Name: Venkata Lahari Goruputi

Role: DevOps Intern

Task: Prometheus and Grafana

Installing Prometheus and Grafana in EC2 Instance

✓ Prometheus and Grafana are most popular open-source tools for monitoring especially for Kubernetes. Prometheus is the logs and metrics collector and Grafana will give us dashboards to visualize the details collected by Prometheus in the form graphs, tables and charts etc.

What is Prometheus:

➢ Prometheus and Grafana are most popular open-source tools for monitoring especially for Kubernetes. Prometheus is the logs and metrics collector and Grafana will give us dashboards to visualize the details collected by Prometheus in the form graphs, tables and charts etc.

Know More About Prometheus

➢ **Key Features of Prometheus:**

- **Multi-dimensional Data Model:**
  - Metrics are stored with labels, allowing for flexible querying and aggregation.
- **PromQL:**
  - Prometheus Query Language enables powerful querying of metrics.
- **Service Discovery:**
  - Built-in support for discovering targets dynamically, including integration with Kubernetes and other service discovery mechanisms.
- **Alerting:**
  - Supports alerting based on predefined rules and conditions.
- **Scalability:**
  - Designed to be horizontally scalable to handle large-scale deployments.

What is Grafana:

➢ Grafana is an open-source platform for data visualization and monitoring. It allows users to create, explore, and share dashboards with data from various sources, including Prometheus.

Know More About Grafana

Know More About Grafana Dashboards

- **Key Features of Grafana:**
  - **Visualization:**
    - Rich visualizations including graphs, charts, and tables.
  - **Dashboarding:**
    - Create and customize dashboards to display metrics and logs.
  - **Alerting:**
    - Define alert rules and receive notifications based on data thresholds.
  - **Data Source Integration:**
    - Grafana supports integration with numerous data sources, including Prometheus, to visualize and analyse metrics and logs.
  - **Plugins and Extensions:**
    - Extensible with plugins and extensions for additional features and integrations.

Integration and Use:

✓ Prometheus and Grafana Integration:

- ✓ Prometheus metrics can be easily visualized and analysed in Grafana dashboards. Grafana supports Prometheus as a data source, allowing users to query Prometheus metrics using PromQL and create rich visualizations.
- ✓ **Common Use Cases:**

  - Together, Prometheus and Grafana are commonly used for monitoring cloud-native applications, microservices architectures, containerized environments (like Kubernetes), and traditional infrastructure. They provide deep insights into system performance, resource utilization, and application health.

✓ Installing Prometheus:

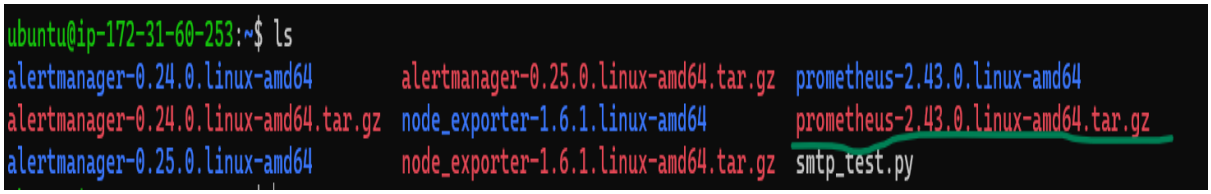
- **Prerequisites:**
  - **EC2 Instance:**
    - Ensure you have an EC2 instance running with appropriate permissions to install software.
  - **SSH Access:**
    - Connect to your EC2 instance via SSH.
    - Logged into ubuntu server via ssh using **[ssh <server-username>@<publicIP-of-the-server>].**
  - **Steps:**
    - After Login to the server, we need to update the packages using **[sudo apt update command].**
    - Then we need to download the stable or latest version of Prometheus from the official page of Prometheus.

Download Prometheus

- **Downloading Prometheus:**
  - Use **[wget]** command to download the tar file use the below links to download the Prometheus tar file.
  - wget https://github.com/prometheus/prometheus/releases/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz

Download Prometheus tar file which I have used

- The above command will download the Prometheus tar file in the home path of the server.



✓ Extracting the downloaded Prometheus tar file:

- After Downloading the tar file, we need to extract the tar file to get all the binaries required to install the Prometheus.
- Use the following command to extract the tar file

[sudo tar xvfz prometheus-2.43.0.linux-amd64.tar.gz]

- The above command will extract the tar file in the home path, after the extraction is done, we will get a folder named **prometheus-2.43.0.linux-amd64** which contains all the binaries and dependencies required to install the Prometheus.

✓ Installation:

- As part of Installation process, we need to move the Prometheus related filed and binaries to the appropriate directories in-order run the Prometheus.
- Follow the below steps for the Installation

```
sudo mv prometheus-2.43.0.linux-amd64/prometheus /usr/local/bin/

sudo mv prometheus-2.43.0.linux-amd64/promtool /usr/local/bin/

sudo mkdir /etc/prometheus

sudo mv prometheus-2.43.0.linux-amd64/consoles /etc/prometheus

sudo mv prometheus-2.43.0.linux-amd64/console_libraries /etc/prometheus
```

- The above commands will move the files to bin folder/PATH variable to let the **PATH** variable know the default location of the Prometheus.
- Because Executables are accessible system-wide by placing them in **/usr/local/bin/.**
- The consoles and **console_libraries** directories contain web console templates and libraries that Prometheus uses for its web UI.

- Configuration and related files are organized in **/etc/prometheus/** for easy management and adherence to standard Linux filesystem hierarchy conventions.
- Prometheus has access to necessary resources like console templates and its configuration file.

```
ubuntu@ip-172-31-60-253:~$ ls -al /usr/local/bin/
total 306252
drwxr-xr-x  2 root         root              4096 Jun 26 04:37 .
drwxr-xr-x 10 root         root              4096 Apr 11 02:07 ..
-rwxr-xr-x  1 alertmanager alertmanager  34546840 Dec 22  2022 alertmanager
-rwxr-xr-x  1 alertmanager alertmanager  27906085 Dec 22  2022 amtool
-rwxr-xr-x  1 ubuntu       ubuntu        20025119 Jul 17  2023 node_exporter
-rwxr-xr-x  1 ubuntu       ubuntu       119498478 Mar 21  2023 prometheus
-rwxr-xr-x  1 ubuntu       ubuntu       111606794 Mar 21  2023 promtool
```

```
ubuntu@ip-172-31-60-253:~$ cd /etc/prometheus/
ubuntu@ip-172-31-60-253:/etc/prometheus$ ls
alert_rules.yml  prometheus.yml
```

✓ Creating User for Prometheus:

- We need to create a dedicated user for Prometheus so that service follows best practices for Unix/Linux system administration. It helps in tracking and managing resources used by specific services and simplifies troubleshooting.
- So, we need create a user and change the ownership and permissions for the user accordingly.

```
sudo useradd --no-create-home --shell /bin/false prometheus

sudo chown -R prometheus:prometheus /etc/prometheus

sudo chown prometheus:prometheus /usr/local/bin/prometheus
```

- The above steps will creates a user names prometheus and gives the prometheus user ownership for the directories which were required.

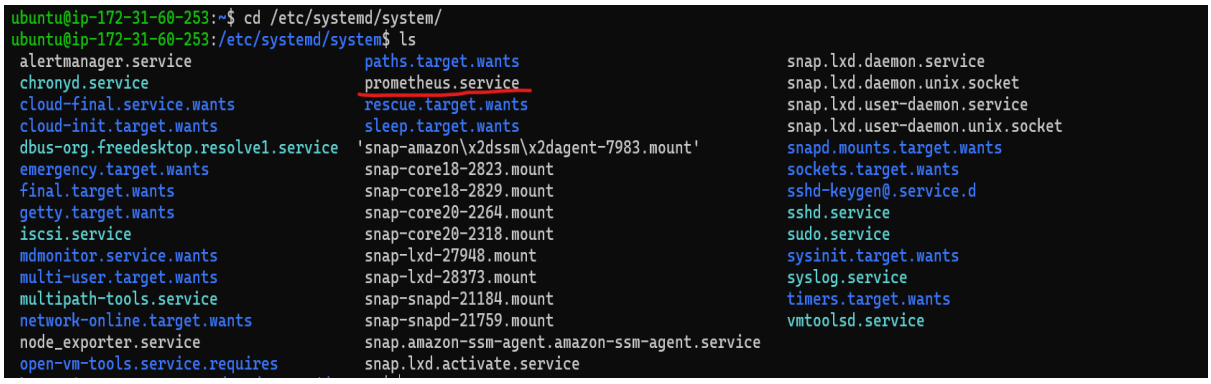
```
ubuntu@ip-172-31-60-253:~$ cat /etc/passwd | grep prome
prometheus:x:1001:1001::/home/prometheus:/bin/false
ubuntu@ip-172-31-60-253:~$
```

✓ Create Prometheus Service:

- Now we need to create the service for Prometheus using **[sudo nano /etc/systemd/system/prometheus.service]** (or) **[sudo vi /etc/systemd/system/prometheus.service]**
- Creating a new systemd service file for Prometheus is essential for managing Prometheus as a background service on a Linux system.
- The above command will create a service file for prometheus in **/etc/systemd/system** path with the service name as **prometheus**, then, we need to

```
[Unit]
Description=Prometheus Monitoring
Wants=network-online.target
After=network-online.target

[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

[Install]
WantedBy=multi-user.target
```

add the above content and save the file using **:wq!** in vi editor and **ctrl+X and Y** in nano editor, this service file contains the location of prometheus and other details.



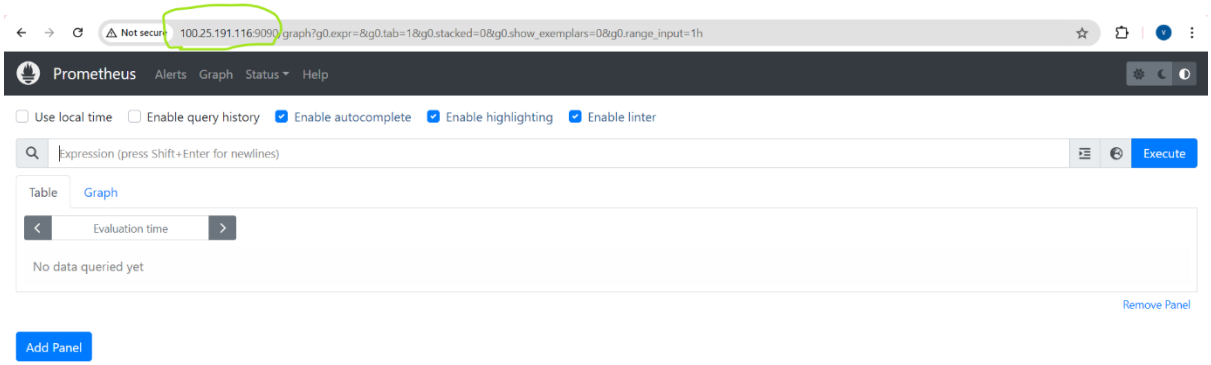
✓ Run and Check Prometheus status:

- Then we need to run commands which will restart the daemon and start and enables the prometheus service.

```
sudo systemctl daemon-reload

sudo systemctl start prometheus

sudo systemctl enable prometheus
```

```
ubuntu@ip-172-31-60-253:~$ sudo systemctl status prometheus.service
● prometheus.service - Prometheus
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-06-30 06:04:40 UTC; 22min ago
   Main PID: 390 (prometheus)
      Tasks: 8 (limit: 9497)
     Memory: 78.8M
        CPU: 831ms
     CGroup: /system.slice/prometheus.service
             └─390 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.templates=>

Jun 30 06:04:42 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:42.466Z caller=main.go:1209 level=info msg="Loading configuration file" filename=/etc>
Jun 30 06:04:42 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:42.483Z caller=main.go:1246 level=info msg="Completed loading of configuration file" >
Jun 30 06:04:42 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:42.483Z caller=main.go:990 level=info msg="Server is ready to receive web requests."
Jun 30 06:04:42 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:42.484Z caller=manager.go:974 level=info component="rule manager" msg="Starting rule >
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.545Z caller=compact.go:510 level=info component=tsdb msg="write block resulted in >
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.546Z caller=head.go:1269 level=info component=tsdb msg="Head GC completed" caller=>
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.585Z caller=compact.go:519 level=info component=tsdb msg="write block" mint=171949>
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.587Z caller=head.go:1269 level=info component=tsdb msg="Head GC completed" caller=>
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.587Z caller=checkpoint.go:100 level=info component=tsdb msg="Creating checkpoint" >
Jun 30 06:04:50 ip-172-31-60-253 prometheus[390]: ts=2024-06-30T06:04:50.603Z caller=head.go:1241 level=info component=tsdb msg="WAL checkpoint complete" f>
lines 1-20/20 (END)
```

- After starting the we need to verify for the prometheus service status.
- Use sudo systemctl status prometheus command for the status verification of prometheus service.
- If our service is up and running, we can now able to access the prometheus UI in the browser using the public IP of the server on port 9090. **http://<public-ip>:9090**.



✓ Configuring Prometheus:

- Prometheus configuration files will be present in **/etc/prometheus** location.
- To configure prometheus we need to edit the prometheus.yml file which we be present in **/etc/prometheus.**

➢ Installing Grafana:

- We can install Grafana in the same instance where we have installed Prometheus.

➢ Download Grafana:

[wget https://dl.grafana.com/oss/release/grafana-9.3.6.linux-amd64.tar.gz]

- Download the Grafana tar file using the above command or use wget and URL of the choice from official documentation.

Download Grafana

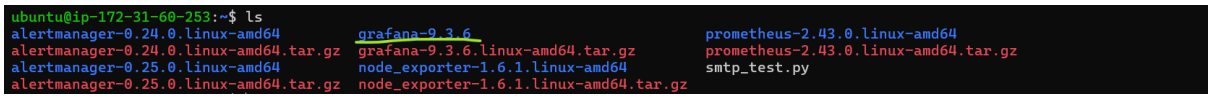
- Here, I have used **grafana-9.3.6.linux-amd64.tar.gz** to download Grafana.

➢ Extracting the downloaded Grafana tar file:

- After Downloading the tar file, we need to extract the tar file to get all the binaries required to install the Grafana.

[sudo tar xvfz grafana-9.3.6.linux-amd64.tar.gz]

- This command will extract the tar file in the home path, after the extraction is done, we will get a folder named **grafana-9.3.6** which contains all the binaries and dependencies required to install the Grafana.



➢ Move the Extracted Files:

- Move the extracted Grafana directory to **/usr/local.**

[sudo mv grafana-9.3.6 /usr/local/grafana]

➢ Create a Symbolic Link:

- Create a symbolic link to make it easier to access Grafana.

```
sudo ln -s /usr/local/grafana/bin/grafana-server /usr/local/bin/grafana-server

sudo ln -s /usr/local/grafana/bin/grafana-cli /usr/local/bin/grafana-cli
```

➢ Create a Grafana User:

- Create a new system user for running Grafana
- We need to create a dedicated user for grafana so that service follows best practices for Unix/Linux system administration.
- It helps in tracking and managing resources used by specific services and simplifies troubleshooting.

- So, we need create a user and change the ownership and permissions for the user accordingly.

```
sudo useradd --no-create-home --shell /bin/false grafana

sudo chown -R grafana:grafana /usr/local/grafana
```

```
ubuntu@ip-172-31-60-253:~$ cat /etc/passwd | grep grafana
grafana:x:115:122::/usr/share/grafana:/bin/false
ubuntu@ip-172-31-60-253:~$
```

➢ Create Grafana Service:

- Now we need to create the service for Grafana. Creating a new systemd service file for Grafana is essential for managing Grafana as a background service on a Linux system.

[sudo nano /etc/systemd/system/grafana.service]

(or)

[sudo nano /etc/systemd/system/grafana.service]

- This above command will create a service file for prometheus in /etc/systemd/system path with the service name as grafana (Generally any service file will present in this location in linux).

```
[Unit]
Description=Grafana instance
After=network.target

[Service]
User=grafana
Group=grafana
Type=simple
ExecStart=/usr/local/grafana/bin/grafana-server --
config=/usr/local/grafana/conf/defaults.ini --homepath=/usr/local/grafana

[Install]
WantedBy=multi-user.target
```

- We need to add the above content in the service file to run grafana as a service.
- After adding the above content in grafana.service file we need to save the file.
- Save the file using **:wq!** in vi editor and with **ctrl+X** and **Y** in nano editor.

```
ubuntu@ip-172-31-60-253:/etc/systemd/system$ ls
 alertmanager.service                    open-vm-tools.service.requires                snap.lxd.activate.service
 chronyd.service                         paths.target.wants                            snap.lxd.daemon.service
 cloud-final.service.wants               prometheus.service                            snap.lxd.daemon.unix.socket
 cloud-init.target.wants                 rescue.target.wants                           snap.lxd.user-daemon.service
 dbus-org.freedesktop.resolve1.service   sleep.target.wants                            snap.lxd.user-daemon.unix.socket
 emergency.target.wants                 'snap-amazon\x2dssm\x2dagent-7983.mount'       snapd.mounts.target.wants
 final.target.wants                      snap-core18-2823.mount                        sockets.target.wants
 getty.target.wants                      snap-core18-2829.mount                        sshd-keygen@.service.d
 grafana.service                         snap-core20-2264.mount                        sshd.service
 iscsi.service                           snap-core20-2318.mount                        sudo.service
 mdmonitor.service.wants                 snap-lxd-27948.mount                          sysinit.target.wants
 multi-user.target.wants                 snap-lxd-28373.mount                          syslog.service
 multipath-tools.service                 snap-snapd-21184.mount                        timers.target.wants
 network-online.target.wants             snap-snapd-21759.mount                        vmtoolsd.service
 node_exporter.service                   snap.amazon-ssm-agent.amazon-ssm-agent.service
```

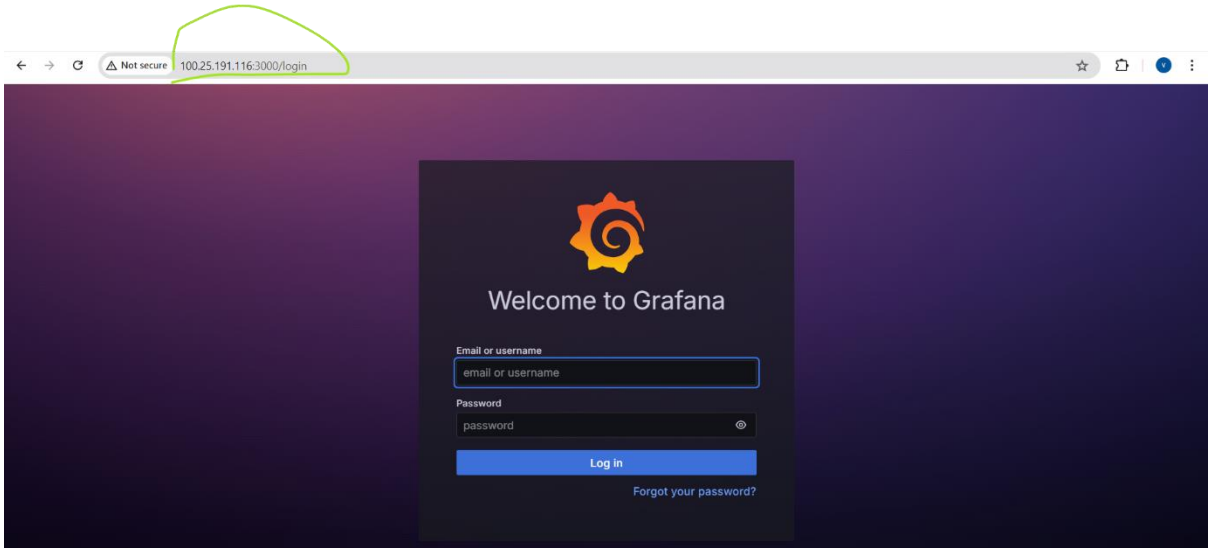
➢ Run and Check Grafana Status:

- After saving the service file we need to start and enable the grafana service.
- Then we need to run commands which will restart the daemon and start and enables the grafana service

```
sudo systemctl daemon-reload

sudo systemctl start grafana

sudo systemctl enable grafana

sudo systemctl status grafana
```

- If our service is up and running, we can now able to access the grafana UI in the browser using the public IP of the server on port 3000. **http://<public-ip>:3000**.

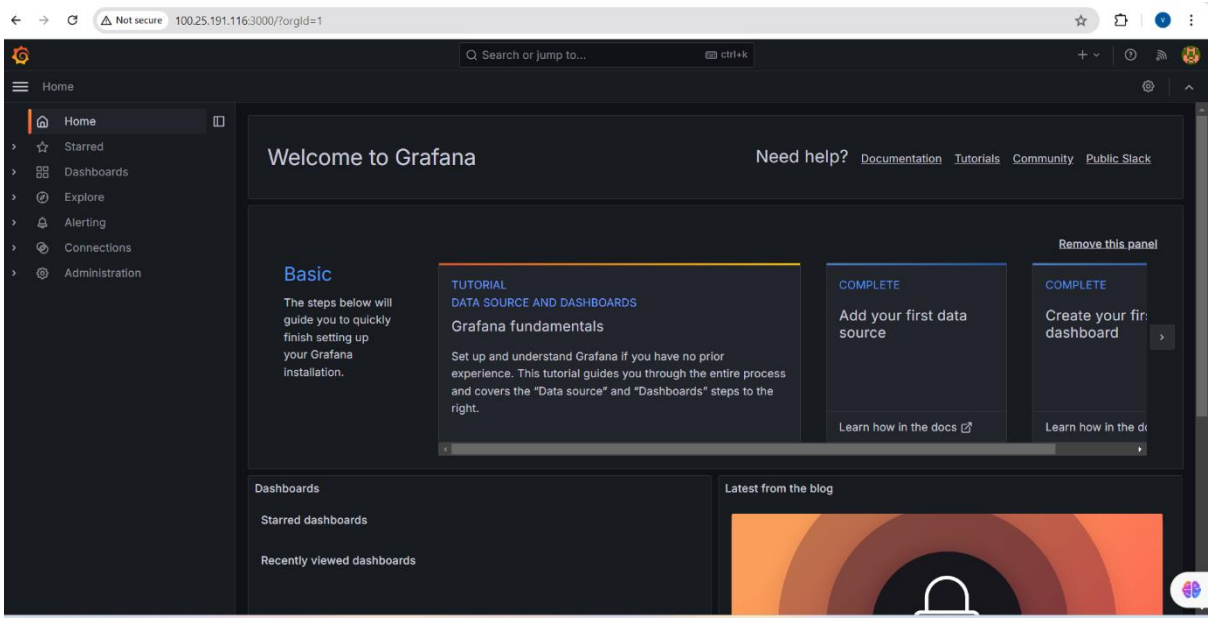
```
ubuntu@ip-172-31-60-253:~$ sudo systemctl status grafana-server.service
● grafana-server.service - Grafana instance
     Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-06-30 06:04:40 UTC; 34min ago
       Docs: http://docs.grafana.org
   Main PID: 379 (grafana)
      Tasks: 13 (limit: 9497)
     Memory: 180.6M
        CPU: 3.230s
     CGroup: /system.slice/grafana-server.service
             └─379 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:de

Jun 30 06:04:45 ip-172-31-60-253 grafana[379]: logger=grafana-apiserver t=2024-06-30T06:04:45.4813274922 level=info msg="Adding GroupVersion featuretoggle.
Jun 30 06:06:18 ip-172-31-60-253 grafana[379]: logger=infra.usagestats t=2024-06-30T06:06:18.5131180332 level=info msg="Usage stats are ready to report"
Jun 30 06:14:44 ip-172-31-60-253 grafana[379]: logger=cleanup t=2024-06-30T06:14:44.5053008822 level=info msg="Completed cleanup jobs" duration=36.868897ms
Jun 30 06:14:44 ip-172-31-60-253 grafana[379]: logger=plugins.update.checker t=2024-06-30T06:14:44.7635108672 level=info msg="Update check succeeded" durat
Jun 30 06:24:44 ip-172-31-60-253 grafana[379]: logger=cleanup t=2024-06-30T06:24:44.4712087722 level=info msg="Completed cleanup jobs" duration=3.634623ms
Jun 30 06:24:44 ip-172-31-60-253 grafana[379]: logger=plugins.update.checker t=2024-06-30T06:24:44.7658090132 level=info msg="Update check succeeded" durat
Jun 30 06:32:55 ip-172-31-60-253 grafana[379]: logger=context userId=0 orgId=0 uname= t=2024-06-30T06:32:55.6657220712 level=info msg="Request Completed" m
Jun 30 06:34:44 ip-172-31-60-253 grafana[379]: logger=cleanup t=2024-06-30T06:34:44.4715130452 level=info msg="Completed cleanup jobs" duration=3.214132ms
Jun 30 06:34:44 ip-172-31-60-253 grafana[379]: logger=plugins.update.checker t=2024-06-30T06:34:44.7634730962 level=info msg="Update check succeeded" durat
Jun 30 06:36:18 ip-172-31-60-253 grafana[379]: logger=infra.usagestats t=2024-06-30T06:36:18.519676792 level=info msg="Usage stats are ready to report"
lines 1-21/21 (END)
```



➢ Access Grafana:

- Login to Grafana using the default credentials then we need to update the password.

- **default username:** **admin**
- **default password:** **admin**



✓ This is the process to install Prometheus and Grafana in Ubuntu-22.04 EC2 instance.

Thanks,

Lahari G