PROJECT -3

Integrate Grafana with Linux Server for high cpu utilization and create a graph in Grafana

What is Grafana? — Overview

Today, almost every application stack usually consists of a number of different applications, each performing a specific role and working together towards a common goal. This is the case whether it be that of a Fortune 500 company or a computer science student trying to complete a tech project.

As such, the stability and reliability of your infrastructure would greatly depend on the performance of each application within that infrastructure.

Understanding the state of these applications is vital to keeping your systems in the best shape possible. This is done by monitoring our applications, in other words, collecting data (usually in the form of time series data) about these applications that would inform you of their current state at all times.

The larger your stack, the more applications you need to monitor to ensure your infrastructure is healthy. This would lead to collecting large amounts of performance data on your application. You would then require a massive amount of effort to parse through and make sense of this data, which is not a pretty task without the right tool.

Grafana was made exactly for this purpose!

1 TableofContents

1 Introduction

Introduction

Grafana is an open-source analytics and interactive visualization web application used for monitoring application performance. It allows users to ingest data from a wide range of sources, query and display it in customizable charts, set alerts for abnormal behavior, and visualize data on dashboards. Its open-source nature, however, requires users to manually maintain their Grafana instance, a task that is handled by MetricFire's Hosted Grafana offering, which provides all the functionality with none of the management overhead.

1.1 KEY TAKEWAYS

1. Grafana is an open-source analytics and interactive visualization web application that allows users to ingest data from various sources, query this data, and display it on customizable charts for easy analysis.
2. Grafana comes with a variety of visualization options to help users view and understand their data, which are split into "panels" that are then used to build the Grafana dashboard.
3. The platform has built-in support for a wide range of notification channels, including email, Slack, PagerDuty, and more, and allows users to set alerts to be notified of abnormal behavior.

Alerts

If you're using Grafana Alerting, then you can have alerts sent through a number of different alert notifiers, including PagerDuty, SMS, email, VictorOps, OpsGenie, or Slack.

Alert hooks allow you to create different notifiers with a bit of code if you prefer some other channels of communication. Visually define alert rules for your most important metrics.

What features does Grafana have?

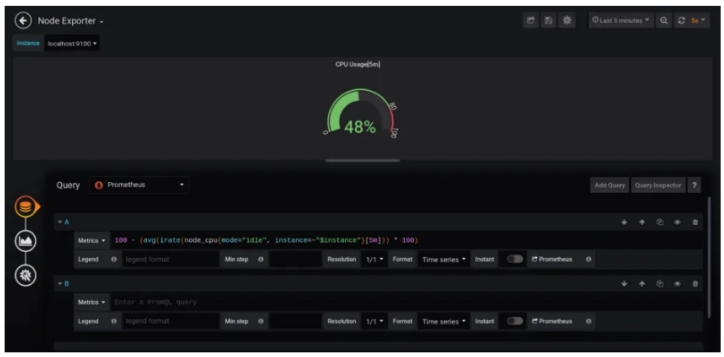
Grafana comes with a plethora of features that provide value straight out of the box. These features are the reason Grafana is arguably one of the most popular visualization software available for metric monitoring, simply because of their ease of use:

1. Visualization

Grafana possesses a huge variety of visualization options to help you view and understand your data easily. These options are split into "panels" which are then used to build the Grafana dashboard.

A panel is the most granular visualization building block in Grafana and is used to display data that has been queried from the data source attributed to that panel. For easier understanding, think of a panel as a space on the dashboard that houses a specific type of visual portrait of information.

This information is being pulled from the data source attributed to that panel and can be a type of graph (gauge, histogram, bar chart, etc.), or logs and alerts. For example, one could create a gauge panel with its data source set to Prometheus, and then query CPU usage data stored in Prometheus to be displayed on this panel. Such a panel would look like this:



You could stack up panels like this to make up a dashboard, with each panel displaying its own piece of information in the manner that best suits you. Worthy of note is the fact that although Grafana ships with some pretty cool panels, you could design and add your own panels using plugins — the power of Open Source.

Check out these MetricFire articles (Grafana plugins and Grafana plugins explained) on how to set up Grafana plugins. For a more in-depth look into Grafana dashboards, look no further than this MetricFire article.

2. Alerting

When monitoring applications, it is essential to be made aware the second something goes wrong or is abnormal. This is vital to keeping your systems healthy and reducing downtime. Grafana has built-in support for a huge number of notification channels, be it email, Slack, PagerDuty, etc., whichever best suits you.

In order to create an alert, you'd have to create and configure an alert rule. This rule serves as a trigger for your alert, such that whenever said rule is broken, an alert notification is sent via the alert notification channel you had configured into the rule. For a more detailed look into Grafana alerts, look no further than this MetricFire article — Grafana alerting.

3. Annotations

Grafana allows you to annotate, or in simple terms, leave notes directly on graphs. This simple but powerful feature provides a way to seamlessly mark important points on your graph. This serves as a reminder for further action in the future, to provide context to an onboarding team member, or to simply mark a special event on your graph.

Think of it as writing a sticky note and placing it directly on your graph, but this time, you don't need any sticky note papers. Further information on annotations can be found in our Grafana annotations tutorial article (yes, we have articles for everything Grafana!).

4. Open Source

Grafana is completely open source and backed by an active vibrant community. This presents some huge benefits to its users such as the flexibility to develop and publish their own plugins or use plugins developed by other people. These plugins are usually easy to install by pretty much downloading the source code and running it manually.

However, its open-source nature does pave the way for some drawbacks. For example, you would have to manually maintain your Grafana instance yourself, manually perform updates, and so on. These drawbacks are completely handled by MetricFire's Hosted Grafana offering, which provides all the functionality Grafana has to offer, with none of the management overhead.

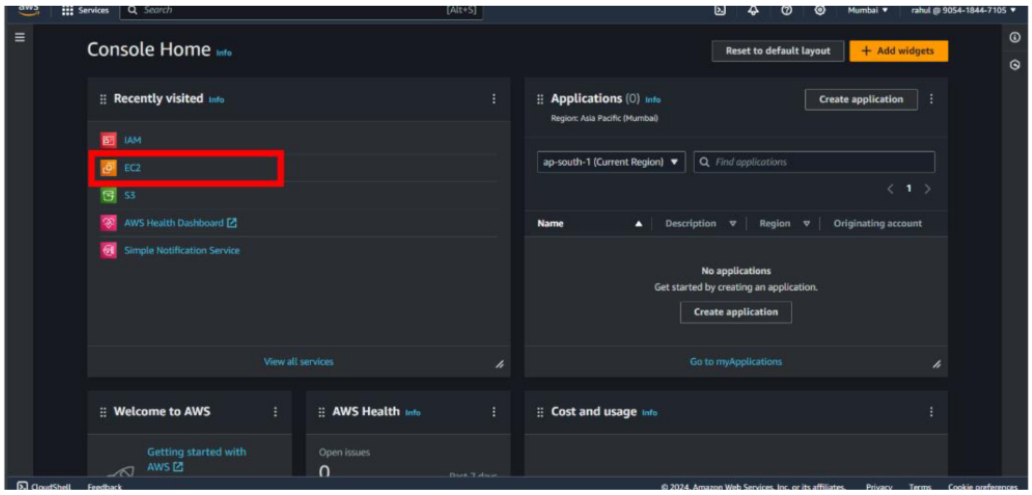
Integrate Grafana with a linux server for high cpu utilization and create a graph of Grafana

❖ Sign in to AWS Management Console:

1. Click on the Open Console button, and you will get redirected to AWS Console in a new browser tab.

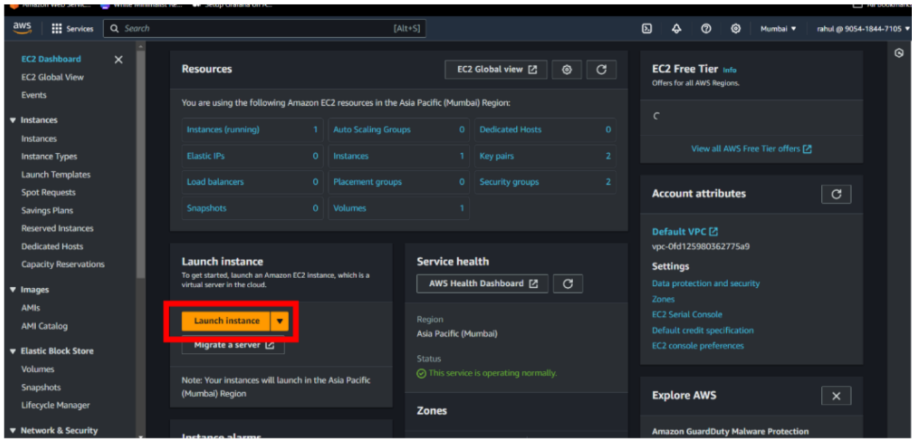
- On the AWS sign-in page, Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
- Now copy your User Name and Password in the Lab Console to the IAM Username and Password in AWS Console and click on the Sign in button.

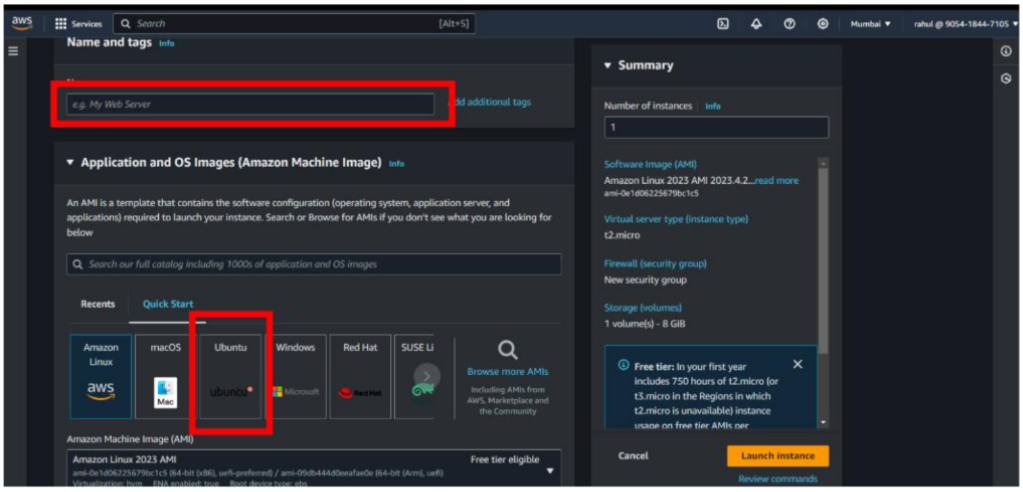
2. Once Signed in to the AWS Management Console, Make the default AWS Region as US East (N. Virginia) us-east-1.



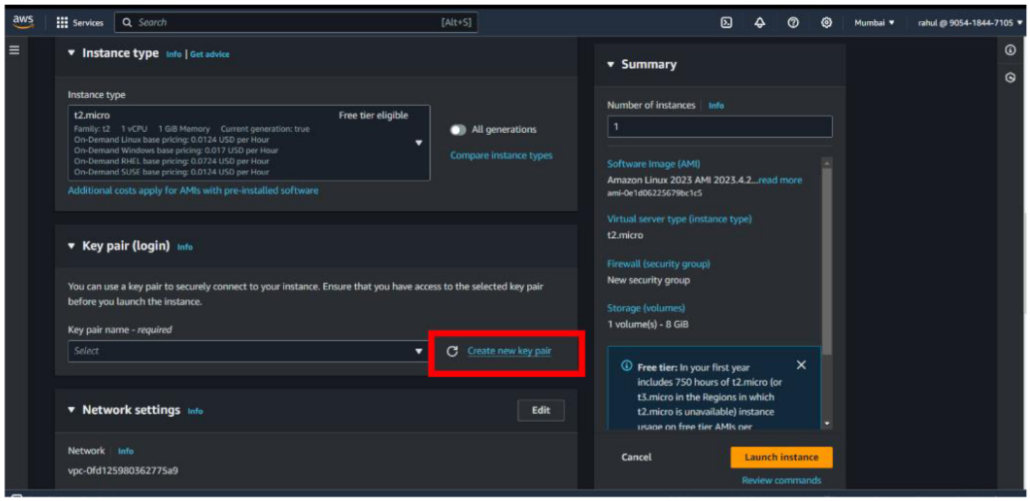
❖ Create an EC2 Instance(ubuntu):

- For creating an EC2 instance follow the following steps as shown in snapshots.

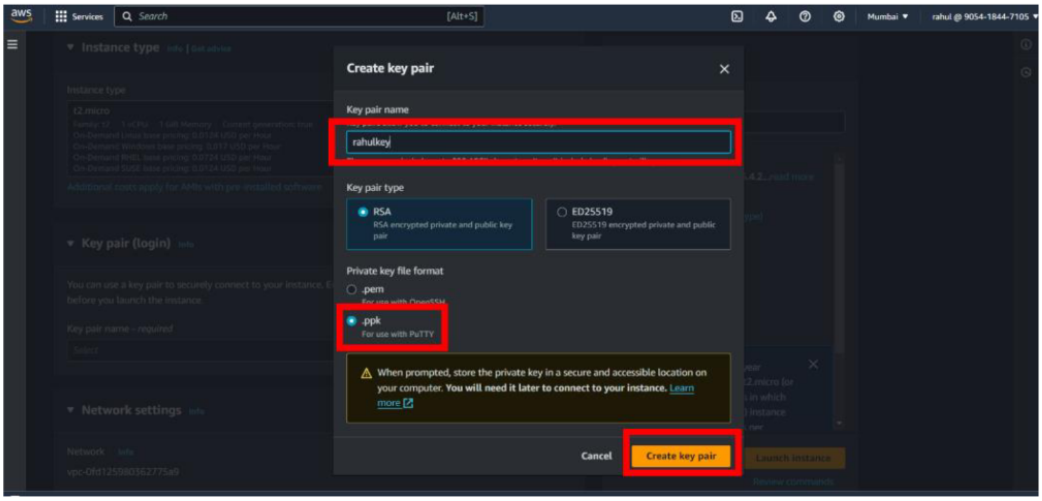




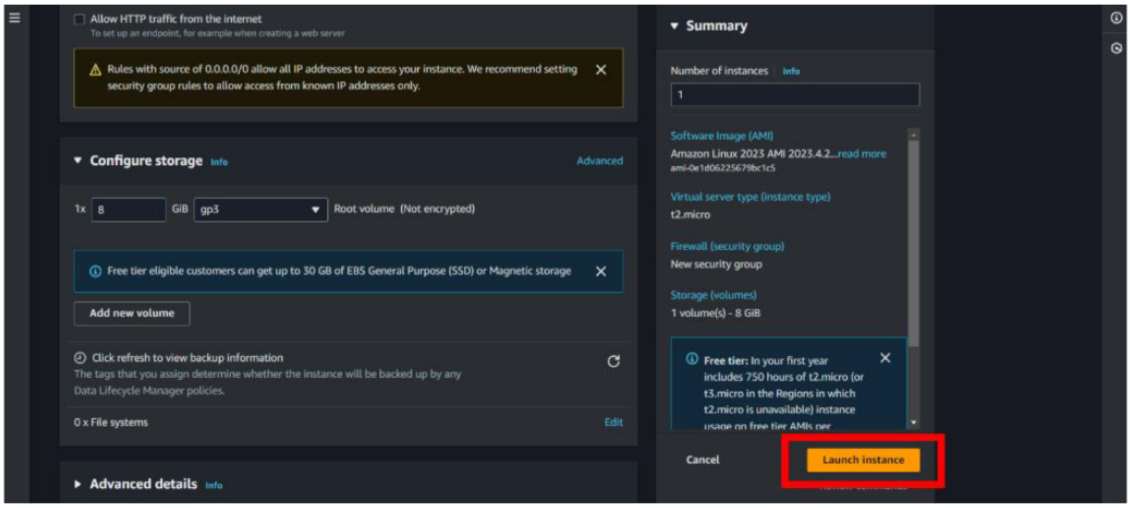
- Provide the EC2 name of your choise and select”**Ubuntu”**as an OS Image.



- Create a new key pair.

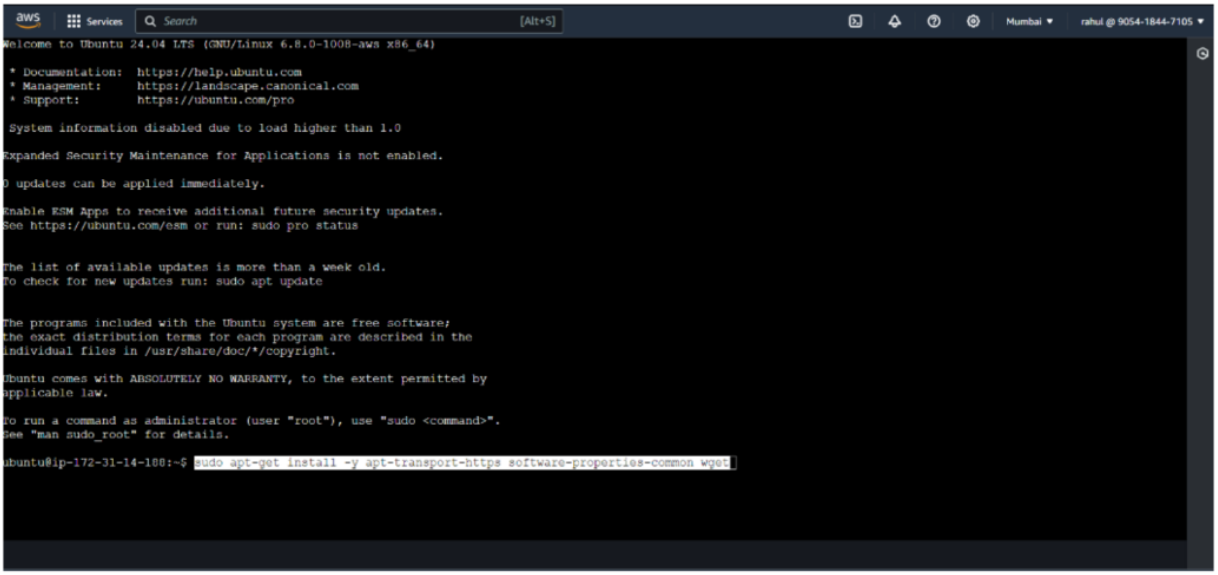


- Scroll down and click on “**LAUNCH INSTANCE** “.

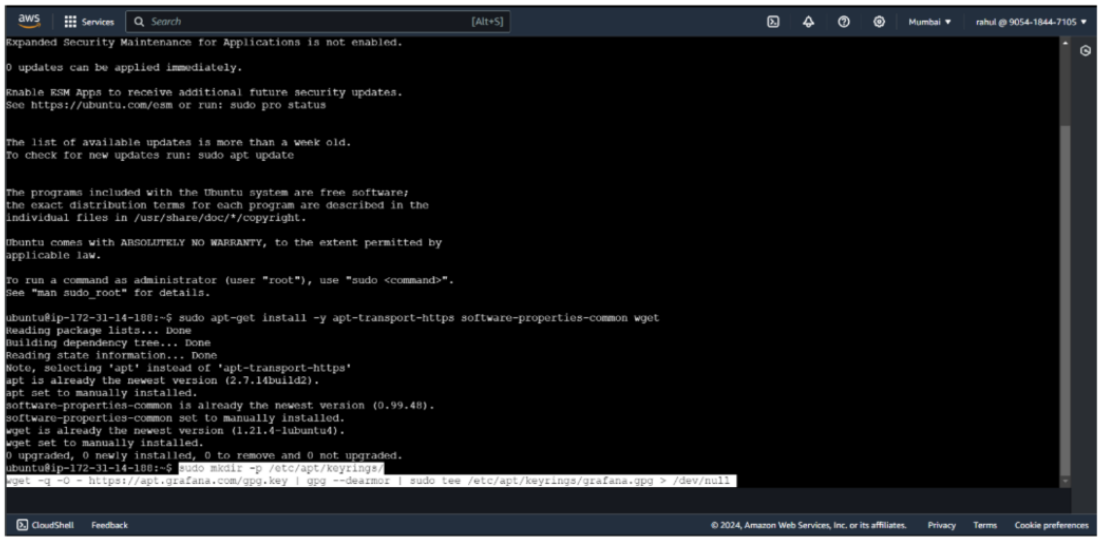


- Then open your instance and connect that instance by putty or on web browser.
- After connecting the instance follow the given command or read Grafana documentation for help.

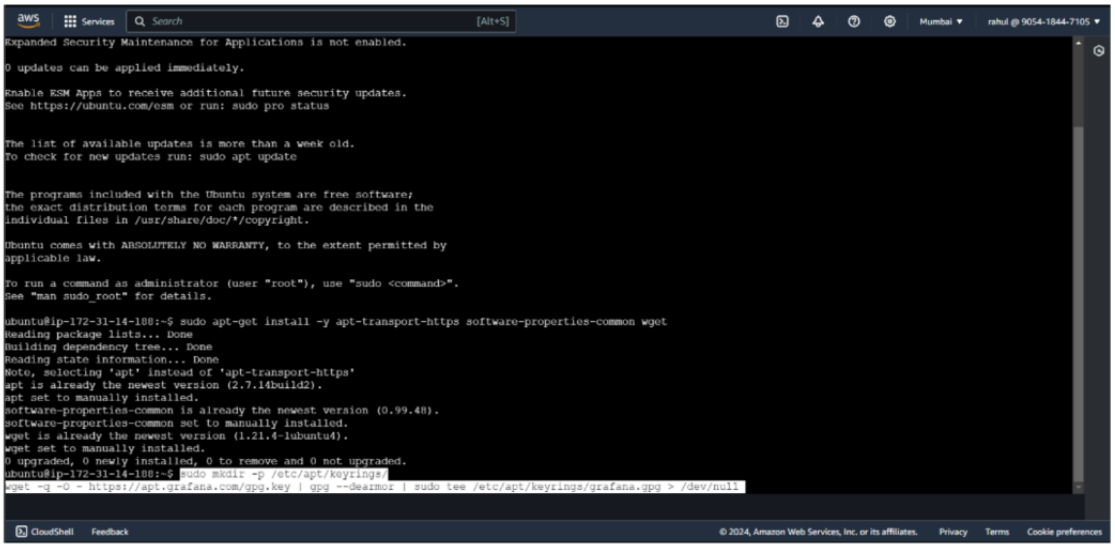
❖ Installation of GRAFANA into instance:



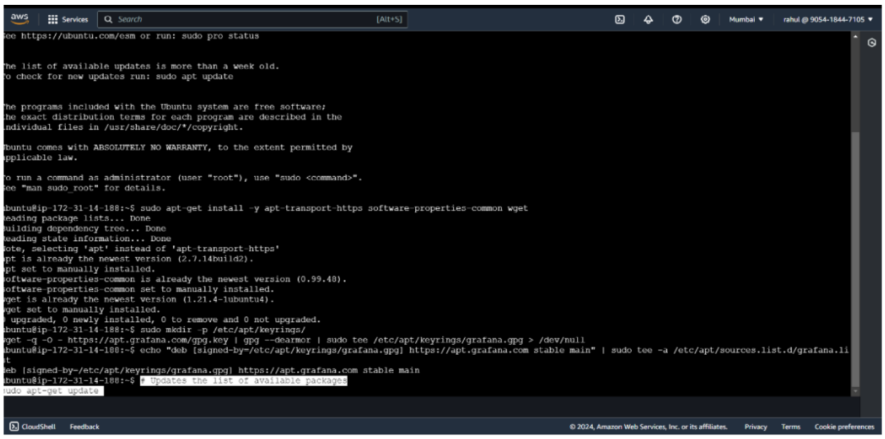
sudo apt-get install -y apt-transport-https software-properties-common wget



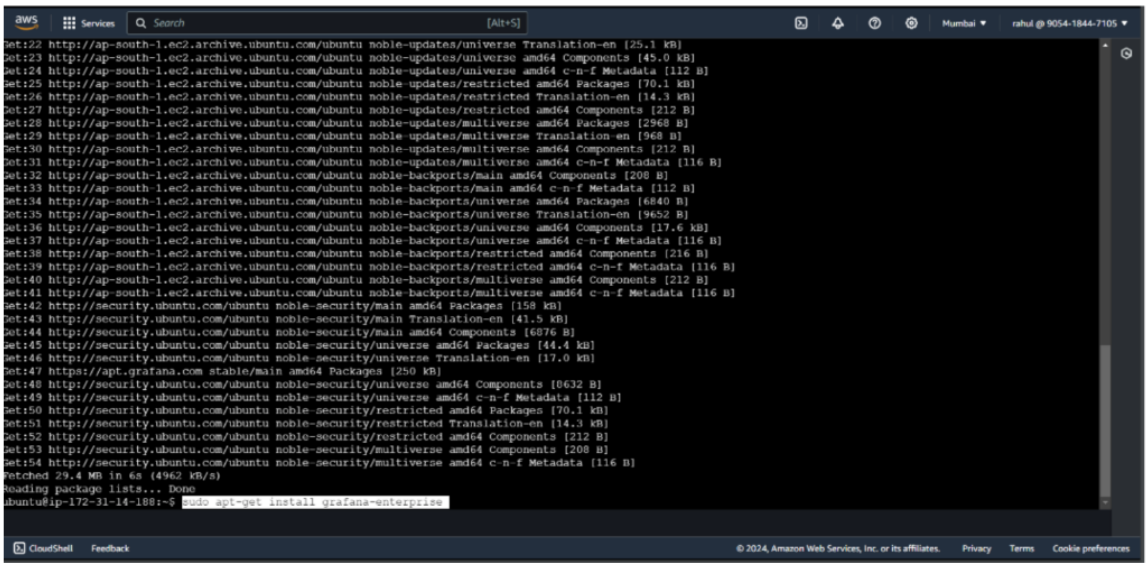
sudo mkdir -p /etc/apt/keyrings/wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null



sudo mkdir -p /etc/apt/keyrings/wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null

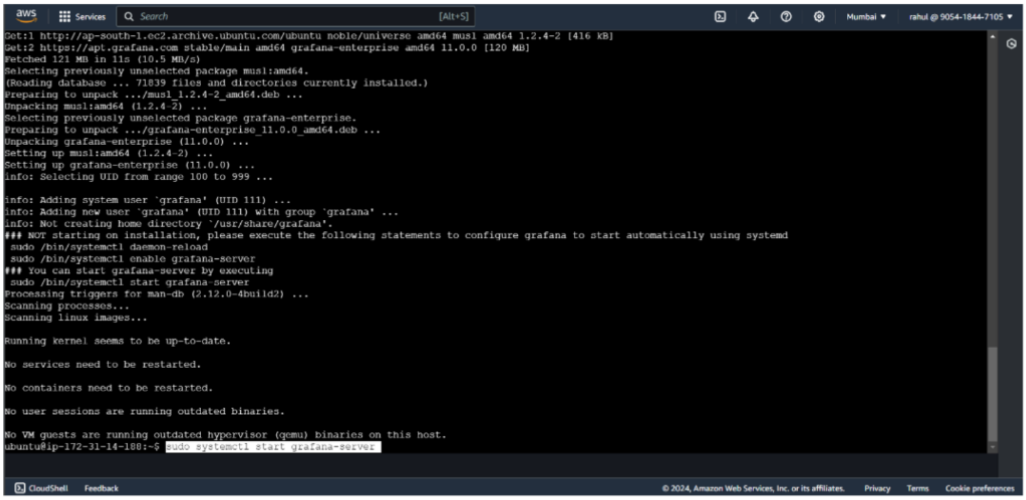


To updates the list of available packages
sudo apt-get update

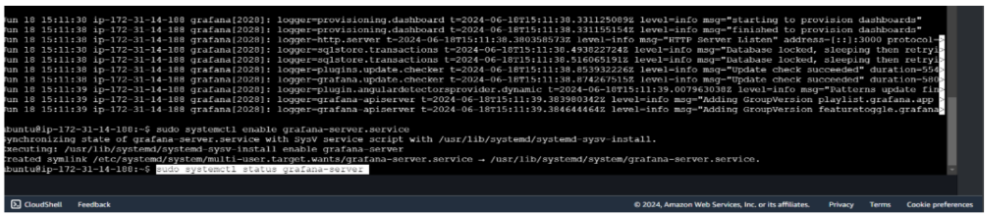


To installs the latest Enterprise release:

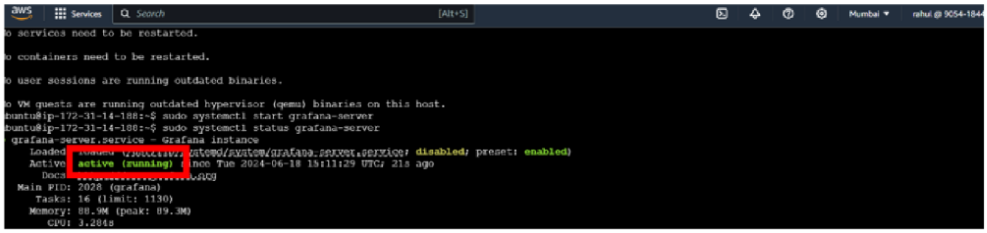
sudo apt-get install grafana-enterprise



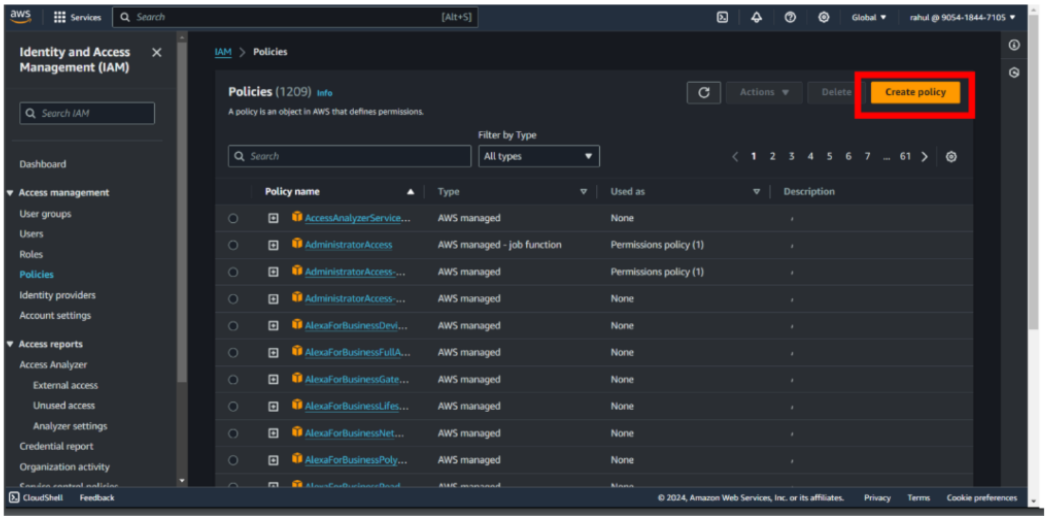
- *sudo systemctl start grafana-server*
- *sudo systemctl enable grafana-server.service*



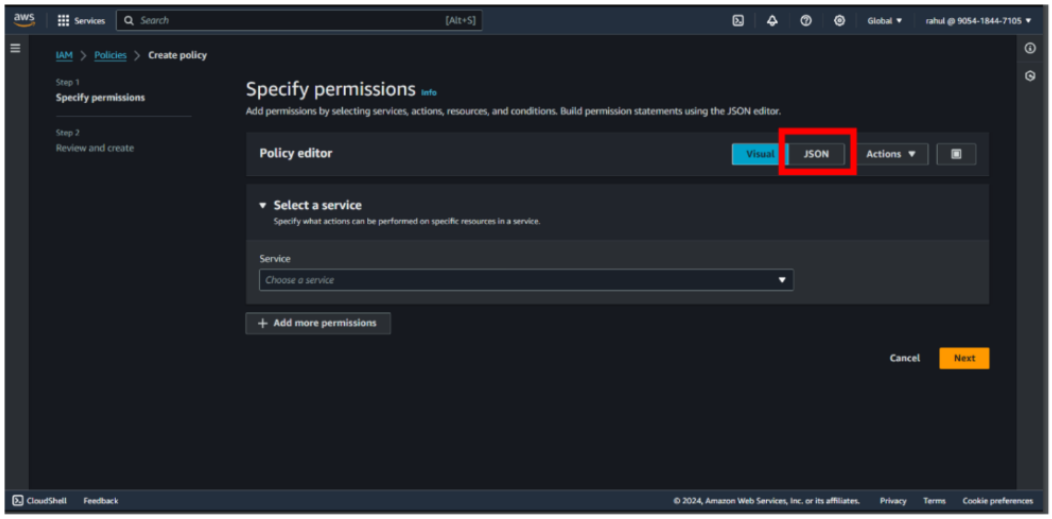
sudo systemctl status grafana-server.service



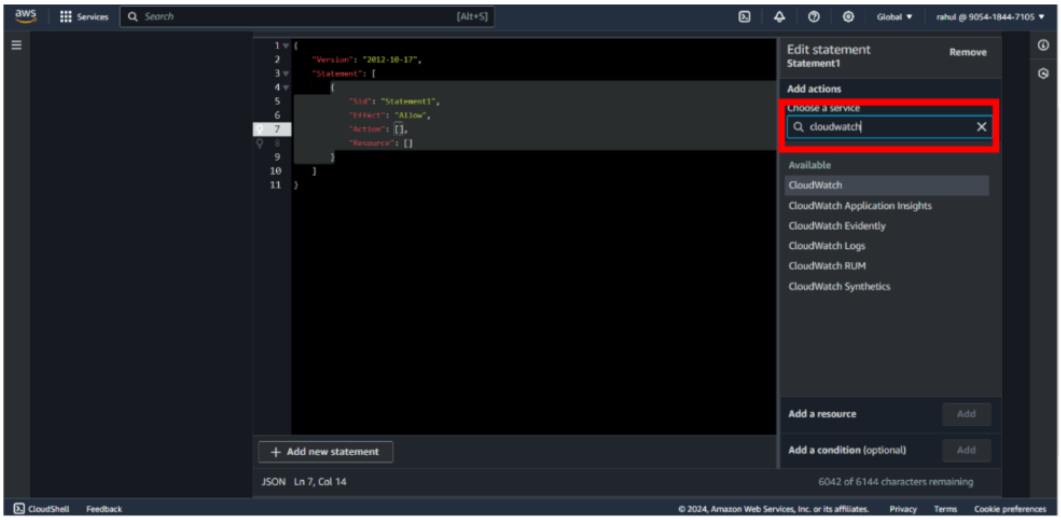
❖ Creating policy:



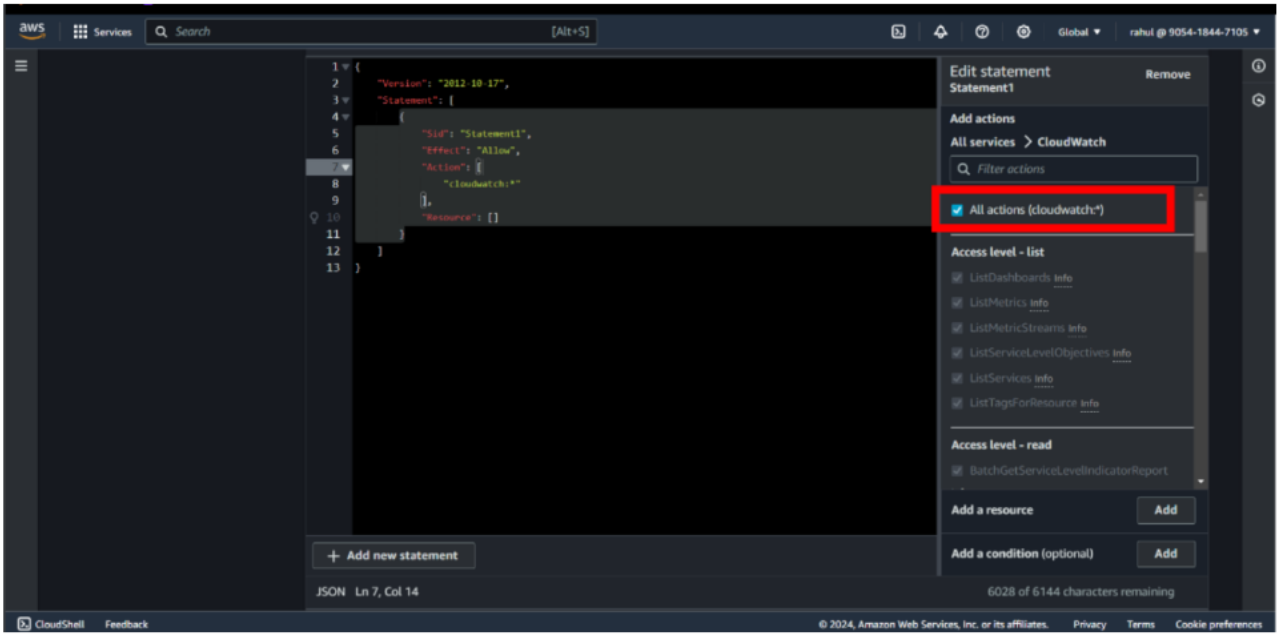
- Then search IAM role and go to policies and click on **Create policies**.



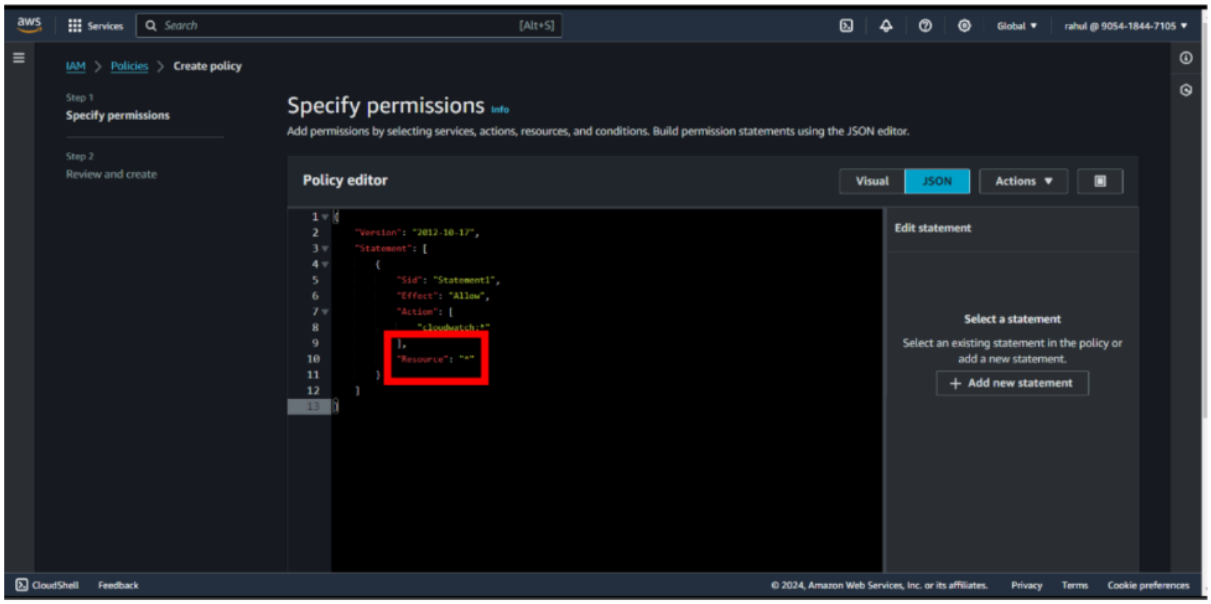
- Then go to **JSON** type.



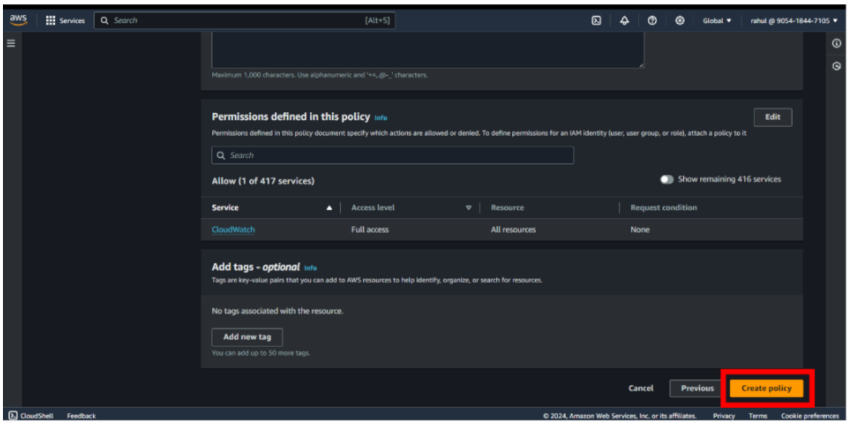
- Click on add action and search for **“Cloudwatch”**.



- After selecting cloudwatch select **“All actions “.**

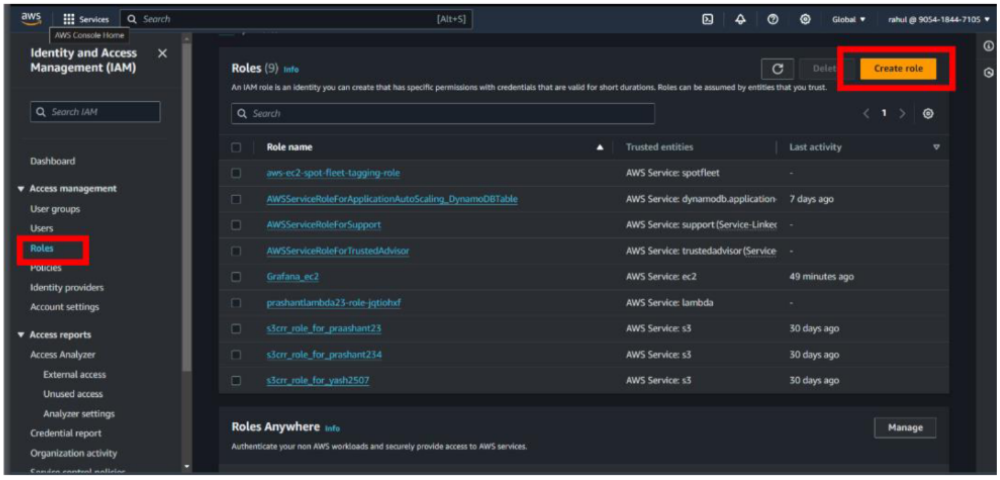


- Then put **“Resource “ : ”*”**

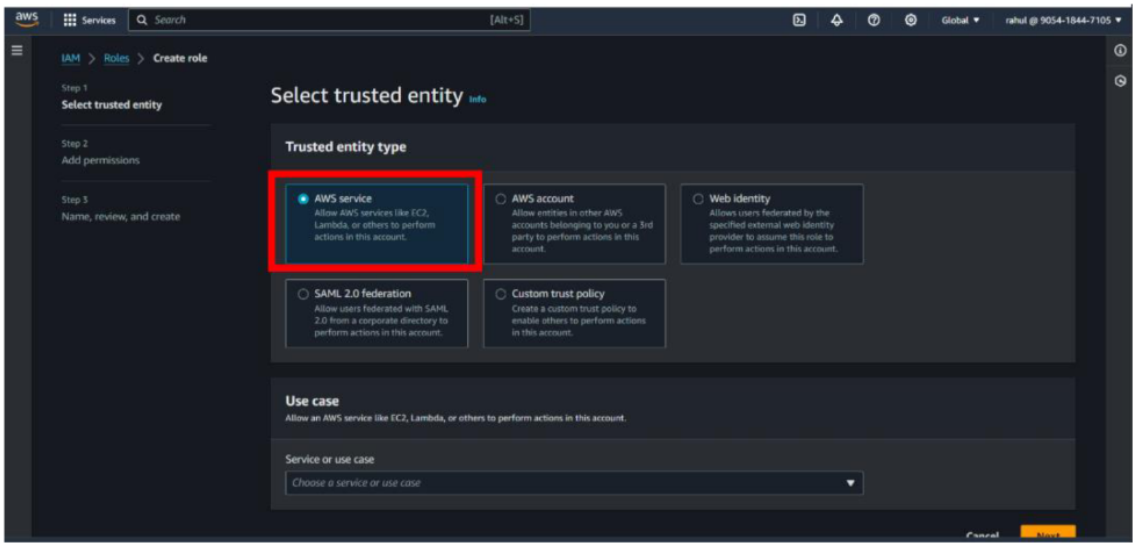


- Scroll down and click on **Create policy.**

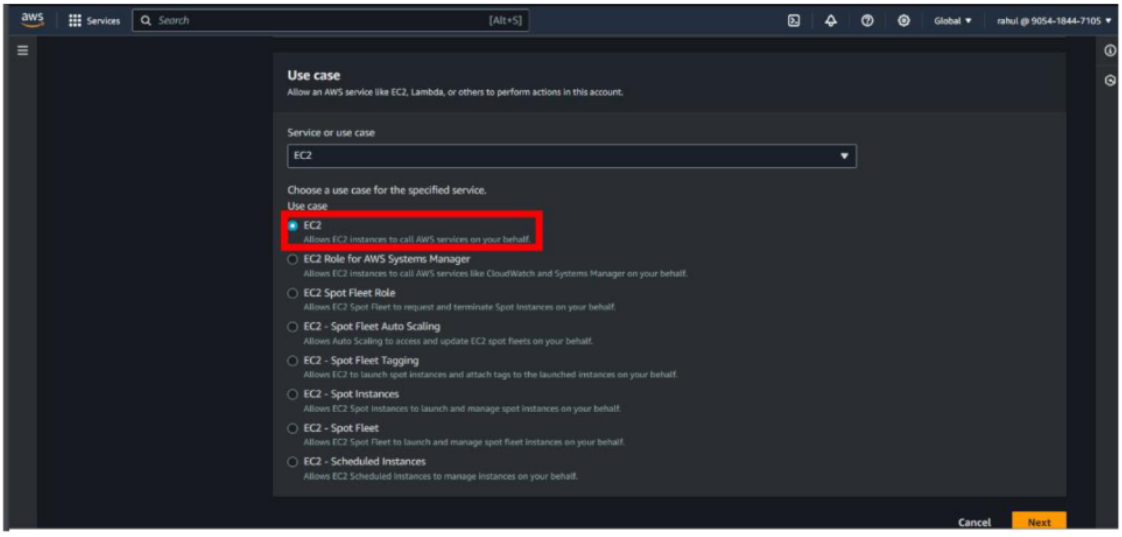
❖ Creating role :



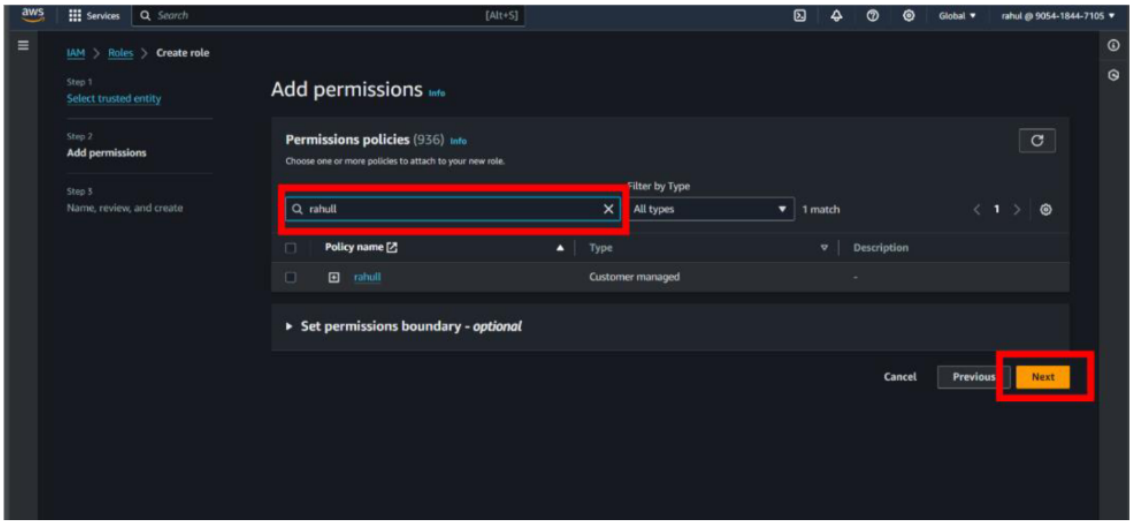
- Then go to roles and click on **Create roles.**



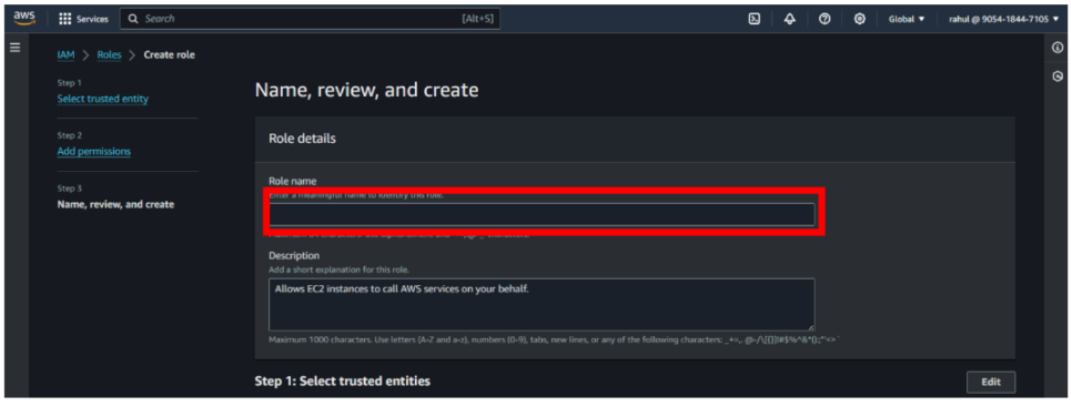
- Then select entity type is **AWS services .**



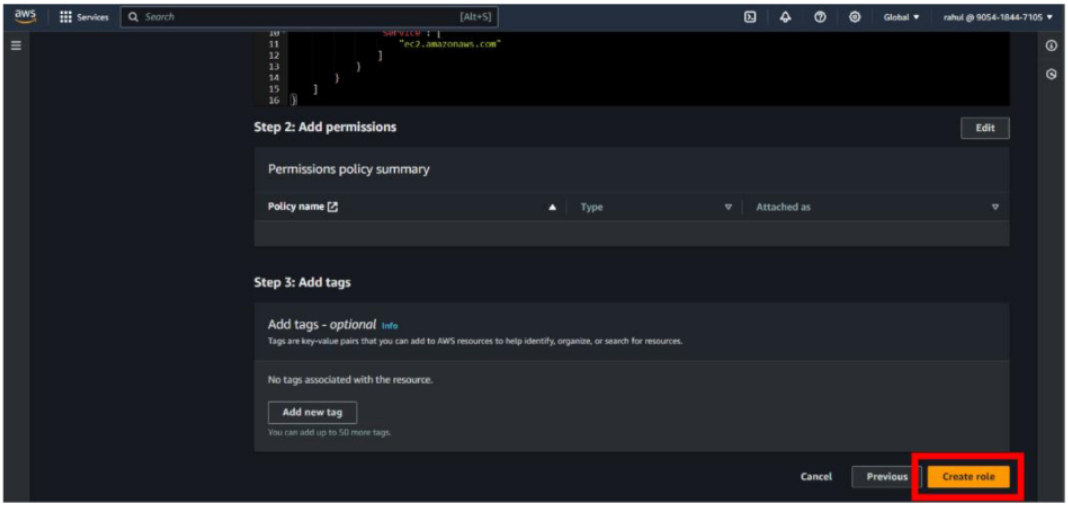
Now , select use case as **EC2 .**



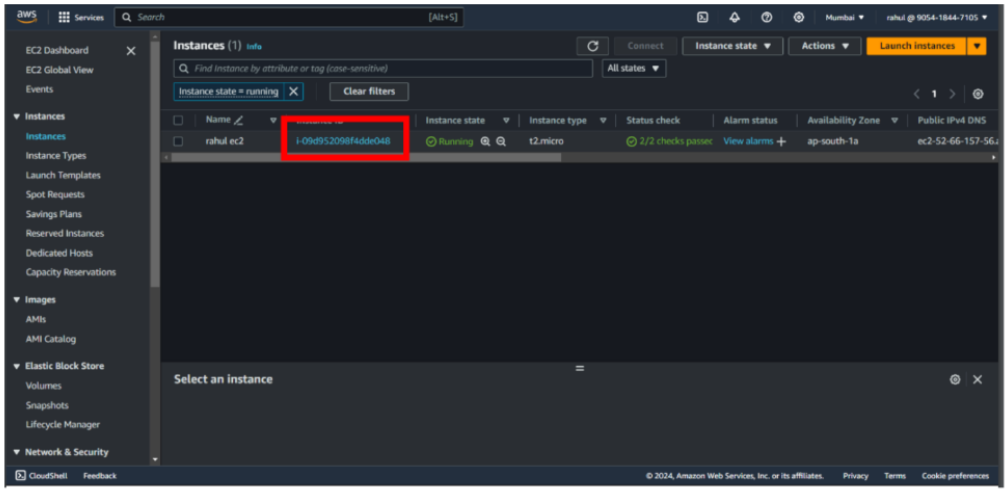
- Select your policy here which you have created previously , then click on **Next**.



- Then give a name of your choise to the role .

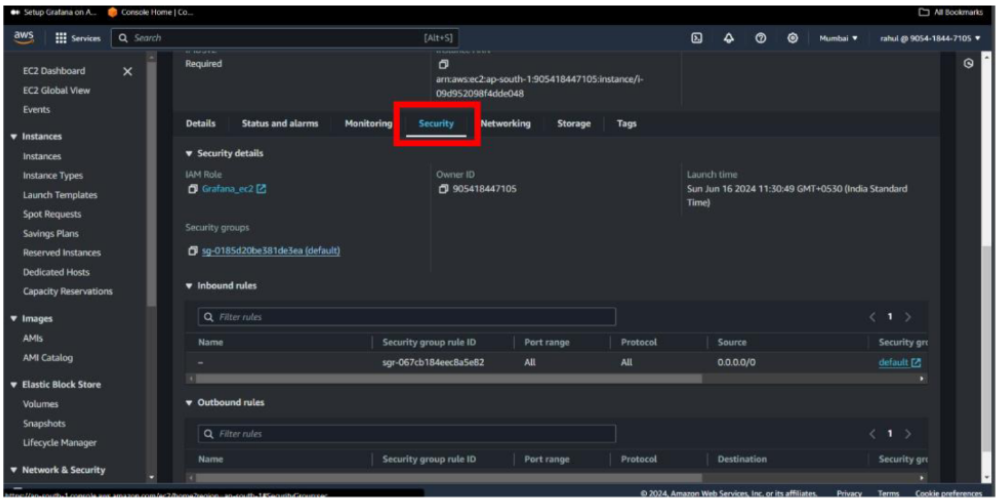


- Then scroll down and click on the **Create role** .

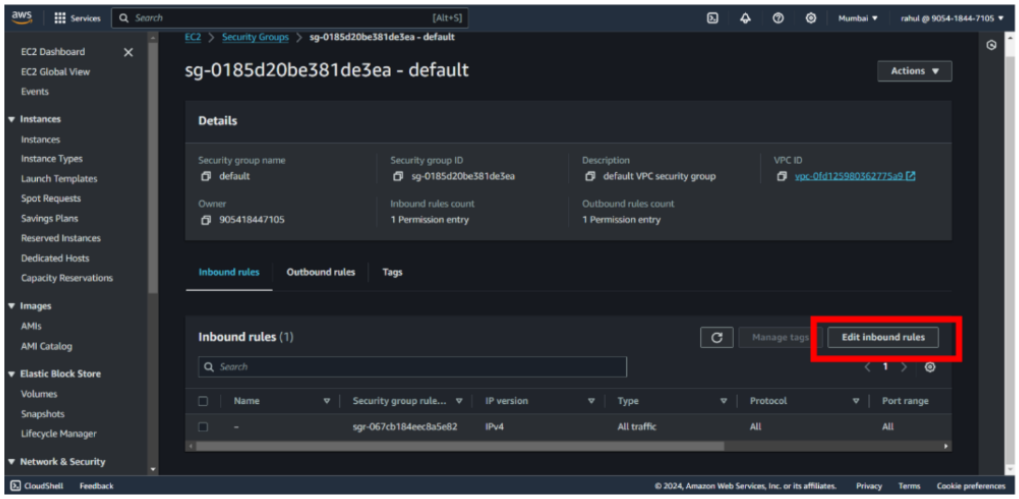


- After creating your role and policy go to instance and open your instance , which you have already created.

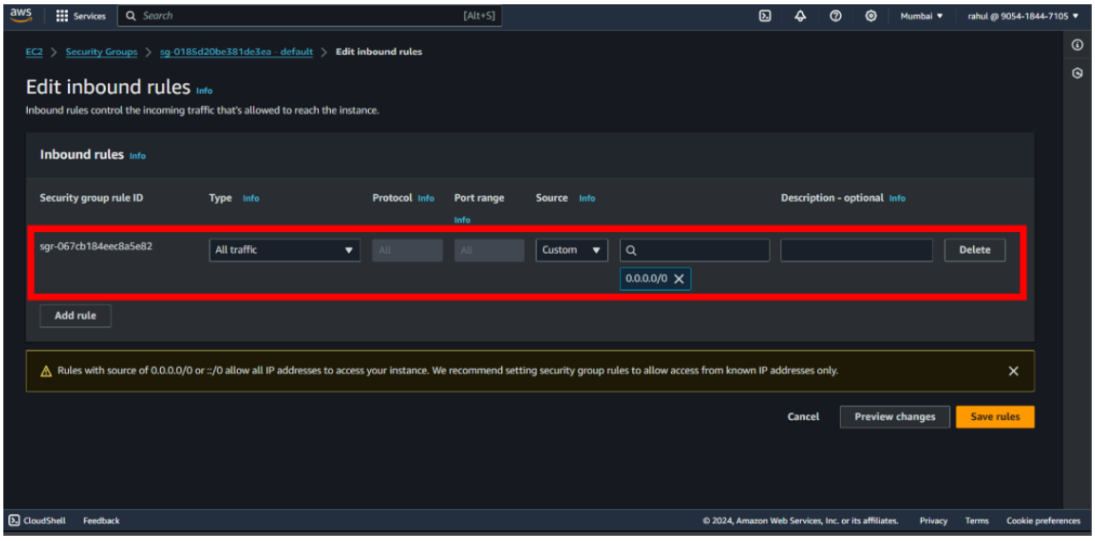
❖ Modifying security rules:



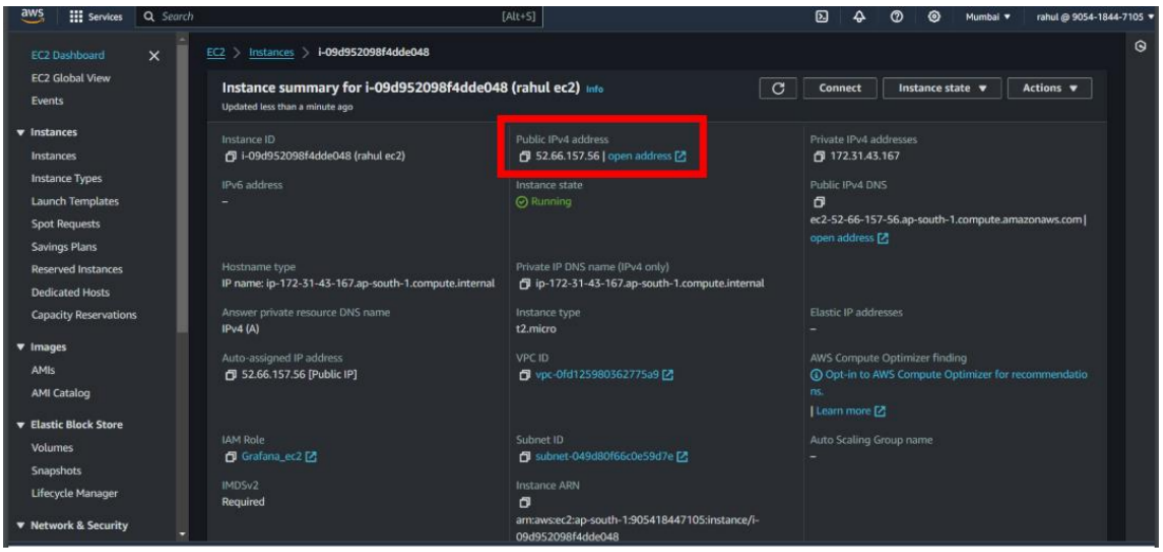
- Scroll down and go to **Security** option.



- Then click on **Edit inbound rule** .

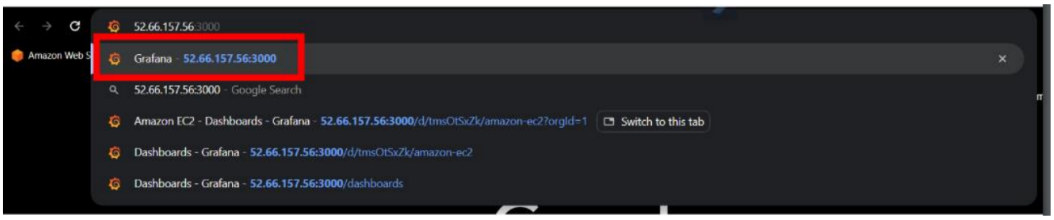


- Then modify your rule **select type = All traffic** and **source =0.0.0.0/0**

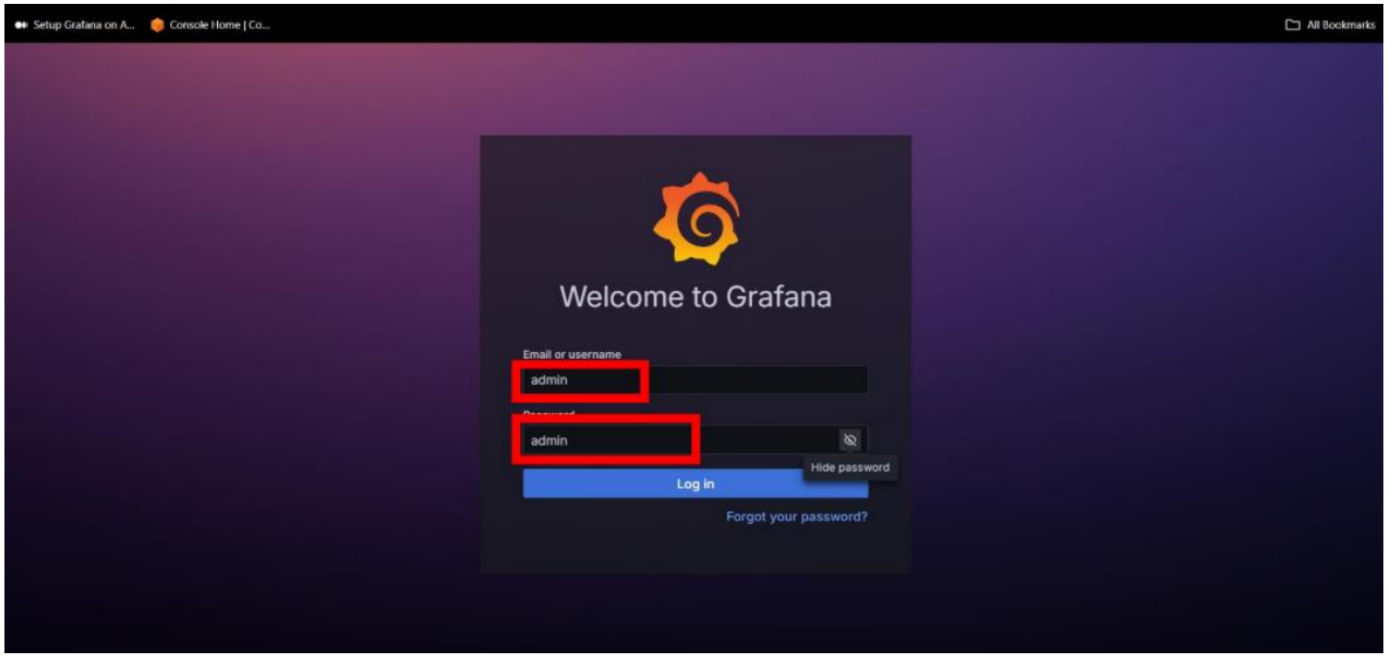


- Then copy your **public IPv4 address** of your instance .

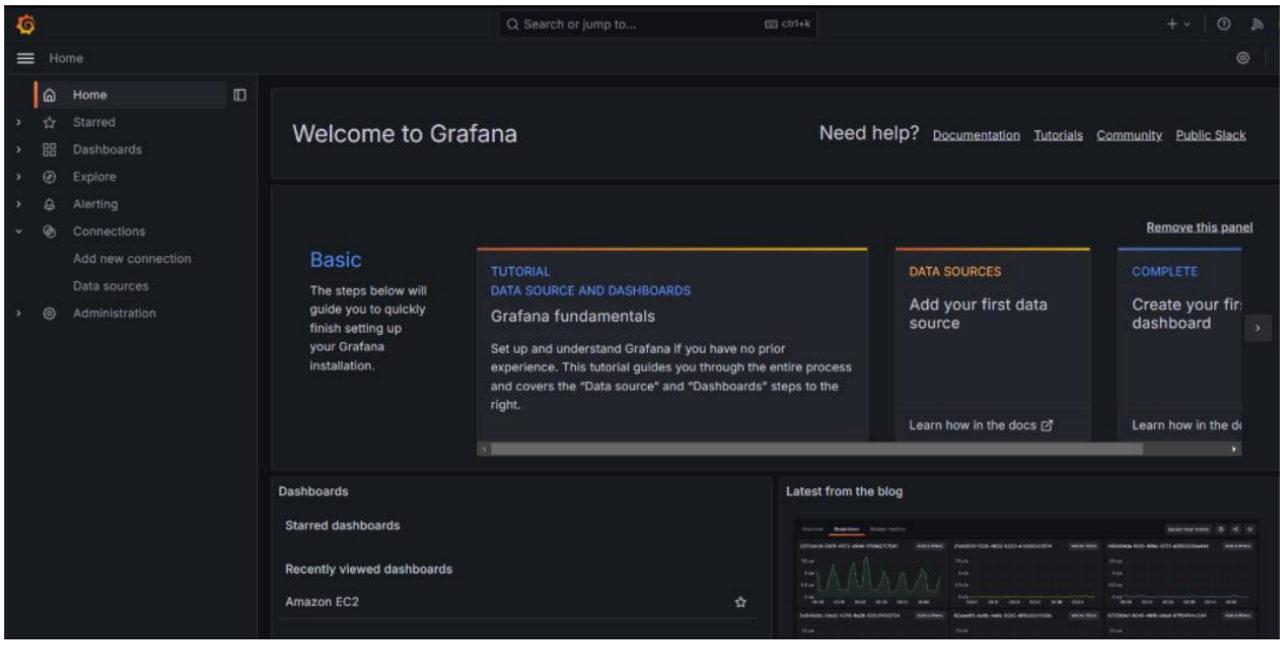
❖ <u>Starting GRAFANA:</u>



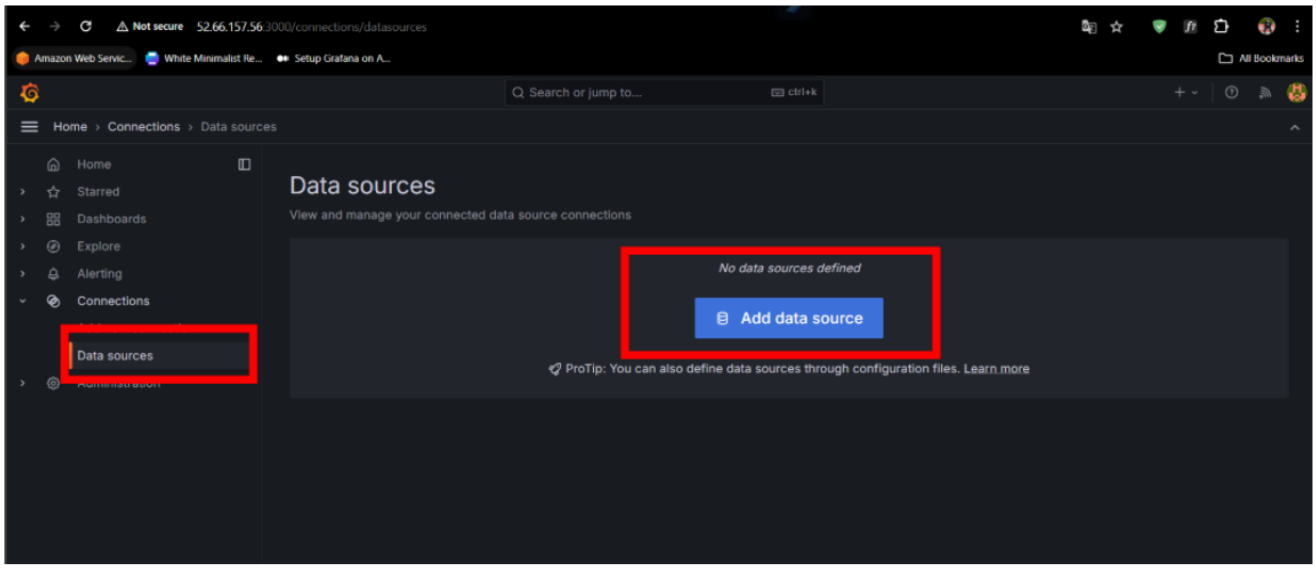
- Then paste your copied ip address and type "**:3000"** after ip address and search it .

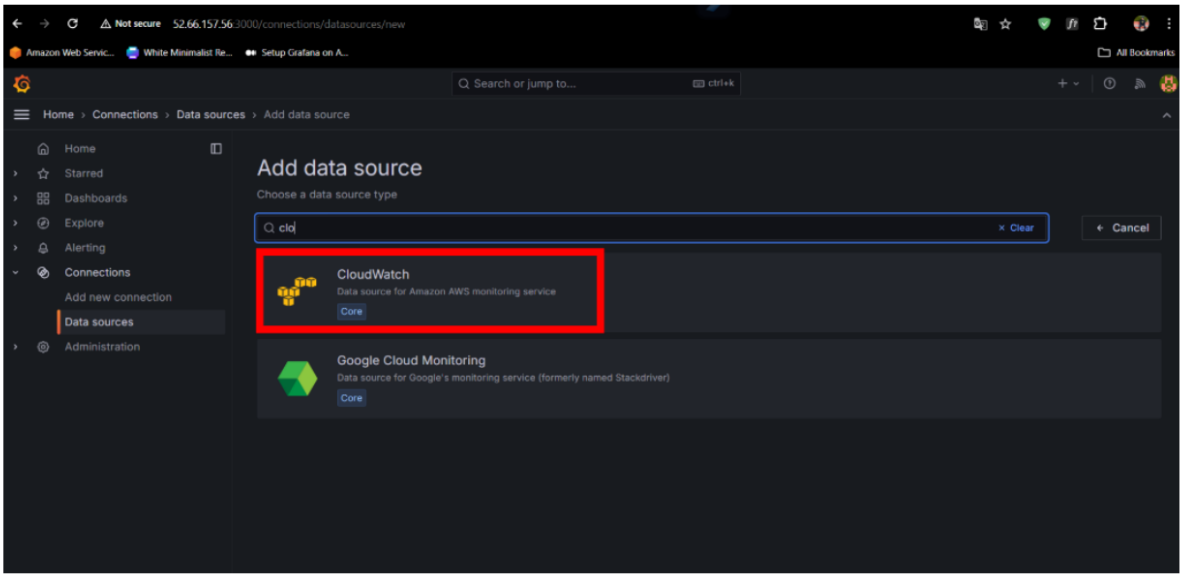


- Then login in Grafana by the help of credientials shown in above snapshot.

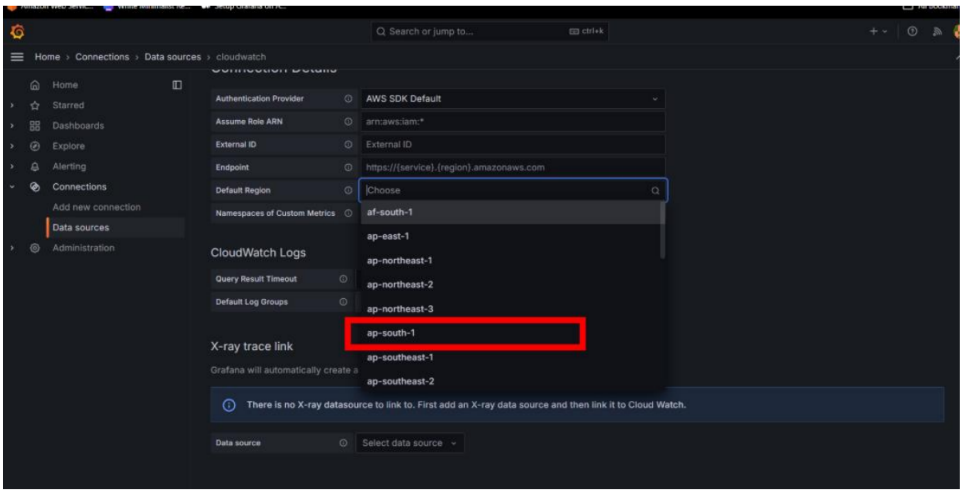


❖ Adding data source to grafana :

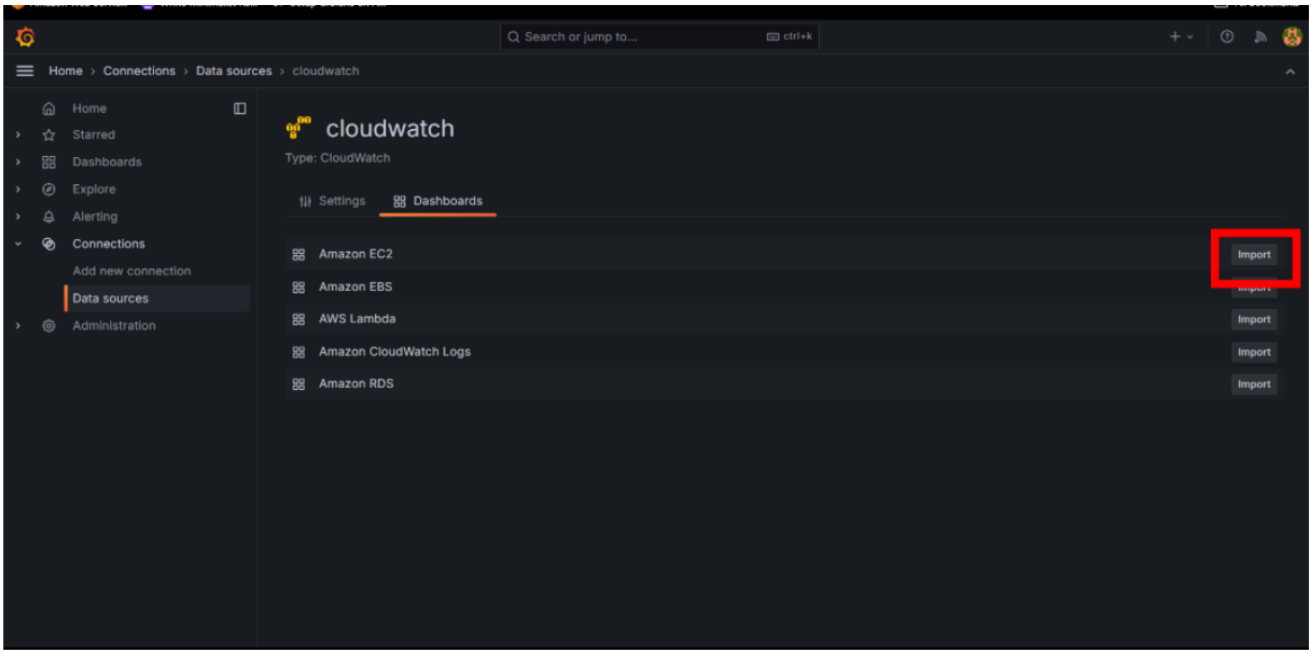




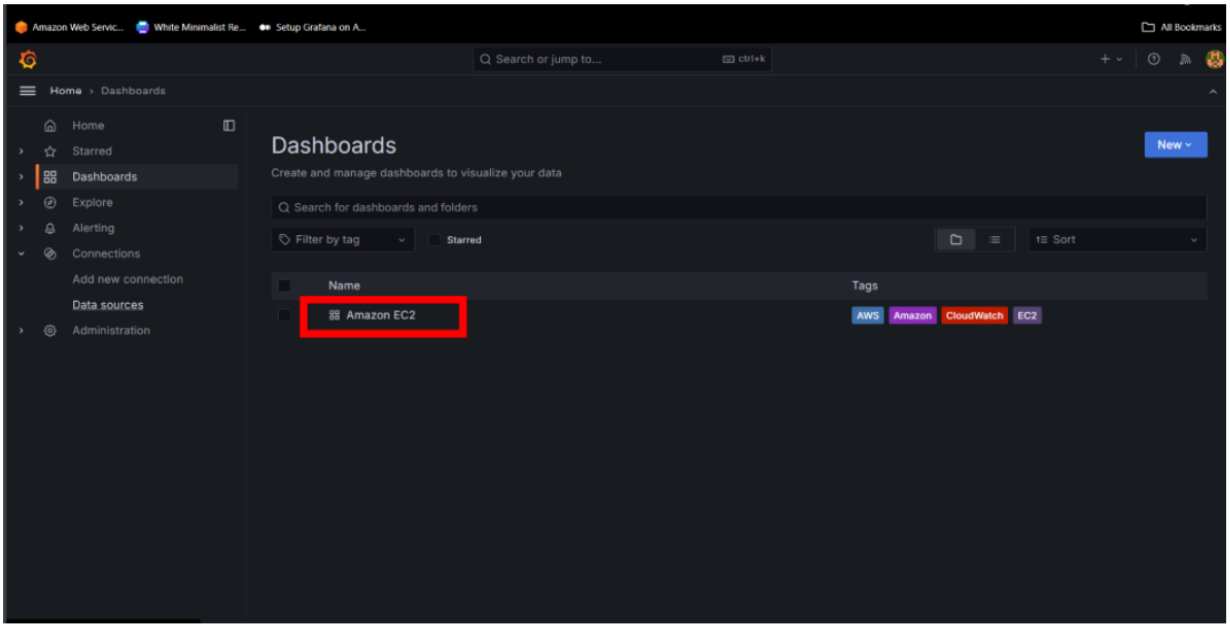
- Now , select “**Cloudwatch**” as an data source .



- Then after selecting cloudwatch select your region .



- Then inside your cloudwatch go to dashboard and **Import EC2**.



- Then go to dashboard and open your **Amazon EC2**.

Capgemini