

Monitoring Task

TASKS

Work Flow:

- Create an EC2 instance with the help of AWS Management Console with linux OS of required configuration.
- Now, Connect an EC2 instance with an help of Windows Terminal or Gitbash or Vbox.
- To connect an EC2 instance the command is:
 - `ssh -i "key_file" ec2-user@"Public_IP_address"`

Key_file --- Key file of the instance with the extension .pem

Public_IP_address --- Public IP address of the instance.

The screenshot shows the AWS Management Console instance details for an EC2 instance. Key information includes:

- Instance ID:** i-0db4d857d36f76e44
- Public IPv4 address:** 13.220.219.6
- Private IP addresses:** 172.31.39.182
- Instance state:** Running
- Instance type:** t3.micro
- VPC ID:** vpc-04af07505f15640b
- Subnet ID:** subnet-062bb0c40773599285
- Instance ARN:** arn:aws:ec2:us-east-1:634018648846:instance/i-0db4d857d36f76e44

1. Install Prometheus and Grafana on a Linux EC2 machine, connect Prometheus to Grafana, and create a dashboard to view metrics.

Step 1: Install Prometheus

- ✓ To begin installing Prometheus, we first create a dedicated user and necessary directories for Prometheus to store its configuration and data.
- ✓ Run the following commands:
 - `sudo useradd --no-create-home --shell /bin/false Prometheus`
 - `sudo mkdir /etc/prometheus /var/lib/Prometheus`

```
[ec2-user@ip-172-31-39-182 ~]$ sudo useradd --no-create-home --shell /bin/false prometheus
[ec2-user@ip-172-31-39-182 ~]$ sudo mkdir /etc/prometheus
[ec2-user@ip-172-31-39-182 ~]$ sudo mkdir /var/lib/prometheus
[ec2-user@ip-172-31-39-182 ~]$ |
```

- ✓ Next, we download the Prometheus software package from the official GitHub release page.
 - ✓ Use the command below to download and extract it:
 - `cd /tmp`
 - `curl -LO`
- <https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz>

```
etheus-2.52.0.linux-amd64.tar.gztar -xvf prometheus-2.52.0.linux-amd64.tar.gz
```

- **cd prometheus-2.52.0.linux-amd64**

```
[ec2-user@ip-172-31-39-182 ~]$ cd /tmp  
[ec2-user@ip-172-31-39-182 tmp]$ curl -LO https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz  
tar -xvf prometheus-2.52.0.linux-amd64.tar.gz  
% Total    % Received % Xferd  Average Speed   Time   Time  Current  
          Dload  Upload Total Spent   Left Speed  
  0     0    0    0    0    0    0 --:--:--:--:--:--:--:--:--:--:--:--:0  
100 99.8M  100 99.8M  0    0  256M    0 --:--:--:--:--:--:368M  
prometheus-2.52.0.linux-amd64/  
prometheus-2.52.0.linux-amd64/LICENSE  
prometheus-2.52.0.linux-amd64/promtool  
prometheus-2.52.0.linux-amd64/console_libraries/  
prometheus-2.52.0.linux-amd64/console_libraries/prom.lib  
prometheus-2.52.0.linux-amd64/console_libraries/menu.lib  
prometheus-2.52.0.linux-amd64/prometheus  
prometheus-2.52.0.linux-amd64/consoles/  
prometheus-2.52.0.linux-amd64/consoles/node-overview.html  
prometheus-2.52.0.linux-amd64/consoles/prometheus.html  
prometheus-2.52.0.linux-amd64/consoles/node.html  
prometheus-2.52.0.linux-amd64/consoles/index.html.example  
prometheus-2.52.0.linux-amd64/consoles/node-disk.html  
prometheus-2.52.0.linux-amd64/consoles/node-cpu.html  
prometheus-2.52.0.linux-amd64/consoles/prometheus-overview.html  
prometheus-2.52.0.linux-amd64/prometheus.yml  
prometheus-2.52.0.linux-amd64/NOTICE  
[ec2-user@ip-172-31-39-182 tmp]$ cd prometheus-2.52.0.linux-amd64  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ |
```

- ✓ Next, we download the Prometheus software package from the official GitHub release page.
- ✓ Use the command below to download and extract it:
 - **sudo cp prometheus /usr/local/bin/**
 - **sudo cp promtool /usr/local/bin/**
 - **sudo cp -r consoles/ console_libraries/ /etc/prometheus/**
 - **sudo cp prometheus.yml /etc/prometheus/**

```
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo cp prometheus /usr/local/bin/  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo cp promtool /usr/local/bin/  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo cp -r consoles/ console_libraries/ /etc/prometheus/  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo cp prometheus.yml /etc/prometheus/  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ |
```

- ✓ We then update ownership of the directories and files to the Prometheus user so it can access and manage them properly:
 - **sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/Prometheus**

```
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus  
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ |
```

- ✓ Now we need to set up Prometheus as a service so it can run in the background. Create the service file using the following command:
 - **sudo tee /etc/systemd/system/prometheus.service > /dev/null <<EOF**
- ```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target
```
- 
- ```
[Service]  
User=prometheus  
ExecStart=/usr/local/bin/prometheus \\
```

```
--config.file=/etc/prometheus/prometheus.yml \\
--storage.tsdb.path=/var/lib/prometheus/
```

[Install]

WantedBy=default.target

EOF

```
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo tee /etc/systemd/system/prometheus.service > /dev/null <<EOF
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

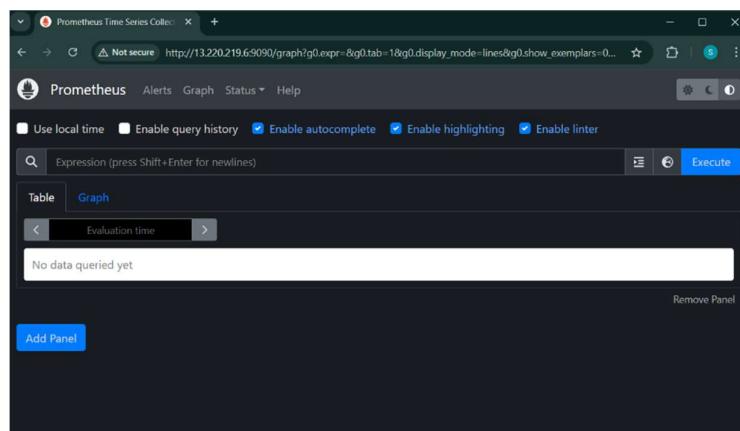
[Service]
User=prometheus
ExecStart=/usr/local/bin/prometheus \
--config.file=/etc/prometheus/prometheus.yml \
--storage.tsdb.path=/var/lib/prometheus/

[Install]
WantedBy=default.target
EOF
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ |
```

- ✓ Finally, reload the systemd service manager and start the Prometheus service:
 - **sudo systemctl daemon-reexec**
 - **sudo systemctl start prometheus**
 - **sudo systemctl enable Prometheus**

```
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo systemctl daemon-reexec
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo systemctl start prometheus
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/default.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
[ec2-user@ip-172-31-39-182 prometheus-2.52.0.linux-amd64]$ |
```

- ✓ At this point, Prometheus is successfully installed and running on your EC2 instance. You can access it by opening your browser and navigating to:
 - **http:// 13.220.219.6:9090**
- ✓ This will open the Prometheus web UI where you can explore available metrics.



Step 2: Install Grafana on the Linux EC2 Instance.

- ✓ First, we downloaded and installed the Grafana package using the official .rpm file. On Amazon Linux 2, we ran the following command:
 - **sudo yum install -y https://dl.grafana.com/oss/release/grafana-10.4.2-1.x86_64.rpm**

```
[ec2-user@ip-172-31-39-182 ~]$ sudo yum install -y https://dl.grafana.com/oss/release/grafana-10.4.2-1.x86_64.rpm
Amazon Linux 2023 Kernel Livepatch repository
grafana-10.4.2-1.x86_64.rpm
Dependencies resolved.
=====
 Package          Architecture Version      Repository   Size
=====
 Installing:
 grafana          x86_64      10.4.2-1    @commandline 108 M
=====
 Installing dependencies:
 auctex           x86_64      1.18.0-4.amzn2023.0.2  amazonlinux    717 k
 fontconfig        x86_64      2.13.94-2.amzn2023.0.2  amazonlinux    273 k
 font-filesystem  noarch     1:2.0.5-12.amzn2023.0.2  amazonlinux    9.5 k
 freetype          x86_64      2.10.1-1.amzn2023.0.2  amazonlinux    42 k
 glib              x86_64      2.92.0-11.amzn2023.0.2  amazonlinux    17 k
 google-fonts-sans-serif  noarch     2.0204001-1.amzn2023.0.2  amazonlinux    593 k
 graphite2         x86_64      1.3.14-7.amzn2023.0.2  amazonlinux    97 k
 harfbuzz          x86_64      7.0.0-1.amzn2023.0.2  amazonlinux    377 k
 libraqm           noarch     3.0-21.amzn2023.0.4   amazonlinux    18 k
 libX11            x86_64      1.8.10-2.amzn2023.0.1  amazonlinux    659 k
 libX11-common    noarch     1:8.10-2.amzn2023.0.1  amazonlinux    147 k
 libXau            x86_64      1.0.9-1.amzn2023.0.1  amazonlinux    53 k
 libXext            x86_64      1.3.6-1.amzn2023.0.1  amazonlinux    42 k
 libXrender         x86_64      0.9.11-6.amzn2023.0.1  amazonlinux    29 k
 libXpixmap         x86_64      1.0.1-1.amzn2023.0.1  amazonlinux    218 k
 libXpm             x86_64      2:1.6.37-10.amzn2023.0.6  amazonlinux    128 k
 libxcb             x86_64      1.17.0-1.amzn2023.0.1  amazonlinux    235 k
 pixman            x86_64      0.43.4-1.amzn2023.0.4  amazonlinux    296 k
 xml-common        noarch     6.6.3-56.amzn2023.0.2  amazonlinux    52 k
=====
 Transaction Summary
=====
 Install 20 Packages
```

- ✓ This command installs the latest stable version of Grafana.
- ✓ Once Grafana was installed, we started the Grafana service using:
 - **sudo systemctl start grafana-server**
- ✓ To make sure Grafana starts automatically on every system reboot, we enabled it with:
 - **sudo systemctl enable grafana-server**

```
[ec2-user@ip-172-31-39-182 ~]$ sudo systemctl start grafana-server
[ec2-user@ip-172-31-39-182 ~]$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /usr/lib/systemd/system/grafana-server.service.
[ec2-user@ip-172-31-39-182 ~]$ |
```

- ✓ After starting the service, Grafana was accessible via a web browser at:
 - **http:// 13.220.219.6:3000**
- ✓ On first login, the default username and password were both admin. Upon logging in, Grafana prompted us to change the default password for security purposes.

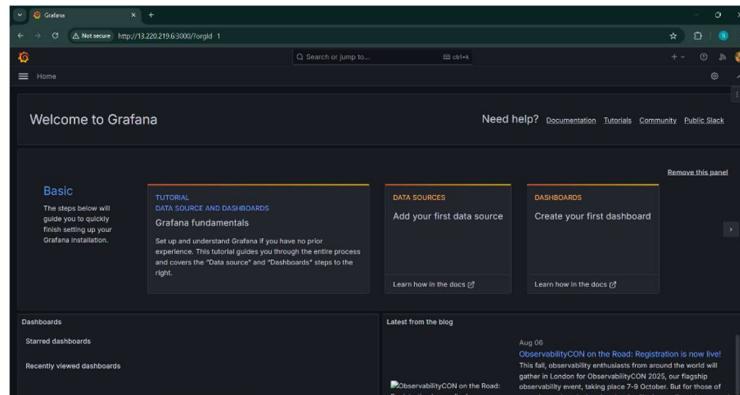


Step 3: Connect Prometheus to Grafana.

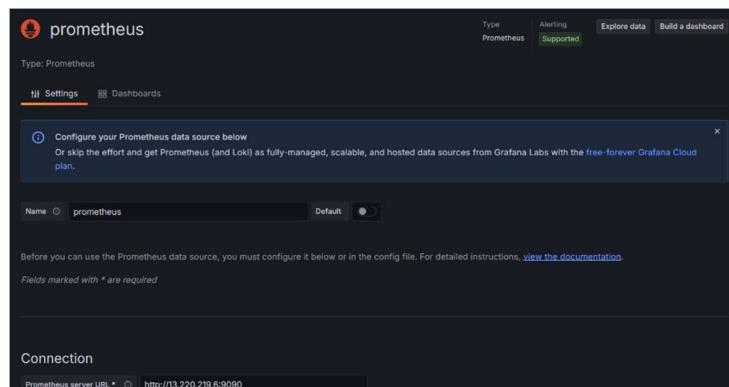
- ✓ After successfully installing Grafana and Prometheus on the EC2 instance, we proceed to connect Prometheus as a data source in Grafana.
- ✓ First, we access the Grafana dashboard by opening the web browser and navigating to:
 - **http:// 13.220.219.6: 3000**
- ✓ We log in using the default credentials:
 - **Username: admin**

- **Password:admin**

(Grafana will prompt to change the password after the first login.)



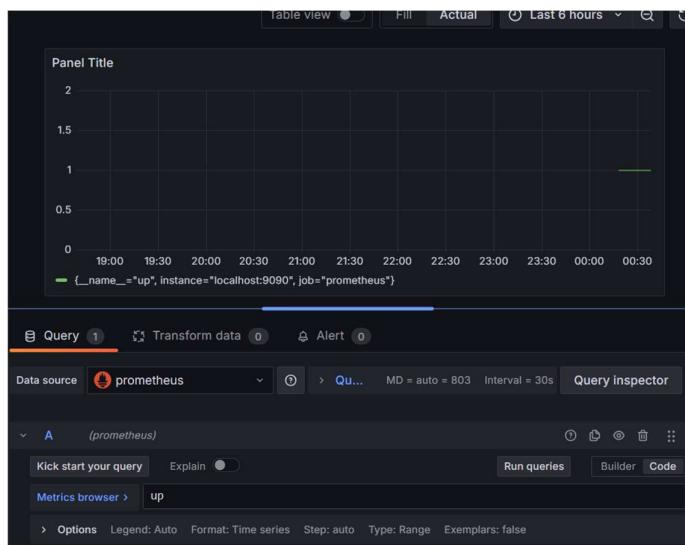
- ✓ Once logged in, we click on the **gear icon** (⚙️) on the left sidebar to go to "**Data Sources**", then click "**Add data source**".
- ✓ From the list of available data sources, we select "**Prometheus**".
- ✓ In the configuration screen, we enter the following in the **URL** field:
 - **http:// 13.220.219.6:9090**
- ✓ This tells Grafana to connect to Prometheus running on the same EC2 instance.
- ✓ We leave all other settings as default and click on the "**Save & Test**" button at the bottom.
- ✓ If the configuration is successful, Grafana displays a green success message saying that the data source is working.
- ✓ This step completes the integration between Grafana and Prometheus, enabling Grafana to query and visualize metrics collected by Prometheus.



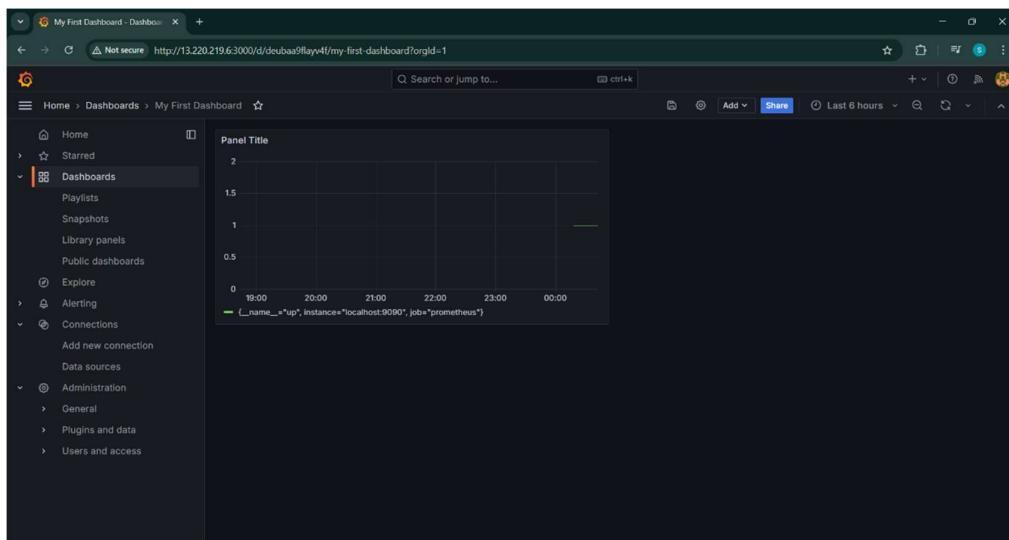
Step 4: Create a Dashboard in Grafana.

- ✓ After connecting Prometheus as a data source in Grafana, we proceed to create a dashboard to visualize metrics.
 - **In the Grafana web interface, we click on the “+” (Create) icon on the left sidebar and select “Dashboard”**
 - **Then we click on “Add new panel” to open the panel editor.**

- In the panel editor, we select Prometheus as the data source from the dropdown at the top.
- Inside the query editor, we type the Prometheus query:
 - up



- ✓ This query shows the status of the targets being monitored (a value of 1 means the target is up and 0 means it's down).
 - Grafana will automatically display a graph based on the query. We then click the “Apply” button at the top right of the panel editor to save the panel to the dashboard.
 - Optionally, we can rename the panel by clicking on the panel title, choosing Edit, and updating the title field.
- ✓ This completes the dashboard setup, and we now have a working Grafana dashboard that displays metrics from Prometheus in real-time.



Conclusion

- ✓ We launched a Linux EC2 instance and installed both Prometheus and Grafana to monitor and visualize system metrics. Prometheus runs on port 9090 and collects metrics, while Grafana (on port 3000) visualizes these metrics. After linking Prometheus as a data source in Grafana, we created a custom dashboard that shows the uptime of the EC2 instance.

***** **TASK COMPLETED** *****