
`--no`	Answers no to all prompts
`--verbose`	Provides detailed output. By default, only informational, warning, and error messages are output to console
`--quiet`	Provides only error message output
`--debug`	Provides detailed output for debugging purposes
`--nolog`	Provides no output to debugging logs. Use only with the support command to avoid recording passwords in log files.

Accessing the Hadoop Command-Line Interface

You can access hcli from the command shell when logged on as the root user, and hcli executes the command as the appropriate user. For example, commands related to the HDFS file system are actually run by the HDFS user, even though you are logged in as the root user. hcli can be run on any node in the cluster, although some commands only work on certain nodes. You can run the commands from any file system location without navigating to the installation directory.

1 Log on as root on any node.
For example, log onto a master node to run `hcli system restart --stale`
For example, log onto any node to run `hcli node info`

Accessing Hadoop Command Help

hcli help provides a list and brief description of all of the commands available in the Hadoop command-line interface. The basic syntax and usage is shown for each command. You can also specify a command name to view information about the parameters for that command.

1 To view help do one of the following:

View Options	Description
All commands	Run `hcli` to see all commands.
Command details	Run a specific command without parameters to see details. For example, run `hcli system`
Command parameter options	Run a command option with `--help`. For example, run `hcli system start --help`

config

Purpose

This command allows you to view and reset values in Hadoop configuration files.

The CONFIGSET should be 'hdfs-site', 'core-site', or any other configuration set managed by Ambari.

Syntax

```
hcli [--option] config [edit [--options] CONFIGSET | export [--options]
CONFIGSET | get CONFIGSET PROPNAME | import [--options] CONFIGSET |
listsaved | load [--options] [--backupindex] | retune | save [--
options] | set CONFIGSET PROPNAME VALUE | show [--option] | vip [--
option]]
```

Parameters

hcli **config** edit [OPTIONS] CONFIGSET

Edit complete configuration set from Ambari in JSON or XML format.

Notice: Importing an incomplete configuration file can render the cluster inoperable.

Note: Exit code is 0 if the file was imported successfully; exit code is 1 if the file could not be imported.

Option	Description
`--format=FORMAT, -f FORMAT`	Export to the specified format, either xml or json.
`--editor=EDITOR, -e EDITOR`	Use the specified editor or value of $EDITOR by default.

hcli **config** export [OPTIONS] CONFIGSET

Export complete configuration set from Ambari in JSON or XML format.

Note: Exit code is 0 if the file was exported successfully; exit code is 1 if the file could not be exported.

Option	Description
`--outputfile=FILE, -o FILE`	Export to the specified file.
`--format=FORMAT, -f FORMAT`	Export to the specified format, either xml or json.

hcli **config** get CONFIGSET PROPNAME

Extract a single property from a global Hadoop configuration file.

Note: The PROPNAME is the key from the key-value pairs in the configuration set.

Note: Exit code is 0 if the property was found; exit code is 1 if the value cannot be returned.

hcli **config** import [OPTIONS] CONFIGSET

Import complete configuration set to Ambari in JSON or XML format.

Notice: Importing an incomplete configuration file can render the cluster inoperable.

Note: Exit code is 0 if the file was imported successfully; exit code is 1 if the file could not be imported.

Option	Description
`--inputfile=FILE, -i FILE`	Export to the specified file.
`--format=FORMAT, -f FORMAT`	Export to the specified format, either xml or json.

hcli **config** listsaved

List all of the saved complete configuration set backups.

Note: Exit code is 0 if the configs listed correctly; exit code is 1 if the config cannot be listed.

hcli **config** load [options] [backupindex]

Restore the selected complete configuration set from backup.

Notice: Restoring an incomplete configuration file can render the cluster inoperable.

Note: The 'backupindex' should be an index number from the saved list of configuration backups.

Note: Exit code is 0 if the config was loaded successfully; exit code is 1 if the config could not be loaded.

Option	Description
`--local`	Load config from local file system with the specified 'backupindex'.
`--file`	Specify the location of the config backup archive.

hcli **config** retune

Apply factory default tuning and overwrite existing configuration settings.

Note: Before applying, a prompt requests backup of the existing configuration.

Note: Exit code 0 indicates the tuning was successfully completed; exit code 1 indicates the tuning could not be changed.

hcli **config** save [OPTIONS]

Save the complete configuration set.

Note: The exit code is 0 if the config was saved; the exit code is 1 if the config could not be saved.

Option	Description
`--label`	Specify the name for the config backup archive.

hcli **config** set CONFIGSET PROPNAME VALUE

Set a single property in a global Hadoop configuration file.

Note: The PROPNAME is the key from the key-value pairs in the configuration set. The property will be either overwritten or created, as necessary.

Note: The exit code is 0 if the property was set; the exit code is 1 if the value cannot be set.

Option	Description
`--label`	Specify the name for the config backup archive.

hcli **config** show [OPTIONS]

Export some Hadoop cluster configuration to stdout.

Note: The exit code is 0 if successful; the exit code is 1 if unsuccessful.

Option	Description
`--json`	Export Hadoop cluster configurations to `stdout` as a JSON dictionary
`--shell`	Produce bash-compatible output

hcli **config** vip [OPTIONS] SERVICENAME

Manipulate Virtual IP addresses in the cluster.

Note: Exit code is 0 if vip manipulation is successful; 1 if vip manipulation is unsuccessful.

Option	Description
`--give`	Release the Virtual IP address for the specified service on this host.
`--take`	Claim the Virtual IP address for the specified service on this host. The command will verify that the Virtual IP address is not currently in use elsewhere in the cluster, so the IP must be released on any other hosts before claiming on a new host.

Example: `hcli config vip --take oozie`

```
Virtual IP 39.64.8.64 claimed successfully.
```

node

Purpose

This command operates on any node in the cluster to control services and provide status.

Syntax

```
hcli [--option] node [info [--option] | runonall [--master | --data |
--edge] [--parallel] [--help] "cmd" [--option] | copytoall src
[additional src] --dest dir [--option] ]
```

Parameters

hcli node info [--*option*]

Display node information about the specified options.

Note: The exit code is 0 if the command is successful.

Option	Description
`--clustername`	Displays the name of the cluster.
`--node_type`	Displays the type of node, for example, a data node or master node.
`--hdp_version`	Displays the HDP version number.
`--teradata-hadoop-tools_version`	Displays the Hadoop tools version number.
`--json`	Outputs all the information in JSON format for cluster name, version numbers, node type.

Example: To display information for all components on the node, run `hcli node info`

```
+-------------------------------+-----------+
| Header                        | Value     |
+-------------------------------+-----------+
| cluster_name                  | c1        |
| teradata-hadoop-tools_version | 2.3.50.x  |
| hdp_version                   | 2.3       |
| node_type                     | master    |
+-------------------------------+-----------+
```

Example: To display the cluster name, run `hcli node info --clustername`.

```
c1
```

hcli node runonall [--master | --data | --edge] [--parallel] [--help] '*cmd*' [--*option*]

Execute a command on multiple Hadoop nodes. The command is required and must be in quotes. If no option is specified, the command runs on all nodes by default.

Option	Description
`-m | --master`	Run on master nodes
`-d | --data`	Run on data nodes
`-e | --edge`	Run on edge nodes
`-p | --parallel`	Run command on nodes in parallel

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the command was unsuccessful.

hcli node copytoall *src* [additional *src*] --dest *dir* [--*option*]

Copy a file or directory to multiple Hadoop nodes. If no option is specified, the file or directory copies to all nodes by default.

- *src* Names the source file or directory. Multiple allowed.
- -D | --dest *dir* Names the destination directory.
- -r | --recursive Copies recursively.

Option	Description
`-m | --master`	Copies file or directory to master nodes
`-d | --data`	Copies file or directory to data nodes
`-e | --edge`	Copies file or directory to edge nodes

Note: Exit code 0 indicates the command succeeded; other exit codes indicate the number of nodes on which the command failed.

Example: `hcli node copytoall /etc/hosts --dest /etc`

system

Purpose

This command adds nodes, implements failover, controls Hadoop services, or reports component version in the cluster.

Syntax

```
hcli [--option] system [addNode -n FQDN --type=type | failover | start
| stop | restart [--stale] | version [option] ]
```

Parameters

hcli system addNode -n *FQDN* --type=DATA | EDGE | MASTER [-h | --help]

Add a new node to the configuration.

- -n *FQDN* Name of the node and Fully Qualified Domain Name added to the configuration.
- --type=*type* Type of the node, data or edge, or master to be added to the configuration.
- -h | --help Show help for this command.

Note: Exit code 0 if adding the node was successful; exit code 1 if the node could not be added.

Example: `hcli system addNode -n HOSTNAME --type=DATA`

hcli system setup_kerberos [OPTIONS]

Important: Run this command before using the Enable Kerberos wizard.

The output of the command provides values you need to enter when using the wizard.

Configure local MIT KDC and prepare the Hadoop cluster for Kerberos enablement via this command and the Enable Kerberos wizard.

Executing this command will install the `krb5-server` package on Master Node #1 and install the `krb5-client` package on all nodes in the Hadoop cluster.

During the execution of this command the user will be prompted to set the password for the KDC database master key and the Kerberos admin principal `kadmin/admin@<REALM>`.

A principal and headless keytab will be created for the 'tdatuser' service account.

The local YARN usercache directories will be cleared.

During the execution of this command the user will be prompted to restart the cluster after making several configuration changes to various Hadoop services. This cluster restart is not mandatory, but if it is skipped, services will have stale configurations until they are restarted.

Option	Description
`-n | --node`	Optionally specify the Fully Qualified Domain Name of the host on which to install the MIT KDC. If not specified, the default host is Master Node #1.
`-r | --realm`	Optionally specify a custom Kerberos realm name. If not specified, the default realm name will be based on the cluster name.

Note:

- This command should be executed from Master Node #1.
- This command cannot be executed if Kerberos is already enabled on the cluster.

- Exit code 0 indicates the Kerberos setup succeeded; exit code 1 indicates the setup failed.

Once the command is complete, run the Enable Kerberos wizard.

hcli system start | stop | restart [--stale] [OPTION]

Start, stop, or restart cluster services using Ambari.

- `start` Starts the cluster services using Ambari
- `stop` Stops the cluster services using Ambari
- `restart [--stale]` Restarts the cluster services using Ambari. The stale option restarts only services with stale configurations.

Option	Description
`--yes`	Automatically answer 'yes' to all confirmation prompts
`--no`	Automatically answer 'no' to all confirmation prompts
`--help`	Show help for this command

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the command failed.

Example: Restart stale services in the cluster and show verbose output.

```
hcli --verbose system restart --stale
```

hcli system version [hdp | ambari | tools | tdh]

Display component version information.

Options	Description
None	Displays version information for all tools
`ambari`	Displays version of the Ambari server
`hdp`	Displays version of the Hadoop component only
`tdh`	Displays information for the Teradata Hadoop Tools only
`tools`	Displays information for the Hadoop Tools component only

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the command failed.

Example: To show all the component versions, run `hcli system version`

```
+----------------+-----------+
| Name                  | Version   |
+-----------------------+-----------+
| HDP                   | 2.3       |
| AMBARI_SERVER         | 2.1.2     |
| teradata-hadoop-tools | 2.3.55.43 |
| teradata-hdp-release  | 2.3.2.0   |
+-----------------------+-----------+
```

support

Purpose

This command allows you to report disk status, configure disks after replacement, install files, and set passwords to match Ambari and PostgreSQL.

Syntax

```
hcli [--option] support [checkDisks [--option] ] | [configureDisk [--
option]] | [install rpm name | patch name [--option]] | [manager_config
[--option]] | [rebalance [--option]] | [tune_coresidence] [--option]]]
| [[--nolog] support [setpgpass password ]
```

Parameters

hcli support checkDisks [OPTIONS]

Report status of disk drives in the cluster.

Option	Description
`--node`	Reports disks from the local node only (default on datanodes)
`--system`	Reports disks from the entire cluster (default on master nodes)
`--json`	Reports all data in JSON format
`--failed`	Reports only disks with abnormal status

Note: Exit code 0 indicates the command succeeded; exit code 1 if the report failed.

hcli support **configureDisk** [--check]

Configure, format, and prepare data node disk configurations for use after a disk drive replacement.

- Create vdisks on RAID controller
- Create filesystem and disk label if not present
- Update mount options in `/etc/fstab`
- Mount data drive filesystems

All Hadoop services on the node must be stopped with Ambari before using this command. Services can be started after the command completes with no errors.

Note: Use the --check option to check that the data node disks are correctly configured and labeled. No configuration changes will be made.

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the number of nodes on which the command failed.

hcli support install --rpm *name* | --patch *name* [--master | --data | --edge | --all | --dry-run]

Install RPMs and patches on nodes in the cluster. Either the RPM and name or the patch and name are required.

- -r | --rpm *name* Installs the specified RPM file.
- -p | --patch *name* Installs the specified patch file.

Option	Description
`-m | --master`	Installs on master nodes

Option	Description
`-d | --data`	Installs on data nodes
`-e | --edge`	Installs on edge nodes
`-a | --all`	Installs on all nodes
`-n | --dry-run`	Tests the installation but does not make changes

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the command failed.

Example: `hcli support install --rpm /tmp/new-package-file.rpm --master --edge`

hcli support **manager_config** [OPTION]

Modify various Ambari related settings.

Option	Description
`-p | --port` *port number*	Update the port number used to connect to Ambari.
`-h | --host=`*host fqdn*	Update the hostname used to connect to Ambari.
`-p | --password=`*admin password*	Update the Ambari administrator password.

Notice: When using the `--password` option, use the `--nolog` option to avoid recording passwords in debugging logs.

Note: Exit code 0 indicates the Ambari setting updated successfully; exit code 1 indicates the setting could not be updated.

Example: hcli --nolog support manager_config --password=*admin password*

hcli support rebalance

Adjust parameters to enable high-priority HDFS rebalancing after expanding or contracting a cluster.

Note: Waits until rebalance is complete before exiting.

Note: Exit code 0 indicates the rebalance succeeded; exit code 1 if the HDFS could not be rebalanced.

hcli [--nolog] support setpgpass *password*

Change the password on postgres and update `postgres/.pgpass` files on the postgres nodes.

Notice: Use the `--nolog` option to avoid recording passwords in debugging logs.

Note: Exit code 0 indicates the command succeeded; exit code 1 indicates the command failed.

Example: `hcli --nolog support setpgpass pswd`

hcli support tune_coresidence [--node | --help]

Tune data nodes in the cluster for coresidence configuration.

Option	Description
`-n | --node` *FQDN*	Specifies the fully-qualified domain name of a single node that is tuned to the coresidence configuration.
`-n` *FQDN-NODE1* `-n` *FQDN-NODE2*	Specifies the fully-qualified domain names of all nodes that are tuned to the coresidence configuration.
`-h | --help`	Shows help for this command.

Note: Exit code 0 indicates that the tuning for coresidence configuration has completed; exit code 1 indicates an exception has occurred.

Example: `hcli support tune_coresidence`

Example: `hcli support tune_coresidence --node <FQDN> [-h|--help]`

Example: `hcli support tune_coresidence -n <FQDN-NODE1> -n <FQDN-NODE2> [-h|--help]`

test

Purpose

The command runs service checks and performance tests on the cluster.

Syntax

```
hcli [--option] test | [flume [--options] SIZE] | [test ingest [--options]] [ smoke [--service] | terasort size[K |M |G |T] | testdfsio size[K |M |G |T]]
```

Parameters

hcli test **flume** [--options] *size*[K |M |G |T]

Run flume test with various file sizes.

- K - kilobytes
- M - megabytes
- G - gigabytes
- T - terabytes

Note: Exit code is 0 if the test completed successfully; exit code is 1 if the test failed.

Option	Description
`--hdfs`	Dump flume records to HDFS
`--checkpoint arg`	Checkpoint after every *arg* record
`--timeout arg`	Fail if test takes longer than *arg* seconds
`--batchsize arg`	Use flume batch size of *arg* records

Example: `hcl test flume --hdfs 10M`

hcli test ingest [--options]

Perform a parallel load or unload test of dummy data into Hadoop for the specified time.

Note: Exit code is 0 if the test completed successfully; exit code is 1 if the test failed.

Option	Description
`--block-size=`*size*	Buffer block size for loan/unload threads
`--file-size=`*size*	Bytes to write in each file before rolling over to a new file
`--time=`*secs* | `-t` *secs*	Number of seconds to measure load/unload performance
`--ramp=`*secs* | `-r` *secs*	Number of seconds to start load/unload before starting measurement
`--dfs-block-size=`*size*	Override DFS block size (must be a factor of 2)
`--threads=`*N*	Use the specified number of threads

Example: `hcli test ingest -t 3600 --threads=24`

hcli test smoke [*service*]

Run basic service checks for Ambari. Results are output as they occur and as a table of service check results.

- No service listed. Run service checks for all installed services.
- `NAMENODE RESOURCEMANAGER` Run service checks only for the Namenode and Resource Manager services.

Example: Run service checks for only the Namenode and Resource Manager services:

```
hcli test smoke NAMENODE RESOURCEMANAGER
```

hcli test terasort *size*[K |M |G |T]

Run the Hadoop Teragen, Terasort, and Teravalidate of specified size

- K - kilobytes
- M - megabytes
- G - gigabytes
- T - terabytes

Example: `hcli test terasort 2T`

hcli test testdfsio *size*[K |M |G |T]

Run Hadoop DFSIO test with various file sizes.

- K - kilobytes
- M - megabytes
- G - gigabytes
- T - terabytes

Note: Exit code is 0 if the test completed successfully; exit code is 1 if the test failed.

Example: `hcli test testdfsio 100M`

CHAPTER 4

Reference Information

Log File Locations

- HCLI logs are located in `/var/opt/teradata/hadoop-tools/logs/`
- HDP and Ambari logs are located in `/var/opt/teradata/log/`*`service/component`* for the following services:
  - Ambari Metrics
  - Falcon
  - Flume
  - HBase
  - HDFS
  - Hive
  - MapReduce2
  - Oozie
  - WebHCat
  - YARN
  - ZooKeeper

Service Ports

Service	Port
Accumulo	
Accumulo GC	50091
Accumulo Monitor	4560
HTTP Monitor	50095
Master Server	9999
Proxy Server	42424
Shutdown	4445
Tablet Server	9997
Tracer	12234
Ambari	
Agent	8440

Service	Port
Agent (secure)	8441
Ambari Agent Ping	8670
Metrics RPC	60200
Metrics Server Webapp	6188
Server UI/ REST API	8081
Server UI/ REST APlr (secured)	8181
Falcon	
Application Port	15000
Application Port (secure)	15443
EmbeddedMQ Port	61616
HBase	
HBase Master	60000
HBase Master Thrift	8080
HBase Master Thrift UI	8085
HBase Regionserver Thrift	9090
HBase Regionserver Thrift UI	9095
Master Web UI	60010
Regionserver	60020
Regionserver Web UI	60030
HDFS	
Checkpoint RPC	50100
Checkpoint Web UI	50105
Datanode data transfer	50010
Datanode metadata operations	50020
Datanode Web UI (http)	50075
Datanode Web UI (https)	50475
JournalNode RPC	8485
JournalNode Web UI	8480
JournalNode Web UI (secure)	8481
Namenode metadata	8020
Namenode RPC	8020
Namenode Service RPC	None

Service	Port
Namenode Web UI (http)	50070
Namenode Web UI (https)	50470
Secondary Namenode	50091
Secondary Namenode	50090
ZKFC	8019
Hive/ WebHcat	
Hiveserver2	10000
Hive Web UI	9999
Metastore	9083
WebHcat Web UI	50111
Kafka	
Alternate Advertised Port	None
Broker	9092
Knox	
Gateway	8442
LDAP Server	33389
MapReduce2	
HistoryServer2	19888
HistoryServer2 (secure)	19890
HistoryServer2 Admin API	10033
HistoryServer2 API	10020
Shuffle Port	13562
Oozie	
Oozie	11000
Oozie (secure)	11443
Oozie Admin Port	11001
Phoenix	
Query Server	8765
PostgreSQL	
Database Server	5432
Spark	
History Server	18080

Service	Port
Storm	
DRPC Invocations Port	3773
DRPC Port	3772
Logviewer Port	8000
Nimbus Thrift Port	6627
Supervisor Slots Ports	6700, 6701, 6702, 6703
UI Port	8080
Web UI	8744
YARN	
Application History Server	8188
Application Timeline Server	10200
NodeManager	45454
NodeManager Localizer	8040
NodeManager Web UI	8042
NodeManager Web UI (secure)	8044
ResourceManager	8025
ResourceManager	8032
ResourceManager	8050
ResourceManager	8141
Resource Manager Admin	8033
ResourceManager Resource Tracker	8031
ResourceManager Scheduler	8030
ResourceManager Web UI	8088
ResourceManager Web UI (secure)	8090
Zookeeper	
UI	4040
ZK Client	2181
ZK Leader Election	3888
ZK Peer Connection	2888