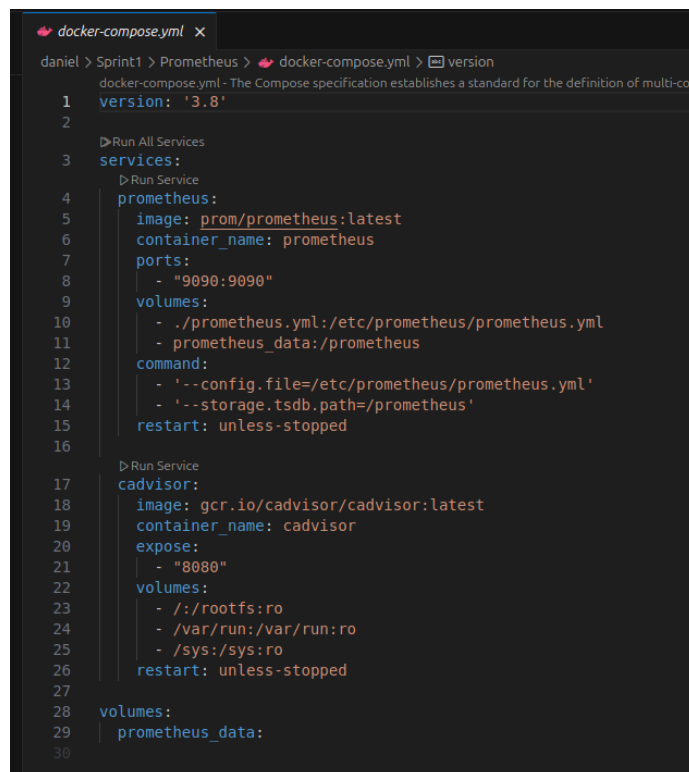


Objetivo

Configurar Prometheus para recolectar métricas de servicios en contenedores mediante Docker Compose y visualizar dichas métricas a través de su interfaz web.

Crear el entorno de trabajo

1. Se creó una carpeta de proyecto (**Prometheus/**) para contener todos los archivos necesarios.
2. Dentro de esta carpeta se guardaron:
 - **docker-compose.yml**: Define los servicios de Prometheus y cAdvisor.
 - **prometheus.yml**: Archivo de configuración personalizado para Prometheus.



```
1 version: '3.8'
2
3 services:
4   prometheus:
5     image: prom/prometheus:latest
6     container_name: prometheus
7     ports:
8       - "9090:9090"
9     volumes:
10      - ./prometheus.yml:/etc/prometheus/prometheus.yml
11      - prometheus_data:/prometheus
12     command:
13       - '--config.file=/etc/prometheus/prometheus.yml'
14       - '--storage.tsdb.path=/prometheus'
15     restart: unless-stopped
16
17   cadvisor:
18     image: gcr.io/cadvisor/cadvisor:latest
19     container_name: cadvisor
20     expose:
21       - "8080"
22     volumes:
23       - /:/rootfs:ro
24       - /var/run:/var/run:ro
25       - /sys:/sys:ro
26     restart: unless-stopped
27
28 volumes:
29   prometheus_data:
```

```
! prometheus.yml x
daniel > Sprint1 > Prometheus > ! prometheus.yml > {} global
prometheus.json - Prometheus configuration File (prometheus.json)
1 global:
2   scrape_interval: 15s
3
4 scrape_configs:
5   - job_name: 'cadvisor'
6     static_configs:
7       - targets: ['cadvisor:8080']
8
```

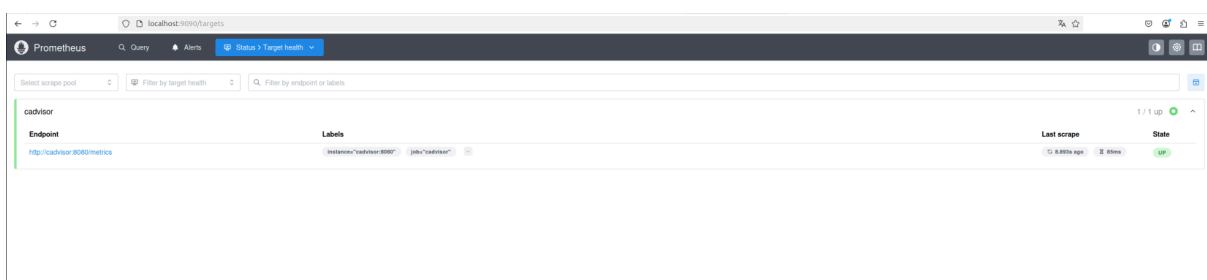
Desde la terminal, en la carpeta del proyecto, se ejecutó “docker-compose up -d”. Esto levantó los contenedores de Prometheus y cAdvisor en segundo plano.

A mayores, levantamos los contenedores que teníamos configurados de ejercicios anteriores para poder analizar sus métricas más adelante.

```
root@daniel-MS-7E28:/home/daniel/Sprint1/Prometheus# docker-compose up -d
Recreating cadvisor ... done
Starting prometheus ... done
root@daniel-MS-7E28:/home/daniel/Sprint1/Prometheus# cd ..
root@daniel-MS-7E28:/home/daniel/Sprint1# cd DockerCompose/
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# ls
docker-compose.yml  nginx  php  src
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose up -d --build
Building php
Step 1/2 : FROM php:7.4-fpm
--> 38f2b691dcb8
Step 2/2 : RUN docker-php-ext-install mysqli
--> Using cache
--> f7ced234d426
Successfully built f7ced234d426
Successfully tagged dockercompose_php:latest
Starting mysql ... done
Starting php ... done
Starting nginx ... done
Starting phpmyadmin ... done
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                               NAMES
9d6a1aad4c1a   gcr.io/cadvisor/cadvisor:latest     "/usr/bin/cadvisor -..." 37 seconds ago Up 36 seconds (healthy) 8080/tcp                                              cadvisor
598b2b99e59a   prom/prometheus:latest              "/bin/prometheus -..." 4 minutes ago  Up 36 seconds  0.0.0.0:9090->9090/tcp, [::]:9090->9090/tcp             prometheus
ca5627eac952   nginx:latest                         "/docker-entrypoint..." 44 hours ago  Up 4 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp                 nginx
17c1a3378eac   phpmyadmin/phpmyadmin               "/docker-entrypoint..." 44 hours ago  Up 4 seconds  0.0.0.0:8081->80/tcp, [::]:8081->80/tcp                 phpmyadmin
5c572d139bc4   dockercompose_php                   "docker-php-entrypoi..." 44 hours ago  Up 5 seconds  9000/tcp                                              php
056ffafbb286   mysql:5.7                           "docker-entrypoint.s..." 44 hours ago  Up 5 seconds  3306/tcp, 33060/tcp                                    mysql
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose#
```

Acceder a Prometheus

Se accedió a la interfaz web de Prometheus desde el navegador en **http://localhost:9090**



Verificar la recolección de métricas

- En el campo de búsqueda de Prometheus se ejecutó la siguiente consulta: “container_memory_usage_bytes”
- Esto mostró una gráfica con el consumo de memoria de cada contenedor en ejecución.
- La gráfica permitió verificar que Prometheus estaba recolectando correctamente métricas desde cAdvisor.

