Graphite/Prometheus and Grafana

Overview

CPS system and application statistics and Key Performance Indicators (KPI) are collected by the system and can be displayed using a browser-based graphical metrics tool. This chapter provides a high-level overview of the tools CPS uses to collect and display these statistics.

The list of statistics available in CPS is consolidated in an Excel spreadsheet. After CPS is installed, this spreadsheet can be found in the following location on the Cluster Manager VM:

```
/var/qps/install/current/scripts/documents/QPS_statistics.xlsx
```

Prometheus

Prometheus is an application which is a part of monitoring solution in CPS. It is used to actively gather statistics from the running virtual machines and application services.

Prometheus application resides on both pcrfclient VMs. It scrapes statistics from collectd exporter after every configured interval and stores in `/var/data/Prometheus` directory on pcrfclient VMs.

To learn more about Prometheus, refer to: https://prometheus.io/docs/introduction/overview/.

Enable Prometheus

This following sections provides information on how to enable Prometheus on CPS system.

- By default, Prometheus is disabled on system. You need to configure Prometheus to start its operation.
- You can configure Prometheus using CSV based configurations or API based configurations.
- By default, statistics granularity is set to 10 seconds. To change it, you need to configure statistics granularity. Support is present for both CSV/API based installations.

Note It is recommended to keep the default statistics granularity. If you want to change the value, contact your Cisco Technical representative.

- After enabling Prometheus, you must add Prometheus data source in Grafana.
- When Prometheus is enabled on the system, existing dashboards created with graphite will not work. You must use Prometheus queries to create new dashboard on the system.

CSV Based Installation Configuration Parameters

Table 1: CSV Based Installation Parameters

Parameter	Description
enable_prometheus	This parameter is used to enable/disable Prometheus in CPS. Default: disabled Possible Values: enabled, disabled
stats_granularity	This parameter is used to configure statistics granularity in seconds. Default: 10 seconds Possible Values: Positive Number

For example, in case of CSV based installations, you can configure **Configuration.csv** with the following parameters to enable Prometheus on Cluster Manager:

```
cat /var/qps/config/deploy/csv/Configuration.csv | tail -5
db_authentication_admin_passwd,72261348A44594381D2E84ADDD1E6D9A,
db_authentication_passwd_encryption,true,
db_authentication_readonly_passwd,72261348A44594381D2E84ADDD1E6D9A,
enable_prometheus,enabled,
stats_granularity,10,
```

After configuring the parameters, run the following commands to import the new configuration to VMs:

```
/var/qps/install/current/scripts/import/import_deploy.sh
/var/qps/install/current/scripts/upgrade/reinit.sh
```

API Based Installation Parameters

Table 2: API Based Installation Parameters

Parameter	Description
enablePrometheus	This parameter is used to enable/disable Prometheus in CPS. Default: disabled Possible Values: enabled, disabled
statsGranularity	This parameter is used to configure statistics granularity in seconds. Default: 10 seconds Possible Values: Positive Number

In case of API based installations, you need to use `api/system/config/config` PATCH API from Cluster Manager.

For example:

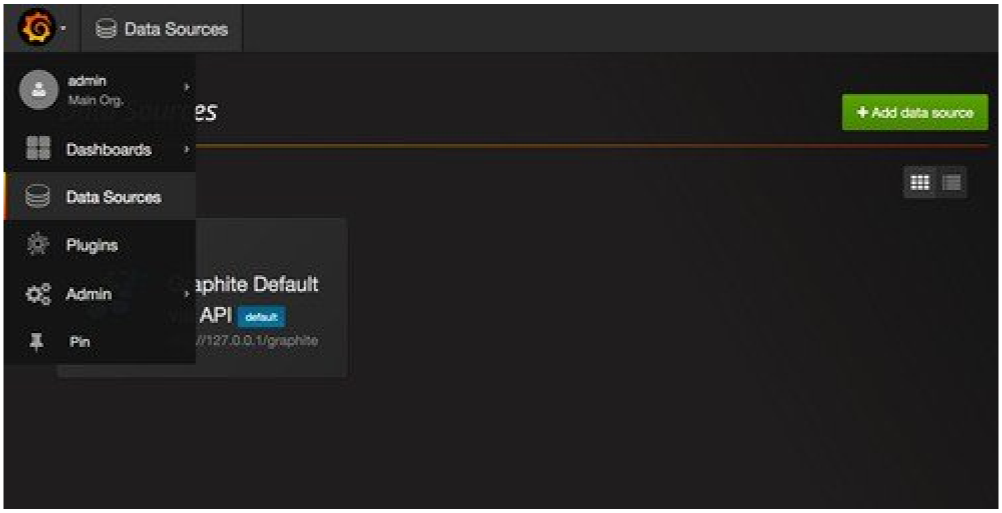
```
cat prom.yaml
 enablePrometheus: "enabled"
 statsGranularity: "10"
curl -i -X PATCH http://installer:8458/api/system/config/config -H "Content-Type:
application/yaml" --data-binary @prom.yaml
 HTTP/1.1 200 OK
  Date: Fri, 20 Apr 2018 08:38:20 GMT
  Content-Length: 0
```

Add Datasource in Grafana for Prometheus

Step 1 Login to Grafana with admin credentials.

Step 2 Click on the Grafana logo to open the sidebar menu.

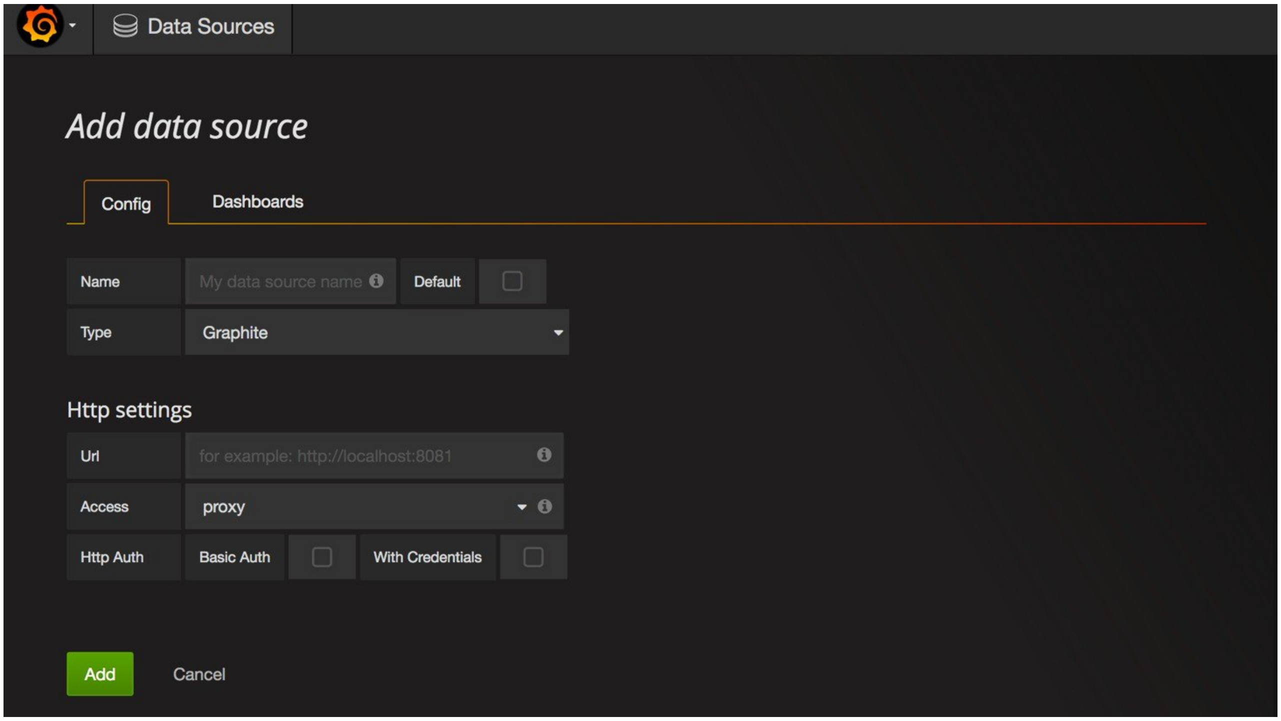
Figure 1: Sidebar Menu



Step 3 Click on **Data Sources** in the sidebar.

Step 4 Click on **Add data source**.

Figure 2: Add data source



Add data source

Name	My data source name	Default	□
Type	Graphite		

Http settings

Url	for example: http://localhost:8081		
Access	proxy		
Http Auth	Basic Auth □	With Credentials	□

Step 5 From **Type** drop-down list, select **Prometheus**.

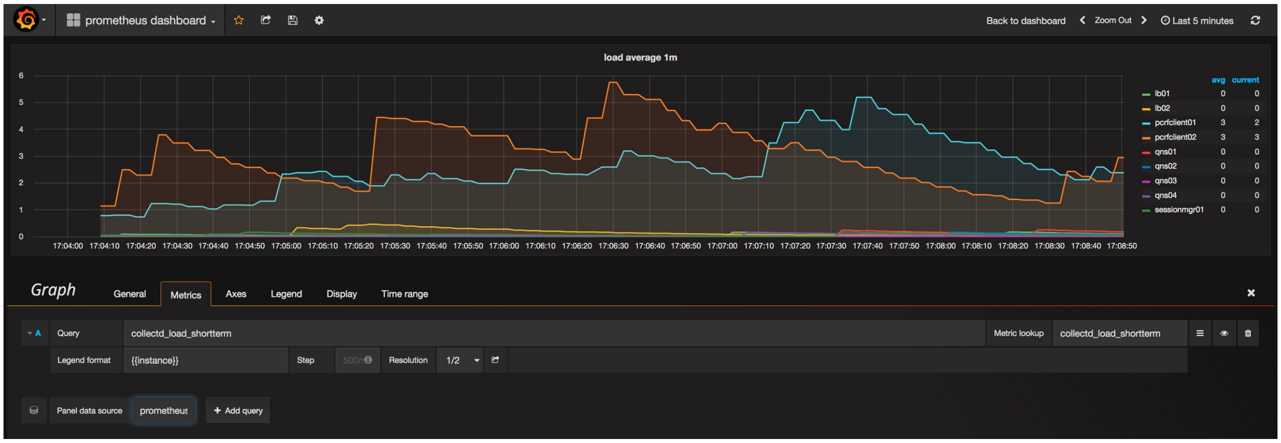
Step 6 Set the appropriate Prometheus server URL (for example, http://localhost:9090/).

Step 7 Click **Add** to save the new data source.

Step 8 Create graph with Prometheus as a data source.

For example, sample graph which gives 1 min load average of VMs.

Figure 3: Sample Graph



	avg	current
lb01	0	0
lb02	0	0
pcrfclient01	3	2
pcrfclient02	3	3
qns01	0	0
qns02	0	0
qns03	0	0
qns04	0	0
sessionmgr01	0	0

A Query: collectd_load_shortterm — Metric lookup: collectd_load_shortterm

Legend format: {{instance}} — Step: 500m — Resolution: 1/2

Panel data source: prometheus

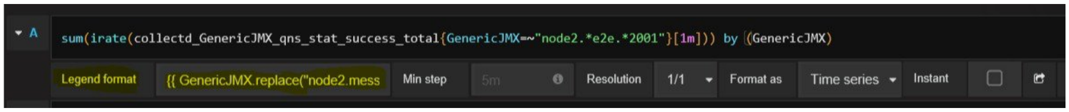
Note Legend format: The legend format needs to be modified for the upgraded version of Grafana:

- `Replace {{ GenericJMX.replace('nodeX.messages.e2e_', 'nodeX-') }} with {{GenericJMX}}`

  where, node*X* is node1/node2/node3/node4.

  Example:

  From



```
sum(irate(collectd_GenericJMX_qns_stat_success_total{GenericJMX=~"node2.*e2e.*2001"}[1m])) by (GenericJMX)
```

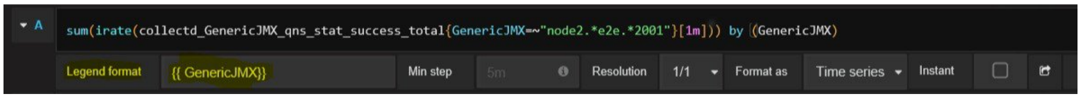
Legend format: {{ GenericJMX.replace("node2.mess

Min step: 5m

Resolution: 1/1

Format as: Time series

  To



```
sum(irate(collectd_GenericJMX_qns_stat_success_total{GenericJMX=~"node2.*e2e.*2001"}[1m])) by (GenericJMX)
```

Legend format: {{ GenericJMX}}

Min step: 5m

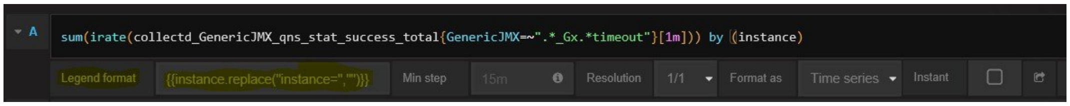
Resolution: 1/1

Format as: Time series

- `Replace {{instance.replace("instance=","")}} with {{instance}}`

  Example:

  From



```
sum(irate(collectd_GenericJMX_qns_stat_success_total{GenericJMX=~".*_Gx.*timeout"}[1m])) by (instance)
```

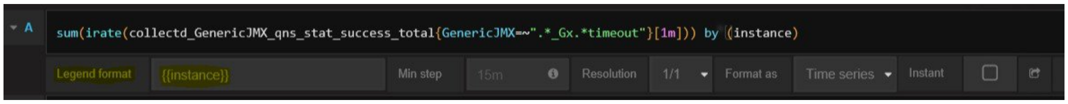
Legend format: {{instance.replace("instance=","")}}

Min step: 15m

Resolution: 1/1

Format as: Time series

  To



```
sum(irate(collectd_GenericJMX_qns_stat_success_total{GenericJMX=~".*_Gx.*timeout"}[1m])) by (instance)
```

Legend format: {{instance}}

Min step: 15m

Resolution: 1/1

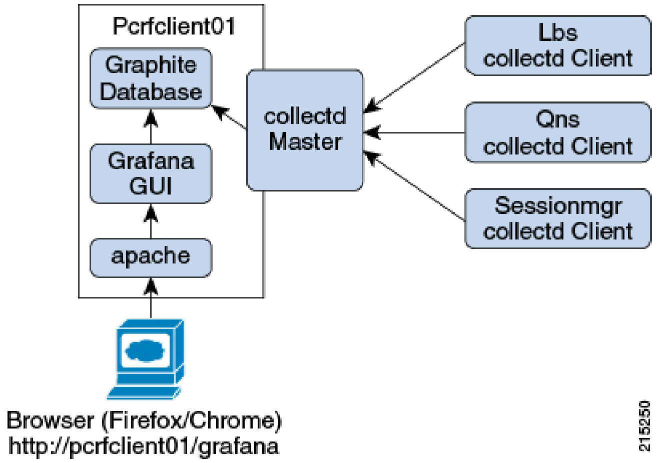
Format as: Time series

Graphite

Collected clients running on all CPS Virtual Machines (such as Policy Server (QNS), Policy Director (LB), and sessionmgr) push data to the Collected master on the pcrfclient01. The Collected master node in turn forwards the collected data to the Graphite database on the pcrfclient01.

The Graphite database stores system-related statistics such as CPU usage, memory usage, and Ethernet interface statistics, as well as application message counters such as Gx, Gy, and Sp.

Figure 4: Graphite



Pcrfclient01 and pcrfclient02 collect and store these bulk statistics independently.

As a best practice, always use the bulk statistics collected from pcrfclient01. Pcrfclient02 can be used as a backup if pcrfclient01 fails.

In the event that pcrfclient01 becomes unavailable, statistics will still be gathered on pcrfclient02. Statistics data is not synchronized between pcrfclient01 and pcrfclient02, so a gap would exist in the collected statistics while pcrfclient01 is down.

Note It is normal to have slight differences between the data on pcrfclient01 and pcrfclient02. For example, pcrfclient01 generates a file at time *t* and pcrfclient02 generates a file at time t +/- clock drift between the two machines.

Note Based on the retention period configured in `/etc/carbon/storage-schemas.conf` in pcrfclient VMs, same time period graph may look different after some time.

Default retention period is 10s:1d,60s:60d (10 seconds data points for 1 day and 60 seconds data points for 60 days). As per the default retention, first day 6 data points are available for each minute and after that, those 6 data points are aggregated and only one data point is available for a minute in Graphite.

Additional Graphite Documentation

To learn more about Graphite, refer to: http://graphite.readthedocs.org/en/latest/

For a list of all functions that can be used to transform, combine and perform computations on data stored in Graphite, refer to: http://graphite.readthedocs.org/en/latest/functions.html

Grafana

Grafana is a third-party metrics dashboard and graph editor.

Grafana provides a graphical or text-based representation of statistics and counters collected in the Prometheus database. To use Prometheus in Grafana, refer to http://docs.grafana.org/features/datasources/prometheus/.

Additional Grafana Documentation

This chapter provides information about the CPS implementation of Grafana. For more information about Grafana, or access the general Grafana documentation, refer to: http://docs.grafana.org.

Configure Grafana Users using CLI

In CPS 7.0.5 and higher releases, users must be authenticated to access Grafana. No default users are provided. In order to access Grafana, you must add at least one user as described in the following sections.

The steps mentioned in the sections describe how to add and delete users who are allowed view-only access of Grafana. In order to create or modify dashboards, refer to Grafana Administrative User, on page 9.

After adding or deleting a Grafana user, manually copy the `/var/broadhop/.htpasswd` file from the pcrfclient01 VM to the pcrfclient02 VM.

Also, run `/var/qps/bin/support/grafana_sync.sh` to synchronize the information between two OAM (pcrfclient) VMs.

There is no method to change the password for a Grafana user; you can only add and delete users. The `change_passwd.sh` script cannot be used to change the password for Grafana users.

Note The `change_passwd.sh` script changes the password on all the VMs temporarily. You also need to generate an encrypted password. To generate encrypted password, refer to *System Password Encryption* in *CPS Installation Guide for VMware*. The encrypted password must be added in the `Configuration.csv` spreadsheet. To make the new password persisent, execute `import_deploy.sh`. If the encrypted password is not added in the spreadsheet and `import_deploy.sh` is not executed, then after running `reinit.sh` script, the qns-svn user takes the existing default password from `Configuration.csv` spreadsheet.

Log on to the pcrfclient01 VM to perform any of the following operations.

Add User

Run the following command on Cluster Manager VM:

```
/usr/bin/htpasswd -s /var/www/html/htpasswd <username>
```

When prompted for a password, enter and reenter the password. This step updates htpasswd file and forces SHA encryption of the password.

After creating graphite/grafana user, CPS user needs to execute `/var/broadhop/sync_htpasswd.sh` on pcrfclient VMs or `reinit.sh` on Cluster Manager VM to synchronize created user with pcrfclient VMs.

Note Any Grafana user created using CLI on pcrfclient VM (using old method) gets overwritten after Puppet execution.

Note This user is for Grafana authentication, which you will see when you open Grafana URL https://<lbvip01>:443/grafana.

This user has nothing to do with Grafana administrative user **admin**. By default, Grafana user **admin** is reserved only for administrative activities.

Note This user can be used in Basic Authentication in Configuring Graphite User Credentials in Grafana, on page 17.

Delete User

Run the following command to delete Graphite or Grafana user:

```
/usr/bin/htpasswd -D /var/www/html/htpasswd <username>
```

After deleting graphite/grafana user, CPS user needs to execute `/var/broadhop/sync_htpasswd.sh script` on pcrfclient VMs or `reinit.sh` on Cluster Manager VM to synchronize deleted user with pcrfclient VMs.

Note Any Grafana user created or deleted using CLI on pcrfclient VM (using old method) gets overwritten after Puppet execution.

Connect to Grafana

Use the following URL to access Grafana.

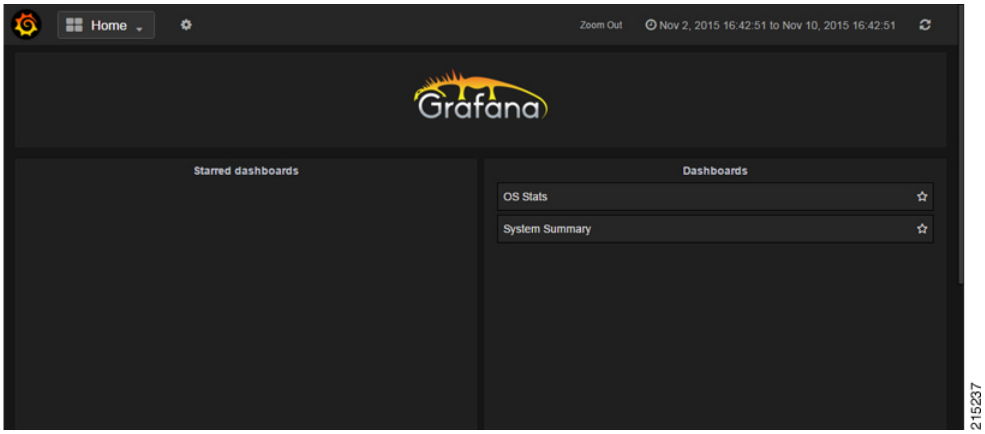
- HA: https://<*lbvip01*>:443/grafana

  Deprecated URL: https://<*lbvip01*>:9443/grafana

- All in One: http://<*ip*>:80/grafana

When prompted, enter the username and password of a user you created in Configure Grafana Users using CLI, on page 7.

Figure 5: Grafana Home Screen



Grafana Administrative User

Note

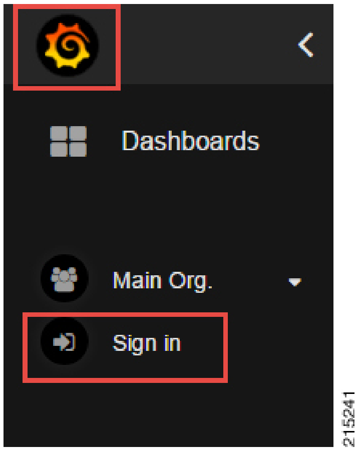
- Grafana administrative user **admin** is independent user stored in `grafana.db`. The admin user is used for administrative tasks such as:
  - Adding/updating data source for Graphite or Prometheus.
  - Adding another administrative user.
  - Change password of existing administrative users.
- You can add user using `htpasswd` utility that is used for authentication for logging into Grafana for normal use.Grafana administrative user has nothing to do with any user added using `htpasswd` utility.

Log in as Grafana Admin User

To create or modify dashboards in Grafana, you must log in as the Grafana administrative user.

Step 1 Click the Grafana logo in the upper left corner of your screen.

Figure 6: Grafana Logo



Step 2 Click **Sign In**.

Step 3 Enter the administrative username and password: `admin/admin`

Change Grafana Admin User Credentials

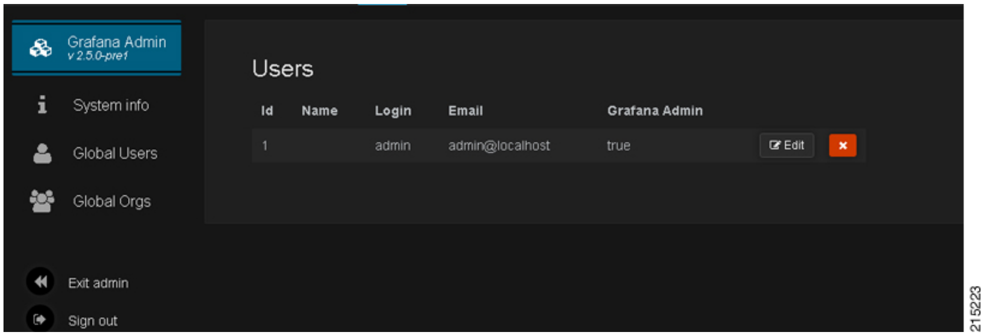
Step 1 Log in as the administrative user (`admin/admin`).

Step 2 Click the Grafana logo, then click **Grafana admin**.

Step 3 Click **Global Users**.

Step 4 Click **Edit**.

Figure 7: Changing Grafana Admin User Credentials



Users

Id	Name	Login	Email	Grafana Admin		
1		admin	admin@localhost	true	Edit	×

Add a Grafana User

Note The steps mentioned here can be performed only by **administrative** user.

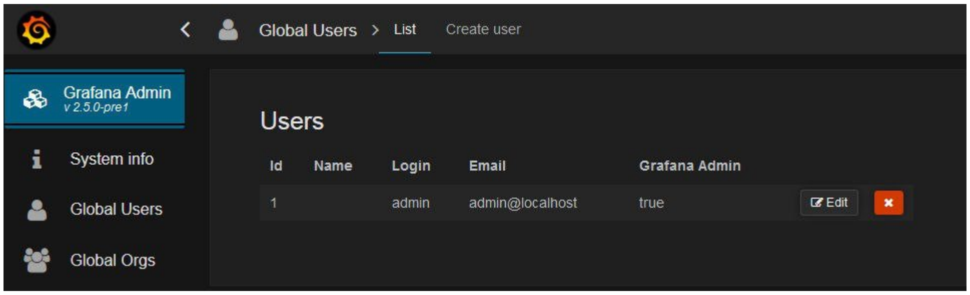
Step 1 Click the Grafana logo in the upper left corner of your screen.

Step 2 Click **Sign in**. Enter the administrative username and password.

Step 3 Click **Grafana admin** from the left side to open the **System info** pane on the right side.

Step 4 Click **Global Users** to open a pane. By default, the **List** tab appears displaying the list of users currently configured in Grafana.

Figure 8: List Tab

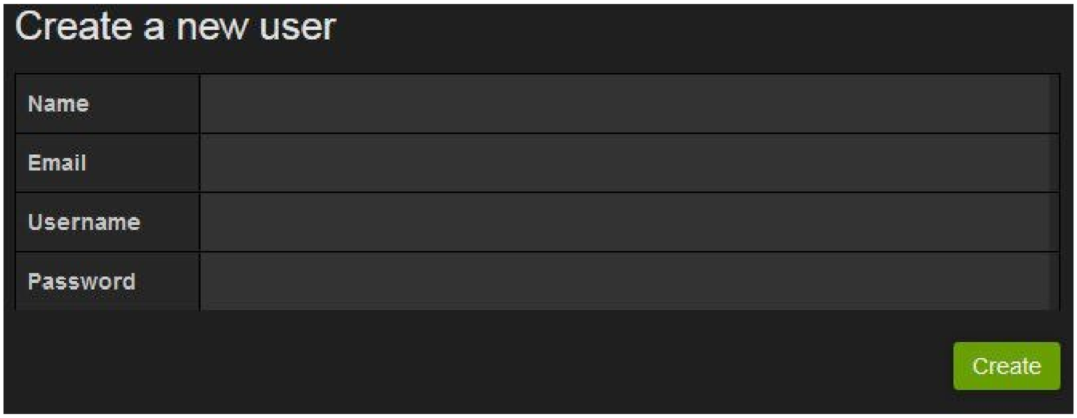


Users

Id	Name	Login	Email	Grafana Admin
1		admin	admin@localhost	true

Step 5 Click **Create user** at the top to open **Create a new user** pane.

Figure 9: Create a new user



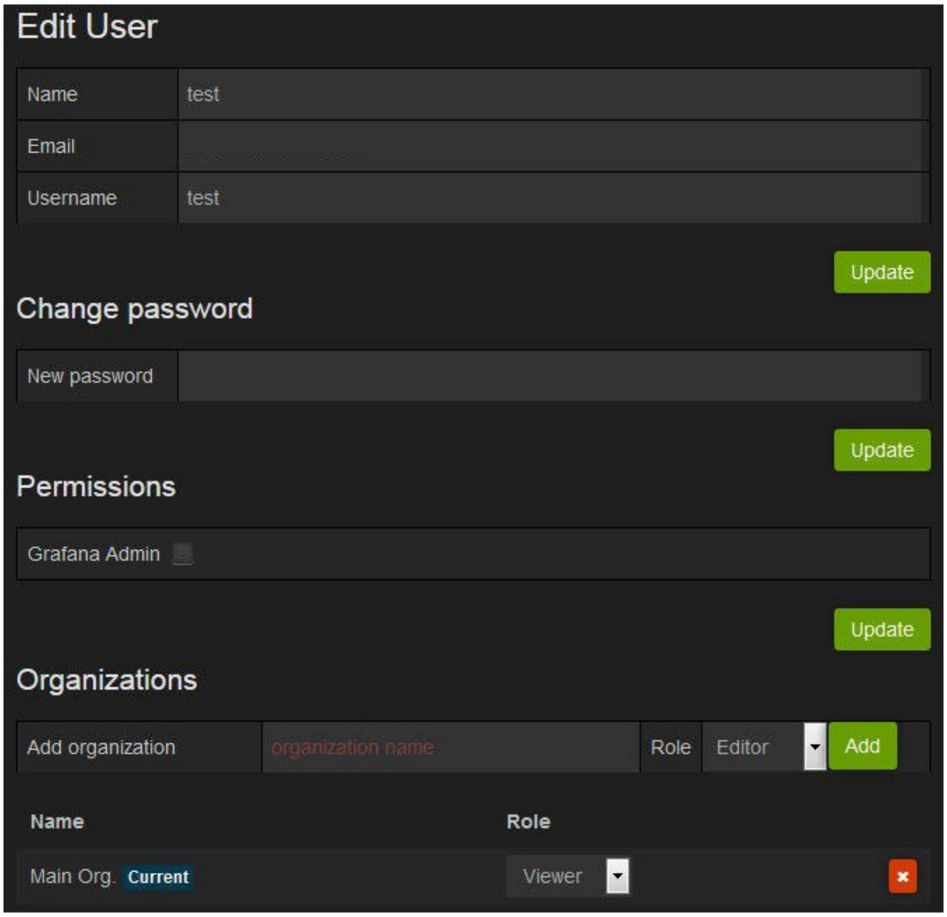
Step 6 Enter the required parameters in *Name*, *Email*, *Username* and *Password* fields.

Step 7 Click **Create** to create the grafana user.

Step 8 You will see the newly added user in the **List** tab. By default, the new user will have only **Viewer** rights.

Step 9 Click **Edit** to open **Edit User** pane. Only administrative user can update/modify the user properties.

Figure 10: Edit User Information



Edit User

Name: test

Email:

Username: test

Change password

New password:

Permissions

Grafana Admin □

Organizations

Add organization: organization name — Role: Editor

Name	Role
Main Org. Current	Viewer

Change the Role of Grafana User

You can also change the rights of the user from the main page.

Note The steps mentioned here can be performed only by **administrative** user.

Click **Main Org.** drop-down list to select **Users**. This will open **Organization users** pane, where you can change the role of a user from **Role** drop-down list.

The user can have Admin/Viewer/Editor/Read Only Editor roles.

- **Admin:** An admin user can view, update and create dashboards. Also the admin can edit and add data sources and organization users.
- **Viewer:** A viewer can only view dashboards, not save or create them.

- **Editor:** An editor can view, update and create dashboards.
- **Read Only Editor:** This role behaves just like the Viewer role. The only difference is that you can edit graphs and queries but not save dashboards. The Viewer role has been modified in Grafana 2.1 so that users assigned this role can no longer edit panels.

Add an Organization

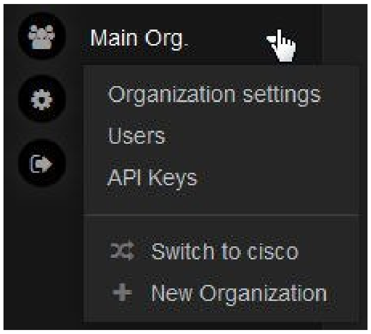
Grafana supports multiple organizations in order to support a wide variety of deployment models, including using a single Grafana instance to provide service to multiple potentially untrusted Organizations.

In many cases, Grafana will be deployed with a single Organization. Each Organization can have one or more Data Sources. All Dashboards are owned by a particular Organization.

Note The steps mentioned here can be performed only by **administrative** user.

Step 1 Click **Main Org.** drop-down list to select **New Organization**.

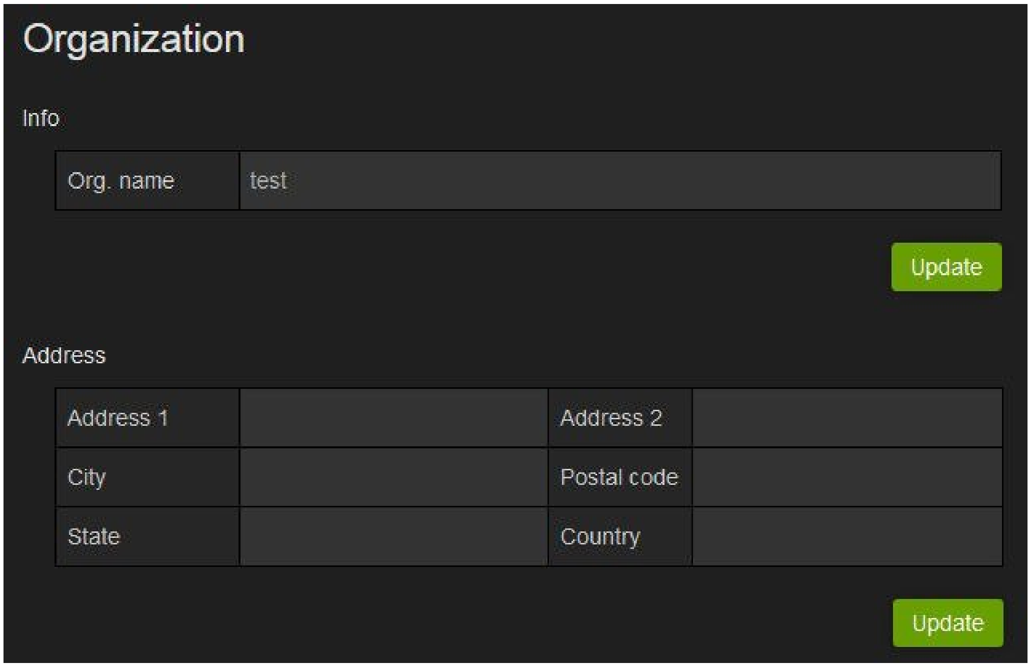
Figure 11: New Organization



Step 2 This will open a new pane **Add Organization**. Enter organization name in *Org. name* field. For example, test.

Step 3 After adding the name, click **Create** to open **Organization** pane.

Figure 12: Organization



Organization

Info

Org. name	test

Address

Address 1		Address 2	
City		Postal code	
State		Country	

In this pane, you can modify the organization name and other organization information. After modifying the information, click **Update** to update the information.

Move Grafana User to another Organization

Note The steps mentioned here can be performed only by **administrative** user.

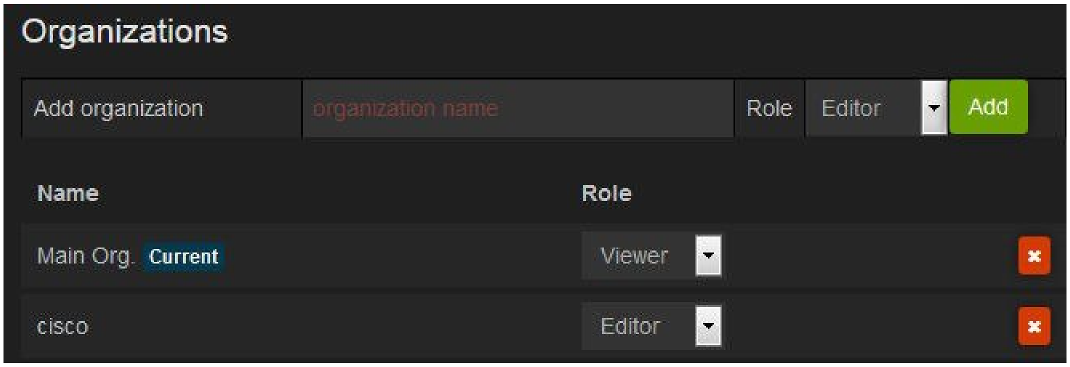
Step 1 Click **Grafana admin** from the main page to **System Info** page.

Step 2 Click **Global Users** from the left pane to open **Users** pane on the right.

Step 3 Click **Edit** against the user for whom you want to make the changes.

Step 4 Under **Organizations** section, you can add the user to some other organizations.

Figure 13: Move User to another Organization



Name	Role
Main Org. Current	Viewer
cisco	Editor

Step 5 In *Add organization* field, you need to enter the name of the new organization.

Step 6 You can also change the role of the user from the **Role** drop-down list.

Step 7 After adding the required information, click **Add** to add the user into a new organization.

Step 8 In the above example, you can see that the user is added to the new organization. If you want to remove the user from pervious organization, click the **red cross** at the end.

Configure Grafana for First Use

After an initial installation or after upgrading an existing CPS deployment which used Grafana, you must perform the steps in the following sections to validate the existing data sources.

Migrate Existing Grafana Dashboards

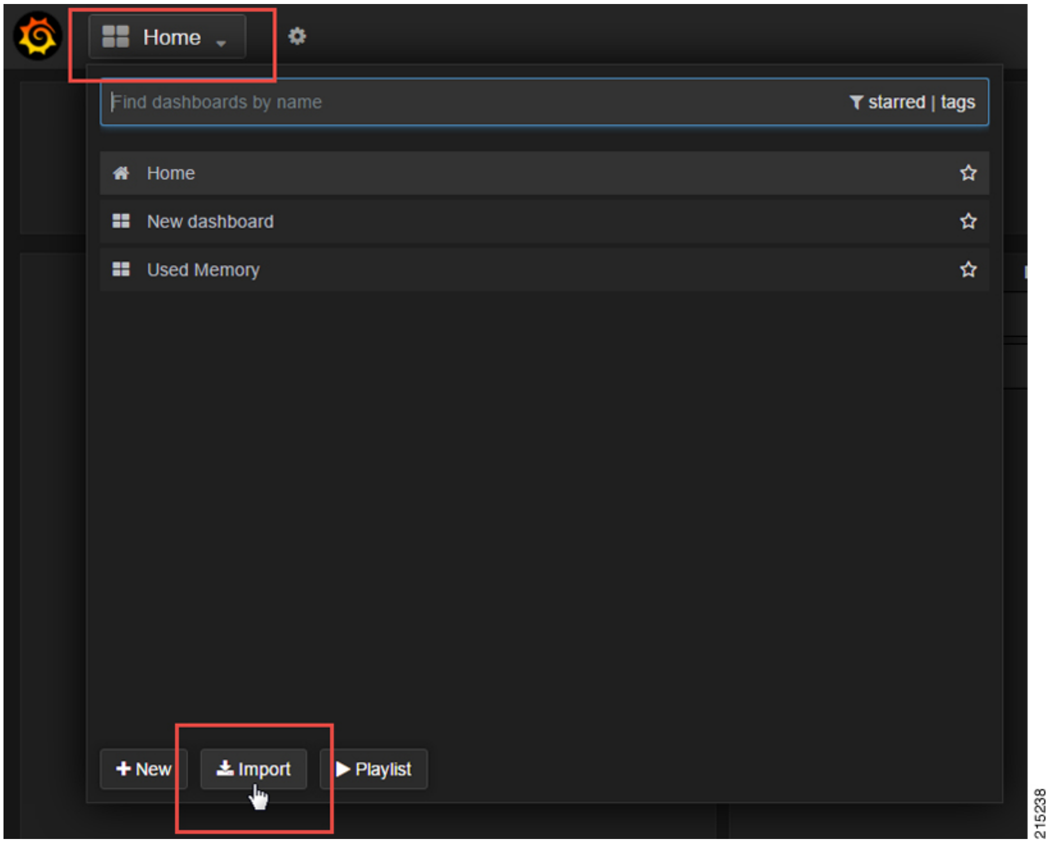
During an upgrade of CPS (and Grafana), saved dashboard templates remain intact.

After upgrading an existing CPS deployment, you must manually migrate any existing Grafana dashboards.

Step 1 Sign in as the Grafana Administrative User. For more information, refer to Grafana Administrative User, on page 9.

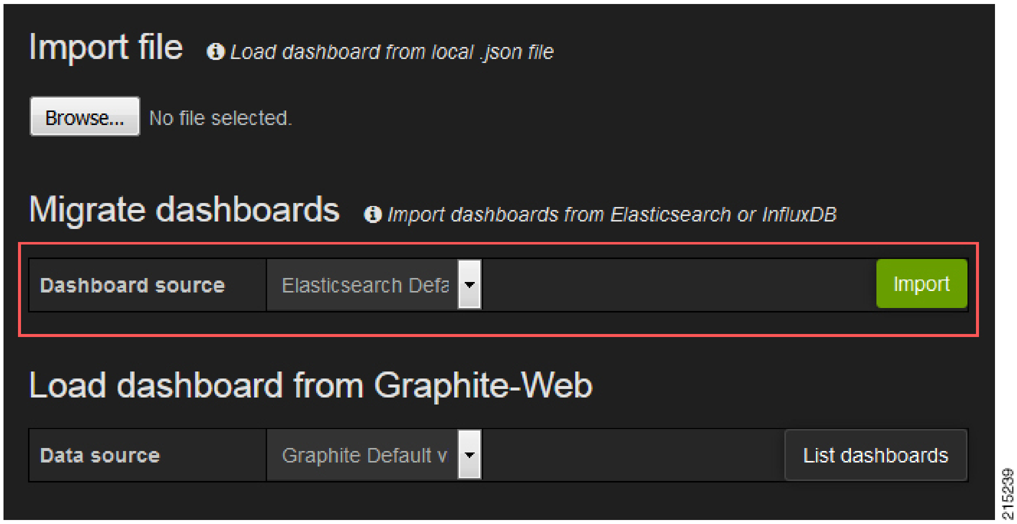
Step 2 Click **Home** at the top of the Grafana window and then click **Import** as shown below:

Figure 14: Import



Step 3 In the Migrate dashboards section, verify that **Elasticsearch Def** (Elasticsearch Default via API) is listed, then click **Import**.

Figure 15: Import File



Step 4 All existing dashboards are imported and should now be available.

Configuring Graphite User Credentials in Grafana

Step 1 Log into Grafana with admin credentials.

Step 2 Click Grafana home icon.

Step 3 Select Data Sources.

Note Default link to datasources is https://<LB VIP>/grafana/datasources.

Step 4 Select Graphite default table is added.

Step 5 In HTTP Auth table, select basic auth.

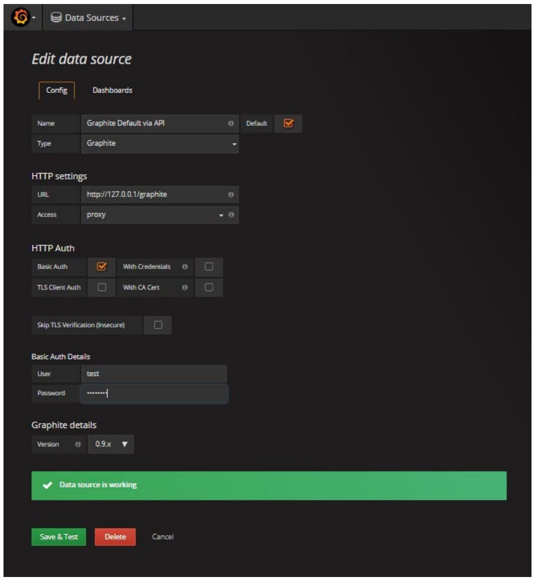
After selected new **Basic Auth Details** table is added.

Note Add/update username/password, which you created using htpasswd Add User, on page 7.

Step 6 Enter Graphite DB credentials.

Step 7 Click **Save and Test**.

After successful testing, "Data source is working message" message is displayed:



Edit data source

Name: Graphite Default via API   Default ☑

Type: Graphite

HTTP settings

URL: http://127.0.0.1/graphite

Access: proxy

HTTP Auth

Basic Auth ☑   With Credentials □

TLS Client Auth □   With CA Cert □

Skip TLS Verification (Insecure) □

Basic Auth Details

User: test

Password: ••••••••

Graphite details

Version: 0.9.x

✓ Data source is working

Note

- The above graphite configuration screenshot is a sample configuration. The options can vary depending on the CPS version installed.
- Grafana supports Graphite data source credential configuration capability. Graphite data source requires common data source credential to be configured using Grafana for Grafana user. Data source credential must be configured before upgrade/migration starts or after fresh installation. If you fail to add the user, then Grafana will not have an access to Graphite database and you will get continuous prompts for Graphite/Grafana credentials.
- All Grafana users configured will be available after migration/upgrade or fresh installation. However, you need to configure the graphite data source in Grafana UI.

Accessing Graphite Database Using CLI

All requests to Graphite database require a valid username and password. You need to use `-u` flag in the request followed by the username and password.

```
curl -u <username>:<password> -G http://< GRAPHITEURL>
```

Note Password must be provided in plain text format.

Changing Default graphite_default User Password

`graphite_default` user is created during the fresh installation. Currently, CPS doesn't automatically sychronize the `graphite_default` user password change, once CPS is deployed. If you want to change the password for `graphite_default` user, peform the following steps. The steps must be executed from Cluster Manager.

Before you begin

Take the backup of old password files for future reference.

```
/var/www/html/htpasswd
```

```
/root/.graphite_default
```

Step 1 Remove old password entry for the default user (graphite_default).

```
/usr/bin/htpasswd -D /var/www/html/htpasswd graphite_default
```

Step 2 Create new password for the default user.

```
/usr/bin/htpasswd -s /var/www/html/htpasswd graphite_default
```

Step 3 Create encrypted password.

```
/var/qps/bin/support/mongo/encrypt_passwd.sh NEW_PASSWORD
```

where, *NEW_PASSWORD* is the password used for `/usr/bin/htpasswd -s /var/www/html/htpasswd graphite_default`

Step 4 Update `/root/.graphite_default` with the encrypted password.

```
echo <encryptedpassword> > /root/.graphite_default
```

where, <*encryptedpassword*> is the encrypted password for `graphite_default` user.

Step 5 Check the facter parameter graphite_default before updating.

The following is a sample output.

```
facter | grep graphite_default
graphite_default => 458B2FE273EAA3A93450B186227C8543
```

Step 6 Update the facter parameters to change `graphite_default` with the latest password.

For OpenStack: `/var/qps/bin/support/config_cluman.sh`

For VMware: `/var/qps/install/current/scripts/import/import_deploy.sh`

Step 7 Build the changes.

```
/var/qps/bin/build/build_all.sh
```

Step 8 Verify the facter parameter for `graphite_default` is updated with the latest `/root/.graphite_default` content.

The following is a sample output.

```
facter | grep graphite
graphite_default => 238B2FE273EAA3A93450B186227C8143
```

Step 9 Run `vm-init` on pcrfclient01 and pcrfclient02 VMs to update `graphite_default` for all the users.

```
ssh pcrfclient01 "/etc/init.d/vm-init”
ssh pcrfclient02 "/etc/init.d/vm-init”
```

Manual Dashboard Configuration using Grafana

Grafana enables you to create custom dashboards which provide graphical representations of data by fetching information from the Prometheus database. Each dashboard is made up of panels spread across the screen in rows.

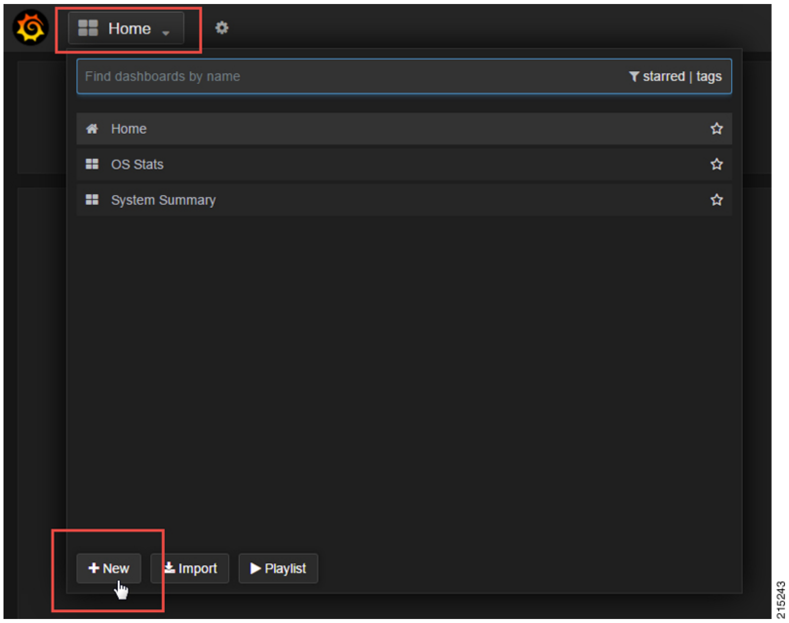
Note CPS includes a series of preconfigured dashboard templates. To use these dashboards, refer to Updating Imported Templates.

Create a New Dashboard Manually

Step 1 Sign-in as a Grafana Administrative user. For more information, see Grafana Administrative User, on page 9.

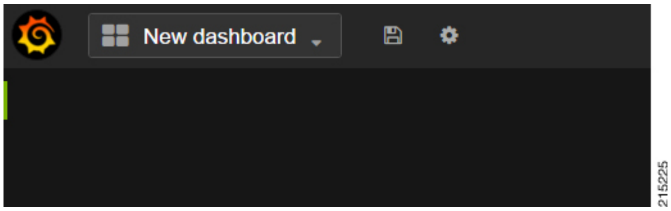
Step 2 Click **Home** at the top of the Grafana window and select **New** as shown below:

Figure 16: Home



A blank dashboard is created.

Figure 17: Blank Dashboard



Step 3 At the top of the screen, click the gear icon, then click **Settings**.

Figure 18: Gear Icon

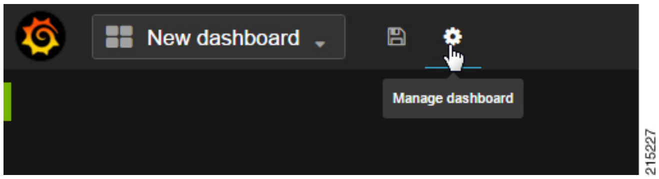
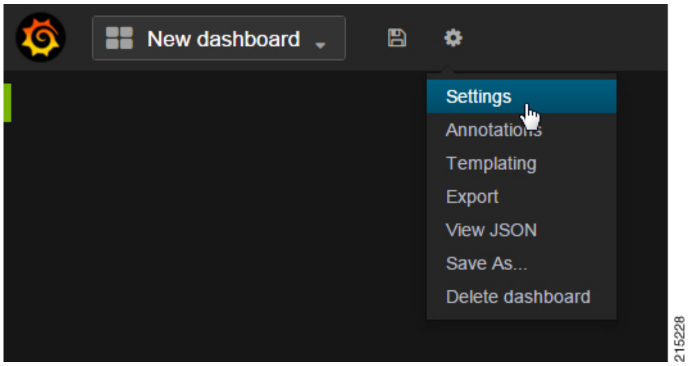
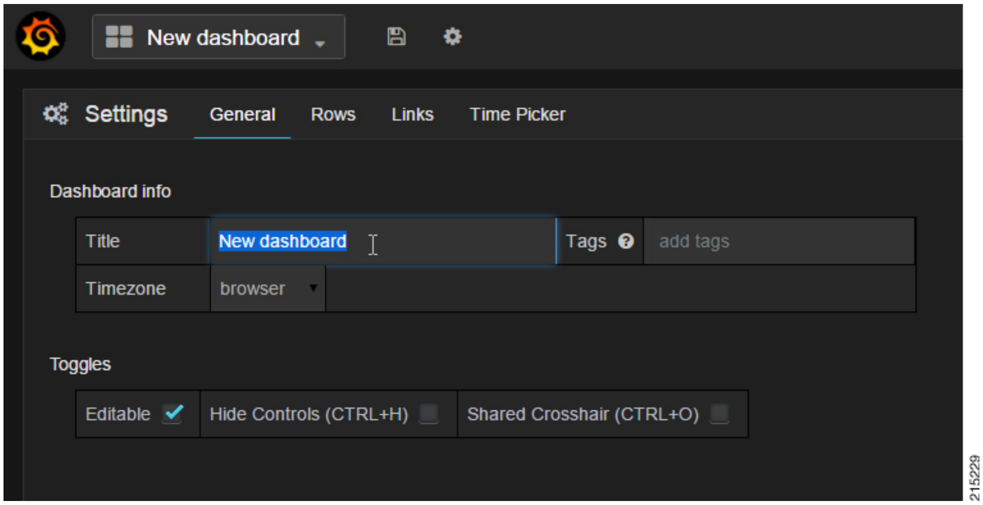


Figure 19: Settings



Step 4 Provide a name for the dashboard and configure any other Dashboard settings. When you have finished, click the **X** icon in the upper right corner to close the setting screen.

Figure 20: Name for Dashboard



Dashboard info

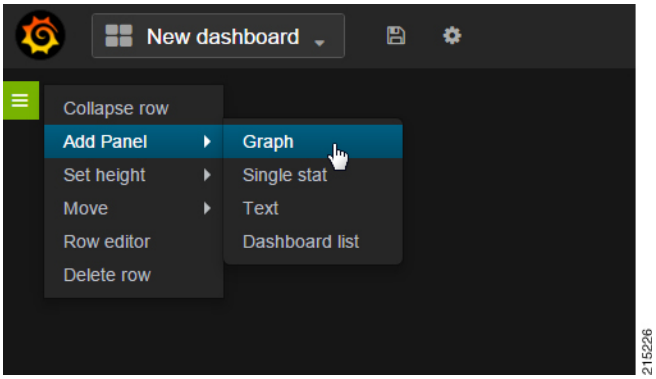
Title	New dashboard	Tags	add tags
Timezone	browser		

Toggles

Editable ☑ Hide Controls (CTRL+H) □ Shared Crosshair (CTRL+O) □

Step 5 To add a graph to this dashboard, hover over the green box on the left side of the dashboard, then point to **Add Panel**, then click **Graph**.

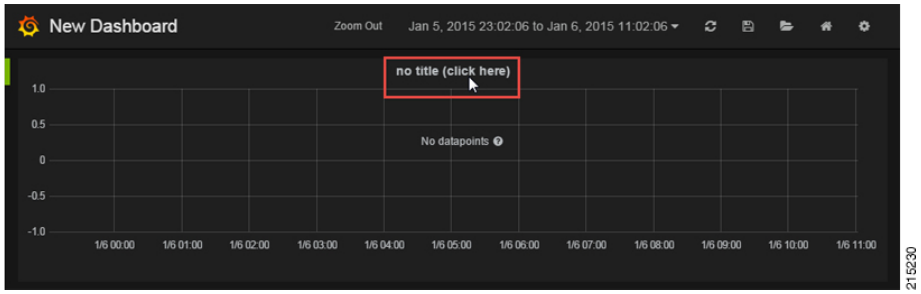
Figure 21: Add Graph to Dashboard



Configure Data Points for the Panel

Step 1 Click the panel title, as shown below, then select **Edit**.

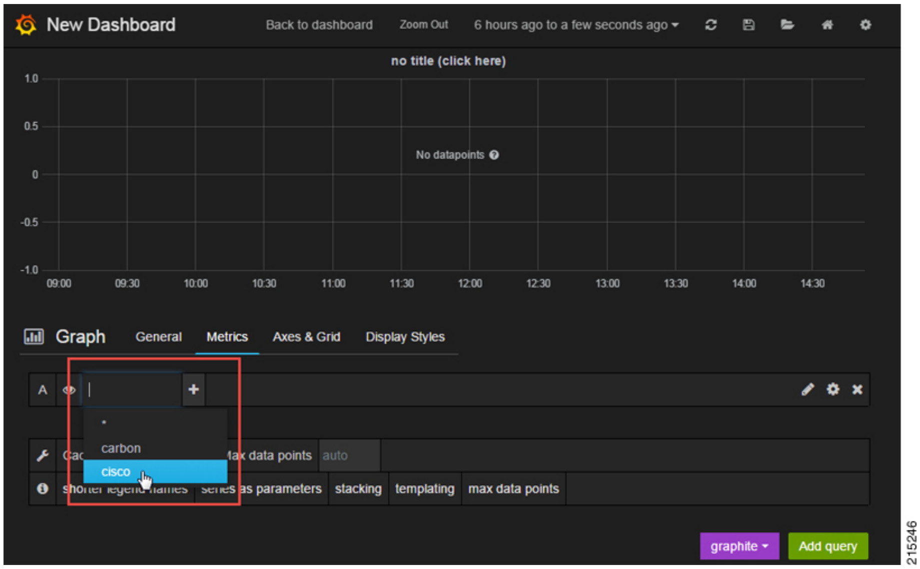
Figure 22: Edit



Step 2 Select the necessary metrics by clicking on the select metric option provided in the query window. A drop-down list appears from which you can choose the required metrics.

Select metrics by clicking select metric repeatedly until the lowest level of the hierarchy.

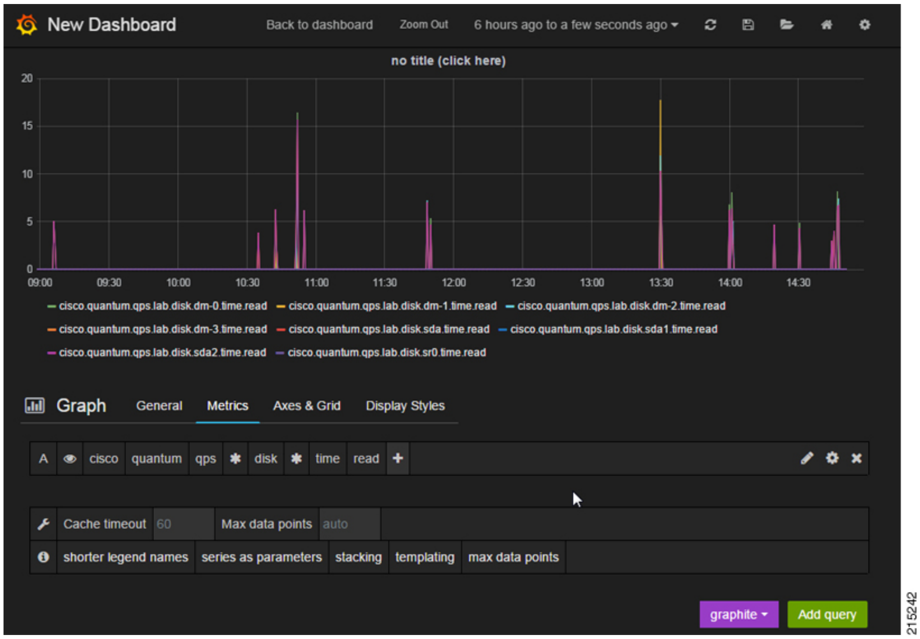
Figure 23: Metric Selection



Note Clicking the '*' option in the drop-down list selects all the available metrics.

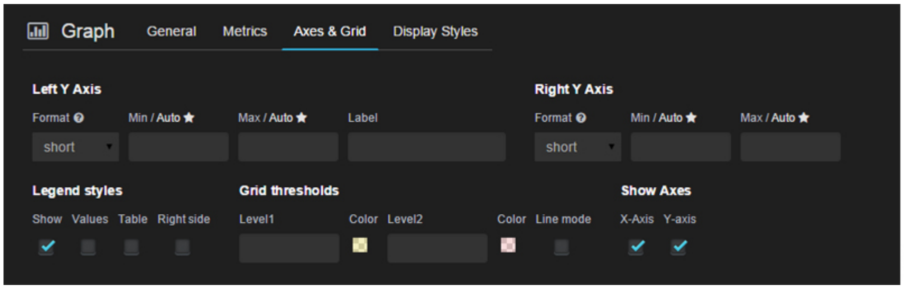
Step 3 Click the '+' tab to add aggregation functions for the selected metrics. the monitoring graph is displayed as shown below.

Figure 24: Aggregation Functions


New Dashboard
Back to dashboard
Zoom Out
6 hours ago to a few seconds ago ▾
no title (click here)
20
15
10
5
0
09:00
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
14:30
cisco.quantum.qps.lab.disk.dm-0.time.read
cisco.quantum.qps.lab.disk.dm-1.time.read
cisco.quantum.qps.lab.disk.dm-2.time.read
cisco.quantum.qps.lab.disk.dm-3.time.read
cisco.quantum.qps.lab.disk.sda.time.read
cisco.quantum.qps.lab.disk.sda1.time.read
cisco.quantum.qps.lab.disk.sda2.time.read
cisco.quantum.qps.lab.disk.sr0.time.read
Graph
General
Metrics
Axes & Grid
Display Styles
A
cisco
quantum
qps
✱
disk
✱
time
read
+
Cache timeout
60
Max data points
auto
shorter legend names
series as parameters
stacking
templating
max data points
graphite ▾
Add query
215242

Step 4 The x-axis and y-axis values can be configured in the **Axes & Grid** tab.

Figure 25: Axes and Grid



Left Y Axis

Format: short | Min / Auto | Max / Auto | Label

Right Y Axis

Format: short | Min / Auto | Max / Auto

Legend styles: Show ✓ | Values □ | Table □ | Right side □

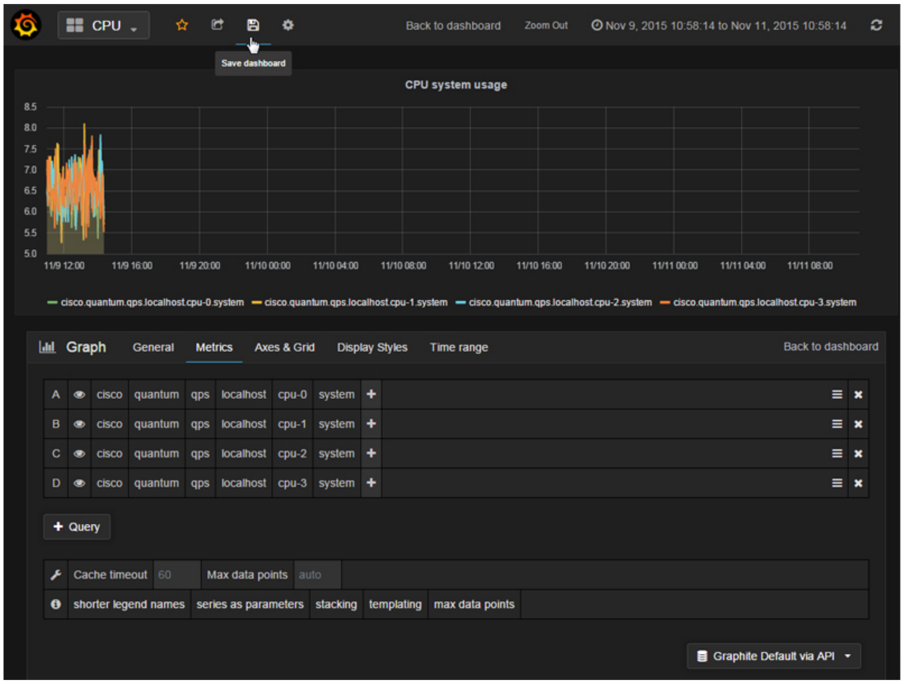
Grid thresholds: Level1 | Color | Level2 | Color | Line mode □

Show Axes: X-Axis ✓ | Y-axis ✓

Step 5 Click the disk icon (Save dashboard) at the top of the screen, as shown in the following image.

Note The changes to this dashboard are lost if you do not click the **Save** icon.

Figure 26: Save

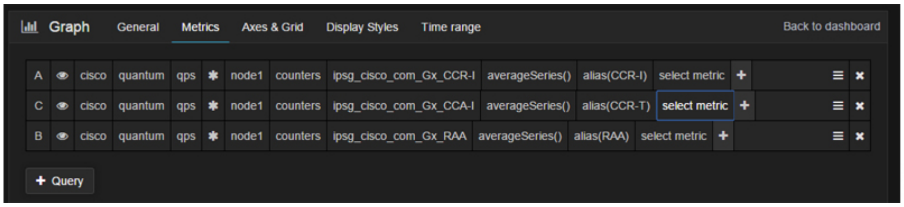


```
A  cisco  quantum  qps  localhost  cpu-0  system
B  cisco  quantum  qps  localhost  cpu-1  system
C  cisco  quantum  qps  localhost  cpu-2  system
D  cisco  quantum  qps  localhost  cpu-3  system
```

Cache timeout: 60 | Max data points: auto

Graphical representation of application-messages such as - CCR, CCA, Gx, Gy, LDAP, Rx messages and so on, can be configured in the dashboard panel by using the queries shown in the below figure.

Figure 27: Graphical Representation



```
A  cisco  quantum  qps  *  node1  counters  ipsg_cisco_com_Gx_CCR-I  averageSeries()  alias(CCR-I)  select metric
C  cisco  quantum  qps  *  node1  counters  ipsg_cisco_com_Gx_CCA-I  averageSeries()  alias(CCR-T)  select metric
B  cisco  quantum  qps  *  node1  counters  ipsg_cisco_com_Gx_RAA  averageSeries()  alias(RAA)  select metric
```

Configure Useful Dashboard Panels

The following section describes the configuration of several useful dashboard panels that can be used while processing Application Messages. Configure the dashboard panel as shown in the screens below.

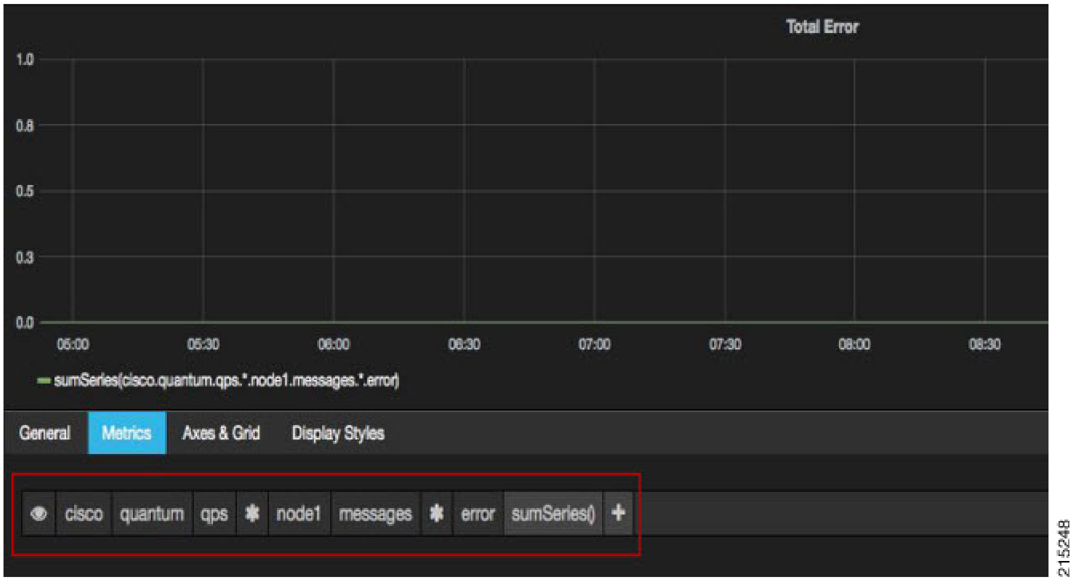
For more information on panels, see http://docs.grafana.org/features/panels/dashlist/.

Note It is recommended to have panel option in Grafana dashboard. This is required so that when the dashboard is loaded only necessary graphs can be expanded for which statistics need to be seen. This reduces load on pcrfclient VMs as they do not need to fetch lot of statistics.

Total Error:

This dashboard panel lists the errors found during the processing of Application Messages. To configure Total Error dashboard panel, create a panel with name 'Total Error' and configure its query as shown:

Figure 28: Total Error Dashboard



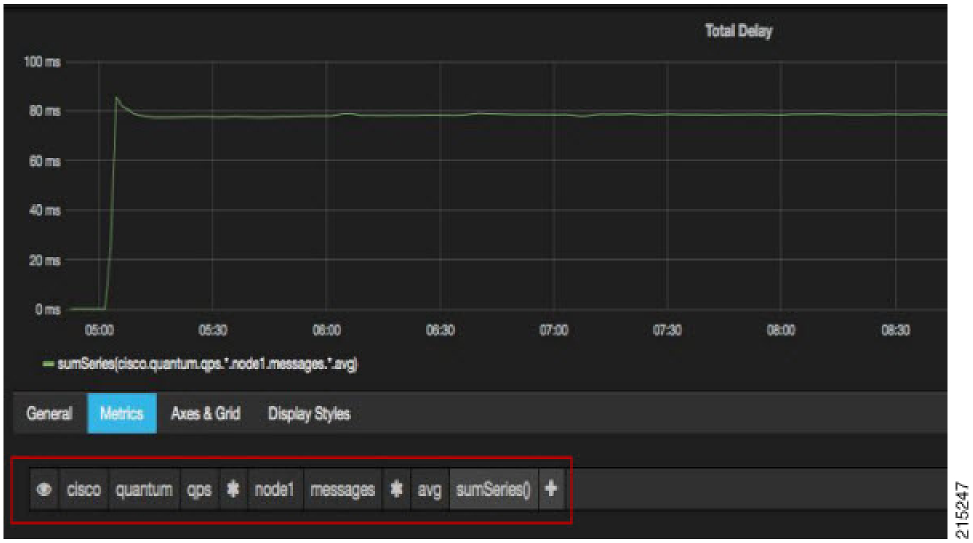
sumSeries(cisco.quantum.qps.*.node1.messages.*.error)

cisco quantum qps * node1 messages * error sumSeries()

Total Delay:

This dashboard panel displays the total delay in processing various Application Messages. To configure Total Delay dashboard panel, create a panel with name Total Delay and configure its query as shown:

Figure 29: Total Delay Dashboard



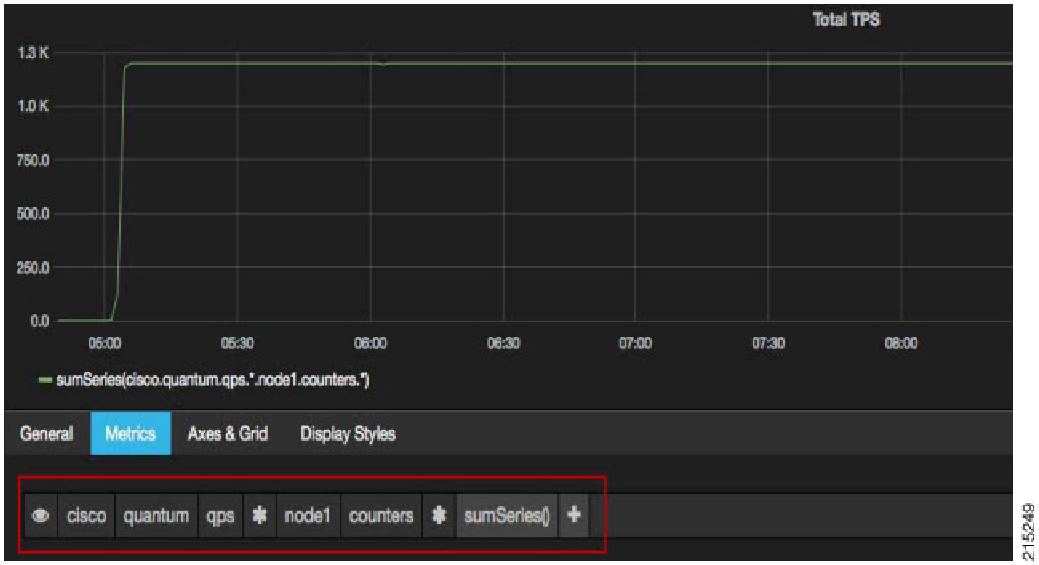
sumSeries(cisco.quantum.qps.*.node1.messages.*.avg)

cisco quantum qps * node1 messages * avg sumSeries() +

Total TPS:

This panel displays the total TPS of CPS system. Total TPS count includes all Gx, Gy, Rx, Sy, LDAP and so on. The panel can be configured as shown below:

Figure 30: Total TPS



sumSeries(cisco.quantum.qps.*.node1.counters.*)

cisco quantum qps * node1 counters * sumSeries() +

Updating Imported Templates

Some of the preconfigured templates (such as Diameter statistics panels) have matrices configured which are specific to a particular set of Diameter realms. These panels need to be reconfigured to match customer specific Diameter realms.

For example, the Gx P-GW panel in the Diameter Statistics dashboard does not fetch the stats and displays the message “No Datapoints”. The probable reasons could be:

- Matrices used in query uses matrices specific to particular Diameter realm which is different on customer setup.
- No application call of such type has ever landed on CPS Policy Directors (LBs) (no Diameter call from the P-GW has ever landed on Policy Director after the Grafana setup).

Copy Dashboards and Users to pcrfclient02

As a best practice, the internal Grafana database should be kept in sync between pcrfclient01 and pcrfclient02. This sync operation should be performed after any dashboard or Grafana user is migrated, updated, added or removed.

Under normal operating conditions, all Grafana operations occur from pcrfclient01. In the event of a pcrfclient01 failure, pcrfclient02 is used as backup, so keeping the database in sync provides a seamless user experience during a failover.

The following steps copy all configured Grafana dashboards, Grafana data sources, and Grafana users configured on pcrfclient01 to pcrfclient02.

Log in to the pcrfclient01 VM and run the following command:

```
/var/qps/bin/support/grafana_sync.sh
```

As a precaution, the existing database on pcrfclient02 is saved as a backup in the `/var/lib/grafana` directory.

Configure Garbage Collector KPIs

The following sections describe the steps to configure Garbage Collector (GC) KPIs in Grafana:

- Backend changes: Changes in the collectd configuration so that GC related KPIs will be collected by collectd and stored in graphite database.
- Frontend changes: Changes in Grafana GUI for configuring metrics for GC graph.

Backend Changes

Check if the following changes are already present in the jmxplugin.conf file. If already configured, then skip this section and move to configuring the Grafana dashboard.

Step 1 Edit `/etc/puppet/modules/qps/templates/collectd_worker/collectd.d/jmxplugin.conf` on the Cluster Manager VM as described in the following steps.

Step 2 Verify that the JMX plugin is enabled. The following lines must be present in the `jmxplugin.conf` file.

```
JVMARG has path for jmx jar

JVMARG

-Djava.class.path=/usr/share/collectd/java/collectd-api.jar/usr/share/collectd/java/generic-jmx.jar
```

```
And GenericJMX plugin is loaded

LoadPlugin org.collectd.java.GenericJMX
```

Step 3 Add an Mbean entry for garbage collector mbean in GenericJMX plugin so that statistics from this mbean will be collected.

```
  # Garbage collector information
<MBean "garbage_collector">
      ObjectName "java.lang:type=GarbageCollector,*"
      InstancePrefix "gc-"
      InstanceFrom "name"
<Value>
        Type "invocations"
        #InstancePrefix ""
        #InstanceFrom ""
        Table false
        Attribute "CollectionCount"
</Value>
<Value>
        Type "total_time_in_ms"
        InstancePrefix "collection_time"
        #InstanceFrom ""
        Table false
        Attribute "CollectionTime"
</Value>
</MBean>
```

Step 4 For every “Connection” block in `jmxplugin.conf` file add the entry for garbage collector mbean.

For example:

```
  <Connection>
    InstancePrefix "node1."
  ServiceURL "service:jmx:rmi:///jndi/rmi://localhost:9053/jmxrmi"
  Collect "garbage_collector"
  Collect "java-memory"
  Collect "thread"
  Collect "classes"
  Collect "qns-counters"
  Collect "qns-actions"
  Collect "qns-messages"
  </Connection>]
```

Step 5 Save the changes to the `jmxplugin.conf` file then synchronize the changes to all CPS VMs as follows:

a) Go to the `/var/qps/install/current/scripts/build/` directory on the Cluster Manager and execute the following script:

```
./build_puppet.sh
```

b) Go to the `/var/qps/install/current/scripts/upgrade/` directory on the Cluster Manager and execute the following command:

```
./reinit.sh
```

c) Restart the collectd service on all VMs by running the following command on each VM in the CPS cluster:

```
monit restart collectd
```

Frontend Changes

The frontend changes must be done in the Grafana GUI.

Step 1 Create a new Grafana dashboard. For more information, see Manual Dashboard Configuration using Grafana, on page 19.

Step 2 In the **Metrics** tab of the new dashboard, configure queries for GC related KPIs.

The query needs to be configured in the following format:

```
cisco.quantum.qps.<hostname>.node*. gc*.total_time_in_ms-collection_time
cisco.quantum.qps.<hostname>.node*.gc*.invocations
```

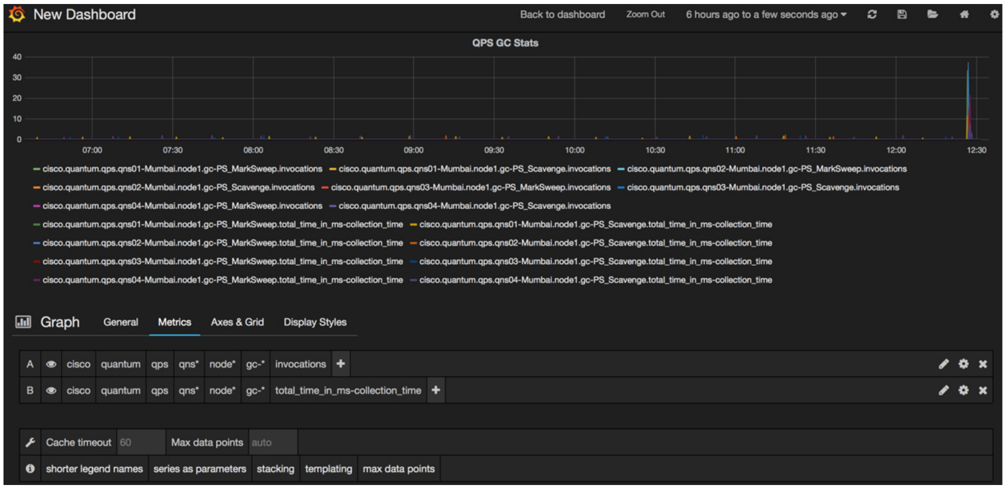
where, *<hostname>* is regular expression for the name of hosts from which KPI needs to be reported.

If this is a High Availability (HA) CPS deployment, KPIs need to be reported from all Policy Server (QNS) VMs.

Assuming the Policy Server (QNS) VMs have “qns” in their hostname, then a regular expression would be *qns*. This would report data for all VMs that have a hostname containing “qns” (qns01 qns02 and so on).

- HA Setup

Figure 31: On HA Setup



- cisco.quantum.qps.qns01-Mumbai.node1.gc-PS_MarkSweep.invocations
- cisco.quantum.qps.qns01-Mumbai.node1.gc-PS_Scavenge.invocations
- cisco.quantum.qps.qns02-Mumbai.node1.gc-PS_MarkSweep.invocations
- cisco.quantum.qps.qns02-Mumbai.node1.gc-PS_Scavenge.invocations
- cisco.quantum.qps.qns03-Mumbai.node1.gc-PS_MarkSweep.invocations
- cisco.quantum.qps.qns03-Mumbai.node1.gc-PS_Scavenge.invocations
- cisco.quantum.qps.qns04-Mumbai.node1.gc-PS_MarkSweep.invocations
- cisco.quantum.qps.qns04-Mumbai.node1.gc-PS_Scavenge.invocations
- cisco.quantum.qps.qns01-Mumbai.node1.gc-PS_MarkSweep.total_time_in_ms-collection_time
- cisco.quantum.qps.qns01-Mumbai.node1.gc-PS_Scavenge.total_time_in_ms-collection_time
- cisco.quantum.qps.qns02-Mumbai.node1.gc-PS_MarkSweep.total_time_in_ms-collection_time
- cisco.quantum.qps.qns02-Mumbai.node1.gc-PS_Scavenge.total_time_in_ms-collection_time
- cisco.quantum.qps.qns03-Mumbai.node1.gc-PS_MarkSweep.total_time_in_ms-collection_time
- cisco.quantum.qps.qns03-Mumbai.node1.gc-PS_Scavenge.total_time_in_ms-collection_time
- cisco.quantum.qps.qns04-Mumbai.node1.gc-PS_MarkSweep.total_time_in_ms-collection_time
- cisco.quantum.qps.qns04-Mumbai.node1.gc-PS_Scavenge.total_time_in_ms-collection_time

A cisco quantum qps qns* node* gc-* invocations

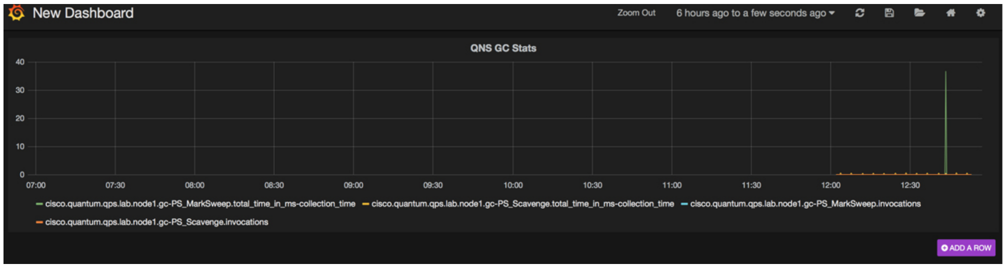
B cisco quantum qps qns* node* gc-* total_time_in_ms-collection_time

Cache timeout 60 Max data points auto

shorter legend names | series as parameters | stacking | templating | max data points

An example statistics graph is shown below.

Figure 32: Example Graph



- cisco.quantum.qps.lab.node1.gc-PS_MarkSweep.total_time_in_ms-collection_time
- cisco.quantum.qps.lab.node1.gc-PS_Scavenge.total_time_in_ms-collection_time
- cisco.quantum.qps.lab.node1.gc-PS_MarkSweep.invocations
- cisco.quantum.qps.lab.node1.gc-PS_Scavenge.invocations

Step 3 Save the dashboard by clicking on Save icon.

Export and Import Dashboards

Existing dashboard templates can be exported and imported between environments. This is useful for sharing Grafana dashboards with others.

Export Dashboard

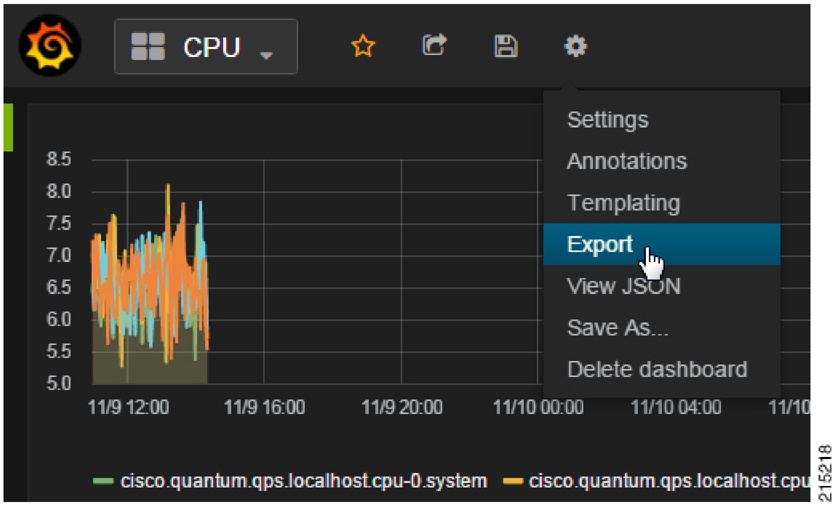
This topic describes how to export a dashboard configuration to a file.

Step 1 Sign-in as a Grafana Administrative User.

Step 2 Open the dashboard to be exported.

Step 3 Click the gear icon at the top of the page, and then select **Export** to save the dashboard configuration on your local system.

Figure 33: Export



Step 4 If prompted, select the location on your local system to save the dashboard template, and click **OK**.

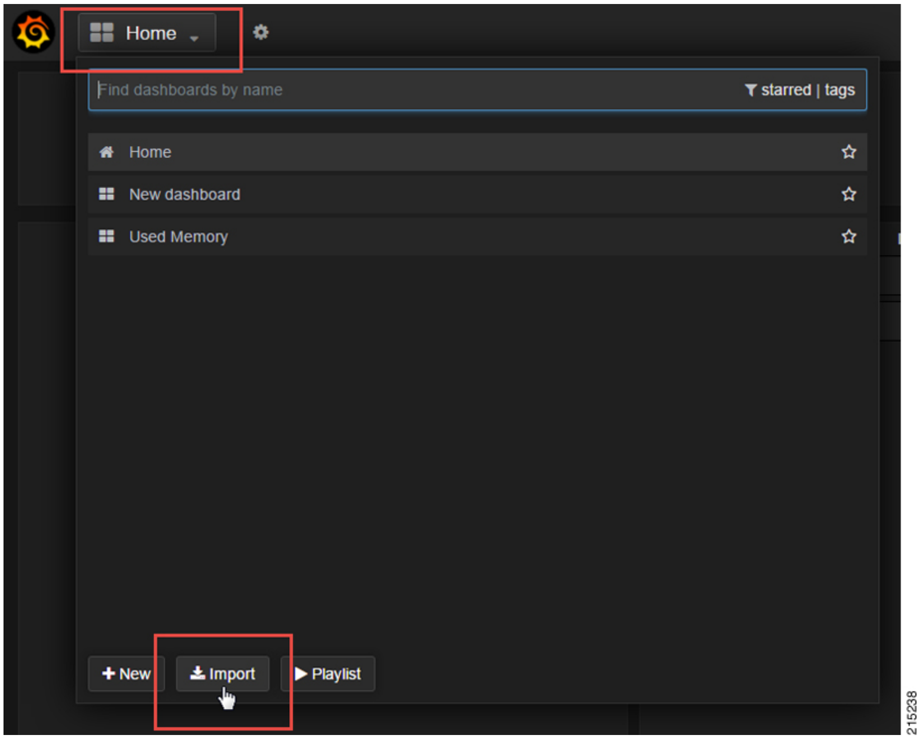
Import Dashboard

This topic describes how to import a dashboard from a file.

Step 1 Sign-in as a Grafana Administrative User.

Step 2 Click **Home** at the top of the Grafana window, and then click **Import** as shown below.

Figure 34: Import



Step 3 Click **Choose File**.

Step 4 Select the file on your local system to save the dashboard template and click **Open**.

Step 5 After the dashboard is loaded, click the disk icon (Save dashboard) at the top of the screen to save the dashboard.

Note Your changes to this dashboard are lost if you do not save the dashboard.

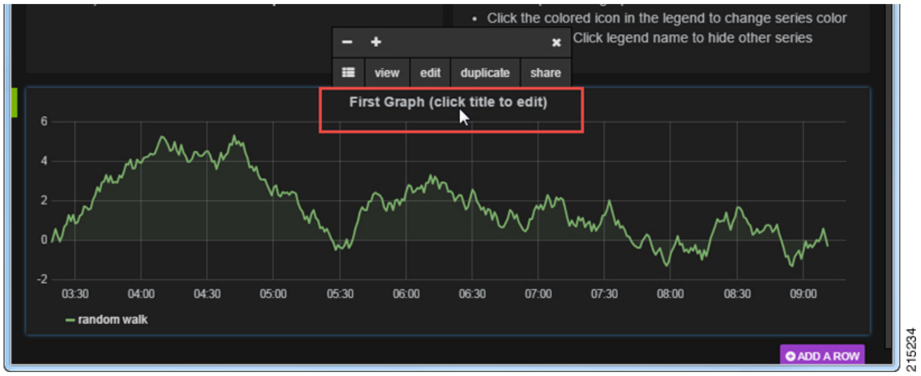
The data to be imported in the dashboard should be in the correct format. Grafana does not throw any error if incorrectly formatted data is loaded.

Export Graph Data to CSV

This topic describes how to export the data in a graph panel to a CSV file.

Step 1 Click the title of the graph as shown below to open the graph controls.

Figure 35: Title

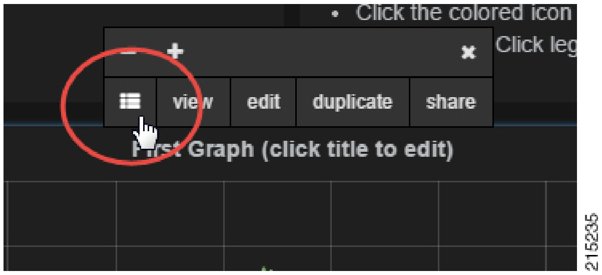


Click the colored icon in the legend to change series color

Click legend name to hide other series

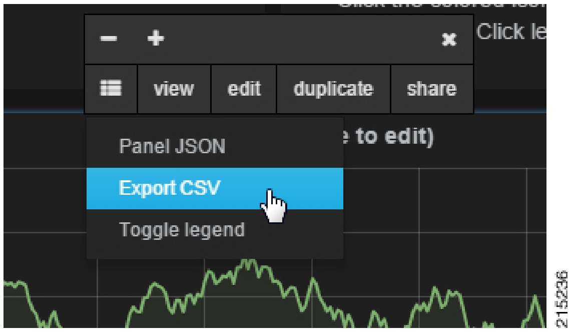
Step 2 Click the rows button to open another menu.

Figure 36: Rows



Step 3 Click **Export CSV**.

Figure 37: Export



A `grafana_data_export.csv` file is downloaded by your browser.

Session Consumption Report

Introduction

This feature generates the session consumption report and stores the data into a separate log. The total number of sessions limited by the license, the total number of active sessions, and total transactions per second are documented at regular time intervals into the log. The core license number is derived from the license file that has the total number of sessions limited by the license. The active session count and the transaction count has been taken from Grafana using the graphite query. A single entity of the feature mainly prints the current time stamp with the statistics values.

Data Collection

The session and TPS count is collected from the graphite API with a JSON response. The JSON response is then parsed to get the counter, which is then logged into the consolidated log. The sample URL and the JSON response are given below:

```
> curl
-u graphite_default:$graphite_default_passwd -G "http://localhost/graphite/render?
target=cisco.quantum.qps.pcrfclient01.set_session_count_total.records&from=-20second&until=-0hour&format=json"
> [{"target":
"cisco.quantum.qps.localhost.set_session_count_total.records", "datapoints": [[3735.42,
1455148210], [3748.0, 1455148220]]}]
> curl
-u graphite_default:$graphite_default_passwd -G "http://localhost/graphite/render?
target=sumSeries(cisco.quantum.*.*.node*.messages.e2e*.success)&from=-20second&until=-0hour&format=json"
> [{"target":
"sumSeries(cisco.quantum.*.*.node*.messages.e2e*.success)", "datapoints": [[2345.34324,
1455148210], [2453.23445453,
1455148220]]}]
```

Logging

Data logging is done using the logback mechanism. The consolidated data that is generated is stored in a separate log file named `consolidated-sessions.log` inside the `/var/log/broadhop` directory along with other logs. The data entries are appended to the log every 90 seconds. The logs generated are detailed and have the counter name and the current value with the time stamp.

Performance

The codebase pulls the JSON response from the Graphite API. The overhead by the codebase adds an average of 350 ms of time.

Log Rotation

A log rotation policy is applied on the logs generated for the session Consumption Report. The file size limitation for each log file is 100 MB. The limitation on number of log files is 5. The logs get rotated after reaching the limitations. One file contains a little more than two years of data, so five such files can contain 10 years of data until the first file get replaced.

Sample Report

```
2016-02-15 20:30:01 - TPS_COUNT: 6440.497603         SESSION_COUNT: 200033.0
LICENSE_COUNT: 10000000
2016-02-15 20:31:31 - TPS_COUNT: 6428.235699999999   SESSION_COUNT: 201814.0
LICENSE_COUNT: 10000000
2016-02-15 20:33:01 - TPS_COUNT: 5838.386624000001   SESSION_COUNT: 204818.0
LICENSE_COUNT: 10000000
2016-02-15 20:34:31 - TPS_COUNT: 6266.777699999999   SESSION_COUNT: 208719.0
LICENSE_COUNT: 10000000
2016-02-15 20:36:01 - TPS_COUNT: 6001.863687         SESSION_COUNT: 211663.0
LICENSE_COUNT: 10000000
2016-02-15 20:37:31 - TPS_COUNT: 6528.9450540000025  SESSION_COUNT: 213976.0
LICENSE_COUNT: 10000000
2016-02-15 20:39:01 - TPS_COUNT: 6384.073428         SESSION_COUNT: 218851.0
LICENSE_COUNT: 10000000
2016-02-15 20:40:31 - TPS_COUNT: 6376.373494000002   SESSION_COUNT: 220515.0
LICENSE_COUNT: 10000000
2016-02-15 20:42:01 - TPS_COUNT: 6376.063389999998   SESSION_COUNT: 222308.0
LICENSE_COUNT: 10000000
2016-02-15 20:43:31 - TPS_COUNT: 6419.310694000001   SESSION_COUNT: 223146.0
LICENSE_COUNT: 10000000
2016-02-15 20:45:01 - TPS_COUNT: 6455.804928         SESSION_COUNT: 222546.0
LICENSE_COUNT: 10000000
2016-02-15 20:46:31 - TPS_COUNT: 6200.357029999999   SESSION_COUNT: 223786.0
LICENSE_COUNT: 10000000
2016-02-15 20:48:02 - TPS_COUNT: 6299.090987         SESSION_COUNT: 223973.0
LICENSE_COUNT: 10000000
2016-02-15 20:49:31 - TPS_COUNT: 6294.876452         SESSION_COUNT: 226629.0
LICENSE_COUNT: 10000000
2016-02-15 20:51:01 - TPS_COUNT: 6090.202965999999   SESSION_COUNT: 227581.0
LICENSE_COUNT: 10000000
2016-02-15 20:52:31 - TPS_COUNT: 6523.586347999997   SESSION_COUNT: 228450.0
LICENSE_COUNT: 10000000
2016-02-15 20:54:01 - TPS_COUNT: 5842.613997000001   SESSION_COUNT: 229334.0
LICENSE_COUNT: 10000000
2016-02-15 20:55:31 - TPS_COUNT: 6638.526543         SESSION_COUNT: 232683.0
LICENSE_COUNT: 10000000
2016-02-15 20:57:01 - TPS_COUNT: 6073.7797439999995  SESSION_COUNT: 230466.0
LICENSE_COUNT: 10000000
2016-02-15 20:58:31 - TPS_COUNT: 6354.272679999999   SESSION_COUNT: 234070.0
LICENSE_COUNT: 10000000
2016-02-15 21:00:03 - TPS_COUNT: 6217.872034999999   SESSION_COUNT: 236139.0
LICENSE_COUNT: 10000000
```

Resync Member of a Replica Set

This procedure can be performed if any one of the members is not in a healthy state. Un-healthy state means that the replica-member is in RECOVERING or other unhealthy state for a longer duration and unable to recover on its own. Additionally, this procedure can be performed to reclaim the disk-space in order to reduce the database fragmentation.

Note Make sure PRIMARY member is available while performing this procedure.

Step 1 Verify the status of primary and secondary member by running the following command:

```
diagnostics.sh --get_replica_status
```

Step 2 Stop Mongo AIDO client by running the following command on sessionmgr VM:

```
monit stop aido_client
```

Step 3 Stop Mongo server from sessionmgr VM by running the following command (portNum is the port number of fragmented member):

```
/etc/init.d/sessionmgr-<portNum> stop
```

Step 4 Clean database directory by removing data directory from path mentioned against `--dbpath` attribute of mongo command. The value can be retrieved by running the following command (using the portNum of the fragmented member):

```
grep -w DBPATH= /etc/init.d/sessionmgr-<portNum>
```

Step 5 Start Mongo AIDO client from Sessionmgr VM by running the following command:

```
/etc/init.d/sessionmgr-<portNum> start
```

Step 6 Start AIDO client on sessionmgr VM by running the following command:

```
monit start aido_client
```