

Deploy Prometheus and Grafana on a single VM

Abstract

This document provides step-by-step guidance for how to deploy Prometheus and Grafana on a single VM. We will deploy Prometheus main components (e.g. Prometheus, Alertmanager, PushGateway, Node Exporter) using a Ubuntu VM on VMware. And then, we will install and run Grafana, and add Prometheus we deployed before as its datasource. Finally, we'll see a Grafana Dashboard monitoring different metrics like CPU usage, total http requests, and so on.

Environment Set Up

1. Download and install the VMware WorkStation 16.2.3, activate it with security key.
2. Download ubuntu-20.04.4-desktop-amd64.iso file, add it to a new created VM on VMware.
3. Install and open the Ubuntu VM, right click and choose 'Open in Terminal'.
4. In terminal, type in the following command to keep all of your packages up to date and install the vim command:

```
$ sudo apt-get update
$ sudo apt-get install vim
```

Deployment Steps

1. Prometheus Deployment

1.1 Prometheus components Download

Direct to the /opt directory make a /software and /module directory:

```
$ cd /opt
$ sudo mkdir software
$ sudo mkdir modules
```

Download the latest version of the prometheus components (all Linux-amd64.tar.gz) under /opt/software.

For Prometheus:

```
$ cd software
$ sudo wget https://github.com/prometheus/prometheus/releases/download/v2.45.0-rc.0/prometheus-2.45.0-rc.0.linux-amd64.tar.gz
```

For PushGateway:

```
$ sudo wget https://github.com/prometheus/pushgateway/releases/download/v1.4.1/pushgateway-1.4.1.linux-amd64.tar.gz
```

For Node Exporter:

```
$ sudo wget https://github.com/prometheus/node_exporter/releases/download/v1.2.2/node_exporter-1.2.2.linux-amd64.tar.gz
```

*For Alertmanager, it's an important component for alerting, but we'll use Grafana to substitute it (will be shown in later article) because Grafana has a stronger UI and alerting system.

1.2 Decompress packages

Use the following command to decompress the three tar.gz files above to /opt/module directory:

```
$ sudo tar -zxvf prometheus-2.45.0-rc.0.linux-amd64.tar.gz -C /opt/module
$ sudo tar -zxvf pushgateway-1.4.1.linux-amd64.tar.gz -C /opt/module
$ sudo tar -zxvf node_exporter-1.2.2.linux-amd64.tar.gz -C /opt/module
```

1.3 Configuration

This step is only necessary for configuring Prometheus, to add the monitoring functions of PushGateway and Node exporter.

Open /opt/module to see all the decompressed component files, enter the Prometheus directory and open prometheus.yml file to configure.

```
$ cd /opt/module/prometheus-2.45.0-rc.0.linux-amd64
$ sudo vim prometheus.yml
```

Add the following configurations to under the scope_config module of prometheus.yml:

```
- job_name: "pushgateway"
  static_configs:
  - targets: ["localhost:9091"]
    labels:
      instance: pushgateway

- job_name: "node exporter"
  static_configs:
  - targets: ["localhost:9100"]
```

1.4 Run components

1.4.1 Running Prometheus: under the /opt/module/prometheus-2.45.0-rc.0.linux-amd64, create a prometheus.log file, and change the permission of it.

```
$ sudo touch prometheus.log
$ sudo chmod 777 prometheus.log
```

Then run Prometheus using the command:

```
$ sudo nohup ./prometheus --config.file=prometheus.yml > ./prometheus.log 2>&1 &
```

If you see this kind of respond like [1] 11456, and it do not exit when typing any other following command, then prometheus is running successfully.

```
chuyanc@chuyanc-vm:/opt/module/prometheus-2.45.0-rc.0.linux-amd64$ sudo nohup ./prometheus --config.file=prometheus.yml > ./prometheus.log 2>&1 &
[1] 11456
```

1.4.2 Running PushGateway: under the /opt/module/pushgateway-1.4.1.linux-amd64, create a prometheus.log file, and change the permission of it.

```
$ cd /opt/module/pushgateway-1.4.1.linux-amd64
$ sudo touch pushgateway.log
$ sudo chmod 777 pushgateway.log
```

Then run PushGateway using the command:

```
$ sudo nohup ./pushgateway --web.listen-address :9091 > ./pushgateway.log 2>&1 &
```

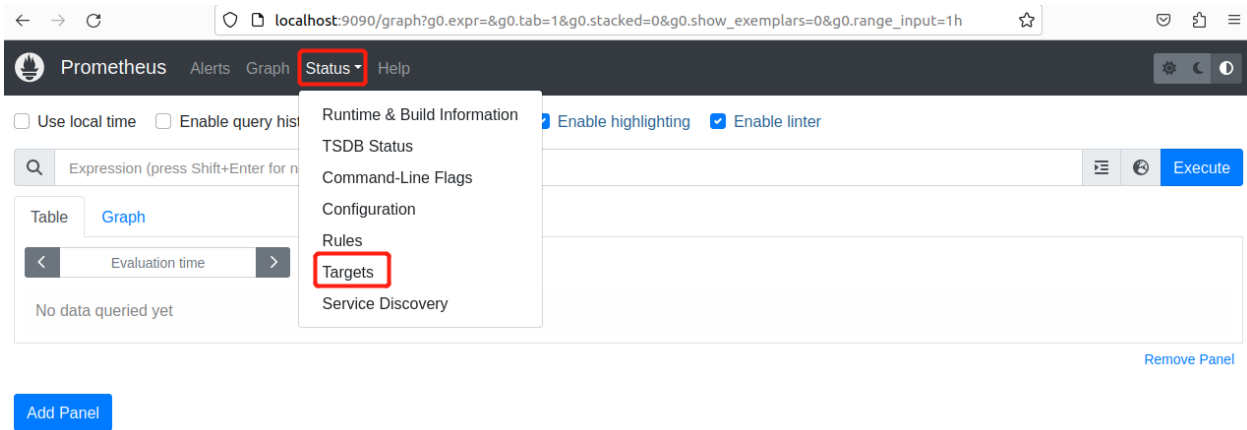
1.4.3 Running the Node Exporter: open the /opt/module/node_exporter-1.2.2.linux-amd64, and simply run it.

```
$ cd /opt/module/node_exporter-1.2.2.linux-amd64
$ ./node_exporter
```


Open the browser and type in: <http://localhost:9100/metrics> we can see the data when monitoring the current machine.

1.4.4




After the previous 2 steps, open the browser and type in: <https://localhost:9090/>, the Prometheus homepage will be shown like this:



And click the Status → Targets, if the three status are up like this, the Prometheus is installed and running.

 Prometheus

AlertsGraphStatus▼Help



Targets

All scrape pools ▼

AllUnhealthyCollapse All

Q

Filter by endpoint or labels

▼

Unknown

▼

Unhealthy

▼

Healthy

node exporter (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	<div>instance="localhost:9100"</div> <div>job="node exporter"</div>	5.522s ago	44.986ms	

prometheus (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	<div>instance="localhost:9090"</div> <div>job="prometheus"</div>	4.666s ago	20.303ms	

pushgateway (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9091/metrics	UP	<div>instance="pushgateway"</div> <div>job="pushgateway"</div>	9.214s ago	4.863ms	

2. Grafana Deployment

2.1 Grafana Download

```
$ cd /opt/software/
$ sudo wget https://dl.grafana.com/enterprise/release/grafana-enterprise-9.5.3.linux-amd64.tar.gz
```

2.2 Decompress packages

Use the following command to decompress the tar.gz file above to /opt/module directory:

```
sudo tar -zxvf grafana-enterprise-9.5.3.linux-amd64.tar.gz -C /opt/module/
```

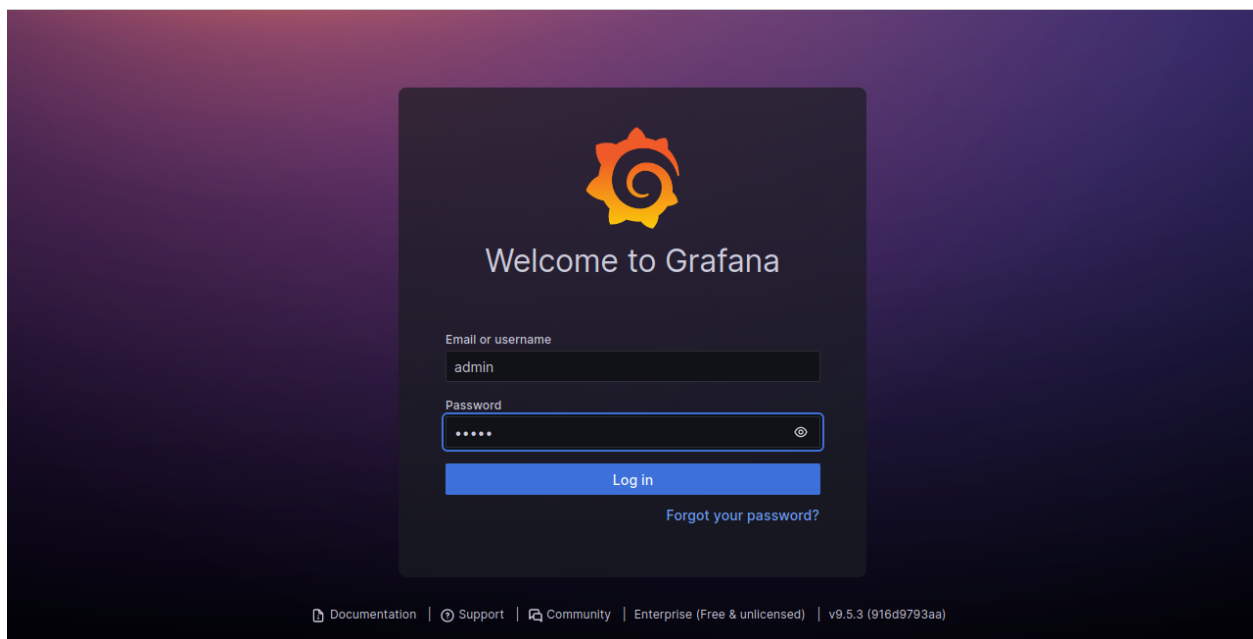
2.3 Run Grafana: under the /opt/module/grafana-9.5.3, create a grafana.log file, and change the permission of it.

```
$ cd /opt/module/grafana-9.5.3  
$ sudo touch grafana.log  
$ sudo chmod 777 grafana.log
```

Then run Grafana using the command:

```
sudo nohup ./bin/grafana-server web > ./grafana.log 2>&1 &
```

Open the browser and type in: <http://localhost:3000/>, we can see the Grafana log in page.

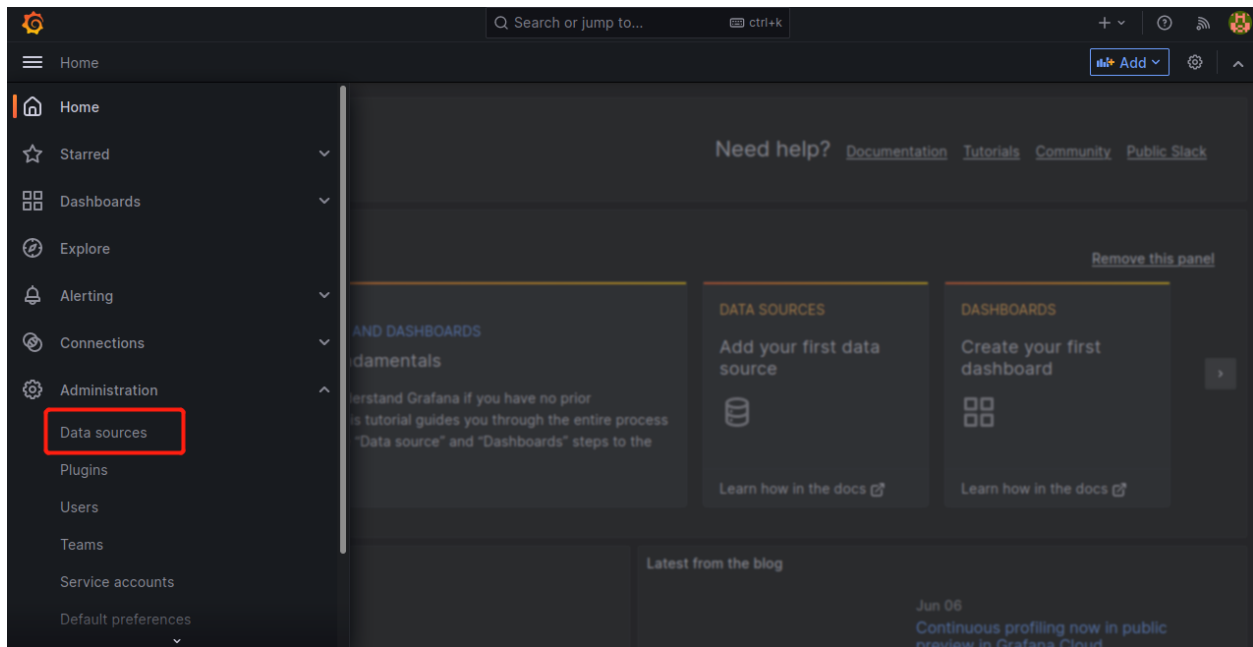


To log in:

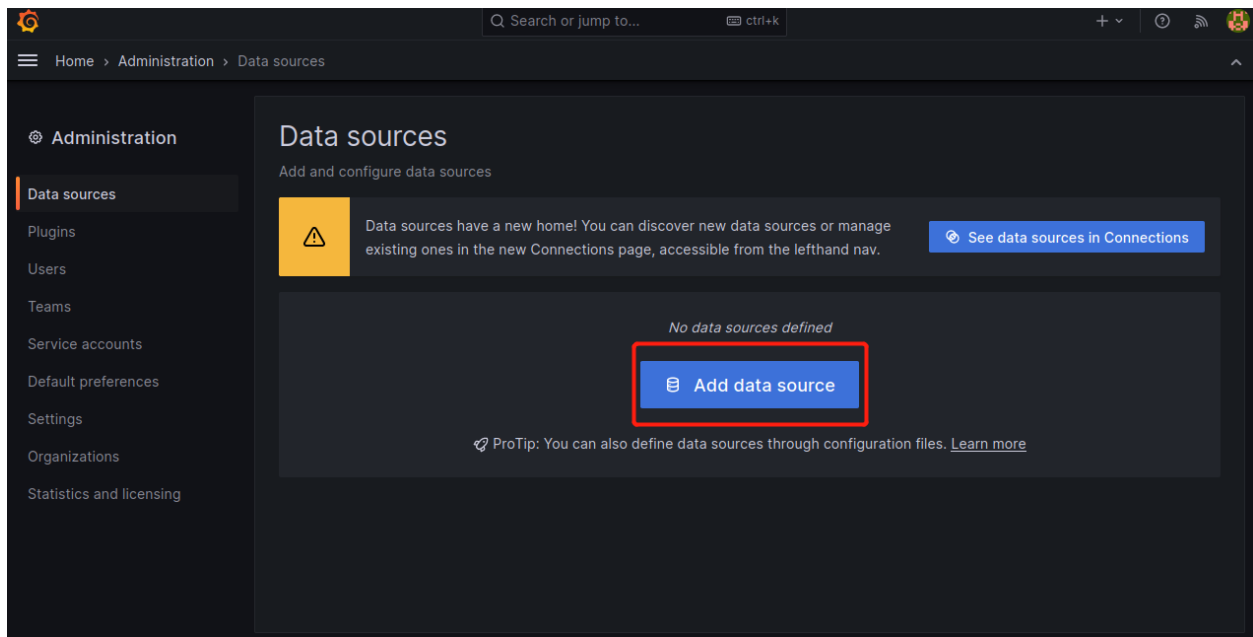
User name: admin Password: admin

2.4 Add Prometheus Data Source

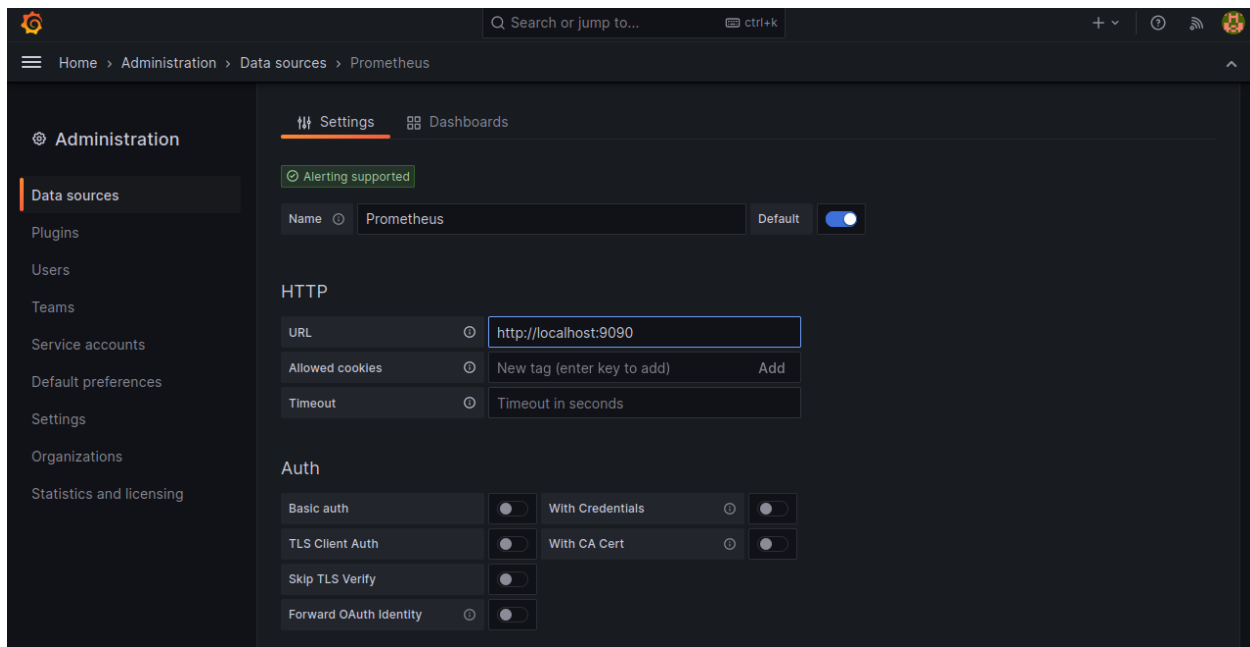
First enter the Homepage and click "Data Sources":



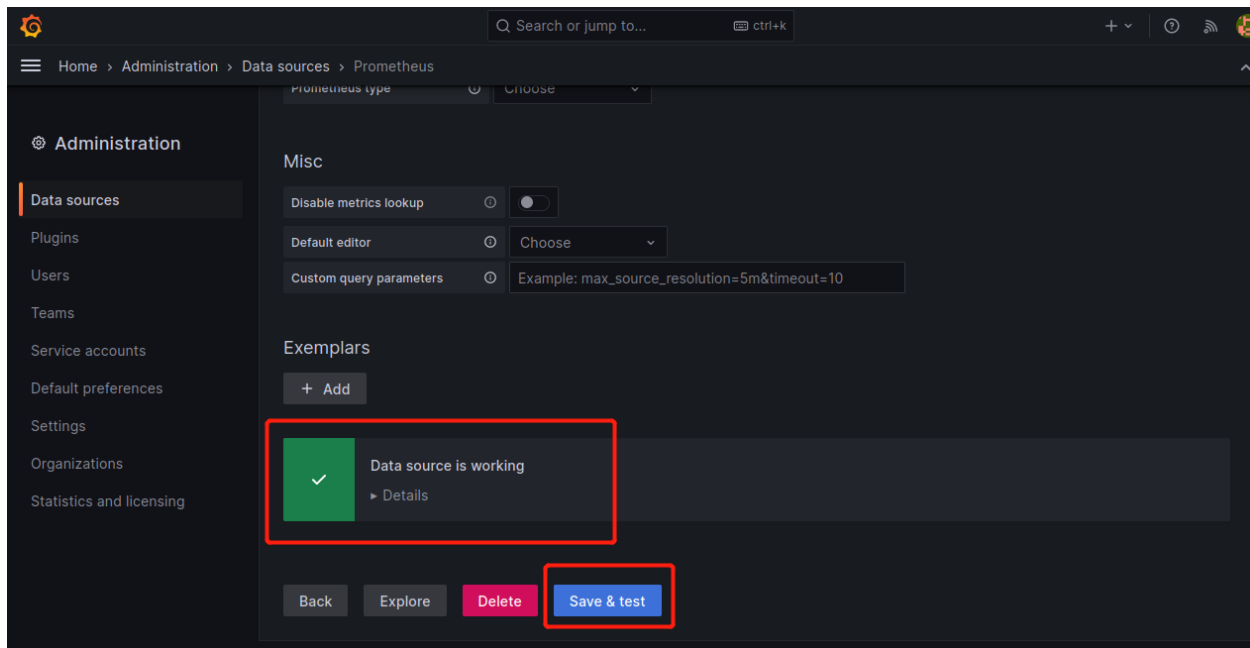
Then click “Add Data Source” and select “Prometheus”:



Then in the Prometheus setting page, enter the data source URL:



Roll to the end and click “Save & Test”:



If the Data source is working, we have successfully connect Prometheus and Grafana for future monitoring works.