**Configuring, Provisioning, and Monitoring Node.js Application = Capstone Project no 2**

**Process:**

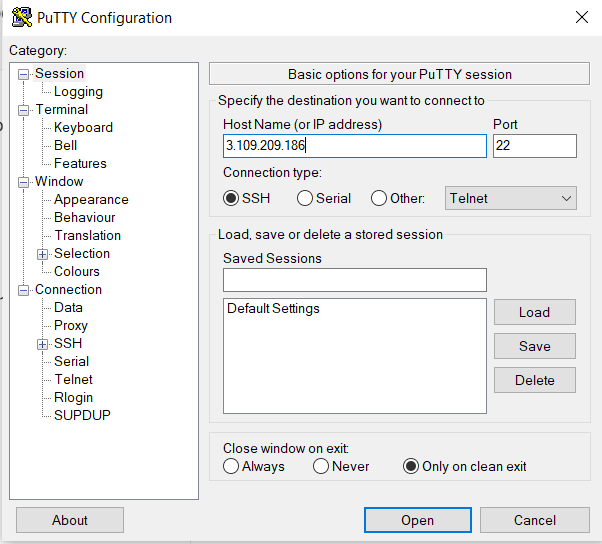
**Set Up AWS Resources**

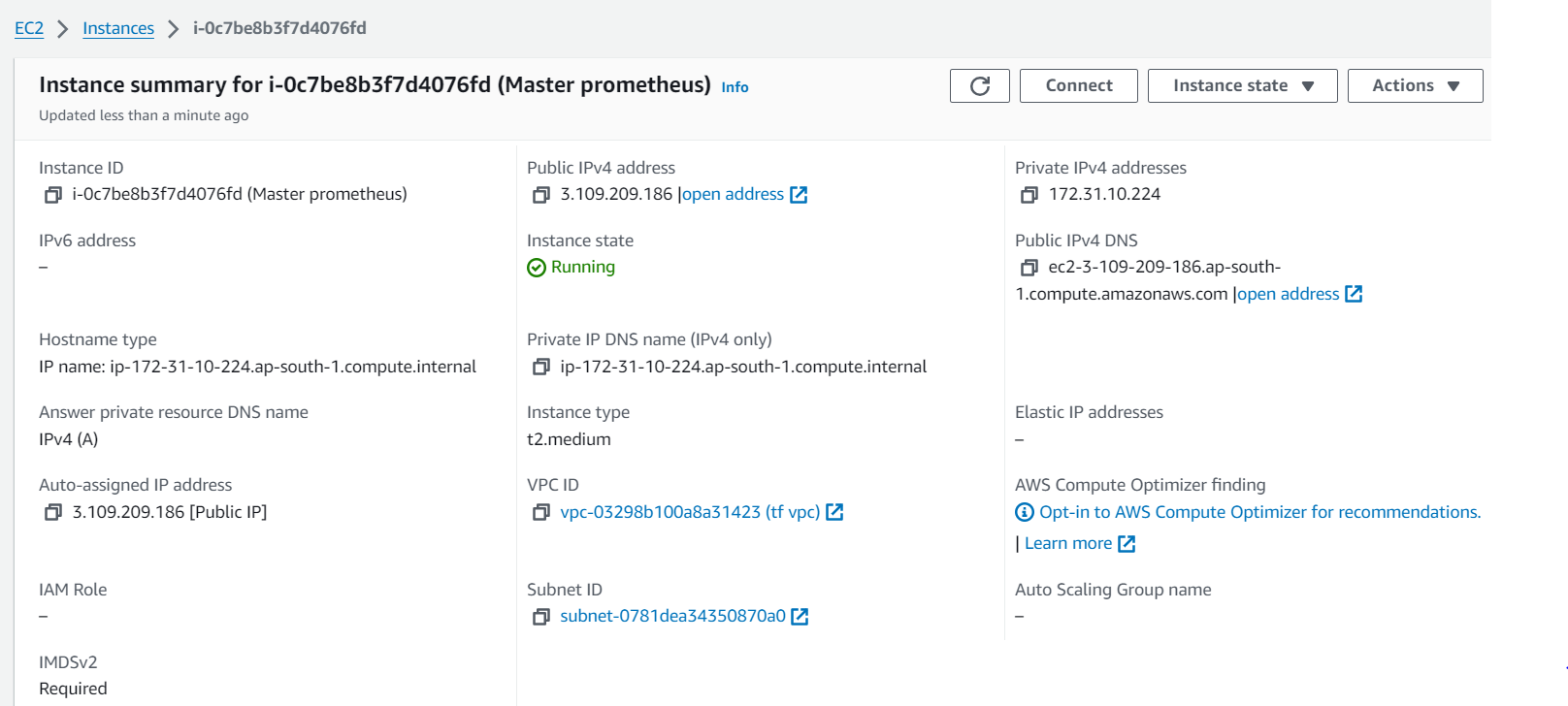
Logged in to AWS Management Console.  
• Launched EC2 Instance: (virtual machine - Ubuntu).

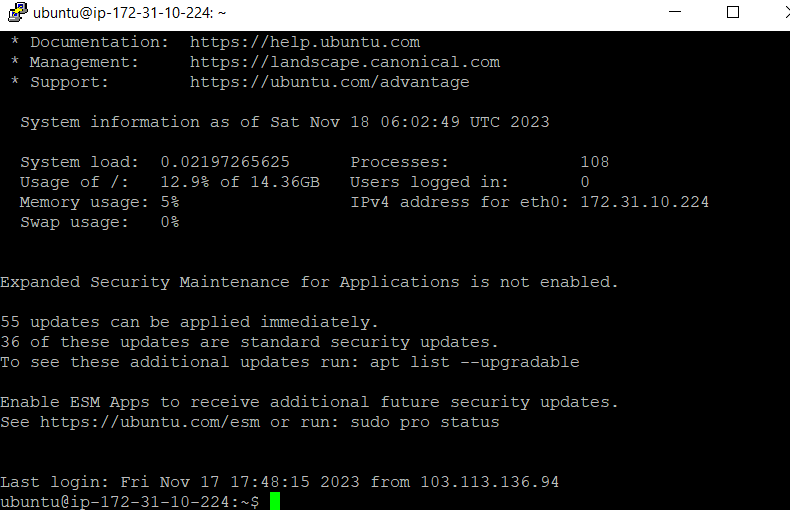
And security groups to allowed SSH (port 22) traffic with instance type - T2.medium and 15Gb disk size and created PPK file.

**Connect to Your EC2 Instance**

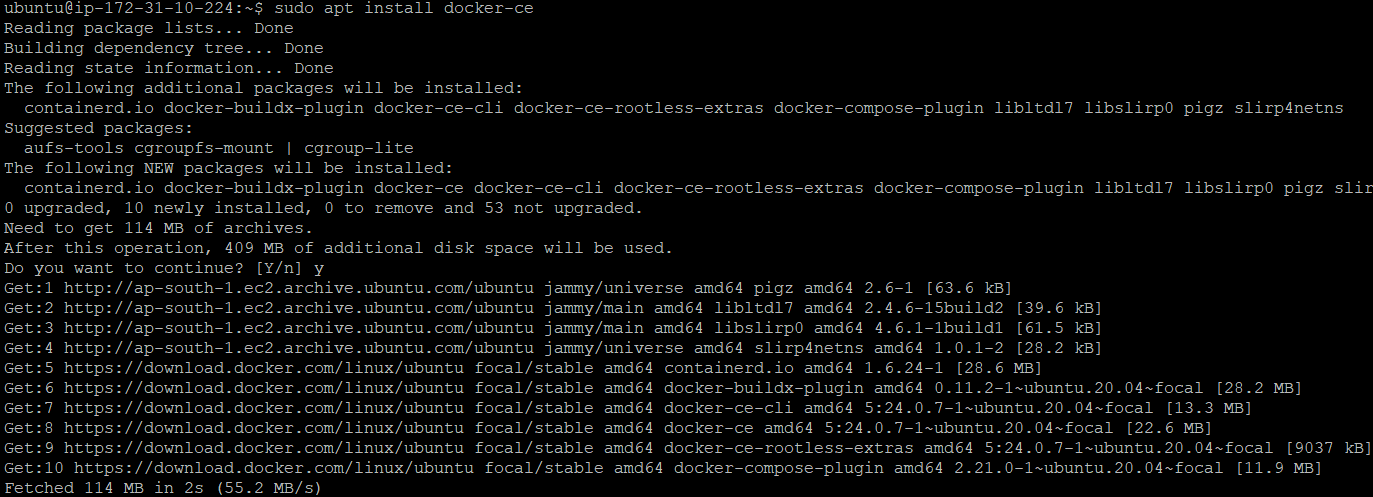
•    Used SSH to connect to EC2 instance using the PPK file via Putty as below.



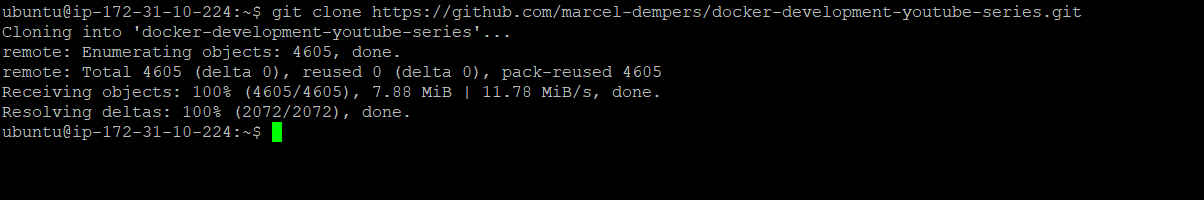




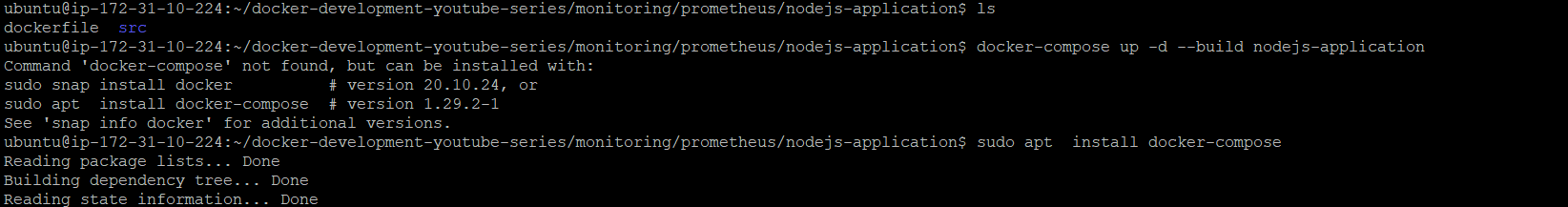
**Installed docker on to the machine as below:**



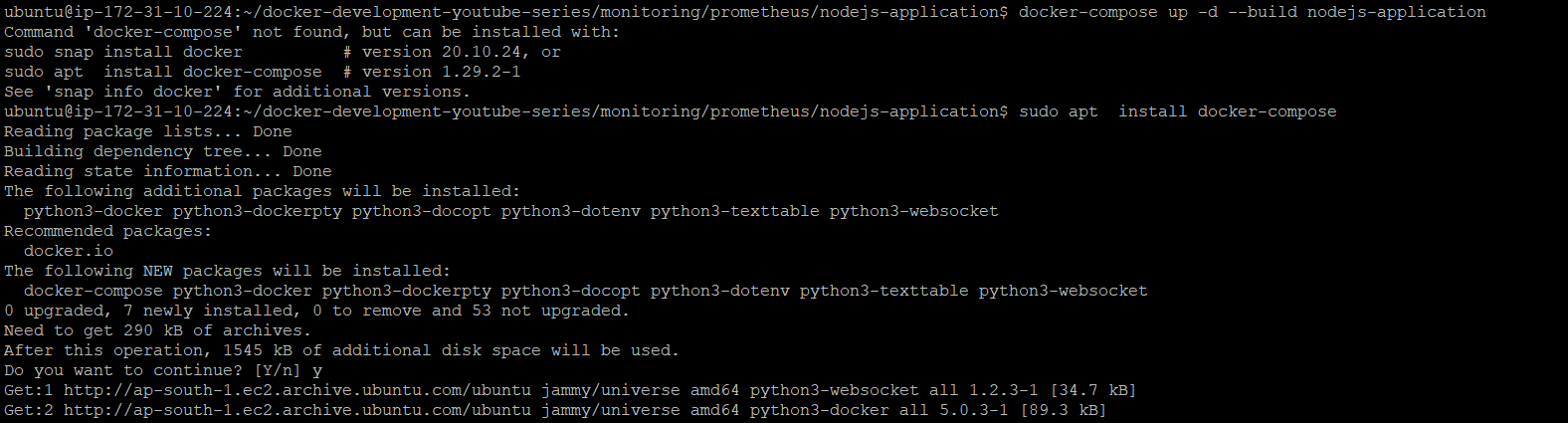
**Cloned the Git repo and code on to the machine as below:**

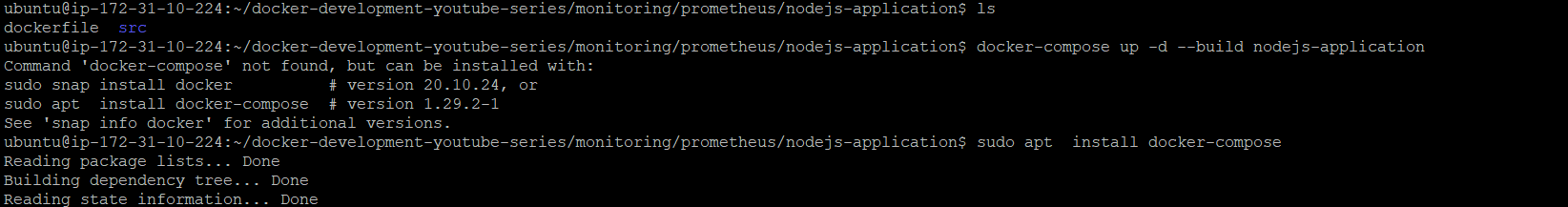


**Change directory to the source code repo for building the docker file :**



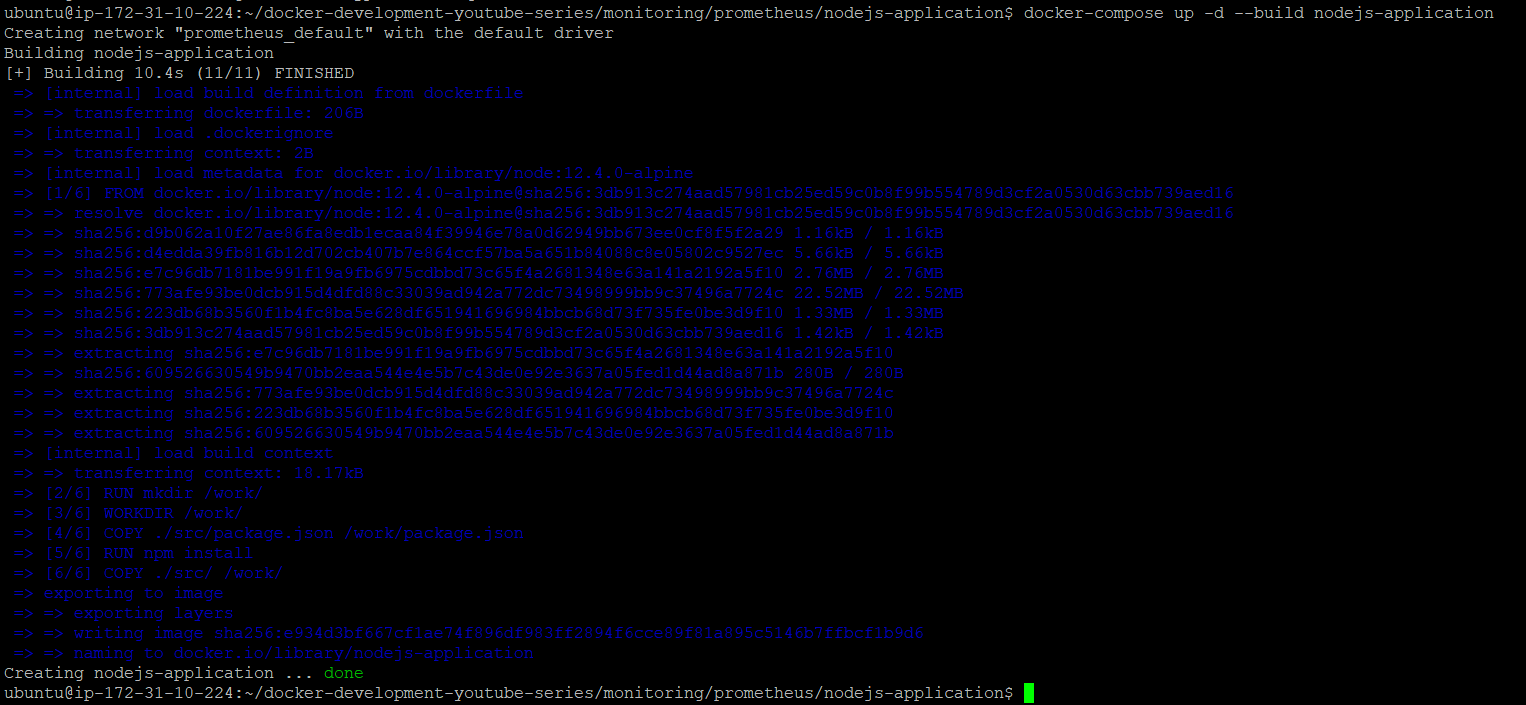
**Here I used docker compose concept as we are deploying multiple docker containers(microservice) so installed docker-compose as below:**





**Below is the command for building the nodejs-application that is building**

**Docker file.**



**Docker File :**

FROM node:12.4.0-alpine as dev

RUN mkdir /work/

WORKDIR /work/

COPY ./src/package.json /work/package.json

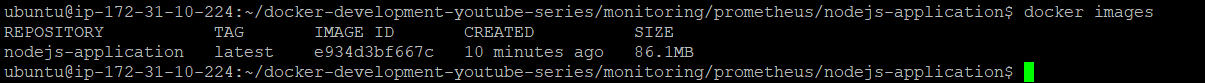
RUN npm install

COPY ./src/ /work/

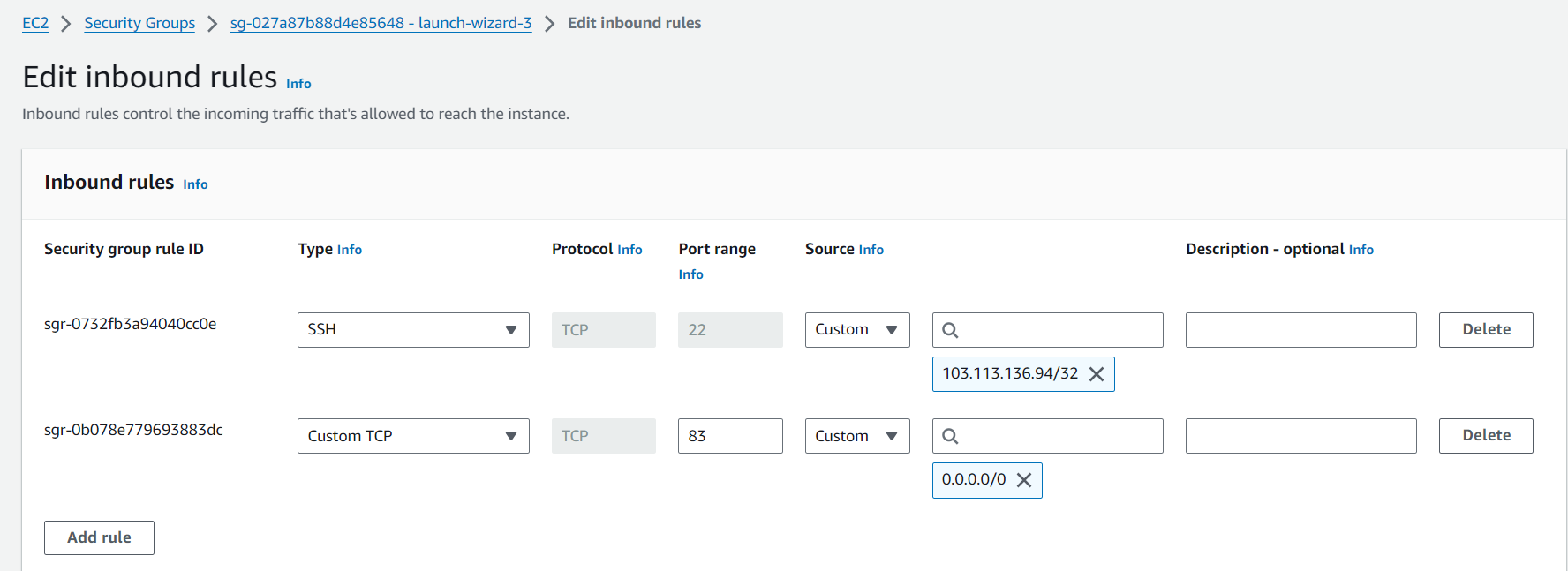
CMD node .

**Now checked running images in the machine using below command**

* docker images

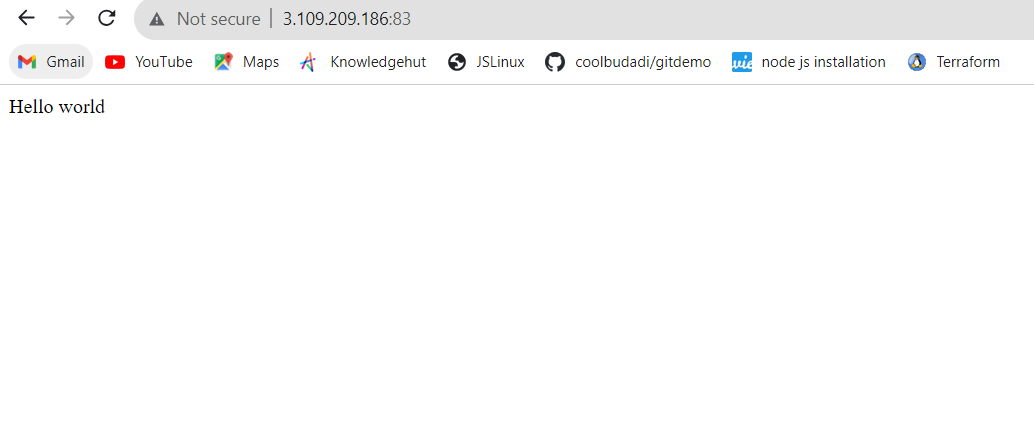


**As app is running on port 83 so added port entry to the SG as below:**

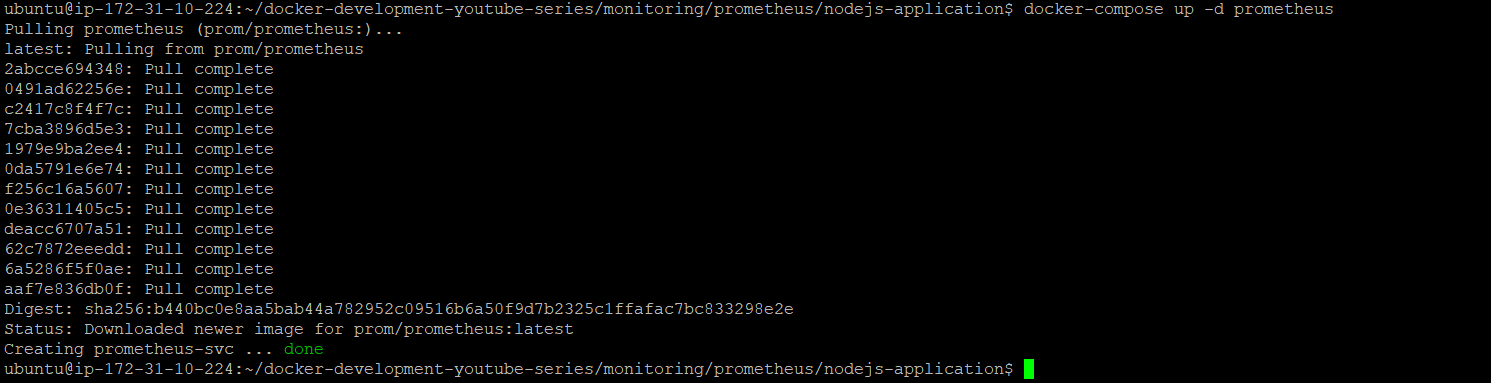


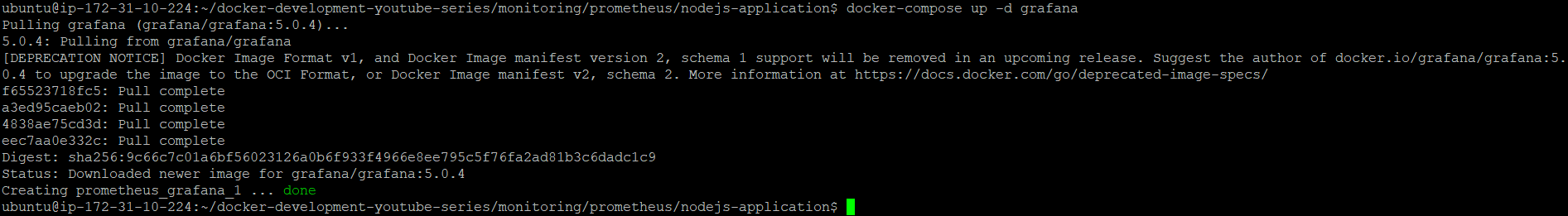
**And visit to website by hit IP and port no as below :**

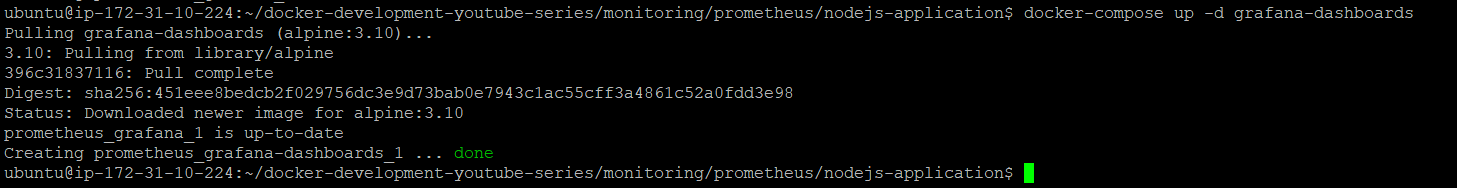
Running App:



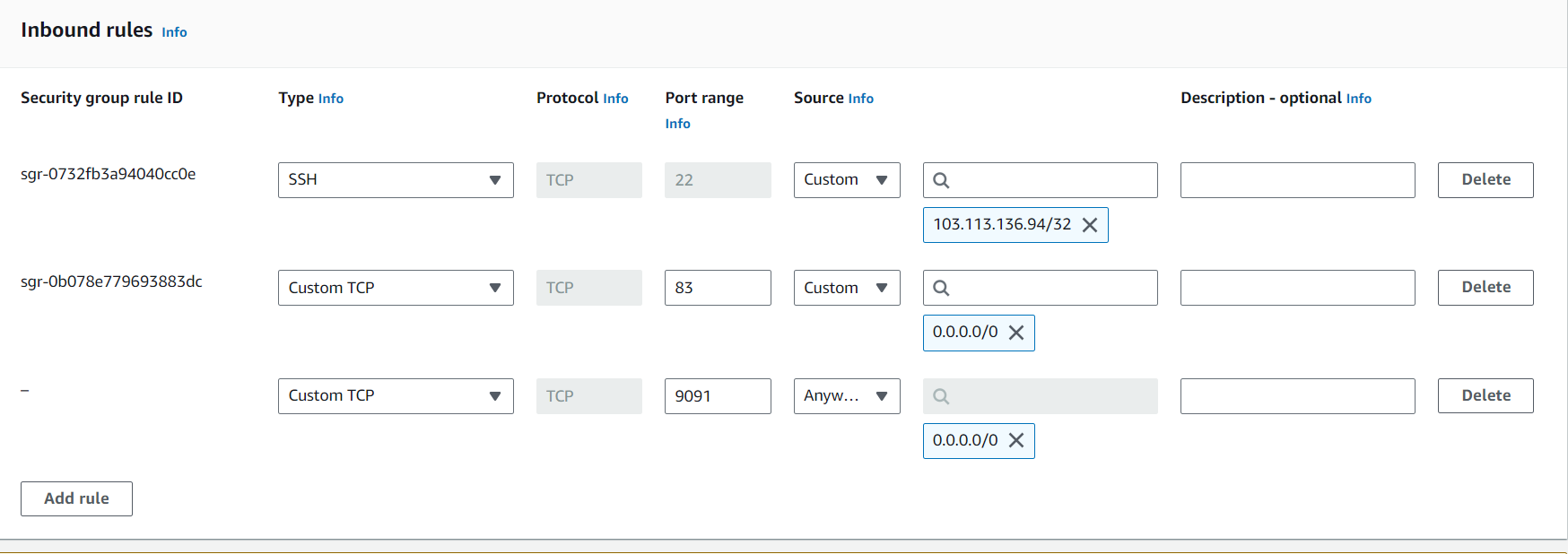
**Downloaded and created Prometheus ,Grafana , Grafana Dashboard services by using docker compose as below:**



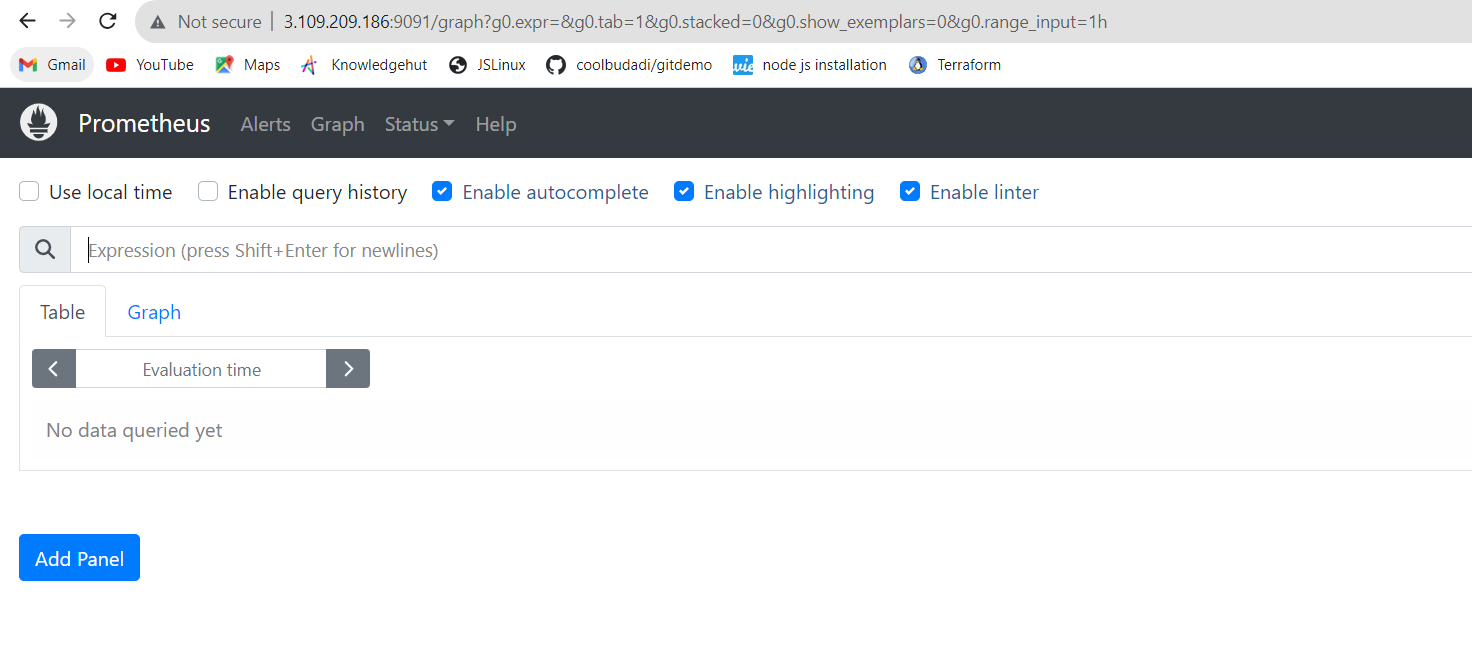




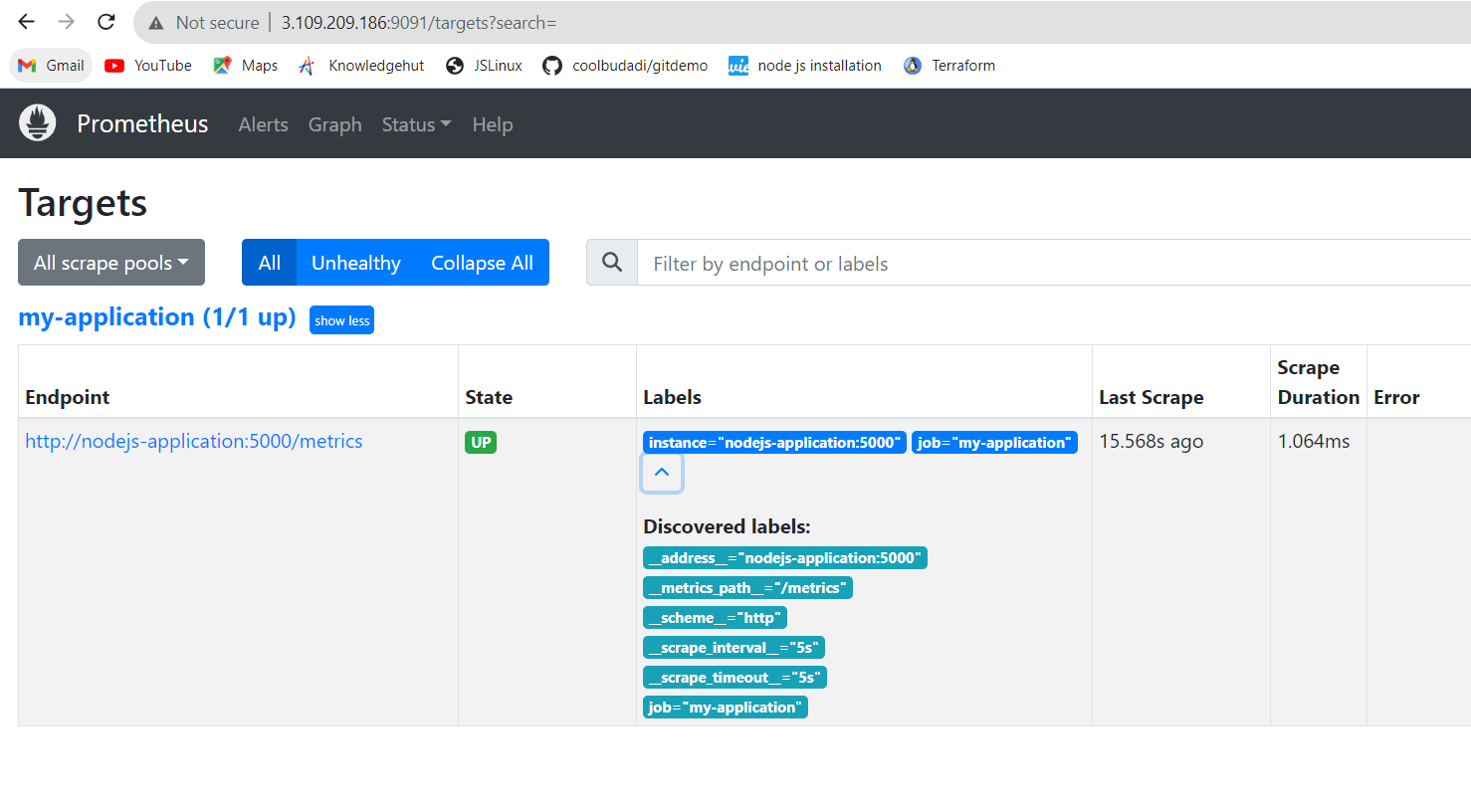
**As Prometheus running on Port 9091 so added as entry to SG as below:**



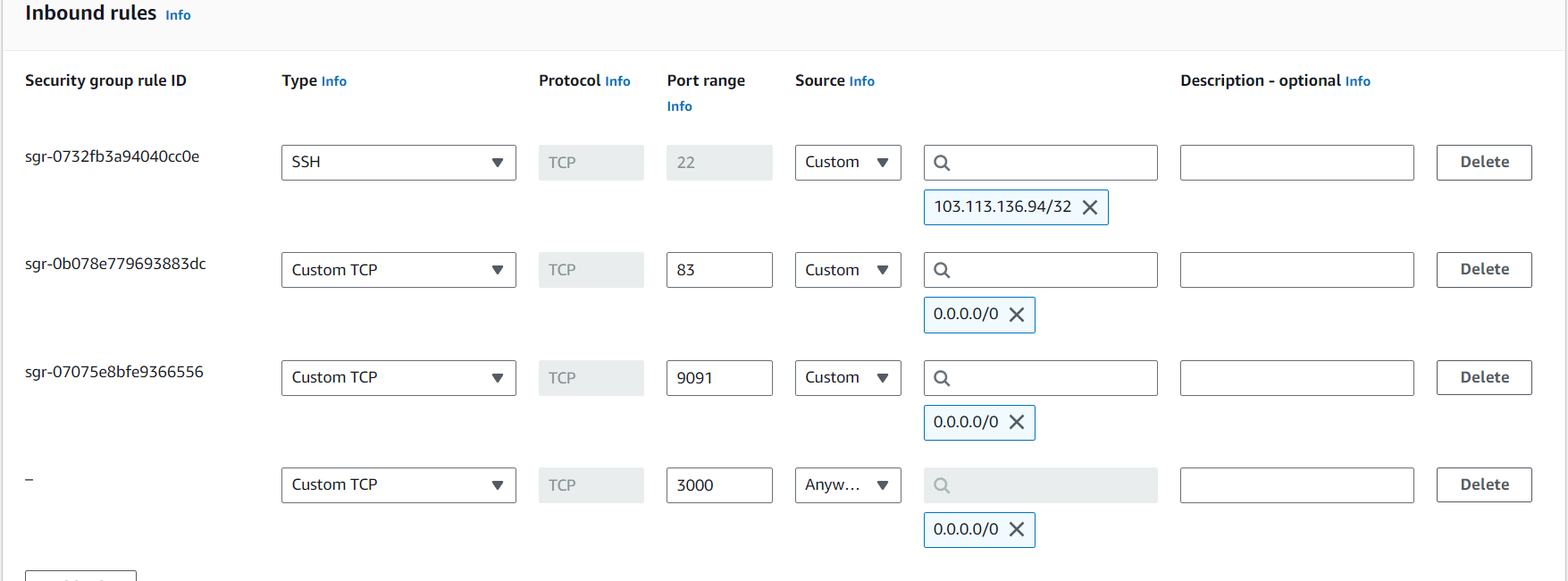
**And got Prometheus server as below:**



**As we can see Targets are showing in UP state.**

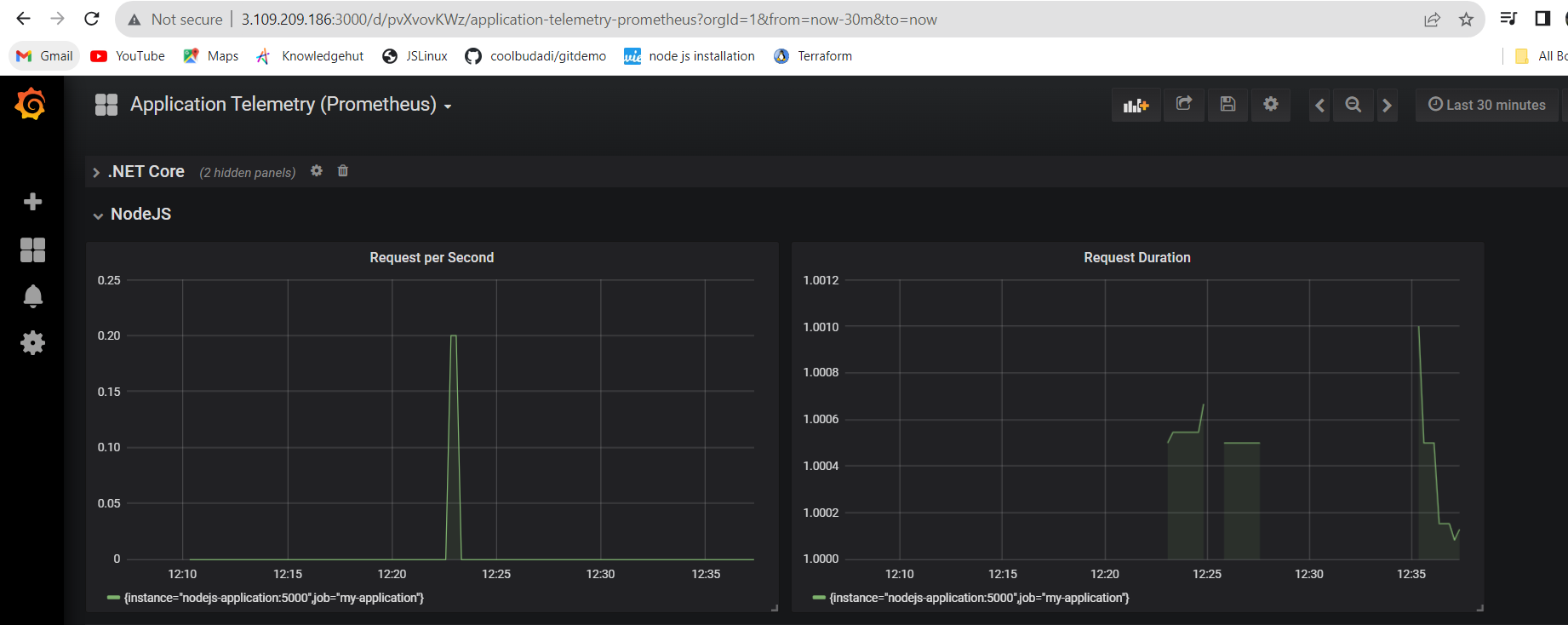


**As Grafana running on Port 3000 so added as entry to SG as below:**



**And we got below result as below:**

**We created Grafana dashboard for monitoring node js application as below:**



Git Repo for application source:

<https://github.com/marcel-dempers/docker-development-youtube-series/tree/master/monitoring/prometheus/nodejs-application>

Prometheus File:

global:

scrape\_interval: 5s

evaluation\_interval: 30s

scrape\_configs:

- job\_name: my-application

honor\_labels: true

static\_configs:

- targets: ['go-application:5000','python-application:5000','dotnet-application:5000','nodejs-application:5000']

Docker Compose File:

version: "3"

services:

nodejs-application:

build:

context: ./nodejs-application

container\_name: nodejs-application

image: nodejs-application

ports:

- "83:5000"

prometheus:

container\_name: prometheus-svc

image: prom/prometheus

ports:

- "9091:9090"

command: --config.file=/etc/prometheus/prometheus.yaml

volumes:

- ./prometheus.yaml:/etc/prometheus/prometheus.yaml

grafana:

image: grafana/grafana:5.0.4

ports:

- "3000:3000"

environment:

- GF\_AUTH\_BASIC\_ENABLED=false

- GF\_AUTH\_ANONYMOUS\_ENABLED=true

- GF\_AUTH\_ANONYMOUS\_ORG\_ROLE=Admin

grafana-dashboards:

image: alpine:3.10

depends\_on:

- grafana

volumes:

- ./grafana-data:/grafana

command: >

/bin/sh -c "

apk add --no-cache curl

echo 'waiting for grafana'

sleep 5s

cd /grafana

curl --request POST http://grafana:3000/api/datasources --header 'Content-Type: application/json' -d @datasources.json

curl --request POST http://grafana:3000/api/dashboards/db --header 'Content-Type: application/json' -d @dashboard.json"