

# POC on Monitoring tools

(Grafana, Prometheus, Alert Manager, Blackbox exporter and Node Exporter)

## Grafana

Grafana is a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources.

## Prometheus

Prometheus is a free software application used for event monitoring and alerting. It records real-time metrics in a time series database built using a HTTP pull model, with flexible queries and real-time alerting.

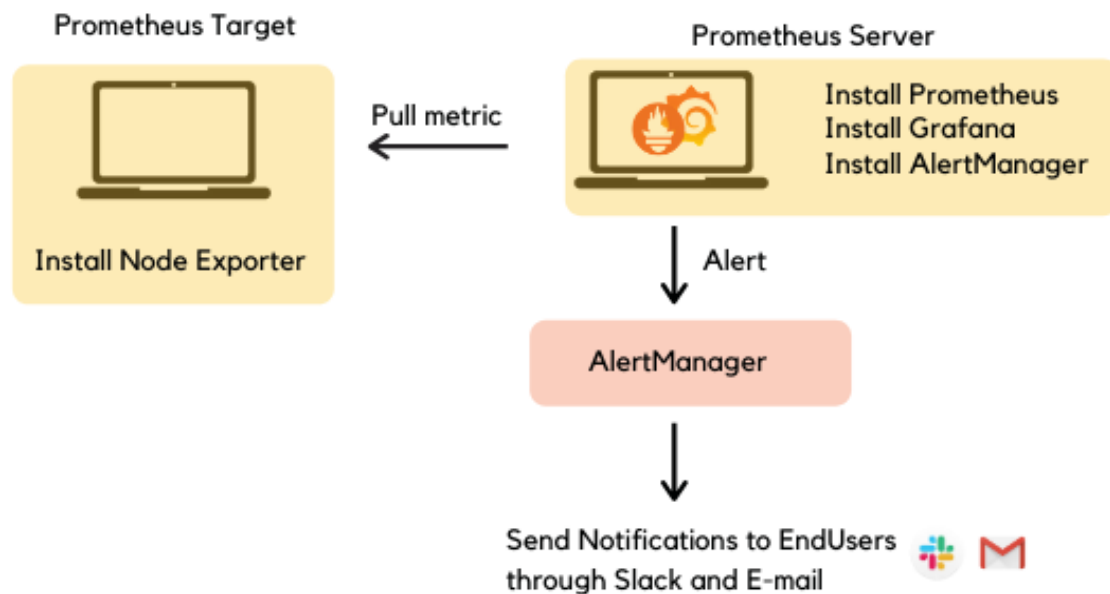
## Prometheus Alert Manager

The Alertmanager handles alerts sent by client applications such as the Prometheus server. It takes care of deduplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

## Node Exporter

Node Exporter is a Prometheus exporter for server level and OS level metrics with configurable metric collectors. It helps us in measuring various server resources such as RAM, disk space, and CPU utilization.

## Working Diagram



## Introduction

This POC includes the step by step procedure of installing different packages like prometheus, grafana, node exporter and alert manager to monitor system resources and generate alerts when a system metric crosses a specific threshold.

## Installing Node Exporter

First, add a node\_exporter user with no login permission and without home directory.

```
$ sudo adduser --no-create-home --disabled-login --shell /bin/false node_exporter
```

Next, download the node exporter binary file.

```
$ wget
```

```
https://github.com/prometheus/node_exporter/releases/download/v1.1.2/node_exporter-1.1.2.linux-386.tar.gz
```

Unzip the downloaded binary file using the following command.

```
$ tar -xvzf node_exporter-1.1.2.linux-386.tar.gz
```

Rename the extracted directory to node\_exporter.

```
$ mv node_exporter-1.1.2.linux-386 node_exporter
```

Go to the node\_exporter directory and copy the executable file to /usr/local/bin directory.

```
$ cd node_exporter
```

```
$ sudo cp node_exporter /usr/local/bin/
```

Change the ownership of the binary executable file.

```
$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
```

Create a service file for node exporter to run it as a daemon and copy the following configuration there.

```
$ sudo nano /etc/systemd/system/node_exporter.service
```

```
[Unit]
```

```
Description=Node Exporter
```

```
[Service]
```

```
User=node_exporter
```

```
Group=node_exporter
```

```
Type=simple
```

```
ExecStart= /usr/local/bin/node_exporter
```

```
Restart=on-failure
```

```
RestartSec=5
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Reload the daemon and start the node\_exporter service.

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl start node_exporter.service
```

```
$ sudo systemctl enable node_exporter.service
```

**Note:** By default Webdock Perfect Server stacks allows traffic from specific ports. You need to open the ports explicitly whenever needed.

```
robo@robo-VirtualBox: ~/node_exporter

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

robo@robo-VirtualBox:~$ sudo adduser --no-create-home --disabled-login --shell /bin/false node_exporter
[sudo] password for robo:
Adding user 'node_exporter' ...
Adding new group 'node_exporter' (1001) ...
Adding new user 'node_exporter' (1001) with group 'node_exporter' ...
Not creating home directory '/home/node_exporter'.
Changing the user information for node_exporter
Enter the new value, or press ENTER for the default
    Full Name []: sumit
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
robo@robo-VirtualBox:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.1.2/node_exporter-1.1.2.linux-386.tar.gz
--2022-01-25 12:01:17-- https://github.com/prometheus/node_exporter/releases/download/v1.1.2/node_exporter-1.1.2.linux-386.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/70c28380-7d9f-11eb-9951-aa90133cbde4?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220125%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220125T063118Z&X-Amz-Expires=300&X-Amz-Signature=1466f99e8b97004d633019521511c8dc7827e9755bf8132ff98125933bf4839c&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.1.2.linux-386.tar.gz&response-content-type=application%2Foctet-stream [following]
--2022-01-25 12:01:18-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/70c28380-7d9f-11eb-9951-aa90133cbde4?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220125%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220125T063118Z&X-Amz-Expires=300&X-Amz-Signature=1466f99e8b97004d633019521511c8dc7827e9755bf8132ff98125933bf4839c&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.1.2.linux-386.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8916315 (8.5M) [application/octet-stream]
Saving to: 'node_exporter-1.1.2.linux-386.tar.gz'

node_exporter-1.1.2 100%[=====] 8.50M 421KB/s in 34s

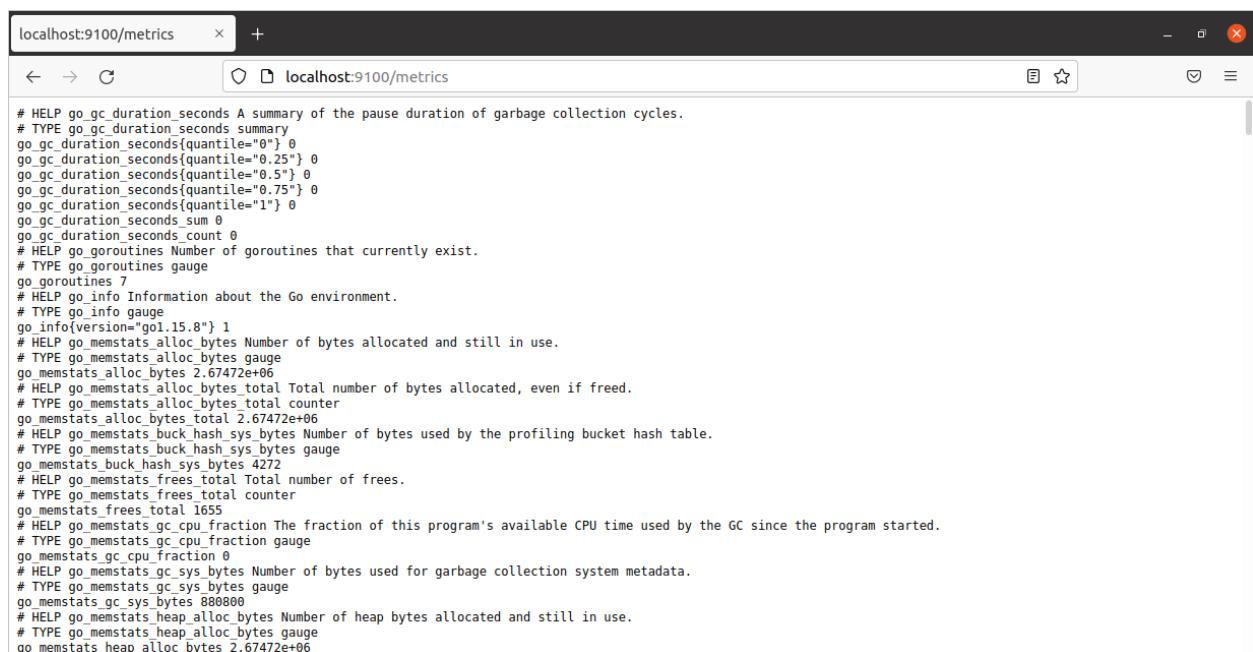
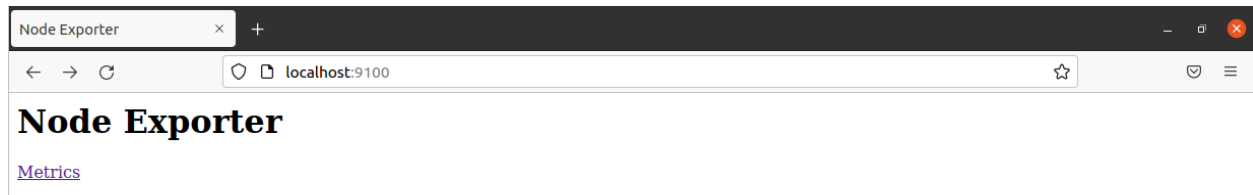
2022-01-25 12:01:54 (254 KB/s) - 'node_exporter-1.1.2.linux-386.tar.gz' saved [8916315/8916315]

robo@robo-VirtualBox:~$ tar -xvzf node_exporter-1.1.2.linux-386.tar.gz
node_exporter-1.1.2.linux-386/
node_exporter-1.1.2.linux-386/LICENSE
node_exporter-1.1.2.linux-386/NOTICE
node_exporter-1.1.2.linux-386/node_exporter
robo@robo-VirtualBox:~$ mv node_exporter-1.1.2.linux-386 node_exporter
robo@robo-VirtualBox:~$ cd node_exporter
robo@robo-VirtualBox:~/node_exporter$ sudo cp node_exporter /usr/local/bin/
robo@robo-VirtualBox:~/node_exporter$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
robo@robo-VirtualBox:~/node_exporter$ sudo nano /etc/systemd/system/node_exporter.service
robo@robo-VirtualBox:~/node_exporter$ sudo systemctl daemon-reload
robo@robo-VirtualBox:~/node_exporter$ sudo systemctl start node_exporter.service
robo@robo-VirtualBox:~/node_exporter$ sudo systemctl enable node_exporter.service
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
robo@robo-VirtualBox:~/node_exporter$ sudo ufw allow 9100/tcp
Rules updated
Rules updated (v6)
robo@robo-VirtualBox:~/node_exporter$
```

In case of a Webdock Perfect Server stack, open the port 9100 on the server.

```
$ sudo ufw allow 9100/tcp
```

Now check the port 9100 of the server, it will list the system metrics in raw form.



## Installing Prometheus

Create a prometheus user with no login permission and without home directory.

```
$ sudo adduser --no-create-home --disabled-login --shell /bin/false prometheus
```

Download the Prometheus binary file.

```
$ wget
```

<https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz>

Unzip the downloaded binary file.

```
$ tar -xvzf prometheus-2.26.0.linux-amd64.tar.gz
```

Change the directory name to prometheus.

```
$ mv prometheus-2.26.0.linux-amd64 prometheus
```

Go to the prometheus directory.

```
$ cd prometheus
```

Create configuration and data directories for Prometheus.

```
$ sudo mkdir -p /etc/prometheus/{rules,rules.d,files_sd} /var/lib/prometheus
```

Copy the binary files to the /usr/local/bin directory.

```
$ sudo cp prometheus promtool /usr/local/bin
```

Copy the libraries and console file to the /etc/prometheus directory

```
$ sudo cp -r consoles/ console_libraries/ /etc/prometheus/
```

Copy the configuration file to the /etc/prometheus directory.

```
$ sudo cp prometheus.yml /etc/prometheus/
```

Change the ownership of the files and directories.

```
$ sudo chown -R prometheus:prometheus /etc/prometheus/ /var/lib/prometheus  
/usr/local/bin/{promtool,prometheus}
```

Create a service file to allow Prometheus to run in the background

```
$ sudo nano /etc/systemd/system/prometheus.service
```

[Unit]

Description=Prometheus

[Service]

User=prometheus

Group=prometheus

Type=simple

ExecStart= /usr/local/bin/prometheus \

--config.file=/etc/prometheus/prometheus.yml \

--storage.tsdb.path=/var/lib/prometheus \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console\_libraries \

--web.listen-address=0.0.0.0:9090

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

Reload the daemon and start the service.

```
$ sudo systemctl daemon-reload
$ sudo systemctl start prometheus.service
$ sudo systemctl enable prometheus.service
```

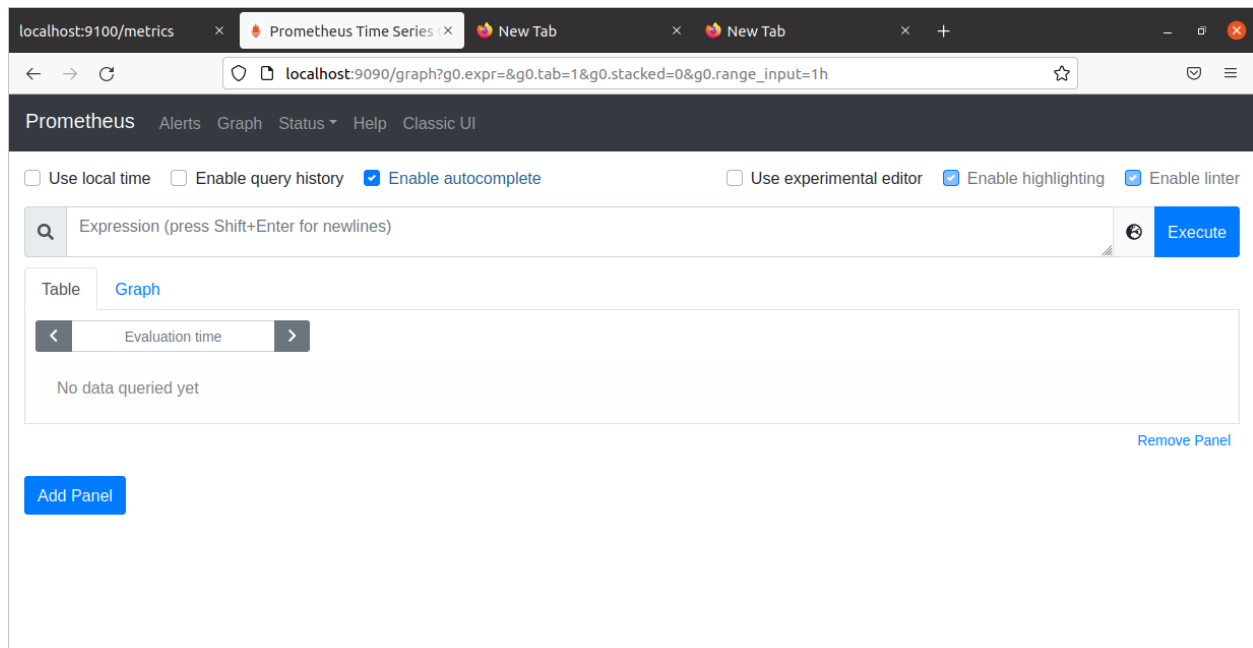
In case of Webdock Perfect Server stack, open the port 9090.

```
$ sudo ufw allow 9090/tcp
```

```
robo@robo-VirtualBox: ~
robo@robo-VirtualBox:~$ sudo adduser --no-create-home --disabled-login --shell /bin/false prometheus
[sudo] password for robo:
Adding user 'prometheus' ...
Adding new group 'prometheus' (1002) ...
Adding new user 'prometheus' (1002) with group 'prometheus' ...
Not creating home directory '/home/prometheus'.
Changing the user information for prometheus
Enter the new value, or press ENTER for the default
    Full Name []: prometheus
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
robo@robo-VirtualBox:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz
--2022-01-25 12:35:30-- https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/4394b400-922b-11eb-9d10-18661b9cfc8a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220125%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220125T070531Z&X-Amz-Expires=300&X-Amz-Signature=334d189f83a7e3f3ca0bec6530a8055acd9633985317742db78304245a3474a4&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=6838921&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.26.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2022-01-25 12:35:31-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/4394b400-922b-11eb-9d10-18661b9cfc8a?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220125%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220125T070531Z&X-Amz-Expires=300&X-Amz-Signature=334d189f83a7e3f3ca0bec6530a8055acd9633985317742db78304245a3474a4&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=6838921&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.26.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
robo@robo-VirtualBox: ~/prometheus
robo@robo-VirtualBox:~$ tar -xvzf prometheus-2.26.0.linux-amd64.tar.gz
prometheus-2.26.0.linux-amd64/
prometheus-2.26.0.linux-amd64/consoles/
prometheus-2.26.0.linux-amd64/consoles/index.html.example
prometheus-2.26.0.linux-amd64/consoles/node-cpu.html
prometheus-2.26.0.linux-amd64/consoles/node-disk.html
prometheus-2.26.0.linux-amd64/consoles/node-overview.html
prometheus-2.26.0.linux-amd64/consoles/node.html
prometheus-2.26.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.26.0.linux-amd64/consoles/prometheus.html
prometheus-2.26.0.linux-amd64/console_libraries/
prometheus-2.26.0.linux-amd64/console_libraries/menu.lib
prometheus-2.26.0.linux-amd64/console_libraries/prom.lib
prometheus-2.26.0.linux-amd64/prometheus.yml
prometheus-2.26.0.linux-amd64/LICENSE
prometheus-2.26.0.linux-amd64/NOTICE
prometheus-2.26.0.linux-amd64/prometheus
prometheus-2.26.0.linux-amd64/promtool
robo@robo-VirtualBox:~$ mv prometheus-2.26.0.linux-amd64 prometheus
robo@robo-VirtualBox:~$ cd prometheus
robo@robo-VirtualBox:~/prometheus$ sudo mkdir -p /etc/prometheus/{rules,rules.d,files_sd} /var/lib/prometheus
robo@robo-VirtualBox:~/prometheus$ sudo cp prometheus promtool /usr/local/bin
robo@robo-VirtualBox:~/prometheus$ sudo cp -r consoles/ console_libraries/ /etc/prometheus/
robo@robo-VirtualBox:~/prometheus$ sudo cp prometheus.yml /etc/prometheus/
robo@robo-VirtualBox:~/prometheus$ sudo chown -R prometheus:prometheus /etc/prometheus/ /var/lib/prometheus /usr/local/bin/{promtool,prometheus}
robo@robo-VirtualBox:~/prometheus$ sudo nano /etc/systemd/system/prometheus.service
robo@robo-VirtualBox:~/prometheus$ sudo systemctl daemon-reload
robo@robo-VirtualBox:~/prometheus$ sudo systemctl start prometheus.service
robo@robo-VirtualBox:~/prometheus$ sudo systemctl enable prometheus.service
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
```

Visit port 9090 of the server in a browser (e.g. <http://mysite.com:9090>) and it will show the following web page.



## Installing Grafana

Update the system.

```
$ sudo apt-get update -y
```

Add gpg key for Grafana.

```
$ sudo apt-get install gnupg2 curl -y
```

```
$ curl https://packages.grafana.com/gpg.key | $ sudo apt-key add -
```

Add apt-repository for Grafana.

```
$ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
```

Update the apt-repositories on ubuntu.

```
$ sudo apt-get update -y
```

Install Grafana using the following command.

```
$ sudo apt-get install grafana -y
```

Reload the daemon and start the Grafana server.

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl start grafana-server.service
```

```
$ sudo systemctl enable grafana-server.service
```



In case of a Webdock Perfect Server stack, open the port 3000.

\$ sudo ufw allow 3000/tcp

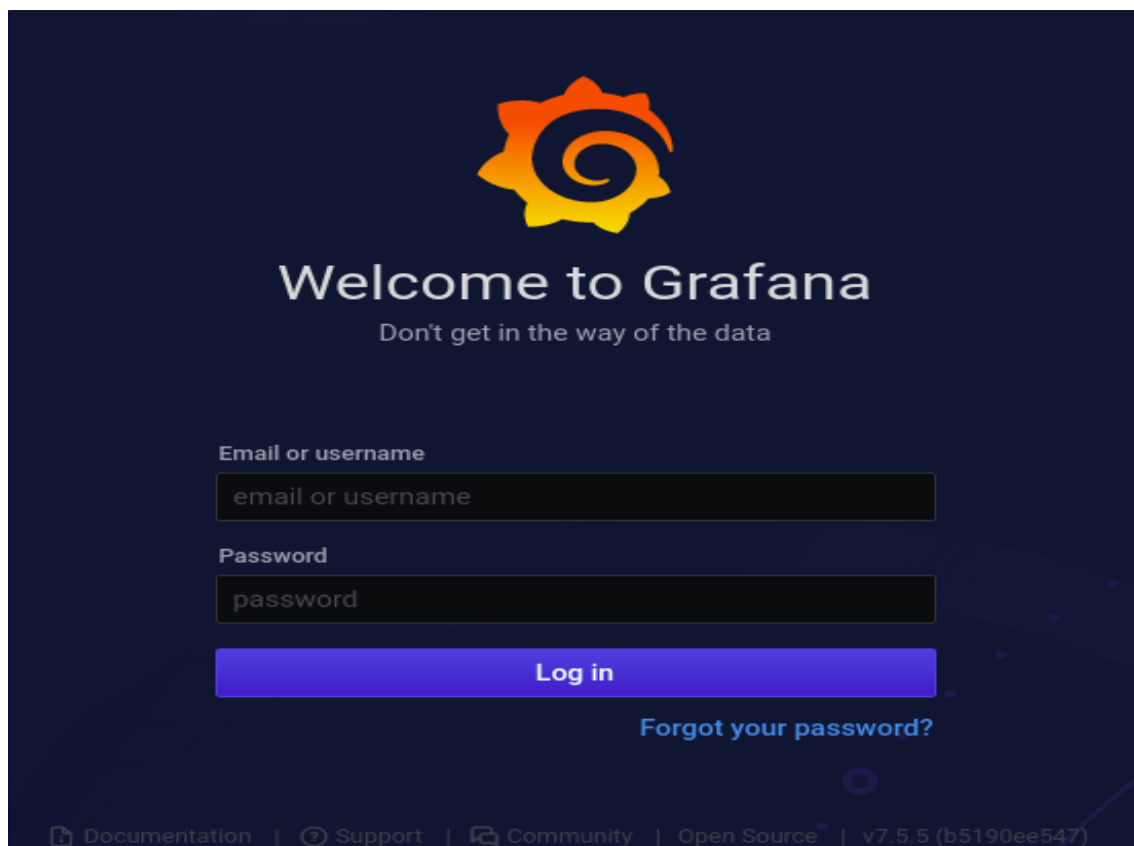
```
robo@robo-VirtualBox: ~/prometheus
robo@robo-VirtualBox:~/prometheus$ sudo apt-get update -y
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [66.5 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 223 kB in 13s (17.1 kB/s)
Reading package lists... Done
robo@robo-VirtualBox:~/prometheus$ sudo apt-get install gnupg2 curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl gnupg2
The following packages will be upgraded:
  libcurl4
1 upgraded, 2 newly installed, 0 to remove and 188 not upgraded.
Need to get 166 kB/400 kB of archives.
After this operation, 464 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.7 [161 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 gnupg2 all 2.2.19-3ubuntu2.1 [4,584 B]
Fetched 166 kB in 11s (14.8 kB/s)
(Reading database ... 181252 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.7_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.7) over (7.68.0-1ubuntu2.6) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.7) ...
Setting up gnupg2 (2.2.19-3ubuntu2.1) ...
```

```
robo@robo-VirtualBox: ~/prometheus
robo@robo-VirtualBox:~/prometheus$ curl https://packages.grafana.com/gpg.key | sudo apt-key add -
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 1694 100 1694 0 0 490 0 0:00:03 0:00:03 --:--:-- 490
OK
robo@robo-VirtualBox:~/prometheus$ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://packages.grafana.com/oss/deb stable InRelease [12.1 kB]
Get:6 https://packages.grafana.com/oss/deb stable/main amd64 Packages [28.0 kB]
Fetched 40.1 kB in 12s (3,298 B/s)
Reading package lists... Done
robo@robo-VirtualBox:~/prometheus$ sudo apt-get update -y
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://packages.grafana.com/oss/deb stable InRelease
Reading package lists... Done
robo@robo-VirtualBox:~/prometheus$ sudo apt-get install grafana -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  grafana
0 upgraded, 1 newly installed, 0 to remove and 188 not upgraded.
```



```
robo@robo-VirtualBox: ~/prometheus
31% [1 grafana 28.5 MB/72.7 MB 39%]
Get:1 https://packages.grafana.com/oss/deb stable/main amd64 grafana amd64 8.3.4 [72.7 MB]
73% [1 grafana 66.3 MB/72.7 MB 91%]
Get:1 https://packages.grafana.com/oss/deb stable/main amd64 grafana amd64 8.3.4 [72.7 MB]
Fetched 6,456 kB in 1h 25min 7s (1,264 B/s)
Selecting previously unselected package grafana.
(Reading database ... 181265 files and directories currently installed.)
Preparing to unpack .../grafana_8.3.4_amd64.deb ...
Unpacking grafana (8.3.4) ...
Setting up grafana (8.3.4) ...
Adding system user 'grafana' (UID 127) ...
Adding new user 'grafana' (UID 127) with group 'grafana' ...
Not creating home directory '/usr/share/grafana'.
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
sudo /bin/systemctl start grafana-server
Processing triggers for systemd (245.4-4ubuntu3.11) ...
robo@robo-VirtualBox:~/prometheus$ sudo systemctl daemon-reload
[sudo] password for robo:
robo@robo-VirtualBox:~/prometheus$ sudo systemctl start grafana-service
Failed to start grafana-service.service: Unit grafana-service.service not found.
robo@robo-VirtualBox:~/prometheus$ sudo systemctl start grafana-server.service
robo@robo-VirtualBox:~/prometheus$ sudo systemctl enable grafana-server.service
Synchronizing state of grafana-server.service with SysV
service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.
robo@robo-VirtualBox:~/prometheus$ sudo ufw allow 3000/tcp
Rules updated
Rules updated (v6)
robo@robo-VirtualBox:~/prometheus$
```

Visit port 3000 of the server in a browser (e.g. <http://mysite.com:3000>) of the server and it will display a login page.



## Adding Targets to Prometheus

Open the prometheus.yml file.

```
$ sudo nano /etc/prometheus/prometheus.yml
```

Add the following configuration in the file.

global:

```
scrape_interval: 15s
```

```
evaluation_interval: 15s
```

scrape\_configs:

```
- job_name: 'System Resources'
```

```
  scrape_interval: 10s
```

```
  static_configs:
```

```
  - targets: ['localhost:9100']
```

```
- job_name: 'Prometheus'
```

```
  scrape_interval: 10s
```

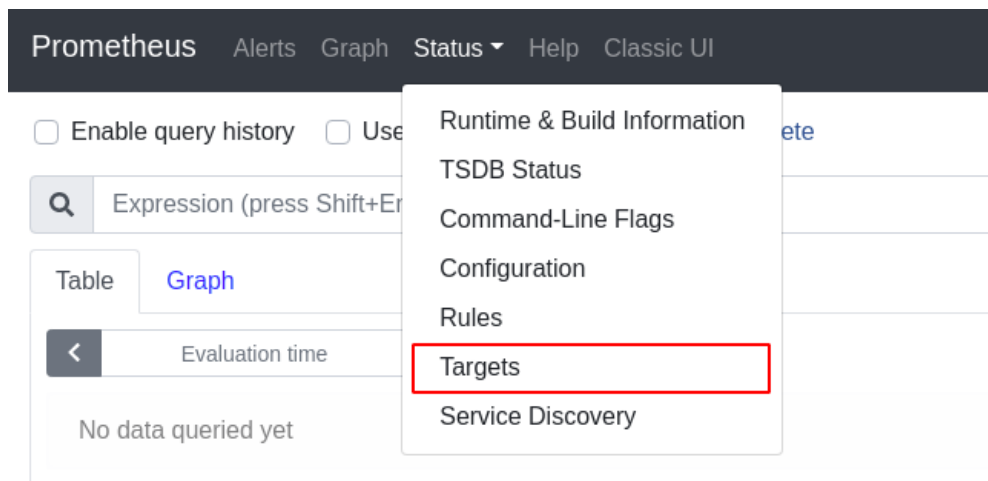
```
  static_configs:
```

```
  - targets: ['localhost:9090']
```

Restart the prometheus service.

```
$ sudo systemctl restart prometheus
```

Visit port 9090 of the server in a browser (e.g. <http://mysite.com:9090>) and list all the targets.

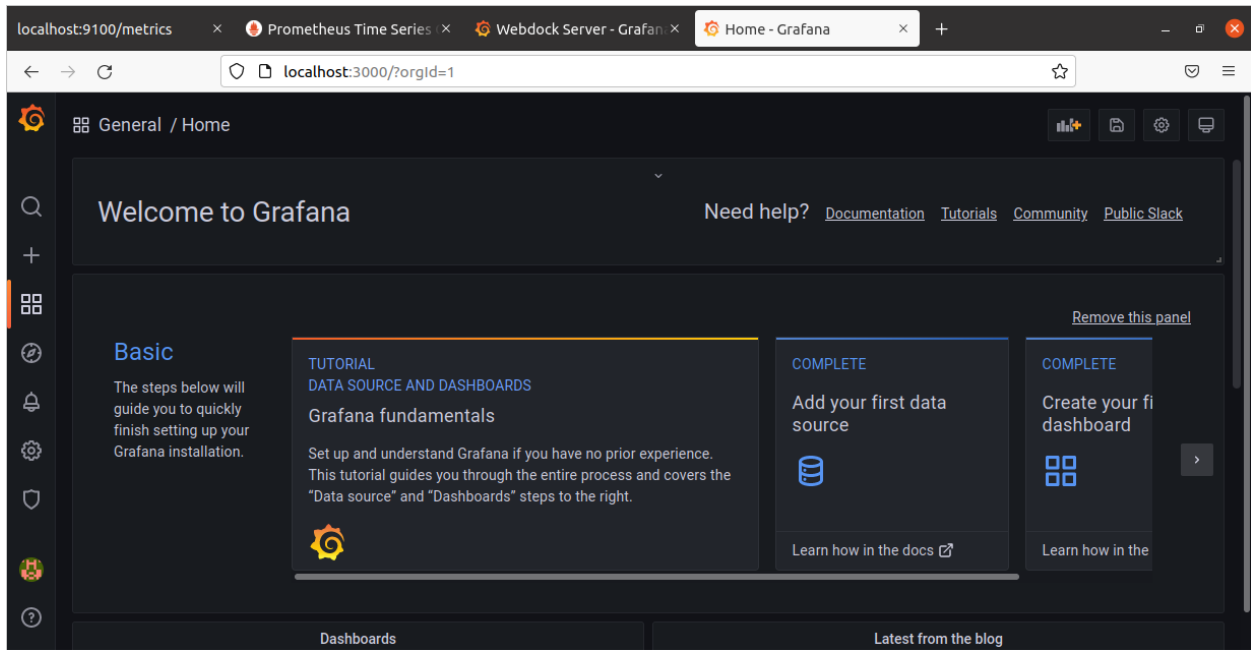


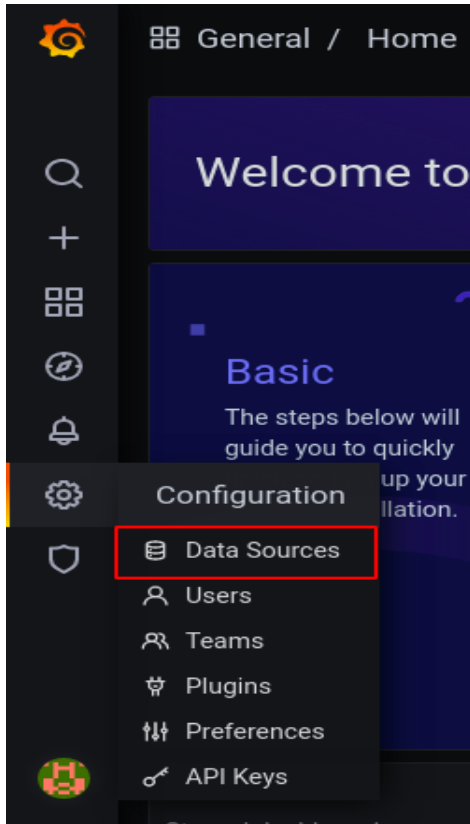
Add Panel

Prometheus Alerts Graph Status ▾ Help Classic UI					
Targets					
All Unhealthy Collapse All					
System Resources (1/1 up) <a href="#">show less</a>					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9100/metrics">http://localhost:9100/metrics</a>	UP	instance="localhost:9100" job="System Resources"	9.997s ago	110.398ms	
prometheus (1/1 up) <a href="#">show less</a>					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	instance="localhost:9090" job="prometheus"	3.706s ago	5.792ms	

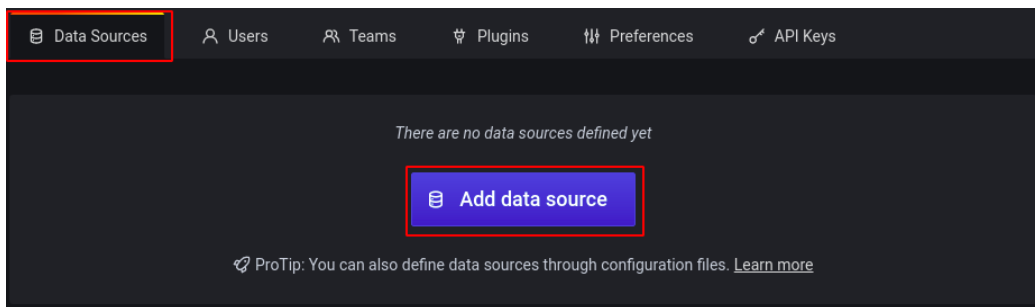
Adding Data Source and Dashboard to Grafana

Log into the Grafana Dashboard using the default username and password (i.e admin). Hover over the settings icon on the left side menu and then click on ‘Data Sources’.

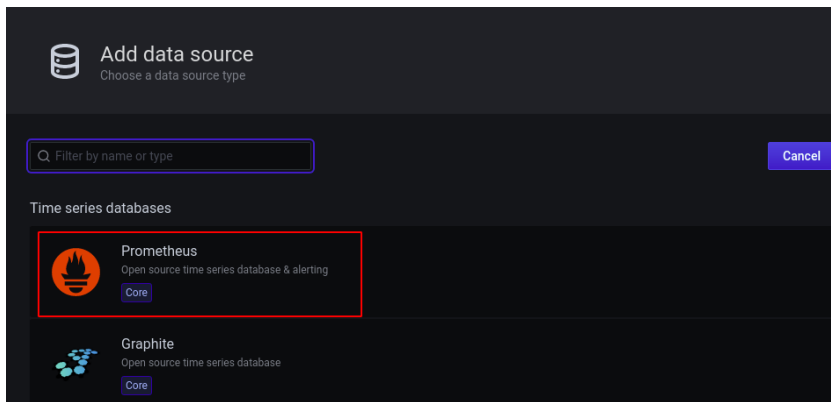




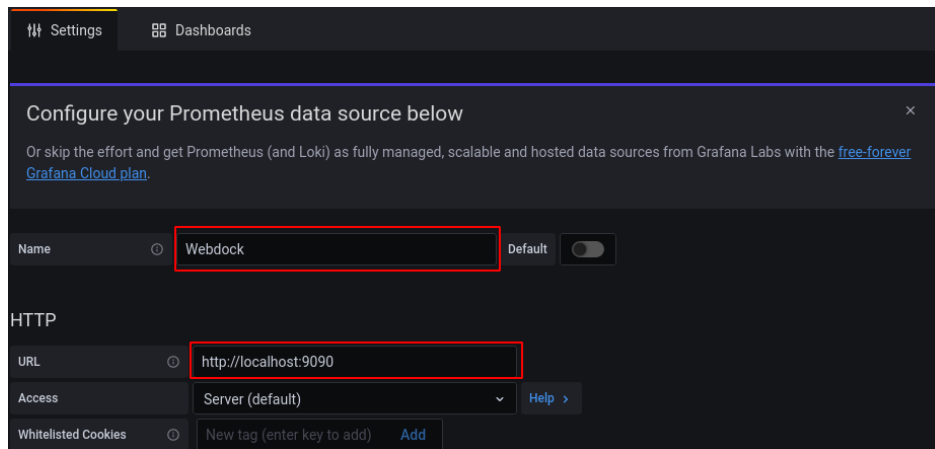
It will open a configuration page. Select the 'Data Sources' tab and click on 'Add data source'.



On the 'Add data source' page select the Prometheus as data source.

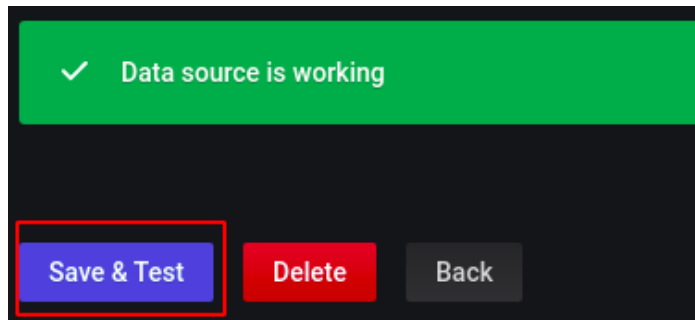


Provide the name and the url of the Prometheus data source.

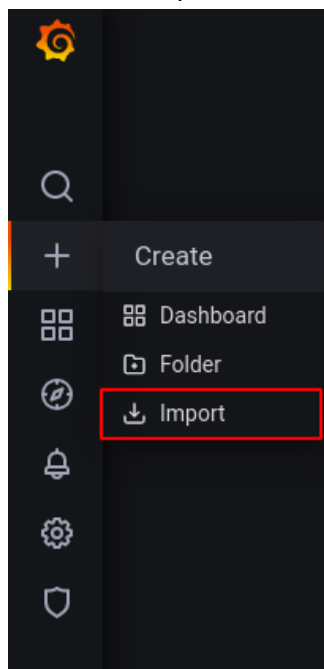


The screenshot shows the 'Configure your Prometheus data source' form in Grafana. The form has a title bar with a close button. Below the title, there is a link to the Grafana Cloud plan. The form contains several fields: 'Name' with the value 'Webdock', 'URL' with the value 'http://localhost:9090', 'Access' set to 'Server (default)', and 'Whitelisted Cookies' with a text input and an 'Add' button. A 'Default' toggle switch is also present.

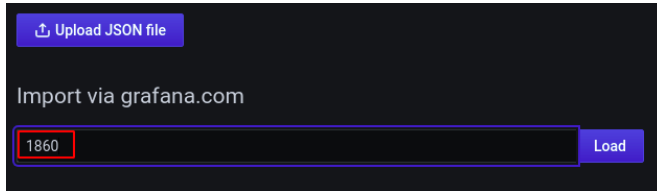
Click on the 'save & test' button at the end of the page and it will show the status of the Data Source.



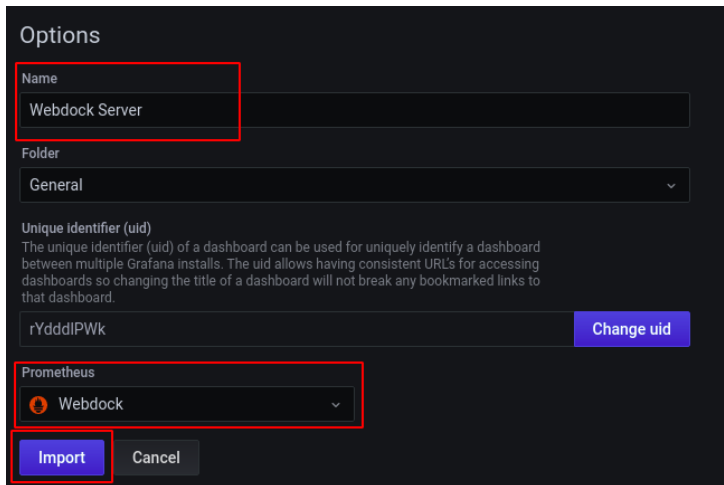
In order to import Grafana dashboard, hover over the 'Add' symbol and click on 'import'.



Enter the Grafana dashboard id and click on 'load'.

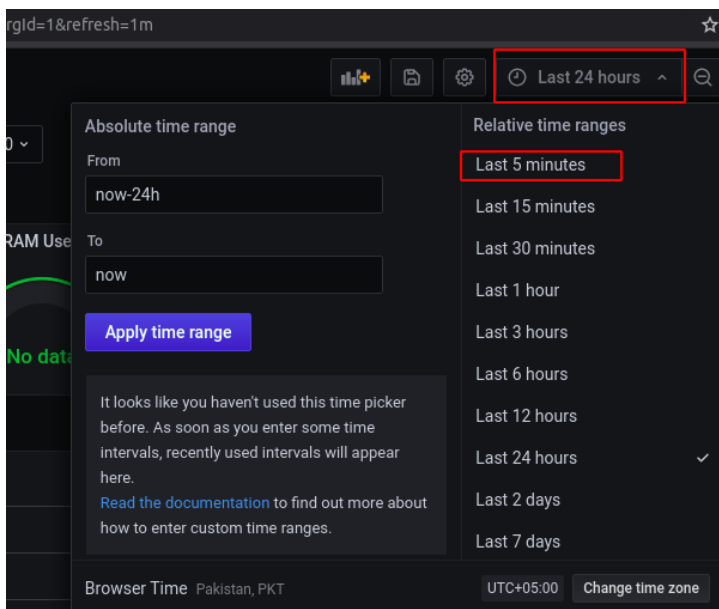
A screenshot of the Grafana dashboard import interface. At the top, there is a blue button labeled 'Upload JSON file'. Below it, the text 'Import via grafana.com' is displayed. A text input field contains the number '1860', which is highlighted with a red rectangular box. To the right of this field is a blue button labeled 'Load'.

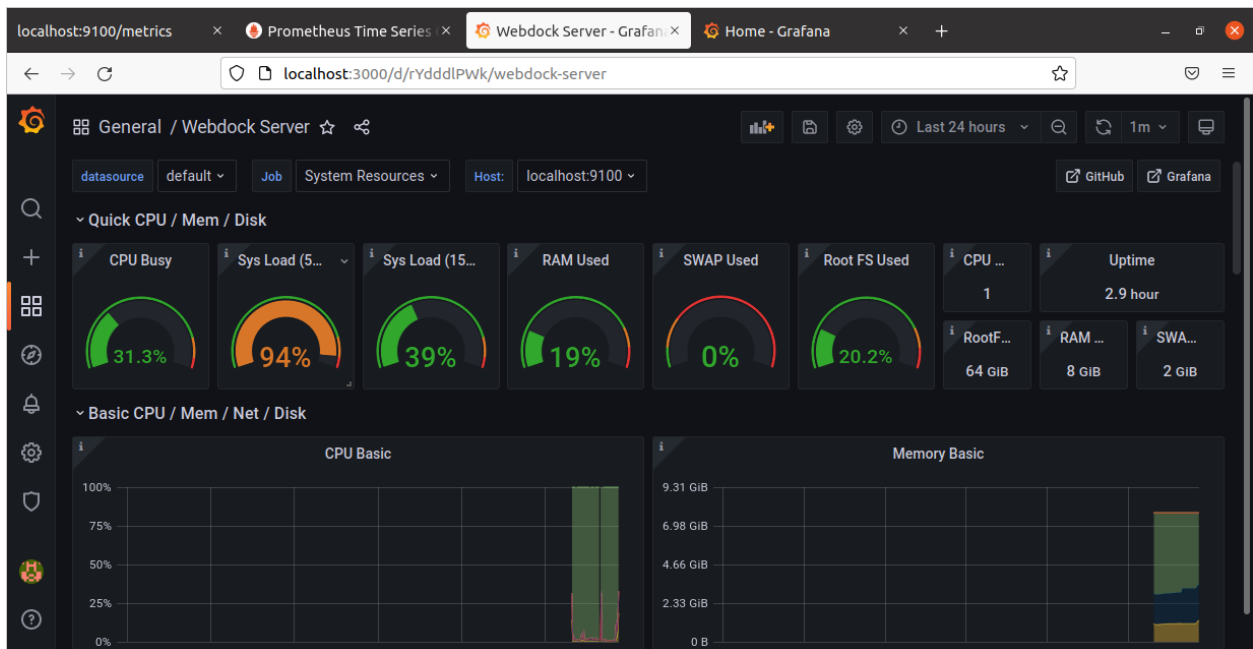
After loading the dashboard, it will ask for the dashboard name and the Prometheus data source. Click on import after providing the required details.

A screenshot of the 'Options' screen in Grafana. The 'Name' field contains 'Webdock Server' and is highlighted with a red box. Below it, the 'Folder' dropdown is set to 'General'. A section for 'Unique Identifier (uid)' explains its purpose and shows a default value 'rYdddIPWk' with a 'Change uid' button. The 'Prometheus' dropdown is set to 'Webdock' and is also highlighted with a red box. At the bottom, there are 'Import' and 'Cancel' buttons, with 'Import' being highlighted by a red box.

## Monitoring System Resources using Grafana

After importing the dashboard, it will display a monitoring dashboard. Click on the time range selector on the top right corner of the dashboard and select a time range.

A screenshot of the Grafana dashboard's time range selector. The top bar shows 'Last 24 hours' with a dropdown arrow, highlighted by a red box. A modal is open showing 'Absolute time range' (From: now-24h, To: now) and 'Relative time ranges'. In the relative ranges list, 'Last 5 minutes' is highlighted with a red box. Other options include 'Last 15 minutes', 'Last 30 minutes', 'Last 1 hour', 'Last 3 hours', 'Last 6 hours', 'Last 12 hours', 'Last 24 hours' (which has a checkmark), 'Last 2 days', and 'Last 7 days'. An 'Apply time range' button is at the bottom of the modal. A message at the bottom of the modal states: 'It looks like you haven't used this time picker before. As soon as you enter some time intervals, recently used intervals will appear here. Read the documentation to find out more about how to enter custom time ranges.' The bottom of the screen shows 'Browser Time Pakistan, PKT' and 'UTC+05:00' with a 'Change time zone' button.



- > CPU / Memory / Net / Disk (7 panels)
- > **Memory Meminfo** (15 panels)
- > Memory Vmstat (4 panels)
- > System Timesync (4 panels)
- > System Processes (7 panels)
- > System Misc (7 panels)
- > Hardware Misc (3 panels)
- > Systemd (2 panels)
- > Storage Disk (8 panels)
- > Storage Filesystem (5 panels)
- > Network Traffic (17 panels)
- > Network Sockstat (5 panels)
- > Network Netstat (11 panels)



## Installing Prometheus Alert Manager

Create an alert\_manager user without a home directory and no login permissions.

```
$ sudo adduser --no-create-home --disabled-login --shell /bin/false alert_manager
```

Download the alert manager binary file.

```
$ wget
```

```
https://github.com/prometheus/alertmanager/releases/download/v0.22.0-rc.2/alertmanager-0.22.0-rc.2.linux-amd64.tar.gz
```

Unzip the downloaded binary file.

```
$ tar -xvzf alertmanager-0.22.0-rc.2.linux-amd64.tar.gz
```

Rename the extracted directory.

```
$ mv alertmanager-0.22.0-rc.2.linux-amd64 alertmanager
```

Go to the alertmanager directory.

```
$ cd alertmanager
```

Create a data directory for alertmanager.

```
$ sudo mkdir /var/lib/alertmanager
```

Copy the amtool and alertmanager executable files to /usr/local/bin directory.

```
$ sudo cp amtool alertmanager /usr/local/bin/
```

Copy the alertmanager.yml file to the /etc/prometheus/ directory.

```
$ sudo cp alertmanager.yml /etc/prometheus/
```

Change the ownership of the executable files and data directory.

```
$ sudo chown -R alert_manager:alert_manager /usr/local/bin/{amtool,alertmanager}
/var/lib/alertmanager
```

Create a service file to run the alertmanager in the background.

```
$ sudo nano /etc/systemd/system/alertmanager.service
```

```
[Unit]
```

```
Description=Alert Manager
```

```
[Service]
```

```
User=alert_manager
```

```
Group=alert_manager
```

```
Type=simple
```

```
ExecStart= /usr/local/bin/alertmanager \
```

```
--config.file=/etc/prometheus/alertmanager.yml \
```

```
--storage.path=/var/lib/alertmanager/ \
```

--cluster.listen-address=127.0.0.1:9094

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

Reload the daemon and start the service.

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl start alertmanager.service
```

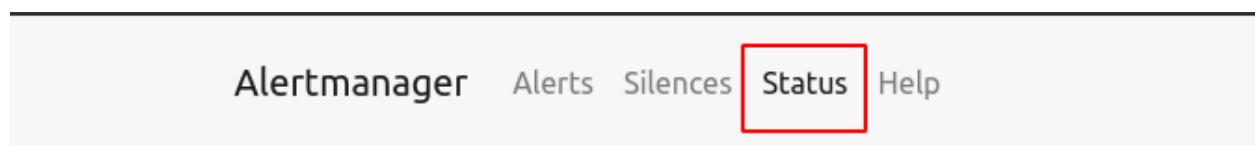
```
$ sudo systemctl enable alertmanager.service
```

In case of a Webdock Perfect Server stack, open port 9094 and 9093.

```
$ sudo ufw allow 9094/tcp
```

```
$ sudo ufw allow 9093/tcp
```

Visit port 9093 of the server in a browser (e.g. <http://mysite.com:9093>) and check the alert manager cluster status.



# Status

**Uptime:** 2021-05-22T06:29:31.167Z

## Cluster Status

**Name:** 01F69C7ECWNCNRWMK14Z00SBFS

**Status:** ready

**Peers:**

- **Name:** 01F69C7ECWNCNRWMK14Z00SBFS  
**Address:** 127.0.0.1:9094

## Configuring Alert Manager

Open the prometheus.yml file and append the following configuration.

```
$ sudo nano /etc/prometheus/prometheus.yml
```

alerting:

alertmanagers:

- static\_configs:

- targets:

- 'localhost:9093'

rule\_files:

- "rules.yml"

Create a file named 'rules.yml' at /etc/prometheus directory and add the alert rules there.

```
$ sudo nano /etc/prometheus/rules.yml
```

groups:

- name: alert.rules

rules:

- alert: InstanceDown

expr: up == 0

for: 1m

labels:

severity: "critical"

annotations:

summary: "Endpoint {{ \$labels.instance }} down"

description: "{{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minutes."

- alert: HostOutOfMemory

expr: node\_memory\_MemAvailable\_bytes / node\_memory\_MemTotal\_bytes \* 100 < 25

for: 5m

labels:

severity: warning

annotations:

summary: "Host out of memory (instance {{ \$labels.instance }})"

description: "Node memory is filling up (< 25% left)\n VALUE = {{ \$value }}\n LABELS: {{ \$labels }}"

- alert: HostOutOfDiskSpace

expr: (node\_filesystem\_avail\_bytes{mountpoint="/" } \* 100) / node\_filesystem\_size\_bytes{mountpoint="/" } < 50

for: 10s

labels:

severity: warning

annotations:

summary: "Host out of disk space (instance {{ \$labels.instance }})"

description: "Disk is almost full (< 50% left)\n VALUE = {{ \$value }}\n LABELS: {{ \$labels }}"

- alert: HostHighCpuLoad

expr: (sum by (instance) (irate(node\_cpu\_seconds\_total{job="System Resources",mode="idle"}[5m]))) > 80

for: 5m

labels:

severity: warning

annotations:

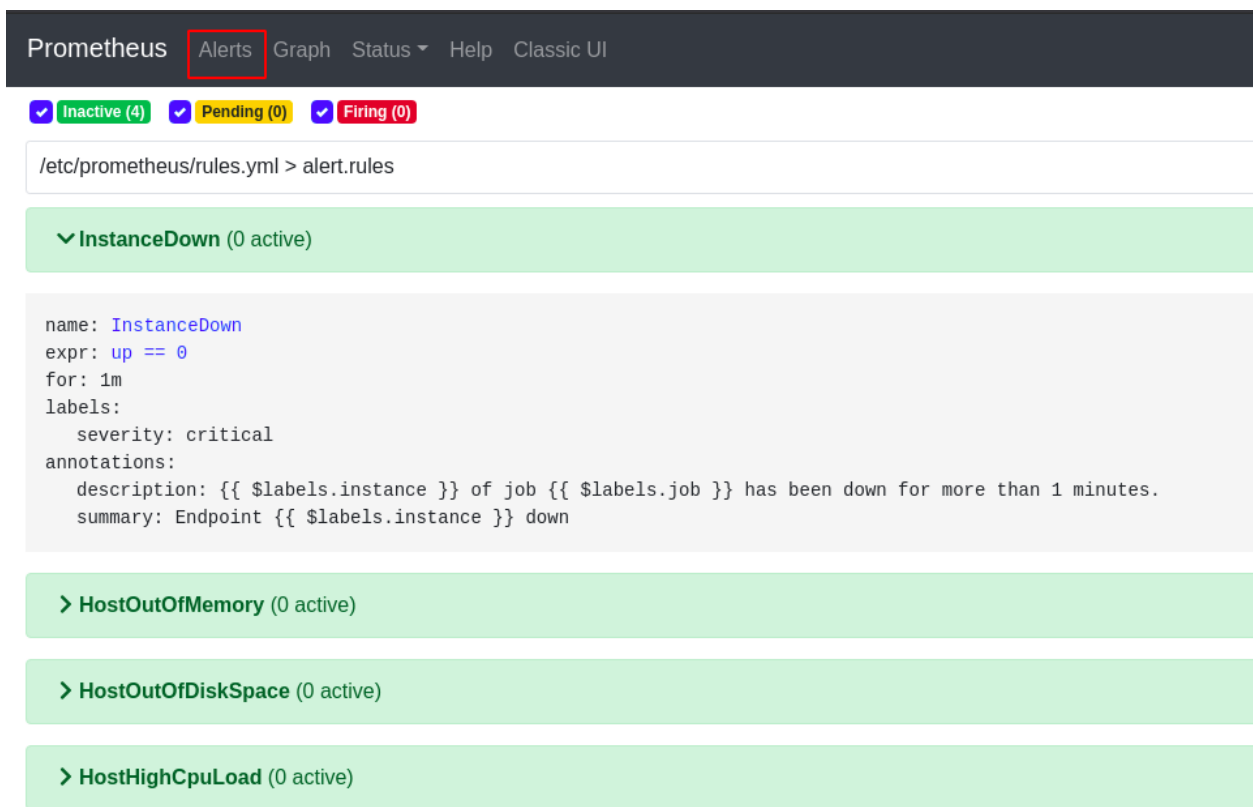
summary: "Host high CPU load (instance {{ \$labels.instance }})"

description: "CPU load is > 80%\n VALUE = {{ \$value }}\n LABELS: {{ \$labels }}"

Restart the Prometheus service to apply these changes.

\$ sudo systemctl restart prometheus.service

Visit port 9090 of the server in a browser (e.g. <http://mysite.com:9090> and click on the 'Alert' button and it will show all the alerts configured in the 'rules.yml' file.



The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus' and 'Alerts' (highlighted with a red box), along with 'Graph', 'Status', 'Help', and 'Classic UI'. Below the navigation bar, there are three status indicators: 'Inactive (4)' (green), 'Pending (0)' (yellow), and 'Firing (0)' (red). A search bar contains the text '/etc/prometheus/rules.yml > alert.rules'. The main content area displays a list of alert rules, each with a green header bar indicating its status and the number of active instances. The first rule, 'InstanceDown', is expanded, showing its configuration details. The other three rules, 'HostOutOfMemory', 'HostOutOfDiskSpace', and 'HostHighCpuLoad', are collapsed.

Prometheus Alerts Graph Status Help Classic UI

✓ Inactive (4) ✓ Pending (0) ✓ Firing (0)

/etc/prometheus/rules.yml > alert.rules

▼ InstanceDown (0 active)

name: InstanceDown  
expr: up == 0  
for: 1m  
labels:  
  severity: critical  
annotations:  
  description: {{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minutes.  
  summary: Endpoint {{ \$labels.instance }} down

> HostOutOfMemory (0 active)

> HostOutOfDiskSpace (0 active)

> HostHighCpuLoad (0 active)

## Configuring Email Notification

Open the alertmanager.yml file and add the email configuration.

```
$ sudo nano /etc/prometheus/alertmanager.yml
```

global:

```
  resolve_timeout: 1m
```

route:

```
  group_by: ['alertname']
```

```
  group_wait: 10s
```

```
  group_interval: 10s
```

```
  repeat_interval: 1h
```

```
  receiver: 'email_notification'
```

receivers:

```
- name: 'email_notification'
```

```
  email_configs:
```

```
    - to: "to"
```

```
      from: "from"
```

```
      smarthost: smtp.gmail.com:587
```

```
      auth_username: "email"
```

```
      auth_identity: "email"
```

```
      auth_password: "password"
```

```
      send_resolved: true
```

inhibit\_rules:

```
- source_match:
```

```
    severity: 'critical'
```

```
  target_match:
```

```
    severity: 'warning'
```

```
  equal: ['alertname', 'dev', 'instance']
```

Restart the alert manager and prometheus services.

```
$ sudo systemctl restart prometheus.service
```

```
$ sudo systemctl restart alertmanager.service
```

Now whenever system resources cross a specific threshold defined in rules.yml file, the status of alerts will change from 'inactive' to 'pending'.

Prometheus Alerts Graph Status ▾ Help Classic UI

✓ Inactive (1)

✓ Pending (3)

✓ Firing (0)

/etc/prometheus/rules.yml > alert.rules

> InstanceDown (0 active)

> HostOutOfMemory (1 active)

> HostOutOfDiskSpace (1 active)

> HostHighCpuLoad (1 active)

And after some time, the status will be in 'Firing' state and an email notification will be sent.

Prometheus Alerts Graph Status ▾ Help Classic UI

✓ Inactive (1)

✓ Pending (0)

✓ Firing (3)

/etc/prometheus/rules.yml > alert.rules

> InstanceDown (0 active)

> HostOutOfMemory (1 active)

> HostOutOfDiskSpace (1 active)

> HostHighCpuLoad (1 active)

## Installing Blackbox for URL Monitoring

Download the alert manager binary file.

```
$ wget
```

```
https://github.com/prometheus/blackbox_exporter/releases/download/v0.19.0/blackbox_exporter-0.19.0.linux-amd64.tar.gz
```

Unzip the downloaded binary file.

```
$ tar -xvzf blackbox_exporter-0.19.0.linux-amd64.tar.gz
```

Go to the alertmanager directory.

```
$ cd blackbox_exporter-0.19.0.linux-amd64
```

Create a config file

```
$ sudo nano monitor_website.yml
```

modules:

```
http_2xx:
```

```
  prober: http
```

```
  timeout: 5s
```

```
  http:
```

```
    valid_http_versions: ["HTTP/1.1", "HTTP/2.0"]
```

```
    valid_status_codes: [] # Defaults to 2xx
```

```
    method: GET
```

Create a service file for the Blackbox exporter

```
$ sudo nano /etc/systemd/system/blackbox.service
```

[Unit]

```
Description=Blackbox Exporter Service
```

```
Wants=network-online.target
```

```
After=network-online.target
```

[Service]

```
Type=simple
```

```
User=root
```

```
ExecStart=/home/ubuntu/blackbox_exporter-0.19.0.linux-amd64/blackbox_exporter \
```

```
--config.file=/home/ubuntu/blackbox_exporter-0.19.0.linux-amd64/monitor_website.yml \
```

```
--web.listen-address=":9115"
```

```
Restart=always
```

[Install]

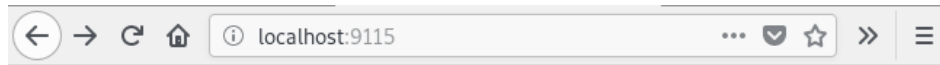
```
WantedBy=multi-user.target
```



Start blackbox service

```
$ sudo systemctl start blackbox.service
```

Open localhost:9115 and check the output as:



# Blackbox Exporter

[Probe prometheus.io for http\\_2xx](#)

[Debug probe prometheus.io for http\\_2xx](#)

[Metrics](#)

[Configuration](#)

## Recent Probes

Module	Target	Result	Debug
--------	--------	--------	-------

Update Prometheus config file

```
$ sudo nano /etc/prometheus/prometheus.yml
```

global:

```
scrape_interval: 15s
```

```
evaluation_interval: 15s
```

scrape\_configs:

```
- job_name: 'prometheus'
```

```
static_configs:
```

```
- targets: ['localhost:9090', 'localhost:9115']
```

```
- job_name: 'blackbox'
```

```
metrics_path: /probe
```

```
params:
```

```
module: [http_prometheus]
```

```
static_configs:
```

```
- targets:
```

```
- https://127.0.0.1:80
```

```
- https://127.0.0.1:81
```

```
relabel_configs:
```

```

- source_labels: [__address__]
  target_label: __param_target
- source_labels: [__param_target]
  target_label: instance
- target_label: __address__
  replacement: 127.0.0.1:9115 # The blackbox exporter's real hostname:port.

```

Restart Prometheus service

\$ sudo systemctl daemon-reload

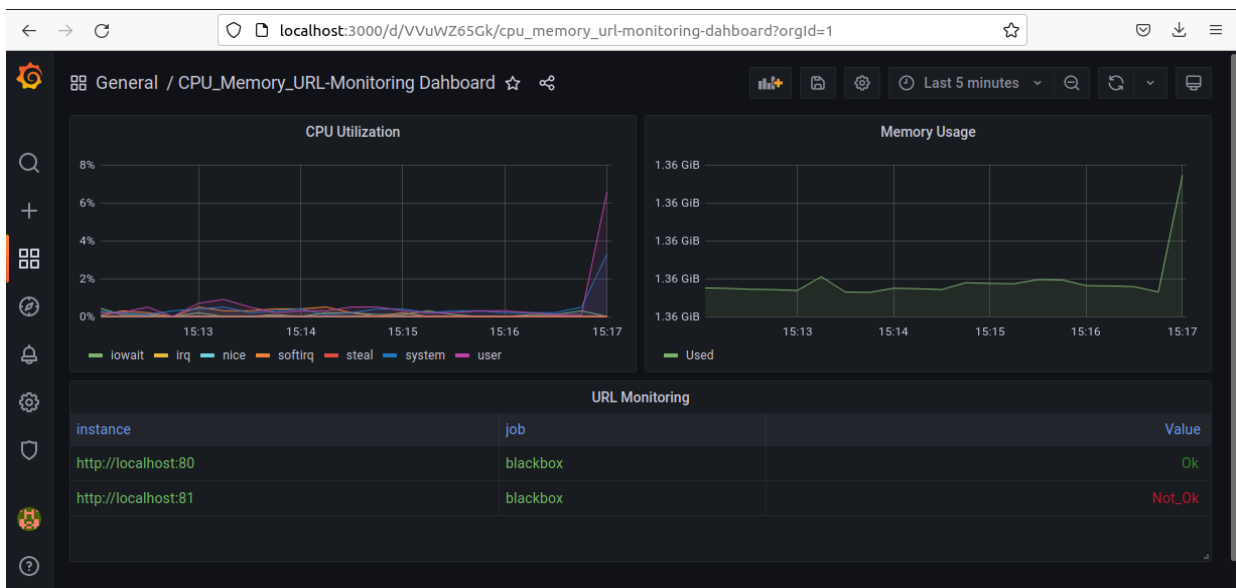
\$ sudo systemctl restart prometheus

Open localhost:9090/targets and check the new target as 9115.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="Prometheus"	7.648s ago	9.000ms	
http://localhost:9115/metrics	UP	instance="localhost:9115" job="System Resources"	2.634s ago	2.598ms	
http://localhost:9100/metrics	UP	instance="localhost:9100" job="System Resources"	1.494s ago	32.098ms	

Goto Grafana and configure new panel as URL monitoring with query as:

Probe\_http\_status\_code



## **Conclusion**

Node exporter, Prometheus and Grafana work together to display all the system metrics on a single dashboard. Prometheus alert manager is a tool used to alert when a specific system metric crosses a threshold specified in rules.yml file. Blackbox exporter is used to collect website related metrics to monitor them.