

Universidad de Costa Rica

Escuela de las Ciencias de la Computación e Informática

CI-0125 Desempeño y Experimentación

Docente: Ariel Mora Jiménez

Grupo 1

Proyecto Sistema de Monitoreo

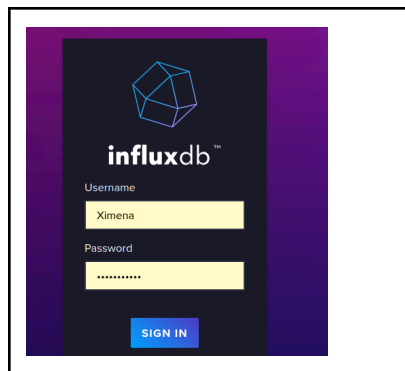
Alumna:

Ximena Marín Sánchez, C14448

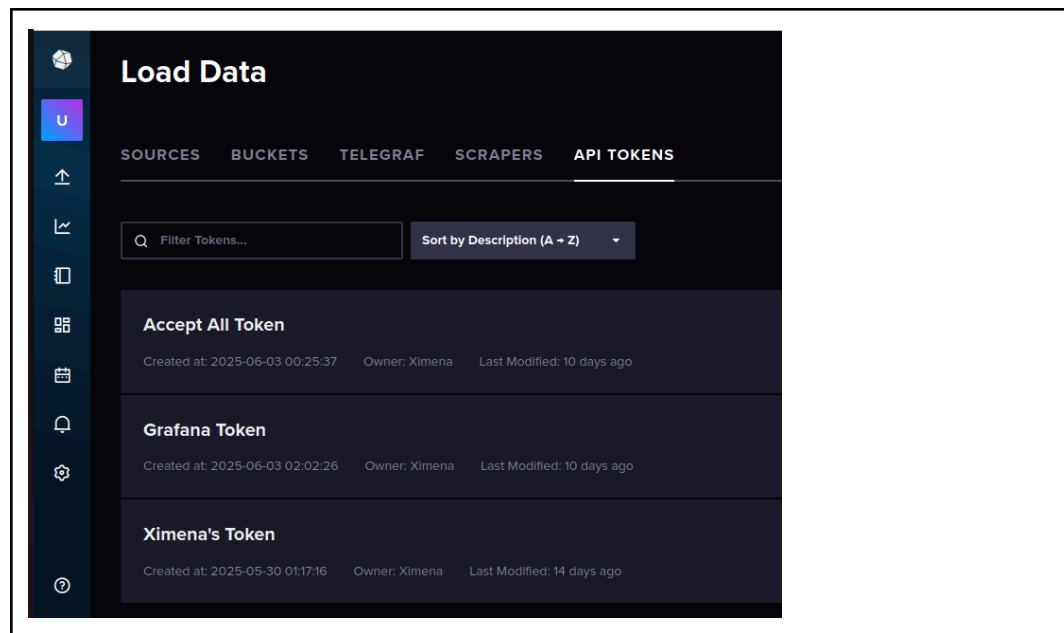
Proyecto - Sistema de monitoreo con Telegraf, InfluxDB y Grafana

Se instala la máquina virtual de Mediación virtual

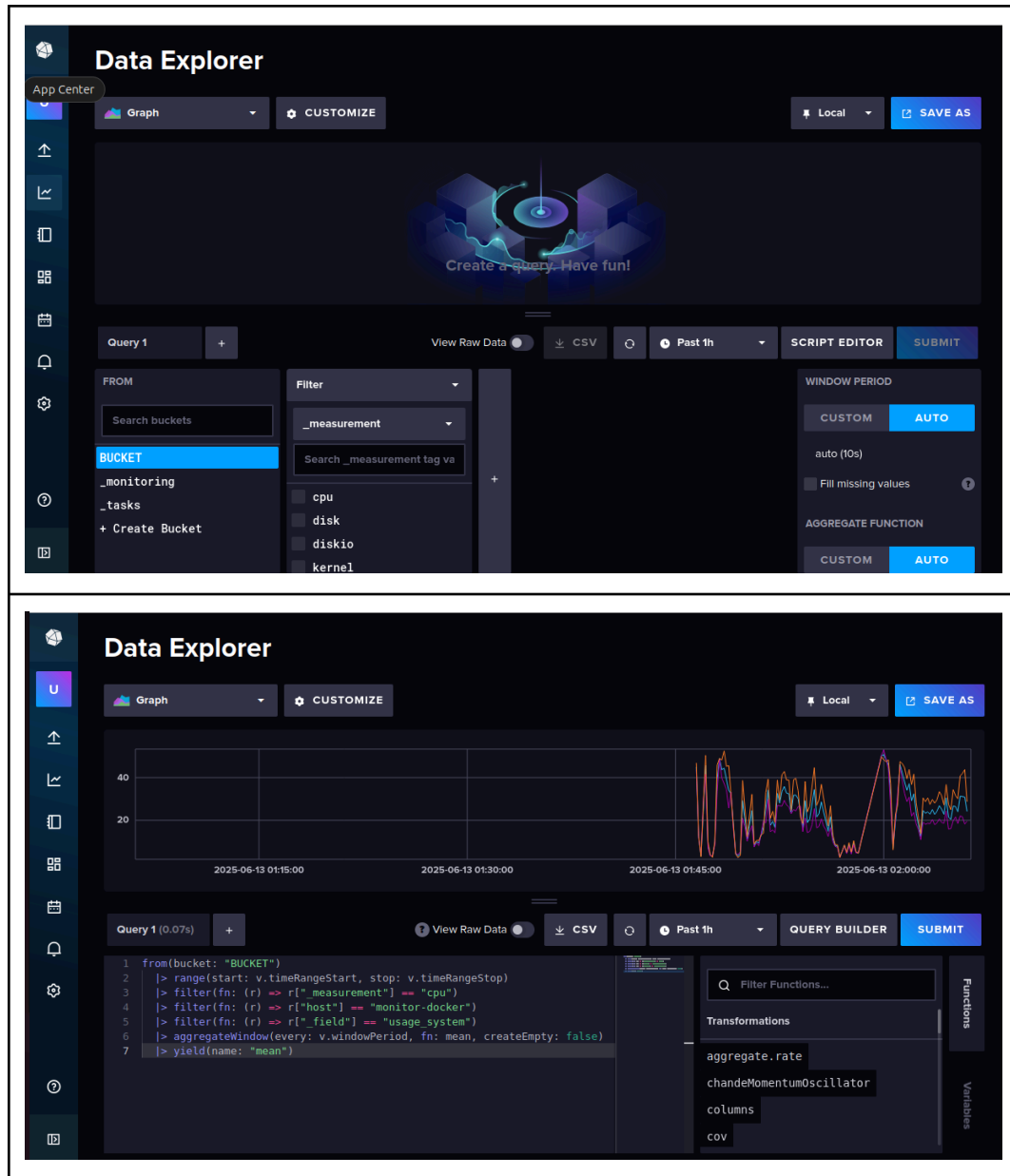
1. Pasos para instalar la imagen de InfluxDB en su versión 2
2. Se entra al siguiente sitio web, a la sección de tag y se escoge el número 2
 - https://hub.docker.com/_/influxdb/tags
 - Se abre la terminal y se escribe:
 - `docker pull influxdb:2`
 - `docker image list`
3. Levante un contenedor de InfluxDB usando el comando `'docker run'`. Considere que es necesario exponer el puerto del contenedor a la máquina host, configurar el uso de volúmenes persistentes, así como configurar el usuario administrador, organización y bucket
 - `docker run -d -p 8086:8086 -v "$PWD/data:/var/lib/influxdb2" -v "$PWD/config:/etc/influxdb2" -e DOCKER_INFLUXDB_INIT_MODE=setup -e DOCKER_INFLUXDB_INIT_USERNAME=Ximena -e DOCKER_INFLUXDB_INIT_PASSWORD=monitor1234 -e DOCKER_INFLUXDB_INIT_ORG=UCR -e DOCKER_INFLUXDB_INIT_BUCKET=BUCKET influxdb: 2`
4. Ingrese a la interfaz web del contenedor de InfluxDB para verificar su funcionamiento
 - a. Se levanta mediante un `docker compose up -d`
 - b. Se ingresa al link: <http://localhost:8086/signin>
 - c. Se muestra la interfaz de inicio de sesión



5. Instalar el cliente de Telegraf directamente en el sistema operativo
 - a. `sudo apt-get update && sudo apt-get install telegraf`
 - b. Adicional:
 - i. Reiniciar telegraf: `sudo systemctl restart telegraf`
 - ii. Levantar telegraf: `sudo systemctl start telegraf`
6. Configure el cliente de Telegraf para que se conecte al contenedor de InfluxDB****
 - a. Creación de Token



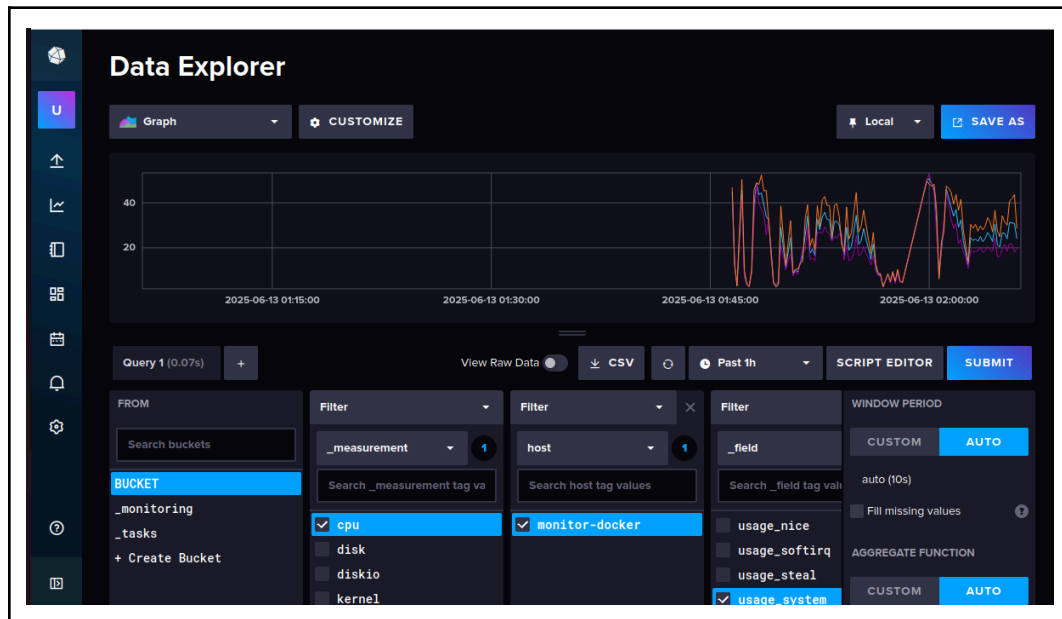
- b. Se crea un token de acceso:**
 - i. *Accept All token en InfluxDB*
 - ii. *oofHbBCibxW5bOuZSu0DgkTX6R7VMJrmLk9BTxsZFOZAqPcyyyc2_m2Q4XrEhmlESxY3CIJpeL7SCgo8tdFJVLw==*
7. Verifique en la interfaz web del contenedor de InfluxDB la conexión de Telegraf
 - a. Data Explorer inicial y ejemplo



8. Usando la interfaz web del contenedor de InfluxDB realice consultas para obtener métricas de CPU, RAM, entre otras

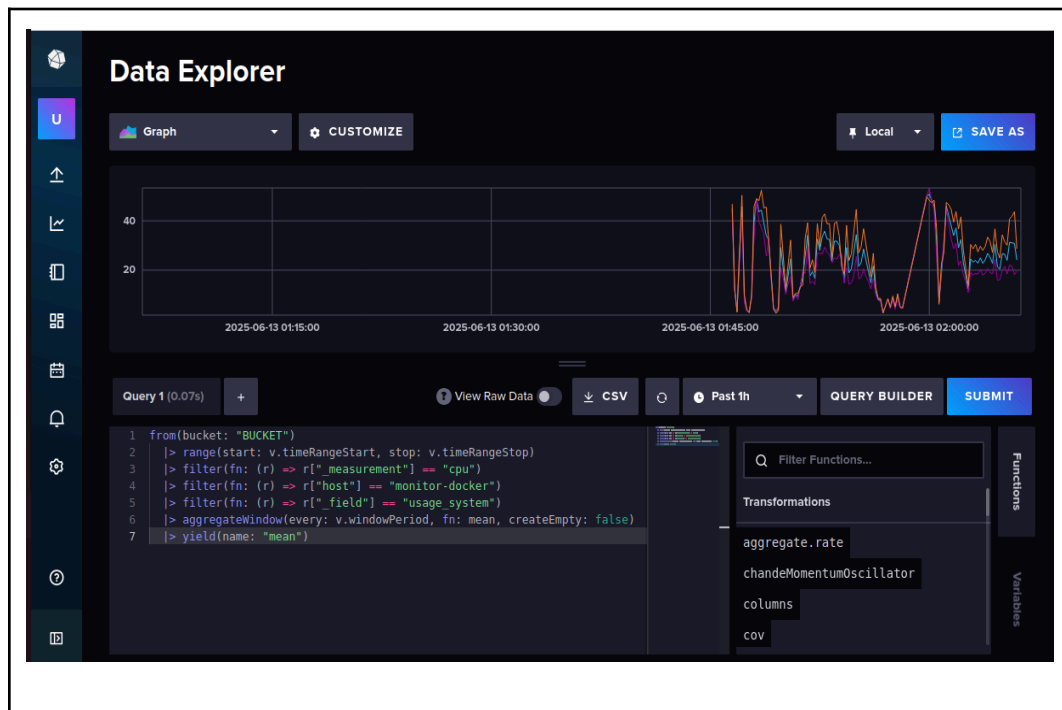
Los siguientes ejemplos mostrados a continuación son con la finalidad de tener una breve visualización de las métricas, se realizaron mayor cantidad de consultas que fueron guardadas en un archivo para posterior uso para crear un dashboard en Grafana de las consultas realizadas.

a. Ejemplo mostrando métricas con CPU

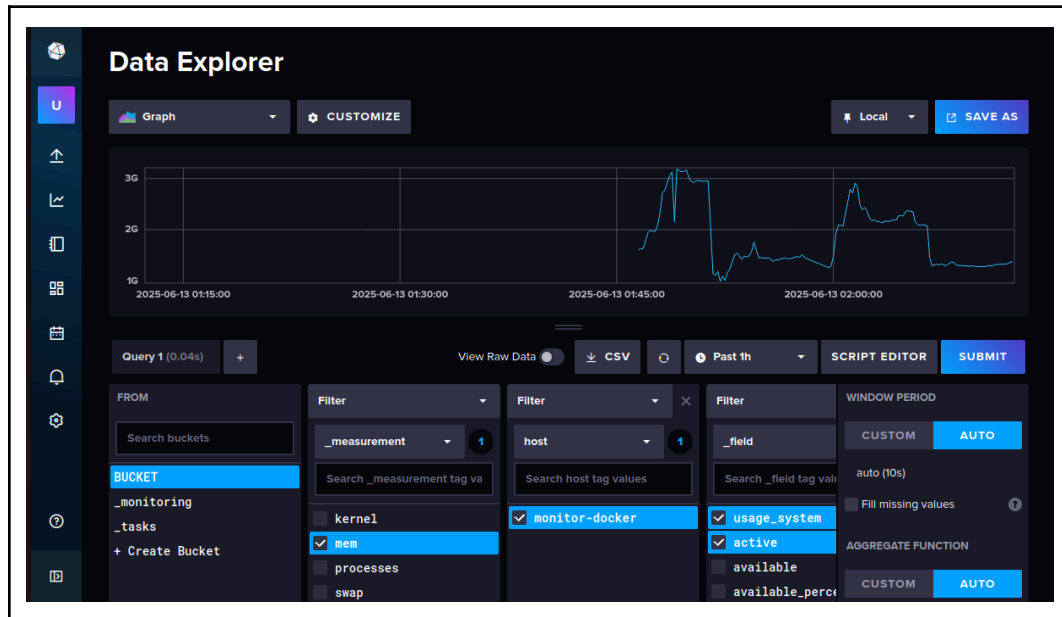


b. Usando el botón de 'Script Editor' se tiene acceso a la consulta en el lenguaje Flux como se muestra en el siguiente ejemplo

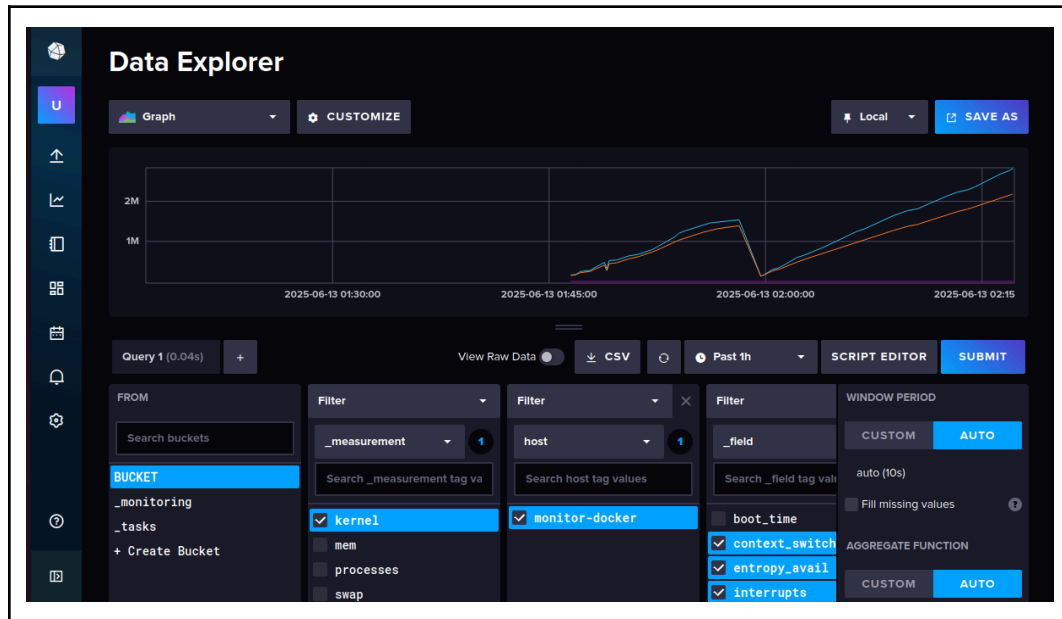
c. Ejemplo FLUX lo que se observa en el recuadro que inicia con
`from(bucket: "BUCKET")`



d. Ejemplo usando memoria:



e. Ejemplo usando kernel



9. Instale la imagen de Grafana en su versión más reciente
 - a. touch docker-compose.yaml
 - b. nano docker-compose.yaml
 - c. En el archivo .yaml se instala la imagen de Grafana
 - d. Extracto del docker-compse.yaml

```

grafana:
  image: grafana/grafana
  ports:
    - 3000:3000
  environment:
    - GF_SECURITY_ADMIN_USER=admin
    - GF_SECURITY_ADMIN_PASSWORD=monitor1234
  volumes:
    - grafana-storage:/var/lib/grafana
  depends_on:
    - influx
  restart: unless-stopped

volumes:
  grafana-storage:

```

10. Usando docker-compose levante contenedores para InfluxDB y Grafana. Considere que en el caso de Grafana es necesario exponer el puerto así como configurar el uso de volúmenes persistentes*****

- a. *docker compose up -d*
- b. *docker compose config*
- c. *docker ps -a*

11. Configure Grafana para que use al contenedor de InfluxDB como fuente de datos*****

- a. *docker exec -it docker-grafana-1 /bin/bash*
- b. Escribir el comando: ping influx
 - i. monitor@monitor-docker:~/Docker\$ docker exec -it docker-grafana-1 /bin/bash
 - ii. 25fc2d1161ed:/usr/share/grafana\$ ping influx
- c. Se accede a data sources en grafana
 - i. <http://localhost:3000/connections/datasources>
- d. En el http se escribe: <http://influx:8086>
- e. Se selecciona FLUX como Query Language
- f. Se configura la información como BUCKET y organization
 - i. Organization: UCR
 - ii. Default Bucket: BUCKET
- g. Se presiona la opción de Save & test

Extracto de la configuración de Data sources

The screenshot shows the Grafana interface with the 'Data sources' menu item highlighted in the left sidebar. The main content area displays the configuration for an InfluxDB data source. At the top, there are toggle switches for 'Skip TLS Verify' and 'Forward OAuth Identity'. Below these are the 'Basic Auth Details' fields: 'User' (admin) and 'Password' (configured), with a 'Reset' button. The 'Custom HTTP Headers' section has an 'Add header' button. The 'InfluxDB Details' section includes fields for 'Organization' (UCR), 'Token' (configured), 'Default Bucket' (BUCKET), 'Min time interval' (10s), and 'Max series' (1000), each with a 'Reset' button. At the bottom of this section are 'Delete' and 'Save & test' buttons.

Home > Connections > Data sources > influxdb

Skip TLS Verify ☐

Forward OAuth Identity ☐

Basic Auth Details

User admin

Password configured Reset

Custom HTTP Headers

+ Add header

InfluxDB Details

Organization UCR

Token configured Reset

Default Bucket BUCKET

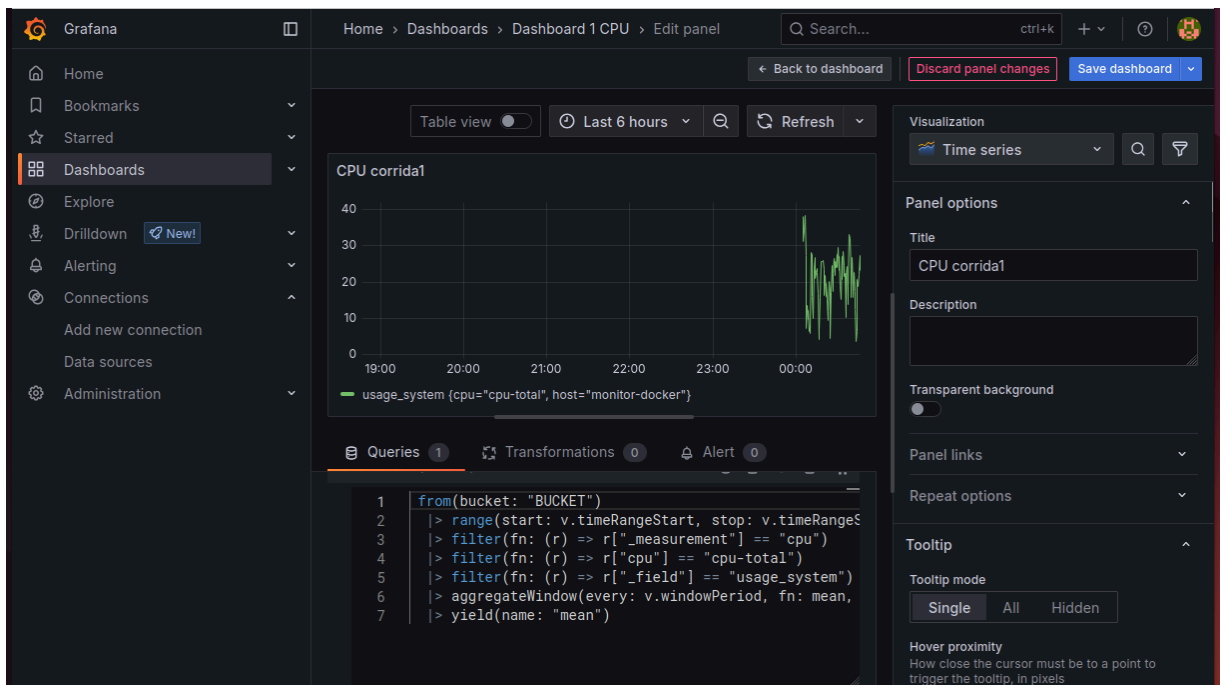
Min time interval 10s

Max series 1000

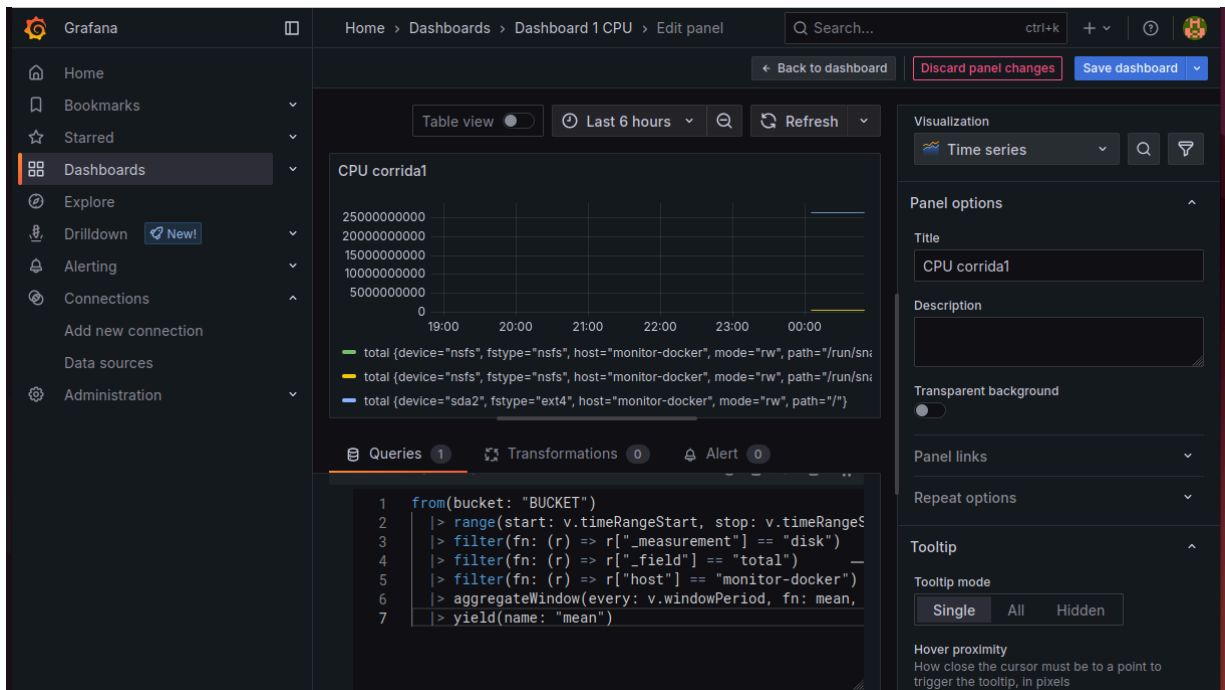
Delete Save & test

12. Cree un dashboard en Grafana utilizando las consultas que realizó en el paso 8

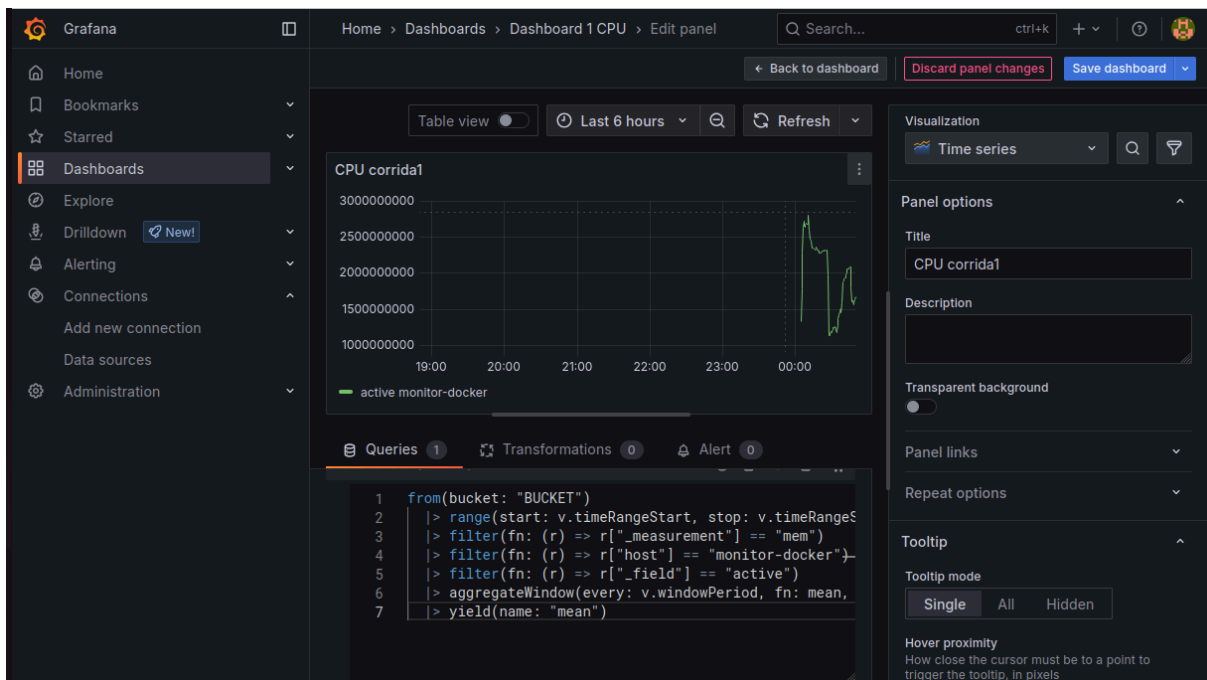
Dashboard de CPU en Grafana



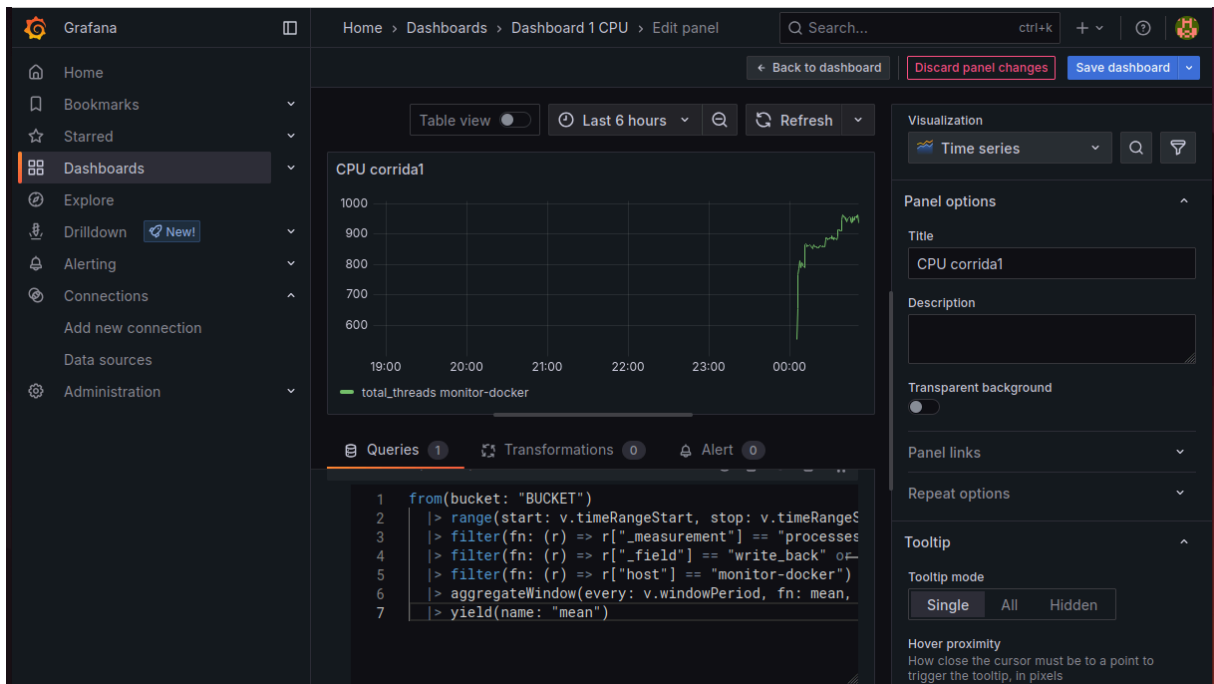
Dashboard de disco en Grafana



Dashboard de memoria en Grafana



Dashboard de procesos en Grafana



Dashboard de kernel en Grafana

