

[Open in app](#)[Sign up](#)[Sign in](#)**Medium**

Search



Unveiling the Power Trio: Telegraf, InfluxDB, Grafana for Seamless Ubuntu Monitoring

Rapidcode Technologies · [Follow](#)

8 min read · Aug 16, 2023

 [Listen](#) [Share](#)

Grafana



influxdb

The Images shown are for illustration purposes only

Are you ready to supercharge your system monitoring capabilities? In this comprehensive tutorial, we will guide you through the process of installing and

configuring the TIG Stack (Telegraf, InfluxDB, and Grafana) on an Ubuntu server. With this powerful trio, you'll be able to monitor system loads, including CPU, memory, disk I/O, and even gather custom metrics from other services.

Let's dive in!

A Few Basics First

Before we get started, let's quickly familiarize ourselves with the key components of the TIG Stack:

Telegraf:

Telegraf is an open-source server agent that collects and reports various system and application metrics. It supports a wide range of input plugins that can gather data from sources like system resources (CPU, memory, disk), network devices, databases, and more. Telegraf then processes this data and sends it to various output destinations, including InfluxDB.

InfluxDB:

InfluxDB is a time-series database specifically designed to store and manage time-series data, which is data that is indexed and queried by time. Telegraf sends the collected metrics to InfluxDB, which stores them in a highly efficient manner. InfluxDB provides a query language (InfluxQL) and APIs for retrieving and analyzing time-series data.

Grafana:

Grafana acts as the visualization and dashboarding component of the stack. It connects to InfluxDB (or other data sources) to pull the time-series data and allows users to create rich and interactive dashboards.

Prerequisites

The server should have Root Privileges, Assuming you have three Ubuntu servers set up on your AWS account, let's break down how we'll distribute the TIG Stack components:

- 1. InfluxDB Server:** We'll install InfluxDB on the first server. InfluxDB is where all the metrics will be stored.

2. Telegraf Server:

The second server will be dedicated solely to Telegraf.

Telegraf will collect metrics exclusively from the server where it's installed, and then send them to InfluxDB on the first server for storage.

3. Grafana Server:

Lastly, the third server will host Grafana. This is where we'll visualize the collected metrics and create awesome dashboards.

By strategically placing each component on separate servers, we ensure efficient performance and a streamlined monitoring process.

Let's move forward and set up the TIG Stack step by step!

Step 1: Install InfluxDB

Let's first install the time series database InfluxDB.

Go to the first server and install InfluxDB by using the following commands.

First, go to the root privileges:

```
sudo su
```

For Ubuntu/Debian users, add the InfluxData repository with the following commands:

```
wget -q https://repos.influxdata.com/influxdata-archive_compatible.key  
echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c influxdata' | gpg --dearmor > influxdata-archive_compatible.gpg  
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compatible.gpg] http://repos.influxdata.com/ influxdb'
```

Then, install and start the InfluxDB service:

```
sudo apt-get update && sudo apt-get install influxdb
```

```
sudo service influxdb start
```

Or if your operating system is using systemd (Ubuntu 15.04+, Debian 8+):

```
sudo apt-get update && sudo apt-get install influxdb
sudo systemctl unmask influxdb.service
sudo systemctl start influxdb
```

Configure InfluxDB

a – Create an admin account on your InfluxDB server

To do so, head over to the InfluxDB CLI and type the following commands:

```
$ influx
Connected to http://localhost:8086 version 1.7.7
InfluxDB shell version: 1.7.7

> CREATE USER admin WITH PASSWORD 'password' WITH ALL PRIVILEGES
> SHOW USERS

user    admin
-----
admin  true
```

```
root@ip-172-31-35-188:/home/ubuntu# influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> CREATE USER admin WITH PASSWORD 'password' WITH ALL PRIVILEGES
> SHOW USERS
user    admin
-----
admin  true
```

b – Create a user account for Telegraf

Now that you have an admin account, create an account for Telegraf.

```
> CREATE USER telegraf WITH PASSWORD 'password' WITH ALL PRIVILEGES  
> SHOW USERS
```

```
admin@raspberrypi:~$> CREATE USER telegraf WITH PASSWORD 'password' WITH ALL PRIVILEGES  
admin@raspberrypi:~$> SHOW USERS  
user      admin  
----      ----  
admin      true  
telegraf  true
```

c – Enable HTTP authentication on your InfluxDB server

HTTP authentication needs to be enabled in the InfluxDB configuration file.

Head over to `/etc/influxdb/influxdb.conf` and edit the following lines.

```
[http]  
# Determines whether HTTP endpoint is enabled.  
enabled = true  
  
# Determines whether the Flux query endpoint is enabled.  
# flux-enabled = false  
  
# Determines whether the Flux query logging is enabled.  
# flux-log-enabled = false  
  
# The bind address used by the HTTP service.  
bind-address = ":8086"  
  
# Determines whether user authentication is enabled over HTTP/HTTPS.  
auth-enabled = true
```

save the file and restart the InfluxDB service.

```
sudo systemctl restart influxdb
```

Step 2: Install and configure Telegraf

Go to the second server and log in as the root user or after login as a normal user, you can go root user by using the following command:

```
sudos su
```

You can install Telegraf by using the following command

```
wget -q https://repos.influxdata.com/influxdata-archive_compat.key  
echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c' | influxdb --keyring /etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg  
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg] https://repos.influxdata.com stable main'  
sudo apt-get update && sudo apt-get install telegraf
```

before starting and enabling Telegraf first you have to configure Telegraf to send metrics to InfluxDB.

Configure InfluxDB configuration and HTTP authentication on Telegraf

Now a user account is created in InfluxDB for Telegraf, we are going to make sure that it uses it to write data.

Head over to the configuration file of Telegraf on the second server(where Telegraf is installed), located at `/etc/telegraf/telegraf.conf`.

Modify the following lines :

```
# # Configuration for sending metrics to InfluxDB  
[[outputs.influxdb]]  
#   ## The full HTTP or UDP URL for your InfluxDB instance.  
#   ##  
#   ## Multiple URLs can be specified for a single cluster, only ONE of the  
#   ## urls will be written to each interval.  
#   # urls = ["unix:///var/run/influxdb.sock"]  
#   # urls = ["udp://127.0.0.1:8089"]  
    urls = ["http://172.31.35.188:8086"]  
#  
#   ## The target database for metrics; will be created as needed.  
#   ## For UDP url endpoint database needs to be configured on server side.  
    database = "telegraf"  
#
```

Scroll down and you can find the HTTP Basic Auth section, modify the following line:

```
## Timeout for HTTP messages.  
# timeout = "5s"  
  
## HTTP Basic Auth  
username = "telegraf"  
password = "password"  
  
## HTTP User-Agent  
# user_agent = "telegraf"
```

save the file and type the following command:

Start and enable service

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

Now Telegraf collect metrics and send them to InfluxDB, you can go to the InfluxDB server and ensure that the database is created or not by following commands:

```
$influx -username telegraf -password password  
  
Connected to http://localhost:8086 version 1.8.10  
InfluxDB shell version: 1.8.10  
> show databases;  
name: databases  
name  
----  
_internal  
telegraf  
> use telegraf  
Using database telegraf  
  
> show measurements;  
name: measurements  
name  
----
```

```
cpu  
disk  
diskio  
kernel  
mem  
processes  
swap  
system  
>
```

You can see the above snap, Telegraf service successfully created a database named “*telegraf*” and stored measurements in it.

Note: If the database is not present that you have to check the security group to add port 8086 for InfluxDB so that Telegraf service can send data to InfluxDB on port 8086

Now its time to visualize measurements on the beautiful dashboard provided by Grafana:

Step 3: Install Grafana

Go to the third server and log in as the root user or after login as a normal user, you can go root user by using the following command:

```
sudos su
```

Complete the following steps to install Grafana from the APT repository:

1. To install the required packages and download the Grafana repository signing key, run the following commands:

```
sudo apt-get install -y apt-transport-https  
sudo apt-get install -y software-properties-common wget  
sudo wget -q -O /usr/share/keyrings/grafana.key https://apt.grafana.com/gpg.i
```

2. To add a repository for stable releases, run the following command:

```
echo "deb [signed-by=/usr/share/keyrings/grafana.key] https://apt.grafana.co
```

3. Run the following command to update the list of available packages:

```
sudo apt-get update
```

4. To install Grafana, run the following command:

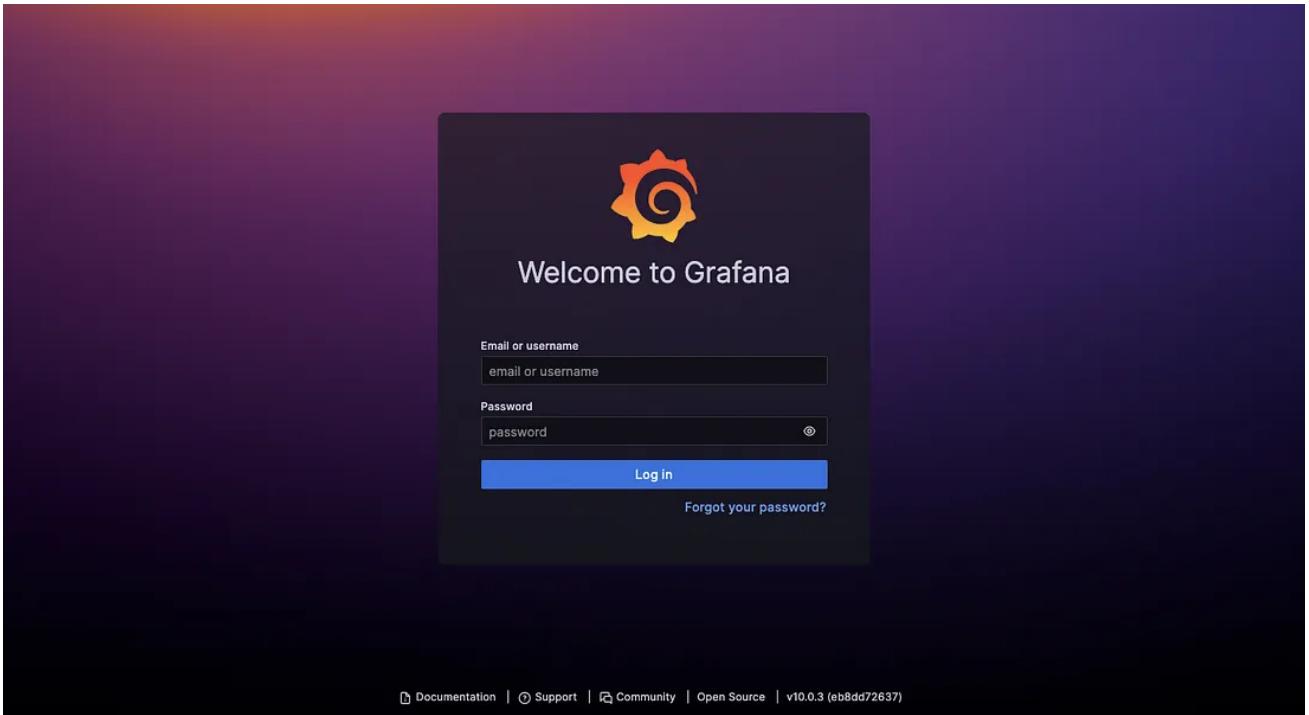
```
# Installs the latest OSS release:  
sudo apt-get install grafana
```

5. And start and enable service:

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

Now hop over to the browser and type the public IP of the third(Grafna) server with port number 3000.

You should have to see the following snap:



Note: If the above snap is not loaded so you have to check port is configured or not in the security group and port is 3000

Type the default username “admin” and password “admin” and it will ask new password if you want to configure it then go ahead and type new password otherwise skip the step and then you will see the following snap:

A screenshot of the Grafana dashboard. The top navigation bar includes a gear icon, a search bar with placeholder "Q. Search or jump to...", a "cmd+k" keyboard shortcut, and various user icons. Below the header, there's a "Welcome to Grafana" message and a "Need help?" section with links to "Documentation", "Tutorials", "Community", and "Public Slack". The main content area is divided into several panels: a "Basic" panel with instructions for setting up Grafana, a "TUTORIAL DATA SOURCE AND DASHBOARDS" panel with a "Grafana fundamentals" section and a "DATA SOURCES" panel with a "Add your first data source" button; a "DASHBOARDS" panel with a "Create your first dashboard" button; and a "Dashboards" sidebar with sections for "Starred dashboards" and "Recently viewed dashboards". At the bottom, there's a "Latest from the blog" section featuring a post titled "How to monitor pool water levels from anywhere with Grafana" dated "Aug 14", showing a dashboard visualization, and a "359" comment count.

Above snap you can see the data source click on it and type the data source as InfluxDB and click on it, and you should see the following snap :

The screenshot shows the Grafana Administration interface with the following details:

- Left sidebar:** Shows the navigation path: Home > Administration > Data sources > InfluxDB.
- Top bar:** Includes a search bar ("Search or jump to..."), a command key ("cmd+k"), and various system icons.
- Main Content Area:**
 - InfluxDB Configuration:** Type: InfluxDB. Status: Alerting supported.
 - General Settings:** Name: InfluxDB, Default: Enabled.
 - Query Language:** Set to InfluxQL.
 - HTTP Settings:** URL: http://172.31.35.188:8086, Allowed cookies: New tag (enter key to add), Timeout: Timeout in seconds.
 - Auth Settings:** Basic auth: Enabled, With Credentials: Enabled. TLS Client Auth: Enabled, With CA Cert: Enabled.

And configure database, URL, and HTTP authentication:

HTTP

URL	<input type="text"/> http://172.31.35.188:8086
Allowed cookies	<input type="text"/> New tag (enter key to add) <input type="button" value="Add"/>
Timeout	<input type="text"/> Timeout in seconds

Auth

Basic auth	<input checked="" type="checkbox"/>	With Credentials <input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/>	With CA Cert <input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>	
Forward OAuth Identity	<input type="checkbox"/>	

Basic Auth Details

User	admin
Password	*****

InfluxDB Details

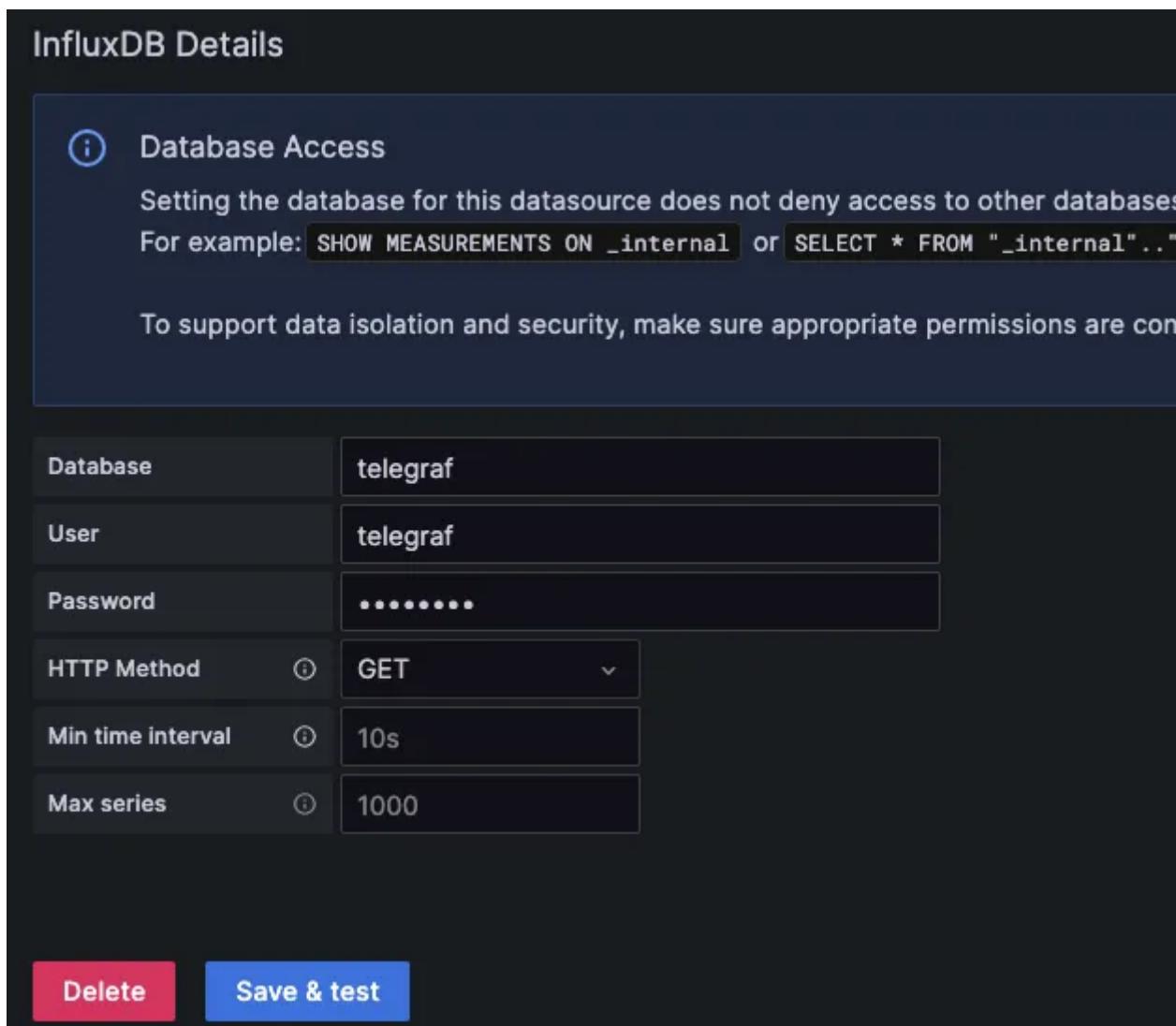
Database Access

Setting the database for this datasource does not deny access to other databases
For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal"...`

To support data isolation and security, make sure appropriate permissions are con

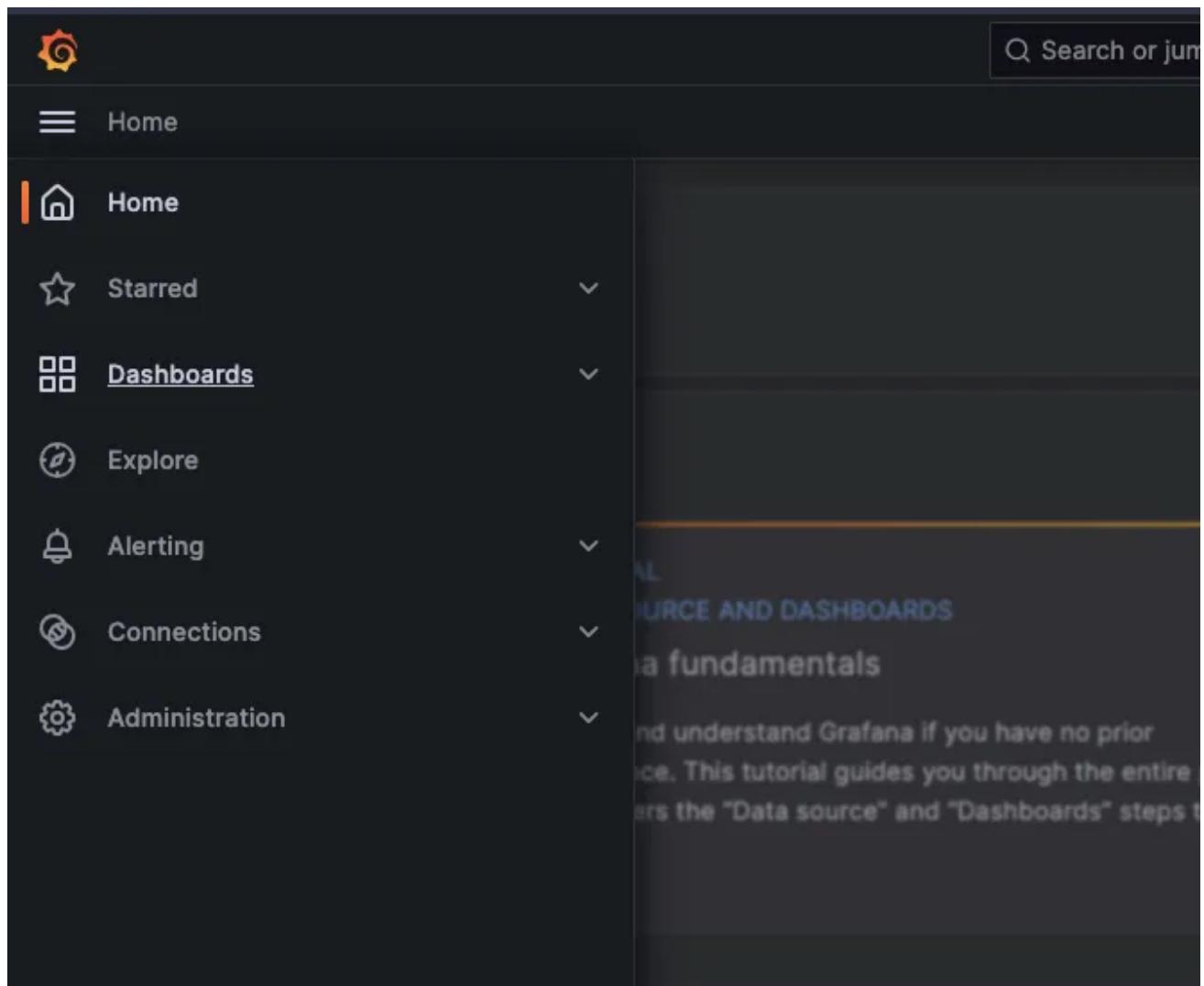
Database	telegraf
User	telegraf
Password	*****
HTTP Method	GET
Min time interval	10s
Max series	1000

Delete **Save & test**

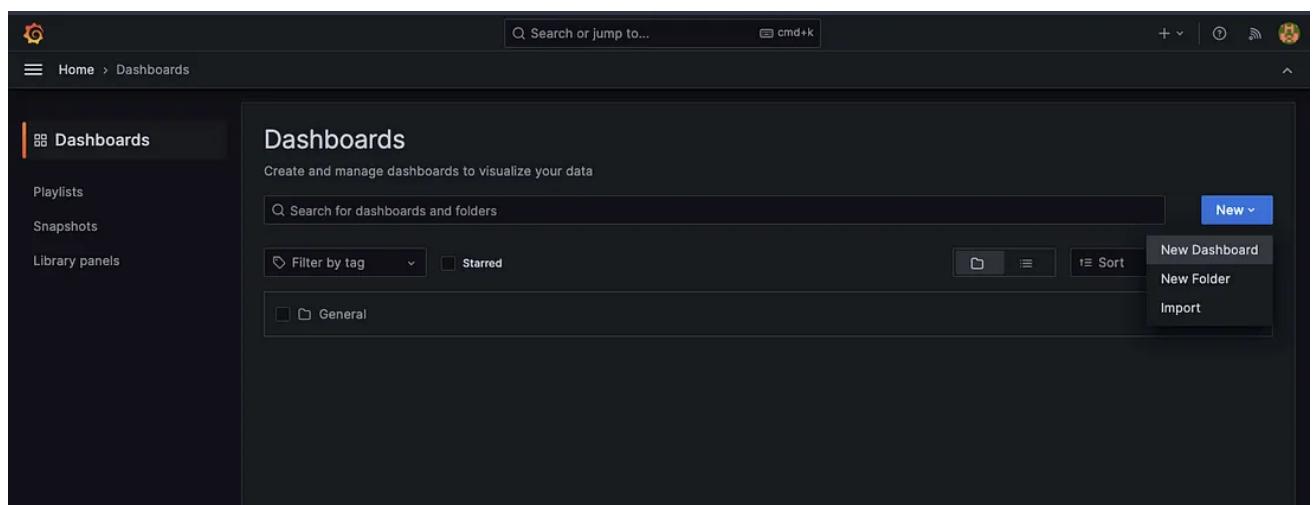


Finally, click on save and test.

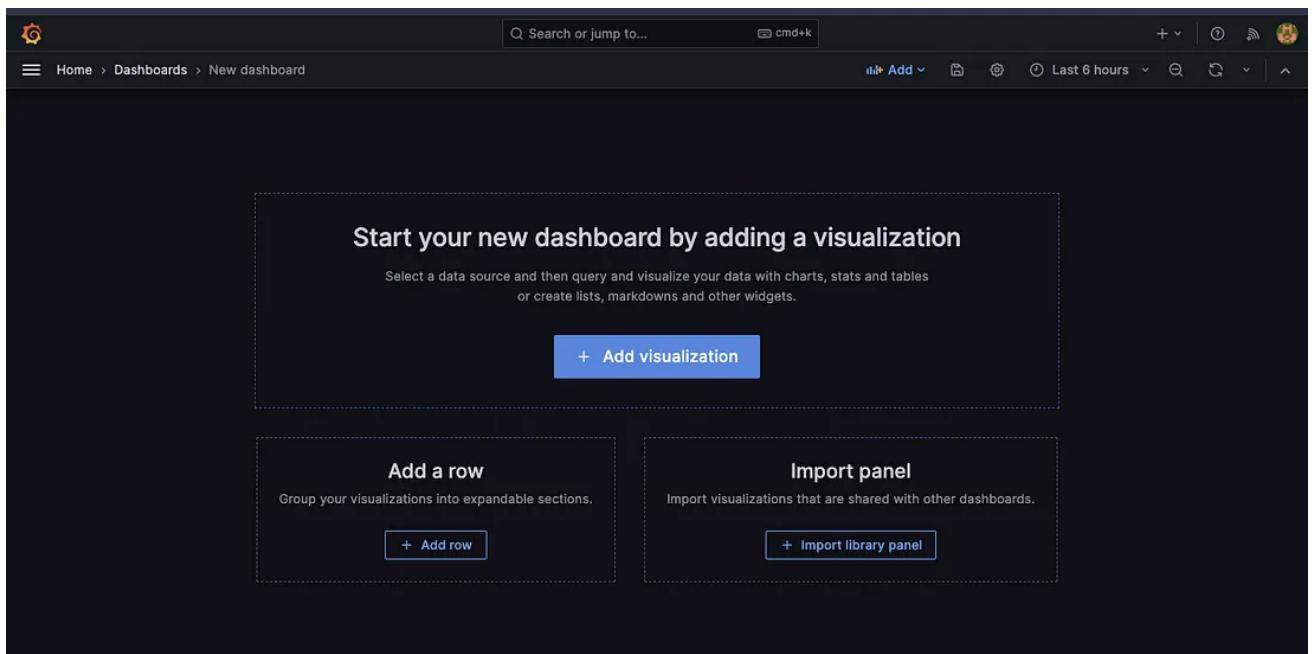
Now click on Dashboard



Click on New Dashboard



Click on Add Visualization and start creating a dashboard:



Now in above snap click on

select measurements->CPU .

field(value)->usage_user.

And click on '+' sign in select row and click selector -> percentile

And on the right side:

the panel option change the Dashboard name “CPU Utilization”.

And in Graph style-> set line width “1.5” and opacity “25” and Connect null values “Always”

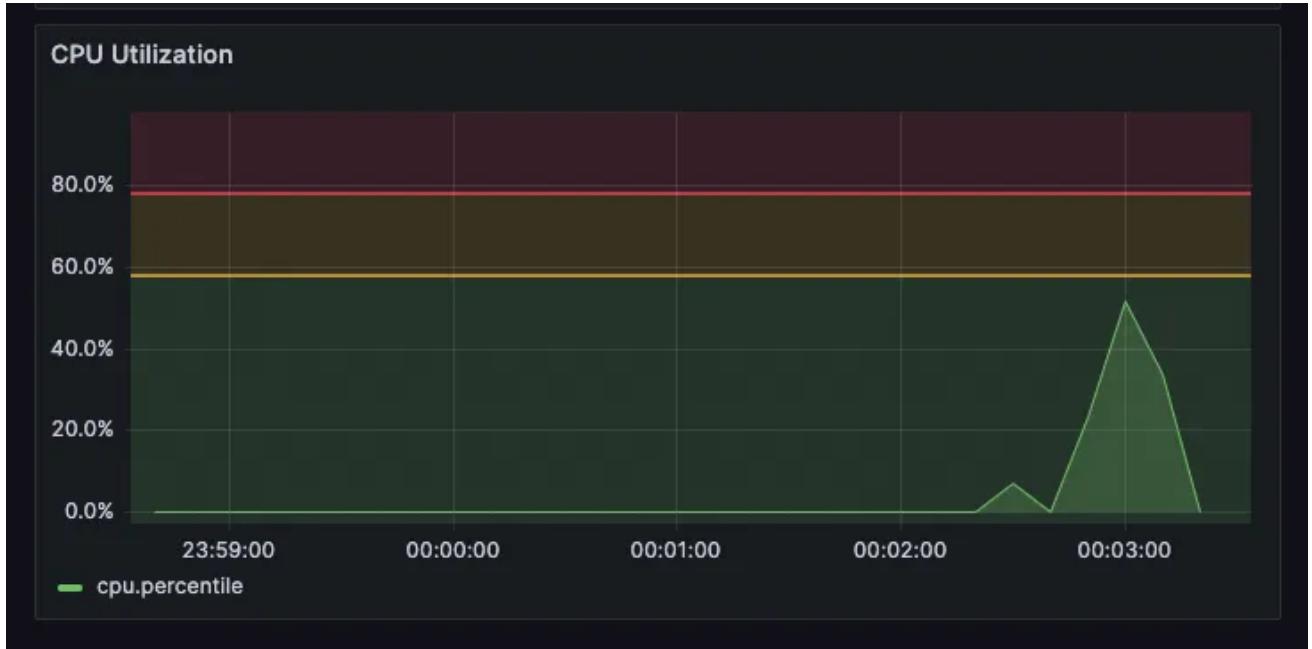
And in the standard option select Unit-> Misc -> Percent(0-100) and min 0 and max 100 and decimal 1.

And select Thresholds Red for 80% and Yellow for 60% .

And select show thresholds as filled regions and lines.

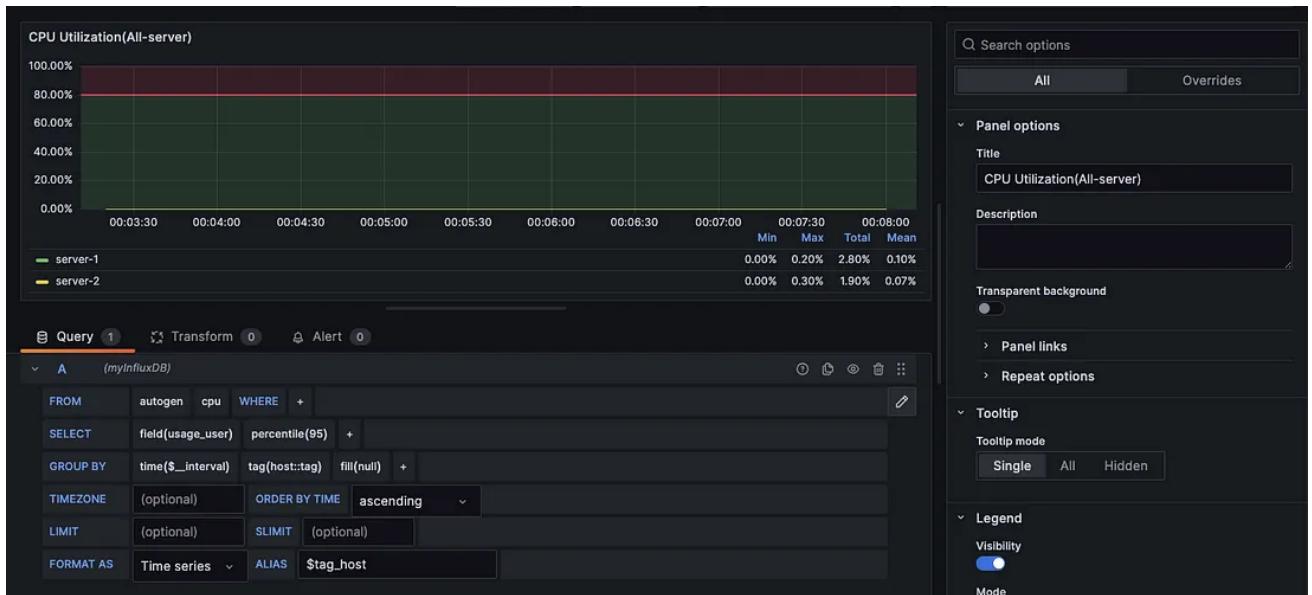
Finally click on the top bar on data update last 15 min and click on apply.

Now you could see your CPU utilization panel on the dashboard

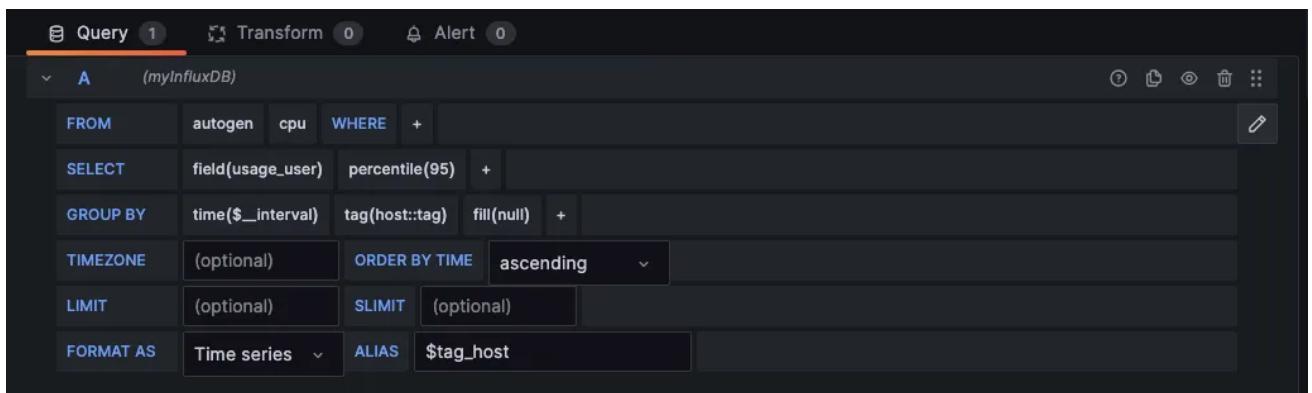


Same you could create for Memory, Process, etc.

For multiple servers, you have to install and configure Telegraf by following the same steps as we did in the Telegraf sections to visualize on Grafana you could change the following query, and the rest of the settings will same as above :

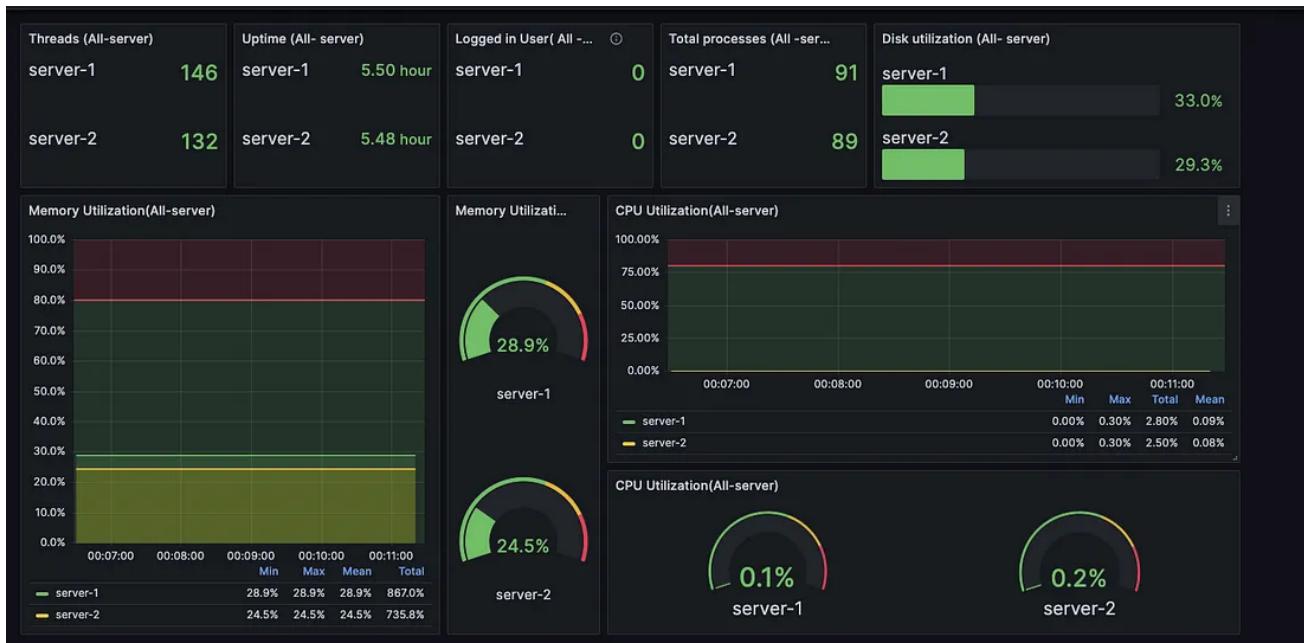


You have to focus on the following part only:



Like above you could create all other panels such as Memory, thread, Disk utilization, Processes, and User logged-in, etc.

After that, your Dashboard should look like the following snap:



Note:- For testing, you could run the following command on servers and visualize the things on the Dashboard:

```
apt install stress-ng
stress-ng --cpu 4 --io 2 --vm 2 --vm-bytes 1G --timeout 60s --fallocate 4 --
```

After applying the above commands you could see the following changing Dashboard.



In summary, the combination of Telegraf, InfluxDB, and Grafana provides a complete solution for collecting, storing, querying, and visualizing time-series data. Telegraf collects metrics from various sources, InfluxDB stores the metrics efficiently, and Grafana enables users to create informative and interactive dashboards for data analysis and monitoring purposes. This stack is commonly used in monitoring infrastructure, applications, and other systems to gain insights into performance and behavior.

[Monitoring](#)[Influxdb](#)[Telegraf](#)[Grafana](#)[DevOps](#)[Follow](#)

Written by Rapidcode Technologies

314 Followers · 1 Following

Architecting the future of innovation and design with cloud-native skills. ⭐ Let's transform your business!

🌟 #Innovation #Perseverance



Responses (1)

What are your thoughts?

Respond



Mano Atech

Mar 5, 2024

...

always i following your article and it's more useful

for my carrier, could you please provided document docker container monitoring same like this article.



[Reply](#)

More from Rapidcode Technologies



docker/compose

Define and run multi-container applications with



 Rapidcode Technologies

Deploying a PHP Web App with Docker Compose, Nginx, and MariaDB

Introduction: In this tutorial, we will walk you through the process of deploying a PHP web application using Docker Compose, Nginx as the...

Sep 23, 2023  169  4

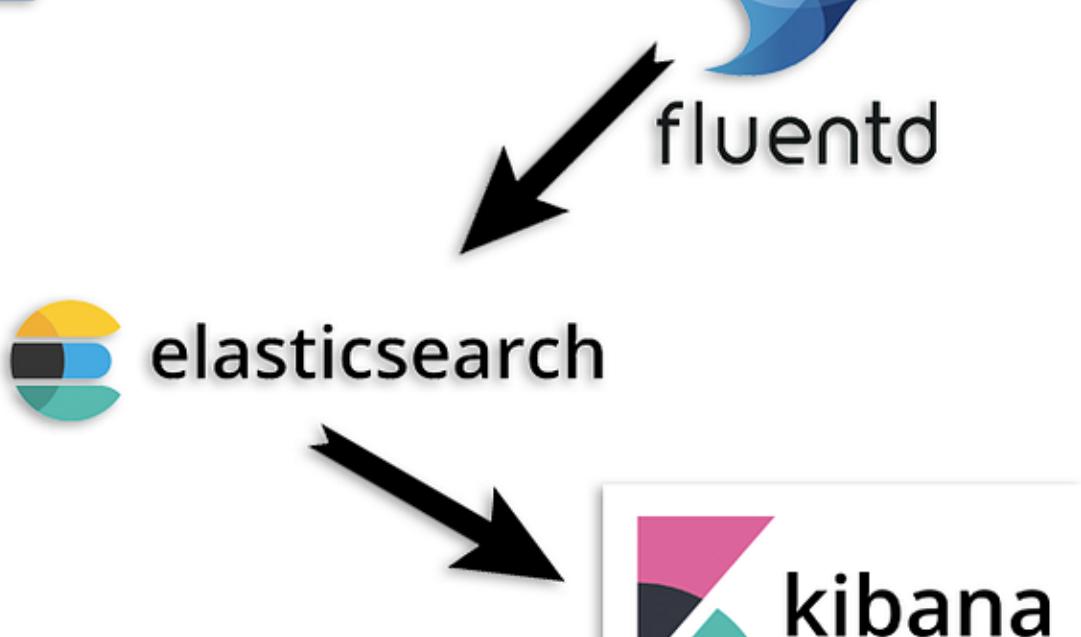
 Rapidcode Technologies

Step-by-Step Guide: Creating an EKS Cluster with Terraform Resources, IAM Roles for Service...

Create AWS VPC

Jun 7, 2023  58  6



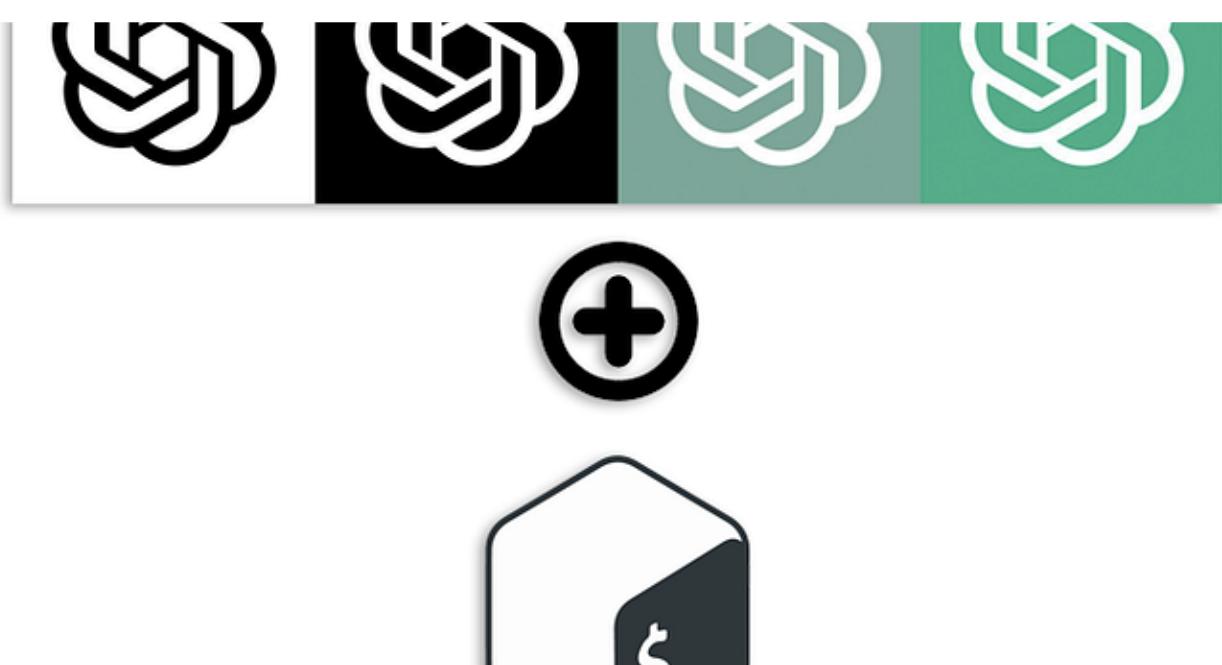


 Rapidcode Technologies

Simplifying Kubernetes Logging with EFK Stack

Logging is critical in application development. We often add log statements to our code for future reference, especially when...

Nov 8, 2023  259  4





Rapidcode Technologies

How to Run ChatGPT using ShellGPT From the terminal

If you want to run ChatGPT in the terminal, you can pass text commands to ChatGPT using the Python program ShellGPT, which will then...

[See all from Rapidcode Technologies](#)

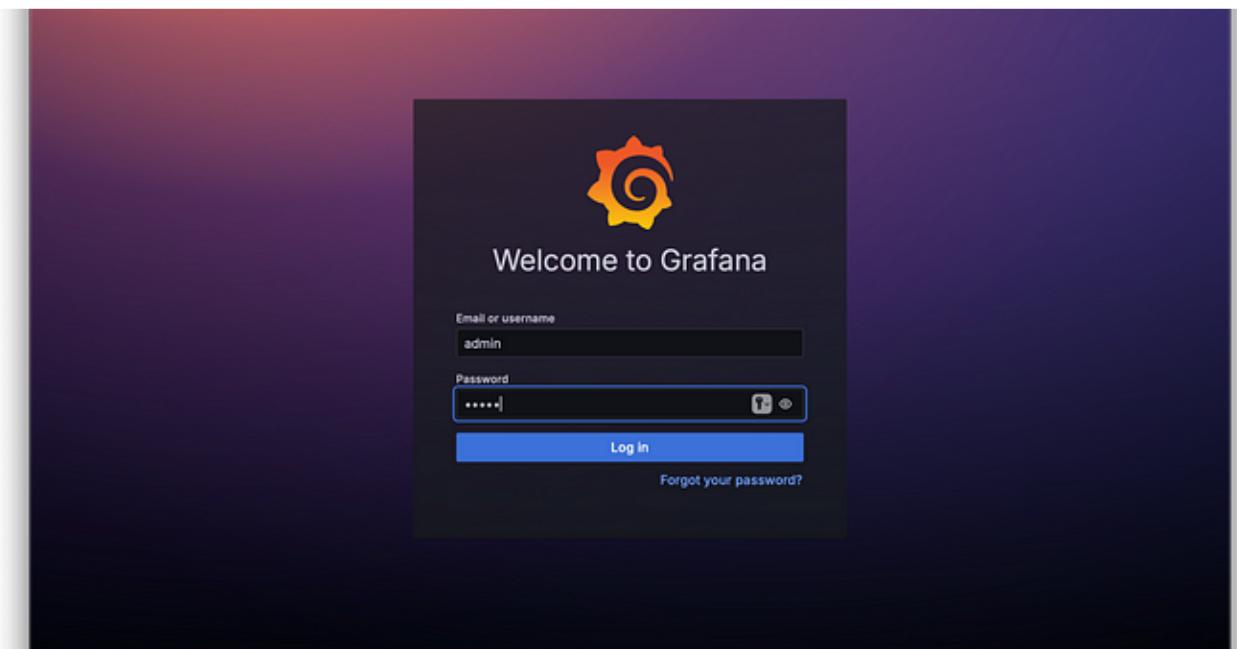
May 24, 2023

35

4



Recommended from Medium



In DevOps.dev by alfathmad

Deploying Monitoring Tools: Prometheus, Alertmanager, Grafana, Node Exporter, and Uptime Kuma with...

In this guide, we will explore how to deploy a complete monitoring stack using Docker Compose. The tools we will set up are Prometheus...



Sep 27, 2024

63



 DevOpsDynamo

Ultimate Guide to Monitoring with Prometheus, Grafana, and Loki on Kubernetes: Ensuring Data...

I've been in Kubernetes setups for a long time, and I've seen the configuration of the monitoring system in many different ways. One of...

Nov 9, 2024

 33

Lists



General Coding Knowledge

20 stories · 1914 saves



Productivity

244 stories · 684 saves



Natural Language Processing

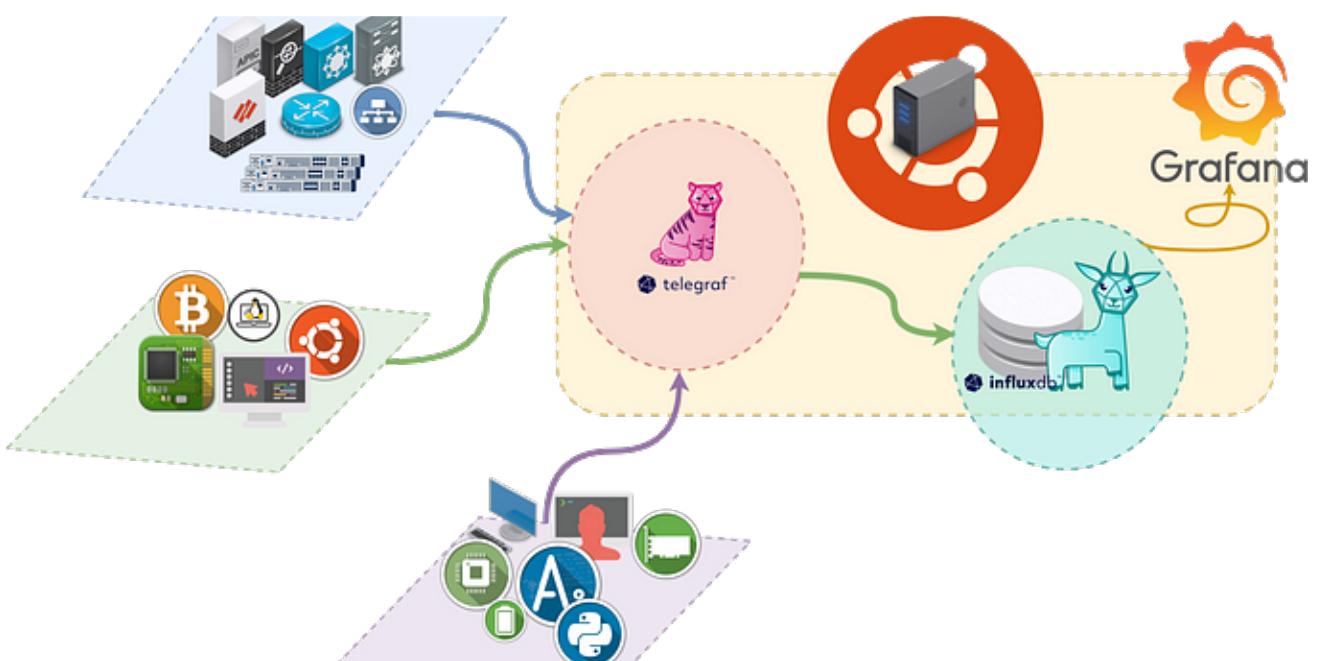
1939 stories · 1591 saves



 In DevPulse by Rsprasangi

Mastering Prometheus and Grafana: A Crucial DevOps Skillset

◆ Sep 19, 2024  1



 r00tb33r

InfluxDB, Telegraf and Grafana

on Linux server

◆ Aug 27, 2024





 Cumhur Akkaya

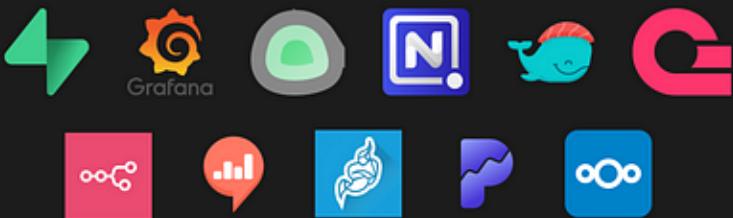
DevOps and Cloud Resources (PDF Books)

In this article;  You may find the pdf links of the resources about DevOps Tools and Cloud Systems. There are 452 pieces in total, for...

⭐ Aug 30, 2024 ⚡ 134 🗣 1



11 Open-Source SaaS Killer Selfhost With Docker



@harendravarma2



@harendra21



@harendra21



In Dev Genius by Harendra

11 Open-Source SaaS Killer—Selfhost With Docker

Selfhost Supabase, Grafana, Uptime Kuma, NocoDB, Dokku, Appwrite, N8N, Redash, Jitsi, Plausible and Nextcloud with docker



Sep 9, 2024

4.1K

27

[See more recommendations](#)