

Pivotal Command Center

Version 2.1

User Guide

Rev: A01

Copyright © 2013 GoPivotal, Inc. All rights reserved.

GoPivotal, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." GOPIVOTAL, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.

Use of Open Source

This product may be distributed with open source code, licensed to you in accordance with the applicable open source license. If you would like a copy of any such source code, Pivotal will provide a copy of the source code that is required to be made available in accordance with the applicable open source license. Pivotal may charge reasonable shipping and handling charges for such distribution.

Revised November 2013

Table of Contents

About Pivotal, Inc.....	1
About this Guide	1
Document Conventions	1
Text Conventions.....	2
Command Syntax Conventions	2
Chapter 1: Overview	4
Pivotal Command Center Overview	4
Pivotal Command Center UI and CLI.....	4
Performance Monitor (nmon).....	5
PostgreSQL Database	5
Architectural Overview	6
Chapter 2: Installing Pivotal Command Center	7
Quick Install.....	7
Supported Platforms	9
Browser Support	9
Product Downloads.....	9
Prerequisites	9
Package Accessibility.....	10
Installation Instructions	11
Install Command Center	11
Install PHD Services	13
Enable the PHD Services.....	13
Launching Pivotal Command Center	14
Next Steps.....	14
Uninstalling Pivotal Command Center.....	14
Upgrading Pivotal Command Center	15
Chapter 3: Using Pivotal Command Center UI	16
Overview.....	16
Logging In.....	17
Login Screen.....	17
Cluster Status Page.....	17
Settings Menu.....	18
Configuring and Deploying a Cluster	18
Post Installation for HAWQ.....	22
Starting the Cluster	23
Initializing HAWQ.....	23
Dashboard	24
Cluster Analysis.....	26
MapReduce Job Monitor.....	28
Job Details.....	29
YARN App Monitor	31
HAWQ Query Monitor	32
Appendix A: Creating a YUM EPEL Repository	33
Appendix B: Best Practises and Troubleshooting	34
nmon Issues.....	34

Job Monitor Page	34
HTTPS Issues.....	34

Preface

This preface includes four sections:

- [About Pivotal, Inc.](#)
- [About this Guide](#)
- [Document Conventions](#)

About Pivotal, Inc.

Greenplum is currently transitioning to a new corporate identity (Pivotal, Inc.). We estimate that this transition will be completed in 2013. During this transition, there will be some legacy instances of our former corporate identity (Greenplum) appearing in our products and documentation. If you have any questions or concerns, please do not hesitate to contact us through our web site,

<http://gopivotal.com/about-pivotal/support>.

About this Guide

This guide includes the following topics:

- [Chapter 1, “Overview”](#)— An overview of the functionality and architecture of Pivotal Command Center.
- [Chapter 2, “Installing Pivotal Command Center”](#)— Instructions to install Pivotal Command Center.
- [Chapter 3, “Using Pivotal Command Center UI”](#)— An overview of the functionality of the Pivotal Command Center console (user interface).
- [Appendix A, “Creating a YUM EPEL Repository”](#)— Instructions how to setup a local yum repository or point your hosts to an EPEL repository.
- [Appendix B, “Best Practises and Troubleshooting”](#).

Document Conventions

The following conventions are used throughout the Pivotal Command Center documentation to help you identify certain types of information.

- [Text Conventions](#)
- [Command Syntax Conventions](#)

Text Conventions

Table 0.1 Text Conventions

Text Convention	Usage	Examples
<i>italics</i>	New terms where they are defined Database objects, such as schema, table, or columns names	The <i>master instance</i> is the postgres process that accepts client connections. Catalog information for Pivotal Command Center resides in the <i>pg_catalog</i> schema.
monospace	File names and path names Programs and executables Command names and syntax Parameter names	Edit the <code>postgresql.conf</code> file. Use <code>gpstart</code> to start Pivotal Command Center.
<monospace italics>	Variable information within file paths and file names Variable information within command syntax	<code>/home/gpadmin/<config_file></code> <code>COPY tablename FROM '<filename>'</code>
monospace bold	Used to call attention to a particular part of a command, parameter, or code snippet.	Change the host name, port, and database name in the JDBC connection URL: <code>jdbc:postgresql://host:5432/m ydb</code>
UPPERCASE	Environment variables SQL commands Keyboard keys	Make sure that the Java <code>/bin</code> directory is in your <code>\$PATH</code> . <code>SELECT * FROM my_table;</code> Press <code>CTRL+C</code> to escape.

Command Syntax Conventions

Table 0.2 Command Syntax Conventions

Text Convention	Usage	Examples
{ }	Within command syntax, curly braces group related command options. Do not type the curly braces.	<code>FROM { 'filename' STDIN }</code>
[]	Within command syntax, square brackets denote optional arguments. Do not type the brackets.	<code>TRUNCATE [TABLE] name</code>

Table 0.2 Command Syntax Conventions

Text Convention	Usage	Examples
...	Within command syntax, an ellipsis denotes repetition of a command, variable, or option. Do not type the ellipsis.	DROP TABLE <i>name</i> [, ...]
	Within command syntax, the pipe symbol denotes an “OR” relationship. Do not type the pipe symbol.	VACUUM [FULL FREEZE]

1. Overview

This document is a User Guide for the Pivotal Command Center (PCC) User Interface.

This chapter provides a brief overview of Pivotal Command Center.

- [Pivotal Command Center Overview](#)
 - [Performance Monitor \(nmon\)](#)
 - [PostgreSQL Database](#)
- [Architectural Overview](#)

Pivotal Command Center Overview

The Pivotal Command Center (PCC) allows an administrative user to configure, deploy, administer, and monitor one or more Pivotal HD clusters. The Command Center has both a graphical user interface and command-line tools to deploy and configure, monitor, and administer Pivotal HD clusters.

For UI operations, see [Chapter 3, “Using Pivotal Command Center UI”](#)

For command line operations, see the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide*.

This release of Command Center allows administering and monitoring of only Pivotal HD Enterprise 1.1 clusters.

PCC provides complete life cycle management for Pivotal HD Clusters. It performs the following two main groups of functions:

- Cluster configuration and deployment
- Cluster monitoring and management.

These functions are served through a set of RESTful web services that run as a web application on Jetty server on the Command Center admin host. This is called `gphdmgr-webservices`. This web application stores its metadata and cluster configuration for Pivotal HD cluster nodes and services in the Pivotal Command Center PostgreSQL database. It makes use of a Puppet Server to perform most of its HD cluster installation and configuration. It also has a polling service that retrieves Hadoop metrics from the cluster and stores them in the Command Center PostgreSQL Database at periodic intervals.

Pivotal Command Center UI and CLI

The PCC UI provides the user with a single web-based graphical user interface to configure and deploy, monitor and manage one or more Pivotal HD cluster. This web application is hosted on a Ruby-on-Rails application which presents the status and metrics of the clusters. The system metrics data is gathered by the Performance Monitor (nmon) component. The Command Center UI invokes the

`gphdmgr-webservice` APIs to retrieve all Hadoop-specific cluster metrics and status information. This includes the Hadoop metrics that was previously retrieved by the polling service.

PCC provides a command-line interface (CLI) for more advanced users to perform installation, configuration and uninstalls. This tool invokes the `gphdmgr-webservice` APIs to install and configure the various Pivotal HD services. The CLI also provides a way to perform other administrative actions such as starting and stopping clusters. For how to use this CLI, please refer to the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide*.

Performance Monitor (nmon)

Pivotal Command Center comes with a Performance Monitor called `nmon` (for node monitor). This makes use of a highly scalable message passing architecture to gather performance metrics from each node that Command Center monitors. This consists of a `nmon` master daemon that runs on the Command Center admin host and an `nmon` daemon that runs on all the cluster nodes that report system metric information to the `nmon` master. This includes metrics such as CPU, memory, disk I/O and network usage information.

The `nmon` master on the admin host dumps the system metrics it receives from the `nmon` agents on the cluster nodes into a PostgreSQL DB. This is then queried by the Command Center UI application to display its cluster analysis graphs.

The `nmon` agents hosts are deployed throughout the cluster during Pivotal HD cluster deployment itself (see *Pivotal HD Enterprise 1.1 Installation and Administrator Guide* for details).

The agents are deployed as services on each host, including on the Pivotal Command Center admin host. To stop or start the `nmon` service run the following as root:

```
# service nmon stop
# service nmon start
```

PostgreSQL Database

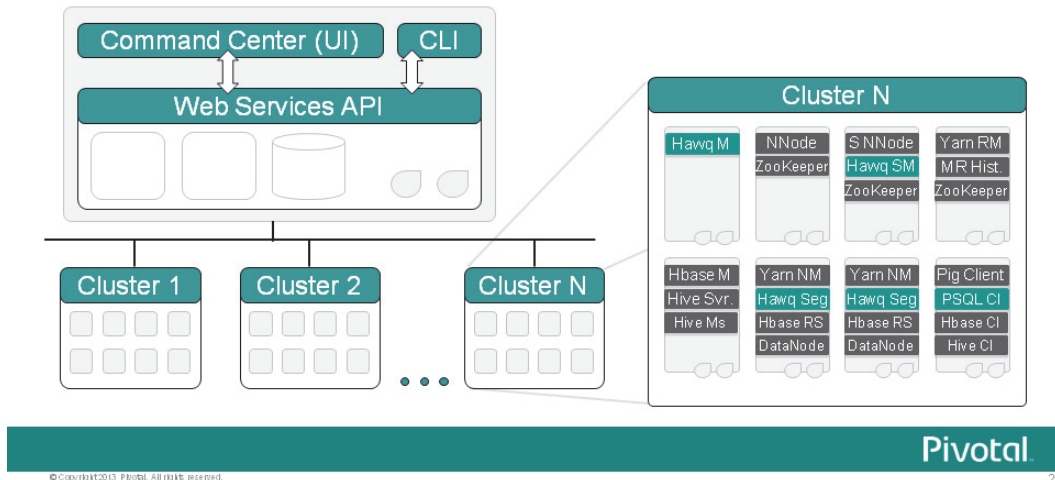
Pivotal Command Center makes use of a PostgreSQL Database to store the following:

- Cluster configurations
- Hadoop cluster metrics
- System metrics of the cluster
- Pivotal Command Center Metadata

Architectural Overview

For more details about Pivotal HD Enterprise 1.1, refer to the *Pivotal HD 1.1 Installation and Administrator Guide*.

Pivotal Command Center - Architecture



2. Installing Pivotal Command Center

This section describes how to install and configure Pivotal HD 2.1.x using the Pivotal Command Center Unified Installer.

This chapter includes the following sections:

- [Quick Install](#)
- [Supported Platforms](#)
- [Browser Support](#)
- [Product Downloads](#)
- [Prerequisites](#)
- [Package Accessibility](#)
- [Installation Instructions](#)
- [Uninstalling Pivotal Command Center](#)
- [Uninstalling Pivotal Command Center](#)
- [Upgrading Pivotal Command Center](#)

Quick Install

Below is an Installation Quick Start, intended for users who are familiar with the product and various prerequisites; more detailed instructions are provided later in this chapter.

Task	Sub-Task
Prerequisites See Prerequisites for more details.	Check JDK version (as root) <pre># java -version</pre> Ensure you're running Oracle Java JDK Version 1.7. If not, download the appropriate version from Oracle.
	Check Yum accessibility (as root) Verify that all hosts have yum access to an EPEL yum repository. <pre># sudo yum list <LIST OF PACKAGES></pre>

Task	Sub-Task
Install Pivotal Command Center See Install Command Center for more details.	(as root) <ol style="list-style-type: none"> Copy tar file to your specified directory on the admin node, for example: <pre># scp ./PCC-2.1.x. version.build.os .x86_64.tar.gz host:/root/phd/</pre> Login as root and untar to that directory: <pre># cd /root/phd # tar --no-same-owner -zxvf PCC-2.1.x. version.build.os .x86_64.tar.gz</pre> Run the installation script from the directory where it is installed: <pre># ./install</pre> As the rest of the installation is done as the gpadmin user, change gpadmin user: <pre># su - gpadmin</pre> Enable Secure Connections
Import the PHD Services packages to the Admin Node See Install PHD Services for more details.	(as gpadmin) <ol style="list-style-type: none"> Copy the Pivotal HD services (PHD, ADS (HAWQ), and PHDTools (optional for USS), tarballs from the initial download location to the gpadmin home directory. Change the owner of the packages to gpadmin and untar the tarballs. <p>For example: If the file is a tar.gz or tgz, use:</p> <pre>tar xzf packagename.tgz</pre> <p>If the file is a tar, use:</p> <pre>tar xf packagename.tar</pre>
Enable the PHD Services See Enable the PHD Services for more details.	(as gpadmin) For each service (PHD, ADS, PHDTools) you are enabling, run the following: <pre># icm_client import -s < PATH TO EXTRACTED PHD SERVICE TAR BALL ></pre>
Launch Pivotal Command Center UI. See Launching Pivotal Command Center for more details.	Launch a browser and enter the host on which you installed PCC: <pre>https://CommandCenterHost:5443</pre> <p>The Command Center login page is launched in your browser. The default username/password is gpadmin/gpadmin.</p>
Configure and Deploy a Cluster See Configuring and Deploying a Cluster for more details.	After logging into PCC, click Add Cluster from the Cluster Status page. Navigate through the Add Cluster Wizard to configure and deploy a cluster.
Post Installation for HAWQ See Post Installation for HAWQ for more details.	(as gpadmin) Exchange keys between HAWQ master and segment hosts: Create a hostfile (HAWQ_Segment_Hosts.txt) that contains the hostnames of all your HAWQ segments. <pre># ssh < HAWQ_MASTER > # source /usr/local/hawq/greenplum_path.sh # /usr/local/hawq/bin/gpssh-exkeys -f ./HAWQ_Segment_Hosts.txt</pre>

Task	Sub-Task
Start the Cluster Starting the Cluster for more details.	Return to the PCC UI and start the cluster from the Cluster Status page.
Initialize HAWQ Initializing HAWQ for more details.	Initialize HAWQ (as gpadmin) ssh to the HAWQ master, then run the following: <pre># source /usr/local/hawq/greenplum_path.sh # /etc/init.d/hawq init</pre>

Supported Platforms

- RHEL 6.1 64-bit, 6.2 64-bit
- CentOS 6.1 64-bit, 6.2 64-bit

Browser Support

The following browsers are supported by Pivotal Command Center 2.1:

- Firefox 16, 19
- IE 8, IE 9, both with Google Chrome Frame
- Chrome 25.0.1364.172

Product Downloads

The following packages are required:

- `PCC-2.1.x.*.version_build_OS.x86_64.tar.gz`

Prerequisites

- Installation of Pivotal Command Center assumes the user has a working knowledge of the following:
 - **Yum:** Enables you to install or update software from the command line. See <http://yum.baseurl.org/>.
 - **RPM (Redhat Package Manager).** See information on RPM at [Managing RPM-Based Systems with Kickstart and Yum](#).
 - **NTP.** See information on NTP at: <http://www.ntp.org>
 - **SSH (Secure Shell Protocol).** See information on SSH at http://www.linuxproblem.org/art_9.html
 - **DNS lookup.** Verify that the admin host is able to reach every cluster node using its hostname and IP address. Verify that every cluster node is able to reach every other cluster node using its hostname and IP address:

```
# ping -c 2 myhost.mycompany.com
```

The return code should be 0

```
# ping -c 2 192.168.1.2
```

The return code should be 0

- **JAVA JDK.** Ensure that you are running Oracle JAVA JDK version 1.7:

As root:

```
# java -version
```

If you are not running the correct JDK, you can download a supported version from the Oracle site at

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Note: Oracle does not seem to be shipping self extracting JDK packages as of release 7. The current release of Pivotal Command Center expects a self extracting (.bin) package. So please restrict the Java version to one that has a .bin package

Note: Once you have installed Command Center, you can use the following command to install a downloaded JDK:

As gpadmin, run:

```
# icm_client import -f <PATH TO JDK>
```

- **YUM.** Verify that all hosts have yum access to an EPEL yum repository. See “[Package Accessibility](#)” for more details.
- **Internet Connectivity.** You will need an active internet connection to install PCC. This is required to pull the software dependencies for the UI.

Package Accessibility

Pivotal Command Center and Pivotal HD Enterprise expect some prerequisite packages to be pre-installed on each host, depending on the software that gets deployed on a particular host. In order to have a smoother installation it is recommended that each host would have yum access to an EPEL yum repository. If you have access to the Internet, then you can configure your hosts to have access to the external EPEL repositories. However, if your hosts do not have Internet access (or you are deploying onto a large cluster), then having a local yum EPEL repo is highly recommended. This will also give you some control on the package versions you want deployed on your cluster. See [Appendix A, “Creating a YUM EPEL Repository”](#) for instructions on how to setup a local yum repository or point your hosts to an EPEL repository.

For Pivotal Command Center 2.1, here is a list of pre-requisites that need to either already be installed on the Command Center admin host or on an accessible yum repository:

- httpd
- mod_ssl
- postgresql
- postgresql-devel
- postgresql-server
- compat-readline5

- createrepo
- sigar
- sudo

Run the following command on the admin node to make sure that you are able to install the prerequisite packages during installation.

```
$ sudo yum list <LIST OF PACKAGES>
```

For example:

```
$ sudo yum list httpd mod_ssl postgresql postgresql-devel
postgresql-server compat-readline5 createrepo sigar sudo
```

If any of them are not available or not already installed, then you may have not added the repository correctly to your admin host.

For the cluster hosts (where you plan to install the cluster), the prerequisite packages depend on the software you will eventually install there, but you may want to verify that the following two packages are installed or accessible by yum on all hosts:

- nc
- postgresql-devel

Installation Instructions

Once you have met the prerequisites, you are ready to begin the installation. Perform the following installation steps as a root user.

Note: Avoid using hostnames that contain capital letters because Puppet has an issue generating certificates for domains with capital letters. Also avoid using underscores as they are invalid characters in hostnames.

If you are upgrading from an earlier version of Pivotal Command Center, see [“Upgrading Pivotal Command Center”](#) on page 15.

Install Command Center

1. Copy the Command Center tar file to your host. For example:

```
# scp ./PCC-2.1.x.version.build.os.x86_64.tar.gz
host:/root/phd/
```

2. Log into the Command Center admin host as root user. `cd` to the directory where the Command Center tar files are located and `untar`. For example:

```
# cd /root/phd
# tar --no-same-owner -zxvf
PCC-2.1.x.version.build.os.x86_64.tar.gz
```

3. Still as root user, run the installation script. This installs the required packages and configures both Pivotal Command Center and starts services.

Important: You must run the installation script from the directory where it is installed, for example: `PCC-2.1.x.version`

For example:

```
# ls
PCC-2.1.x.version
PCC-2.1.x.version.build.os.x86_64.tar.gz
# cd PCC-version
# ./install
```

You will see installation progress information on the screen. Once the installation successfully completes, you will receive an installation success message on your screen.

Once you have configured and deployed a cluster, you can view your cluster status here: `https://<CommandCenterHost>:5443/status`

4. Enable Secure Connections:

Pivotal Command Center uses HTTPS to secure data transmission between the client browser and the server. By default, the installation script generates a self-signed certificate. Alternatively you can provide your own Certificate and Key by following these steps:

- a. Set the ownership of the certificate file and key file to `gpadmin`.
- b. Change the permission to owner read-only (mode 400)
- c. Edit the PCC configuration file
`/usr/local/greenplum-cc/config/commander` as follows:
 Change the path referenced in the variable `PCC_SSL_KEY_FILE` to point to your own key file.
 Change the path referenced in the variable `PCC_SSL_CERT_FILE` to point to your own certificate file.
- d. Restart PCC with the following command:
`service commander restart`

5. Verify that your PCC instance is running by executing the following command:

```
$ service commander status
```

6. From now on you can switch to the `gpadmin` user. You should no longer need to be root for anything else.

```
su - gpadmin
```

Next Steps: [Install PHD Services](#) then [Enable the PHD Services](#).

Starting, Stopping, and Restarting Command Center Services

To stop or restart Command Center services, run the following commands on the Pivotal Command Center admin host:

```
$ service commander stop
$ service commander start
$ service commander restart
```

Install PHD Services

Once you have Pivotal Command Center installed, you need to import and enable the PHD services (PHD, PHDTools, and HAWQ). You can use the import utility to sync the RPMs from the specified source location into the Pivotal Command Center (PCC) local yum repository of the Admin Node. This allows the cluster nodes to access the RPMs.

1. Copy the Pivotal HD, ADS, and PHDTools tarballs from the initial download location to the `gadmin` home directory.
2. Change the owner of the packages to `gadmin` and untar the tarballs. For example:

For PHD, if the file is a `tar.gz` or `tgz`, use:

```
tar zxf PHD-1.1.x-x.tgz
```

If the file is a `tar`, use:

```
tar xf PHD-1.1.x-x.tar
```

For Pivotal ADS, if the file is a `tar.gz` or `tgz`, use

```
tar zxf PADS-1.1.x-x.tgz
```

If the file is a `tar`, use:

```
tar xf PADS-1.1.x-x.tar
```

For PHDTools, if the file is a `tar.gz` or `tgz`, use

```
tar zxf PHDTools-1.1.x-x.tgz
```

If the file is a `tar`, use:

```
tar xf PHDTools-1.1.x-x.tar
```

Enable the PHD Services

1. As `gadmin`, extract the following tarball for Pivotal HD:

```
# icm_client import -s <PATH TO EXTRACTED PHD TAR BALL>
```

For example:

```
# icm_client import -s PHD-1.1.x-x/
```
2. Optional for HAWQ: As `gadmin`, extract the following tar ball for HAWQ and PXF:

```
# icm_client import -s <PATH TO EXTRACTED ADS TAR BALL>
```

Example:

```
# icm_client import -s PADS-1.1.x-x/
```

For more information, see the log file located at:
`/var/log/gphd/gphdmgr/gphdmgr-import.log`
3. Optional for USS: As `gadmin`, extract the following tar ball for USS:

```
# icm_client import -s <PATH TO EXTRACTED PHDTools TAR BALL>
```

Example:

```
# icm_client import -s PHDTools-1.1.x-x/
```

For more information, see the log file located at:

```
/var/log/gphd/gphdmgr/gphdmgr-import.log
```

You are now ready to configure and deploy a cluster from the Pivotal Command Center UI.

Launching Pivotal Command Center

Launch a browser and navigate to the host on which you installed Command Center. For example:

```
https://CommandCenterHost:5443
```

The Command Center login page is launched in your browser. The default username/password is `gpadmin/gpadmin`.

See [Chapter 2, “Using Pivotal Command Center UI”](#) for more details about the application, including how to change the default password.

Next Steps

See [Chapter 2, “Using Pivotal Command Center UI”](#) for details about using the application, including how to change the default password and how to deploy and configure a HD cluster via the Command Center UI.

See the *Pivotal HD 1.1 Enterprise Installation and Administrator Guide* for instructions for using the command-line interface of Pivotal Command Center to deploy and configure a HD cluster.

Uninstalling Pivotal Command Center

Follow the steps below to uninstall Pivotal Command Center and the Pivotal HD cluster:

1. Stop services on all your clusters (See the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide* for detailed steps).
2. Uninstall all your clusters (See the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide* for detailed steps).
3. From the directory where you untarred the Pivotal Command Center, run the uninstall script:

```
# cd /root/phd/PCC-2.1.x.version/
# ./uninstall
```

Upgrading Pivotal Command Center

The following instructions are for updating Pivotal Command Center from version 2.0.1 to 2.1.

Upgrade Notes:

- If you are upgrading to a new version of Pivotal Command Center, make sure you are also upgrading to compatible versions of Pivotal HD and Pivotal ADS (optional).
- See the latest version of the Pivotal Command Center Release notes for Pivotal Interoperability Matrix.
- We recommend that you always back up any critical data before performing any upgrades.

Follow the steps below to upgrade your Pivotal CC to a newer version:

1. [Optional] As `gpadmin`, stop the cluster:

```
# icm_client -l CLUSTERNAME stop
```

2. As `root`, stop the Command Center services:

```
# service commander stop
```

3. Download the new PCC tarball and untar.

4. Run the installer from the new PCC location:

```
# ./install
```

5. Enable Secure Connections.

Pivotal Command Center uses HTTPS to secure data transmission between the client browser and the server. By default, the installation script generates a self-signed certificate. Alternatively you can provide your own Certificate and Key; to do this, follow the instructions provided in step 4., “[Enable Secure Connections](#),” on page 12 of the installation instructions.

- Check the status of `nmon`:

```
# service nmon status
```

If the status anything other than `running`, stop and restart `nmon` as follows:

```
# service nmon stop
```

```
# service nmon start
```

3. Using Pivotal Command Center UI

This section provides an overview of the Pivotal Command Center 2.1 user interface.

Overview

Pivotal Command Center UI is a browser-based application for viewing the status and performance of Pivotal HD clusters. At a high level, the screens consist of:

- [Cluster Status Page](#)—Provides status information about any clusters you have configured and deployed. Also provides access to the Add Cluster Wizard that allows you to configure and deploy clusters from the UI. See [Configuring and Deploying a Cluster](#) for more details.
- [Dashboard](#)—Provides an overview of your Pivotal HD cluster. This screen shows at one glance the most important states and metrics that an administrator needs to know about the Pivotal HD cluster.
- [Cluster Analysis](#)—Provides detailed information about various metrics of your Pivotal HD cluster. This provides cluster-wide metrics all the way down to host-level metrics.
- [MapReduce Job Monitor](#)—Provides details about all, or a filtered set of MapReduce jobs.
- [YARN App Monitor](#)—Provides details about all, or a filtered set of YARN applications.
- [HAWQ Query Monitor](#)—When HAWQ (a revolutionary MPP database on Hadoop solution) is deployed on the cluster, Command Center can show the progress of all actively running queries on HAWQ.

Status indicators

Note that throughout the user interface the following indicators are used to indicate the status of nodes:

- Green: Succeeded
- Blue: Running
- Grey: Stopped/Pending
- Red: Killed/failed

Logging In

Launch a browser and navigate to the host on which you installed Command Center. For example:

```
https://CommandCenterHost:5443
```

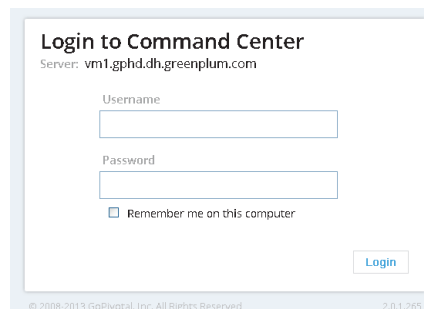
The Command Center login page is launched in your browser. The default username/password is `gpadmin/gpadmin`.

To change the default port (5443), update the `port` settings in the following file:

```
/usr/local/greenplum-cc/config/app.yml
```

Login Screen

The first time you launch the Command Center UI, a login screen appears showing the hostname of the host for the Command Center.



The default admin user/password is `gpadmin/gpadmin`. You can change this password via the [Settings Menu](#).


Click the `Login` button to launch the Command Center UI.

Cluster Status Page

Once you have launched Command Center, the Cluster Status screen appears, displaying a list of available clusters to monitor, the status of each cluster (**started**, **stopped**), and a list of services running on that cluster (Hive, Mahout, and so on).

From this page you can:

- Click **Add Cluster** to launch the Add Cluster Wizard.
- Click the cluster name in the table to view the Dashboard for that cluster.
- From any point within Command Center UI, you can always select a different cluster by using the **Select Cluster** drop-down menu in the upper right corner of the screen.
- Either **Start** or **Stop** a cluster.

Cluster Status			
Add Cluster			
MY CLUSTERS			
Cluster Name	Status	Services	Actions
 PivTest (v2.0)	Started	gfd, GPXF, HAWQ, HBase, HDFS, Hive, Mahout, Pig, YARN, ZooKeeper	Stop

Settings Menu

Click the **gear** icon in the upper right corner of the screen at any time to display the **Settings** menu. From the settings menu you can:

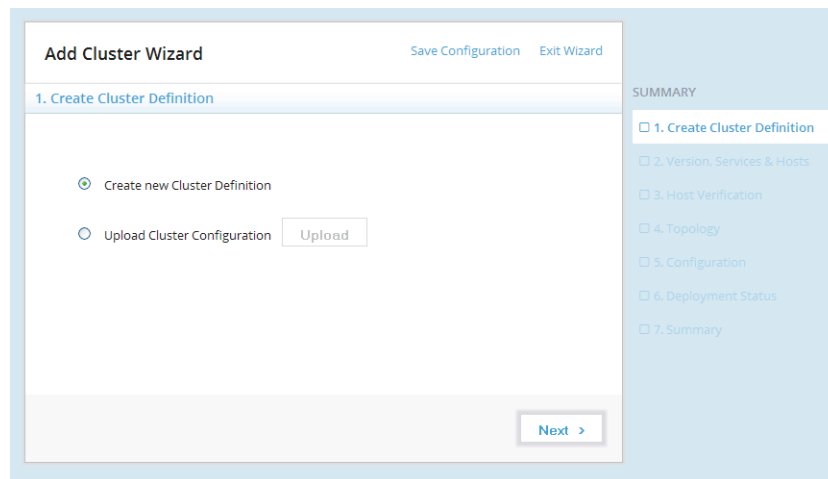
- **Cluster Status.** Click this to go back to a list of available clusters.
- **Change Password.** Click this to change your password.
- **Logout.**

Configuring and Deploying a Cluster

Note: Before you can configure and deploy a cluster, make sure you have already installed and enabled the PHD Services (see [Install PHD Services](#) and [Enable the PHD Services](#)).

After you have logged in to Pivotal Command Center, the Cluster Status page appears. From here, you are able to launch the **Add Cluster Wizard** that enables you to configure and deploy a Pivotal HD Cluster, as follows:

1. Click **Add Cluster**. The Add Cluster Wizard opens:



The Wizard allows you to create a new configuration from scratch or upload and edit any existing configuration. The Summary panel along the right shows you the progress of your configuration and deployment.

2. Create Cluster Definition. Either:

- a. If you are configuring a new cluster, select **Create a new Cluster Definition** then click **Next**.
 - b. If you want to edit an existing cluster; select **Upload Cluster Configuration**, click **Upload**, then navigate to the `clusterConfig.xml` file that you wish to edit; then click **Next**. In this case, the following fields in the Wizard will be populated with the cluster definition properties of that `clusterConfig.xml` file you just uploaded. Follow the instructions below to edit those values.
3. Versions, Services and Hosts:

Enter the following information:

Notes:

If you are editing an existing configuration, some if not all of these fields will be pre-populated. Edit where appropriate.

You need to scroll down to view all the fields on this screen. The **Next** button will not be active until you have entered all the required fields.

- **Name:** Required. Enter a name for this cluster. Special characters are not supported.
- **Hosts:** Required. Enter a new line-separated list of FQND host names. You can also click **Upload** to use a text file containing a new line-separated list of host names.
- **Root Password:** Required. Enter the root password.
- **GP Admin Password:** Required. Enter the `gpadmin` user password. Command Center creates this user on all nodes.
- **JDK Path:** Enter the JDK filename (not the absolute path). For example:
`jdk-6u26-linux-x64-rpm.bin`.
Note: JDK 1.7 is a pre-requisite. If not already installed, you can install using `icm_client import -f`
- **Setup NTP:** Check this box if you want to set up NTP (Network Time Protocol).

- **Disable SELinux:** Check this box if you want to disable SELinux. Recommended.
- **Disable IPTables:** Check this box if you want to disable IPTables. Recommended.
- **Run ScanHosts:** Check this box if you want to run scanhosts. The scanhosts command verifies that prerequisites for the cluster node and provides a detailed report of any missing prerequisites. Running this command ensures that clusters are deployed smoothly.

Click **Next**.

4. Host Verification:

The Host Verification page opens. This step may take a few minutes, it verifies connections to the hosts you just set up. Once the **Eligibility** field changes from **Pending**, to **Eligible** for all hosts, you can click **Next**. You will see any error and informational messages displayed in the comments fields.

Click **Next**.

5. Topology:

This is the section where you specify the roles to be installed on the hosts. For example, you can specify where your hadoop namenode, data node and so on, should be installed. Note that all mandatory roles should have at least one host allocated.

Each service has its own section on this page; you can use the top menu options as shortcuts to those sections on the page, or simply scroll down to each section.

Notes:

You need to click **Enter** or **Tab** before each field is accepted. Once you enter the text and click **Enter** or **Tab**, the text will change appearance and appear enclosed in a box, as shown in the figure below. The entry on the left has been accepted, the entry on the right has not.



At any point during this stage you can click **Save Configuration** at the top right of the page. This saves the configuration file and downloads it. Once saved, a link to the configuration file appears at the bottom of the page. Click that link to open and view the `clusterConfig.xml` file. You cannot edit this xml file directly.

These are the roles that need to have installation nodes defined:

- **CLIENT:** ICM installs Pig, Hive, HBase, and Mahout libraries on this host.
- **HDFS:** Name Node, Secondary Name Node, Data Nodes
- **YARN:** Resource Manager, History Server, Node Managers
- **Zookeeper:** Zookeeper Server
- **HBase:** Hbase Master, HBase Region Servers.
- **Hive:** Hive Master, Hive Metastore
- **HAWQ:** Primary Node, Secondary Node, HAWQ Segment Nodes

- **USS:** Name Node and Catalog
- **PXF:** No hosts to configure. Installed on the client host.
- **Mahout:** No hosts to configure. Installed on the client host.
- **Pig:** No hosts to configure. Installed on the client host.

Click **Next** once you have finished role-mapping.

6. Cluster Configuration:

This page displays a list of all configuration files that define this cluster; the `clusterConfig.xml` (to edit service configuration global values) as well as the service specific configuration files.

All these configuration files are already populated with the values you have already entered; or with default values.

Click any file name to open that configuration file in an editor and enter/edit values.

If you make any changes, click **Save** to return to the Cluster Configuration page.

Once you have completed all your edits, click **Deploy**.

7. Deployment Status:

PIVOTAL COMMAND CENTER

gpadmin | Select Cluster +

Add Cluster Wizard Save Configuration Exit Wizard

6. Deployment Status 100% COMPLETE

Hostname	Status	Role	Messages
centos62-2	INSTALLED	Hive Client	[INFO] Puppet Sync Finished
centos62-2	INSTALLED		[INFO] Puppet Sync Finished
centos62-2	INSTALLED	Hadoop Client	[INFO] Puppet Sync Finished
centos62-2	INSTALLED	ZooKeeper Server	[INFO] Puppet Sync Finished
centos62-2	INSTALLED	Mahout Client	[INFO] Puppet Sync Finished
centos62-2	INSTALLED	HBase Client	[INFO] Puppet Sync Finished
centos62-4	INSTALLED	HAWQ Standby Master	[INFO] Puppet Sync Finished
centos62-3	INSTALLED	Name Node	[INFO] Puppet Sync Finished
centos62-3	INSTALLED		[INFO] Puppet Sync Finished
centos62-3	INSTALLED	HAWQ Master	[INFO] Puppet Sync Finished
centos62-3	INSTALLED	YARN Resource Manager	[INFO] Puppet Sync Finished
centos62-3	INSTALLED	Hive Server	[INFO] Puppet Sync Finished
centos62-3	INSTALLED	HBase Master Server	[INFO] Puppet Sync Finished
centos62-4	INSTALLED		[INFO] Puppet Sync Finished
centos62-4	INSTALLED	Data Node	[INFO] Puppet Sync Finished

Next >

SUMMARY

- ☒ 1. Create Cluster Definition
- ☒ 2. Version, Services & Hosts
- ☒ 3. Host Verification
- ☒ 4. Topology
- ☒ 5. Configuration
- ☒ 6. Deployment Status
- ☐ 7. Summary

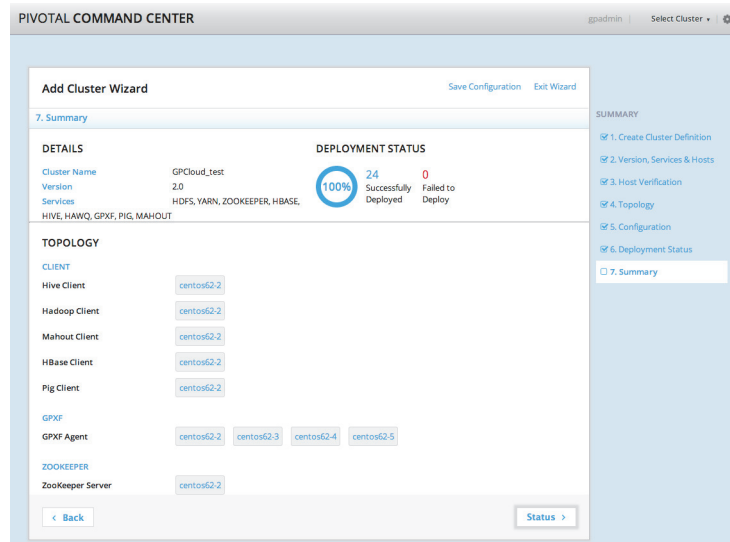
This screen shows the progression of the deployment. Information displayed includes:

- **Hostname**
- **Status**
- **Role**
- **Messages**

Once the deployment is complete, click **Next**.

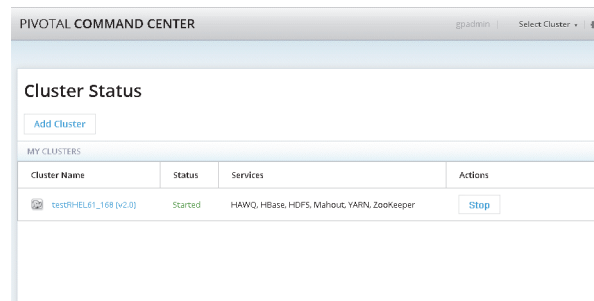
8. Summary.

Once your cluster has successfully deployed, you can view a summary of the cluster, as shown here:



9. Return to the Cluster Status page:

Once you have reviewed this summary, click **Status**, to return to the Cluster Status page. Your new cluster will be listed on this page, with the status of **installed**



If you are deploying HAWQ there is one more manual step you need to take before you can start the cluster, see [Post Installation for HAWQ](#).

Post Installation for HAWQ

You need to exchange SSH keys between HAWQ Master and Segment Nodes to complete HAWQ installation.

1. Create a hostfile (HAWQ_Segment_Hosts.txt) that contains the hostnames of all your HAWQ segments.
2. As gpadmin, execute the following commands from the HAWQ Master.


```
# ssh <HAWQ_MASTER>
# source /usr/local/hawq/greenplum_path.sh
# /usr/local/hawq/bin/gpssh-exkeys -f
# ./HAWQ_Segment_Hosts.txt
```

Next steps: [Starting the Cluster](#) then [Initializing HAWQ](#).

Starting the Cluster

To start your cluster; click **Actions: Start** on the Cluster Status page.

Initializing HAWQ

As gadmin ssh to the HAWQ master, the run the following:

```
# source /usr/local/hawq/greenplum_path.sh
# /etc/init.d/hawq init
```

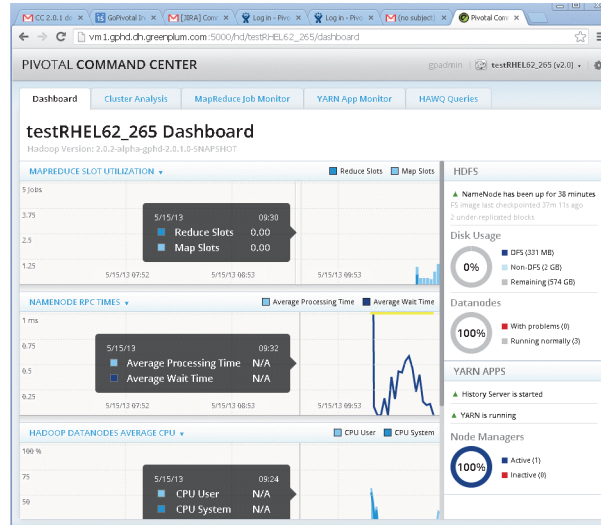
You have now completed your cluster configuration and deployment.

See the following sections of this document for details about using the PCC UI to administer and monitor your cluster.

Dashboard

The dashboard gives you a high level view of a cluster at a glance. You are able to view the status of the most important cluster services, such as HDFS and YARN, and allows you to start and stop each service individually. It also shows you how the most important cluster metrics are trending in a visual way.

The graphs provide a unified view of the state of your system. They are also useful in detecting outliers and pinpointing specific problems that may be present in your system.



The right side of the Dashboard displays the state of the following services, provided they have been deployed for this cluster:

- **HDFS**

For HDFS, the dashboard provides the following information:

- The status of HDFS. You can use the **Actions** dropdown menu to **Start/Stop HDFS** depending on its status.
- When the last NameNode checkpoint occurred.
- The percentage of cluster storage being used by HDFS and how much is free.
- The number of DataNodes that are up and whether they are running normally or with problems

- **YARN**

For YARN, the dashboard provides the following information:

- The status of YARN. You can use the **Actions** dropdown menu to **Start/Stop YARN** depending on its status.
- Whether or not the YARN History Server is running.

Note: The History Server stores a history of the mapreduce jobs run on the cluster.

- The number of NodeManagers that are running.
- **HBase**
For HBase, the dashboard provides the following information:
 - The status of the HBase master. You can use the **Actions** dropdown menu to **Start/Stop HBase** depending on its status.
- **Zookeeper**
For Zookeeper, the dashboard provides the following information:
 - The status of Zookeeper. You can use the **Actions** dropdown menu to **Start/Stop Zookeeper** depending on its status.
 - The Status of the Zookeeper server host.
- **Hive**
For Hive, the dashboard provides the following information:
 - The status of Hive. You can use the **Actions** dropdown menu to **Start/Stop Hive** depending on its status

The Dashboard also provides metrics about:

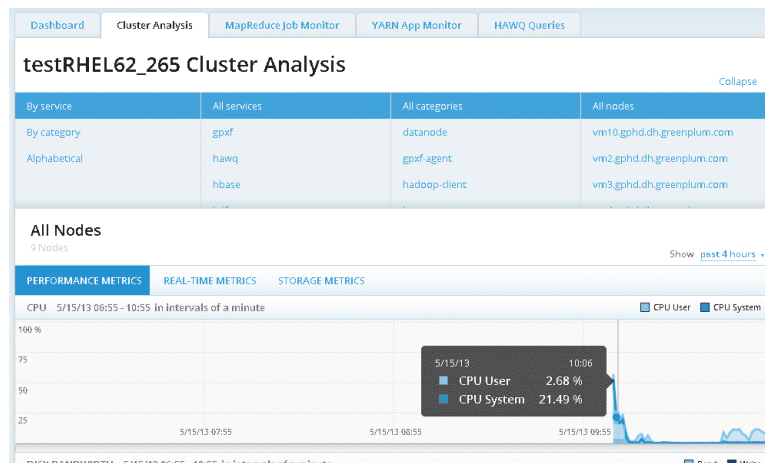
- **Mapreduce Slot Utilization**
- **Namenode RPC Times**
- **Hadoop Datanodes Average CPU**
- **Hadoop Datanodes Average Bandwidth**
- **Namenode Operations Per Second**
- **Hadoop Datanodes Average Disk Bandwidth**
- **Hadoop Datanodes Average Memory**
- **Mapreduce Jobs By Status**

Cluster Analysis

The Cluster Analysis screen provides detailed metrics on your Pivotal HD cluster.

It provides cluster-wide metrics all the way down to host-level metrics. It provides Hadoop-specific metrics, as well as system metrics that you can drill down to if needed.

The Cluster Analysis screen displays the same data that is shown in the dashboard but in greater detail.



By default the Cluster Analysis screen displays the metrics for all services, all categories, and all nodes. You can filter the information displayed by combinations of the following filters:

- **By Service**
Metrics can be filtered by services such as HDFS, YARN, or HAWQ.
- **By Category**
Metrics can be filtered by categories such as:
 - namenode
 - secondarynamenode
 - datanode
 - yarn-resourcemanager
 - yarn-nodemanager
 - mapreduce-historyserver
 - hawq-master
 - hawq-segment
- **Alphabetically**
Metrics can be filtered alphabetically.

Based on the filters you select, the lower part of the Cluster Analysis screen provides detailed graphs that display data related to:

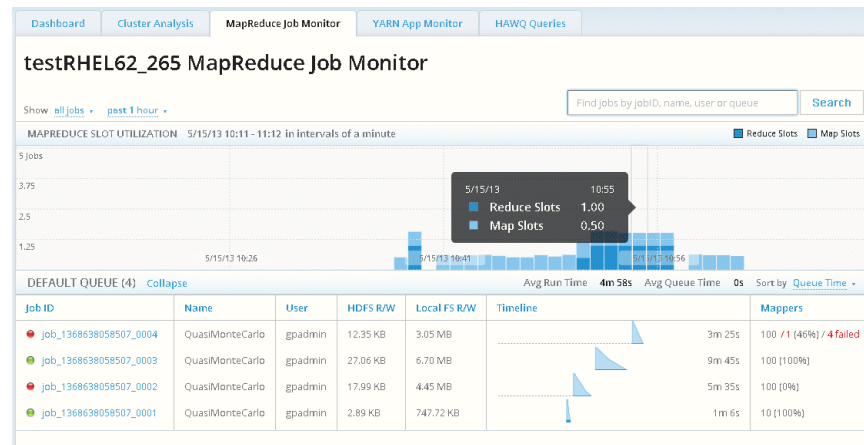
- **CPU**
- **Disk Bandwidth**
- **Network Bandwidth**
- **Memory**
- **Load**
- **Swap Usage**
- **Swap I/O**
- **Network Operations**
- **Disk Operations**

You can view either the **Performance Metrics**, which show the cluster/node utilization over-time, the **Real-time Metrics** which show the current metrics in real-time, or **Storage Metrics**, which show metrics about cluster storage.

If you select Cluster Analysis for **All Nodes** (the default), the Trending Metrics graph for the cluster is displayed:

MapReduce Job Monitor

The Job Monitor screen tracks the MapReduce jobs that are executed in the Pivotal HD cluster when the YARN MapReduce service is running. It provides details about all, or a filtered set of MapReduce jobs.



The MapReduce jobs displayed can be filtered by state and/or time range.

- **By state:**
 - all jobs (set by default)
 - currently pending jobs
 - currently running jobs
 - succeeded jobs
 - failed jobs
 - killed jobs
- **By time range:**
By selecting a preset time range in hours, weeks, months, year, or by specifying a custom time range.

The MapReduce jobs can also be filtered by searching for values for the following:

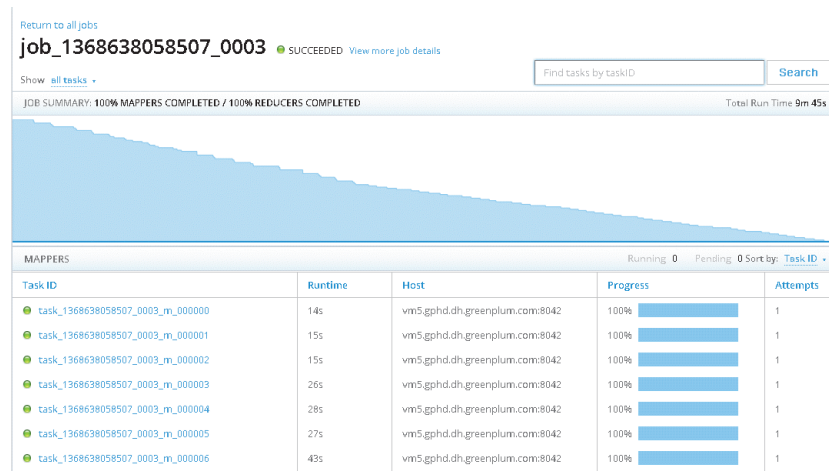
- **jobID**
- **name**
- **user**
- **queue**

Enter your search value in the search bar in the following format: searchKey=searchValue, where searchKey is one of **jobID**, **name**, **user**, or **queue**.

These are substring searches. For example: **jobID=1363920466130** will locate a job with **jobID=job_1363920466130_0002**

Job Details

When you click on any of the jobs in the Job Monitor more details of the job are shown.



This screen displays all the tasks that have been allocated for the selected job and their progress. You can see the mapper and the reducer tasks separately. In the above screen capture, the bars in the JOB SUMMARY section represent the two Mapper tasks that have run, one took 19 seconds, the other, 20 seconds.

Clicking on each task ID will show even more details about that particular task. You can also filter on a particular task ID in the search bar.

task_1368638058507_0003_m_000000 ×

Status ● Succeeded 100% Complete Runtime 5/15/13 10:47 - 10:48 14s elapsed

Attempt ID	Start Time	Runtime	Host	Progress
attempt_1368638058507_0003_m_000000_0	5/15/13 10:47	12s	vm5.gphd.dh.greenplum.com:8042	100%

attempt_1368638058507_0003_m_000000_0

ORG.APACHE.HADOOP.MAPREDUCE.FILESYSTEMCOUNTER	FILE_BYTES_READ:	120
	FILE_BYTES_WRITTEN:	69445
	FILE_LARGE_READ_OPS:	0
	FILE_READ_OPS:	0
	FILE_WRITE_OPS:	0
	HDFS_BYTES_READ:	274
	HDFS_BYTES_WRITTEN:	0
	HDFS_LARGE_READ_OPS:	0
	HDFS_READ_OPS:	4
	HDFS_WRITE_OPS:	0
ORG.APACHE.HADOOP.MAPREDUCE.TASKCOUNTER	MAP_OUTPUT_MATERIALIZED_BYTES:	28
	COMMITTED_HEAP_BYTES:	660602880
	COMBINE_INPUT_RECORDS:	0
	VIRTUAL_MEMORY_BYTES:	1948426240

To see job related counters click on **View more job details** next to the job ID:

job_1382053843582_0001				
Status	SUCCEEDED Analyze Job			User: rgeiger
Runtime	10/21/13 19:59 - 20:00 0s elapsed			Job name: QuasiMonteCarlo
		Counter	Map	Reduce
SHUFFLE ERRORS		BAD_ID:	0	0
		CONNECTION:	0	0
		IO_ERROR:	0	0
		WRONG_LENGTH:	0	0
		WRONG_MAP:	0	0
		WRONG_REDUCE:	0	0
ORG.APACHE.HADOOP.MAPREDUCE.FILESYSTEMCOUNTER		FILE_BYTES_READ:	0	226
		FILE_BYTES_WRITTEN:	691710	69419
		FILE_LARGE_READ_OPS:	0	0
		FILE_READ_OPS:	0	0
		FILE_WRITE_OPS:	0	0
		HDFS_BYTES_READ:	2590	0
		HDFS_BYTES_WRITTEN:	0	215
		HDFS_LARGE_READ_OPS:	0	0

Click the **Analyze Job** link adjacent to the Status field to open a Vaidya report about the selected job, as shown below:

Note: This capability is beta and will be improved in coming releases.


Vaidya report for job_1383346862816_0001
Logged in as: dr.who

- Application
- Job
 - Overview
 - Counters
 - Configuration
 - Map tasks
 - Reduce tasks
 - Vaidya report
- Tools

Job Name:	word count
User Name:	root
Queue:	default
State:	SUCCEEDED
Uberized:	false
Started:	Mon Nov 04 19:26:54 CST 2013
Finished:	Mon Nov 04 19:27:03 CST 2013
Elapsed:	9sec

Vaidya Report

Note: Following diagnostic results and remedies are suggestive in nature and may not be valid or applicable to some jobs under certain situations.

Diagnostic Test Rule	Test Result	Severity	Reference Details	Suggested Remedy
Intermediate Data Compression	FAILED	0.66	<ul style="list-style-type: none"> mapreduce.map.output.compress: false PercentCompressionThreshold: 20.0 Reduce Shuffle Bytes: 0 Map Output Bytes: 0 Average Output Bytes per Map: 0 Compression Percentage: -1.0 Impact: 1.0 	<ul style="list-style-type: none"> It is usually good idea to compress the intermediate map output to reduce the network data transfer from mappers to reducers. Although if compression is not enough, then enabling it may lower the job performance. Typically use either snappy or Lzo compression codec for intermediate data compression for better performance This test will fail if intermediate compression is not enabled for Map-Reduce job OR if enabled but percent compression is less than PercentCompressionThreshold specified.
Output Data Compression	FAILED	0.66	<ul style="list-style-type: none"> mapreduce.output.fileoutputformat.compress: false mapreduce.output.fileoutputformat.compress.codec: org.apache.hadoop.io.compress.DefaultCodec Impact: 1.0 	<ul style="list-style-type: none"> It is usually good to compress the job output to save the space on HDFS. Gzip2 compressed files can be split for processing using multiple maps and hence should be used if the output of one job is to be given as an input to another. Typically it is good to use snappy or LZO codec for intermediate data compression between map & reduce
Single Reducer	FAILED	0.66	<ul style="list-style-type: none"> Total reducers: 1 Impact: 1.0 	<ul style="list-style-type: none"> Make sure you did not forget to set the appropriate number of reduce tasks and thus end up in using default i.e single reduce

About Vaidya

Vaidya is a diagnostic tool installed with PHD for Map/Reduce jobs. After a job is executed successfully, it uses a job history log and job configuration information to identify any performance or scalability problems with the job. Upon execution, it provides a job analysis report indicating specific problems with the job along with the remedy to correct them.

For more information about Vaidya, see the *PHD Enterprise 1.1 Stack and Tool Reference Guide*.

YARN App Monitor

The YARN App Monitor screen tracks YARN applications that are executed in the Pivotal HD Cluster.

CentOs62Build783_34 YARN App Monitor

Show [all apps](#) • [past 1 hour](#) •

Find apps by appID, name or user [Search](#)

Application ID	Name	User	Timeline
application_1363133588036_0640	TeraGen	mapred	31s
application_1363133588036_0641	TeraSort	mapred	2m 9s
application_1363133588036_0642	TeraGen	mapred	48m 21s
application_1363133588036_0643	TeraValidate	mapred	1m 27s
application_1363133588036_0644	TeraGen	mapred	1m 48s
application_1363133588036_0645	TeraSort	mapred	3m 50s
application_1363133588036_0646	TeraSort	mapred	4m 21s
application_1363133588036_0647	TeraGen	mapred	4m 17s
application_1363133588036_0648	TeraGen	mapred	4m 50s
application_1363133588036_0649	TeraValidate	mapred	4m 40s
application_1363133588036_0650	TeraGen	mapred	41m 34s

The YARN applications displayed can be filtered by category and/or time range:

- **By Category:**
 - **all apps** (set by default)
 - **currently pending apps**
 - **currently running apps**
 - **succeeded apps**
 - **failed apps**
 - **killed apps**
- **By Time Range:**
By selecting a preset time range in hours, weeks, months, year, or by specifying a custom time range.

The YARN applications can also be filtered by the following fields by entering it in the search bar in the following format: `searchKey=searchValue`:

- **appID**
- **name**
- **user**

These are substring searches. For example: **appID=1363920466130** will locate the application with **appID=application_1363920466130_0002**

HAWQ Query Monitor

The HAWQ Query monitor is only displayed when HAWQ is installed on the cluster.

This screen displays all **active** queries running on the HAWQ cluster:

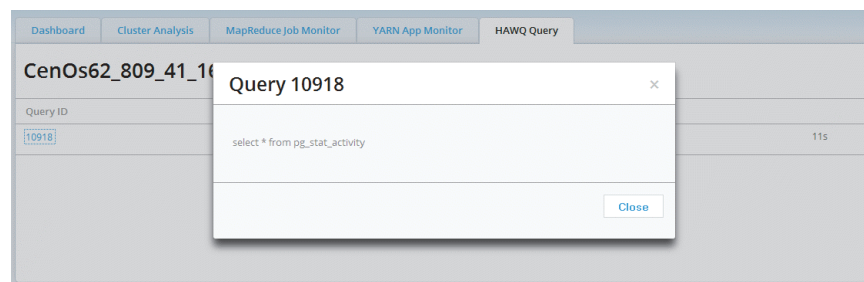
CenOs62_809_41_1616 HAWQ Queries			
Query ID	Database	User	Timeline
10881	postgres	gpadmin	10s

In this release, this screen only displays **active** queries as can be seen when you run:

```
SELECT * FROM pg_stat_activity;
```

on the HAWQ cluster.

Click on a Query ID to get the syntax of that query:



A. Creating a YUM EPEL Repository

Pivotal Command Center and Pivotal HD Enterprise expect some prerequisite packages to be pre-installed on each host, depending on the software that gets deployed on a particular host. In order to have a smoother installation it is recommended that each host would have yum access to an EPEL yum repository. If you have access to the Internet, then you can configure your hosts to have access to the external EPEL repositories. However, if your hosts do not have Internet access (or you are deploying onto a large cluster), then having a local yum EPEL repo would be highly recommended. This will also give you some control on the package versions you want deployed on your cluster.

Following are the steps to create a local yum repo:

1. Mount the RHEL/CentOS DVD on a machine that will act as the local yum repo
2. Install a webserver on that machine (e.g. httpd), making sure that HTTP traffic can reach this machine
3. Install the following packages on the machine:

```
yum-utils
createrep
```

4. Go to the directory where the DVD is mounted and run the following command:

```
# createrepo .
```

5. Create a repo file on each host with a descriptive filename in the `/etc/yum.repos.d/` directory of each host (for example, `CentOS-6.1.repo`) with the following contents:

```
[CentOS-6.1]
name=CentOS 6.1 local repo for OS RPMS
baseurl=http://172.254.51.221/centos/$releasever/os/
$basearch/
enabled=1
gpgcheck=1
gpgkey=http://172.254.51.221/centos/$releasever/os/$basearch
/RPM-GPG-KEY-CentOS-6
```

6. Validate that you can access the local yum repos by running the following command:

```
Yum list
```

B. Best Practises and Troubleshooting

nmon Issues

- If you have to restart the Admin node, ensure that the nmon service is started.
- If you notice any of the clusters are not being fully monitored, perform the following on the Admin node:
 - Make sure the nmon configuration (`/etc/nmon/conf/nmon-site.xml`) includes all the clusters and their hosts. If it doesn't, update it and distribute the updated configuration to all the cluster hosts, then restart nmon on the Admin node as well as on the cluster hosts:


```
sudo service nmon restart
```

```
massh clusterHosts verbose 'sudo service nmon restart'
```

 Where `clusterHosts` contains all the cluster hosts.

Job Monitor Page

- If an application is completed, but on the job monitoring page in the Command Center User Interface, it shows app/job as still running, then check whether History Server is running or not. If it is not running, start it.
Check using: `http://<HistoryServerHost>:19888`

HTTPS Issues

The following errors related to security Keys and Certificates may be issued:

- The `PCC_SSL_KEY_FILE` environment variable must be set.
See the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide* for details.
- The `PCC_SSL_CERT_FILE` environment variable must be set.
See the *Pivotal HD Enterprise 1.1 Installation and Administrator Guide* for details.
- Cannot find `PCC_SSL_KEY_FILE` `"$PCC_SSL_KEY_FILE"`.
Ensure the path is set correctly.
- Cannot find `PCC_SSL_CERT_FILE` `"$PCC_SSL_CERT_FILE"`.
Ensure the path is set correctly.
- Permissions for `$PCC_SSL_KEY_FILE` are too open.
We recommend that your private key files are NOT readable, writable or executable by others.
- Permissions for `$PCC_SSL_CERT_FILE` are too open.
We recommend that your private key files are NOT readable, writable or executable by others.