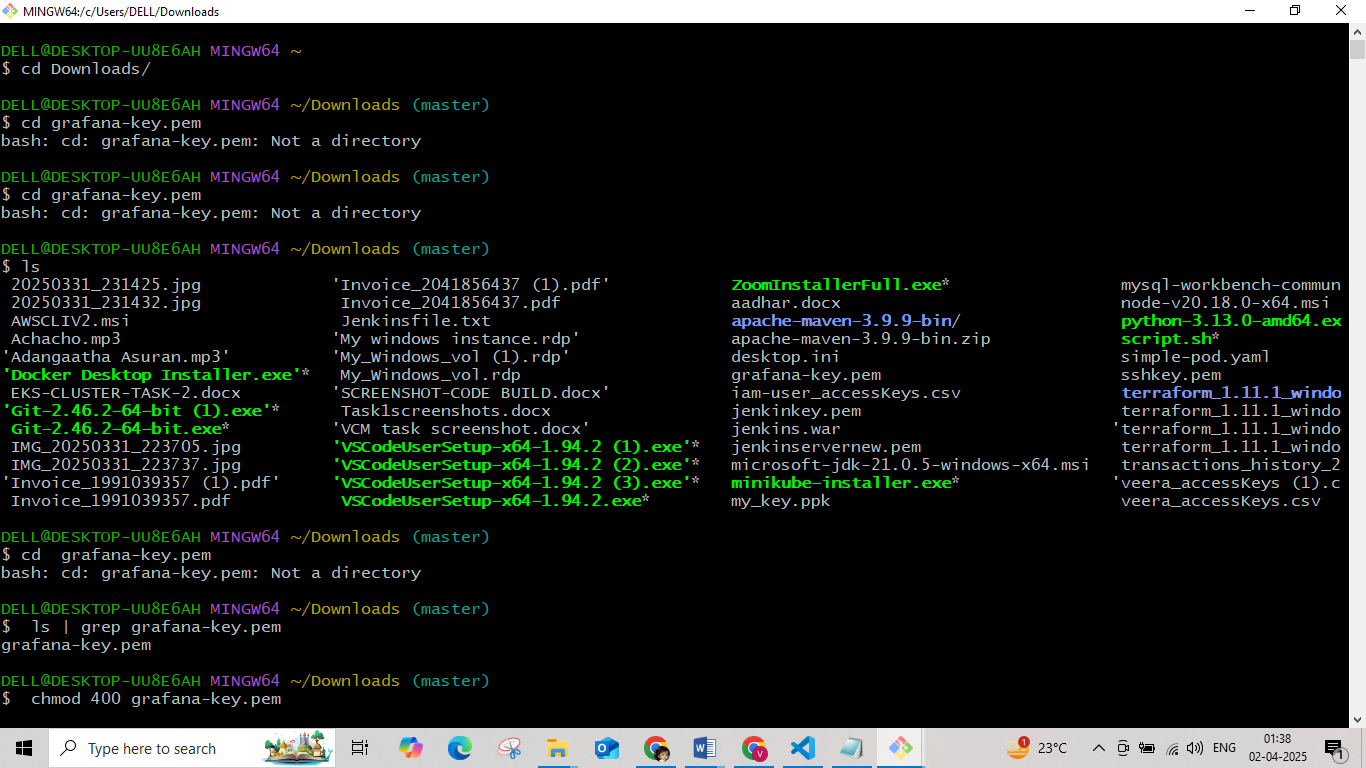
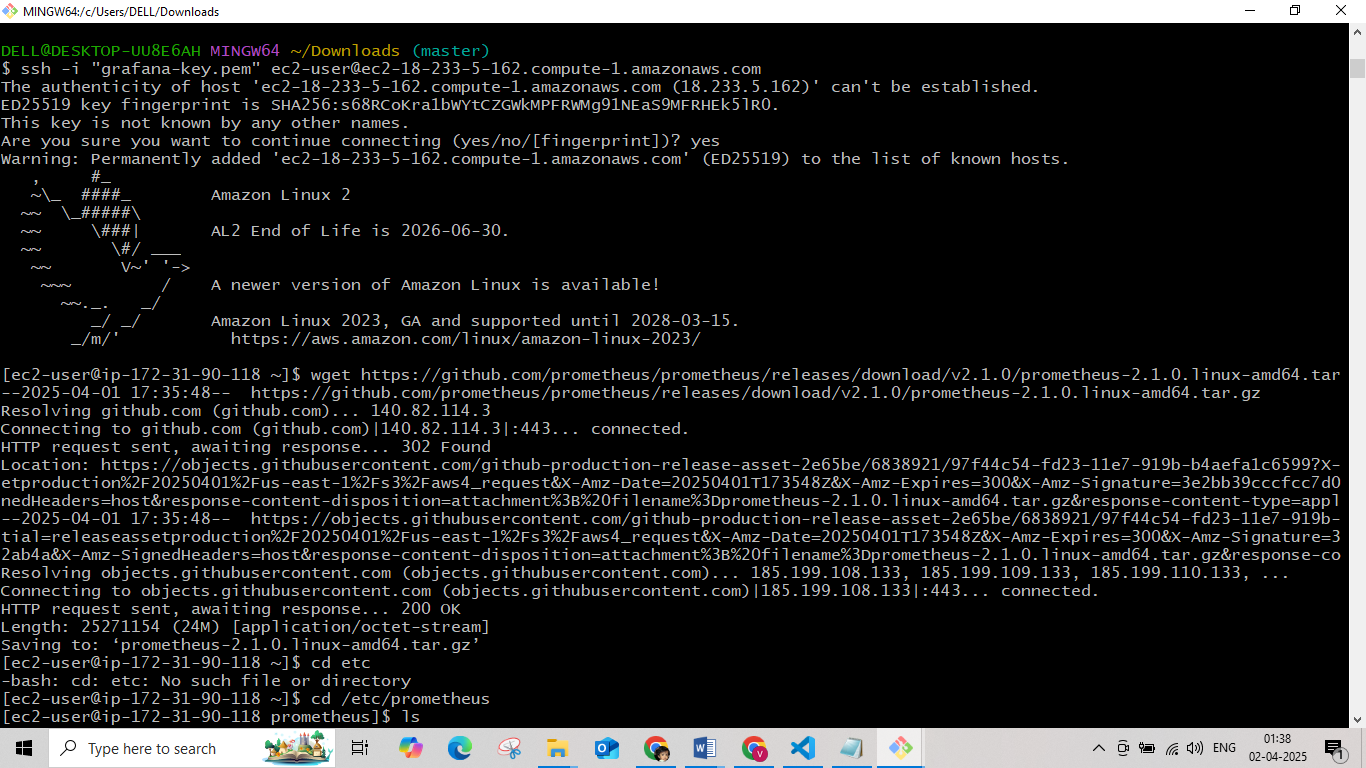
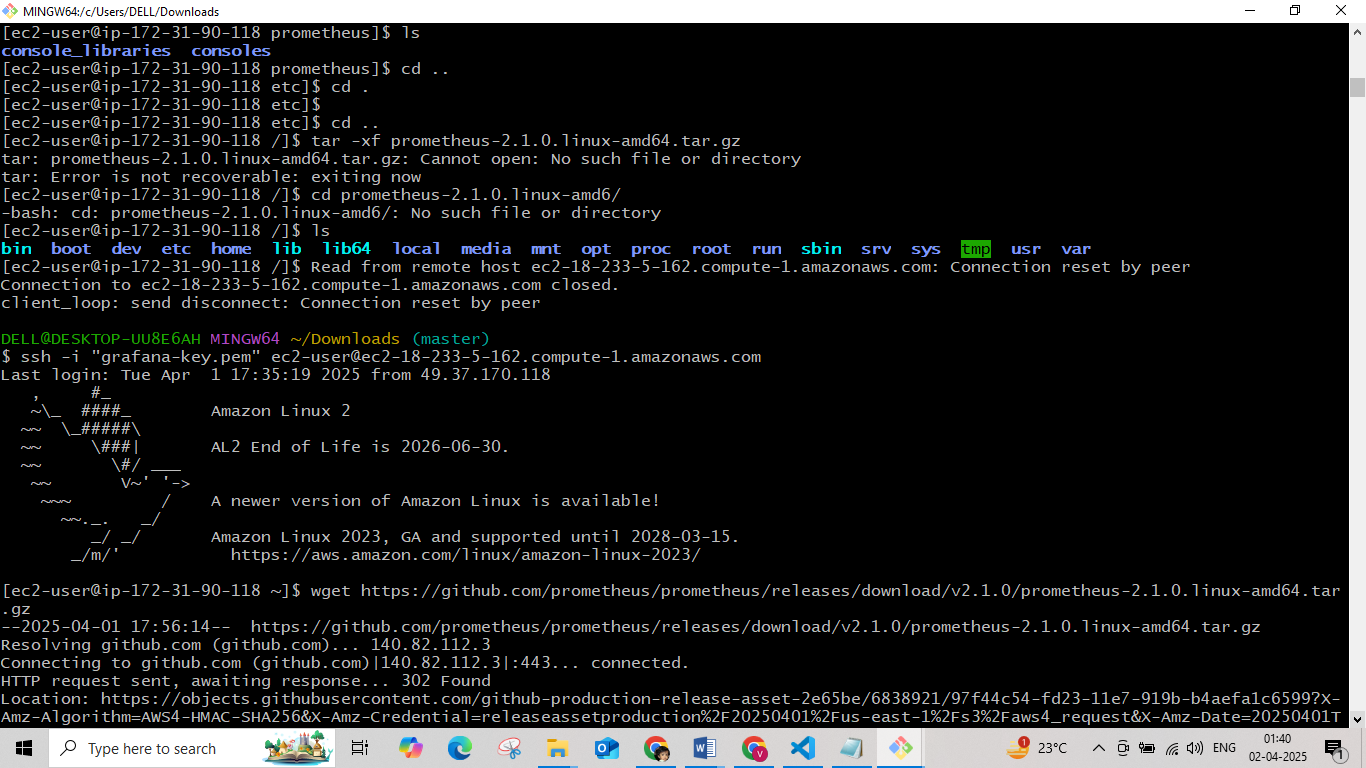
**Install Prometheus**

1. Create EC2 instance for Prometheus and connect it throw SSH



1. Install prometheus usig wget

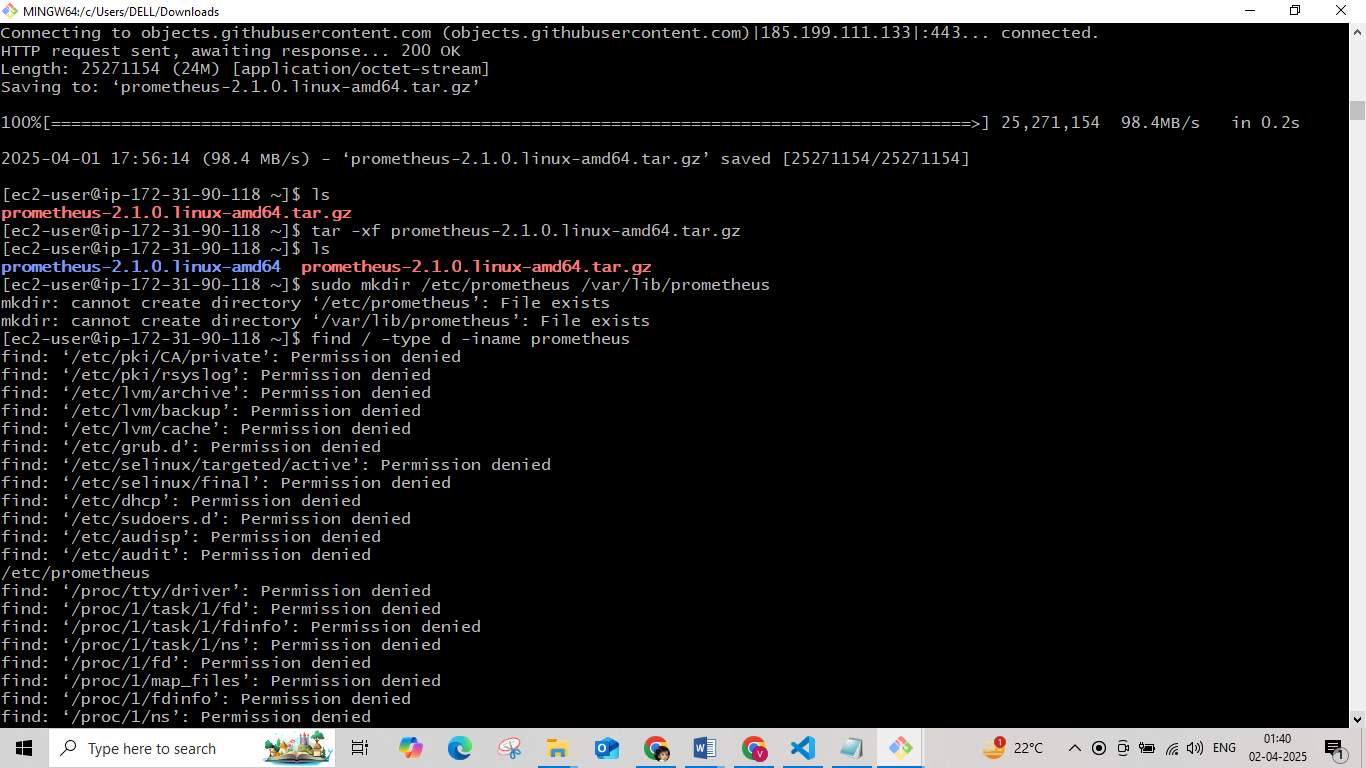




1. Extract the Prometheus archive.

create two directories :

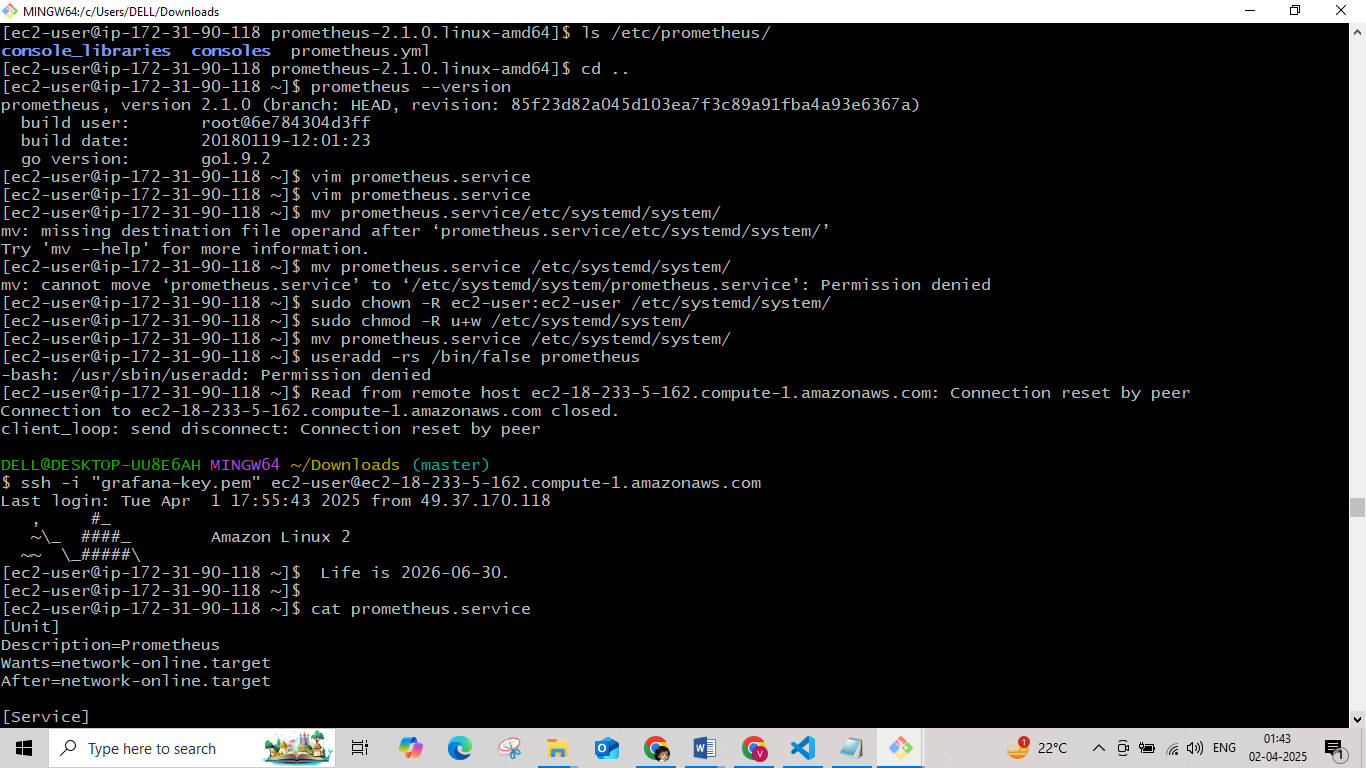
* The “/etc/prometheus” : this directory will store the Prometheus configuration files.
* The “/var/lib/prometheus" : This directory will hold Prometheus data



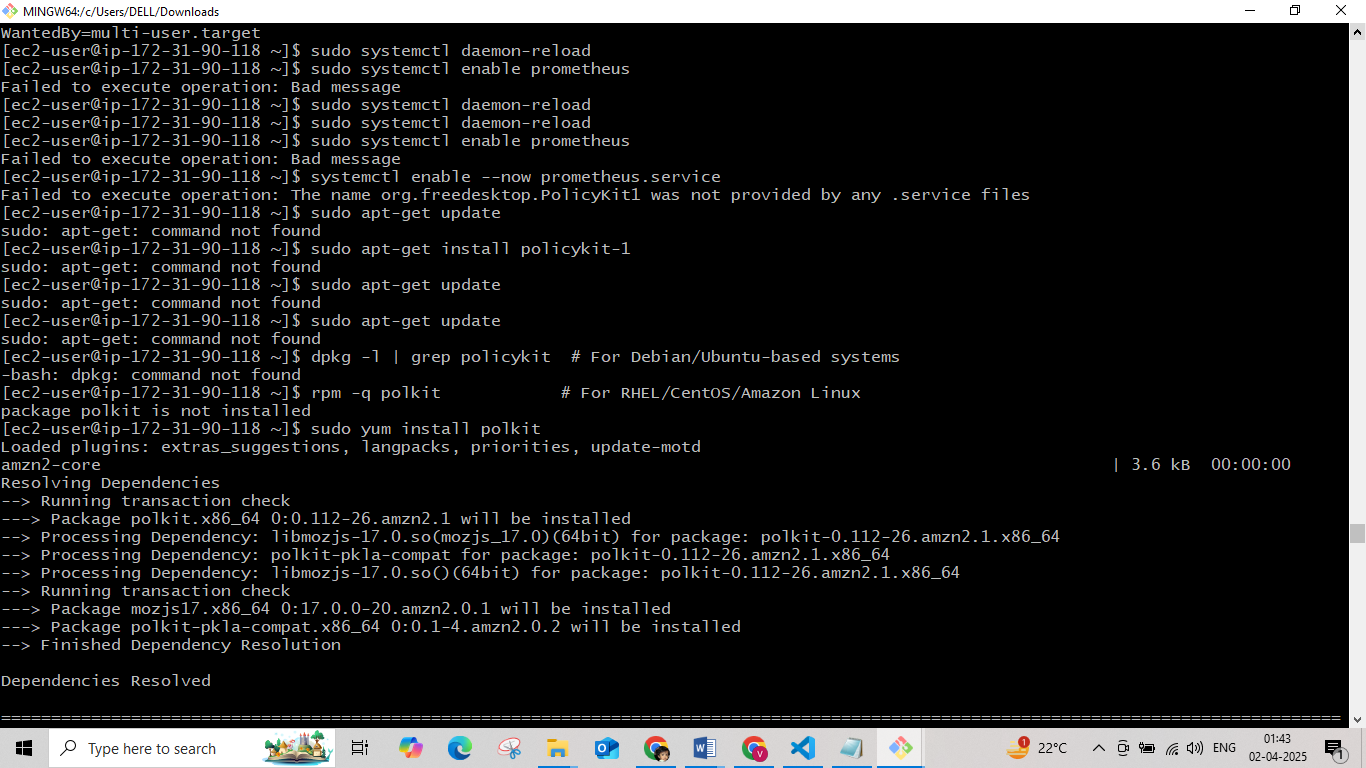
1. Move the **prometheus** and **promtool** binary files to the /usr/local/bin/ directory ( this will makes Prometheus accessible to all users )
2. Next , let’s move the **prometheus.yml** YAML configuration file and the **consoles** and **console\_libraries** directories to the /etc/prometheus . The **consoles** and **console\_libraries** directories contain the resources necessary to create customized consoles ( JavaScript libraries and console templates , this is a more advanced feature and beyond the scope of this article)

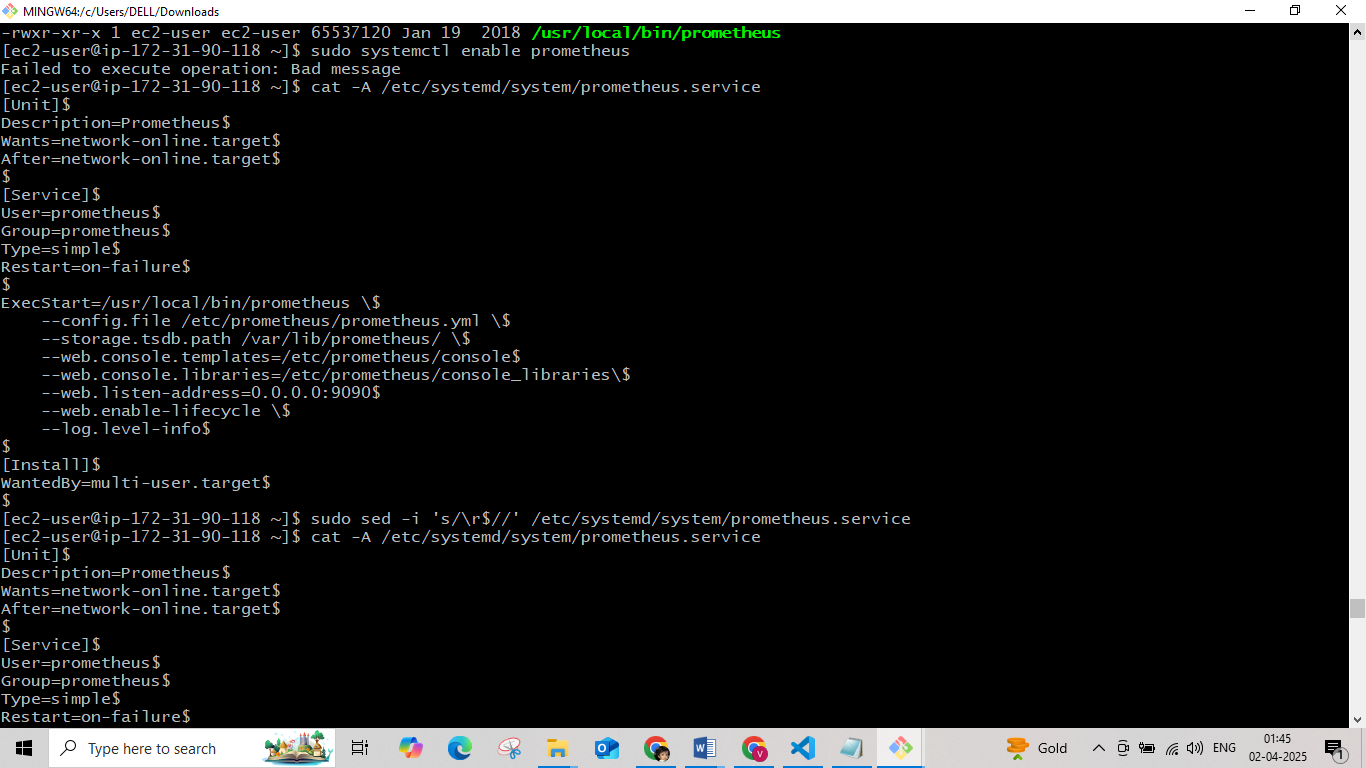


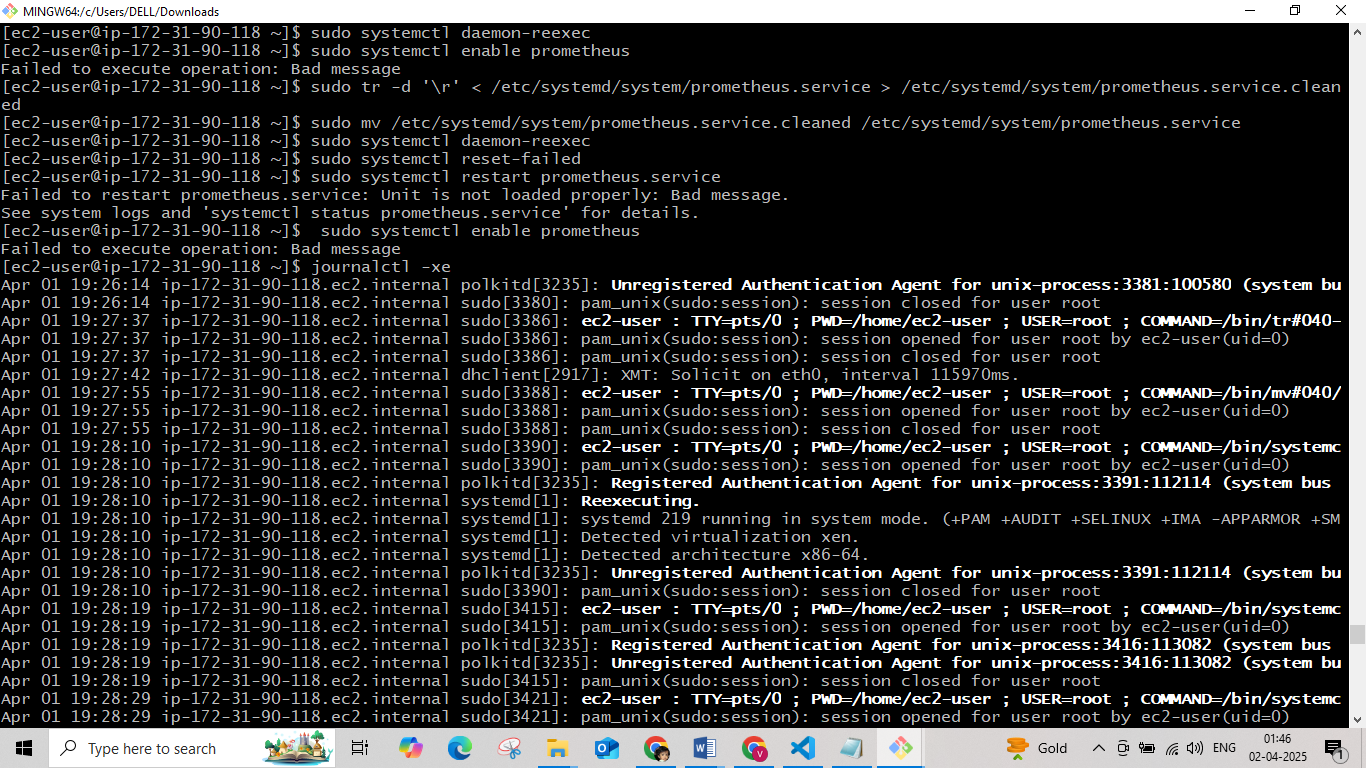
1. To allow Prometheus to run as a service, create a **prometheus.service** file
2. Move the file under the /etc/systemd/system/ and create the user and group Prometheus

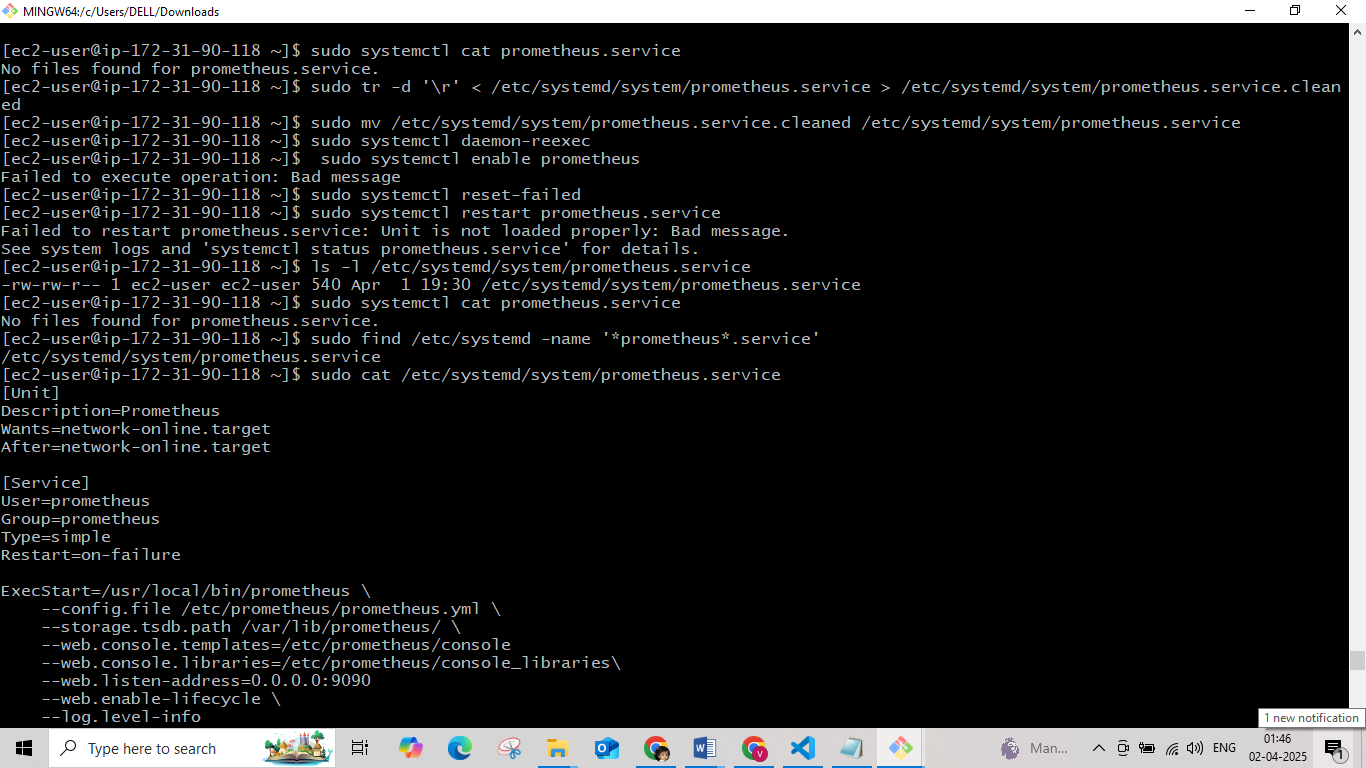


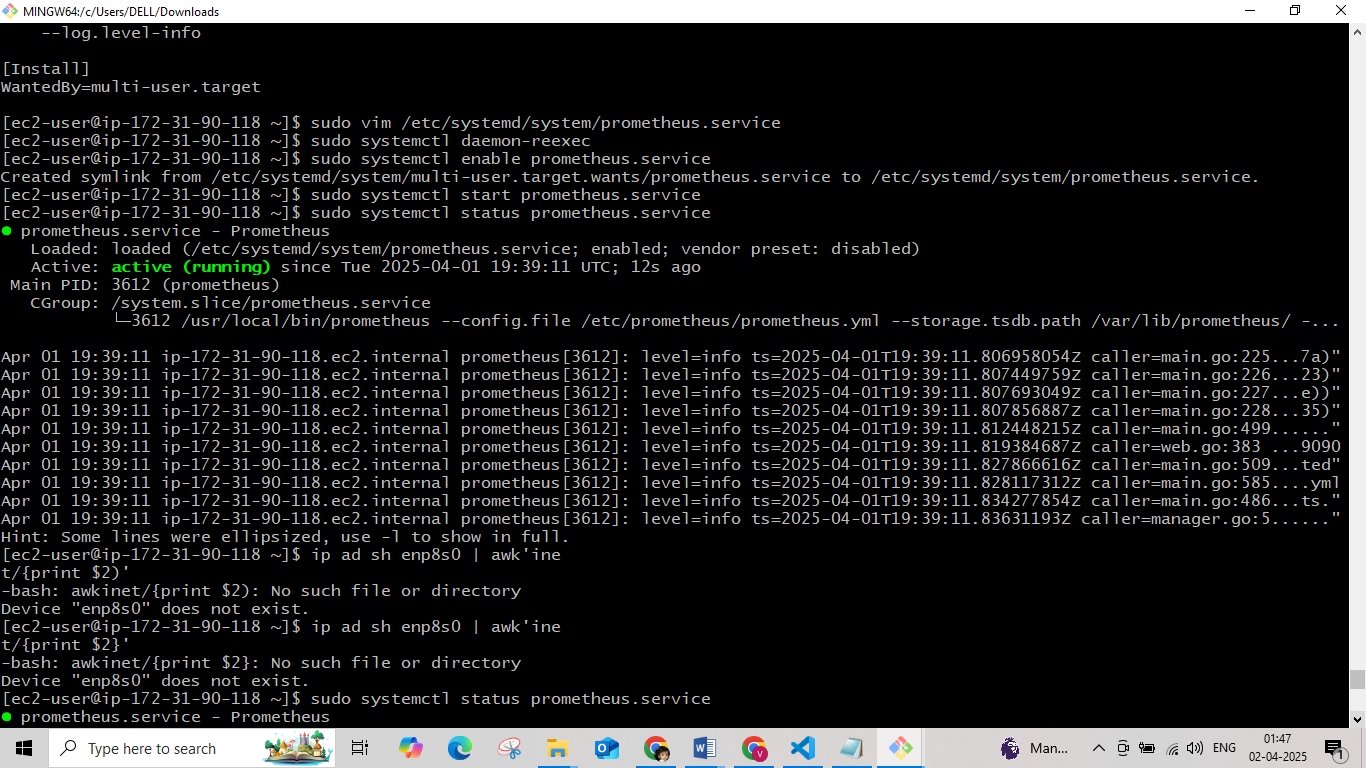
1. Assign the correct ownership and start the service











**Prometheus.service**

[Unit]

Description=Prometheus

After=network.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 \

--web.listen-address=0.0.0.0:9090 \

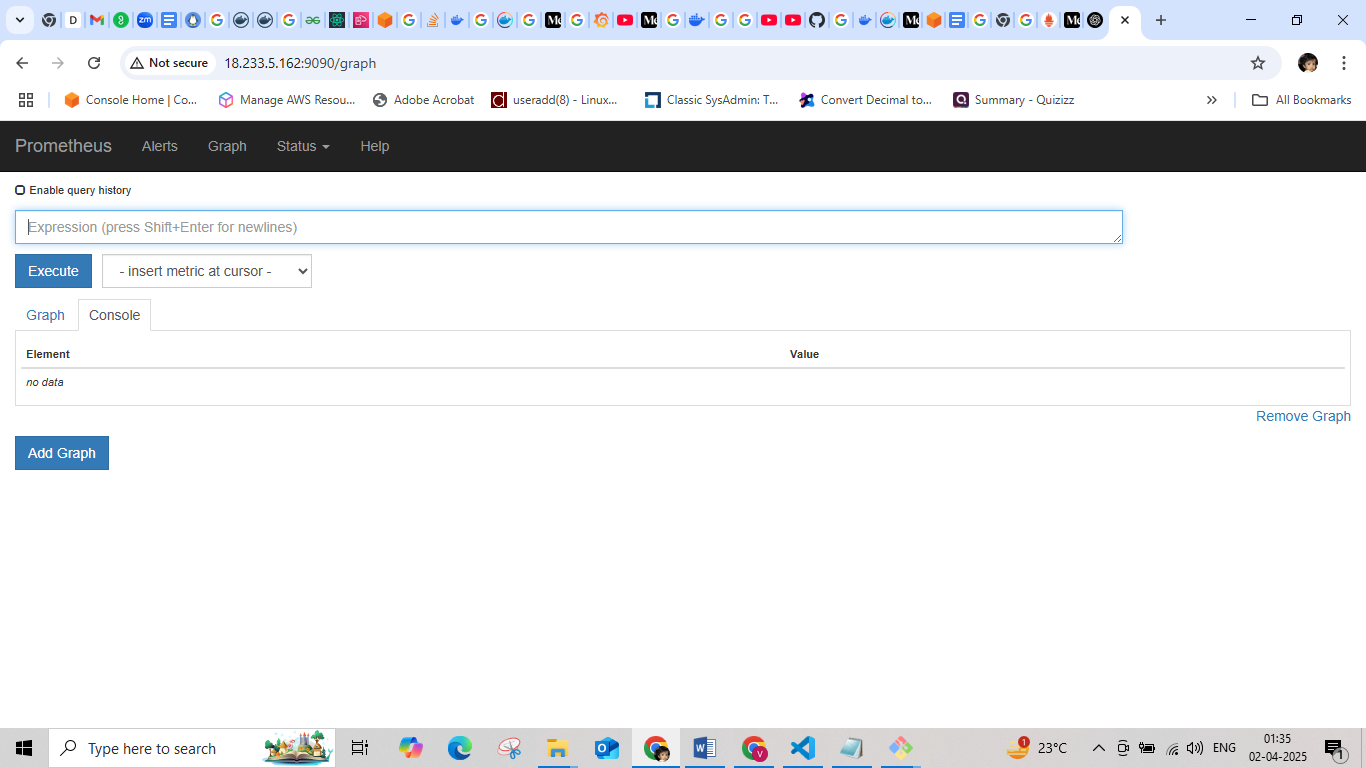
--web.enable-lifecycle \

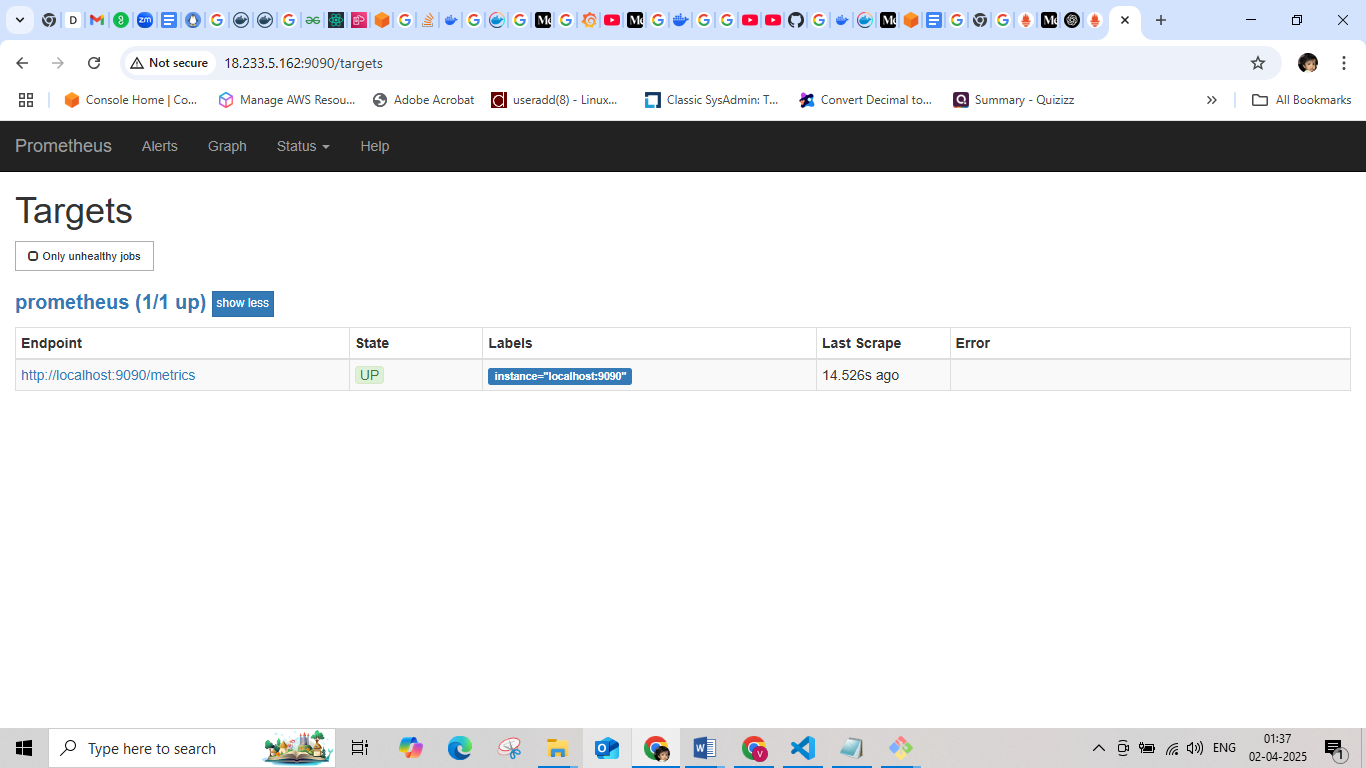
--log.level=info

[Install]

WantedBy=multi-user.target

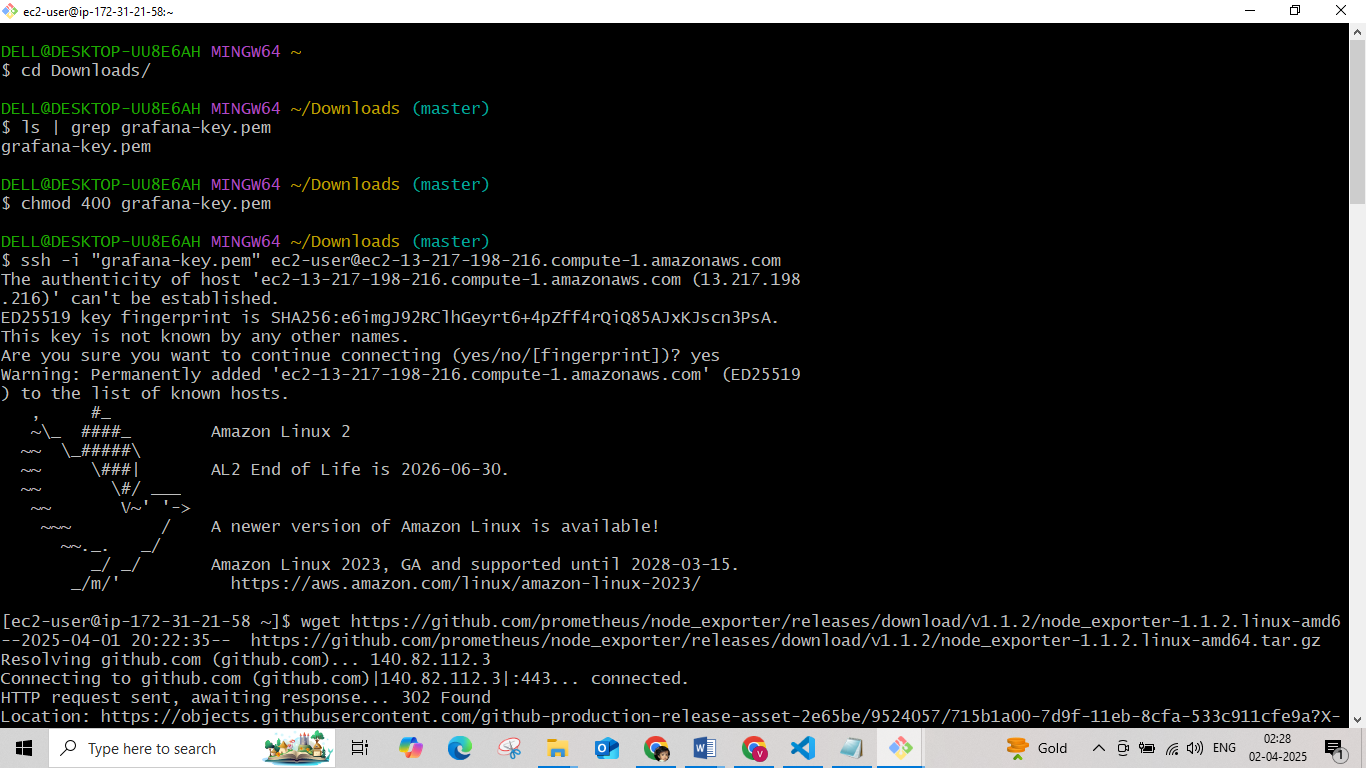
Login to Prometheus browser – http://<ip\_address\_of \_instance>:9090



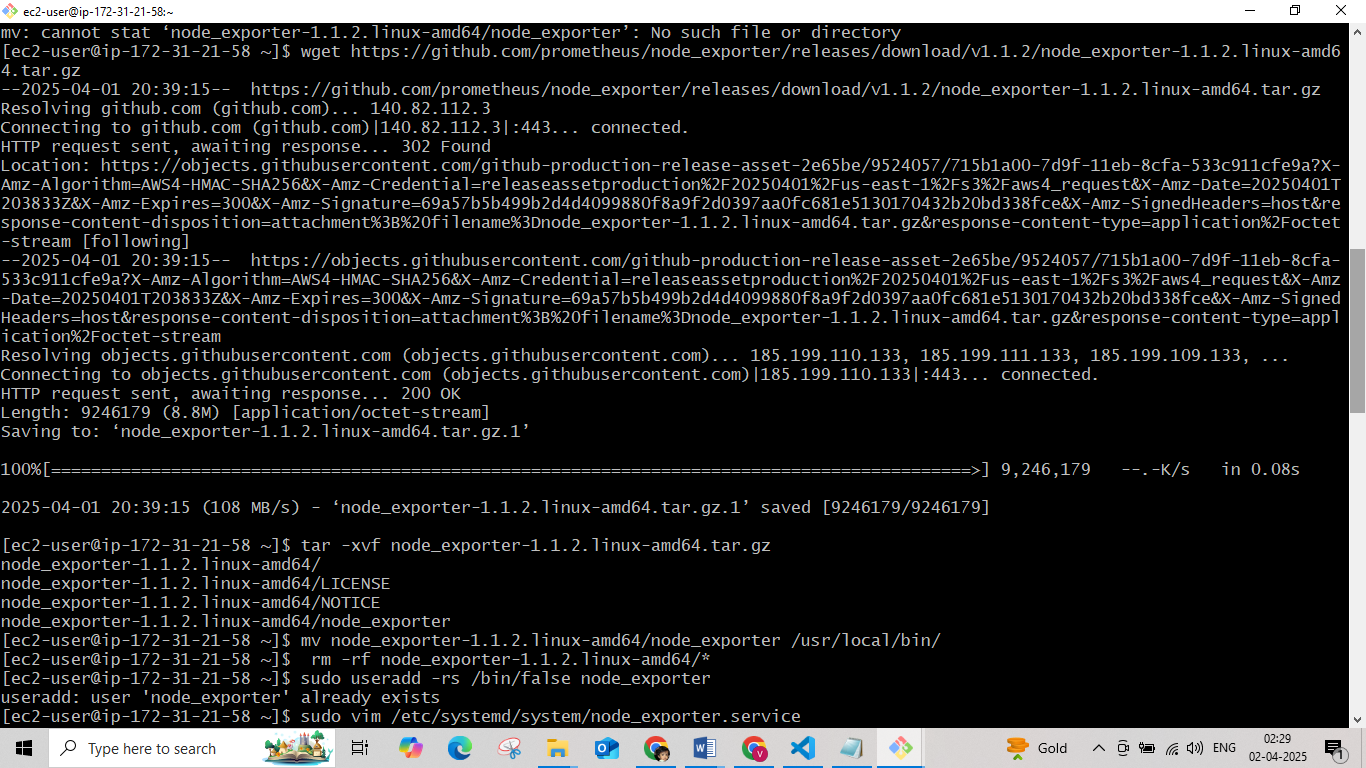


**Install Node-exporter**

* 1. Create an EC2 instance for node-exporter and connect via SSH
  2. Download node-exporter via wget

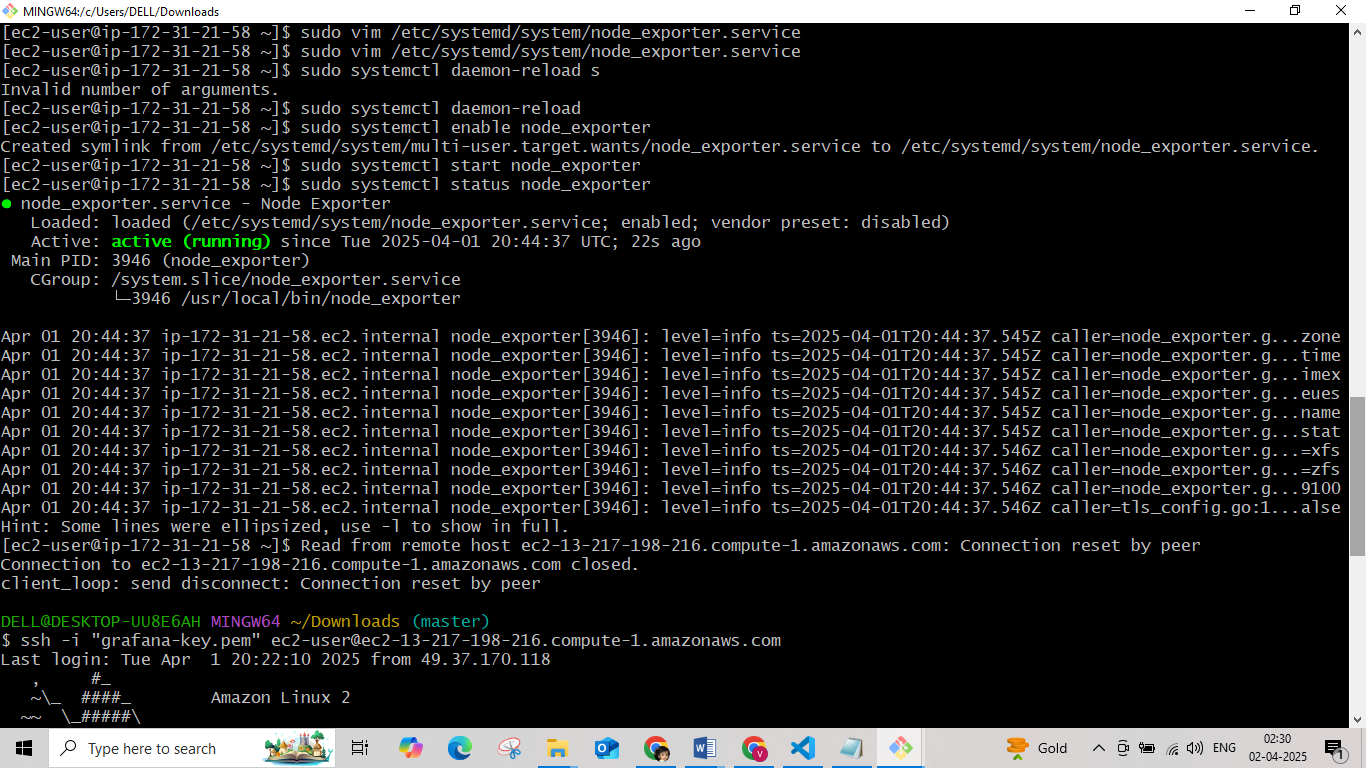


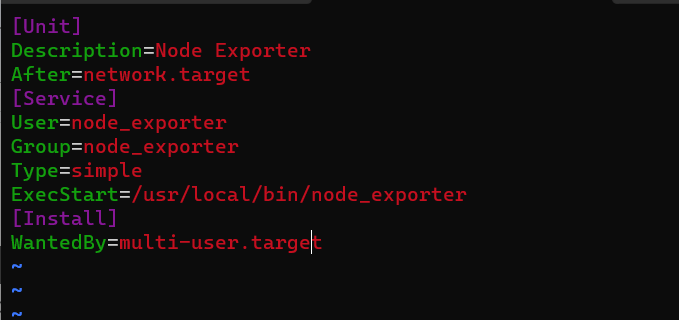
* 1. Extract the downloaded archieve
  2. Move the node\_exporter binary to /usr/local/bin and Remove the residual files
  3. Create users and service files for node\_exporter

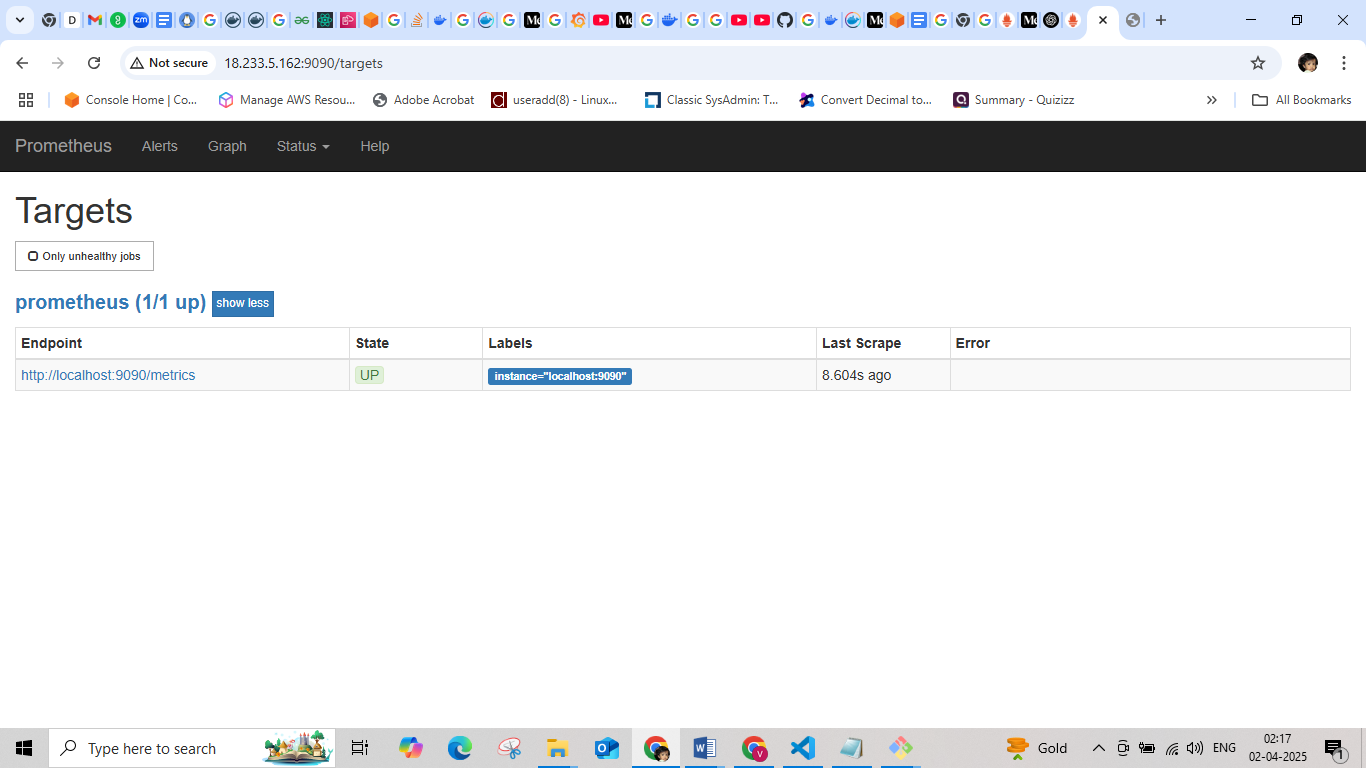


# Create a systemd unit file so that node\_exporter can be started at boot.

# Since we have created a new unit file, we must reload the systemd daemon, set the service to always run at boot, and start it





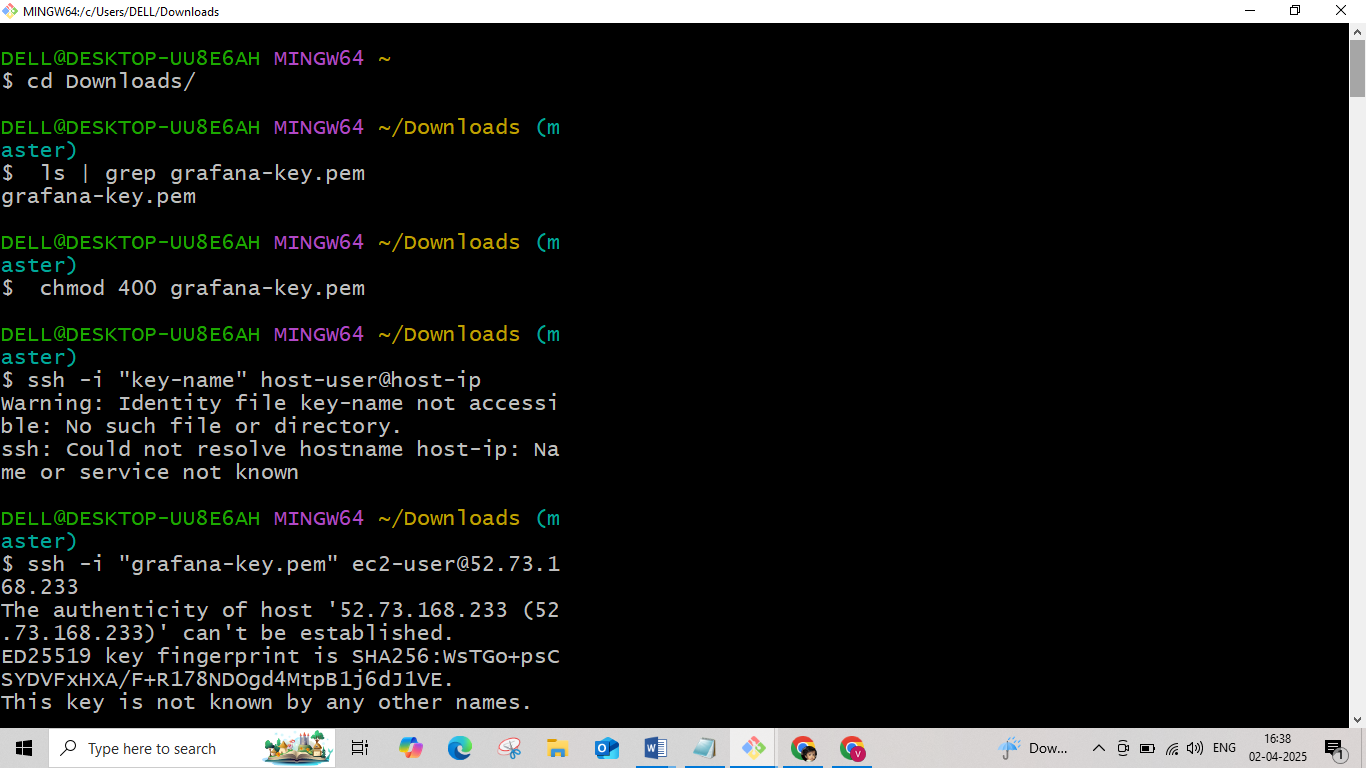


Check the node exporter by http://<ip\_address\_nodeexporter \_instance>:9100

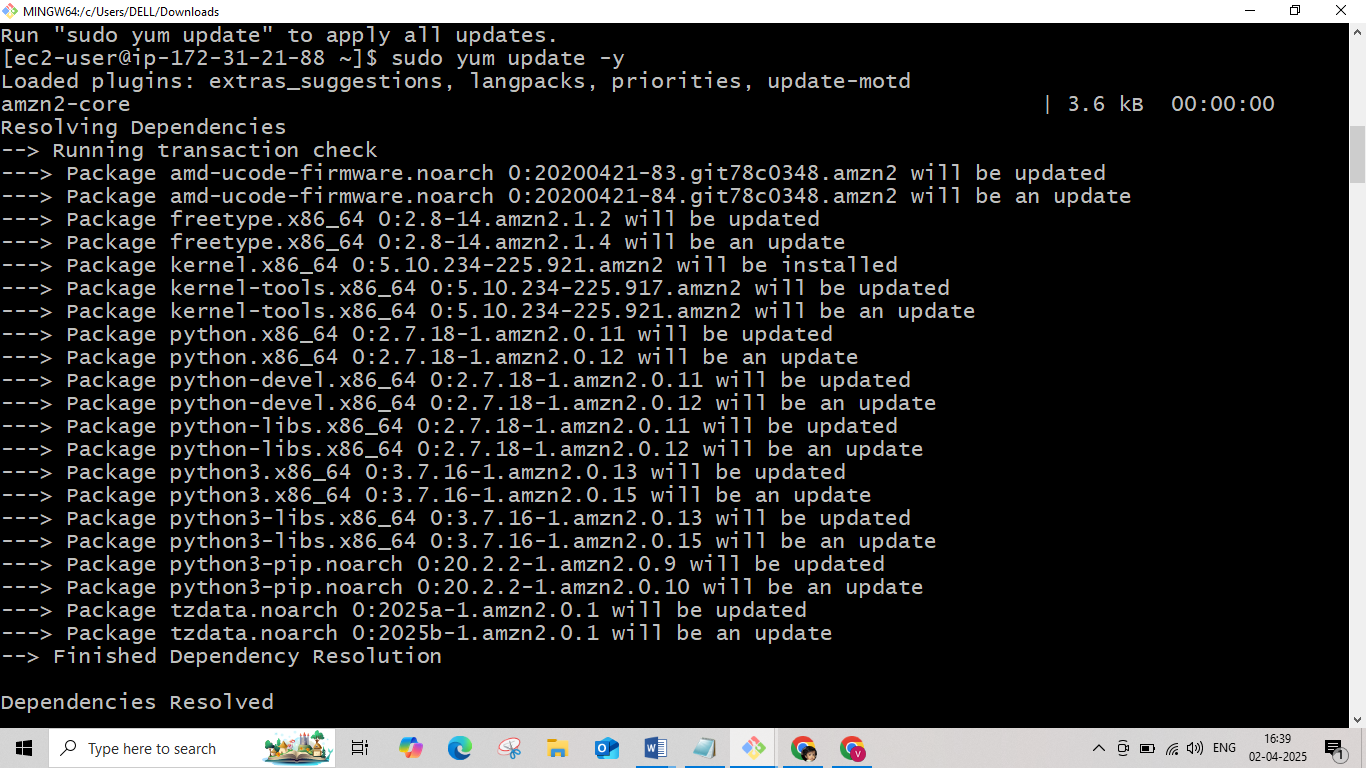


Intall Grafana

1. Create an EC2 instance for Grafana and connect it via SSH



1. Update all install packages

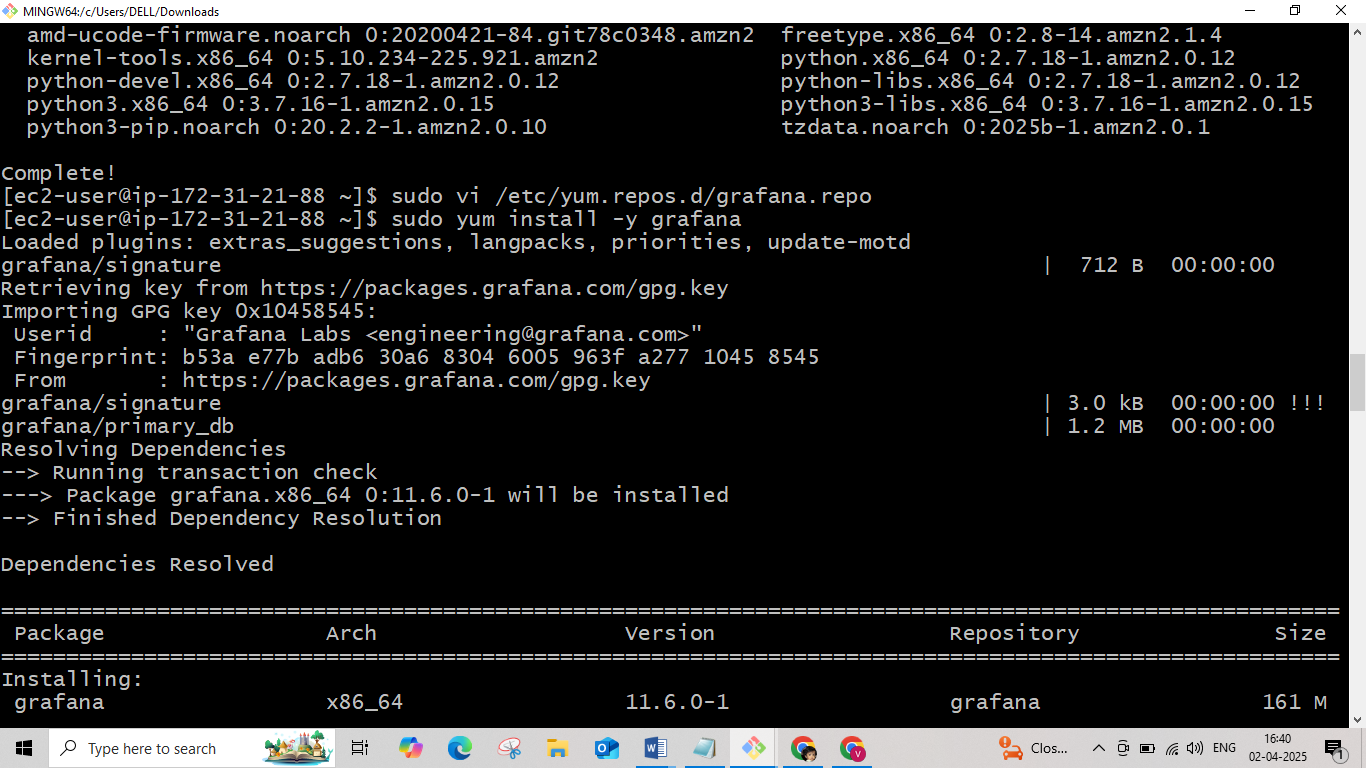


# Add a new YUM repository for the operating system to know where to download Grafana

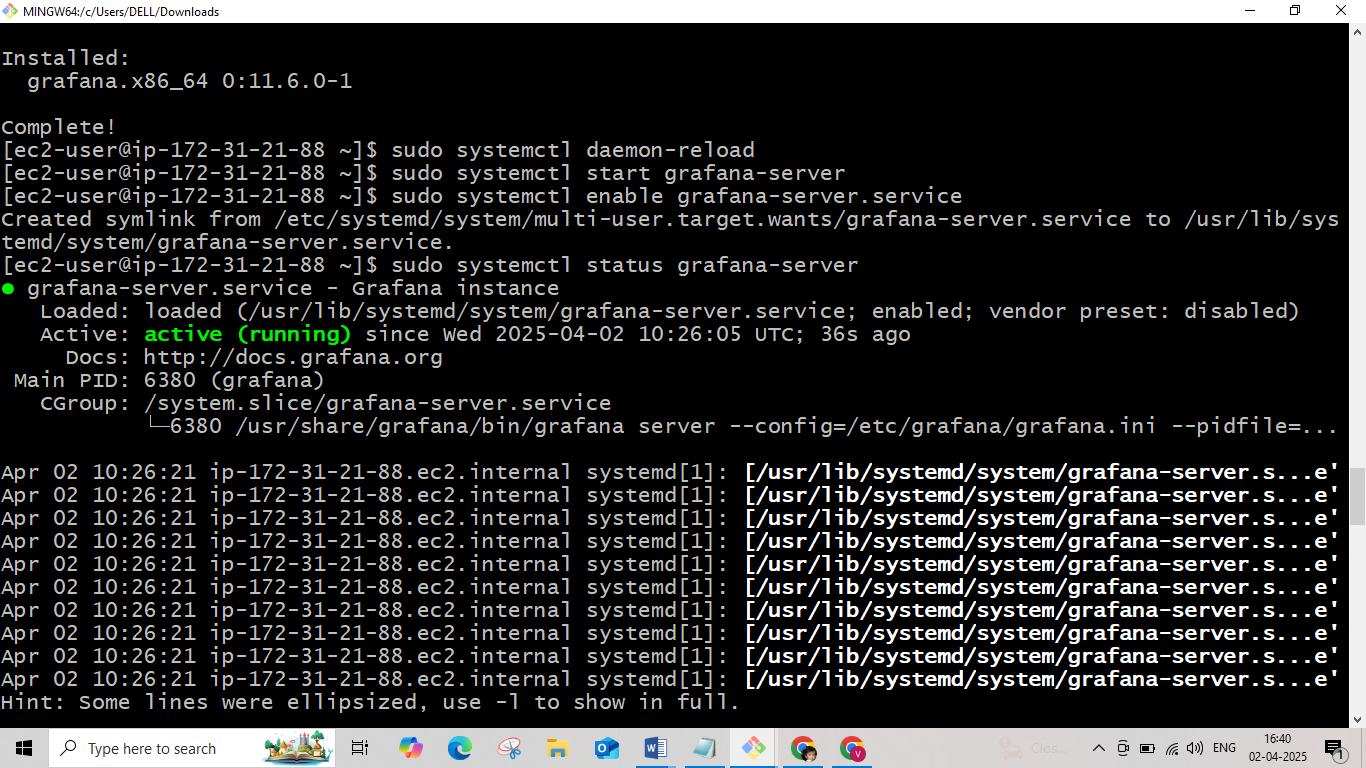
# Add the linesto grafana.repo. This setting will install to the Open Source version of Grafana

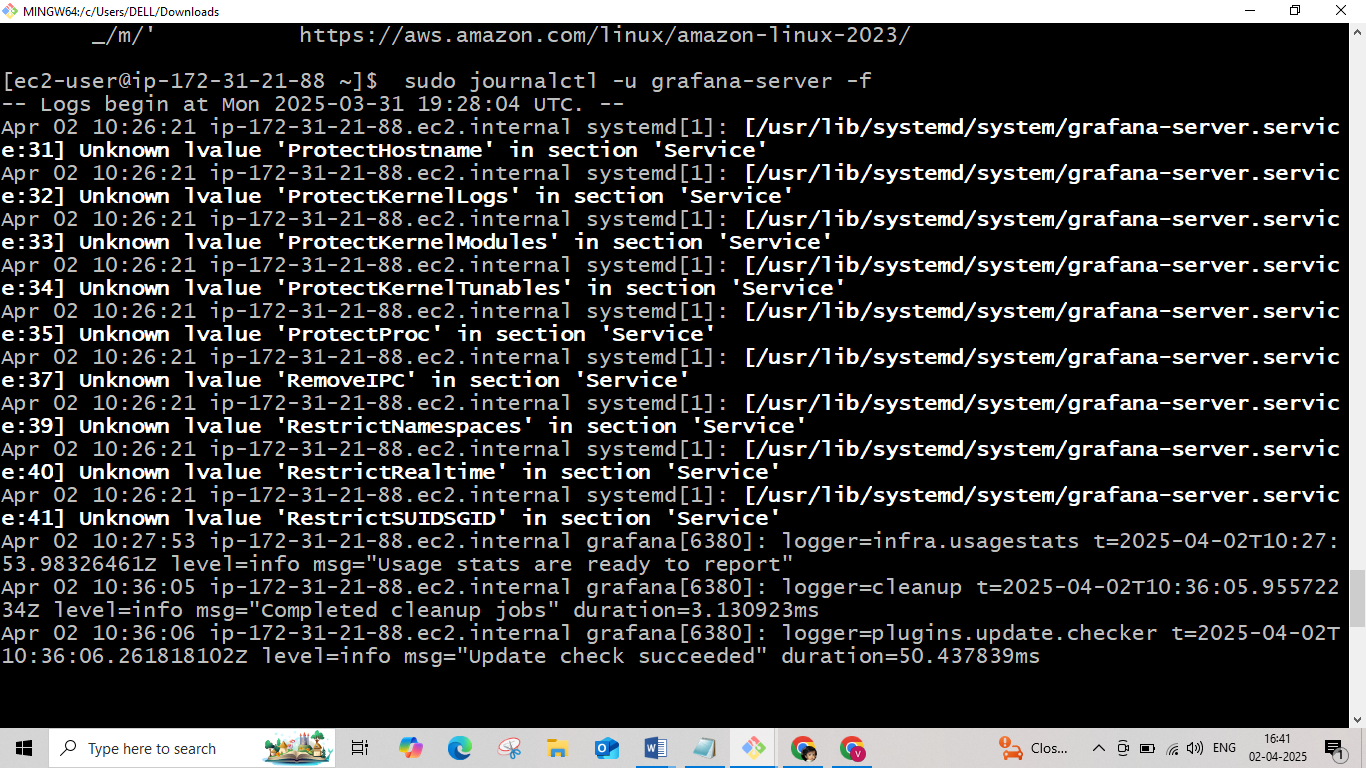
# Install Grafana

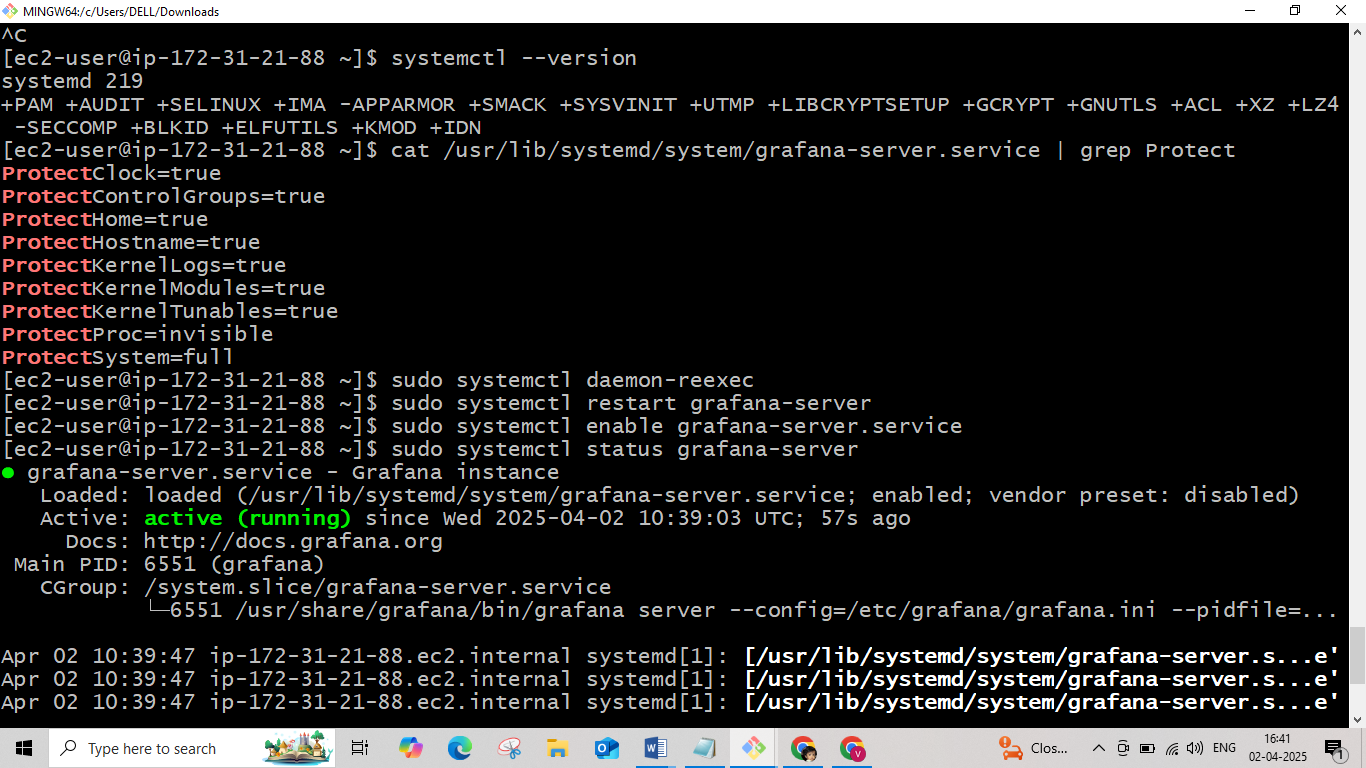
# [grafana] name=grafana baseurl=https://packages.grafana.com/oss/rpm repo\_gpgcheck=1 enabled=1 gpgcheck=1 gpgkey=https://packages.grafana.com/gpg.key sslverify=1 sslcacert=/etc/pki/tls/certs/ca-bundle.crt



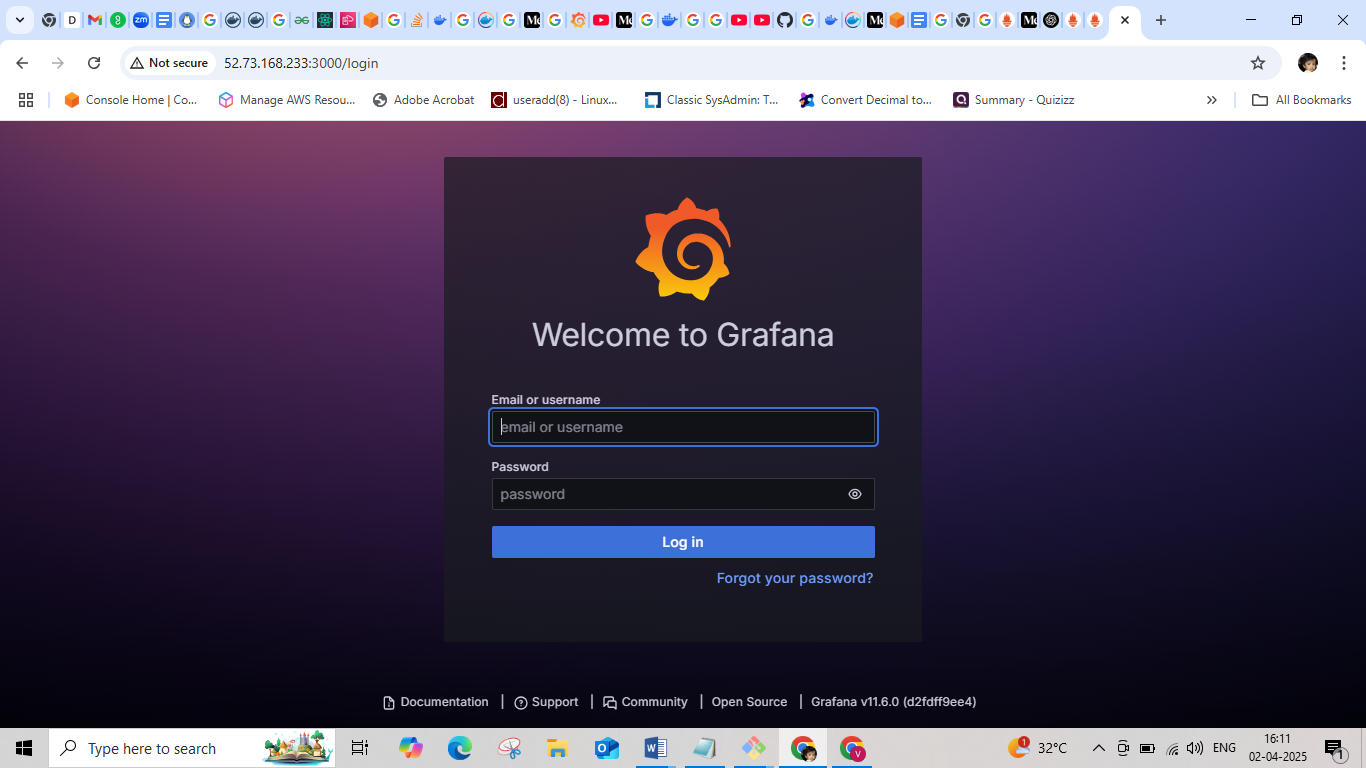
# Reload the systemd to load the new settings. Start the Grafana Server, then check for its status



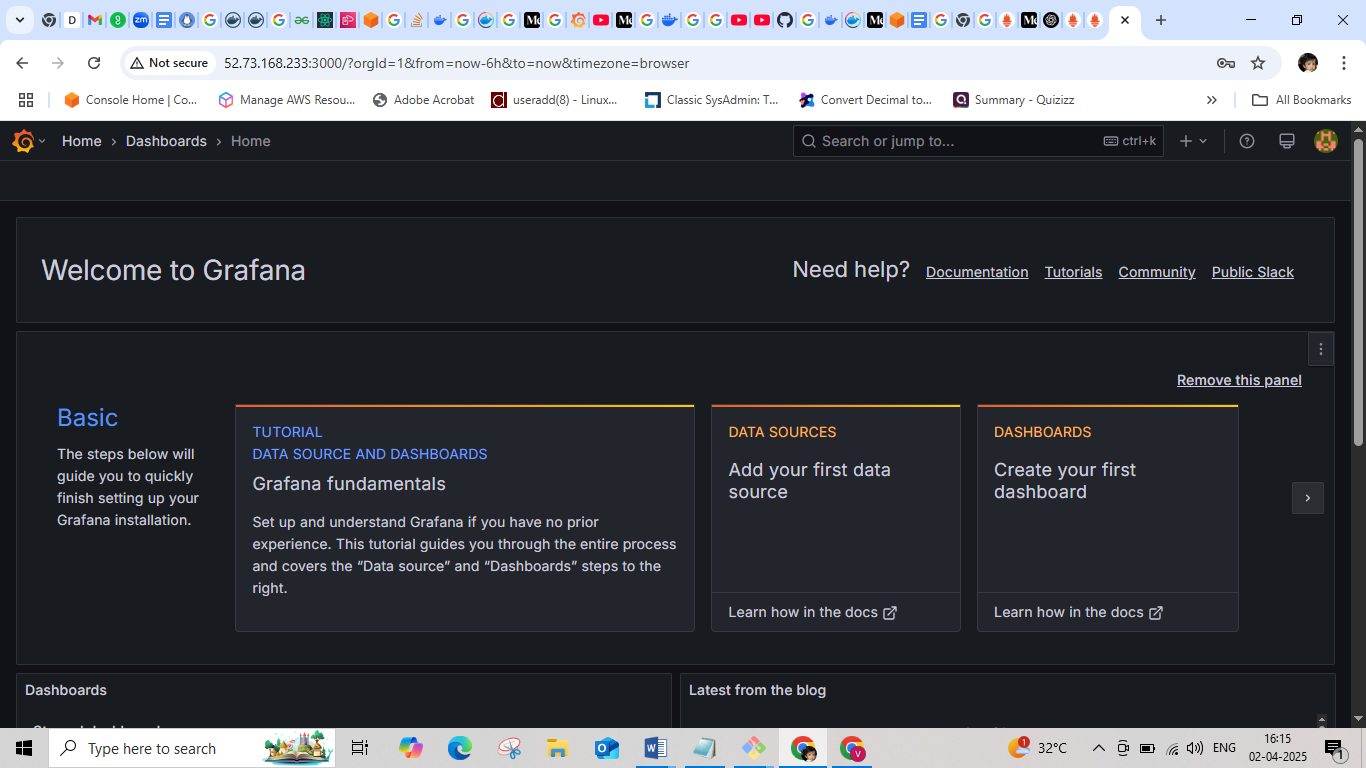




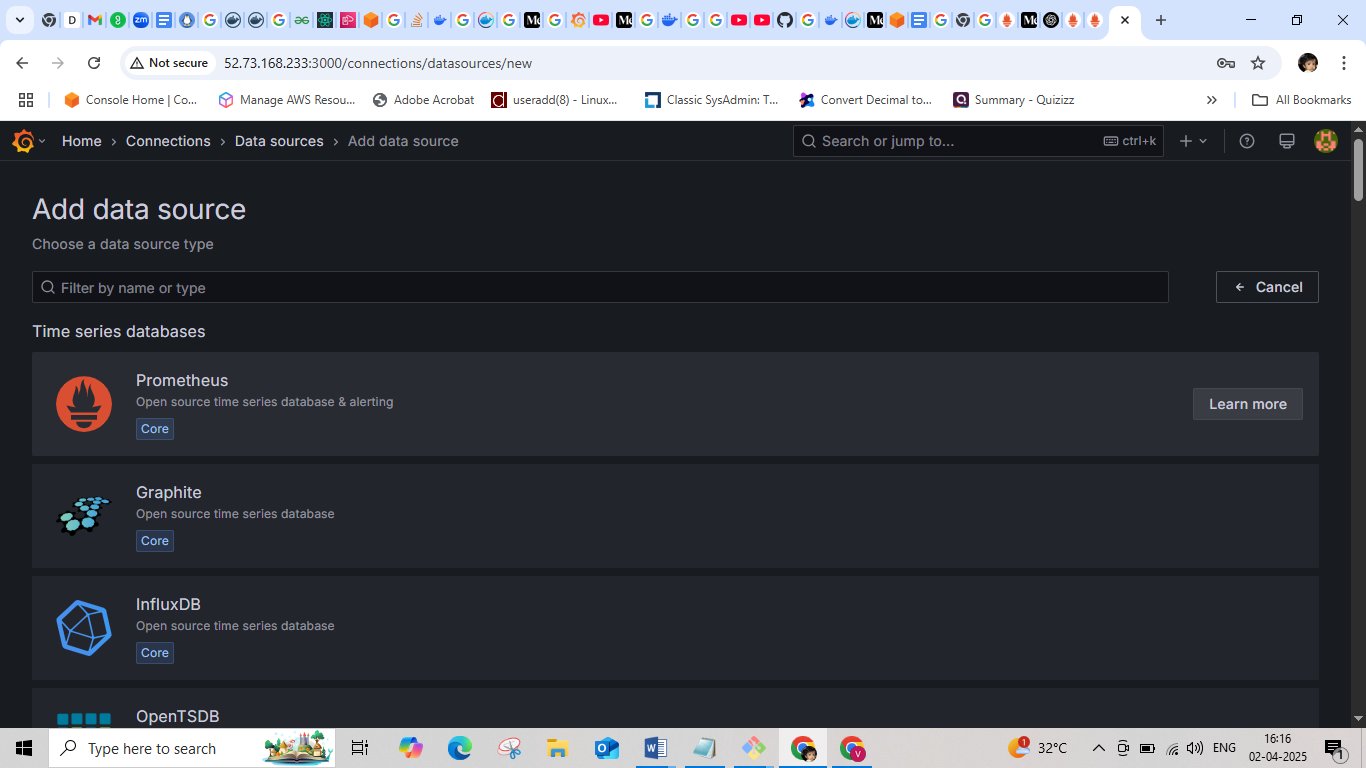
Connect to browser http://<ip\_address\_grafana>:3000



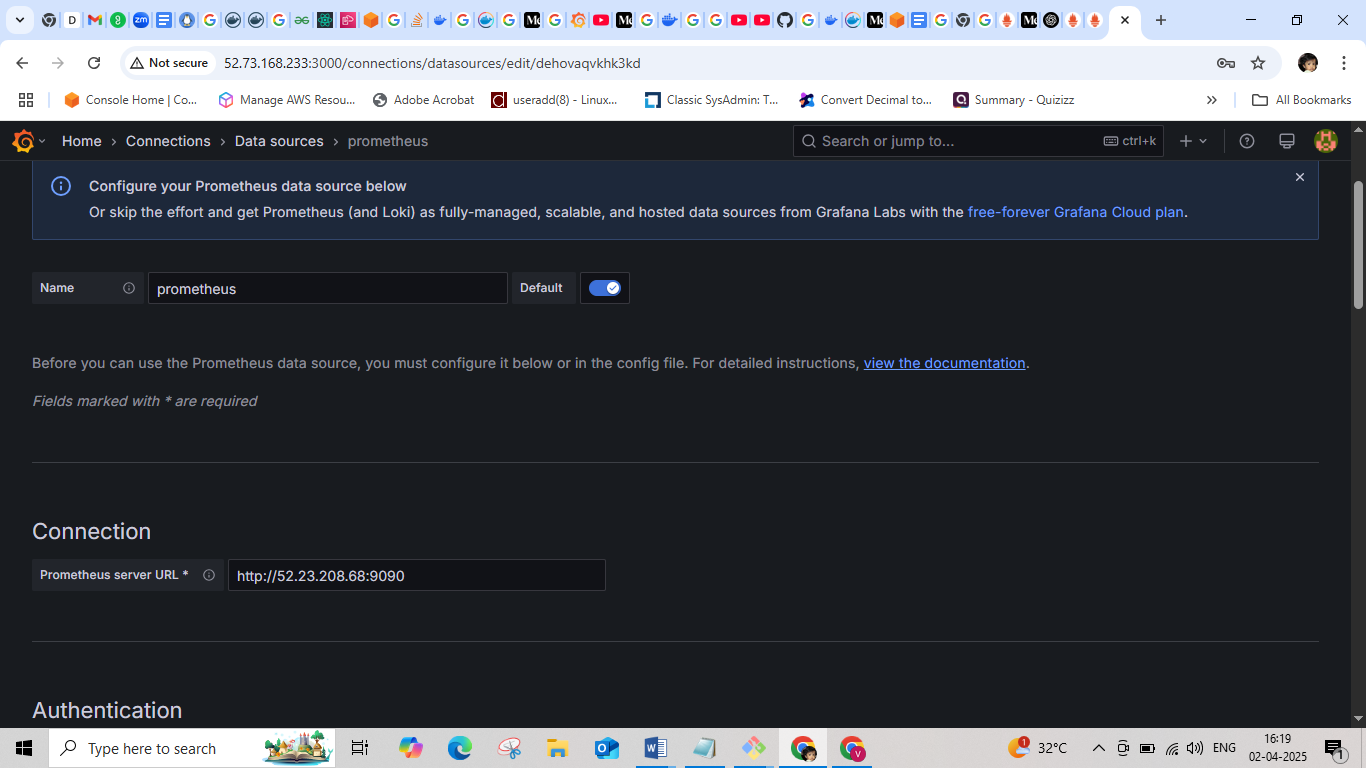
Add Data Sources



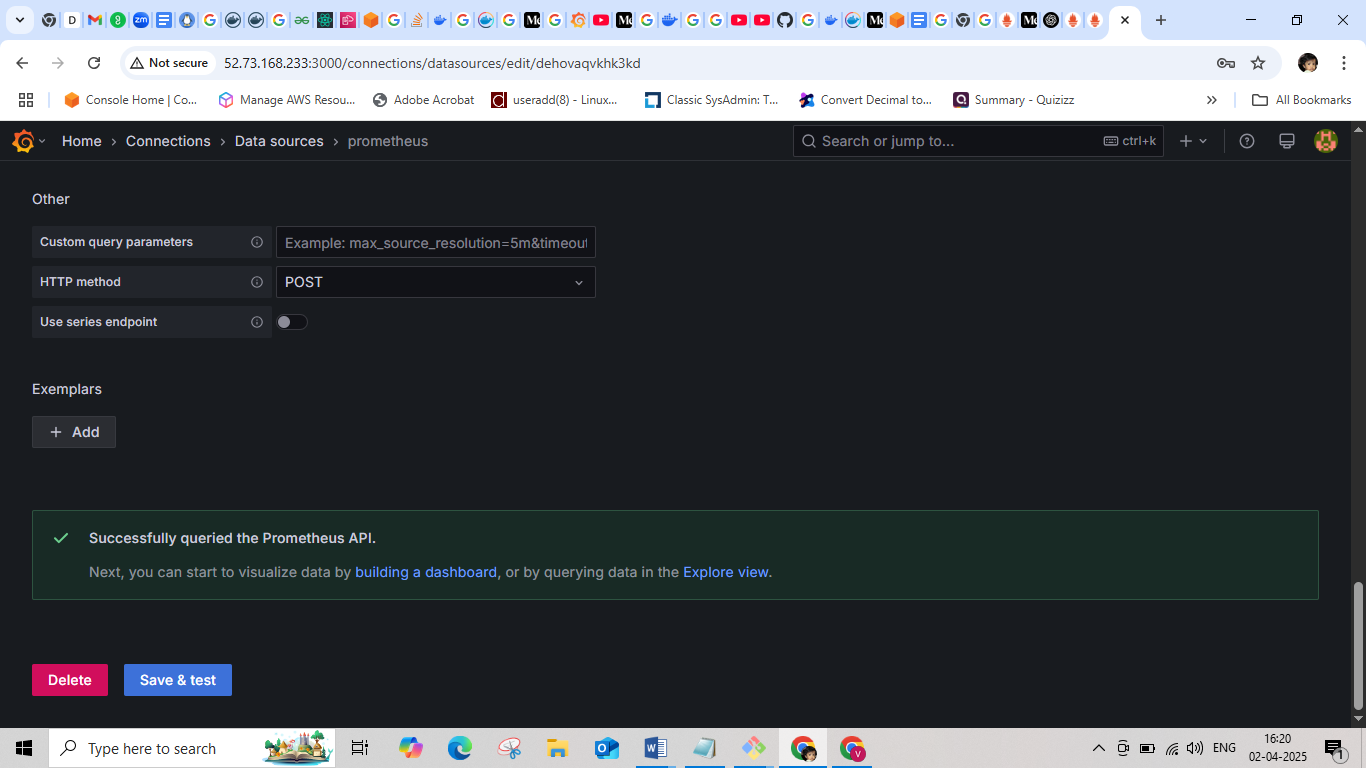
Select Prometheus



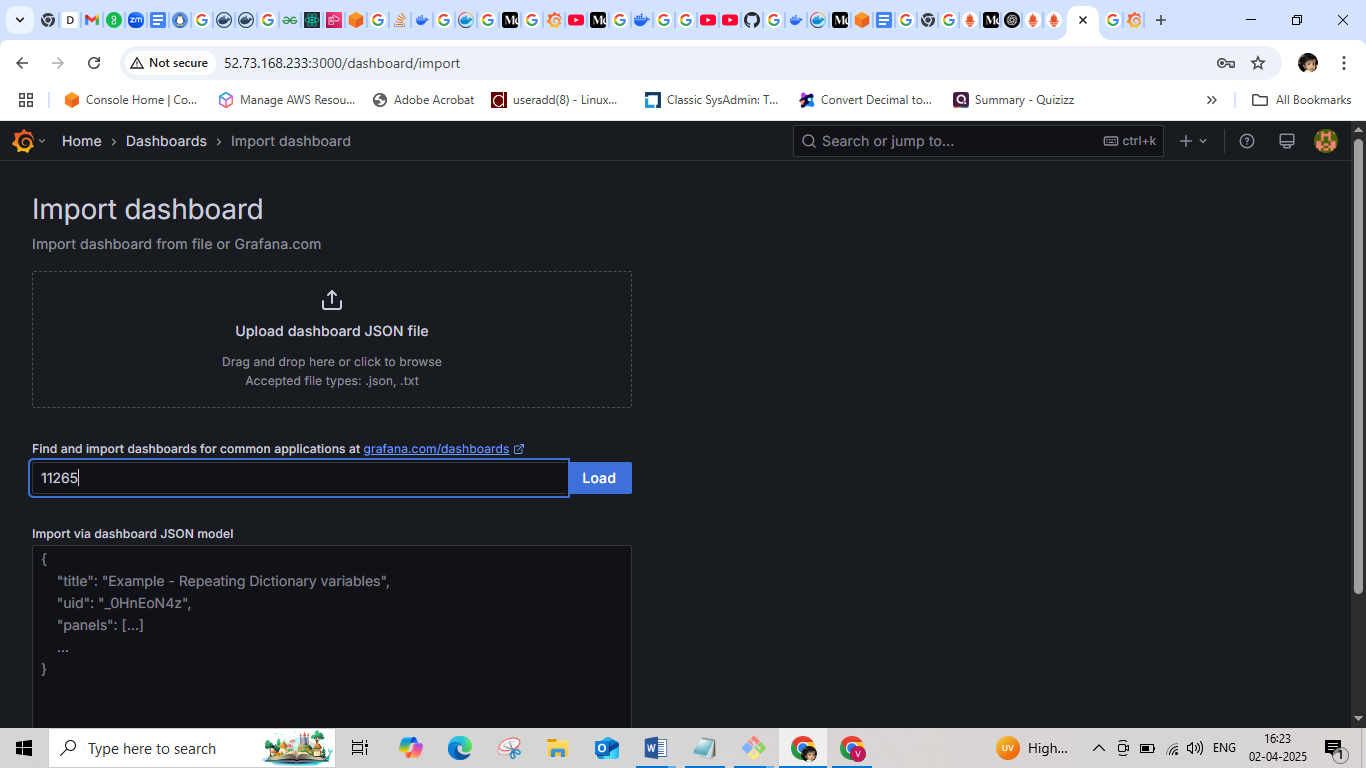
In connection type the url of Prometheus



Add the Prometheus URL as http://Ip\_address of prometheus instance:9090 then save and test it.



Click import dashboard copy the Clipboard id that we want from grafana dashboard



Import the dashboard

