Grafana-Setup:

Step1:

Add the following things in the java code under this directory structure(\src\main\resources\application.properties) for popup the metrics in actuator

management.endpoints.web.exposure.include=\*

management.endpoint.health.show-details=always

C:\Users\Kotic\Downloads\spring-boot-micrometer-master\src\main\resources\application.properties

Step2:

Add the following dependencies in pom.xml under dependencies

<dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-actuator</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>io.micrometer</groupId>

            <artifactId>micrometer-registry-prometheus</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

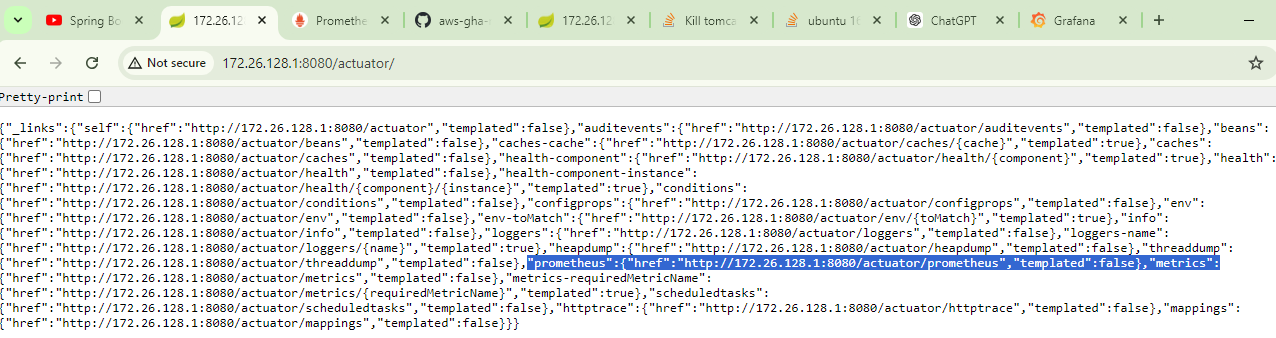
        </dependency>

    </dependencies>

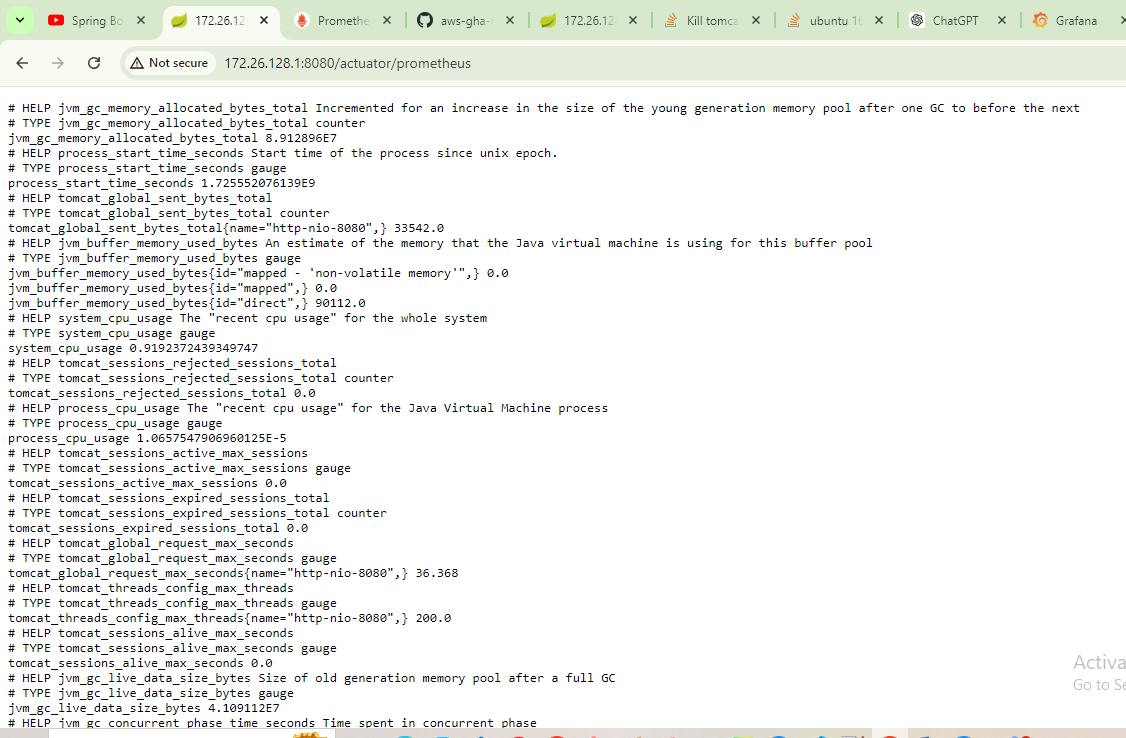
Step3:Build & Run the applications then we can see the no of metrics populated by the actuator like this.

For Build: mvn -B package --file pom.xml

For Run: mvn spring-boot:run



And promethus can contains following metrics



Step4:Adding Prometheus(data source which captures the actuator metrics and stores in a database)

I.First step need to add the prometheu.yml file in the directory structure C:\Users\Kotic\Downloads\spring-boot-micrometer-master\src\main\resources\prometheus.yml

and Prometheus.yml file should contain the following information

# my global config

global:

  scrape\_interval:     15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.

  evaluation\_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.

  # scrape\_timeout is set to the global default (10s).

# Load rules once and periodically evaluate them according to the global 'evaluation\_interval'.

rule\_files:

  # - "first\_rules.yml"

  # - "second\_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:

# Here it's Prometheus itself.

scrape\_configs:

  # The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.

  - job\_name: 'prometheus'

    # metrics\_path defaults to '/metrics'

    # scheme defaults to 'http'.

    static\_configs:

    - targets: ['127.0.0.1:9090']

  - job\_name: 'spring-actuator'

    metrics\_path: '/actuator/prometheus'

    scrape\_interval: 5s

    static\_configs:

    - targets: ['172.26.128.1:8080']

Here in the second job name spring-actuator contains the target which is nothing but ip of the local machine where you are running the java application with port no like 172.26.128.1:8080

II.pull the docker image for the Prometheus for activating the Prometheus.can pull it in anydrive in the machine.

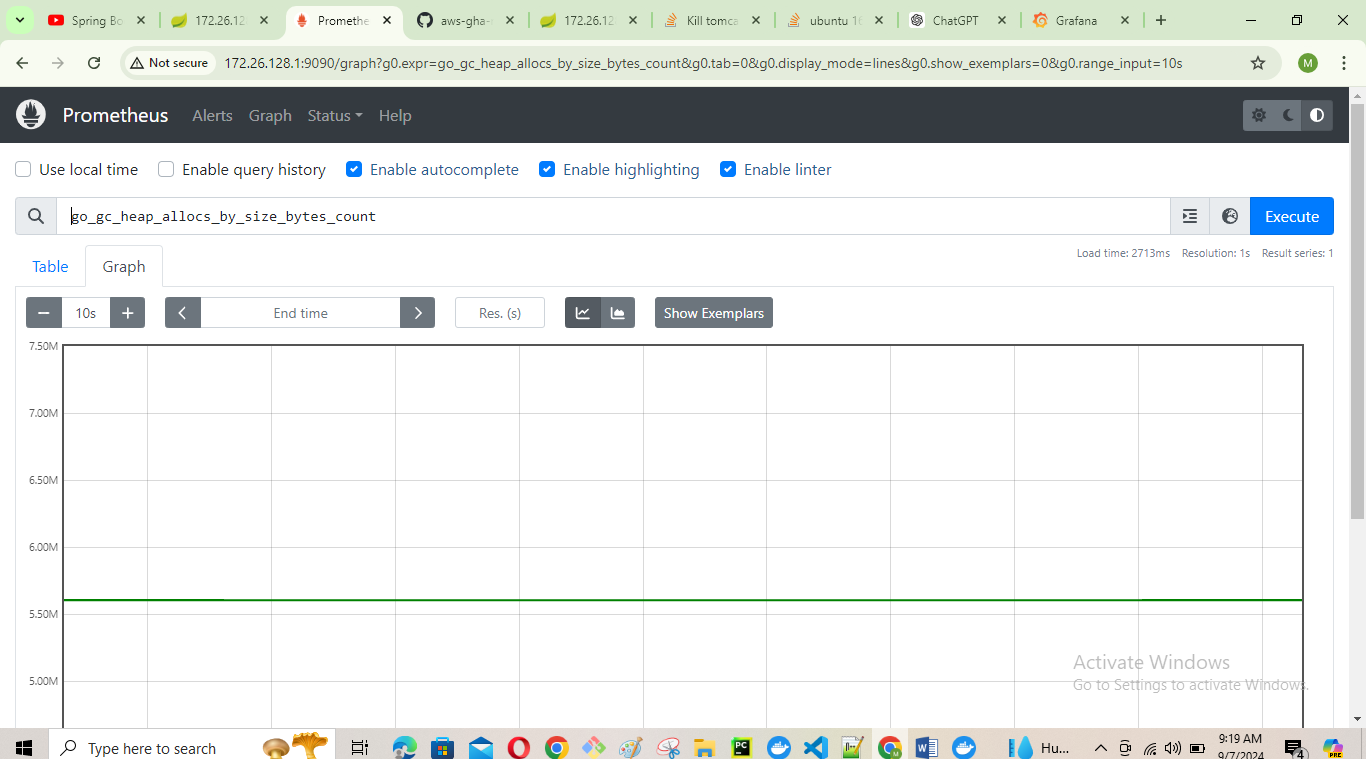
Like ex: docker pull prom/prometheus

III. Run the image to create the Prometheus container by the following command give the path where Prometheus.yml file exists.

Ex: docker run -p 9090:9090 -v C:\Users\Kotic\Downloads\spring-boot-micrometer-master\src\main\resources\prometheus.yml prom/Prometheus.

And Prometheus has port 9090 we will get the details like this

Select any metric and click execute we will get the graph/table



Step4: Add the Grafana to visualize the metrics from the Prometheus like beautiful dashboards.

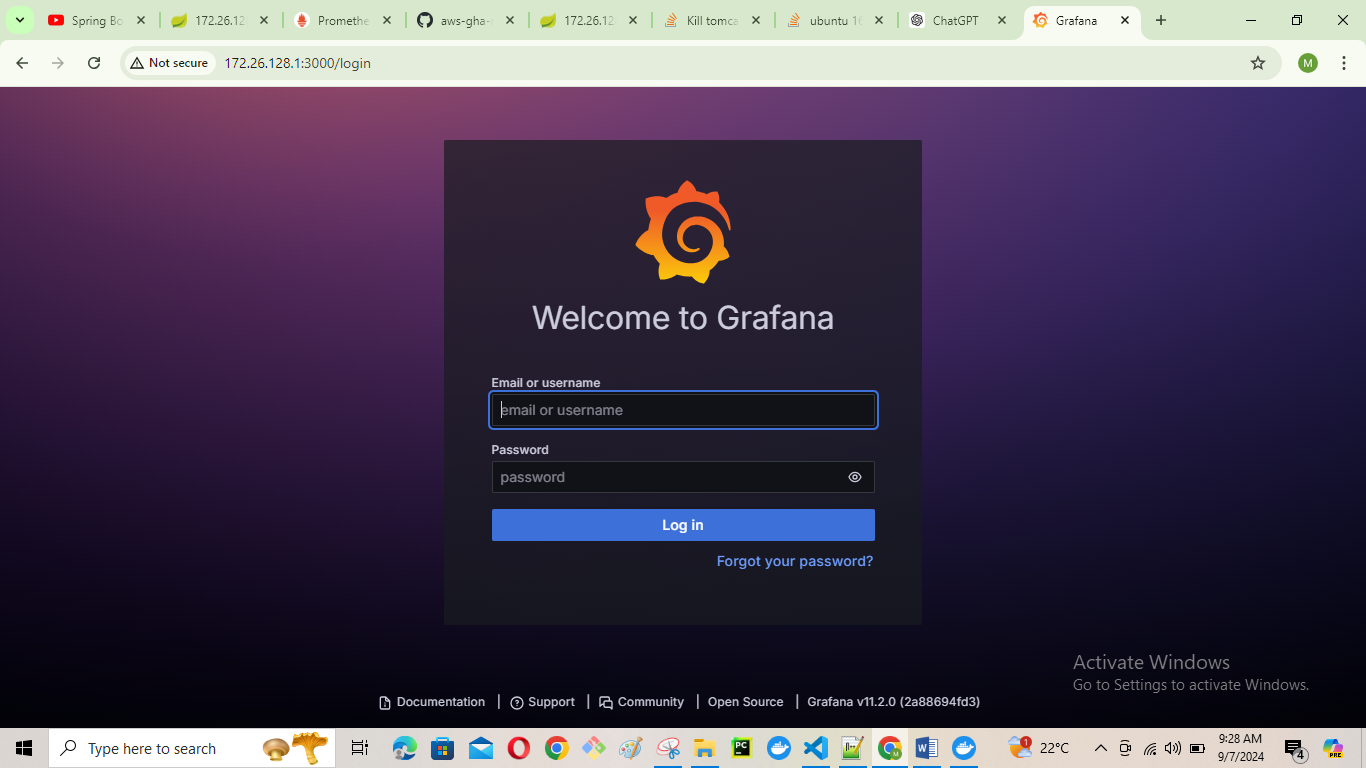
I.Pull the docker image of the grafana

Ex: docker pull grafana/grafana it will pull the latest grafana image

II.Run the docker image to get the grafana container

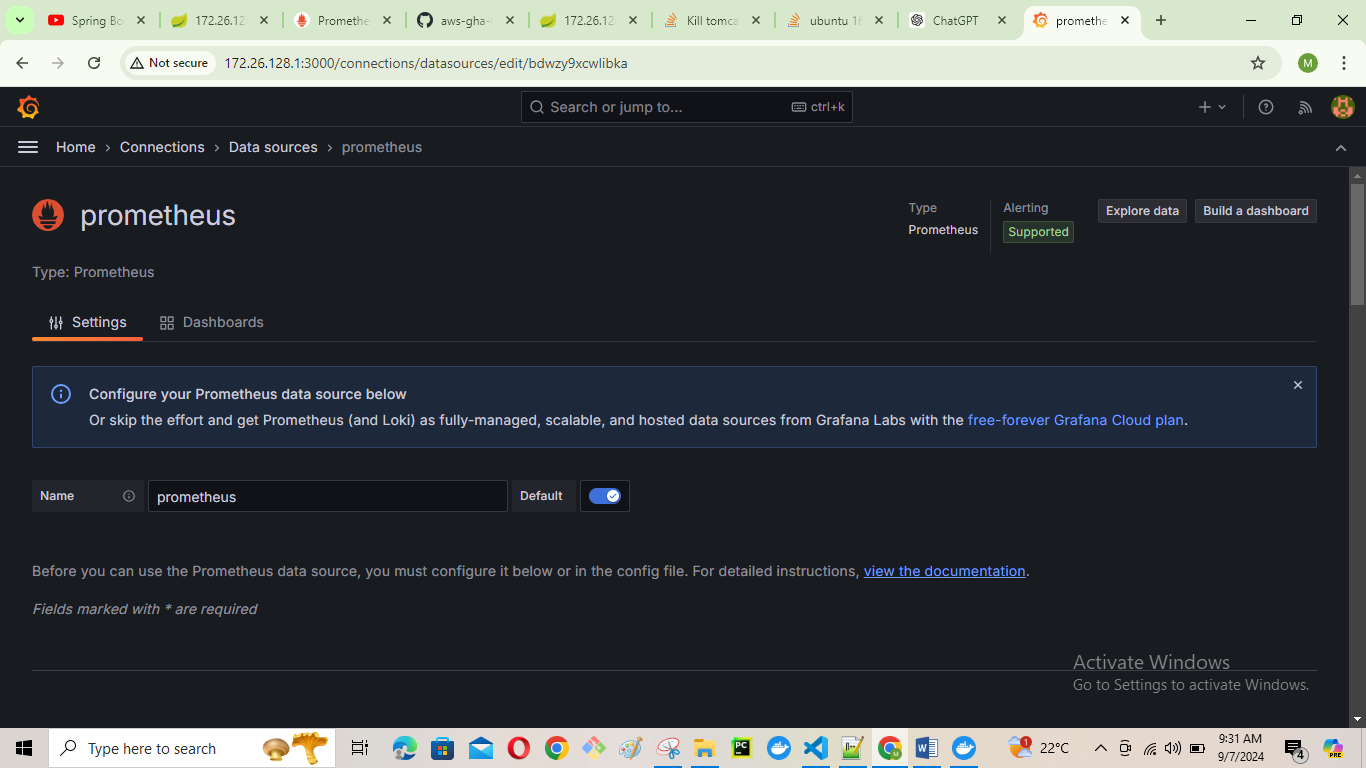
docker run -d --name=grafana12 -p 3000:3000 grafana/grafana

and grafana can be accessible from the 3000 port with system ip.(sometimes 172.26.128.1:3000 is not showing the grafana page then give full url in the browser like this <http://172.26.128.1:3000> then we will see the grafana page)

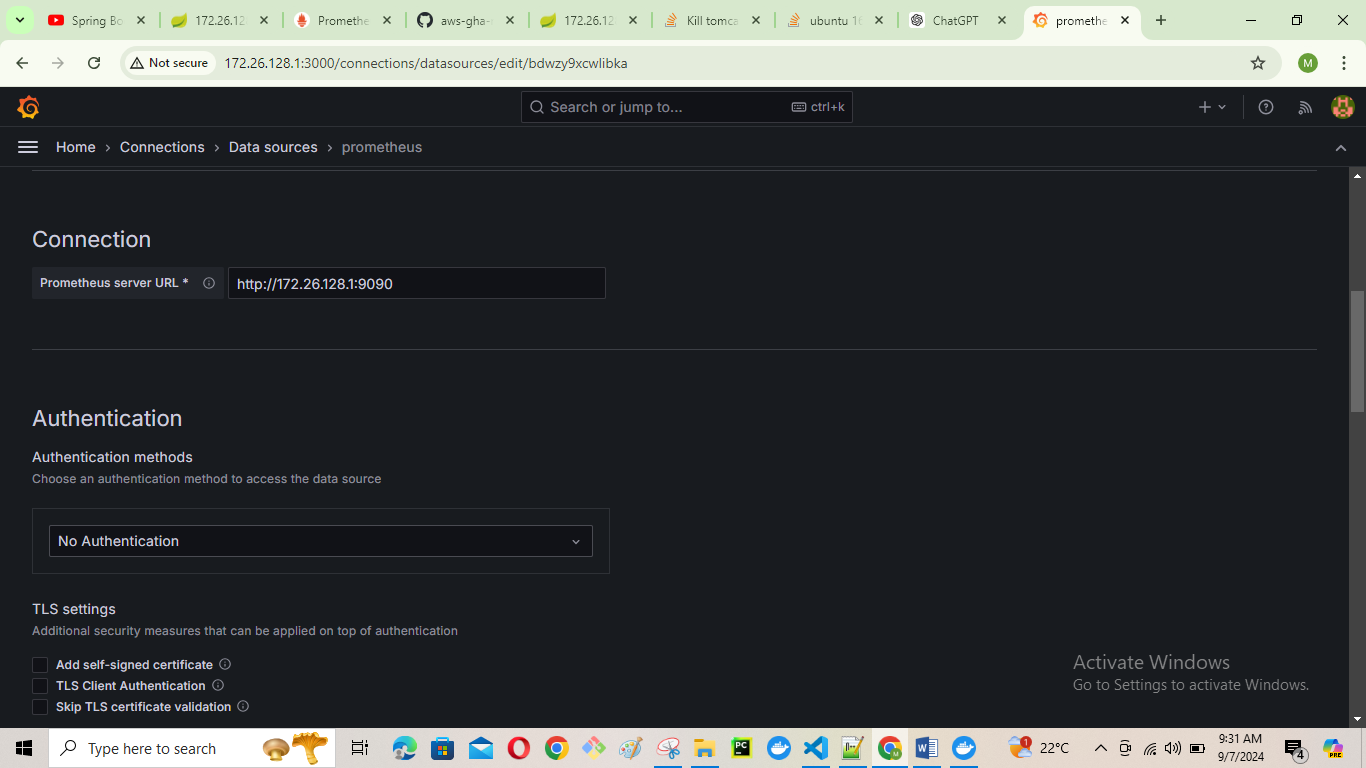


And default credentials for this page is username:admin pwd:admin. After login it will ask to change the credentials if you wish to change it you can.

II. Add the Prometheus datasource to grafana

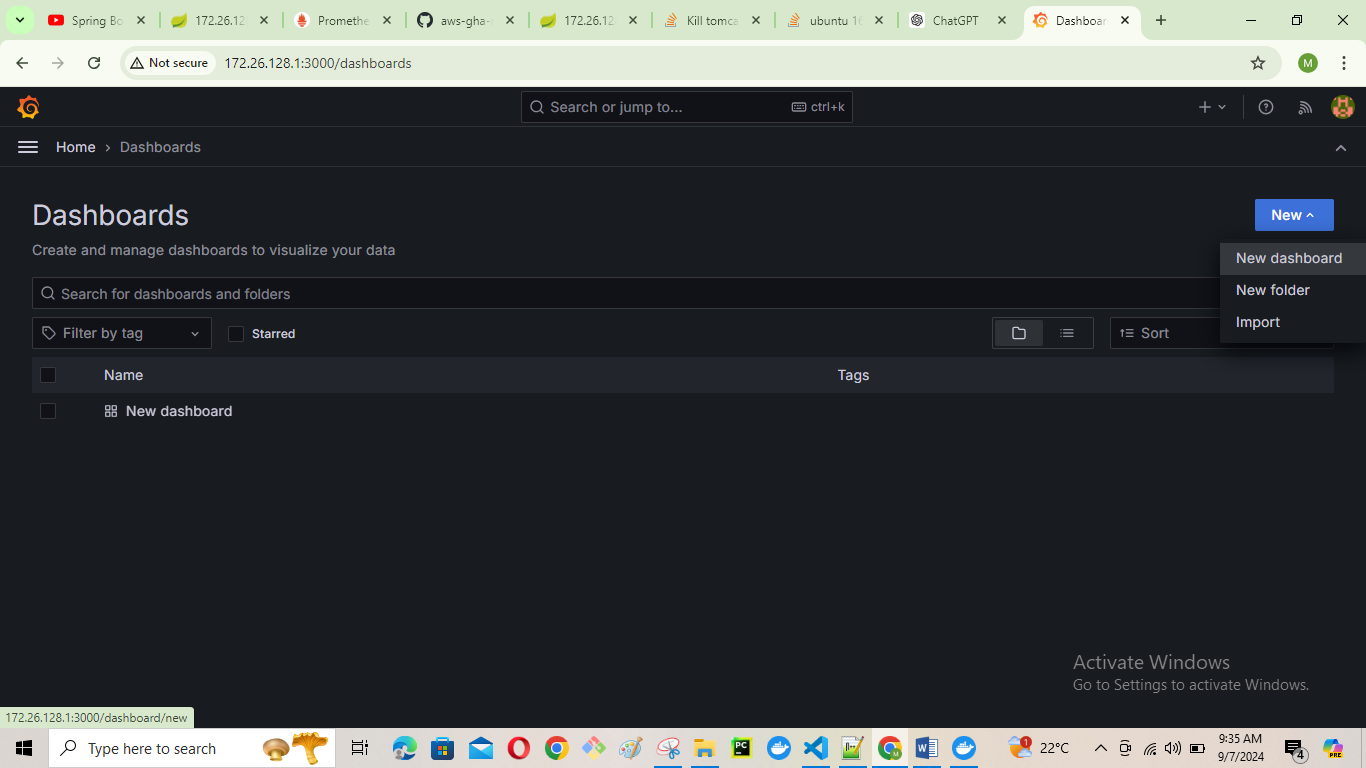


And give the connection details like ip address followed by the port no for Prometheus to the grafana and save it

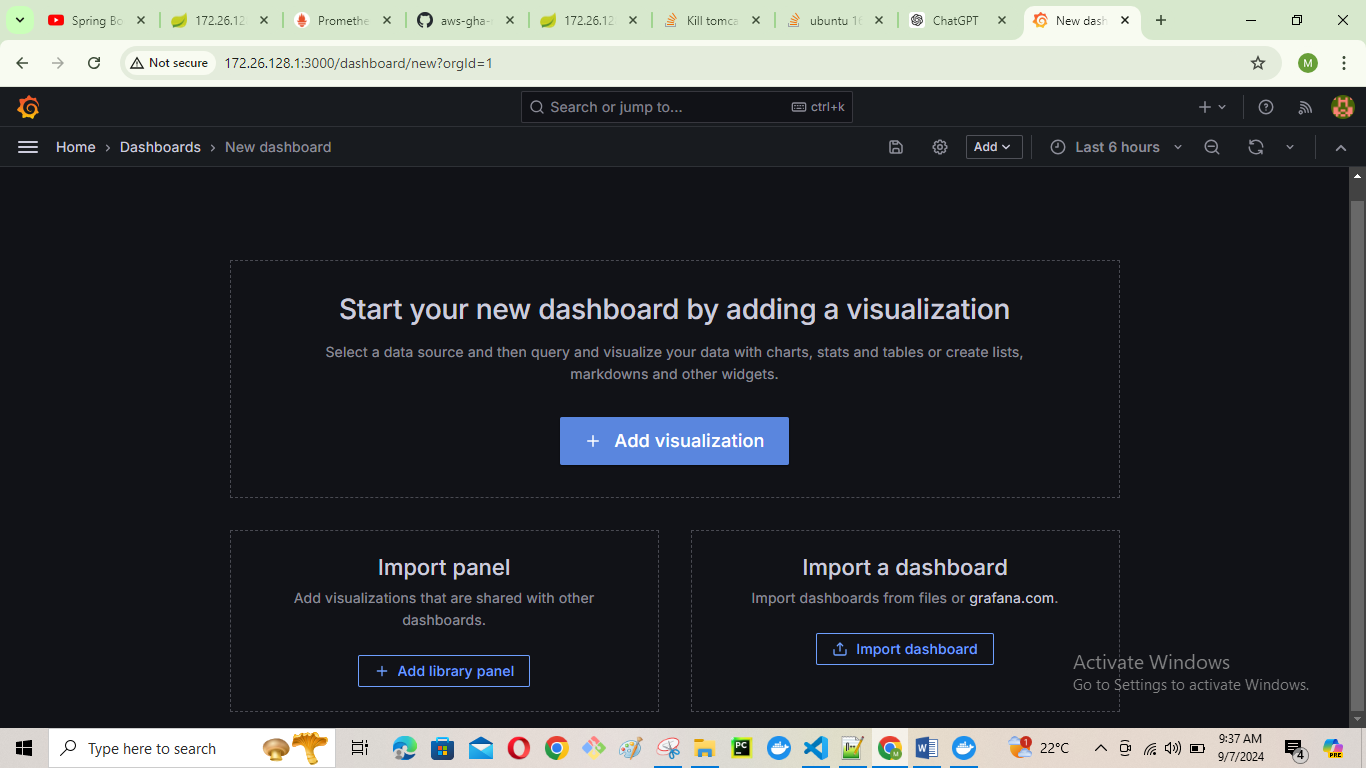


III. Create the dashboard

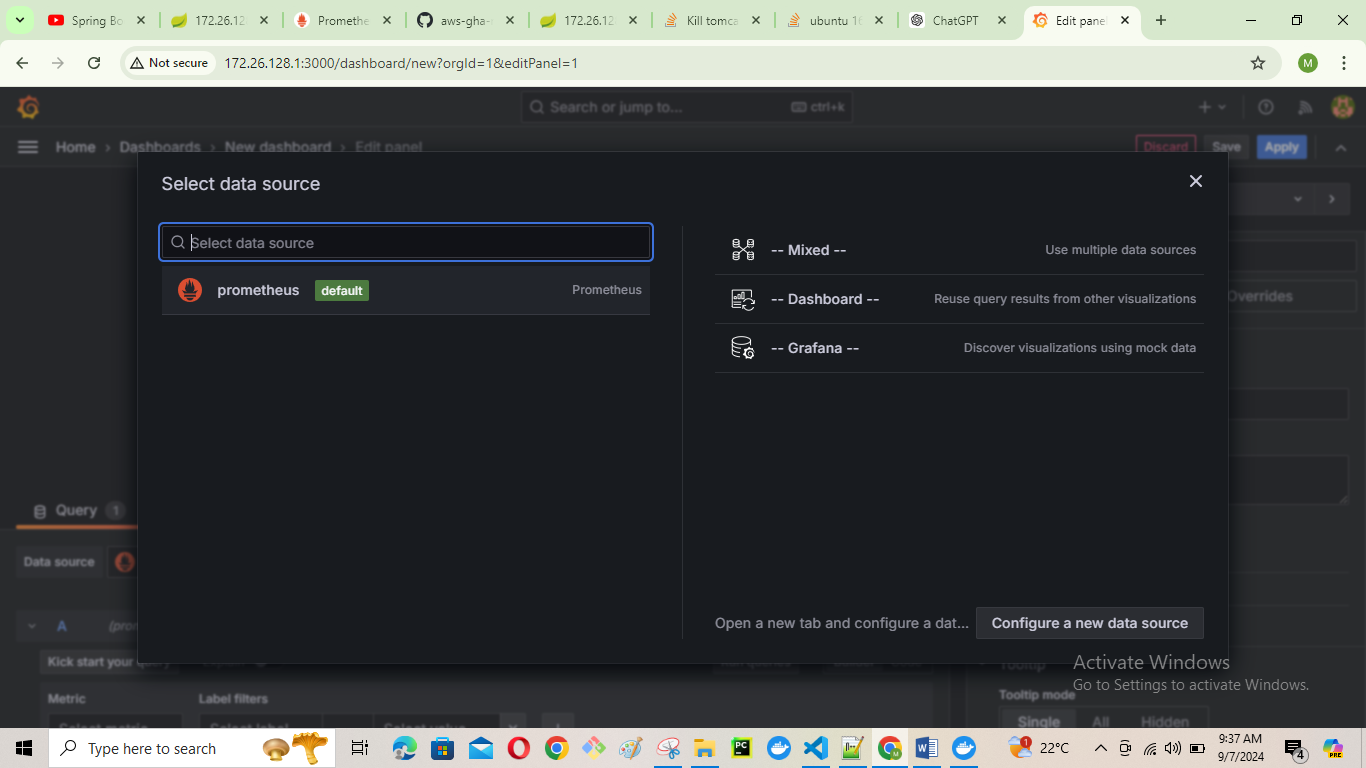
Click the dashboard and click new New dashboard.



Click add visualizations

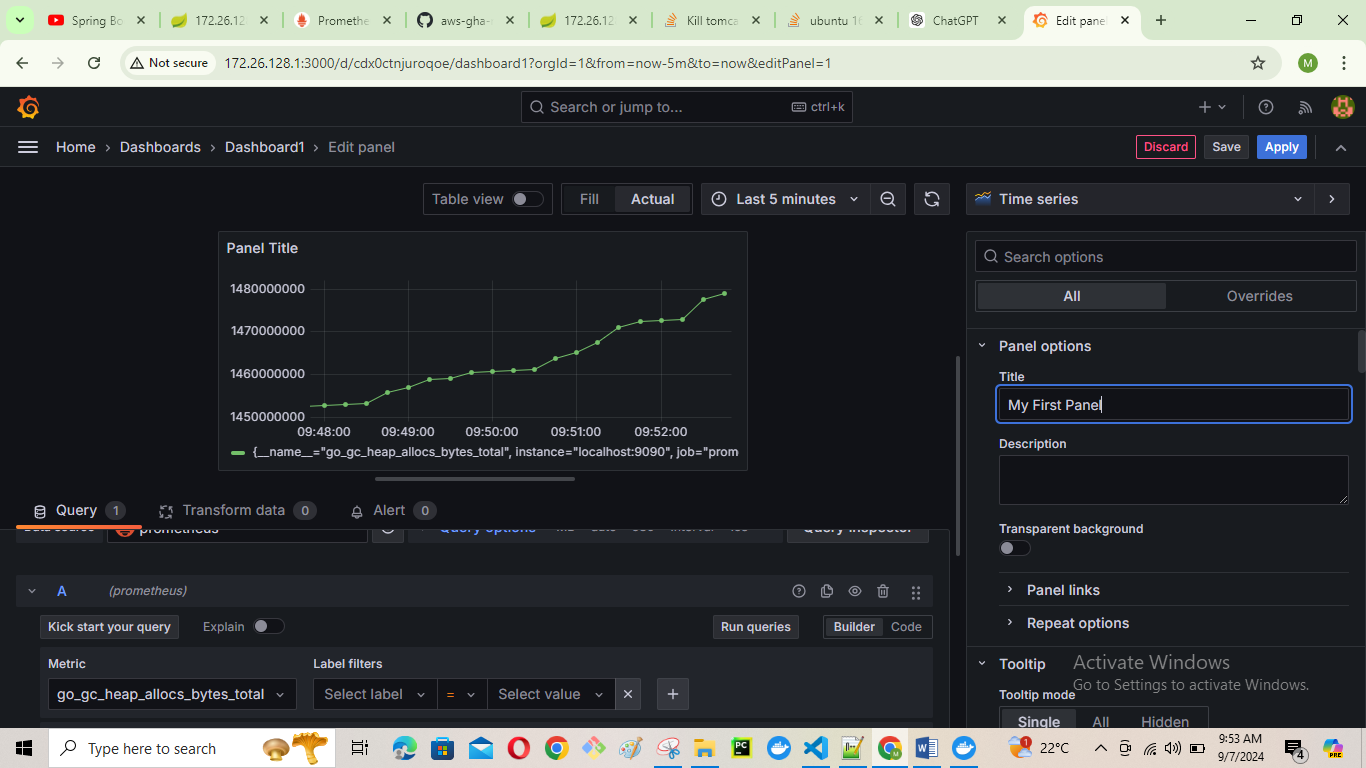


Add data source prometheus



Create a dashboard

And name the panel



Click on metric box it will populate no of metrics and select any one that can be visualize in the panel( in my case "go\_gc\_heap\_frees\_bytes\_total",

If you want to add the Query add it other wise save the Dashboard.

