

Table of Contents

Table of Contents	1
Pivotal Greenplum Command Center Documentation	2
About Pivotal Greenplum Command Center	3
Installing the Greenplum Command Center Software	6
Downloading and Running the Greenplum Command Center Installer	7
Setting the Greenplum Command Center Environment	9
Creating the gpperfmon Database	10
Upgrading Greenplum Command Center	12
Uninstalling Greenplum Command Center	14
Creating Greenplum Command Center Console Instances	15
Greenplum Command Center User Guide	18
Connecting to the Greenplum Command Center Console	19
Dashboard	20
Query Monitor	22
Host Metrics	24
Cluster Metrics	25
Monitoring Multiple Greenplum Database Clusters	26
History	27
System	29
Segment Status	30
Storage Status	32
Admin	33
Permission Levels for GPCC Access	34
Authentication	35
Workload Management	37
Administering Greenplum Command Center	41
About the Command Center Installation	42
Starting and Stopping Greenplum Command Center	43
Administering Command Center Agents	44
Administering the Command Center Database	45
Administering the Web Server	46
Configuring Greenplum Command Center	47
Enabling Multi-Cluster Support	48
Securing a Greenplum Command Center Console Instance	50
Configuring Authentication for the Command Center Console	52
Enabling Authentication with Kerberos	54
Securing the gpmon Database User	59
Utility Reference	61
gpcmdr	62
gpccinstall	64
Configuration File Reference	65
Command Center Agent Parameters	66
Command Center Console Parameters	68
Setup Configuration File	69
Greenplum Database Server Configuration Parameters	71

Pivotal Greenplum Command Center Documentation

Documentation for Pivotal Greenplum Command Center.

- [About Greenplum Command Center](#)
Pivotal Greenplum Command Center is a management tool for the Greenplum Big Data Platform. This section introduces key concepts about Greenplum Command Center and its components.
- [Installing the Greenplum Command Center Software](#)
Downloading and installing the Greenplum Command Center software in your Greenplum cluster and one-time tasks. This section also provides instructions for upgrading Command Center to a new release and migrating Command Center instances.
- [Creating Greenplum Command Center Console Instances](#)
Creating a Command Center Console instance to manage a Greenplum cluster.
- [Using the Greenplum Command Center Web Interface](#)
Using the Greenplum Command Center web user interface to monitor and manage a Greenplum cluster.
- [Administering Greenplum Command Center](#)
System administration information for the Greenplum Command Center components.
- [Utility Reference](#)
Reference information for the two Greenplum Command Center utility programs: the `gpperformon_install` utility that enables the data collection agents and the `gpcmdr` utility that sets up and manages the web application.
- [Configuration File Reference](#)
Reference information for configuration files used with Greenplum Command Center.

About Pivotal Greenplum Command Center

Pivotal Greenplum Command Center is a management tool for the Greenplum Big Data Platform. This section introduces key concepts about Greenplum Command Center and its components.

Introduction

Greenplum Command Center monitors system performance metrics, analyzes system health, and allows administrators to perform some management tasks in a Greenplum environment. The Greenplum Command Center Console is an interactive graphical web application that is installed on a web server, usually on the master host. Users view and interact with the collected Greenplum system data through this application.

Greenplum Command Center is comprised of data collection agents that run on the master host and each segment host. The agents collect data about queries and system utilization and update the Greenplum master host at regular intervals. Greenplum Command Center stores its data and metrics in a dedicated Greenplum database (the Command Center database, `gpcc`) whose information is distributed among the master host and segment hosts like any other Greenplum Database. You can access the data stored in the Command Center database through the Greenplum Command Center Console and through SQL queries.

Note: Command Center requires Greenplum Database to operate because Command Center stores its information in a Greenplum database.

Supported Greenplum Platforms

Greenplum Command Center is currently certified for the Greenplum Data Computing Appliance (DCA) and Greenplum Database software-only environments. Command Center monitors the following for each environment:

Greenplum Data Computing Alliance:

- Greenplum Database Module

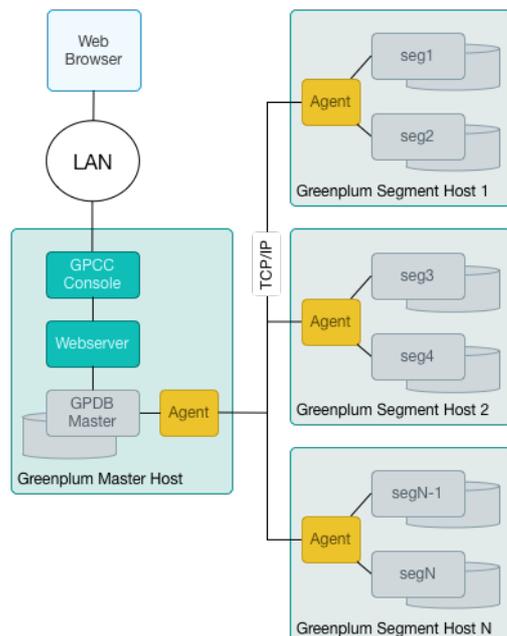
Greenplum Database (software-only environments):

- Greenplum Database

See the *Release Notes* for your Greenplum Command Center release for information about the supported software and hardware versions.

Architecture

The following figure illustrates the Greenplum Command Center architecture.



Greenplum Data Collection Agents

Greenplum data collection agents run on Greenplum segment hosts to collect the query and system statistics to be displayed in the Command Center. The agents are installed with the Greenplum Database distribution but are not enabled until you create the Greenplum Command Center database (gpperfmon) and start them. The agents may also be enabled or disabled by setting the `gp_enable_gpperfmon` server configuration parameter. When this parameter is enabled, the data collection agents run on all Greenplum hosts (master and segments), and start and stop along with Greenplum Database server processes.

The master agent polls all segment agents for system metrics and other data at a configurable interval (called the quantum). The master agent amasses the data from all segments, stores it in flat files, and periodically commits the data in the files to the Greenplum Command Center database.

Greenplum Command Center Database

The Greenplum Command Center database (gpperfmon) is a database within your Greenplum system dedicated to storing and serving system data. Your Greenplum Database installation includes the `gpperfmon_install` utility to install the Command Center database and optionally start the data collection agents.

When this document refers to the Command Center database, it is referring to the database named gpperfmon.

Greenplum administrators can connect to the Command Center database using client programs such as psql or application programming interfaces (APIs) such as JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity). Administrators can also use the Greenplum Command Center Console to view reports on current and historical performance and perform other management tasks.

The Command Center database consists of three sets of tables; *now* tables store data on current system metrics such as active queries, *history* tables store data on historical metrics, and *tail* tables are for data in transition. Tail tables are for internal use only and should not be queried by users. The now and tail data are stored as text files on the master host file system, and the Command Center database accesses them via external tables. The history tables are regular database tables stored within the Command Center (gpperfmon) database. See [gpperfmon Database Reference](#) for the schema definitions of these tables.

Greenplum Command Center Console

Greenplum Command Center provides a browser-native HTML5 graphical console for viewing Greenplum Database system metrics and performing certain database administrative tasks. This browser-based application provides the following functionality:

- Interactive overview of realtime system metrics
- Detailed realtime statistics for the cluster and by server
- Query Monitor view lists queries executing and waiting to execute
- Four permission levels to allow users to view or cancel their own or others' queries, and to view or manage administrative information
- Cluster Metrics view shows synchronized charts of historical system metrics
- History view lists completed queries and system metrics plotted over a selected time period
 - Select a query to view query text and explain plan
- Segment Status view with summaries and details by segment
- Storage Status view with summaries and details by segment data directory
- Admin > Permissions view to see or manage permission levels
- Admin > Authentication view to see or edit the `pg_hba.conf` host-based authentication configuration file

If you have multiple Greenplum environments, you can create separate Command Center instances for them. Each separate console instance operates on a unique port and has its own unique configuration options. For more information, see [Creating Greenplum Command Center Console Instances](#).

Greenplum Command Center Web Service

The Greenplum Command Center Console queries the Command Center database through a web service framework composed of a lightweight go-based web server.

The console setup utility, `gpcmdr`, sets up the web server and web service, prompting you for basic configuration information on the desired port and SSL options. Under normal conditions, the web server and web service API require minimal maintenance and administration, as described in [Web Server](#)

[Administration.](#)

Installing the Greenplum Command Center Software

 The Greenplum Workload Manager installer is included in the Greenplum Command Center installer. Install Workload Manager using its bundled installer. See the [Greenplum Workload Manager documentation](#) for instructions to run the Greenplum Workload Manager installer.

Installation Notes

Greenplum Command Center may be installed on a Dell EMC Data Computing Appliance (DCA) or in a Greenplum Database software environment.

Command Center is compatible with the Data Computing Appliance (DCA), versions 1.2.x, 2.x, and 3.x. Visit [Pivotal Network](#) to download an installer for the most recent 3.x version for your DCA. See the Dell EMC DCA documentation for information about installing the software on your appliance.

The Greenplum Command Center software is typically installed on the Greenplum Database master host. Installing on the master host provides the best performance and security, since the Command Center Console database requests are not passed over the network.

If you are installing the Command Center Console on a remote system (that is, not the same system on which you installed Greenplum Database), you must first install the Greenplum Database binary software files on the remote system. Note that you do not need to initialize the database on the remote system. See the *Greenplum Database Installation Guide* for help installing the Greenplum Database software.

This section contains the following topics:

- [Downloading and Running the Greenplum Command Center Installer](#)
- [Setting the Greenplum Command Center Environment](#)
- [Creating the gpperfmon Database](#)
- [Upgrading the Greenplum Command Center Software](#)
- [Uninstalling Greenplum Command Center](#)

Downloading and Running the Greenplum Command Center Installer

Download and Run the Installer

Important: The `gpadmin` user owns and executes the Greenplum Command Center software, which is installed in the `/usr/local` directory by default. Before you begin installing, ensure that the `gpadmin` user has write permission in the directory where you will install the software. Be sure to prepare the standby master host in the same way. Alternatively, you can run the installer as `root` and, after installation, change the owner of the installation directory and its contents to `gpadmin`.

Follow these steps as `gpadmin` to download and install the Greenplum Command Center software.

1. Download the Greenplum Command Center installer file from the Greenplum Database section of [Pivotal Network](#). Installer files are available for Linux 64-bit platforms.

2. Unzip the installer file.

```
$ unzip greenplum-cc-web-3.X.X-LINUX-x86_64.zip
```

3. Launch the installer with `bash`:

```
$ /bin/bash greenplum-cc-web-3.X.X-LINUX.bin
```

4. Read through the license agreement. When you reach the bottom, type `yes` to accept the license agreement.

5. The installer prompts you to provide an installation path. Press **Enter** to accept the default installation path (`/usr/local/greenplum-cc-web-X.X.X`), or enter an absolute path to another install location. You must have write permission in the location you specify.

6. The installer then asks if you want to install on the standby master.

- Enter `no` if you do not want to install the software to the standby master host now. You can install the software to the standby master later using the `gpccinstall` utility. See [Install Greenplum Command Center Software on Additional Hosts](#).
- Enter `yes` to install the Greenplum Command Center software on the standby master host, then enter the name of the host.

7. The installer completes with a summary of the actions that were performed.

Note:

If you have performed the previous steps as any user other than `gpadmin`, you need to change ownership and permissions of the installation directory before you continue.

Change the ownership of the installation directory:

```
# chown -R gpadmin:gpadmin greenplum-cc-web-X.X.X
```

Change the permissions of the installation directory:

```
# chmod -R 755 greenplum-cc-web-X.X.X
```

Install Greenplum Command Center Software on Additional Hosts

Follow the steps in this section to install the Greenplum Command Center software on the standby master host or on other hosts where you want to run Greenplum Command Center console instances.

Ensure that the host where you want to install Command Center meets these requirements:

- The `gpadmin` user has been created and has passwordless SSH access to the Greenplum cluster. See `gpssh-exkeys` in the *Greenplum Database Utility Reference Guide*.
- The directory where the Command Center software is to be installed has been created and is writable by the `gpadmin` user. The path to this directory must be identical to the path where the Command Center software is installed on the Greenplum master host. The default is `/usr/local`.
- The `~gpadmin/.pgpass` file must be copied from the Greenplum master host. Make sure the `gpadmin` user owns the file and has exclusive read/write

permissions (`chmod 600 ~gpadmin/.pgpass`).

- The Greenplum Database authorization file (`$MASTER_DATA_DIRECTORY/pg_hba.conf`) on the Greenplum Database master and standby hosts must contain entries to allow the gpmon database user, and any other user permitted to use Command Center, to access the gpperfmon database.

Run the `gpccinstall` utility as the `gpadmin` user on the host where you installed the Greenplum Command Center software.

1. Create a text file containing the names of the standby master host or other hosts where the software is to be installed, one name per line. Do not include the name of the host where you ran the installer. Hostnames must be resolvable in DNS. For example:

```
smdw
gpcc
```

2. Source the Greenplum Database and Command Center path files.

```
$ source /usr/local/greenplum-db/greenplum_path.sh
$ source /usr/local/greenplum-cc-web/gpcc_path.sh
```

3. As `gpadmin`, run the `gpccinstall` utility to install Command Center on all hosts listed in the host file you created.

```
$ gpccinstall -f <hostfilename>
```

where `hostfilename` is the name of the host file you created.

When the installation is complete you can log in to the remote host as `gpadmin`, source the `gpcc_path.sh` file in the Command Center installation directory, and set up and manage Command Center instances using the `gpcmdr` utility. See [Creating Greenplum Command Center Console Instances](#) for instructions to create and start a Command Center instance.

Setting the Greenplum Command Center Environment

To enable the `gpadmin` user to execute Command Center utilities such as `gpccmdr`, follow these steps to set up the Greenplum Command Center environment.

1. Add the `GPPERFMONHOME` environment variable to your startup shell profile (such as `~/bashrc`). Set the variable to the Greenplum Command Center home directory.

```
GPPERFMONHOME=/usr/local/greenplum-cc-web  
source $GPPERFMONHOME/gpcc_path.sh
```

Ensure that the `$GPPERFMONHOME/gpcc_path.sh` file has entries for the `greenplum_path.sh` file and the `MASTER_DATA_DIRECTORY` environment variable. See the *Greenplum Database Installation Guide* for details.

2. Save and source the `.bashrc` file:

```
$ source ~/.bashrc
```

Creating the gpperfmon Database

This topic describes how to create the Command Center `gpperfmon` database and enable the data collection agents. This task must be completed one time for the Greenplum Database system, before you create a Greenplum Command Center instance.

When the data collection agents are enabled, their processes are started and stopped (using `gpstart` and `gpstop`) on the Greenplum segment hosts along with the Greenplum Database server processes.

Greenplum provides a `gpperfmon_install` utility that performs the following tasks:

- Creates the Command Center database (`gpperfmon`).
- Creates the Command Center superuser role (`gpmon`).
- Configures Greenplum Database server to accept connections from the `gpmon` role (edits the `pg_hba.conf` and `.pgpass` files).
- Sets the Command Center server configuration parameters in the Greenplum Database server `postgresql.conf` files.

The `gpperfmon_install` utility and the agents are part of the Greenplum Database software. The tasks in this topic can be performed before or after the Command Center software is installed.

Enabling the Collection Agents

1. Log in to the Greenplum master host as the `gpadmin` user.

```
$ su - gpadmin
```

2. Source the path file from the Greenplum Database installation directory:

```
# source /usr/local/greenplum-db/greenplum_path.sh
```

3. Run the `gpperfmon_install` utility with the `--enable` option. You must supply the connection port of the Greenplum Database master server process, and set the password for the `gpmon` superuser that will be created. For example:

```
$ gpperfmon_install --enable --password changeme --port 5432
```

Note:

The `gpperfmon_install` utility creates entries for the `gpmon` user in the `$MASTER_DATABASE/pg_hba.conf` file. See [gpmon User Authentication](#) for notes about restricting the `gpmon` user's access to databases.

The password you specify is saved in a `.pgpass` file in the `gpadmin` user's home directory. See [Changing the gpmon Password](#) for steps to change the `gpmon` password.

4. When the utility completes, restart Greenplum Database server. The data collection agents do not start until the database is restarted.

```
$ gpstop -r
```

5. Using the `ps` command, verify that the data collection process is running on the Greenplum master. For example:

```
$ ps -ef | grep gpmmmon
```

6. Run the following command to verify that the `gpmon` user can authenticate and that the data collection processes are writing to the Command Center database. If all of the segment data collection agents are running, you should see one row per segment host.

```
$ psql -U gpmon gpperfmon -c 'SELECT * FROM system_now;'
```

The data collection agents are now running, and your Greenplum system now has a `gpperfmon` database installed. This is the database where Command Center data is stored. You can connect to it as follows:

```
$ psql gpperfmon
```

Configuring a Standby Master Host (if enabled)

1. Copy the `$MASTER_DATA_DIRECTORY/pg_hba.conf` file from your primary master host to your standby master host. This ensures that the required connection options are also set on the standby master.
2. Copy your `~/.pgpass` file from your primary master host to your standby master host. This file usually resides in the `gpadmin` user's home directory. Note that the permissions on `.pgpass` must be set to 600 (for example: `chmod 0600 ~/.pgpass`).

gpmon User Authentication

The `gpperfmon_install` utility adds the `gpmon` user to the `pg_hba.conf` authorization configuration file entries allowing local connections to any database in the Greenplum cluster. Greenplum Command Center requires that the `gpmon` user have access to `gpperfmon` and every database that Command Center will monitor. Since the `gpmon` role is a Greenplum Database superuser, you may wish to restrict the role from accessing other databases. Edit the `$MASTER_DATA_DIRECTORY/pg_hba.conf` and edit these lines:

```
local gpperfmon gpmon md5
host all gpmon 127.0.0.1/28 md5
host all gpmon ::1/128 md5
```

List `gpperfmon` and the databases you want to monitor with Command Center in the second field:

```
local gpperfmon,userdb1,userdb2 gpmon md5
host gpperfmon,userdb1,userdb2 gpmon 127.0.0.1/28 md5
host gpperfmon,userdb1,userdb2 gpmon ::1/128 md5
```

See [Changing the gpmon Password](#) for steps to change the `gpmon` user's password.

Upgrading Greenplum Command Center

This section provides steps for upgrading Pivotal Greenplum Command Center to a new version.

Upgrading Greenplum Command Center requires installing the new distribution, and then migrating Command Center instances from a previous installation.

A new Greenplum Command Center software release may be installed in the same parent directory as the current release, by default `/usr/local`. The installer updates the symbolic link `greenplum-cc-web` to point to the new release directory and leaves the old release directory in place. After the software is installed, run the `gpcmdr --migrate` command to recreate your Command Center instances.

Install the New Software Release

1. Log in as the `gpadmin` user.
2. Source the `greenplum_path.sh` and `gpcc_path.sh` files from the current release:

```
$ source /usr/local/greenplum-db/greenplum_path.sh
$ source /usr/local/greenplum-cc-web/gpcc_path.sh
```

3. Download the latest Command Center release from [Pivotal Network](#). Installer files are available for Linux 64-bit platforms, and have names in the format:

```
greenplum-cc-web-X.X.X-PLATFORM.zip
```

4. Unzip the installer file. For example:

```
# unzip greenplum-cc-web-X.X.X-PLATFORM.zip
```

5. Launch the installer for the new release with the bash shell:

```
$ /bin/bash greenplum-cc-web-X.X.X-PLATFORM.bin
```

Note: The installer requires write permission in the installation directory (`/usr/local`, by default). If the `gpadmin` user does not have write permission in the installation directory, run the installation as `root`. You will need to change file ownership and permissions after the software is installed.

6. Read through the license agreement. When you reach the bottom, type `yes` to accept the license agreement.
7. The installer prompts you to provide an installation path. Enter a full path or press **ENTER** to accept the default, `/usr/local`. You must have write permission in the directory you specify.
8. If you ran the installation as `root` or any user other than `gpadmin`, change the ownership and permissions of the installation directory:

```
# chown -R gpadmin:gpadmin /usr/local/greenplum-cc-web-versionx.x
# chmod -R 755 /usr/local/greenplum-cc-web-versionx.x
```

Change to the `gpadmin` user before you continue to the next step:

```
# su - gpadmin
```

9. Ensure that you have a current host file listing the names of all of the other hosts participating in the Greenplum Database cluster, including the standby master host. The host names must be resolvable in DNS.
10. As `gpadmin`, run the `gpccinstall` utility to install the new Command Center files on all hosts listed in the host file:

```
$ gpccinstall -f hostfilename
```

where `hostfilename` is the name of the host file you created.

Migrate Command Center Instances

After the new Command Center software is installed, migrate your instances by running the `gpcmdr --migrate` command.

To migrate all instances from a previous installation, run `gpcmdr --migrate` with no arguments. For example:

```
$ gpcmdr --migrate
```

To migrate a single instance, run `gpcmdr --migrate <instance_name>` and, when prompted, provide its full installation path. For example:

```
$ gpcmdr --migrate myinstance
```

Custom changes to the `ssh-wrapper` file are not handled by instance migration. See [Configuration File Reference](#) for information about the `ssh-wrapper` file. If you have set a custom `ssh` path in this file, you must copy it to the current installation.

For example:

```
cp /usr/local/greenplum-cc-web-3.0.0/bin/ssh-wrapper /usr/local/greenplum-cc-web/bin
```

See [Creating Greenplum Command Center Console Instances](#) for instructions to create new instances with `gpcmdr --setup`.

Uninstalling Greenplum Command Center

To uninstall Greenplum Command Center, you must stop both the Command Center Console and disable the data collection agents. Optionally, you may also remove any data associated with Greenplum Command Center by removing your Command Center Console installation and the Command Center database.

1. Stop Command Center Console if it is currently running. For example:

```
$ gpccdr --stop
```

2. Remove the Command Center installation directory from all hosts. For example:

```
$ rm -rf /usr/local/greenplum-cc-web-version
```

3. Disable the Data Collection Agents.

- a. Log in to the master host as the Greenplum administrative user (`gpadmin`):

```
$ su - gpadmin
```

- b. Edit the `$MASTER_DATA_DIRECTORY/postgresql.conf` file and disable the data collection agents:

```
gp_enable_gpperfmon = off
```

- c. Remove or comment out the gpmon entries in `pg_hba.conf`. For example:

```
#local  gpperfmon  gpmon  md5
#host   gpperfmon  gpmon  0.0.0.0/0  md5
```

- d. Drop the Command Center superuser role from the database. For example:

```
$ psql template1 -c 'DROP ROLE gpmon;'
```

- e. Restart the Greenplum Database instance:

```
$ gpstop -r
```

- f. Clean up any uncommitted Command Center data and log files that reside on the master file system:

```
$ rm -rf $MASTER_DATA_DIRECTORY/gpperfmon/data/*
$ rm -rf $MASTER_DATA_DIRECTORY/gpperfmon/logs/*
```

- g. If you do not want to keep your historical Command Center data, drop the gpperfmon database:

```
$ dropdb gpperfmon
```

Creating Greenplum Command Center Console Instances

A Command Center Console instance is a web server providing an HTML5 graphical console application to monitor system metrics and perform some administrative tasks for a single Greenplum Database cluster.

The Command Center Console runs on the gpmonws web server. The default web server port is 28080. Configuration files, log files, and runtime files for each Command Center instance are managed in a subdirectory of the `$GPPERFMONHOME/instances` directory.

If you have multiple Greenplum Database instances, you can create separate Command Center Console instances for each of them. Each separate console instance operates on a unique port and has its own unique configuration options. A multi-cluster view may be enabled to allow you to view status for all clusters. See [Enabling Multi-Cluster Support](#) for more information.

For more information about the web server, see [Web Server Administration](#).

The Command Center Console supports current browser versions of Chrome, Safari, Firefox, and Internet Explorer.

Before You Begin

Ensure that the following prerequisites are satisfied:

- Greenplum Command Center software is installed. See [Install the Greenplum Command Center Software](#).
- The gpperfmon database is created and the data collection agents are running. See [Creating the gpperfmon Database](#).
- The `gpmon` user is created and is able to log in to the `gpperfmon` database (`psql -U gpmon gpperfmon`).
- Any certificates or Kerberos keytab files needed for encryption and user authentication are installed. These files must be readable by the gpadmin user. See [Securing a Greenplum Command Center Console Instance](#).
- If the `gpmon` user is to be authenticated with Kerberos, install the keytab file for the `gpmon` Kerberos principal on the Greenplum master and standby hosts and run `kinit gpmon` before you begin to create the Command Center Console instance. See [Securing the gpmon Database User](#) for more information.

The `gpcmdr --setup` command-line utility sets up the Command Center instance. The command can be run interactively, or you can create a configuration file and run the command non-interactively. If you use a configuration file, you can create multiple Command Center instances at once.

- [Creating an instance interactively](#)
- [Creating an instance with a configuration file](#)

Creating the Greenplum Command Center Instance (Interactive)

Follow the steps below to create a new Command Center Console instance. To accept the displayed default values for any parameters at configuration time, press the **ENTER** key. To monitor multiple Greenplum Database clusters, run the Setup utility separately to create an instance for each cluster.

1. Log in as the Greenplum administrator (`gpadmin`) and source the `<gpcc-install-dir>/gpcc_path.sh` file.
2. With the Greenplum Database instance running, launch Setup.

```
$ gpcmdr --setup
```

3. Setup prompts you to enter an instance name. The instance name can contain letters, digits, and underscores.
4. Setup prompts you to enter a display name for the instance. This name is displayed in the Command Center Console user interface to identify the instance. Enter a display name or press **ENTER** to use the instance name you entered in the previous step.
5. Setup asks if the Greenplum Database master host is remote. If the master instance is on a different host enter `y`, and then enter the master host name and port number at the next prompts. Be sure to enter the host name—not the IP number—if you want to use Greenplum Command Center’s Workload Manager features.
6. Setup asks if you want to enable Kerberos authentication for the new instance. To use this feature, Kerberos authentication must be enabled for Greenplum Database and the Kerberos administrator must have created a keytab file for Command Center. See [Enabling Kerberos Authentication with Greenplum Command Center](#) for details. If you enter `n` you can enable Kerberos authentication later using the `gpcmdr --krbenable` command.

If you choose to enable Kerberos authentication:

- a. At the prompt `Enter web server name for this instance` enter the name of the host from the Kerberos principal. The principal name is in the format `HTTP/<host>@<realm>`. The host must be entered in the same format as the Kerberos principal and should exclude the port number.
 - b. At the prompt `Enter the GPDB Kerberos service name` enter the name of the Kerberos service principal for Greenplum Database, which is in the format `postgres/<host>@<realm>`.
 - c. At the prompt `Choose Kerberos mode` enter the number of the Kerberos mode you want to use. See [Enabling Authentication With Kerberos](#) for a description of these options.
 - d. At the prompt `Enter the path to the keytab file` enter the full path to the keytab containing the web server principal. If you are setting up this instance on the Greenplum master, the keytab file may be the same one used for Greenplum Database. The file must be readable by the gadmin user.
7. Setup asks if you want to enable SSL connections for the Command Center Console. Enabling SSL also enables WebSockets for Command Center connections.

Note: Because database login information is sent over the network, it is strongly recommended to enable SSL to encrypt Command Center communications.

If you enter `y` to enable SSL, Setup asks you to specify the location of your X509 certificate file. Enter the full path to the certificate file. The certificate must be accessible and readable by the user that runs the Command Center web server. The path you enter is added to the Command Center instance's `app.conf` file.
 8. Setup asks if you want to copy this Command Center instance to a standby master. This option is not available for remote installations. If you enter `y`, Setup prompts you to enter the standby master host name.
 9. Setup prompts you to enter the port number for the Command Center instance web server. Each Command Center instance that runs on the host must have a unique port number. The default port number is 28080.
 10. Setup asks if you want to enable SSL for the Web API. Enabling SSL for the Web API ensures that the Command Center application GPCC REST API requests and responses are encrypted. If you enter `y` Setup prompts you to enter the path to the certificate file to use to authenticate the server for Web API requests.

Setup creates the new Greenplum Command Center instance in a subdirectory of `$GPPERFMONHOME/instances` with the name of the instance. See [About the Command Center Installation](#) for a description of the contents of this directory.

Start the new Console instance with the following command:

```
gpcmdr --start <instance-name>
```

See [Connecting to the Greenplum Command Center Console](#) to connect to the instance from a workstation.

See [Configuring Authentication for the Command Center Console](#) for help setting up access for Command Center users.

Creating the Command Center Console Instance (Non-interactive)

It can be useful to run `gpcmdr --setup` non-interactively, taking input from a file. For example, you could install GPCC and create Command Center instances as part of a Greenplum cluster installation script. To accomplish this, create a configuration file and supply it to the `gpcmdr` utility using the `--config_file` option:

```
gpcmdr --setup --config_file file
```

The configuration file is similar to a Windows INI file, containing one or more sections beginning with a section header in square braces. Parameters in the optional `[DEFAULT]` section apply to all subsequent sections and may be overridden. Each section other than `[DEFAULT]` defines a Command Center Console instance to create.

Parameters are specified one-per-line as name-value pairs separated with equals signs (`=`) or colons (`:`). Comments begin with a number sign (`#`) or semicolon (`;`) and continue to the end of the line.

Here is an example configuration file:

```
[DEFAULT]
# defaults apply to all instances
remote_db: false
enable_copy_standby: true
standby_master_host: smdw
enable_kerberos: false
enable_ssl: true
ssl_cert_file: /etc/ssl/certs/cert.pem
```

```
[production]
master_hostname: mdw
instance_name: prod
display_name: Production
master_port: 5432
web_port: 28080
```

```
[development]
master_hostname: mdw
instance_name: dev
enable_copy_standby: false ; override
display_name: Development
master_port: 5532
web_port: 28090
```

See [Setup Configuration File](#) for a detailed description of the setup configuration file syntax and parameters.

Start the Command Center Console Instance

Start the Greenplum Command Center Console instance by entering:

```
gpcmdr --start
```

If you do not specify an instance name, all Command Center Console instances are started. To start a particular instance, you can specify the name of the instance. For example:

```
gpcmdr --start instance_name
```

See [Administering Greenplum Command Center](#) for a complete list of administrative commands.

Greenplum Command Center User Guide

The Greenplum Command Center web interface is a management tool that provides system status and query monitoring facilities for Greenplum Database administrators and users.

Command Center views allow you to instantly view the overall status of the Greenplum Database system. You can drill down to see details about hosts, database segments, queries, and CPU, memory, and disk resource utilization.

The following topics describe the information displayed on each Command Center view.

- [Dashboard](#)
The Dashboard displays when you first log in to the Command Center. It shows an overview of the status of the Greenplum Database cluster the Command Center manages and provides easy access to detailed information about any aspect of system status.
- [Query Monitor](#)
View current running and queued queries. Select a query to view its query text and execution plan. With proper permissions, choose queries and cancel them.
- [Host Metrics](#)
View real-time statistics by server in a table format.
- [Cluster Metrics](#)
View charts of current and recent statistics for all hosts (excluding master and standby).
- [History](#)
View queries and historical charts of statistics for a selected time period, optionally filtered by database and user.
- [System>Segment Status](#)
View a status summary for all primary and mirror segments and details for each segment.
- [System>Storage Status](#)
View the current percentage disk space in use for master and segment hosts, a historical chart of segment host disk usage, and current disk usage by host.
- [Admin>Permissions](#)
View permissions levels for Command Center users. Users with Admin permission can change permission levels.
- [Admin>Authentication](#)
View the Greenplum Database `pg_hba.conf` host-based authentication file. Users with Admin permission can edit the file.
- [Admin>Workload Mgmt](#)
Create, view, edit, and delete Greenplum Workload Manager rules.

Connecting to the Greenplum Command Center Console

You can sign in to the Command Center Console with a name and password or sign in anonymously to view just the **Query Monitor** view.

Open the Command Center Console in a supported browser using the host name and port configured for the Command Center web server. For example, to open a secure Command Center instance on a host named `mdw` at port 28080, enter this URL into your browser:

```
https://mdw:28080
```

- To view the **Query Monitor** page without signing in, click the **View Query Monitor** link. This takes you immediately to the [Query Monitor](#) view. To access additional Command Center features, click **Sign In** on the **Query Monitor** view and sign in with a valid Command Center user name and password.
- To sign in as a Command Center user, enter the user name and password of a Greenplum role that has been configured to allow authentication to Greenplum Command Center, then click **Sign In**. This opens the Dashboard page of the Command Center Console, which provides a graphical system snapshot and a summary view of active queries. See the [Dashboard](#) for information about the Dashboard view.

Dashboard

The **Dashboard** displays when you first sign in to Pivotal Greenplum Command Center. The **Dashboard** provides a quick view of the current system status, Segment Health, Queries, CPU, Memory, and Disk usage. Clicking on a panel provides more detailed information about the metric. The Alerts panel shows the most recent messages from the Greenplum Database log file. Some information is available only to Command Center users with Admin or Operator permission level.

System Information

The following system information is displayed at the top of the page.

Uptime

The elapsed time since the Greenplum Database system was last started.

GPDB Version

The version of the Greenplum Database software the monitored cluster is running.

Connections

The number of active Greenplum Database sessions (client connections).

Server

The name of this Greenplum Command Center instance.

Last Sync

Date and time the data was last synchronized. The Command Center user interface updates views with live data every 15 seconds.

System Summary

The **System Summary** section of the Dashboard provides a quick overview of the status of the database system and segments this Command Center instance monitors.

Database State

Database State is the current state of the Greenplum Database system. The state can be one of the following:

- **Normal:** The database is functioning with no major errors or performance issues.
- **Segment(s) Down:** The database is in change-tracking mode or resync mode. Overall performance and system reliability is greatly reduced. See the *Pivotal Greenplum Database System Administrator Guide* for information about resolving this condition.
- **Database Unreachable:** The Greenplum Performance Monitor agent cannot connect to the database. The database is likely down. See the *Pivotal Greenplum Database System Administrator Guide* for troubleshooting information.
- **Unbalanced:** Some segments are not running in their preferred roles. That is, primaries are running as mirrors and mirrors are running as primaries, resulting in unbalanced processing.
- **Resyncing:** The database is performing a recovery or rebalance operation.

Segment Status

The bar graph in the **Segment Status** section shows the up or down status of all database segments in your Pivotal Greenplum Database system. A color indicator and associated number indicate the number of database segments that are currently in that particular state. Segments can have the following states:

- **Up** (Green)
- **Down** (Red)

Clicking the **Segment Status** panel displays the [Segment Status](#) Command Center page.

Disk Usage Summary

This chart displays total disk usage and disk available for the Greenplum master host and segment hosts at the last synchronization. Hover over the chart to see the amount of disk used, free, and total.

Queries

This graph displays a summary view of active and queued queries for the last 60 minutes. Click on the colored dot next to the **Running** or **Queued** label to toggle the line on or off. At least one line must be visible at all times. Hover over the graph to display the number of queries for each visible line at that point in time.

CPU

This graph displays average CPU usage across the entire cluster, for the last 60 minutes. The graph displays separate lines for system processes and user processes. The user CPU usage includes the Greenplum database master, standby, and segment processes. Click on the colored dot next to the **System** or **User** label to toggle that line on or off. At least one line must be visible at all times.

Hovering the cursor over a line in the graph displays a small window with the percentage of CPU used at that point in time for the visible lines and the total if both the system and user lines are visible.

Memory

This graph displays the average percent of memory used across the entire cluster over the last 60 minutes. Hover over the line to display the percent of memory used at that point in time.

Alerts

Admin and Operator permission levels only

The **Alerts** panel displays recent messages from the Greenplum Database `pg_log` log file. The panel is updated at each synchronization. Filter the messages by severity level using the controls at the top right of the panel.

Query Monitor

The **Query Monitor** view allows you to view detailed information for active queries running on the Greenplum Database system. Users with Admin or Operator permission can see and cancel all users' queries.

Anyone able to access the Command Center web server can sign in anonymously and view the system status and query list on this page. Anonymous users cannot cancel queries or access any other Command Center features.

Data is collected on currently running queries and the query monitor metrics are updated every 15 seconds. The time of the last update and a graphical timer showing the time remaining before the next update are displayed at the top of the page.

With the information available in this view, Greenplum Database administrators can easily:

- Understand how the system is being used — both in real-time and trending over time.
- Identify and diagnose problem queries while they are running, detect skew, find runaway queries, and so on.
- Review and balance the query load on the system by better optimizing and scheduling the query load.
- Cancel queries that disrupt system performance.
- Post a message that is displayed for all Command Center users.

Query Metrics

The Query Monitor table displays the following columns for queries.

Query ID

An identification string for the query. In the Console, this looks like "1295397846-56415-2". Command Center generates this ID by combining the query record's `tmid`, `ssid`, and `cont` fields. (See [queries_*](#) in the *Command Center Database Reference*.)

Status

The status of the query. This can be one of the following:

- Queued: the query has not yet started to execute
- Running: execution has started, is not yet complete
- Done: completed successfully
- Cancelling: cancel request sent, cancel pending
- Cancelled: terminated, no longer running

User

The Greenplum Database user who submitted the query.

Database

The name of the database that was queried.

Submit Time

The time the query was submitted to the query planner.

Queue Time

The amount of time the query has been (or was) in queue awaiting execution.

Run Time

The amount of time since execution began.

CPU %

(Active queries only.) Current CPU percent average for all processes executing this query. The percentages for all processes running on each segment are averaged, and then the average of all those values is calculated to render this metric. Current CPU percent average is always zero in historical and tail data.

CPU Skew

The amount of CPU skew. CPU skew occurs when query executor processes for one segment use a disproportionate amount of CPU compared to processes for other segments executing the query. This value is the coefficient of variation for the CPU used by processes running this query on each segment, multiplied by 100. For example, a value of .95 is shown as 95.

Row Skew

A measure of row skew in the system. Row skew occurs when one segment produces a disproportionate number of rows for a query. This value is the coefficient of variation for the Rows Out metric of all iterators across all segments for this query, multiplied by 100. For example, a value of .95 is shown as 95.

Queue

The name of the resource queue for the query.

Priority

Each query inherits the priority assigned to its resource queue. For more information about Resource Queues and Query Plans, refer to the *Greenplum Database Administrator Guide*.

Using the Query Monitor Controls

- Click a column heading to sort the rows on that column in ascending or descending order.
- Click the checkbox at the left of a row to choose a query to cancel or export. Click the checkbox in the heading row to choose all queries.
- Click **Cancel Query** to cancel selected queries.
- Click **Export** to download a comma-separated values (CSV) text file containing rows for the selected queries. When no queries are selected, all rows are exported. The default file name is `spreadsheet.csv`.
- Click any query ID to see the [Query Details](#), including metrics, the text of the query, and the query plan.
- Users with Admin permission can add a message to display for all Command Center users. The message box has controls to add, edit, and delete the message.

Host Metrics

The **Host Metrics** page displays a table of the hosts in the cluster with statistics collected at the most recent quantum interval. At the top, **Last Sync** displays the time the statistics were last updated.

Click a column header to sort the table by that column. Click again to toggle between ascending and descending sort. Master and standby hosts are not included in the sort and are always displayed following the sorted list of segment hosts.

For each server, the following columns are displayed:

Hostname

The hostname name of the server.

CPU Total/Sys/User (%)

The total percentage of CPU in use is displayed next to a graph illustrating the CPU used for system and user processes. Hover over the table cell to show the percentages used for system and user processes and the percentage CPU idle.

Memory In Use (%)

The percentage of host memory in use is displayed next to a graph illustrating the memory in use and available. Hover over the table cell to see memory used and available in gigabytes.

Memory is calculated as follows:

Total = MemTotal

Free = MemFree + Buffers + Cached

Used = Total - Free

Disk R (MB/s) | Skew

Disk read rate in megabytes per second is displayed next to a graph of calculated disk read skew. Hover over the table cell to see a Low/Medium/High rating for disk skew.

Disk W (MB/s) | Skew

Disk write rate in megabytes per second is displayed next to a graph of calculated disk write skew. Hover over the table cell to see a Low/Medium/High rating for disk write skew.

Net R (MB/s) | Skew

Network read rate in megabytes per second is displayed next to a graph of calculated network read skew. Hover over the table cell to see a Low/Medium/High rating for network read skew.

Net W (MB/s) | Skew

Network write rate in megabytes per second is displayed next to a graph of calculated network write skew. Hover over the table cell to see a Low/Medium/High rating for network write skew.

About Skew Calculations

Disk and Network skew ratings are calculated as each server's standard deviation from the mean calculated from all segment hosts.

Low

Value is within 1 standard deviation from the mean. (Note: if the variance of the set is less than 3, skew is considered low regardless of deviation from mean.)

Moderate

Value is between 1 and 2 standard deviations from the mean.

Very High

Value is greater than 3 standard deviations from the mean.

Cluster Metrics

The **Cluster Metrics** page shows consolidated statistics for all segment hosts in the Greenplum cluster. Master and standby master hosts are excluded from the metrics.

The charts display metrics for the last time period set by the control in the top right corner of the screen.

Use the **Show/hide Charts** control to choose which metrics to display.

Hover over any of the charts to see values for the metrics at a point in time in pop-up boxes. The charts are synchronized so that hovering over any chart shows the same point in time in all charts.

The current value of a metric is shown in the upper right corner of its chart.

On charts with multiple metrics, toggle the display for a line on or off by clicking the line's label in the legend at the top right of the chart. At least one line must be displayed. All lines are redisplayed at the next quantum interval.

The page has charts for the following metrics:

Queries

The number of queries running and the number of queries queued to run.

CPU

The percentage CPU used by system processes and the percentage CPU used by user processes.

Memory

Percentage of memory in use.

Memory is calculated as follows:

Total = MemTotal

Free = MemFree + Buffers + Cached

Used = MemTotal - Free

Disk I/O

Disk read and write rates in megabytes per second.

Network

Network I/O read and write rates in megabytes per second. Network metrics include traffic over all NICs (network interface cards), including internal interconnect and administrative traffic.

Load

System load average for 1-minute, 5-minute, and 15-minute periods.

Swap

Percentage of swap space used.

Monitoring Multiple Greenplum Database Clusters

The Greenplum Command Center Multi-cluster view displays health status for multiple Greenplum Database clusters. The charts can be divided into categories.

Each Greenplum Database cluster that appears in the Multi-cluster view must have its own Command Center instance. The Multi-cluster view is hosted on a single, designated master Command Center instance. The master instance retrieves the health status data from each of the clusters' Command Center instances.

Clicking the status panel for a cluster loads that cluster's Command Center into a new browser window. If auto-login is enabled in the multi-cluster configuration file, the login screen is bypassed and the Dashboard is displayed.

Multi-cluster support is enabled by deploying a configuration file to the Command Center instance directories. See [Enabling Multi-Cluster Support](#) for instructions to set up the Multi-cluster page.

By default, any Command Center user may view multi-cluster status. Access to clusters can be restricted in the multi-cluster configuration file.

The Multi-cluster view displays the following information for each cluster.

Cluster Name

The name for the cluster, as specified in the multi-cluster configuration file.

Database State

The current state of the Greenplum Database cluster. The state can be one of the following:

- **Normal:** The database is functioning with no major errors or performance issues.
- **Segment(s) Down:** The database is in change-tracking mode or resync mode. Overall performance and system reliability is greatly reduced. See the *Pivotal Greenplum Database System Administrator Guide* for information about resolving this condition.
- **Database Unreachable:** The Greenplum Performance Monitor agent cannot connect to the database. The database is likely down. See the *Pivotal Greenplum Database System Administrator Guide* for troubleshooting information.
- **Unbalanced:** Some segments are not running in their preferred roles. That is, primaries are running as mirrors and mirrors are running as primaries, resulting in unbalanced processing.
- **Resyncing:** The database is performing a recovery or rebalance operation.

Uptime

The elapsed time since the Greenplum Database system was last started.

GPDB Version

The version of the Greenplum Database software each monitored cluster is running.

Connections

The number of active Greenplum Database sessions (client connections).

Active Queries

The number queries queued or currently executing in the database.

History

The **History** page allows you to display system metrics and queries executed during a specified time period. Queries may also be filtered by database and/or user.

Set the time period to display by entering dates and times in the **From** and **To** date and time fields. You can enter dates by typing them into the date field or by choosing from the pop-up calendar. Enter 24-hour times in HH:MM format.

To restrict queries that display in the query table at the bottom of the page, enter a Greenplum database name in the **Database** field, a user name in the **User** field, or both. Filtering by database and user only affects the queries displayed in the table. The metrics displayed in charts include all activity during the selected time period.

Click **Search** to display results that match your criteria.

You can click and drag on a chart to zoom in on a time range. Click **Search** to update the query list and charts to the selected range.

Scroll charts left or right by hovering over the edge of the chart and clicking an arrow. Click < or > to move in half steps. Click < or > to move in full steps.

In the query list, select or hover over a query to highlight its queued and run time in the charts.

Charts of the following metrics are available. Show or hide them at any time with the checklist at the upper right of the view.

Queries

The number of queries running and the number of queries queued to run.

CPU

The percentage CPU used by system processes and the percentage CPU used by user processes.

Memory

Percentage of memory in use.

Disk I/O

Disk read and write rates in megabytes per second.

Network

Network I/O read and write rates in megabytes per second. Network metrics include traffic over all NICs (network interface cards), including internal interconnect and administrative traffic.

Load

System load average for 1-minute, 5-minute, and 15-minute periods.

Swap

Percentage of swap space used.

Query Metrics

The Query table displays queries that were active during the specified time period, including queries that started before or finished after the specified time. However, queries that are still active are not included in the table; these queries can be viewed on the [Query Monitor](#) page.

The query table has the following columns:

Query ID

An identification string for the query. In the Console, this looks like "1295397846-56415-2".

Status

The final status of the query. This can be one of the following:

- Done
- Cancelled

User

The Greenplum Database user who submitted the query.

Database

The name of the database that was queried.

Submit Time

The time the query was submitted to the query planner.

Queued Time

The amount of time a query spent in the queue before it was executed.

Run Time

The amount of time the query required to produce a result.

End Time

The time the query completed or was cancelled.

CPU Skew

The amount of CPU skew. CPU skew occurs when query executor processes for one segment use a disproportionate amount of CPU compared to processes for other segments executing the query. This value is the coefficient of variation for the CPU used by processes running this query on each segment, multiplied by 100. For example, a value of .95 is shown as 95.

Row Skew

A measure of row skew in the system. Row skew occurs when one segment produces a disproportionate number of rows for a query. This value is the coefficient of variation for the Rows Out metric of all iterators across all segments for this query, multiplied by 100. For example, a value of .95 is shown as 95.

Queue

The name of the resource queue for the query.

Priority

Each query inherits the priority assigned to its resource queue.

For more information about Resource Queues and Query Plans, refer to the *Greenplum Database Administrator Guide*.

System

The **System** view provides system metrics for individual hosts in the Greenplum cluster and for the entire cluster. Click **Host Metrics** to display metrics for each host. Click **Cluster Metrics** to display consolidated metrics for the entire cluster.

[Segment Status](#)

[Storage Status](#)

Segment Status

The **Segment Status** page provides a health overview for the Greenplum Database segments and details for each primary and mirror segment.

Segment Summary

Greenplum Database is most efficient when all segments are operating in their preferred roles. The **Segment Summary** panel tells you the overall segment status and if any mirrors are acting as primaries.

The **Segment Summary** panel provides the following information:

Database State

The database state can be one of the following:

- **Normal:** The database is functioning with no major errors or performance issues.
- **Segment(s) Down:** The database is in change-tracking mode or resync mode. Overall performance and system reliability is greatly reduced. See the *Pivotal Greenplum Database System Administrator Guide* for information about resolving this condition.
- **Database Unreachable:** The Greenplum Performance Monitor agent cannot connect to the database. The database is likely down. See the *Pivotal Greenplum Database System Administrator Guide* for troubleshooting information.
- **Unbalanced:** Some segments are not running in their preferred roles. That is, primaries are running as mirrors and mirrors are running as primaries, resulting in unbalanced processing.
- **Resyncing:** The database is performing a recovery or rebalance operation.

Mirrors Acting as Primary

The number of mirror segments acting as primary segments.

Recommended Actions

Suggests actions to perform to restore the cluster to balance. These include:

- Recover and Rebalance
- Rebalance

These actions are executed from the command line using the `gprecoverseg` Greenplum management utility. See `gprecoverseg` in the *Pivotal Greenplum Database Utility Reference* for more information.

Total Segments

The total number of primary and mirror segments in the Greenplum cluster.

Segment Hosts

The total number of segment hosts in the Greenplum cluster.

Segment Health

The **Segment Health** panel contains charts for Greenplum Database segments' status, replication mode, and preferred roles.

Status

Numbers of segments that are down and up.

Replication Mode

A chart that shows the number of segments in each of the possible replication modes.

- **Not Syncing:** The primary segment and mirror segment are active and all changes to the primary segment have been copied to the mirror using a file block replication process.
- **Change Tracking:** If a primary segment is unable to copy changes to its mirror segment using the file replication process, it logs the unsent changes locally so they can be replicated when the mirror again becomes available. This can happen if a mirror segment goes down or if a primary segment goes down and its mirror segment automatically assumes the primary role.
- **Resyncing:** When a down segment is brought back up, administrators initiate a recovery process to return it to operation. The recovery process synchronizes the segment with the active primary and copies the changes missed while the segment was down.
- **Synced:** Once all mirrors and their primaries are synchronized, the system state becomes synchronized.

Preferred Roles

The red portion of the Preferred Role chart shows the numbers of segments that not operating in their preferred primary or mirror roles. If the chart is not solid green, the performance of the Greenplum cluster is not optimal.

Primary and mirror segments are distributed evenly among the segment hosts to ensure that each host performs an equivalent share of the work and primary segments and their mirror segments reside on different segment hosts. When a primary segment goes down, its mirror on another host in the cluster automatically assumes the primary role, increasing the number of primary segments running on that host. This uneven distribution of the workload will affect query performance until the down segment is restored and the segments are returned to their original, preferred, roles.

Segment Table

The table at the bottom of the **Segment Status** page contains a detailed row for every primary and mirror segment in the Greenplum Cluster. The table has the following columns for each segment:

Hostname

The name of the segment host where the segment is running.

Address

The network interface on the segment host for the segment.

Port

The port number assigned to the segment.

DBID

The unique identifier for the segment instance.

ContentID

The content identifier for the segment, from 0 to the number of segments minus 1. A primary segment and its mirror have the same ContentID. The master and standby master, which have ContentID -1, are excluded from the table.

Status

“UP” if the segment is running, “DOWN” if the segment has failed or is unreachable.

Role

The segment’s current role, either “primary” or “mirror”.

Preferred Role

The segment’s intended role, either “primary” or “mirror”.

Replication Mode

The replication status for the segment. See [Segment Health](#) for possible values.

Last Event[[Total]

The date and time of last segment health-related activity. Click to display a list of recent events.

Storage Status

The **Storage Status** page shows current historical disk usage for Greenplum master and segment hosts.

Disk Usage Summary

You can see current disk space in use, space free, and total space in the Disk Usage Summary panel. Disk space metrics for the segment hosts (GP Segments) and the master (GP Master) are shown in separate bar charts.

The GP Segments bar chart shows combined disk space for all segments.

The GP Masters bar chart shows combined disk space for master and standby master.

Hover over either of the charts to see the space used, free, and total in gigabytes and as a percentage of the total.

GP Segments Usage History

The GP Segments Usage History panel presents a chart of percentage of disk space in use for the time period set by the control in the panel header.

Hover over the chart to see the percentage disk in use on any given point.

Storage Status Table

The Storage Status table provides current disk space usage metrics for each host and by data directory within hosts.

Admin

The **Admin** view provides the ability to manage authentication and authorization for Greenplum Command Center Console and Greenplum Database users.

- [Permissions](#)
View the Greenplum Command Center permission level for Greenplum Database users. Administrators can change user's permission levels.
- [Authentication](#)
View the Greenplum Database host-based authentication file (`pg_hba.conf`). Administrators can change the file.

Permission Levels for GPCC Access

The **Permissions Levels for GPCC Access** screen allows users with Operator Basic, Operator, or Admin permission to view permissions for Command Center users. Users with Admin permission can set permissions for all users.

Viewing User Permissions

Initially, all Greenplum Database login users are included in the list with their current permission levels.

- To filter by role name, enter all or part of the user's database role name in the *Role Name* field. The filter performs a simple substring search and displays users with matching role names. Click the **Role Name** label to reverse the search order.
- To filter for users with a specific permission level, choose the permission level from the **Permission Level** list.
- Role Name and Permission Level filters can be used together.
- To reset the filters, remove all text from the *Role Name* field and choose **Filter by...** from the **Permission Level** list.

Changing User Permission Levels

Users with Admin permission can change permission levels.

1. Use the **Role Name** and **Permission Level** filters to display the roles you want to change.
2. Check the box next to a role name to select the user, or check the box in the heading to select all displayed users.
3. Select the new permissions level for each user from the list in the **Permission Level** column, or select a new permission level for all selected users from the **Change Selected to...** list.

Authentication

The **System>Authentication** screen allows users with Operator Basic, Operator, and Admin permission to view the Greenplum Database host-based authentication file, `pg_hba.conf`.

Users with Admin permission can add, remove, change, and move entries in the file. The Command Center UI validates entries to ensure correct syntax. Previous versions of the file are archived so that you can restore an earlier version or audit changes.

See [Authentication](#) for an overview of user authentication options for Greenplum Database and Greenplum Command Server.

See [pg_hba.conf file](#) in the PostgreSQL documentation for a detailed description of the contents of the `pg_hba.conf` file.

Viewing the Host-Based Authentication File

Choose **Admin>Authentication** to display the content of the Greenplum Database `pg_hba.conf` file.

The `pg_hba.conf` file contains a list of entries that specify the characteristics of database connection requests and authentication methods. When Greenplum Database receives a connection request from a client, it compares the request to each entry in the `pg_hba.conf` entry in turn until a match is found. The request is authenticated using the specified authentication method and, if successful, the connection is accepted.

Editing the Host-Based Authentication File

Command Center users with the *Admin* permission can edit the `pg_hba.conf` file. Note that any changes you make are lost if you move to another screen before you save them.

- To change an existing entry, click anywhere on the entry. Edit the fields and click **Save** to save your changes, or **Cancel** to revert changes.
- To move an entry up or down in the list, click on the  symbol, drag the line to the desired location, and release.
- To add a new entry to the end of the file, click **Add New Entry** at the bottom of the screen. Edit the fields and click **Save** to save your changes, or **Cancel** to abandon the new entry.
- To add a new entry after an existing entry, highlight the existing entry and click . Edit the fields and click **Save** to save your changes, or **Cancel** to abandon the new entry.
- To copy an entry, select the entry and click . A copy of the selected entry is added below the selected entry and displayed for editing. Edit the fields and click **Save** to save your changes, or **Cancel** to abandon the copy.
- To add a comment to the file, add an entry by clicking **Add New Entry** or  and then choose `#` from the `Type` list.
- To toggle an entry between active and inactive, select the line and click the **active/inactive** toggle control to the right. This action adds or removes a comment character (`#`) at the beginning of the entry.
- To remove an entry, highlight the line and click . The entry is displayed with strikethrough text. You can restore the entry by highlighting it and clicking **undelete**. The entry is permanently removed when you click **Save config and update GPDB**.
- To finish editing, click **Save config and update GPDB**. Then click **Save and Update** to save your changes or click **Cancel** to return with your edits intact.

When you select **Save and Update**, the `pg_hba.conf` file is saved and refreshed in Greenplum Database. Note that existing client connections are unaffected.

Loading a Previous Version of the Host-Based Authentication File

When you save a new version of the `pg_hba.conf` file, a copy is saved in the Greenplum Database `$MASTER_DATA_DIRECTORY/pg_hba_archive` directory as `pg_hba.conf-<timestamp>`.

To view an archived version of the `pg_hba.conf` file, click **Load versions...** and click the timestamp for the version to display.

To revert to a previous version of the file, load the previous version and then click **Save config and update GPDB**. The configuration is refreshed in Greenplum Database and saved as a new version in the archive directory.

Workload Management

The **System>Workload Mgmt** screen allows users with Admin or Operator permission to view Greenplum Workload Manager rules. Users with Admin permission can add, edit, or delete rules.

 The **System>Workload Mgmt** screen is present if, when you sign in, Greenplum Command Center detects that Workload Manager is running on the Greenplum Database master host. If the page is not available, make sure that Workload Manager is running and then sign out of Greenplum Command Center and sign in again.

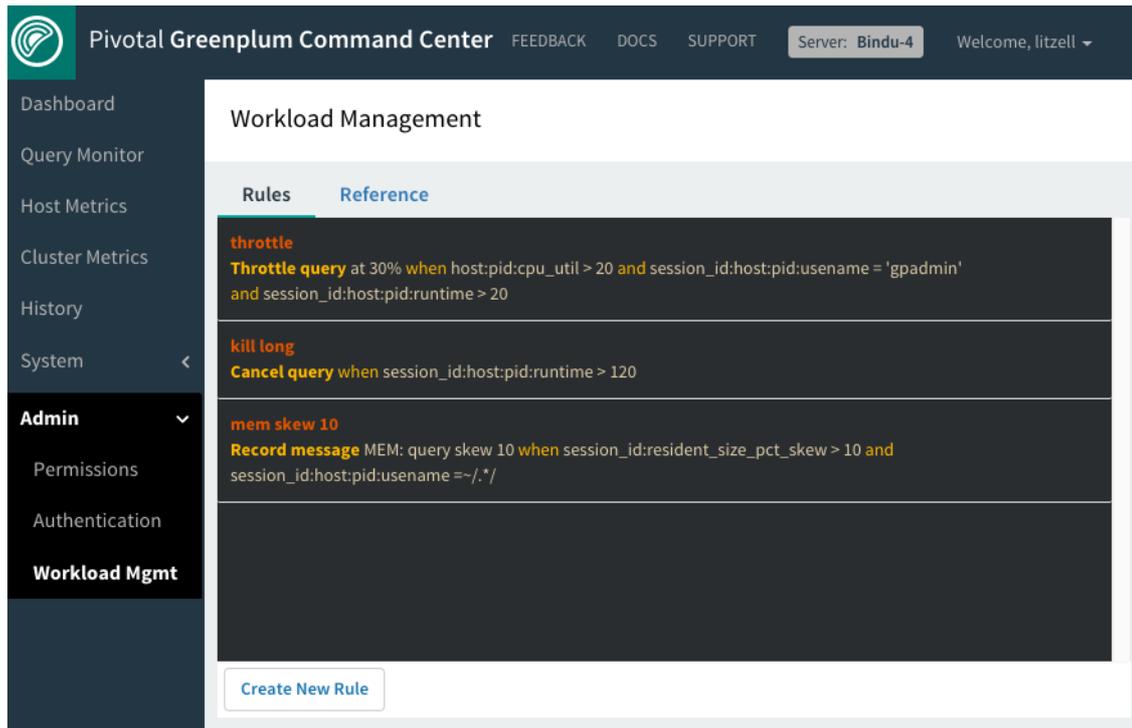
Rules Tab

On the **Rules** tab you can see the Workload Manager rules. If you have Admin permission, you can also manage rules. Any changes you save to the rules are immediately active in Workload Manager.

Viewing Rules

Choose **Admin>Workload Mgmt** to display the currently active Workload Manager rules.

The current set of rules is displayed, one row per rule.



The top line of a rule entry is the name of the rule, a symbolic label used to reference the rule. The name must be unique.

The second line of the rule display shows the following information:

- the rule's action and its argument, if any
- the rule's `when` clause, a Boolean expression that determines when the rule is matched
- the rules' optional `including` clause, a list of metrics to include in the context

Actions

The rule action describes the action that Workload Manager performs when the rule is matched. There are four actions:

- **Record** - Records a message in the `gp_wlm_records` table, along with metrics from the Greenplum Database session or query that matched the rule. The text of the message the rule records is shown between “Record” and the `when` keyword. One row is added to the `gp_wlm_records` table when the rule is matched and another row when the rule stops matching. This action corresponds to the Workload Manager `gpdb_record` action.
- **Cancel query** - Cancels the query that matches the criteria specified by the `when` clause. When a query matches this rule on any host or segment, Workload Manager calls the PostgreSQL `pg_cancel_backend()` function, which cancels the query. The event is logged in the `gp_wlm_events` table. This action corresponds to the Workload Manager `pg_cancel_backend` action.
- **Terminate session** - Terminates the user session that matches the criteria specified by the `when` clause. When a query matches this rule, Workload Manager calls the PostgreSQL `pg_terminate_backend()` function, which ends the client’s session. The event is logged in the `gp_wlm_events` table. This action corresponds to the Workload Manager `pg_terminate_backend` action.
- **Throttle CPU** - Holds a query to a maximum share of CPU on a host, specified as a percentage of CPU utilization between 0 and 100. When a query first matches this rule on any host, Workload Manager begins to throttle the amount of CPU consumed by all of the query’s processes running on that host to the specified percentage. Workload Manager writes a `START` record in the `gp_wlm_events` table when the rule is matched. When the rule no longer matches, Workload Manager stops throttling and writes an `END` record in the `gp_wlm_events` table.

When clause

The `when` clause is a Boolean expression that selects Greenplum Database query processes. The Boolean expression allows comparing numeric- and string-valued metrics with constant values using standard operators such as `<`, `>`, and `=`. String-valued metrics can be matched using regular expressions. Comparisons can be joined with `AND` and `OR` operators and parentheses can be used to force the order in which they evaluate.

When Workload Manager matches a `when` expression to a query process the rule’s action is applied.

Including clause

The `including` clause is a comma-separated list of metrics that are to be added to the rule’s context. Use this clause to capture metrics that are not used in the `when` clause of the rule. The include metrics are saved in the `context` column of the `gp_wlm_records` or `gp_wlm_events` table when the rule is matched. Omit the scope portion of the metric in the `including` clause; Workload Manager determines the scope from the metric name.

Editing a Rule

1. Click anywhere in a rule to edit the rule. The rule’s action (and argument, if any) and the `when` clause are presented for editing.

The screenshot shows the Pivotal Greenplum Command Center interface. The 'Workload Management' section is active, displaying a list of rules. One rule, 'throttle', is selected and its details are shown in a modal window. The rule's action is 'Throttle query' with the argument 'at max cpu 30'. The 'when' clause is 'host:pid.cpu_util > 20 and session_id:host:pid.username = 'gpadmin' and session_id:host:pid'. A pop-up menu is open, showing a list of metrics that can be used in the 'when' clause, such as 'session_id:host:pid', 'session_id:host:pid:application_name', 'session_id:host:pid:backend_start', 'session_id:host:pid:client_addr', 'session_id:host:pid:client_port', 'session_id:host:pid:current_query', 'session_id:host:pid:datid', 'session_id:host:pid:datname', 'session_id:host:pid:gpdb_segment_role', and 'session_id:host:pid:query_start'.

- Choosing a new action from the Action list changes the argument field to the type required by the new action or removes the field for actions with no argument.
- While editing text in the `when` field, you can choose metrics and operators from a pop-up list, filtered to match initial characters you type.

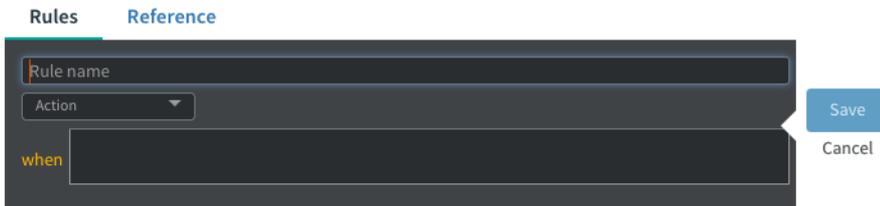
2. Click **Save** to save your changes or **Cancel** to abandon your changes.

Workload Manager validates the rule before it is accepted. If there are any errors, a message is displayed and the editing mode remains active.

Creating a New Rule

1. Click **Create New Rule** to add a new rule. An empty rule editing form is displayed.

Workload Management



The screenshot shows the 'Rules' tab in the Workload Management interface. It features a form with the following elements: a 'Rule name' text input field, an 'Action' dropdown menu, and a 'when' section with a large text area for defining the rule's conditions. To the right of the form are two buttons: 'Save' and 'Cancel'.

2. Enter a unique rule name in the **Rule name** field.
3. Complete the form as described in [Editing a Rule](#).
4. Click **Save** to save the new rule or **Cancel** to abandon the rule.

Workload Manager validates the rule before it is accepted. If there are any errors, a message is displayed and the editing mode remains active.

Duplicating a Rule

Duplicating a rule is useful if you want to create a new rule similar to an existing rule. You can also duplicate a rule if you want to change the rule's name, since editing an active rule's name is not allowed. Just duplicate the rule, give the copy a new name and save it, and then delete the old rule.

1. Click **Duplicate** to the right of the rule you want to copy. The rule is displayed in a rule editing form.
2. Change the rule name to a new unique name.
3. Edit the rule as described in [Editing a Rule](#).
4. Click **Save** to save the new rule or **Cancel** to abandon the rule.

Workload Manager validates the rule before it is accepted. If there are any errors, a message is displayed and the editing mode remains active.

Deleting a Rule

1. Click **Delete** to the right of the rule you want to delete. A **Confirm rule removal** dialog box is displayed.
2. Click **Delete rule** to delete the rule or **Cancel** to keep the rule.

Reference Tab

The **Reference** tab is a quick reference that illustrates the anatomy of a Workload Manager rule, describes syntax for the parts of a rule, and provides examples of rules. The quick reference includes links to more detailed information in the latest Greenplum Workload Manager documentation.

Workload Management

Rules Reference

Anatomy of a Workload Manager Rule

Take **an action** when **my conditions** are met. Also include **additional metrics**.

Action

Record	log that the rule was triggered
Throttle query	limit the % of CPU a query can consume
Cancel query	cancel the offending query
Terminate session	kill the entire user session

Condition (when)

Boolean expression that identifies queries by metric values

“When” metrics are referenced with the format `scope:metric`. See [Metric Reference](#) for details. The condition expression can include any number of AND or OR clauses.

Additional Metrics (including)

List of metrics to capture in addition to your condition metrics (optional)

“Including” metrics are referenced solely by metric name. See [Metric Reference](#) for details.

Example Rules

Record high cpu utilization queries

Creates record in `wlm_records` when `gpadmin` user runs a query and its total cpu utilization on a host exceeds 100%.

```
Record CPU over 100 pct
Record with message Too much cpu for gpadmin when
```

Rule Syntax

Action

In GPCC, just select your desired action and provide any necessary argument values.

Condition—the WHEN clause

“When” metrics are referenced with the format `scope:metric`. See [Metric Reference](#) for details on what metrics are available and which `scope:metric` string to use for each.

Metrics can be compared to values using the following operators:

=	Number or quoted string	Matches when values are exactly equal
!=	Number or quoted string	Matches when values are not equal
=~	Regular expression enclosed in slashes <code>scope:metric =~ /.*/</code>	Regex match against string value using Posix regular expression syntax
>	Number	Greater than
<	Number	Less than
>=	Number	Greater than or equal to
<=	Number	Less than or equal to

Additional Metrics—the INCLUDING clause

This optional clause includes desired metrics not captured by default or by virtue of being mentioned in the “when” clause.

Administering Greenplum Command Center

System administration information for the Greenplum Command Center.

- [About the Command Center Installation](#)
- [Starting and Stopping Greenplum Command Center](#)
- [Administering Command Center Agents](#)
- [Administering the Command Center Database](#)
- [Administering the Web Server](#)
- [Configuring Greenplum Command Center](#)
- [Enabling Multi-Cluster Support](#)
- [Securing a Greenplum Command Center Console Instance](#)

About the Command Center Installation

The installation and setup procedures create a software installation directory and a directory containing files and folders to support each Greenplum Command Center Console instance.

Software Installation Directory

The following files and first-level subdirectories are copied into the installation folder that you specified when you installed Greenplum Command Center Console. This location is referred to as `$GPPERFMONHOME`.

- `gpcc_path.sh` – file containing environment variables for Command Center
- `bin` – program files for Greenplum Command Center
- `etc` – contains `ddl.sql`
- `gpcc-wlm-<version>-<platform>.bin` – installer for [Greenplum Workload Manager](#)
- `instances` – contains a subdirectory of resources for each Greenplum Database instance monitored by the console
- `lib` – library files for Greenplum Command Center
- `open_source_licenses_GPCC.txt` – licenses for open source components used by Greenplum Command Center
- `www` – web server and user interface files

Instances Directory

The `$GPPERFMONHOME/instances` directory contains subdirectories named for each instance created during console setup. The `conf` subdirectory contains configuration files that you can edit. Other files and folders are used by the web services for the instance, and should not be modified or deleted.

Each subdirectory contains the following files and first-level subdirectories:

- `conf` – multi-cluster configuration file, `clusters.conf`
- `webserver` – web server logs for this instance and symbolic links to web server files in the installation directory

Starting and Stopping Greenplum Command Center

Greenplum Command Center includes the command center console and the command center agents.

Starting and Stopping Command Center Agents

Whenever the Greenplum Database server configuration parameter `gp_enable_gpperfmon` is enabled in the master `postgresql.conf` file, the Command Center agents will run and collect data. These agents are automatically stopped and started together with the Greenplum Database instance.

To disable the Command Center data collection agents, you must disable the `gp_enable_gpperfmon` parameter, and restart the Greenplum Database instance.

Starting and Stopping Command Center Console

Use the following `gpccmdr` commands to start, stop and restart Greenplum Command Center Console instances:

```
$ gpccmdr --start ["instance name"]
```

```
$ gpccmdr --stop ["instance name"]
```

```
$ gpccmdr --restart ["instance name"]
```

If you do not specify an instance name, all instances are started, stopped, or restarted at once. You can check the status of instances using:

```
$ gpccmdr --status ["instance name"]
```

Administering Command Center Agents

This topic describes basic agent administration tasks, including adding hosts and viewing agent log files.

Adding and Removing Hosts

Segment agents on new hosts are detected automatically by the master agent. Whenever `gp_enable_gpperfmon` is enabled on the master, the master monitor agent automatically detects, starts, and begins harvesting data from new segment agents.

To verify the addition of a new monitored host, you can check for the new hostname in the Greenplum Command Center Console System Metrics view. Alternately, you can query the `system_now` table for the row containing current metrics for each host. For example:

```
# SELECT * FROM system_now WHERE hostname='new_hostname';
```

Viewing and Maintaining Master Agent Log Files

Log messages for the master agent are written to the following file by default:

```
$MASTER_DATA_DIRECTORY/gpperfmon/logs/gpmmon.log
```

To change the log file location, edit the `log_location` parameter in `gpperfmon.conf`.

On the segment hosts, agent log messages are written to a `gpsmon.log` file in the segment instance's data directory. For a host with multiple segments, the agent log file is located in the data directory of the first segment, as listed in the `gp_configuration` table by dbid. If the segment agent is unable to log into this directory, it will log messages to the home directory of the user running Command Center (typically `gpadmin`).

Configuring Log File Rollover

At higher logging levels, the size of the log files may grow dramatically. To prevent the log files from growing to excessive size, you can add an optional log rollover parameter to `gpperfmon.conf`. The value of this parameter is measured in bytes. For example:

```
max_log_size = 10485760
```

With this setting, the log files will grow to 10MB before the system rolls over the log file. The timestamp is added to the log file name when it is rolled over. Administrators must periodically clean out old log files that are no longer needed.

Administering the Command Center Database

Data collected by Command Center agents is stored in a dedicated database called `gpperfmon` within the Greenplum Database instance. This database requires the typical database maintenance tasks, such as clean up of old historical data and periodic `ANALYZE`.

See the [gpperfmon Database Reference](#) section for a reference of the tables and views in the `gpperfmon` database.

Connecting to the Command Center Database

Database administrators can connect directly to the Command Center database (`gpperfmon`) using any Greenplum Database-compatible client program (such as `psql`). For example:

```
$ psql -d gpperfmon -h master_host -p 5432 -U gpadmin
```

Backing Up and Restoring the Command Center Database

The history tables of the Command Center database (`gpperfmon`) can be backed up and restored using the Greenplum Database parallel backup and restore utilities (`gpcrondump`, `gpdrestore`). See the *Greenplum Database Utility Guide* for more information.

Because the Command Center database has a low number of tables, you may prefer to devise a backup plan using the table-level backup features of `gp_dump`. For example, you can create scripts to run `gp_dump` to back up the monthly partitions of the historical data tables on a monthly schedule. Alternately, you can back up your Command Center database at the database level.

Maintaining the Historical Data Tables

All of the `*_history` tables stored in the Command Center database (`gpperfmon`) are partitioned into monthly partitions. A January 2010 partition is created at installation time as a template partition (it can be deleted once some current partitions are created). The Command Center agents automatically create new partitions in two month increments as needed. Administrators must periodically drop partitions for the months that are no longer needed in order to maintain the size of the Command Center database.

See the *Greenplum Database Administrator Guide* for more information on dropping partitions of a partitioned table.

Administering the Web Server

The gpmonws web server is installed in the `www` directory of your Greenplum Command Center installation.

Configuring the Web Server

The web server configuration file is stored in `$GPPERFMONHOME/instances/instance_name/webserver/conf/app.conf`. Some of the parameters in this configuration file are set by the `gpcmdr` setup utility, including the web server port and SSL options. See the *Web Server Parameters* section of [Configuration File Reference](#) for a description of the parameters in this file.

Viewing and Maintaining Web Server Log Files

Web server access and error log messages are written to `$GPPERFMONHOME/instances/<instance_name>/webserver/logs/gpmonws.log`.

If you experience errors viewing the Greenplum Command Center Console, refer to this file for more information.

To prevent the web server log from growing to excessive size, you can set up log file rotation using `logrotate` or `cronolog`.

Configuring Greenplum Command Center

Configuration parameters for Greenplum Command Center are stored in the Agent and Console configuration files.

Agent Configuration

Changes to these files require a restart of the Greenplum Database instance (`gpstop -r`).

- `$MASTER_DATA_DIRECTORY/gpperfmon/conf/gpperfmon.conf`
- `$MASTER_DATA_DIRECTORY/postgresql.conf`

Console Configuration

Changes to these files require a restart of Command Center Console (`gpcmdr --restart`).

- `$GPPERFMONHOME/instances/<instance_name>/conf/clusters.conf`
- `$GPPERFMONHOME/instances/<instance_name>/webserver/conf/app.conf`

See the [Configuration File Reference](#) section for a description of the configuration parameters in these files.

You should not need to manually edit any of the files. Running the Command Center setup utility will make all the necessary modifications to these configuration files.

Enabling Multi-Cluster Support

Multi-cluster functionality allows you to view the status of multiple Greenplum Database clusters at one time in the Command Center user interface. The clusters can be organized into groups.

Typically, one Greenplum cluster is designated the *master cluster*; its Command Center instance hosts the multi-cluster view for all Command Center-managed clusters.

See [Monitoring Multiple Greenplum Database Clusters](#) for information about accessing the multi-cluster view.

Setting Up Multiple Clusters

Multi-cluster support is enabled with a multi-cluster configuration file. There is a template for this configuration file in the instance directory of each Command Center instance at the following location:

```
$GPCCHOME/instances/<instance_name>/conf/clusters.conf
```

Locate the `clusters.conf` template on the Command Center instance you want to identify as the master instance. This will be the web server that hosts the multi-cluster web page.

The configuration file consists of the following values separated by colons:

```
SERVER : HOST : PORT : TABGROUP : AUTOLOGIN : SSL : ACCESS
```

For example:

```
Miracle:www.example.com:28080:Production:True:false:gpdb_role,accounting_role
Deforest:192.51.100.186:28080:Development:False:false
Grandalpha:grandalpha:32020:Development:False:false
```

All fields except the last, **ACCESS**, are required.

SERVER

The server value is a primary key, used to uniquely identify each Greenplum Database cluster.

The name may not contain special characters, other than the space character, underscore (`_`), or hyphen (`-`).

Command Center throws an error if there are any entries with the same primary key.

HOST

This is the host name or IP address for the cluster's Command Center UI.

PORT

The Command Center port number.

TABGROUP

This field is used to divide Greenplum Database clusters into categories; for example, **Testing**, **Production**, and **Deployment**.

AUTOLOGIN

This field enables automatic login to the cluster from the multi-cluster view. When automatic login is enabled, clicking the cluster's health chart on the multi-cluster page loads the cluster's Command Center UI in a new browser window.

The **AUTOLOGIN** field is a true or false value. It is not case-sensitive.

SSL

This field indicates whether SSL is enabled for the host. It takes a true or false value and is not case-sensitive. Any other value is an error, which will be shown in the UI.

Important: All hosts must have the same SSL configuration. SSL must be enabled or disabled for all hosts.

ACCESS

This optional field contains a comma-separated list of database roles that can see the **Multiple GPDB Clusters** view. If this field is empty, *all* users can see the view.

Securing a Greenplum Command Center Console Instance

A Greenplum Command Center Console instance can be secured by encrypting network traffic between the web server and users' browsers, authenticating Command Center users, and managing users' permissions to access Command Center features.

SSL/TLS Encryption

Greenplum Command Center supports SSL/TLS encryption to secure connections between browsers and the Command Center web server. When SSL is enabled, Command Center uses the WebSockets API, enabling long-lived, full-duplex connections, in addition to encryption.

To enable SSL, you should have a signed certificate for the Command Center web server in place when you create the Command Center instance. Place your certificate on the server where Command Center is installed, for example in the `/etc/ssl/certs` directory of the Greenplum master host. You import the certificate when you create a Command Center instance with the `gpccmdr --setup` command. The locations of the certificate and private key files are saved in

the `$GPPERFMONHOME/instances/<instance_name>/webserver/conf/app.conf` configuration file for the command center instance. See [Command Center Console Parameters](#) for details.

You can request a certificate from your organization's internal certificate authority or a commercial certificate authority, or you can use a self-signed certificate you create yourself with a cryptography suite such as OpenSSL. If you create a self-signed certificate, note that clients will have to override a security warning when they first connect to the Command Center web server.

Authentication Options

Users logging in to Greenplum Command Center are authenticated with the Greenplum Database host-based authentication system. Users can enter credentials as a user name and password or, if Kerberos authentication is configured, by authenticating with Kerberos on their workstation before browsing to the Command Center web server.

Note: Greenplum Command Center does not accept logins from the `gpadmin` user, or from local users configured with trust authentication in the `pg_hba.conf` file. Allowing trust authentication for remote logins is discouraged because it is insecure.

Database users must first be added to the Greenplum Database by using commands such as `CREATE ROLE` or `CREATE USER`. The `LOGIN` privilege is required. This example creates a login user with an encrypted password:

```
CREATE ROLE cc_user WITH LOGIN ENCRYPTED PASSWORD 'changeme';
```

The `pg_hba.conf` configuration file determines how authentication will proceed. This file contains a list of entries that are compared to attributes of the user's connection request, including the type of connection, network location of the originating host, database name, and login user name. When a match is found, the authentication method specified in the entry is applied.

The `pg_hba.conf` file can be viewed by Operators and edited by Admins in the Command Center console on the [Admin>Authentication](#) page.

The `md5` and `password` authentication methods authenticate the user name and password with the Greenplum Database `pg_roles` system table. The `md5` method requires the password to be MD5-encoded when sent over the network, so it is preferred over the `password` method, which sends the password in clear text.

The `ldap` authentication method authenticates the user name and password with an LDAP server. The LDAP server and parameters are specified in the options field of the `pg_hba.conf` entry. See the PostgreSQL [LDAP authentication](#) documentation for the format of the LDAP options.

The `gss` authentication method is used for Kerberos authentication. To use Kerberos with Command Center, Kerberos authentication must be enabled for the Greenplum Database system and the Command Center instance must also be configured. Users authenticate with the Kerberos KDC on their workstations (using `kinit`, for example) before connecting to the Command Center web server. The role name in Command Center is the user's Kerberos principal name.

For details about setting up Kerberos authentication, see [Enabling Kerberos Authentication with Greenplum Command Center](#).

See the PostgreSQL [Authentication methods](#) documentation for additional details of the authentication options.

Authorization

Command Center manages permission levels using Greenplum Database roles and groups. The Basic, Operator Basic, and Operator permission levels correspond to the `gpcc_basic`, `gpcc_operator_basic`, and `gpcc_operator` group roles in the database. The Admin permission level is conferred to roles that have the `SUPERUSER` privilege. A user who has not been added to any of the groups and does not have `SUPERUSER` privilege has the most restrictive permission level, Self Only.

Greenplum Database superusers can manage permission levels on the Command Center [Admin>Authorization](#) page. Superusers can also directly assign users roles in the database by using the `ALTER USER`, `ALTER GROUP`, and related commands to add or remove users from groups and add or remove the `SUPERUSER` privilege. If a role is configured for more than one permission level, Command Center uses the highest permission level.

Command Center users have the following capabilities, according to their permission levels:

Self Only

Users can view metrics and view and cancel their own queries.

Any Greenplum Database user successfully authenticated through the Greenplum Database authentication system can access Greenplum Command Center with Self Only permission. Higher permission levels are required to view and cancel other's queries and to access the System and Admin Control Center screens.

Basic

Allows users to view metrics, view all queries, and cancel their own queries.

Users with Basic permission are members of the Greenplum Database `gpcc_basic` group.

Operator Basic

Allows users to view metrics, view their own and others' queries, cancel their own queries, and view the System and Admin screens.

Users with Operator Read-only permission are members of the Greenplum Database `gpcc_operator_basic` group.

Operator

Allows users to view their own and others' queries, cancel their own and other's queries, and view the System and Admin screens.

Users with Operator permission are members of the Greenplum Database `gpcc_operator` group.

Admin

Allows users access to all views and capabilities in the Command Center.

Greenplum Database users with the `SUPERUSER` privilege in Greenplum Database have Superuser permissions in Command Center.

Configuring Authentication for the Command Center Console

Greenplum Command Center users are Greenplum Database users, authenticated using the standard Greenplum Database host-based authentication system. When authentication is properly configured, a user can use the same credentials to log into a database with a database client such as `psql` and into the Command Center web interface with a browser.

To create a new Command Center user, first you have to create a Greenplum Database user, then edit the Greenplum host-based authentication configuration file (`pg_hba.conf`) to give that user access to Command Center.

A Greenplum Database user who can authenticate via the `pg_hba.conf` file can log in to Greenplum Command Center and view or cancel their own queries and view metrics screens. A user's Command Center permission level determines if additional Command Center features are accessible. See [Authorization](#) for information about permissions.

Note: Greenplum Command Center does not accept logins from the `gpadmin` user, or from local users configured with trust authentication in the `pg_hba.conf` file. Allowing trust authentication for remote logins is discouraged because it is insecure.

The following are steps to create new Command Center users in an interactive `psql` session. With the exception of the `CREATE ROLE` command to create a new database user, all of these steps can be performed in the Command Center on the **Admin>Permissions** or **Admin>Authorization** screens.

See the *Greenplum Database Administrator Guide* for more detailed information about creating database users and roles.

1. Login as `gpadmin` on the master host.

2. Start `psql`:

```
$ psql
```

3. Enter the `CREATE ROLE` command to create a user:

```
# CREATE ROLE cc_user WITH LOGIN ENCRYPTED PASSWORD 'changeme';
```

To create an Admin user - a role with superuser privileges in the database and Greenplum Command Center:

```
# CREATE ROLE cc_admin WITH LOGIN ENCRYPTED PASSWORD 'changeme' SUPERUSER CREATEDB;
```

4. For users other than Admin, set the permission level by adding the user to a Command Center group role:

To create a Basic user: add the user to the `gpcc_basic` role:

```
# GRANT gpcc_basic TO cc_user;
```

To create an Operator Basic user - add the user to the `gpcc_operator_basic` role:

```
# GRANT gpcc_operator_basic TO cc_user;
```

To create an Operator user - add the user to the `gpcc_operator` role:

```
# GRANT gpcc_operator TO cc_user;
```

5. Grant permissions to a group by granting the role to the group:

```
# CREATE ROLE cc_users;
# GRANT cc_users to cc_user;
# GRANT gpcc_operator to cc_users;
```

6. Verify that roles were created successfully using the following command:

```
# \du
```

The new users you created are returned along with the attributes you specified.

7. Edit the `pg_hba.conf` file to give new users access to databases and the Command Center. Open the file in an editor:

```
$ vi $MASTER_DATA_DIRECTORY/pg_hba.conf
```

8. Scroll to the bottom of the file and insert the following lines to give the new users access from any IP address using password authentication:

```
host gpperfmon cc_user 127.0.0.1/28 md5
host gpperfmon cc_admin 127.0.0.1/28 md5
```

List additional databases the users can access after `gpperfmon`, or replace `gpperfmon` with `all` to allow the users to access any database.

Note: If you subsequently have issues logging in to Command Center it may be due to your specific environment; check the `$GPPERFMONHOME/instances/instance_name/logs/gpmonws.log` log file for authentication errors. Edit the `pg_hba.conf` file based on the error message and your specific environment.

9. Save the file and exit the editor.
10. Enter the following command to reload Greenplum Database processes.

```
# gpstop -u
```

Enabling Authentication with Kerberos

If you have enabled Kerberos authentication for Greenplum Database, you can set up Greenplum Command Center to accept connections from Kerberos-authenticated users.

Greenplum Database and Command Center include support for the Generic Security Service Applications Program Interface (GSS-API) standard. A related standard, Simple and Protected GSS-API Negotiation Mechanism (SPNEGO), describes the protocol GSS-API clients and servers use to agree on the method of authentication.

With a SPNEGO-compliant web application such as Command Center, the client and server agree on the authentication method on the client's initial HTTP request. If Kerberos authentication is not supported on both ends of the connection the server falls back to basic authentication, and displays a login form requesting a user name and password. If a user has authenticated on the workstation with Kerberos and has a valid ticket granting ticket, the web browser offers the user's credential to the Command Center web server. A Kerberos-enabled Command Center web server is configured to handle the authenticated user's connection request in one of three modes, called strict, normal, or gpmon-only.

Strict

Command Center has a Kerberos keytab file containing the Command Center service principal and a principal for every Command Center user. If the principal in the client's connection request is in the keytab file, the web server grants the client access and the web server connects to Greenplum Database using the client's principal name. If the principal is not in the keytab file, the connection request fails.

Normal

The Command Center Kerberos keytab file contains the Command Center principal and may contain principals for Command Center users. If the principal in the client's connection request is in Command Center's keytab file, it uses the client's principal for database connections. Otherwise, Command Center uses the `gpmon` user for database connections.

gpmon-only

The Command Center uses the `gpmon` database role for all Greenplum Database connections. No client principals are needed in the Command Center's keytab file.

If you have set up Kerberos authentication for Greenplum Database, most of the configuration required to enable Command Center Kerberos authentication has been done. The Command Center Kerberos configuration builds upon the Greenplum Database Kerberos setup.

Kerberos authentication can be enabled by responding to prompts when you set up a new Command Center instance with the `gpcmdr --setup` command, or you can use the `gpcmdr --krbenable <instance-name>` command to enable Kerberos for an existing Command Center instance.

Before You Begin

Kerberos authentication must be enabled for Greenplum Database. See [Using Kerberos Authentication](#) for instructions. Make sure the following prerequisites are met before you continue:

- The `krb5-workstation` package and associated libraries (`libkrb5*`) must be installed on the Greenplum master host and each client workstation.
- The date and time on the Greenplum master host and all client workstations must be synchronized with the KDC.
- The `krb5.conf` configuration file must be the same on the KDC host, the Greenplum master host, and client workstations.
- The KDC database must have a service principal for Greenplum Database. The default service name for Greenplum Database is `postgres/<master-host>@<realm>`. You can choose a service name other than `postgres`, but it must match the value of the `krb_srvname` parameter in the `$MASTER_DATA_DIRECTORY/postgresql.conf` file.
- A keytab file with the Greenplum Database principal must be installed on the Greenplum master host and identified by the `krb_server_keyfile` parameter in the `$MASTER_DATA_DIRECTORY/postgresql.conf` file.
- Each client workstation must have a keytab file containing their Kerberos principal, `<username>@<realm>`.

Add Command Center Principals to the KDC Database

Before you configure a Command Center instance for Kerberos authentication, you must create the required Kerberos principals. All of the principals used with Command Center are created in the Greenplum Database Kerberos realm. Command Center users can use the same Kerberos principal to log in to Command Center and Greenplum Database.

Command Center Service Principal

A service principal is needed for the Command Center web server. This principal has the format `HTTP/<host>@<realm>`. For example, if users access Command Center at the URL `http://mdw.example.com:28080`, the `<host>` part of the service key is `mdw.example.com` and the `<realm>` part is the Greenplum Database Kerberos realm, for example `GPDB-KRB.EXAMPLE.COM`.

Note that Kerberos authentication only works if Command Center users enter the host in the same format specified in the Kerberos service principal. If the principal specifies the FQDN, for example, using the host's IP address in the browser URL will not work; the web server will fall back to basic authentication, presenting a login screen.

Greenplum Database gpmon User

Command Center uses the `gpmon` Greenplum role to access the `gpservermon` database, which contains data presented in the Command Center UI.

You can choose to authenticate the `gpmon` user with Kerberos or with basic authentication. To use Kerberos, you must create a principal for the `gpmon` user.

If you choose to use basic authentication you do not need a Kerberos principal for the `gpmon` user. The `gpmon` user will authenticate with Greenplum Database using the password saved in the `~gpadmin/.pgpass` file on the host running the Command Center instance. See [Changing the gpmon Password](#) for instructions to manage the `gpmon` password.

Command Center Users

Add Kerberos principals for any Command Center users who do not already have principals in the KDC for Greenplum Database.

Adding Kerberos Principals

To add the required principals, perform the following steps as root on the KDC server.

1. Start `kadmin.local`.

```
kadmin.local
```

2. Add a principal for the Command Center web service. Be sure to specify the `<gpcc-host>` in the same format that users should enter the host in their browsers.

```
kadmin.local: addprinc HTTP/<gpcc-host>@<realm>
```

3. If you want the `gpmon` database user to use Kerberos authentication, add a `gpmon` principal.

```
kadmin.local: addprinc gpmon@<realm>
```

4. Add principals for any new Command Center users.

```
kadmin.local: addprinc cc_user1@<realm>
```

Repeat for each new Command Center user.

5. Enter `quit` to exit `kadmin.local`.

Set Up Keytab Files

After you have created all of the Kerberos principals needed, you create and distribute keytab files. Keytab files contain Kerberos principals and encrypted keys based on the principals' Kerberos passwords. Keytab files are needed for Greenplum Database, the Command Center instance, and each Command Center and database user.

The Command Center instance is usually installed on the Greenplum master and, when this is true, a single keytab file can be shared by Greenplum Database and the Command Center instance. Running Command Center on the Greenplum master is recommended, since it confines authentication with the database to a single host.

If you install the Command Center instance on a host other than the Greenplum master, you will need to create a separate keytab file. Make sure this keytab can be accessed by the operating system user running Greenplum Command Center.

You must also create a keytab file for each Greenplum Database or Command Center user containing just the user's principal. This keytab file is installed

on the user's workstation to enable the user to authenticate to Kerberos.

Command Center Instance on the Greenplum Master Host

If the Greenplum Command Center web server is running on the Greenplum Database master host, Command Center can share the Greenplum Database keytab file. You need to create a keytab file that contains the following principals:

- Service key for the `postgres` process on the Greenplum Database master host, for example `postgres/mdw.example.com@GPDB.EXAMPLE.COM`.
- Service key created for Command Center in the previous section, for example `HTTP/mdw.example.com@GPDB.EXAMPLE.COM`.
- A principal for every Kerberos-authenticated Greenplum Database or Command Center user.

All service keys and principals should be in the Greenplum Database realm.

To create a keytab file for Greenplum Database and Command Center, perform the following steps as root on the KDC server.

1. Start `kadmin.local`.

```
kadmin.local
```

2. Create a keytab file and add the Greenplum Database service key, the command center service key, and all database and Command Center users.

```
kadmin.local: ktadd -k gpdb-kerberos.keytab postgres/mdw.example.com@GPDB.EXAMPLE.COM HTTP/mdw.example.com@GPDB.EXAMPLE.COM
```

You can enter one or more principals with each `ktadd` command. You can specify a wildcard using the `-glob` option. For example this command adds all principals in the `GPDB.EXAMPLE.COM` realm, including service principals and admin users.

```
kadmin.local: ktadd -k gpdb-kerberos.keytab -glob *@GPDB.EXAMPLE.COM
```

3. Enter `quit` to exit `kadmin.local`.
4. Copy the keytab you created to the Greenplum Database master host, replacing the old keytab file. The location of the file is given by the `krb_server_keyfile` parameter in the `$MASTER_DATA_FILE/postgresql.conf` file. Set the permissions on the file so that it can be read only by the `gpadmin` user.
5. Update any entries required for new Greenplum Database principals in the `pg_hba.conf` file and `pg_ident.conf` files. See [Update the Greenplum Database pg_hba.conf File](#) for details.

Command Center Instance on a Separate Host

If the Command Center web server is on a different host than the Greenplum Database master, you need separate keytab files for Greenplum Database and Command Center. The keytab file for Greenplum Database may not require any updates, but you will need to create a keytab file for Command Center.

- The Greenplum Database keytab file must contain the Greenplum Database service key and all principals for users with database access.
- The Command Center keytab file contains the Command Center service key and principals for users that have Command Center access. Users with Command Center access must also have Greenplum Database access, so user principals in the Command Center keytab file must also be in the Greenplum Database keytab file.

Update the Greenplum Database keytab if you created new database roles and principals for Command Center. For example, if you want to use Kerberos authentication for the `gpmon` user, you must create a principal and add it to both the Greenplum Database and Command Center keytab files.

To create the keytab file for Command Center, perform the following steps as root on the KDC host.

1. Start `kadmin.local`.

```
kadmin.local
```

2. Create a keytab file and add the Command Center service key.

```
kadmin.local: ktadd -k gpcc-kerberos.keytab HTTP/mdw.example.com@GPDB.EXAMPLE.COM
```

- If you want to authenticate the `gpmon` user with Kerberos, add the `gpmon` principal.

```
kadmin.local: ktadd -k gpcc-kerberos.keytab gpmon@GPDB.EXAMPLE.COM
```

- Add principals for all Command Center users:

```
kadmin.local: ktadd -k gpcc-kerberos.keytab cc_user1@GPDB.EXAMPLE.COM cc_user2@GPDB.EXAMPLE.COM
```

You can enter one or more principals with each `ktadd` command.

- Enter `quit` to exit `kadmin.local`.
- Copy the keytab you created to the the host running Command Center, for example:

```
$ scp gpcc-kerberos.keytab gpadmin@<host-name>:/home/gpadmin
```

- Update any entries required for new principals in the `pg_hba.conf` file and `pg_ident.conf` files on the Greenplum master. See [Update the Greenplum Database pg_hba.conf File](#).

Update the Greenplum Database pg_hba.conf File

The Greenplum Database `$MASTER_DATA_DIRECTORY/pg_hba.conf` configuration file determines which authentication methods to use to allow database access.

If you created new Command Center users, you may need to add an entry to allow access via Command Center. The entry for an individual user has this format:

```
host database <user-name> <gpcc CIDR> gss [options]
```

Authentication for the `gpmon` user needs to be set up in the `pg_hba.conf` file in one of the following ways.

Basic authentication

The `/home/gpadmin/.gpss` file contains the password for `gpmon` to use. See [Changing the gpmon Password](#) for details. An entry in the `pg_hba.conf` file specifies the md5 authentication method for `gpmon`:

```
local all gpmon md5
```

Trust authentication

On the Greenplum Database host only, the `gpmon` user can access databases without authentication:

```
local all gpmon trust
```

The `/home/gpadmin/.gpss` file is not needed.

Kerberos authentication

A Kerberos principal has been created for the `gpmon` user and added to the Greenplum Database and Command Center keytab files.

```
host all gpmon <gpcc CIDR>] gss [options]
```

Remove any existing reject rules for `gpmon`:

```
host all gpmon <auth-method> reject
```

See [Using Kerberos Authentication](#) for more information about the `pg_hba.conf` file.

Enable Kerberos for the Command Center Instance

Set up the Command Center instance to use the Command Center keytab file you created.

If you are creating a *new* Command Center instance with the `gpcmdr --setup` command, answer `Y` to the prompt `Enable kerberos login for this instance?`, and enter the Command Center host name and path to the keytab file at the prompts. See [Create the Greenplum Command Center Instance](#) for complete instructions.

If you are adding Kerberos authentication to an existing Command Center instance, use the `gpcmdr --krbenable <instance-name>` command. For example, if your Command Center instance is named `my-gpcc` enter this command:

```
$ gpcmdr --krbenable my-gpcc
```

Enter the Command Center host name and path to the keytab file at the prompts. See the [gpcmdr Reference](#) for more information.

Authenticating With Kerberos on the Client Workstation

To use Kerberos Command Center authentication, the user must have authenticated with Kerberos using the `kinit` command-line tool.

The user then accesses the Command Center web server with a URL containing the host name in the format specified in the Command Center service principal and the port number, for example `http://gpcc.example.com:28080`.

The web browser must be configured to use the SPNEGO protocol so that it offers the user's Kerberos principal to the web browser. The method for configuring web browsers varies with different browsers and operating systems. Search online to find instructions to set up your browser and OS.

Securing the gpmon Database User

The Greenplum Database `gpmon` user is a superuser role used to manage the `gpperfmon` database. The `gpperfmon_install` utility, which must be run once before you can create a Command Center Console instance, creates the `gpmon` role.

Greenplum Database uses the `gpmon` role to update the `gpperfmon` database with data collected by agents running on the segment hosts. The Command Center web server uses the `gpmon` role to connect to the `gpperfmon` database as well as databases monitored by the Command Center instance.

When `gpperfmon_install` creates the `gpmon` role, it prompts for a password, which it then adds to the `.pgpass` file in the `gpadmin` user's home directory. The entry in the `.pgpass` file is similar to the following:

```
*:5432:gpperfmon:gpmon:changeme
```

See [The Password File](#) in the PostgreSQL documentation for details about the `.pgpass` file.

In the `$MASTER_DATA_DIRECTORY/pg_hba.conf` authentication file, `gpperfmon_install` creates these entries:

```
local gpperfmon gpmon md5
host all gpmon 127.0.0.1/28 md5
host all gpmon ::1/128 md5
```

 The `gpperfmon_install` utility in Greenplum 4.3.x does not create the IPV6 entry. If IPV6 is enabled in you Greenplum cluster you may need to add this entry.

If you authenticate users with Kerberos, you can also set up Kerberos authentication for the `gpmon` role on the Greenplum master and standby hosts. Kerberos authentication is supported with TCP connections only; `local` entries use Linux sockets and authenticate with the `.pgpass` file password, even if you have enabled Kerberos for `host` entries.

Changing the gpmon Password

To change the `gpmon` password, follow these steps:

1. Log in to Greenplum Database as a superuser and change the `gpmon` password with the `ALTER ROLE` command:

```
# ALTER ROLE gpmon WITH ENCRYPTED PASSWORD 'new_password';
```

2. On the Greenplum master host, update the password in the `.pgpass` file in the `gpadmin` home directory (`~/.pgpass`). Replace the existing password in the line or lines for `gpmon` with the new password.

```
*:5432:gpperfmon:gpmon:new_password
```

3. Ensure that the `.pgpass` file is owned by `gpadmin` and RW-accessible by `gpadmin` only.

```
$ chown gpadmin:gpadmin ~/.pgpass
$ chmod 600 ~/.pgpass
```

4. Restart Greenplum Command Center with the `gpcmdr` utility.

```
$ gpcmdr --restart
```

 If Command Center is installed on the Greenplum standby master host or on a remote host other than the Greenplum master, be sure to also update the `.pgpass` file on those hosts.

Authenticating gpmon with Kerberos

If you authenticate Greenplum Database and Command Center users with Kerberos, you can also authenticate the `gpmon` user with Kerberos.

1. On the KDC, create a keytab file containing the Kerberos principal for the `gpmon` user, just as you would for any Kerberos-authenticated client. Install the file on the Greenplum master and standby hosts.
2. Update the entries for `gpmon` in the `$MASTER_DATA_DIRECTORY/pg_hba.conf` file to use the `gss` authentication method.

```
host all gpmon 0.0.0.0/0 gss include_realm=0 krb_realm=GPDB.EXAMPLE.COM
```

Note that `local` entries in `pg_hba.conf` cannot be authenticated with Kerberos. If there is a `local` entry for the `gpmon` user, it will use the `.pgpass` file to authenticate with the database. See [The pg_hba.conf file](#) in the PostgreSQL documentation for complete `pg_hba.conf` file documentation.

3. Log in to the master host as `gpadmin` and authenticate the `gpmon` user.

```
$ kinit gpmon
```

4. Create the Kerberos-enabled Command Center Console instance. See [Creating Greenplum Command Center Console Instances](#) for steps to create an instance.

Utility Reference

Reference information for the Greenplum Command Center utility programs.

- [gpperfmon-install](#) - creates the `gpperfmon` database and enables the data collection agents
- [gpcmdr](#) - sets up and manages the Greenplum Command Center web application
- [gpccinstall](#) - installs the Greenplum Command Center software on additional hosts

gpcmdr

Configures and manages instances of the Command Center Console.

```
gpcmdr [--ssh_full_path <path>]
  --setup [[<section_header>] --config_file <path>]
  | --start [<instance_name>]
  | --stop [<instance_name>]
  | --restart [<instance_name>]
  | --migrate [<instance_name>]
  | --remove [<instance_name>]
  | --status [<instance_name>]
```

Option	Description
<code>--setup</code>	Configures console components on the installation host. With this option, <code>gpcmdr</code> prompts for values to configure the components and writes the values to <code>app.conf</code> . For more information on these configuration parameters, see Configuration File Reference .
<code>--config_file</code>	Sets the path to a configuration file to use to set up new Command Center instances. This option must be used with the <code>--setup</code> option. See Setup Configuration File for information about the format and content of this configuration file. If <code>section_header</code> is supplied, <code>gpcmdr</code> only sets up the instance defined in the named section in the configuration file. Otherwise, <code>gpcmdr</code> sets up all instances in the configuration file.
<code>--start</code>	Starts the specified instance (or all instances by default) and its associated web service.
<code>--stop</code>	Stops the specified instance (or all instances by default) and its associated web service.
<code>--migrate</code>	Copies Command Center instances (or all instances by default) from a previous installation.
<code>--remove</code>	Removes the specified instance and its associated database schema.
<code>--restart</code>	Restarts the specified instance (or all instances by default) and its associated web service.
<code>--status</code>	Displays the status, either <code>Running</code> or <code>Stopped</code> , of the web service.
<code>--version</code>	Displays the version of the <code>gpcmdr</code> utility.
<code>--ssh_full_path</code>	Sets the full path to the ssh command. Use this to override the ssh command found on the path.

Description

The `gpcmdr` utility sets up and configures Command Center Console instances, starts and stops instances, and provides status information.

You can set up a new Command Center Console instance interactively or, by providing a configuration file, non-interactively.

For actions `--start`, `--stop`, `--restart`, `--migrate`, and `--status` you can specify a console instance name. If you do not specify a name, the action applies to all existing console instances.

On the `--start` option, `gpcmdr` creates the `gpcc_basic`, `gpcc_operator_basic`, and `gpcc_operator` database roles if they do not already exist.

The `--migrate` option prompts you to enter the path to the Command Center installation with instances you want to migrate. The utility checks whether the instance to copy already exists in the new location before copying. If the instance exists in the new location, a prompt asks whether you want to overwrite the instance.

Examples

Interactively create a new Command Center Console instance:

```
$ gpcmdr --setup
```

Set up the Command Center Console instance defined in the `[development]` section of a configuration file:

```
$ gpcmdr --setup development gpccinstances.cfg
```

Check the status of all Command Center Console instances:

```
$ gpcmdr --status
```

gpccinstall

Installs the Greenplum Command Center software on additional hosts.

```
gpccinstall { --help | -h | -? }
gpccinstall { --file | -f } <hostfile>
```

<code>--help</code> <code>-h</code>	Displays usage information.
<code>--file</code> <code>-f</code> <i>hostfile</i>	Specifies the path to a text file containing a list of hosts where the Command Center software is to be installed.

Description

Use the `gpccinstall` utility to install the Greenplum Command Center software on additional hosts, for example on the Greenplum standby master host.

The `gpccinstall` utility should be run by the `gpadmin` user. The Command Center software must be installed on the current host and the `$GPPERFMONHOME` environment variable must be set to the directory where the Command Center software is installed. The software will be installed to the same directory on the target hosts. The `gpadmin` user must have write permission in the parent directory of the Command Center installation directory on all hosts, that is, `$GPPERFMONHOME/..`.

The `<hostfile>` argument is the name of a text file containing a list of hosts, one host name per line, where you want to install the software. The hosts must be set up for passwordless ssh access. The `gpadmin` user must have write access to the directory where the software is to be installed.

Example

The following example installs Greenplum Command Center software on hosts named `sdw` and `gp spare`:

```
# Create a hosts file
$ echo sdw > hosts.txt
$ echo gp spare >> hosts.txt
# Install GPCC on the hosts
gpccinstall -f hosts.txt
```

Configuration File Reference

References for Greenplum Command Center configuration files.

Configuration parameters for Greenplum Command Center are stored in the following files:

```
$MASTER_DATA_DIRECTORY/gpperfmon/conf/gpperfmon.conf
```

Stores configuration parameters for the Greenplum Command Center agents.

```
$GPPERFMONHOME/instances/<instance_name>/webserver/conf/app.conf
```

Stores configuration parameters for the Command Center web application and web server.

```
$MASTER_DATA_DIRECTORY/postgresql.conf
```

Stores configuration parameters to enable the Greenplum Command Center feature for Greenplum Database server.

```
$GPPERFMONHOME/bin/ssh-wrapper
```

Greenplum Command Center normally finds the `ssh` command on the path. If your environment has an incompatible implementation of this command on the path, you can provide the absolute path to your version in the `ssh-wrapper` script, located at

```
$GPPERFMONHOME/bin/ssh-wrapper .
```

For example:

```
ssh="/home/me/bin/myssh"
```

Any system user with write permissions to these directories can edit these configuration files.

Command Center Agent Parameters

The `$MASTER_DATA_DIRECTORY/gpperfmon/conf/gpperfmon.conf` file stores configuration parameters for the Command Center agents. For configuration changes to these options to take effect, you must save `gpperfmon.conf` and then restart Greenplum Database server (`gpstop -r`).

To enable the Command Center agents within Greenplum Database server, you must also set the Greenplum Database Server Configuration Parameters. See the [gppermon_install](#) reference for details.

log_location

Specifies a directory location for Command Center log files. Default is `$MASTER_DATA_DIRECTORY/gpperfmon/logs`.

min_query_time

Specifies the minimum query run time in seconds for statistics collection. Command Center logs all queries that run longer than this value in the `queries_history` table. For queries with shorter run times, no historical data is collected. Defaults to 20 seconds.

If you know that you want to collect data for all queries, you can set this parameter to a low value. Setting the minimum query run time to zero, however, collects data even for the numerous queries run by Command Center itself, creating a large amount of data that may not be useful.

min_detailed_query_time

Specifies the minimum iterator run time in seconds for statistics collection. Command Center logs all iterators that run longer than this value in the `iterators_history` table. For iterators with shorter run times, no data is collected. Minimum value is 10 seconds.

This parameter's value must always be equal to, or greater than, the value of `min_query_time`. Setting `min_detailed_query_time` higher than `min_query_time` allows you to log detailed query plan iterator data only for especially complex, long-running queries, while still logging basic query data for shorter queries.

Given the complexity and size of iterator data, you may want to adjust this parameter according to the size of data collected. If the `iterators_*` tables are growing to excessive size without providing useful information, you can raise the value of this parameter to log iterator detail for fewer queries.

max_log_size

This parameter is not included in `gpperfmon.conf`, but it may be added to this file for use with Greenplum Command Center.

To prevent the log files from growing to excessive size, you can add the `max_log_size` parameter to `gpperfmon.conf`. The value of this parameter is measured in bytes. For example:

```
max_log_size = 10485760
```

With this setting, the log files will grow to 10MB before the system rolls over to a new log file.

partition_age

The number of months that Greenplum Command Center statistics data will be retained. The default is 0, which means we won't drop any data.

quantum

Specifies the time in seconds between updates from Command Center agents on all segments. Valid values are 10, 15, 20, 30, and 60. Defaults to 15 seconds.

If you prefer a less granular view of performance, or want to collect and analyze minimal amounts of data for system metrics, choose a higher quantum. To collect data more frequently, choose a lower value.

ignore_qexec_packet

When set to true, Greenplum Command Center agents do not collect performance data in the `gpperfmon` database `queries_*` tables:

`rows_out`, `cpu_elapsed`, `cpu_currpct`, `skew_cpu`, and `skew_rows`. The default setting, true, reduces the amount of memory consumed by the `gpmmon` process. Set this parameter to false if you require this additional performance data.

smdw_aliases

This parameter allows you to specify additional host names for the standby master. For example, if the standby master has two NICs, you can enter:

```
smdw_aliases=smdw-1,smdw-2
```

This optional fault tolerance parameter is useful if the Greenplum Command Center loses connectivity with the standby master. Instead of continuously retrying to connect to host `smdw`, it will try to connect to the NIC-based aliases of `smdw-1` and/or `smdw-2`. This ensures that the Command Center Console can continuously poll and monitor the standby master.

Command Center Console Parameters

Each instance of the Command Center Console has a configuration file located at `$GPPERFMONHOME/instances/<instance_name>/webserver/conf/app.conf`.

After editing this file, reload the configuration by restarting the Command Center Console instance (`gpperfmon --restart <instance_name>`).

`appname = gpmonws`

The web server binary file. Do not change.

`listentcp4 = [true | false]`

When `true`, the address type is tcp4. The default is `true`.

`runmode = [prod | dev | test]`

The application mode, which can be `dev`, `prod` or `test`. The default is `dev`. In `dev` mode Command Center shows user friendly error pages. User friendly error pages are not rendered in `prod` mode.

`session = [true | false]`

Use sessions to manage user experience. The default is `true`. Sessions are stored in memory.

`enablexsrf = [true | false]`

Enable CSRF protection.

`xsrffix = <seconds>`

CSRF expire time. The default is `43200` seconds.

`xsrftoken = <token_string>`

The CSRF token.

`rendertype = json`

The render type of web server. Do not change.

`printallsqls = [true | false]`

Print all backend gpperfmon SQL to the web server console. The default is `false`.

`sessionname = webserver_gpperfmon_instance_<instance_name>`

The name of the session in Greenplum Database. Do not change.

`display_name = <display_name>`

The display name for console.

`master_host = <hostname>`

The Greenplum Database host name. The default is `localhost`.

`master_port = <port>`

The Greenplum Database master port. The default is `5432`.

`HTTPSCertFile = </path/to/cert.pem>`

The full path to the server's SSL certificate, if SSL is enabled.

`HTTPSKeyFile = </path/to/cert.pem>`

The server's private key file if SSL is enabled.

`EnableHTTPS = [true | false]`

Enable listening on the secure SSL port. The default is `true`.

`EnableHTTP = [true | false]`

Enable listening on the HTTP port. Default is `false`.

`httpsport = [port]`

The web server port. The default is 28080.

Setup Configuration File

A setup configuration file contains properties used to define one or more Greenplum Command Center instances when `gpccmdr` is run with the `--config_file` option.

The configuration file uses the Python configuration file format, similar to the Microsoft INI file format. The file contains sections, introduced by a `[section]` header, and followed by `name: value` or `name=value` entries, one per line. Comments begin with a `#` or `;` character and continue through the end of the line. A `[DEFAULT]` section sets default values for parameters that may be overridden in other sections.

See [Setting Up Command Center Instances with a Configuration File](#) for more information.

Parameters

`remote_db`

`True` if the instance is to run on a different host. Default: `False`.

`master_host`

The name of the host where the Greenplum Command Center Console is to be set up, if `remote_db` is `True`.

`instance_name`

The name of the instance to set up. This will become the name of a subdirectory in the `instances` directory where the instances configuration and log files are stored. Instance names may contain letters, digits, and underscores and are not case sensitive.

`display_name`

The name to display for the instance in the Command Center user interface. Display names may contain letters, digits, and underscores and *are* case sensitive. `instance_name` is used for `display_name` if this parameter is not provided.

`master_port`

The Greenplum Database master port. Default: `5432`.

`webservice_port`

The listen port for the Command Center `gpmomws` server. This port must be different for each Command Center Console instance on the host. Default: `28080`.

`enable_ssl`

`True` if client connections to the Command Center web server should be secured with SSL. Default: `False`.

`ssl_cert_file`

If `enable_ssl` is `True`, set this parameter to the full path to a valid certificate in PEM file format.

`enable_kerberos`

Set to `True` to enable Kerberos authentication.

`webservice_url`

The web server hostname, from the Kerberos HTTP service principal.

`keytab`

Path to the keytab file containing Kerberos principals for the Command Center web server and users.

`enable_copy_standby`

Set to `True` to have `gpccmdr` install the instance configuration on the Greenplum standby master host.

`standby_master_host`

The name of the Greenplum standby master host. Required when `enable_copy_standby` is `True`.

Examples

This example configuration sets up two Command Center instances, `prod` and `dev`. Parameters in the `[DEFAULT]` section apply to all instances and may be overridden by parameters in the `[production]` and `[development]` sections.

```
[DEFAULT]
remote_db=false
master_port=5432
ssl_cert_file=/tmp/cert.pem

[development]
instance_name=development
enable_copy_standby=false
webserver_port=28080
enable_ssl=false

[production]
instance_name=production
display_name=OurProduction
remote_db=true
master_host=mdw
master_port=15432
webserver_port=28081
enable_ssl=true
enable_copy_standby=true
standby_master_host=smdw
```

This example configuration has only one instance, so the `[DEFAULT]` section and section headers are not necessary.

```
instance_name=single
display_name=SINGLE
remote_db=true
master_host=10.152.10.149
master_port=5432
webserver_port=28082
enable_ssl=true
ssl_cert_file=/tmp/cert.pem
enable_copy_standby = true
standby_master_host=192.0.2.156
```

Greenplum Database Server Configuration Parameters

The following parameters must be uncommented and set in the server configuration file (`postgresql.conf`) in order to enable the Command Center data collection agents:

- `gp_enable_gpperfmon` and `gpperfmon_port` must be set in both the master and segment `postgresql.conf` files.
- `gp_enable_gpperfmon` and `gp_enable_gpperfmon` only need to be set in the master `postgresql.conf` file.

After changing these settings, the Greenplum Database instance must be restarted for the changes to take effect.

`gp_enable_gpperfmon`

Turns on the Command Center data collection agent for a segment. Must be set in all `postgresql.conf` files (master and all segments).

`gpperfmon_port`

The default port for the Command Center agents is 8888, but you can set this parameter to a different port if required (master and all segments).

`gp_gpperfmon_send_interval`

Sets the frequency in seconds that the Greenplum Database server processes send query execution updates to the Command Center agent processes.

`gp_external_enable_exec`

This parameter is enabled by default and must remain enabled. It allows the use of external tables that execute OS commands or scripts on the segment hosts. The Command Center agents use this type of external tables to collect current system metrics from the segments.

`gpperfmon_log_alert_level`

Controls which message levels are written to the `gpperfmon` log. Each level includes all the levels that follow it. The later the level, the fewer messages are sent to the log. The default value is `warning`.