Pivotal™

Pivotal Command Center 2.0 User Guide

Rev: A01

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, EMC will provide a copy of the source code that is required to be made available in accordance with the applicable open source license. EMC may charge reasonable shipping and handling charges for such distribution. Please direct requests in writing to EMC Legal, 176 South St., Hopkinton, MA 01748, ATTN: Open Source Program Office.

Date: 9/26/13

Table of Contents

Preface

This preface includes four sections:

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, "Using Pivotal Command Center UI"— An overview of the functionality of the Pivotal Command Center console (user interface).
- Appendix 3, "Best Practises -Troubleshooting".

Document Conventions

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

Text Conventions

Table 0.1 Text Conventions

Text Convention	Usage	Examples
italics	New terms where they are defined Database objects, such as schema, table, or columns names	The *master instance* is the `postgres` process that accepts client connections. Catalog information for Pivotal Command Center resides in the *pg_catalog* schema.
`monospace`	File names and path names Programs and executables Command names and syntax Parameter names	Edit the `postgresql.conf` file. Use `gpstart` to start Pivotal Command Center.
`<monospace italics>`	Variable information within file paths and file names Variable information within command syntax	`/home/gpadmin/<`*`config_file`*`>` `COPY` *`tablename`* `FROM '<`*`filename`*`>'`
`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: `jdbc:postgresql://`**`host:5432/mydb`**
`UPPERCASE`	Environment variables SQL commands Keyboard keys	Make sure that the Java `/bin` directory is in your `$PATH`. `SELECT * FROM` *`my_table`*`;` Press `CTRL+C` 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.	`FROM` **`{`** `'`*`filename`*`'` `|` `STDIN` **`}`**
`[ ]`	Within command syntax, square brackets denote optional arguments. Do not type the brackets.	`TRUNCATE` **`[`** `TABLE` **`]`** *`name`*

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 name [, ...]`
`|`	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. Refer to the *Pivotal HD Enterprise 1.0 Installation and Administrator Guide* for installation and configuration information for PCC.

This chapter provides a brief overview of Pivotal Command Center.

Pivotal Command Center Overview

The Pivotal Command Center (PCC) allows an administrative user to administer and monitor one or more Pivotal HD clusters. The Command Center has command-line tools to deploy and configure Pivotal HD clusters, as well as an intuitive graphical user interface that is designed to help the user view the status of the clusters and take appropriate action. This release of Command Center allows administering and monitoring of only Pivotal HD Enterprise 1.0.x clusters.

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

- Cluster installation, configuration and uninstalls (currently via the CLI and documented in the *Pivotal HD Enterprise 1.0 Installation and Administrator Guide.)*
- Cluster monitoring and management.

These functions are served through a set of RESTful web services that run as a web application on an Apache-Tomcat 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 monitor and manage one or more Pivotal HD clusters. 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 start and stop the clusters. For how to use this CLI, please refer to the *Pivotal HD Enterprise 1.0 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 1.0 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
```

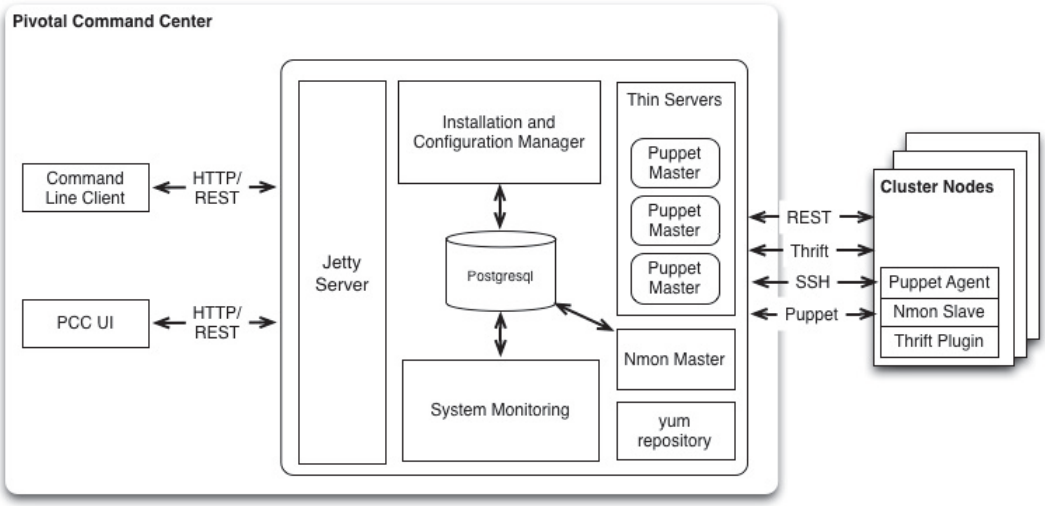
PostgreSQL Database

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 1.0.x, refer to the *Pivotal HD 1.0 Installation and Administrator Guide*.



2. Using Pivotal Command Center UI

This section provides an overview of the Pivotal Command Center 2.0 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:

- 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. This has hadoop-specific metrics such as MapReduce slot utilization and NameNode performance, as well as system metrics such as CPU, memory, disk and network statistics.
- 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
```

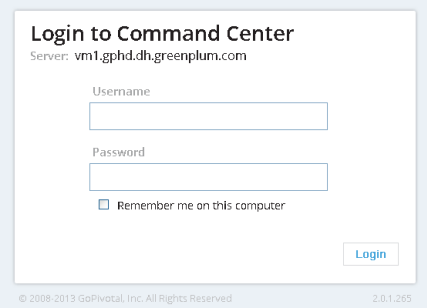
Browser Support

The following browsers are supported by Pivotal Command Center 2.0:

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

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.



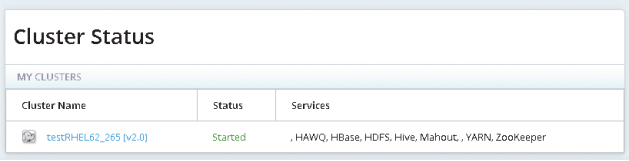
The default admin user/password is `gpadmin`/`gpadmin`. You can change this password via the Settings Menu.

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

Selecting a Cluster

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).

- Click the cluster name in the table to select a 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.



Settings Menu

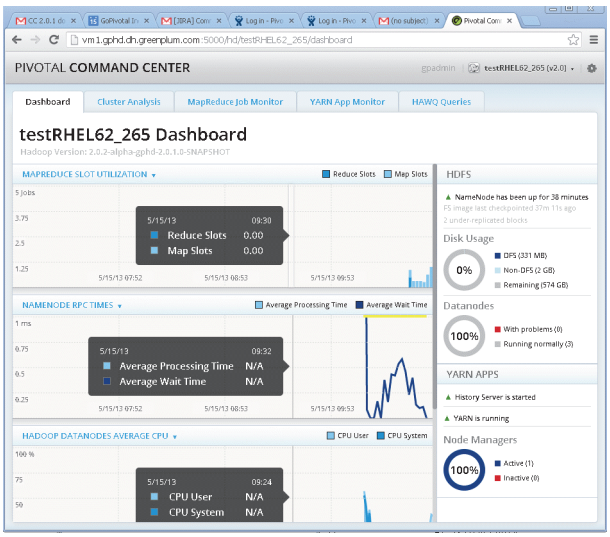
Click the **gear** icon in the upper right corner of the screen at any time to display the **Settings** menu. From the settings menu you can:

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

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. 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 both HDFS and YARN services. It answers the following questions:

- Is HDFS up?
- When did the last NameNode checkpoint occur?
- What percentage of cluster storage is used by HDFS and how much is free?
- How many DataNodes are up and are they running normally or with problems?
- Is YARN up?
- Is the History Serverup?
  Note: The History Server stores a history of the mapreduce jobs run on the cluster.
- How many NodeManagers are up?

The Dashboard provides metrics about:

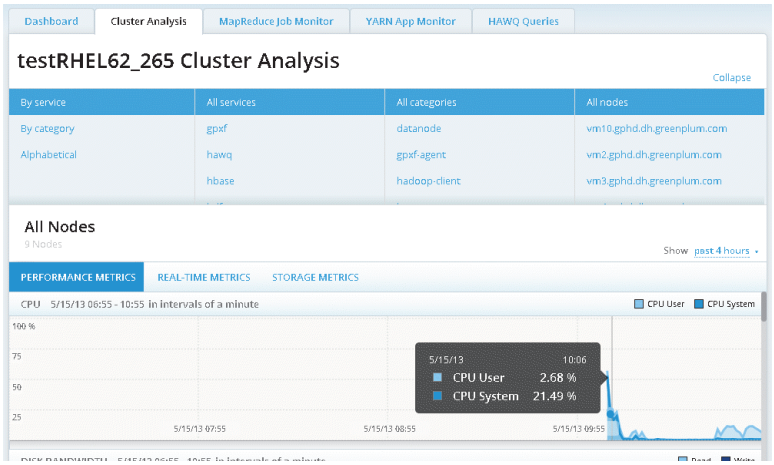
- **Mapreduce Slot Utilization**
- **Namenode RPC Times**
- **Hadoop Datanodes Average CPU**
- **Hadoop Datanodes Average Bandwidth**
- **Namenode Operations Per Second**
- **Hadoop Datanodes Average Disk Bandwidth**
- **Hadoop Datanodes Average Memory**
- **Mapreduce Jobs By Status**

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:

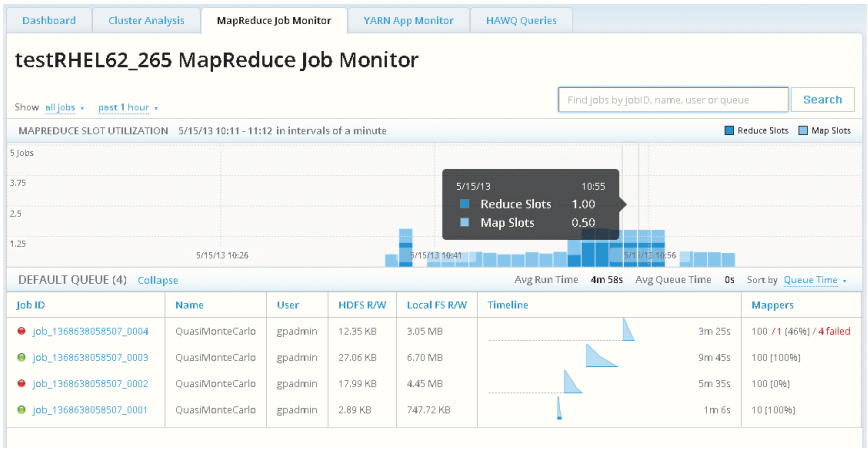
- **Mapreduce Slot Utilization**
- **Namenode RPC Times**
- **Avg Namenode File Operations Per Second**
- **Mapreduce Jobs by Status**
- **Segment CPU**
- **Disk Bandwidth**
- **Network Bandwidth**
- **Segment Memory**
- **Load**
- **Swap Usage**
- **Swap I/O**
- **Network Operations**
- **Disk Operations**

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:

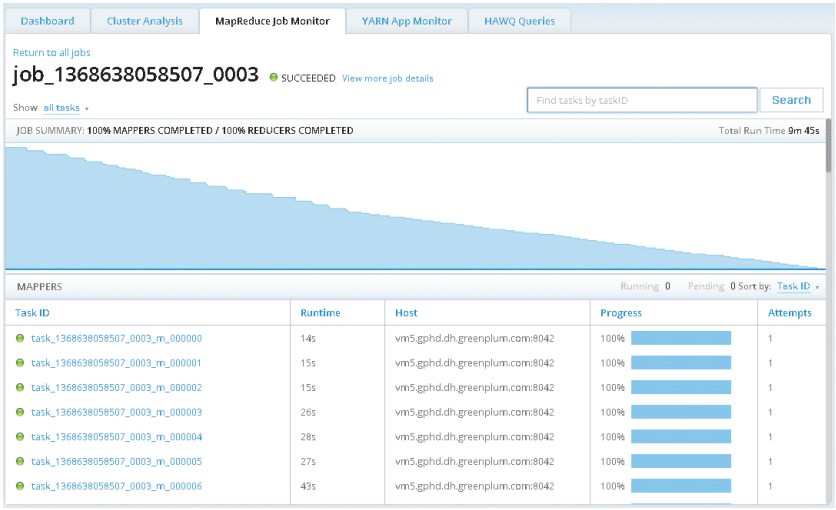
- **jobID**
- **name**
- **user**
- **queue**

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**

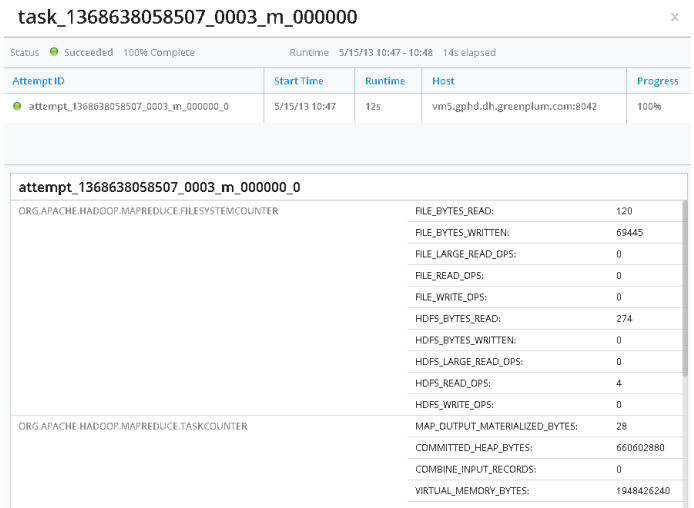
Job Details

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

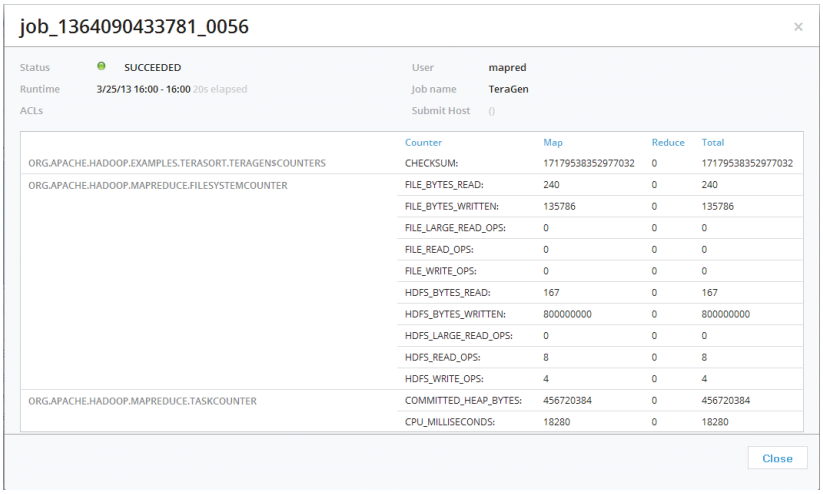


This screen displays all the tasks that are have been allocated for the selected job and their progress. You can see the mapper and the reducer tasks separately. In the above screen capture, the bars in the JOB SUMMARY section represent the two Mapper tasks that have run, one took 19 seconds, the other,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

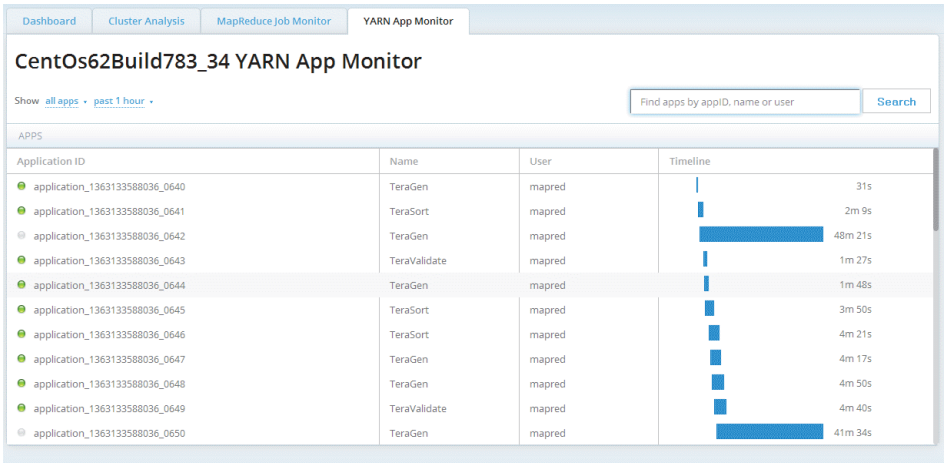


To see job related counters click on **View more job details** next to the job ID:



YARN App Monitor

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



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`:

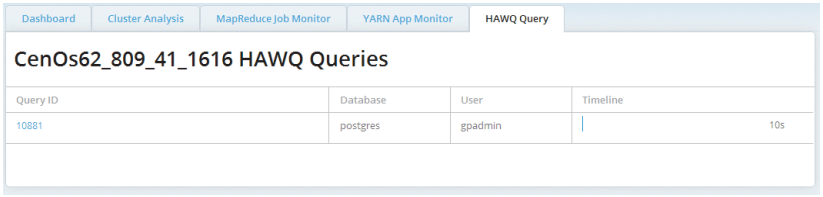
- **appID**
- **name**
- **user**

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.

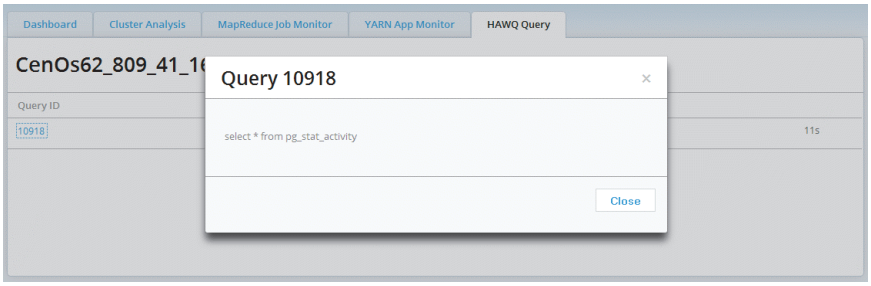
This screen displays all **active** queries running on the HAWQ cluster:



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:



3. Best Practises -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/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 1.0 Installation and Administrator Guide* for details.
- `The PCC_SSL_CERT_FILE environment variable must be set.`
- See the *Pivotal HD Enterprise 1.0 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.