Monitoring tool setup

Glance

1.Create a docker file for glance

```
vim docker-compose.yml
```

2.Add the following content

```
version: '3'
services:
  glances:
    image: nicolargo/glances
    ports:
      - "61208:61208"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    command: ["glances", "-w"]
```

This docker-compose.yml file pulls the official Glances Docker image from Docker Hub, maps the necessary port for the web interface, and mounts the Docker socket for container monitoring.

3.Run the Glances container

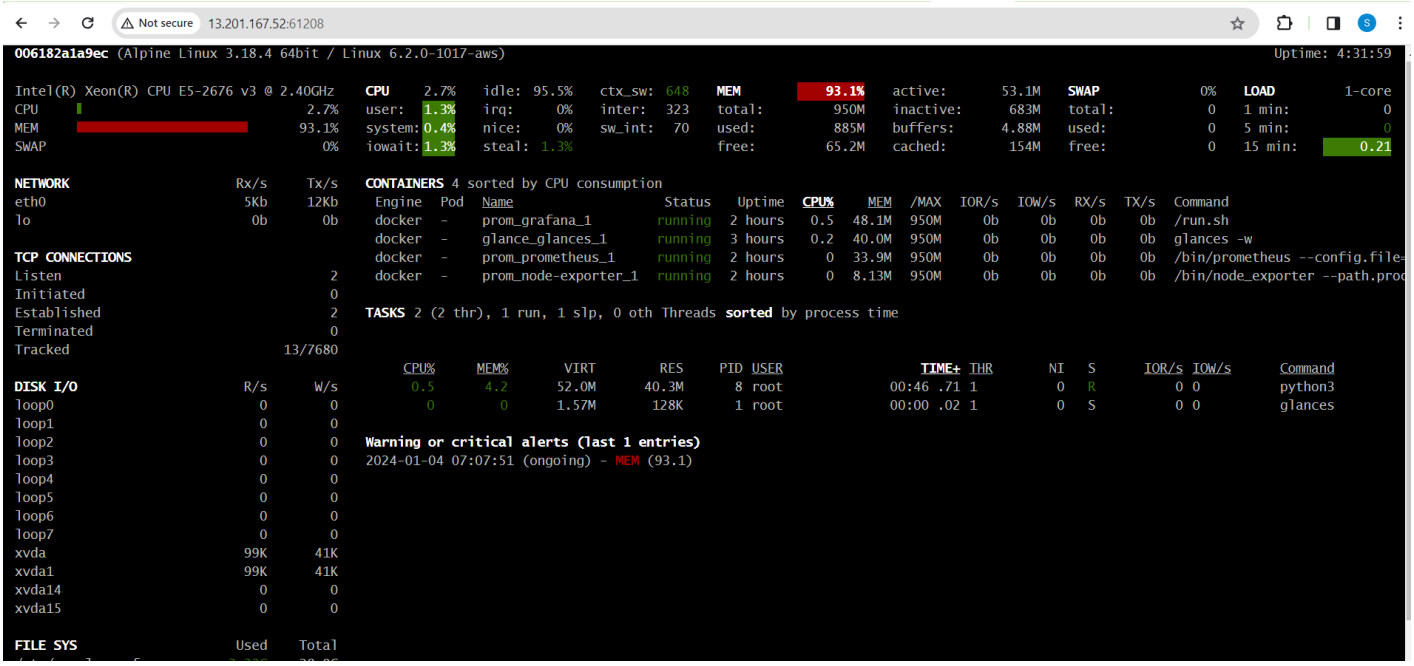
Open a terminal in the directory where you saved the docker-compose.yml file and run the following command:

```
docker-compose up -d
```

This command will download the Glances Docker image and start the container in the background.

4.Access the Glances web interface

Open your web browser and go to `http://localhost:61208`. You should see the Glances web interface, where you can monitor various system metrics.



Promethus and Grafana with Node-Exporter

1.Install Docker and Docker Compose

Make sure you have Docker and Docker Compose installed on your system.

```
apt install docker docker-compose -y
```

2.Create a directory for your monitoring setup Create a new directory and navigate into it.

```
mkdir monitoring
cd monitoring
```

3.Create a `docker-compose.yml` file with the following content

```
vim docker-compose.yml
```

```
version: '3'
services:
  prometheus:
    image: prom/prometheus
    ports:
      - "9090:9090"
    volumes:
      - ./prometheus:/etc/prometheus
```

```
    command:
      - "--config.file=/etc/prometheus/prometheus.yml"
      - "--storage.tsdb.path=/prometheus"
      - "--web.enable-lifecycle"

  grafana:
    image: grafana/grafana
    ports:
      - "3000:3000"
    environment:
      - GF_SECURITY_ADMIN_PASSWORD=admin
    depends_on:
      - prometheus

  node-exporter:
    image: prom/node-exporter
    ports:
      - "9100:9100"
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    command:
      - '--path.procfs=/host/proc'
      - '--path.sysfs=/host/sys'
      - '--collector.filesystem.ignored-mount-points="^/(sys|proc|dev|host|etc)($$|/)"'
```

This docker-compose.yml file defines two services - Prometheus and Grafana. Prometheus will run on port 9090, and Grafana will run on port 3000. The Prometheus configuration file (prometheus.yml) and Grafana admin password are also specified.

Inside the monitoring directory, create directory for prometheus

```
mkdir prometheus
cd prometheus
```

4.Create Prometheus Configuration

Create a prometheus.yml file in the prometheus directory . This file will define the Prometheus configuration. Here's a basic example:

```
```

```
vim prometheus.yml
```

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['node-exporter:9100']
```

Save the file in the prometheus directory (which is mounted into the Prometheus container).

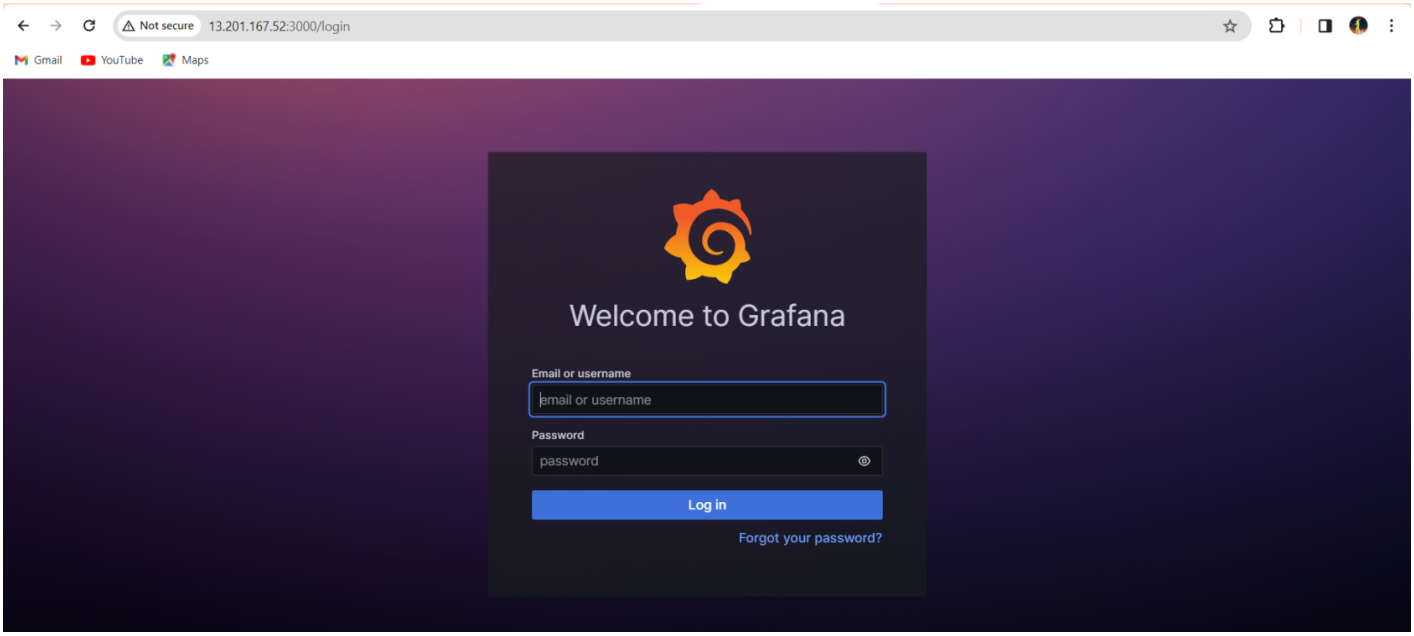
5.Run the Docker containers

```
docker-compose up -d
```

This command will download the necessary Docker images and start Prometheus and Grafana in the background.

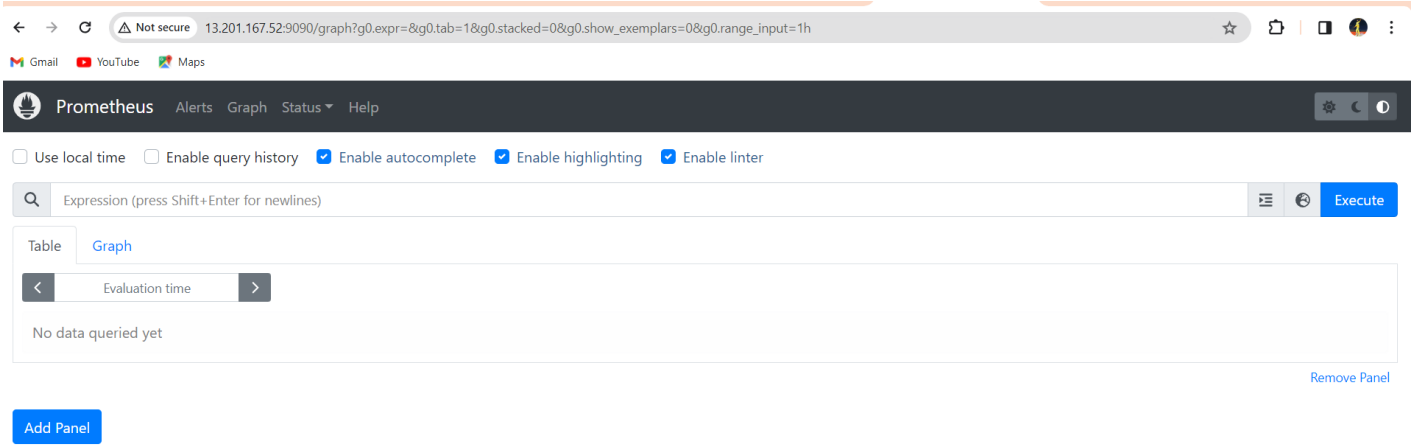
6.Access Grafana

Open your web browser and go to `http://localhost:3000`. Log in with the default credentials (admin/admin). Change the password as needed.



7.Access Prometheus

Open your web browser and go to `http://localhost:9090`

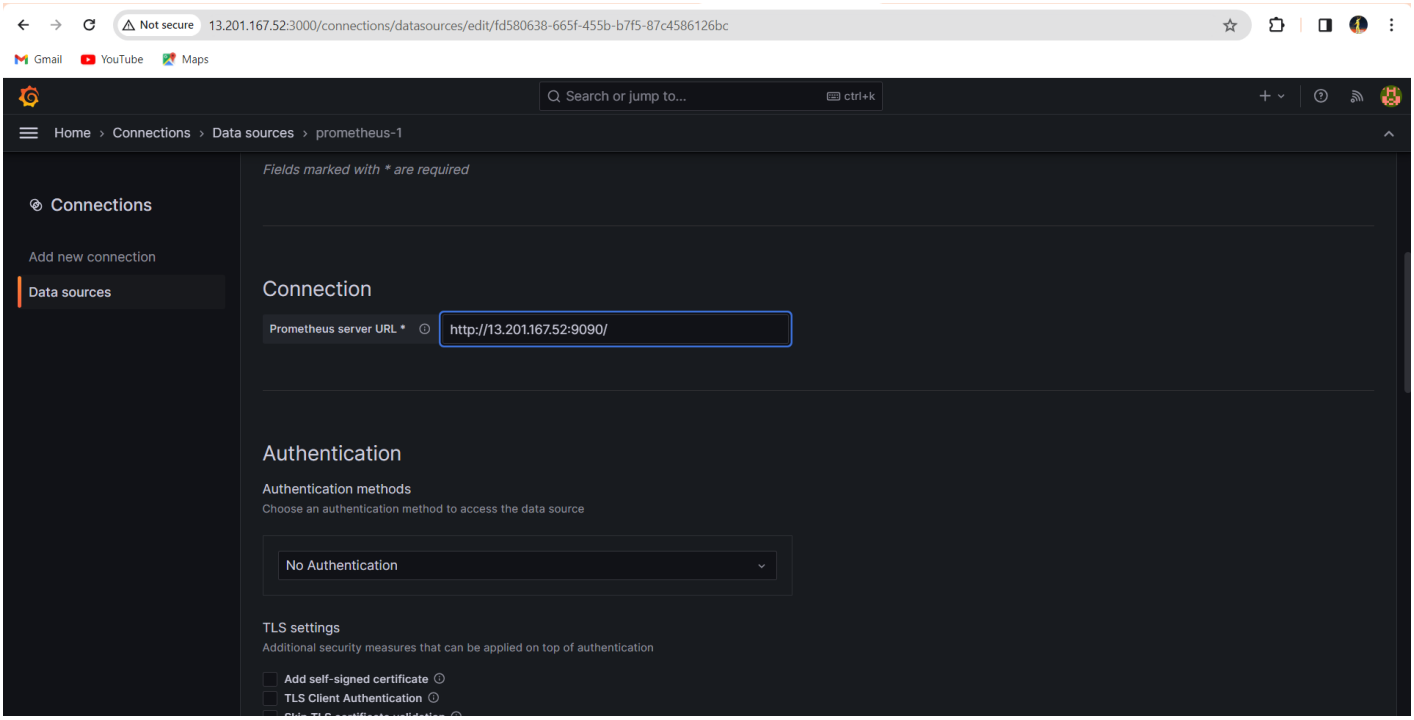


8.Access node-exporter

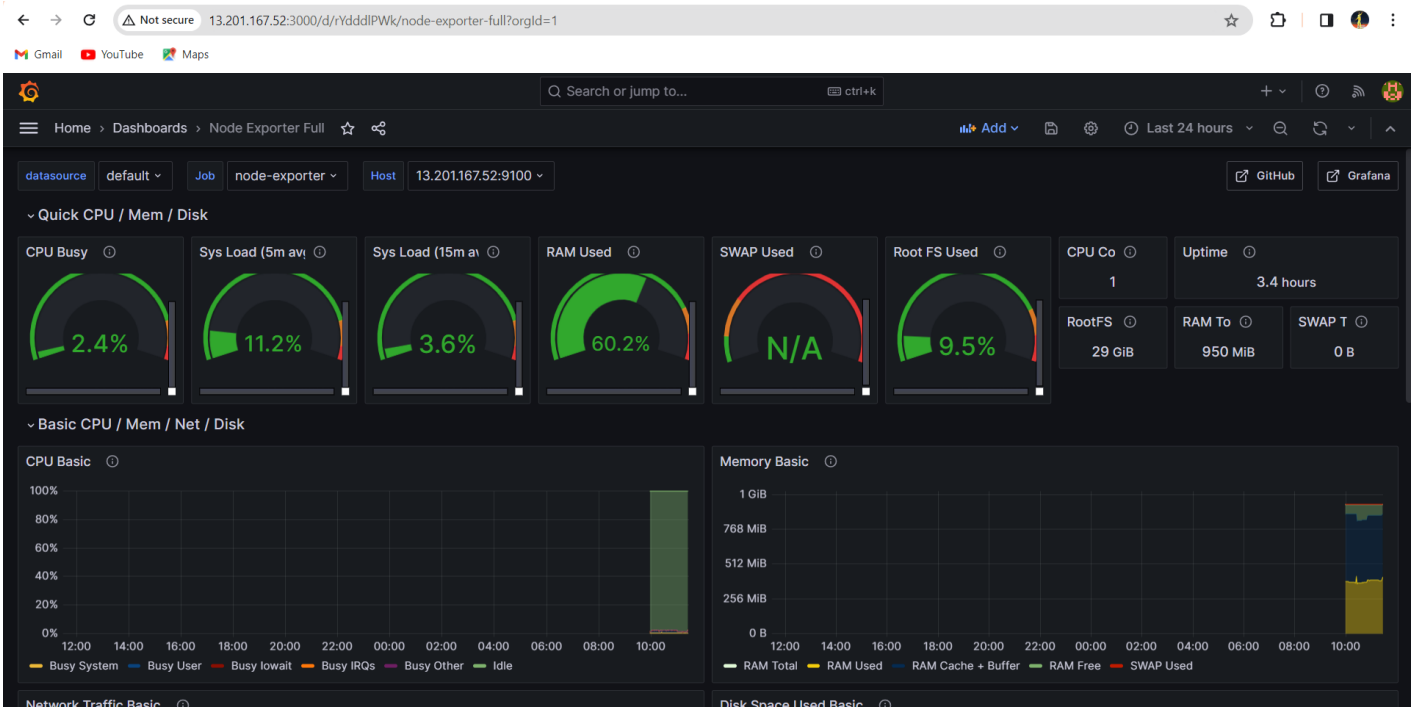
Open your web browser and go to `http://localhost:9100`

9.Add Prometheus as a Data Source

In Grafana, go to "Settings" -> "connection" --> "Data Sources" -> "Add your first data source"-> "Select Promethus." Choose Prometheus and set the URL to `http://prometheus:9090`.



10.Add the dashboard for Node-Exporter In Grafana, (+) symbol on right side corner-> click -> "select import dashboard" -> For node-exporter (1860),want another dashboard means search in google,-> "give load"->"select promethus default"->give import"



Set the email alerts using grafana

11.login docker grafana

Go to server, login docker grafana

```
docker exec -it -u root <cont_name> /bin/bash
```

12.Edit the grafana ini file

```
vim /etc/grafana/grafana.ini
```

Change the following line based on your needs

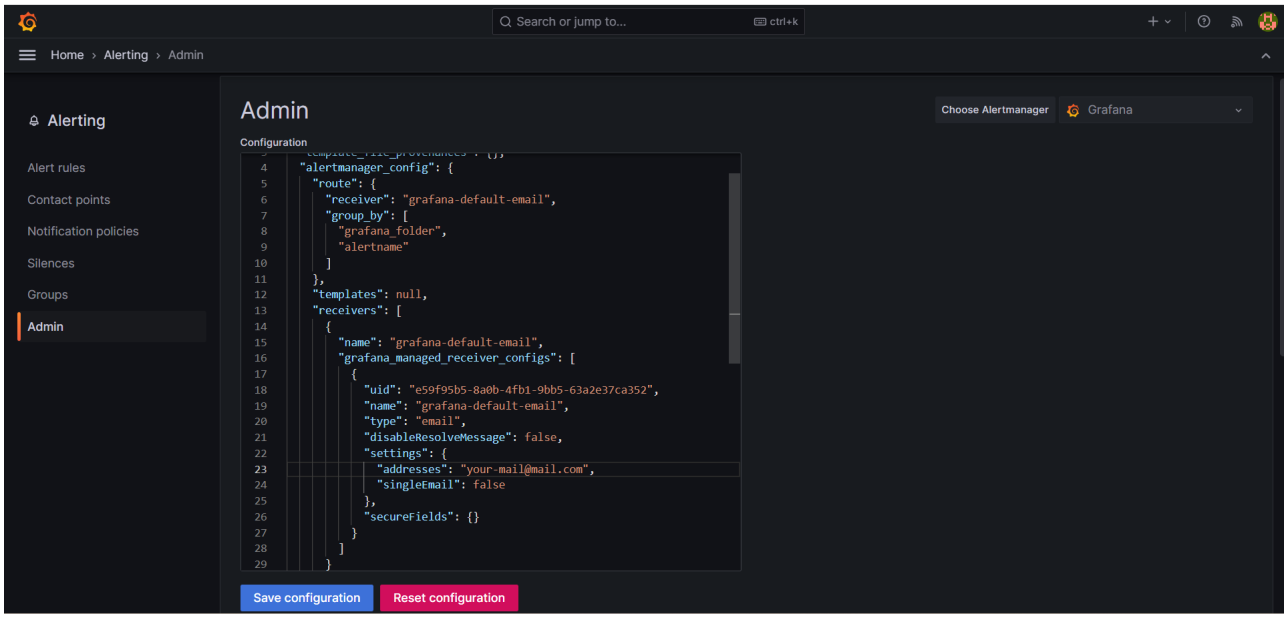
Go to smtp line, modify the changes

```
[smtp]
enabled = true
host = smtp.gmail.com:587
user = your@gmail.com
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
password = your_app_password
```

13.Set the alert in grafana

Go to settings-> "alerting" -> "new alert rule"

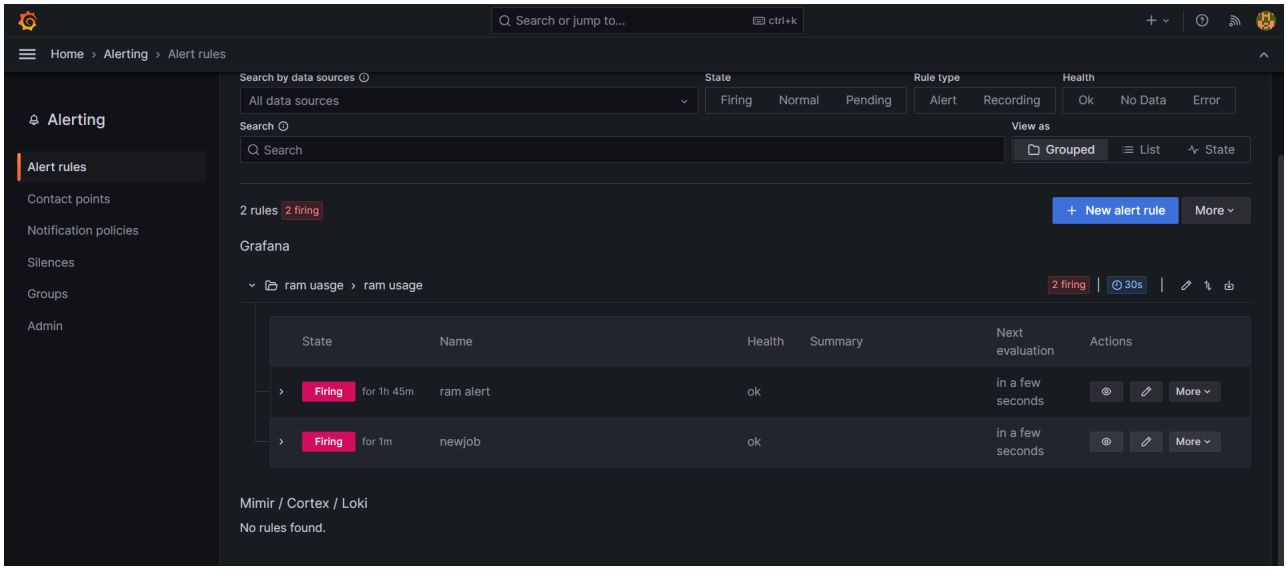
1. Enter alert rule name
2. Define query and alert condition
   a.select the metrics
   b.If you add another metrics means -> "click add query"-> add your query
3. Go to Expression section
   a. In reduce (input give query part(A or B or others) -->in function part give (Last, mode: strict)
   b. In Threshold(input give reduce part(B or C like that) --> Is above section give the time condition
4. Set evaluation behavior
   a. Select folder or create a new folder
   b.select or create Evaluation group
   c. Set the pending time
5. Save the rule
6. Go to admin section in alerting
   Replace your receiving mail id
   Save the configuration



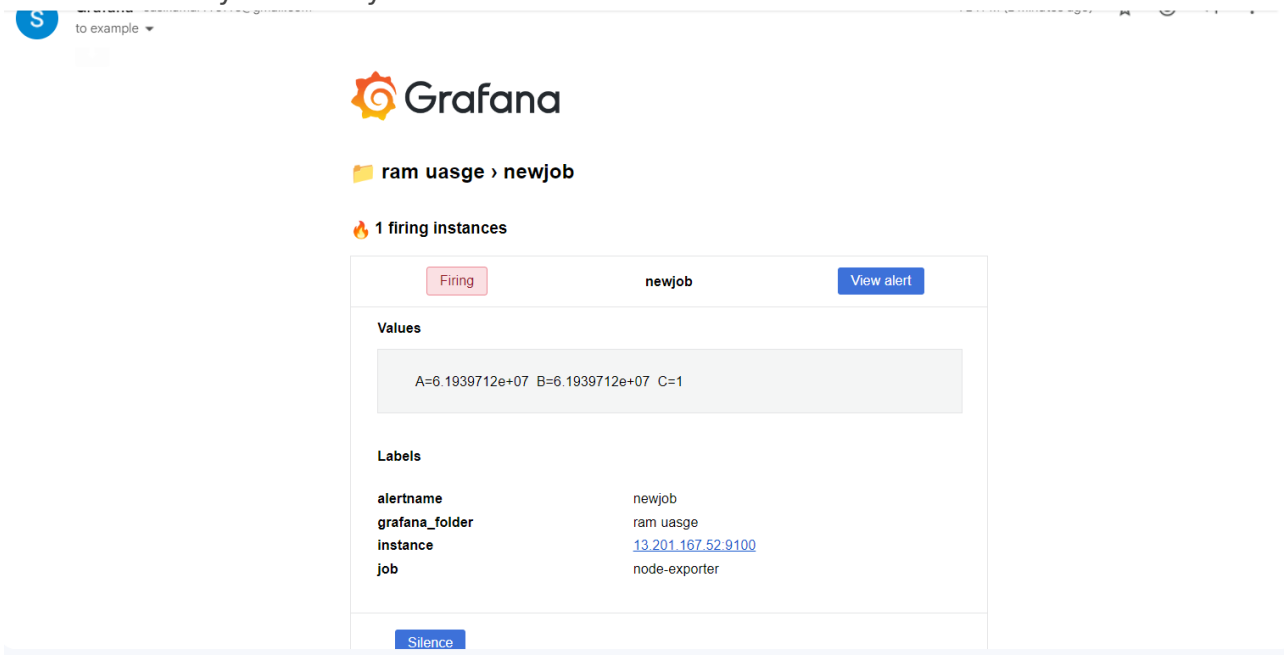
   If the usages are increase, it will notify on your mail

14.Check the mail

1. Check the alert will firing



2. Check the notify alerts in your mail



This is a basic setup, and you may need to customize configurations based on your specific needs and environment.

Nagios Monitoring tool

1.Create a Directory

Create a directory to organize your Nagios configuration files. For example:

```
mkdir nagios-config
cd nagios-config
```

2.Create Docker Compose file

Create a docker-compose.yml file in the nagios-config directory:

```
vim docker-compose.yml
```

```
version: '3'
services:
  nagios:
    image: jasonrivers/nagios:latest
    ports:
      - "8080:80"
    environment:
      - NAGIOS_TIMEZONE=UTC
    volumes:
      - ./config:/opt/nagios/etc
```

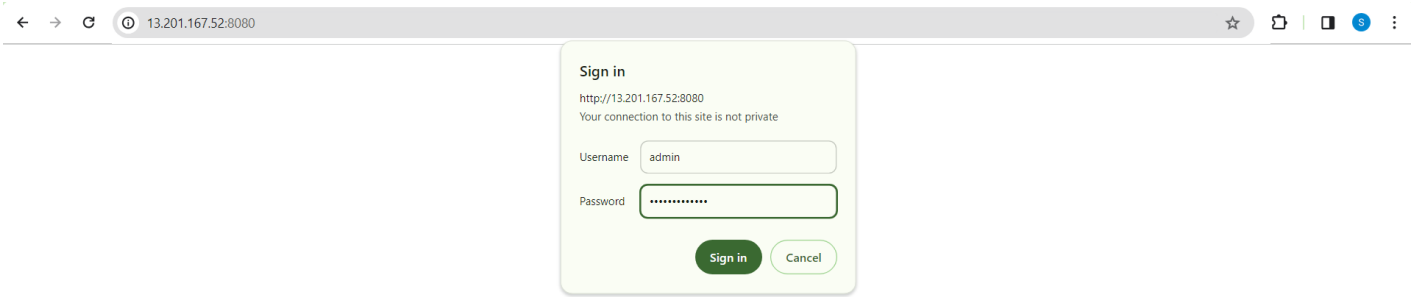
This configuration mounts a local directory (./config) into the Nagios container, allowing you to customize Nagios configurations.

3.Run Docker Compose

Run the following command in the nagios-config directory to start Nagios:

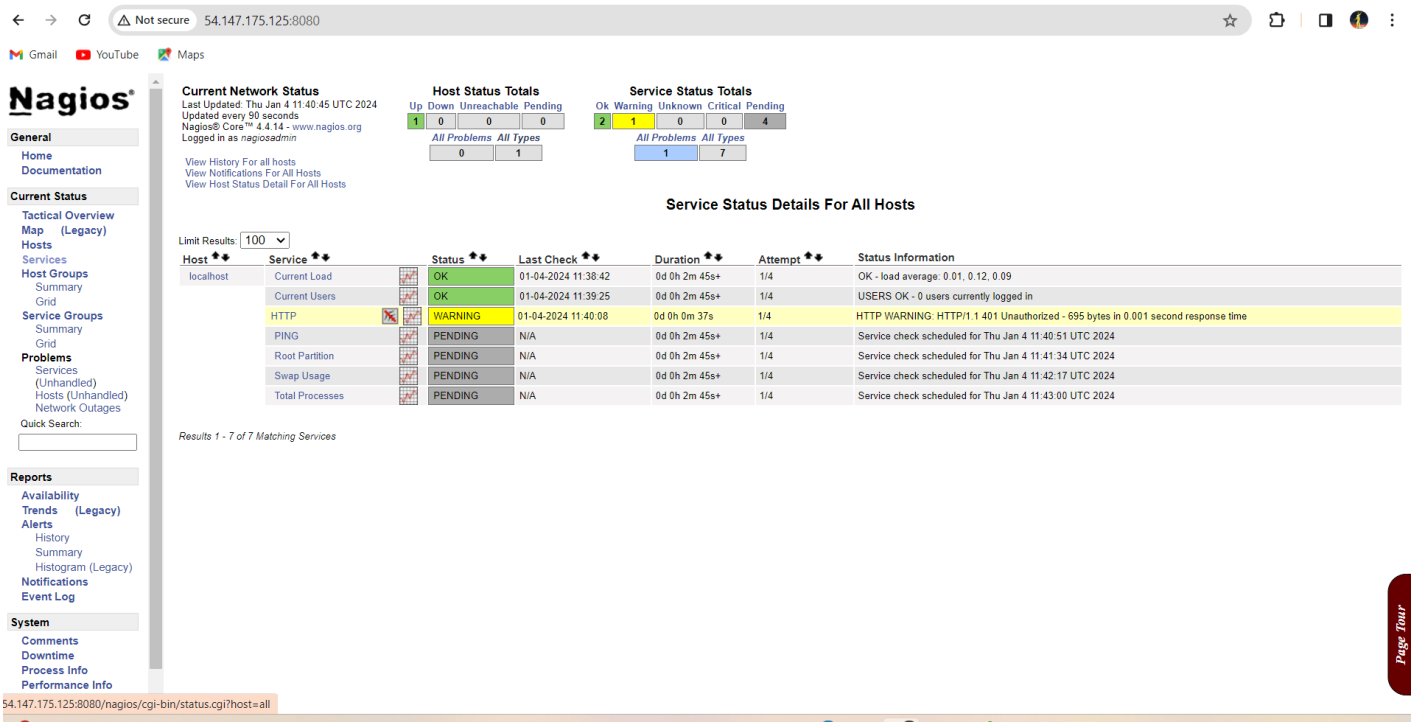
```
docker-compose up -d
```

4.Access Nagios Web Interface Open your web browser and go to `http://localhost:8080/nagios`. Log in with the credentials specified in the docker-compose.yml file (admin/adminpassword).



5.Check the services

In Nagios, go to services on left side



You want add the remote host means modify the files based on needs

```
vi /path/to/nagios-config/config/etc/file.cfg
```

Revision #24

Created Wed, Jan 3, 2024 11:34 AM by sasi

Updated Tue, Feb 20, 2024 10:23 AM by sasi