**PASO1: Crear infrastructura**

* Crear una maquina EC2 del tipo t3.small con la AMI “Amazon Linux 2”
* Durante la creacion asegurate de:
  + Crear una clave en la seccion “par de claves”, y descargarla
  + En la seccion de “grupo de seguridad” Añadir los puertos 3000 y 9090 a las reglas de entrada del grupo de seguridad

**PASO 2: Instalar dependencias**

* Instalar docker y docker compose con los siguientes comandos

|  |
| --- |
| #Actualizar librerias  sudo yum update –y  #instalar docker  sudo yum install -y docker  #arrancar docker  sudo service docker start  #añadir nuestro usuario al grupo docker  sudo usermod -a -G docker $(whoami)  #instalar pip3  sudo yum install -y python3-pip  #descargarse el binario para utilizar docker-compose  sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  #dar permisos de ejecucion al archivo binario  sudo chmod +x /usr/local/bin/docker-compose  #comprobar que docker compose esta instalado  docker-compose --version |

**PASO 3: Añadir configuracion**

* Crear un archivo **docker-compose.yml** con el siguiente contenido

|  |
| --- |
| version: '3'  services:  grafana:  image: grafana/grafana:latest  ports:  - "3000:3000"  volumes:  - grafana-data:/var/lib/grafana  - ./provisioning:/etc/grafana/provisioning  environment:  - GF\_SECURITY\_ADMIN\_USER=admin  - GF\_SECURITY\_ADMIN\_PASSWORD=admin  networks:  - monitoring    node-exporter:  image: prom/node-exporter:latest  ports:  - "9100:9100"  networks:  - monitoring    prometheus:  image: prom/prometheus:latest  ports:  - "9090:9090"  volumes:  - ./prometheus.yml:/etc/prometheus/prometheus.yml  networks:  - monitoring    networks:  monitoring:    volumes:  grafana-data: |

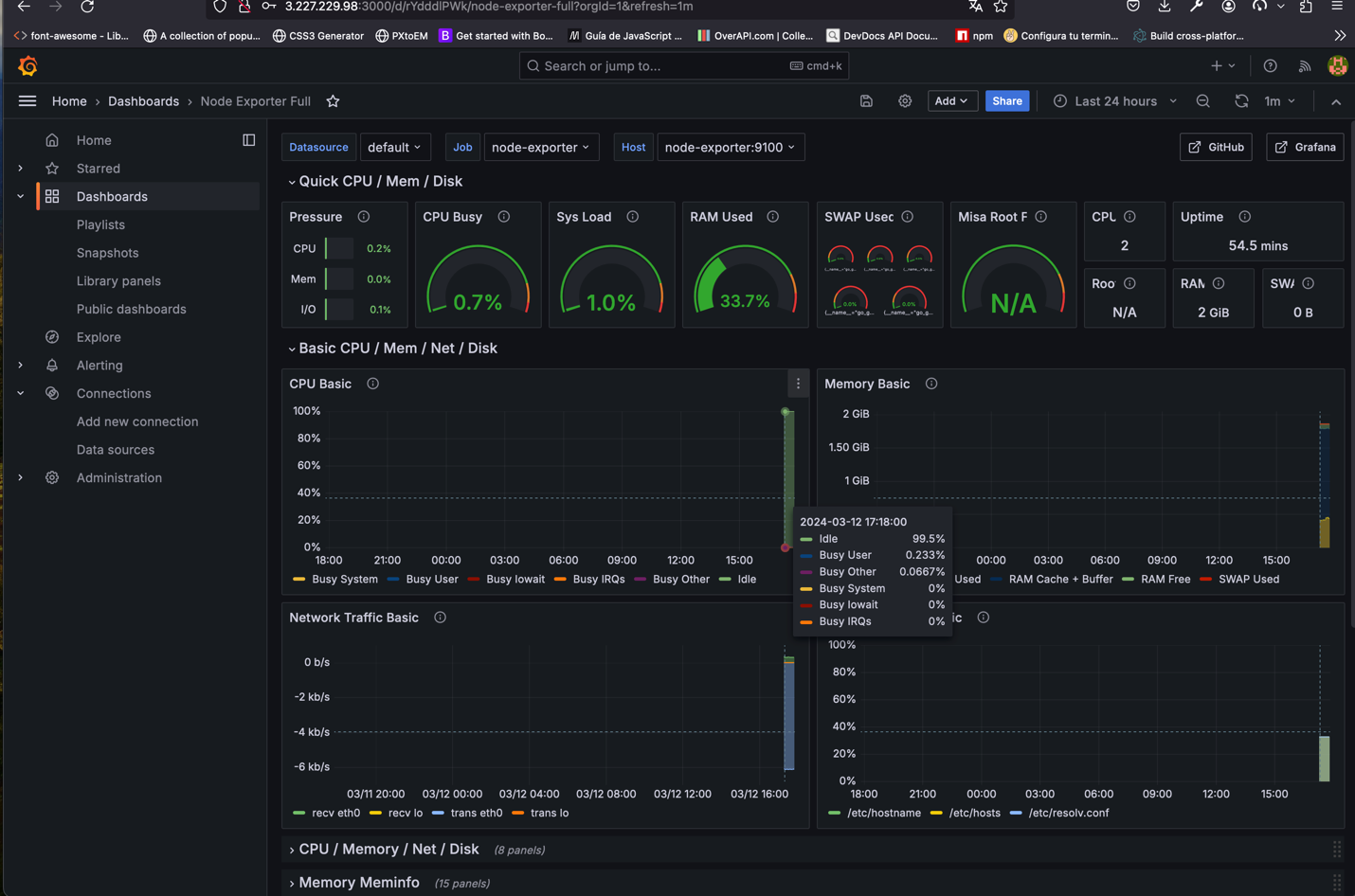
* En este archivo de Docker Compose:
  + El servicio Grafana se define con la imagen de Grafana, exponiendo el puerto 3000 para la interfaz web. Monta un volumen para los datos de Grafana y un directorio de aprovisionamiento que contiene la configuración del origen de datos.
  + El servicio Node Exporter ejecuta Prometheus Node Exporter, exponiendo métricas en el puerto 9100.
  + El servicio Prometheus ejecuta el propio Prometheus, exponiendo el puerto 9090. Monta un archivo prometheus.yml para la configuración.
* Crea un archivo **prometheus.yml** con el siguiente contenido

|  |
| --- |
| global:  scrape\_interval: 15s  scrape\_configs:  - job\_name: 'prometheus'  static\_configs:  - targets: ['localhost:9090']  - job\_name: 'node-exporter'  static\_configs:  - targets: ['node-exporter:9100'] |

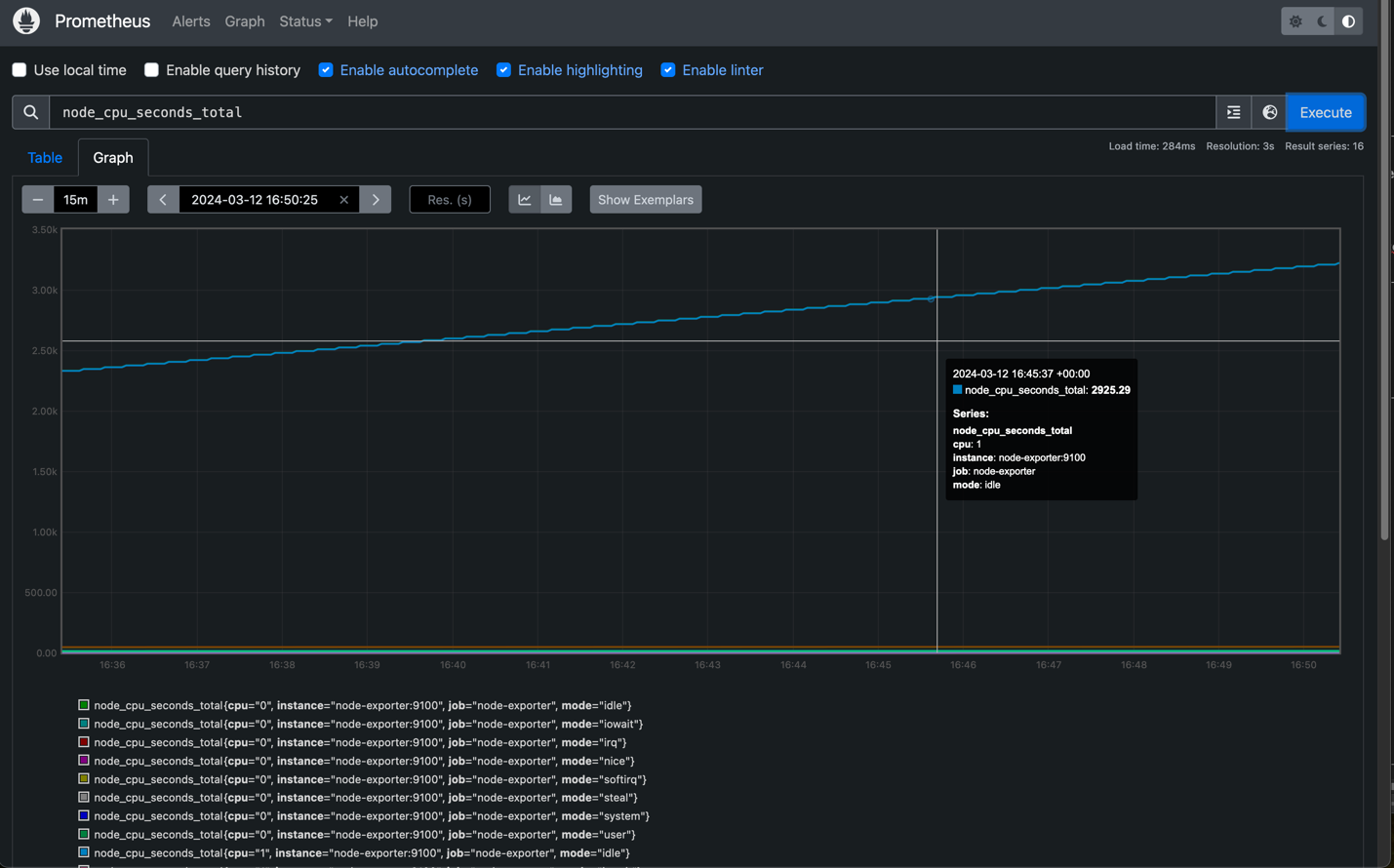
* Crea un directorio con el nombre **provisioning, con el comando** 
  + **mkdir provisioning**
* Dentro de este directorio añade un archivo **datasources.yaml** con el siguiente contenido

|  |
| --- |
| apiVersion: 1  datasources:  - name: Prometheus  type: prometheus  access: proxy  url: http://prometheus:9090 |

* Para desglegar todo el Stack, ejecuta el comando
  + **sudo docker-compose up**



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Urls:

IpInstancia:3000 “Grafana”

IpInstancia:9090 “Prometheus”

En grafana la configuración del origen de datos: http://prometheus:9090