Project 7 - Metrics Collection System

Goal:

The goal of this project is to create a metrics collection and graphing system. This will allow you to visually see what the system utilization is for a given host or across an entire environment.

Instructions:

Create a Virtual Machine

First, start a command line session on your local machine. Next, move into the working folder you created for this course.

```
cd linuxclass
```

Initialize the vagrant project using the usual process of creating a directory, changing into that directory, and running "vagrant init". We'll name this vagrant project "grafana".

```
mkdir grafana
cd grafana
vagrant init jasonc/centos8
```

Configure the Virtual Machine

Edit the Vagrantfile and set the hostname of the virtual machine to "grafana". Also, assign the IP address of 10.23.45.40 to the machine.

```
config.vm.hostname = "grafana"
config.vm.network "private_network", ip: "10.23.45.40"
```

Start the Virtual Machine

Now you're ready to start the VM and connect to it.

```
vagrant up
vagrant ssh
```

Install InfluxDB

We'll use InfluxDB to store the metrics collected from our systems. Install InfluxDB using an RPM package.

```
sudo dnf install -y http://mirror.linuxtrainingacademy.com/grafana/influxdb-1.8.2.x86_64.rpm
```

Internet download location:

https://dl.influxdata.com/influxdb/releases/influxdb-1.8.2.x86_64.rpm

Now that InfluxDB is installed, start and enable it.

```
sudo systemctl start influxdb
sudo systemctl enable influxdb
```

Install Telegraf

Next install Telegraf on the host and start feeding some data into InfluxDB. We'll use an RPM to install Telegraf.

```
sudo dnf install -y http://mirror.linuxtrainingacademy.com/grafana/telegraf-1.15.3-1.x86_64.rpm
```

Internet download location:

https://dl.influxdata.com/telegraf/releases/telegraf-1.15.3-1.x86_64.rpm

Even though Telegraf ships with a few input plugins enabled by default, let's enable even more now.

First, open the Telegraf configuration file for editing.

```
sudo nano /etc/telegraf/telegraf.conf
```

Uncomment the following lines in the file by removing the pound sign (#) from the beginning of each of the lines. NOTE: These lines are not next to each other in the file. You'll have to search for each of these lines. If you are using the nano editor, you can type Ctrl-W to initiate a search, type the text you are looking for, and then hit <ENTER> to jump to the next instance of that text in the file.

```
# [[inputs.conntrack]]
# [[inputs.internal]]
# [[inputs.interrupts]]
# [[inputs.linux_sysctl_fs]]
# [[inputs.net]]
# [[inputs.netstat]]
# [[inputs.nstat]]
```

Once they have been uncommented, they will look like this:

```
[[inputs.conntrack]]
[[inputs.internal]]
[[inputs.interrupts]]
[[inputs.linux_sysctl_fs]]
[[inputs.net]]
[[inputs.netstat]]
[[inputs.nstat]]
```

Now we're ready to start and enable the Telegraf service.

```
sudo systemctl start telegraf
sudo systemctl enable telegraf
```

Let's make sure that data from Telegraf is being stored in InfluxDB. To do that, we'll use the InfluxDB command line tool, `influx`.

```
$ influx
> show databases
```

You should see a database named telegraf in the list of databases. If not, make sure the telegraf service is running (`sudo systemctl restart telegraf`) and check the system log for any error messages (`sudo cat /var/log/messages`).

Now let's look at the types of metrics being stored in the telegraf database.

```
> use telegraf
> show measurements
```

You should see a list of the measurements Telegraf is collecting and writing to the database such as cpu, disk, and so on. If you are familiar with traditional SQL databases, you can think of a measurement as a table.

To retrieve data from a measurement, use the `select` statement. The following is an InfluxQL statement which is very similar to a SQL statement. The command displays the data from all the fields in the swap measurement that occurred within the previous minute. You can think of a field as a column in a traditional SQL database. Run the following command.

```
> select * from swap where time > now() - 1m
```

The output displayed to your screen is all the data that is available about the swap measurement from the last minute.

Exit the influx client:

```
> exit
```

Troubleshooting Tip

If there is no output from the select statement, that means no recent data has been received and stored. In that case, be sure to check that Telegraf service is running. Exit out of the influx client by typing `exit` and hitting `<ENTER>`. That will place you back at the normal Linux command line prompt. Next, execute `sudo systemctl start telegraf`. If Telegraf doesn't start, check the `/var/log/messages` file with `sudo grep "grafana telegraf" /var/log/messages`.

Install Grafana

Let's install Grafana, so we can create dashboards and graphs using InfluxDB as our source.

```
sudo dnf install -y http://mirror.linuxtrainingacademy.com/grafana/grafana-7.2.0-1.x86_64.rpm
```

Internet download location:
https://dl.grafana.com/oss/release/grafana-7.2.0-1.x86_64.rpm

Now you're ready to start and enable the grafana-server service.

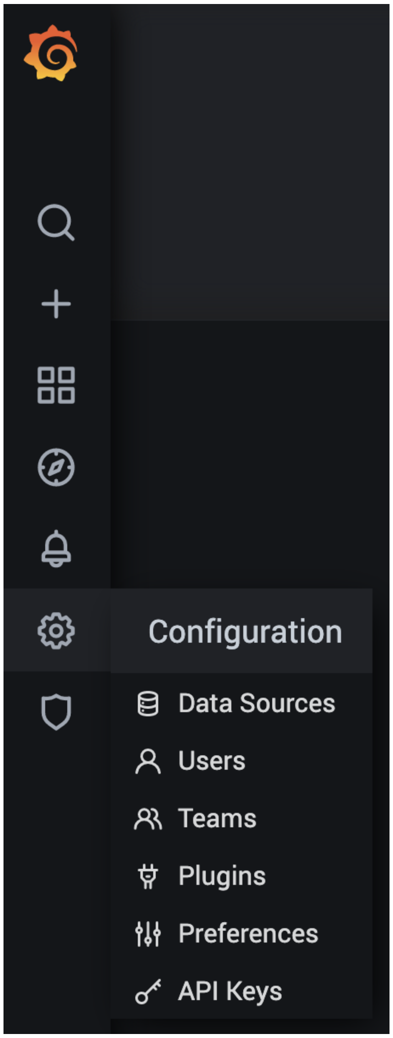
```
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

Open a web browser and connect to http://10.23.45.40:3000. Log in with the username of "admin" and the password of "admin."

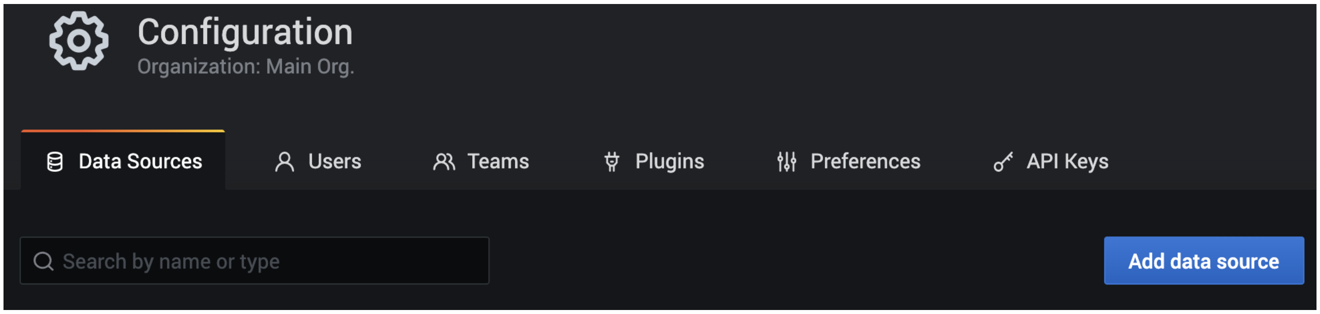
You'll be prompted to change your password, but simply click on "Skip" for now.

Create a Data Source

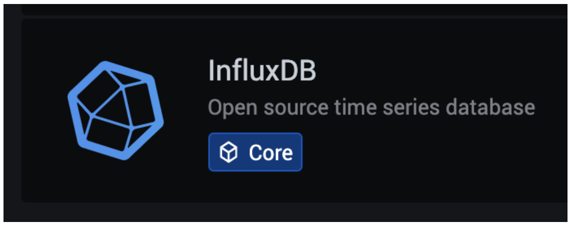
Click on the gear icon in the menu bar on the left. It will take you to the Configuration page for Grafana.



Click on the "Add data sources" button.



Click on "InfuxDB".



Now fill out the form. You will leave most fields at their default value.

Name: InfluxDB-Telegraf
Default: Click to turn the selector on.
Query Language: InfluxQL

URL: http://localhost:8086
Access: Server (default)

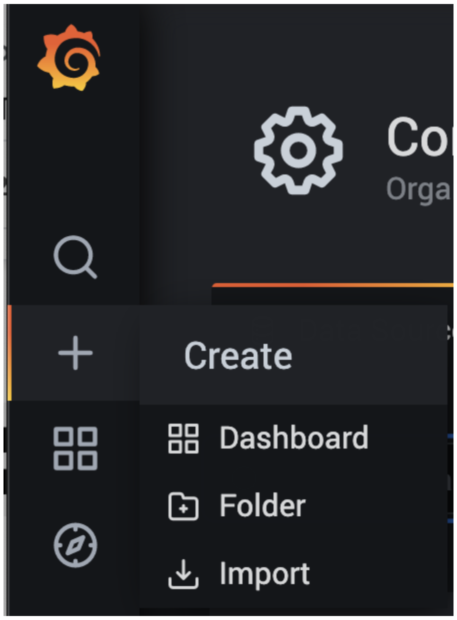
Database: telegraf

Click the "Save & Test" button.

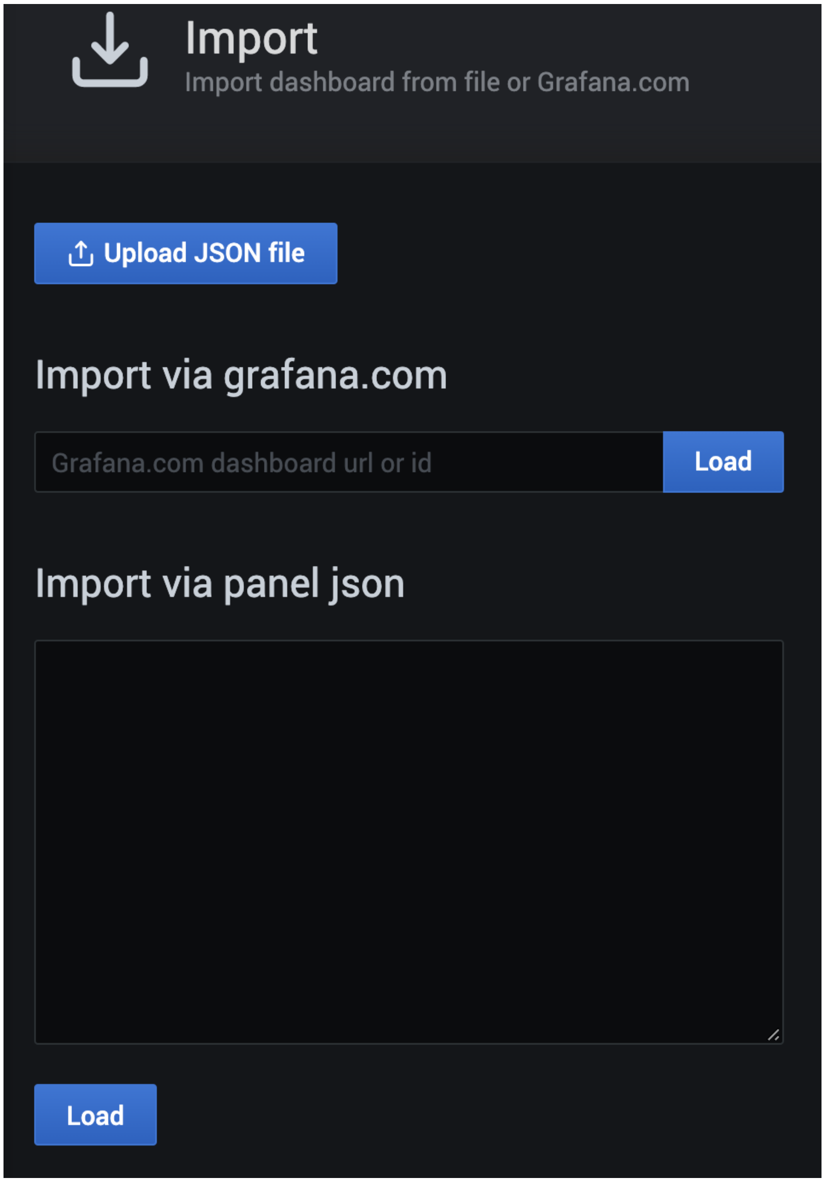
Now Grafana can access the metrics stored in InfluxDB.

Import a Dashboard

Hover over the plus sign in the menu on the left of your screen. It will expand into a menu when you hover over it. From there, click on "Import".

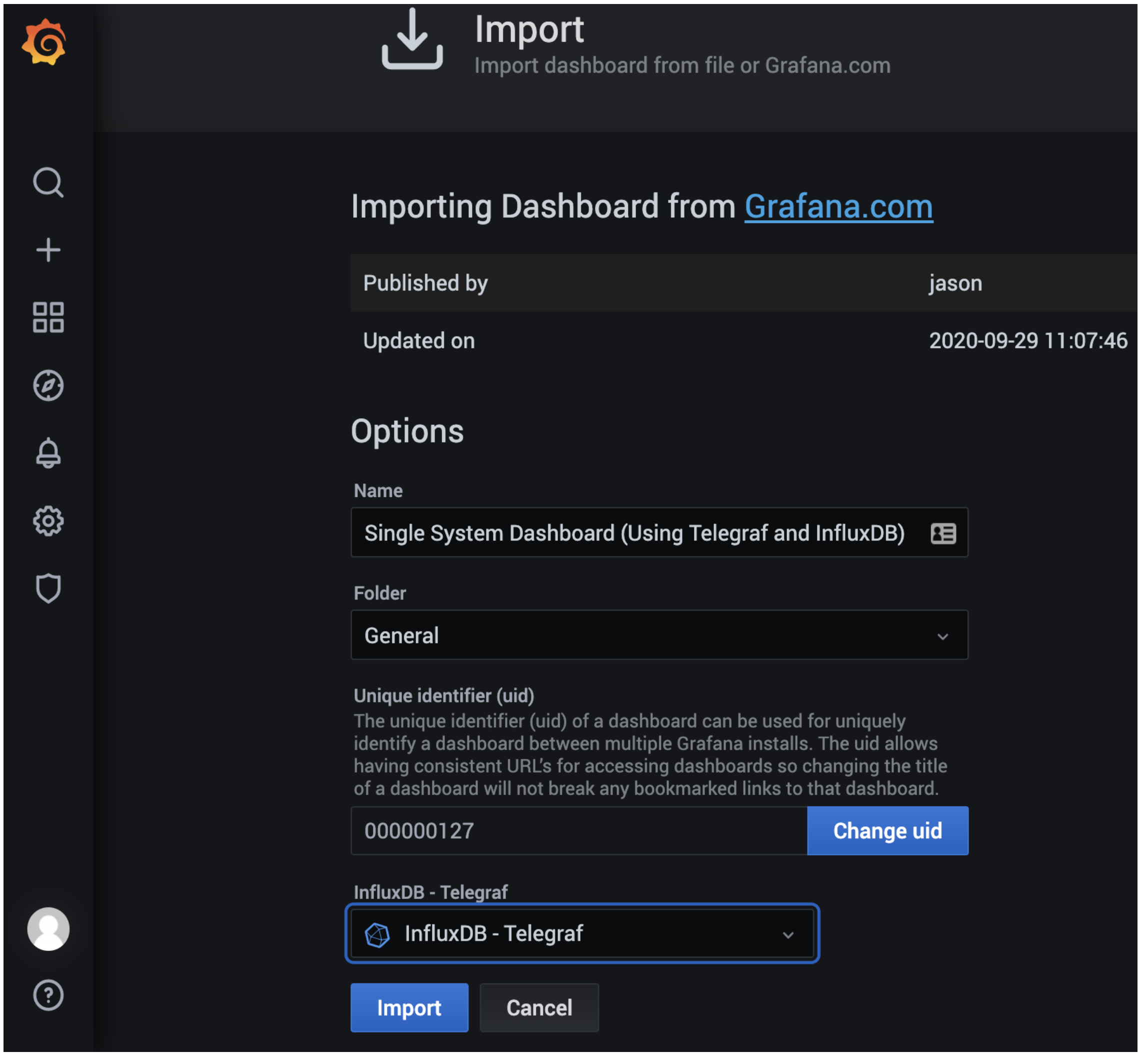


In the "Grafana dashboard URL or id" field, enter "13095" and click the "Load" button next to it.



NOTE: If for some reason this dashboard is not available, you can download the JSON dashboard from this URL: http://mirror.linuxtrainingacademy.com/grafana/13095.json You would then click "Upload JSON File" and navigate to where you saved the 13095.json file on your local computer.

On the next screen in the "Select an InfluxDB data source" box, select "InfluxDB-Telegraf". Then click on the "Import" button.



Importing Dashboard from Grafana.com

Published by	jason
Updated on	2020-09-29 11:07:46

Options

Name
Single System Dashboard (Using Telegraf and InfluxDB)

Folder
General

Unique identifier (uid)
The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

000000127

InfluxDB - Telegraf
InfluxDB - Telegraf

Now you will see a dashboard displaying system performance information for the grafana host.

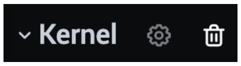
Troubleshooting Tip

You may notice that some graphs are blank and may even report "No data". Some blank graphs may be normal. For example, if the "TCP aborts" graph reports "No data" then that means there were no TCP aborts that have occurred in the selected time range so no such events were recorded and saved to the database. That is the reason there is no data to graph. It's actually a good thing that that particular graph is blank. This is an ideal situation and is quite common.

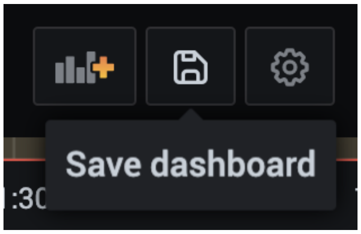
However, if the "Interrupts" graph is blank, that likely means the corresponding Telegraf input plugin has not been enabled for that measurement. Make sure the "`[[inputs.interrupts]]`" line in the `/etc/telegraf/telegraf.conf` file is uncommented and then restart the Telegraf service with `sudo systemctl restart telegraf`. After a couple of minutes, data should start to appear on the graph. If there are many blank graphs, make sure you've enabled the input plugins listed above from earlier in this project.

Customize the Imported Dashboard

Let's say that you're not interested in the graphs displayed in the "Kernel" section. Move your mouse over the word "Kernel". When your mouse is hovering over "Kernel," you will see two icons appear. One is a gear which represents the settings for that row of graphs and the other is a trash can. Click the trash can (or garbage bin) icon. Next, click the "Yes" button to confirm the deletion of the row and all the graphs in that row.



Click the save icon at the top of your screen. The icon looks like a 3.5" floppy disk if you happen to be old enough to remember such things. You'll be presented with a "Save dashboard" dialog box. Click the "Save" button to save your changes.



Customizing existing dashboards is the quickest way to create your own visualizations. You can find more dashboards on Grafana's website: https://grafana.com/grafana/dashboards Be aware that the available dashboards are of varying quality. Also, you have to choose a dashboard that uses the same data source type as you are using. So, if you are going to experiment with other publicly available dashboards, make sure to select ones that use InfluxDB as their data source.

Install Telegraf on Another Host

Let's bring up another system, configure Telegraf on it, and have it send its metrics to InfluxDB as well. Then we'll go back to Grafana and watch our dashboard update with data from this host.

Open up a new command line session. Make sure the kanboard virtual machine is running and connect to it.

```
cd linuxclass
cd kanboard
vagrant up
vagrant ssh
```

Your prompt should like this:

```
[vagrant@kanboard ~]$
```

Install Telegraf on the kanboard host to start feeding its data into InfluxDB.

```
sudo dnf install -y http://mirror.linuxtrainingacademy.com/grafana/telegraf-1.15.3-1.x86_64.rpm
```

Because InfluxDB is not running on the same host, you need to tell Telegraf where to send its data. Of course, we'll tell Telegraf to send its data to the grafana VM.

First, open the Telegraf configuration file for editing.

```
sudo nano /etc/telegraf/telegraf.conf
```

Add the following line below the "[[outputs.influxdb]]" line in the configuration file:

```
  urls = ["http://10.23.45.40:8086"]
```

It should look like this once you are done editing this section of the configuration file.

```
[[outputs.influxdb]]
  urls = ["http://10.23.45.40:8086"]
```

This tells Telegraf the IP address and port of InfluxDB in our environment.

The dashboard and graphs we are using can display data from many Telegraf plugins. Some of those graphs require data from Telegraf input plugins that are not enabled by default. Let's enable them now.

Uncomment the following lines in the Telegraf configuration file by removing the pound sign (#) from the beginning of each of the lines. NOTE: These lines are not next to each other in the file. You'll have to search for each of these lines. If you are using the nano editor, you can type Ctrl-W to initiate a search, type the text you are looking for, and then hit <ENTER> to jump to the next instance of that text in the file.

```
# [[inputs.conntrack]]
# [[inputs.internal]]
# [[inputs.interrupts]]
# [[inputs.linux_sysctl_fs]]
# [[inputs.net]]
# [[inputs.netstat]]
# [[inputs.nstat]]
```

Once they have been uncommented, they will look like this:

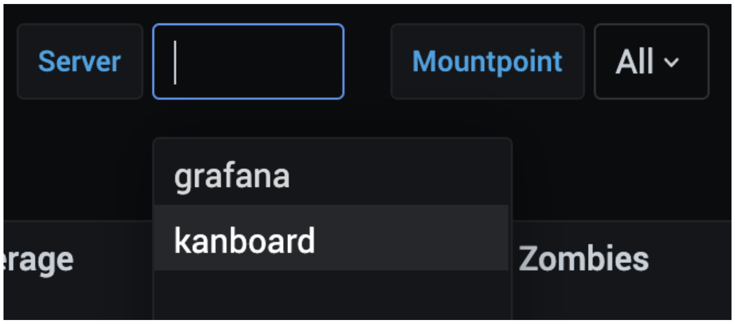
```
[[inputs.conntrack]]
[[inputs.internal]]
[[inputs.interrupts]]
[[inputs.linux_sysctl_fs]]
[[inputs.net]]
[[inputs.netstat]]
[[inputs.nstat]]
```

Now we're ready to start and enable the Telegraf service on the kanboard host.

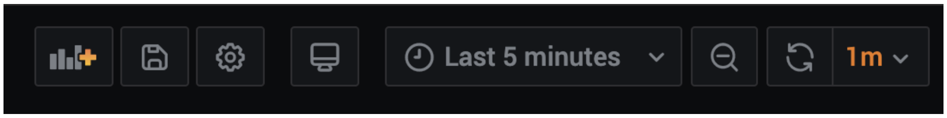
```
sudo systemctl start telegraf
sudo systemctl enable telegraf
```

Return to your web browser and reload the dashboard page by clicking the reload button on your web browser or by typing Ctrl-R for Windows users or CMD-R for mac users.

At the top of your browser you should see a selector box for "Server". When you click on the word "Server" you will see the grafana VM as well as our newly added VM of kanboard. If you don't see "kanboard" as an option, wait a couple of minutes and reload the web page again. If you still don't see the kanboard host, check the `/etc/telegraf/telegraf.conf` file for typing mistakes, paying close attention to the "urls" line. If you needed to make any changes to the configuration file, restart the telegraf service with `sudo systemctl restart telegraf`.



By the way, you can select a time range using the "time picker" in the top right-hand corner to zoom in. If the time window is too large you may not see the data in the graphs. Here I choose to look at the last 5 minutes:



Send Icinga Metrics to InfluxDB

Let's create an InfluxDB database on the grafana VM to store metrics from the Icinga service. **Switch to the command line session associated with the grafana VM**. Make sure you are connected to the grafana VM. Your prompt should like this:

```
[vagrant@grafana ~]$
```

Once you have confirmed you are connected to the grafana host, run the following commands to create an InfluxDB database named icinga2:

```
$ influx
> create database icinga2
> show databases
> exit
```

Now that you have a database where you can store Icinga metrics, tell Icinga to send its metrics there.

Open up a new command session, ensure that the icinga VM is running and connect to it.

```
cd linuxclass
cd icinga
vagrant up
vagrant ssh
```

Make sure you are connected to the icinga VM. Your prompt should like this:

```
[vagrant@icinga ~]$
```

Icinga2 comes with an InfluxDB plugin that allows it to send the metrics it's collecting to InfluxDB. To enable that plugin, run the following command.

```
sudo icinga2 feature enable influxdb
```

Now we need to tell Icinga where to send its metrics. Edit the configuration file for the InfluxDB plugin.

```
sudo nano /etc/icinga2/features-enabled/influxdb.conf
```

Add the following lines below the "`object InfluxdbWriter "influxdb" {`" line in the configuration file:

```
  host = "10.23.45.40"
  database = "icinga2"
  enable_send_metadata = true
```

The configuration file should look like this after you are done editing the file.

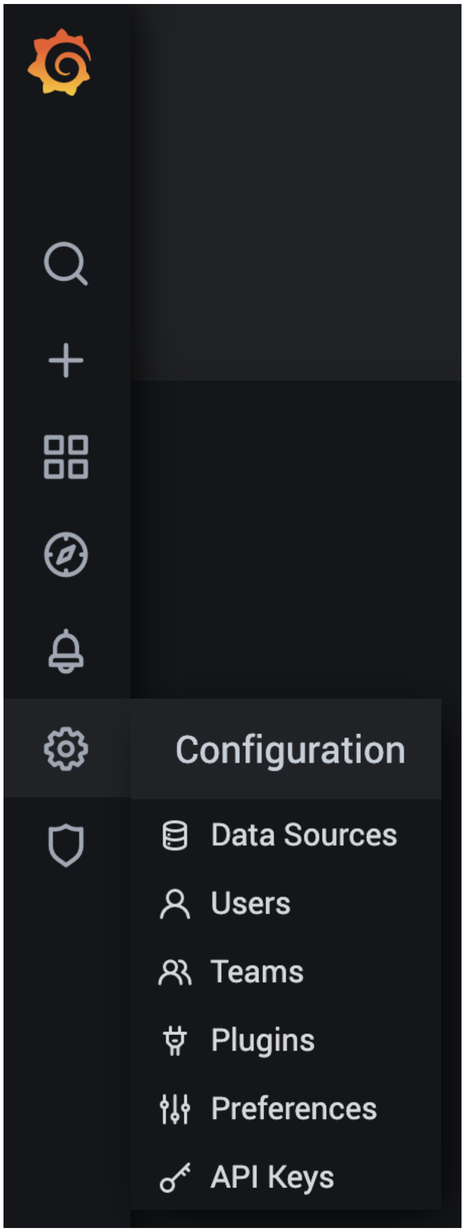
```
/**
 * The InfluxdbWriter type writes check result metrics and
 * performance data to an InfluxDB HTTP API
 */

object InfluxdbWriter "influxdb" {
  host = "10.23.45.40"
  database = "icinga2"
  enable_send_metadata = true
  //host = "127.0.0.1"
  //port = 8086
  //database = "icinga2"
  //flush_threshold = 1024
  //flush_interval = 10s
  //host_template = {
  //  measurement = "$host.check_command$"
  //  tags = {
  //    hostname = "$host.name$"
  //  }
  //}
  //service_template = {
  //  measurement = "$service.check_command$"
  //  tags = {
  //    hostname = "$host.name$"
  //    service = "$service.name$"
  //  }
  //}
}
```

Restart the icinga service to allow it to start sending data to InfluxDB.

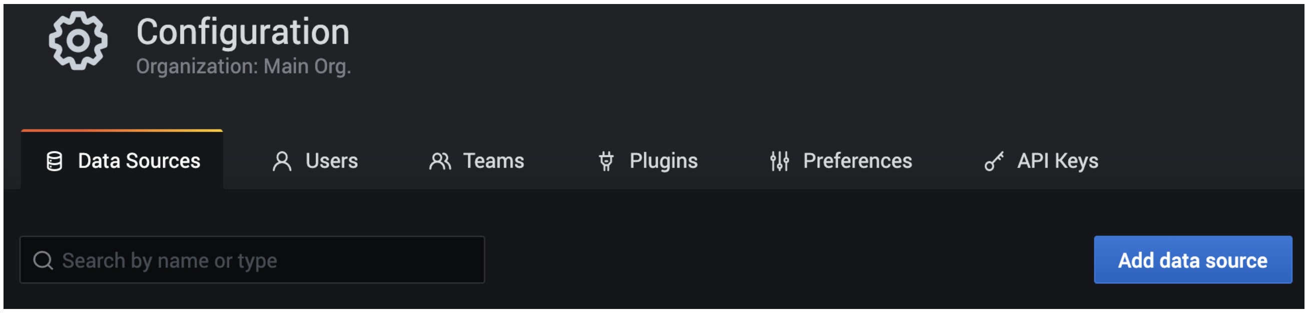
```
sudo systemctl restart icinga2
```

Create a Data Source for the Icinga Data

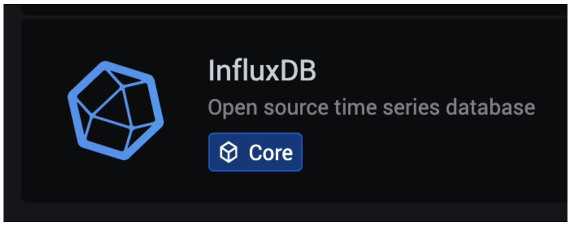
Return to your web browser. Click on the gear icon in the menu bar on the left. It will take you to the Configuration page for Grafana.



Click on the "Add data sources" button.



Click on "InfuxDB".



Now fill out the form. You will leave most fields at their default value.

Name: InfluxDB-Icinga
Query Language: InfluxQL

URL: http://localhost:8086
Access: Server (default)

Database: icinga2

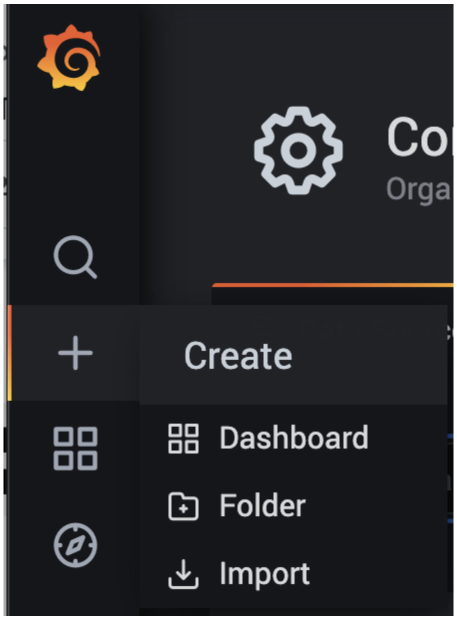
Click the "Save & Test" button.

Now Grafana can access the Icinga metrics stored in InfluxDB.

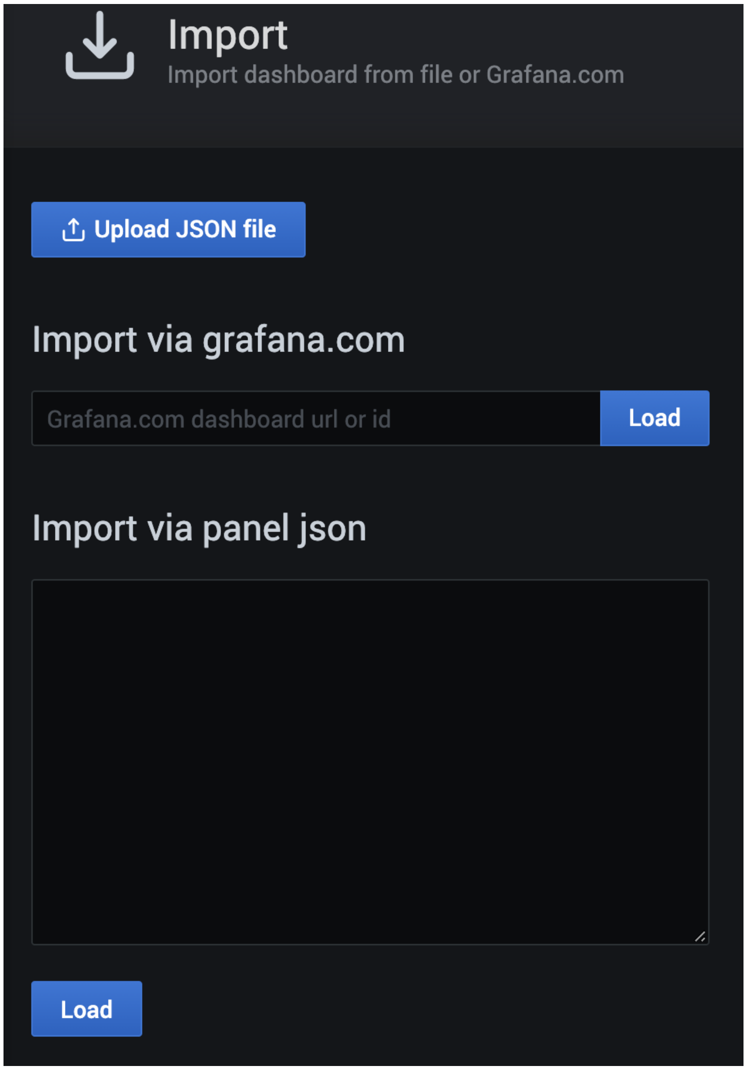
Import a Dashboard for the Icinga Data

Now that we have told Grafana about the Icinga data stored in InfluxDB, it's time to import a new dashboard that can display those metrics.

Hover over the plus sign in the menu on the left of your screen. It will expand into a menu when you hover over it. From there, click on "Import".

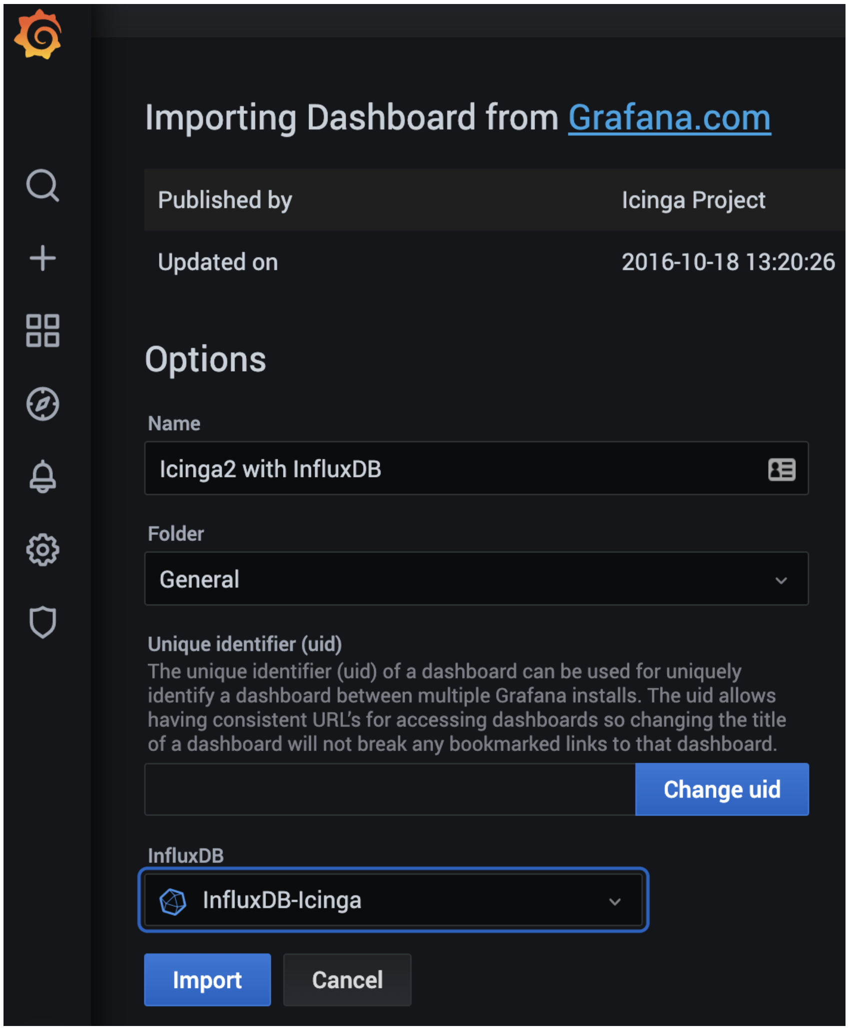


In the "Grafana dashboard URL or id" field, enter "381" and click the "Load" button next to it.



NOTE: If for some reason this dashboard is not available, you can download the JSON dashboard from this URL: http://mirror.linuxtrainingacademy.com/grafana/381.json You would then click "Upload JSON File" and navigate to where you saved the 381.json file on your local computer.

On the next screen in the "Select an InfluxDB data source" box, select "InfluxDB-Icinga". Then click on the "Import" button.



Importing Dashboard from Grafana.com

Published by: Icinga Project

Updated on: 2016-10-18 13:20:26

Options

Name: Icinga2 with InfluxDB

Folder: General

Unique identifier (uid)
The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

InfluxDB: InfluxDB-Icinga

Now you will be presented with a dashboard of the metrics collected by Icinga.

Configuring Telegraf on Icinga

While we're logged into the icinga server, we can configure telegraf on it as well. We're going to repeat the process from before.

Make sure you are connected to the icinga VM. Your prompt should like this:

```
[vagrant@icinga ~]$
```

Install Telegraf on the icinga host to start feeding its data into InfluxDB.

```
sudo dnf install -y http://mirror.linuxtrainingacademy.com/grafana/telegraf-1.15.3-1.x86_64.rpm
```

Because InfluxDB is not running on the same host, you need to tell Telegraf where to send its data. Of course, we'll tell Telegraf to send its data to the grafana VM.

First, open the Telegraf configuration file for editing.

```
sudo nano /etc/telegraf/telegraf.conf
```

Add the following line below the "[[outputs.influxdb]]" line in the configuration file:

```
  urls = ["http://10.23.45.40:8086"]
```

It should look like this once you are done editing this section of the configuration file.

```
[[outputs.influxdb]]
  urls = ["http://10.23.45.40:8086"]
```

This tells Telegraf the IP address and port of InfluxDB in our environment.

The dashboard and graphs we are using can display data from many Telegraf plugins. Some of those graphs require data from Telegraf input plugins that are not enabled by default. Let's enable them now.

Uncomment the following lines in the Telegraf configuration file by removing the pound sign (#) from the beginning of each of the lines. NOTE: These lines are not next to each other in the file. You'll have to search for each of these lines. If you are using the nano editor, you can type Ctrl-W to initiate a search, type the text you are looking for, and then hit <ENTER> to jump to the next instance of that text in the file.

```
# [[inputs.conntrack]]
# [[inputs.internal]]
# [[inputs.interrupts]]
# [[inputs.linux_sysctl_fs]]
# [[inputs.net]]
# [[inputs.netstat]]
# [[inputs.nstat]]
```

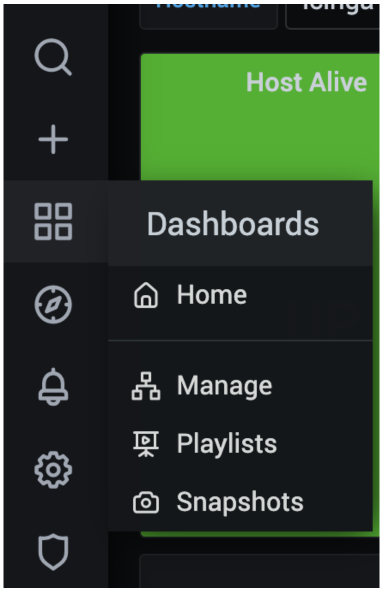
Once they have been uncommented, they will look like this:

```
[[inputs.conntrack]]
[[inputs.internal]]
[[inputs.interrupts]]
[[inputs.linux_sysctl_fs]]
[[inputs.net]]
[[inputs.netstat]]
[[inputs.nstat]]
```

Now we're ready to start and enable the Telegraf service on the icinga host.

```
sudo systemctl start telegraf
sudo systemctl enable telegraf
```

Return to your web browser. Click the dashboard icon in the left-hand menu.

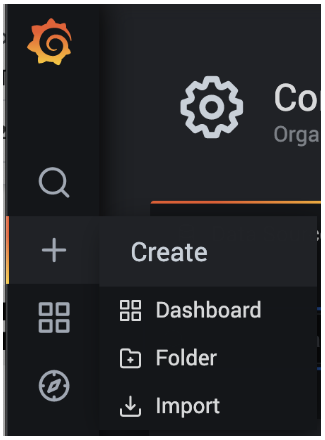


Click on the "Single System Dashboard (Using Telegraf and InfluxDB)" dashboard.

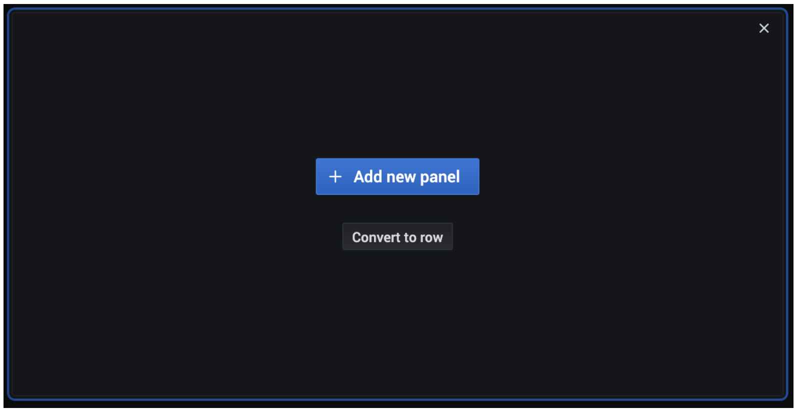
At the top of your browser you should see a selector box for "Server". When you click on the word "Server" you will see the grafana VM, the kanboard VM, and our newly added icinga VM. If you don't see "icinga" as an option, wait a couple of minutes and reload the web page again. If you still don't see the icinga host, check the `/etc/telegraf/telegraf.conf` file for typing mistakes, paying close attention to the "urls" line. If you needed to make any changes to the configuration file, restart the telegraf service with `sudo systemctl restart telegraf`.

Create Your Own Dashboard

Let's create a very simple dashboard from scratch, just so you know how the process works. First, click the plus "+" icon in the left-hand menu.



Next, click the "Add new panel" button.

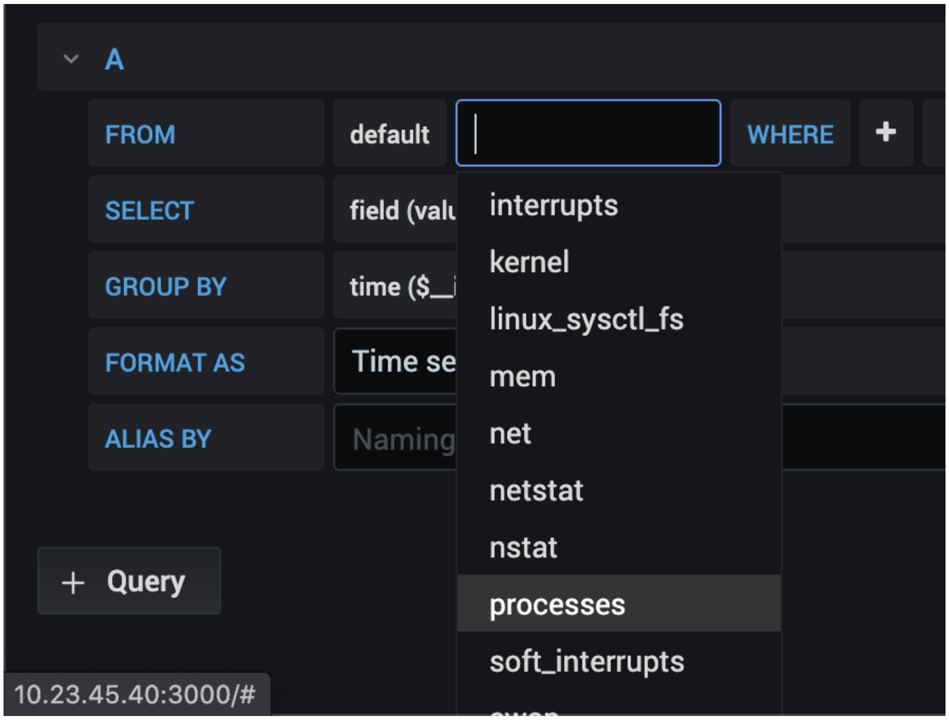


You'll be taken to the "Edit Panel" screen. We're going to use the Query inspector at the bottom of the page to create a graph.

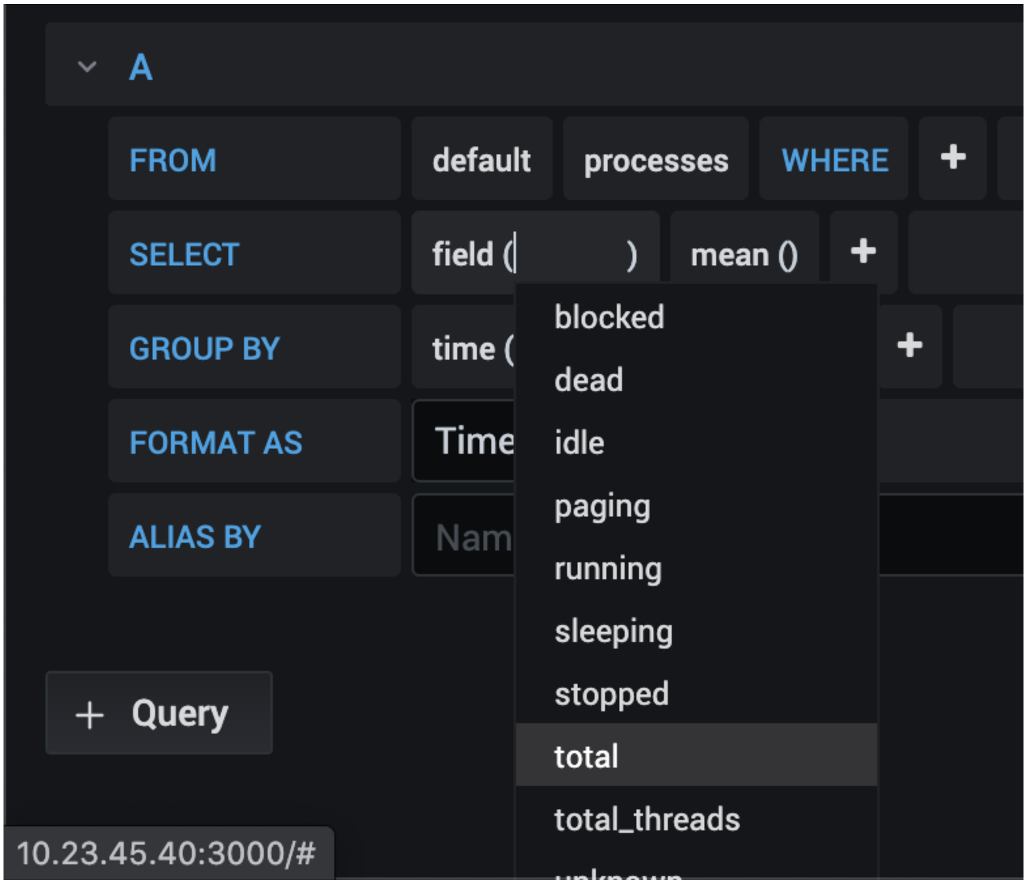
Let's create a graph that will show how many processes are running on each of our systems in our environment.

In the "FROM" line click "select measurement", scroll down the list of measurements, and click on "processes".

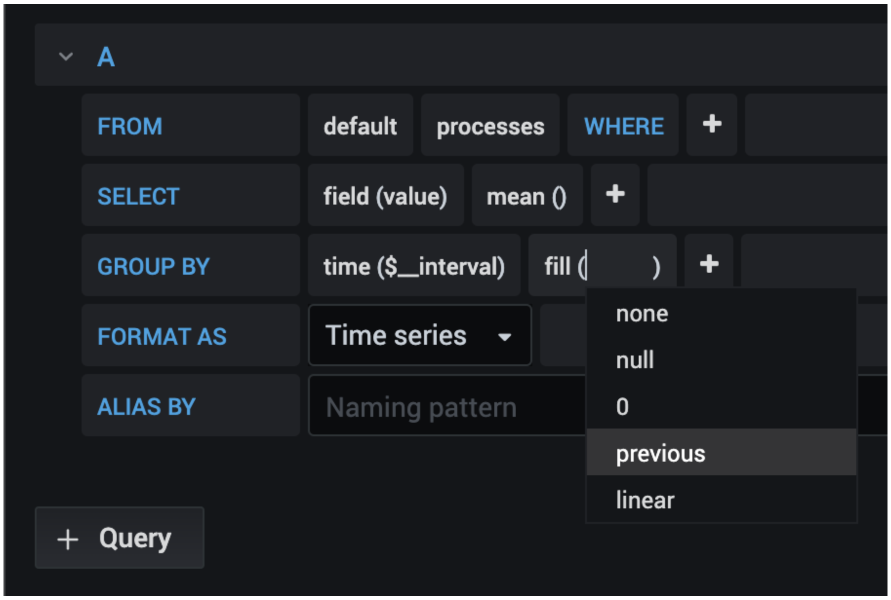
NOTE: If you do not see a "FROM" line, then click on "default" and select "InfluxDB-Telegraf" in the data source drop-down menu.



On the "SELECT" line click on "field(value)" and click on "total" in the fields drop-down.



On the "GROUP BY" line select "fill(null)" and change null to "previous".



A

FROM default processes WHERE +

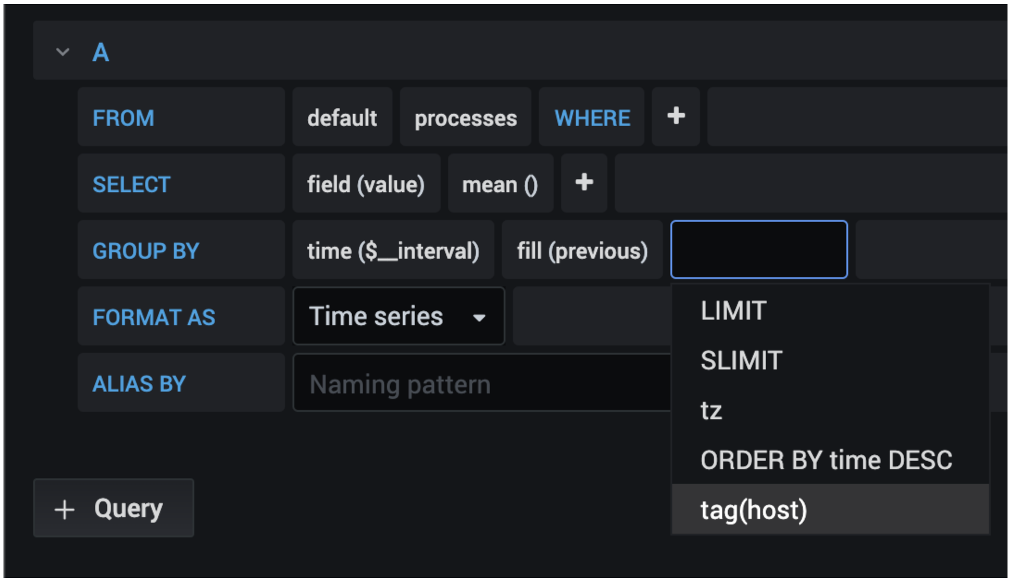
SELECT field (value) mean () +

GROUP BY time ($__interval) fill ( ) +

FORMAT AS Time series

ALIAS BY Naming pattern

Alos, on the "GROUP BY" line click on the plus sign "+" and select "tag(host).



A

FROM default processes WHERE +

SELECT field (value) mean () +

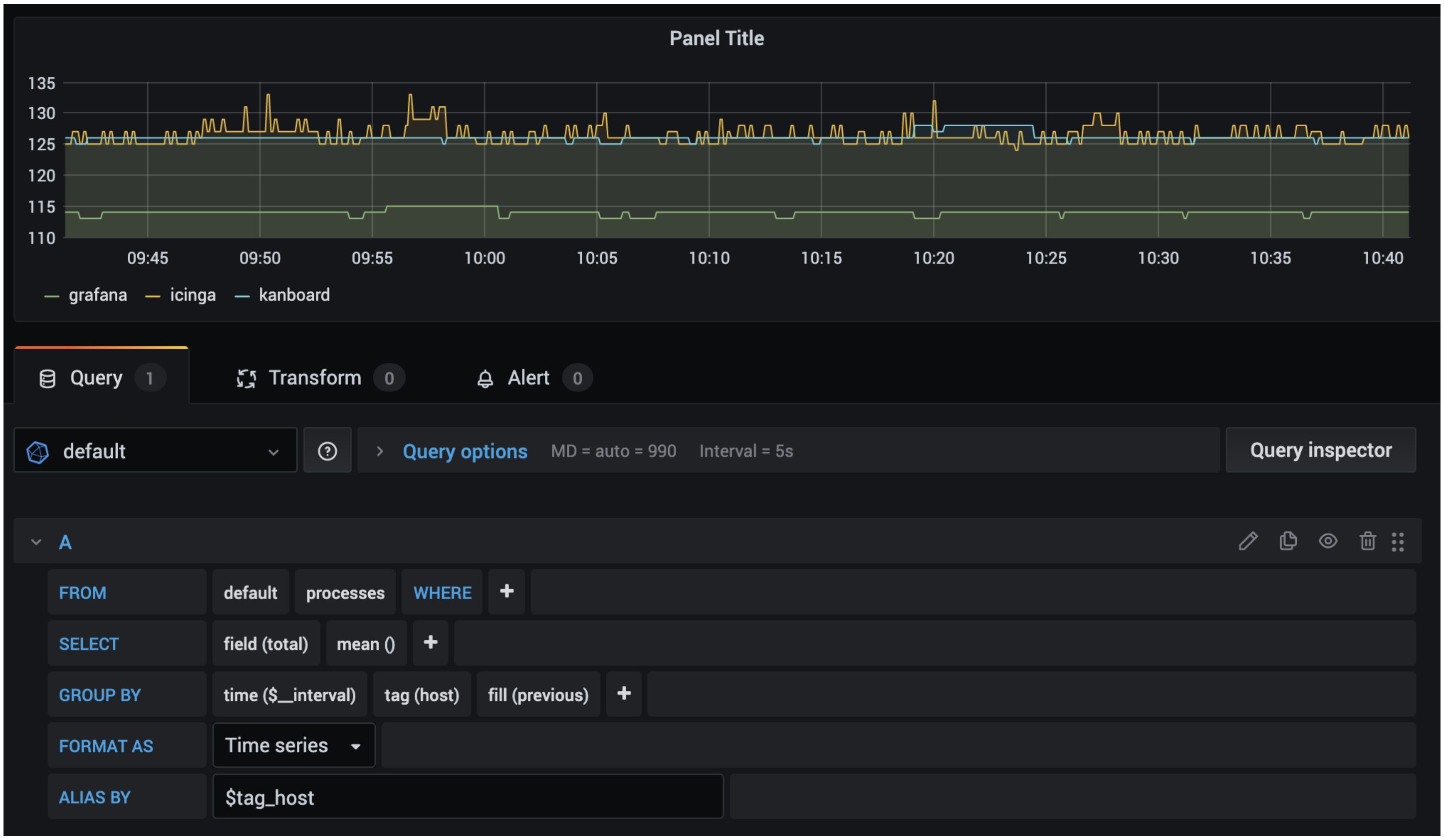
GROUP BY time ($__interval) fill (previous)

FORMAT AS Time series

ALIAS BY Naming pattern

Finally, on the "ALIAS BY" line, type "$tag_host". Click your mouse outside the "ALIAS BY" line to make the change active.

At this point, the query and graph will look something like this:



MD = auto = 990 Interval = 5s

FROM default processes WHERE +

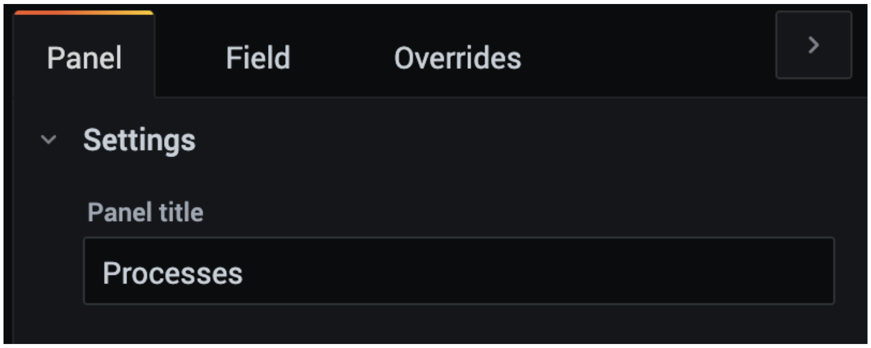
SELECT field (total) mean () +

GROUP BY time ($__interval) tag (host) fill (previous) +

FORMAT AS Time series

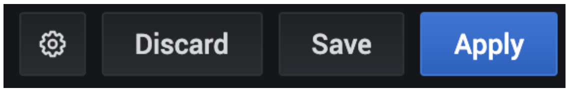
ALIAS BY $tag_host

In the right-hand side of the screen under "Panel" change the "Panel title" to "Processes".

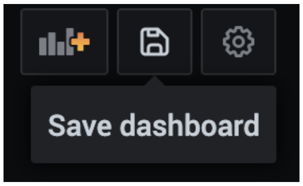


Panel title: Processes

In the top-right hand corner, click the "Apply" button to save our graph.

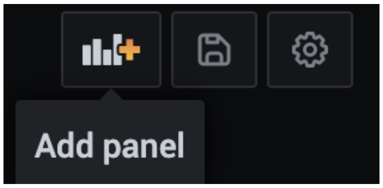


Now you will be returned to the main dashboard page. Click on the "Save Dashboard" icon in the top right of your screen.



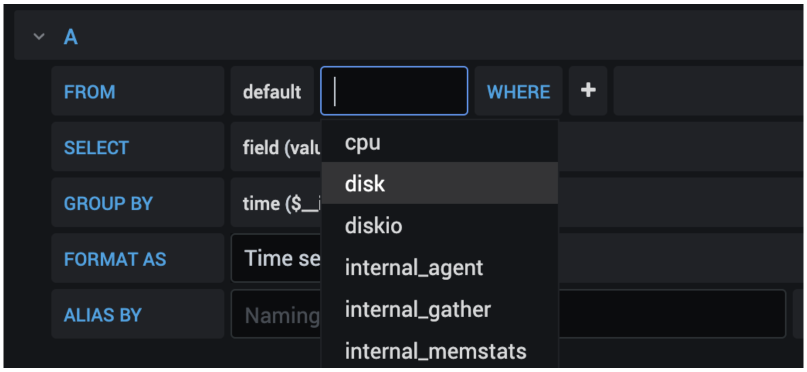
Give your dashboard a name. For example, you could use "My Dashboard". Click the "Save" button.

Let's add one more graph that displays how much free disk space is available in the root (/) file system for each of our hosts. Click Add Panel icon.

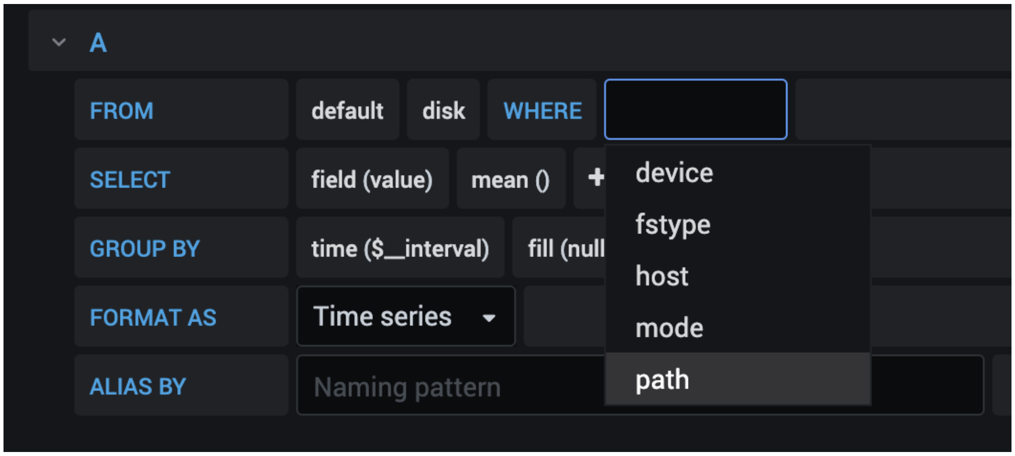


Click the "Add new panel" button that appears.

On the "FROM" line select the "disk" measurement.



Also on the "FROM" line click the plus sign "+" and select "path".



A

FROM default disk WHERE

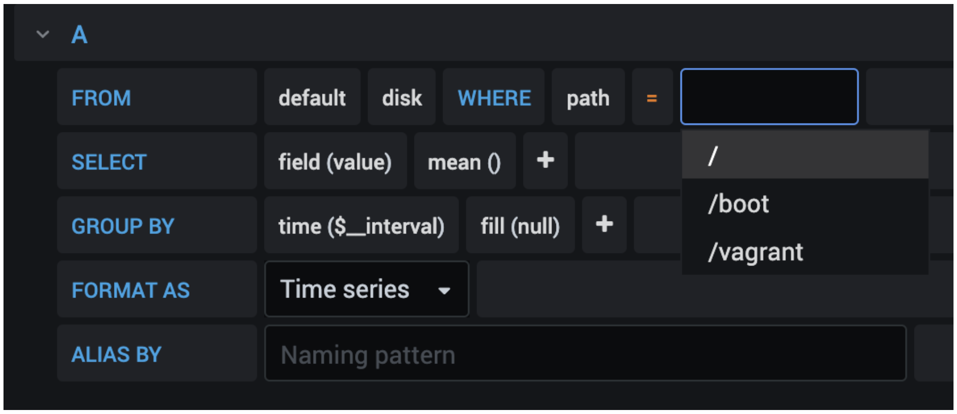
SELECT field (value) mean () +

GROUP BY time ($__interval) fill (null

FORMAT AS Time series

ALIAS BY Naming pattern

Finally, on the "FROM" line, click on "select tag value" and then click on "/".



A

FROM default disk WHERE path =

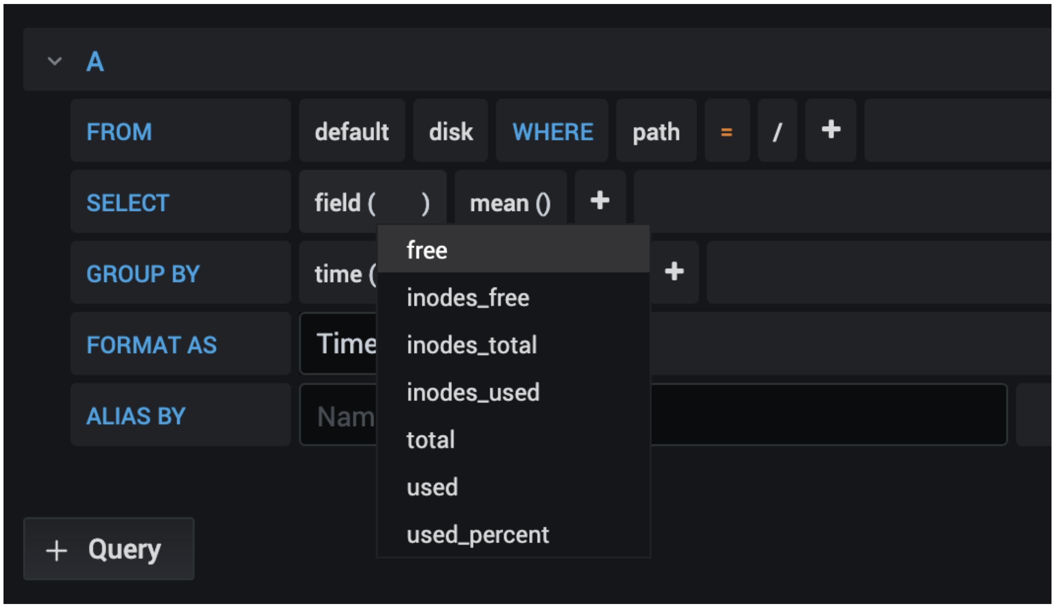
SELECT field (value) mean () +

GROUP BY time ($__interval) fill (null) +

FORMAT AS Time series

ALIAS BY Naming pattern

On the "SELECT" line click "value" and select "free".



A

FROM default disk WHERE path = / +

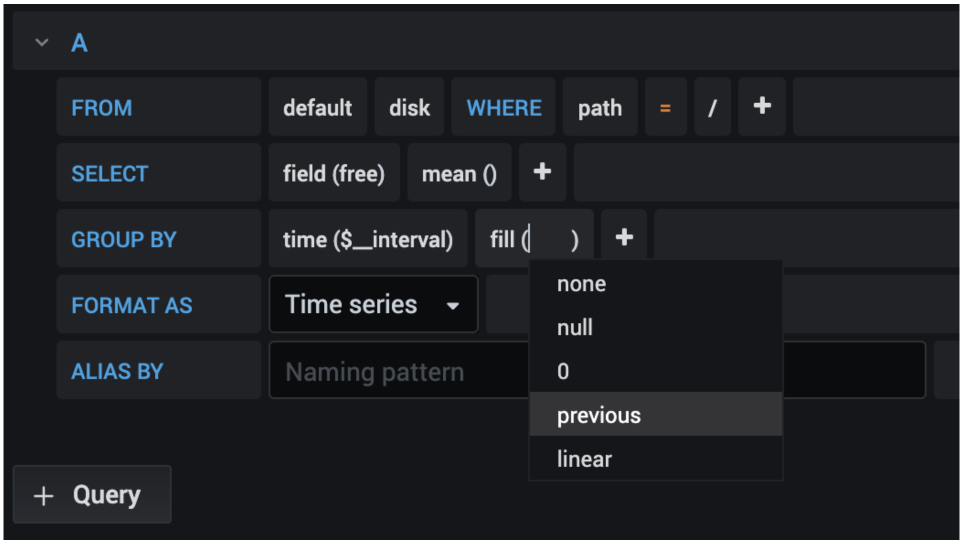
SELECT field ( ) mean () +

GROUP BY time (

FORMAT AS Time

ALIAS BY Nam

On the "GROUP BY" line, click on "null" and then click on "previous".



A

FROM default disk WHERE path = /

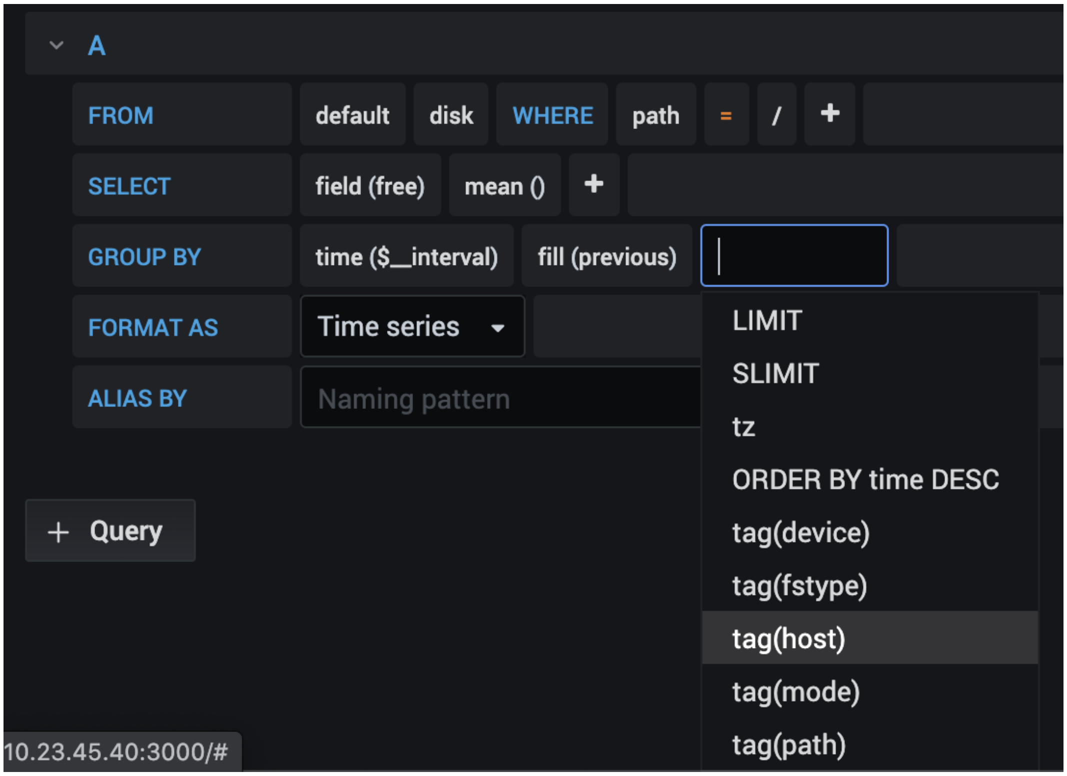
SELECT field (free) mean ()

GROUP BY time ($__interval) fill ( )

FORMAT AS Time series

ALIAS BY Naming pattern

Also on the "GROUP BY" line click the plus sign "+" and then click on "tag(host)".



A

FROM default disk WHERE path = /

SELECT field (free) mean ()

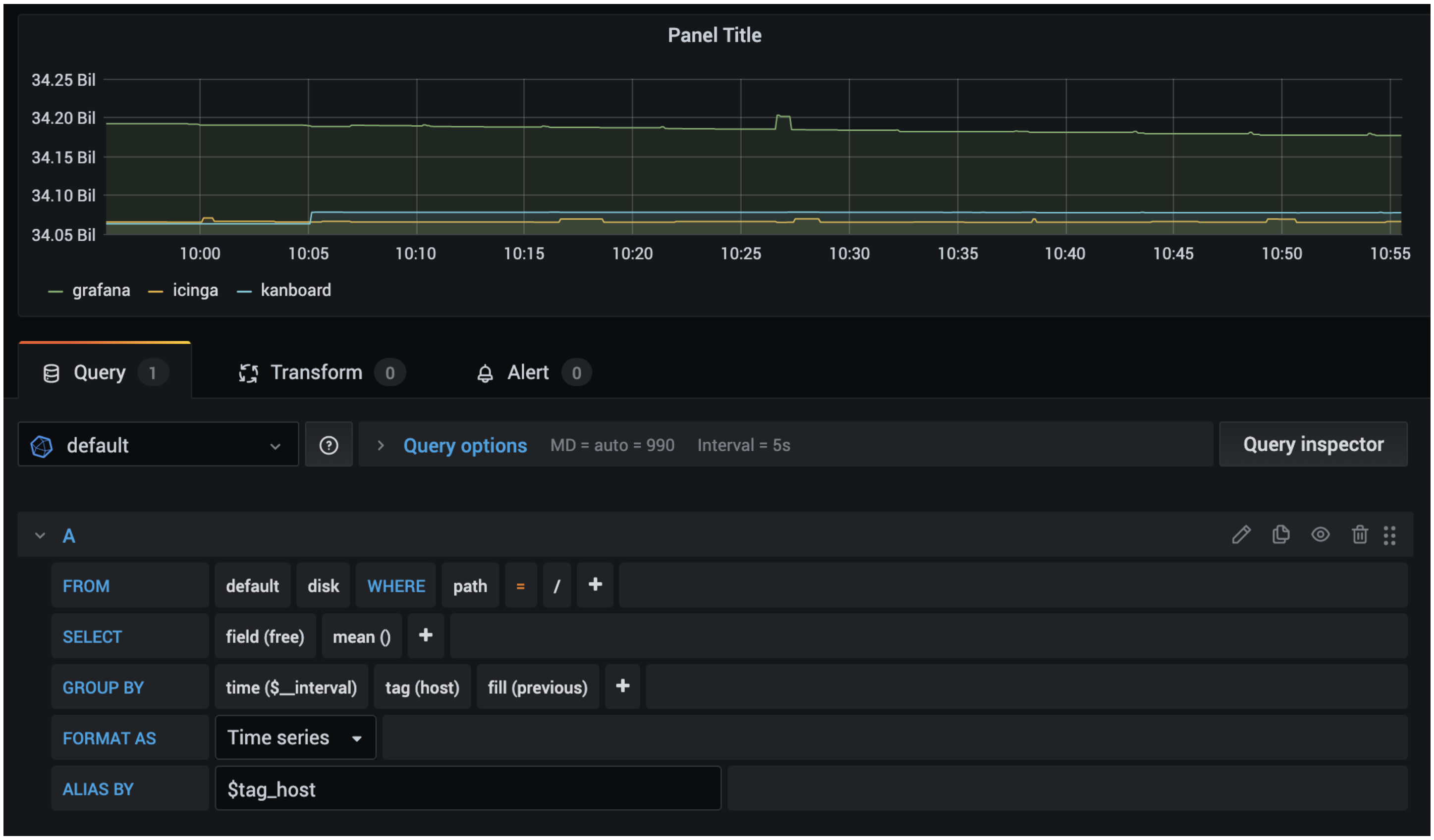
GROUP BY time ($__interval) fill (previous)

FORMAT AS Time series

ALIAS BY Naming pattern

Finally, on the "ALIAS BY" line, type "$tag_host". Click your mouse outside the "ALIAS BY" line to make the change active.

At this point, the query and graph will look something like this:



MD = auto = 990 Interval = 5s

FROM default disk WHERE path = / +

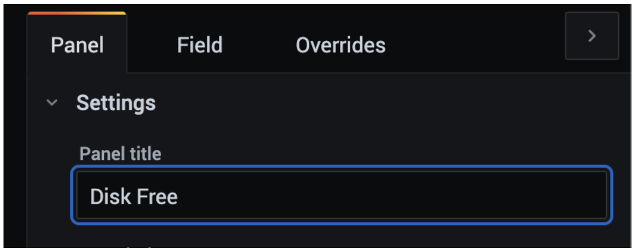
SELECT field (free) mean () +

GROUP BY time ($__interval) tag (host) fill (previous) +

FORMAT AS Time series

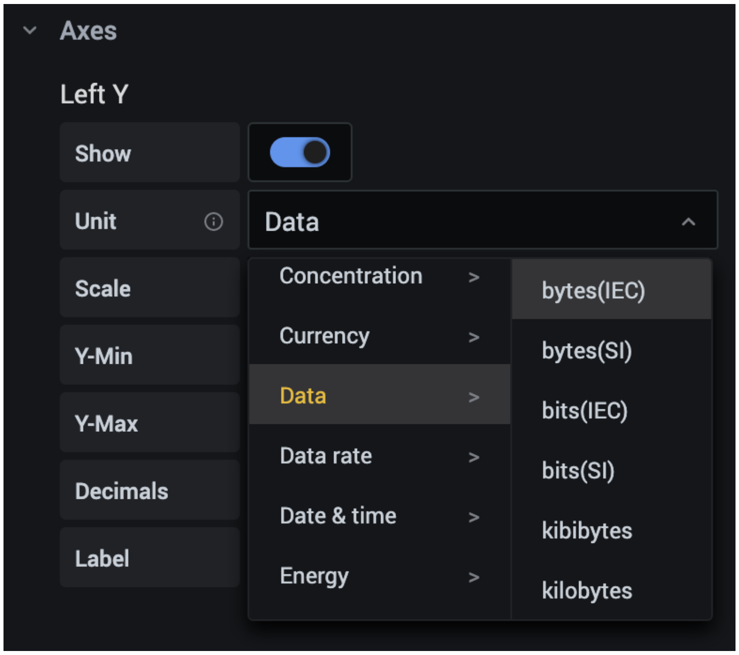
ALIAS BY $tag_host

In the right-hand side of the screen under "Panel" change the "Panel title" to "Disk Free".

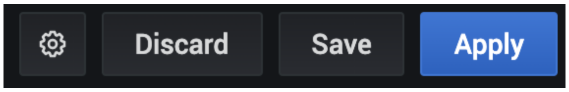


Panel title: Disk Free

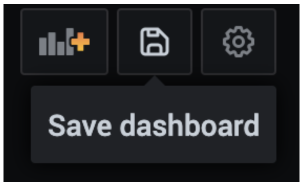
Under the "Axes", then "Left Y", change the "unit" from "short" to "Data > bytes(IEC)".



In the top-right hand corner, click the "Apply" button to save our graph.



Now you will be returned to the main dashboard page. Click on the "Save Dashboard" icon in the top right of your screen.



You can enter an optional note. Click the "Save" button.

By the way, you can resize and rearrange the graphs in your dashboard.

Explore

From this point forward, experiment on your own by creating more graphs and dashboards and/or updating existing dashboards you find on https://grafana.com/grafana/dashboards.

You can even add more hosts and send their metrics to InfluxDB.