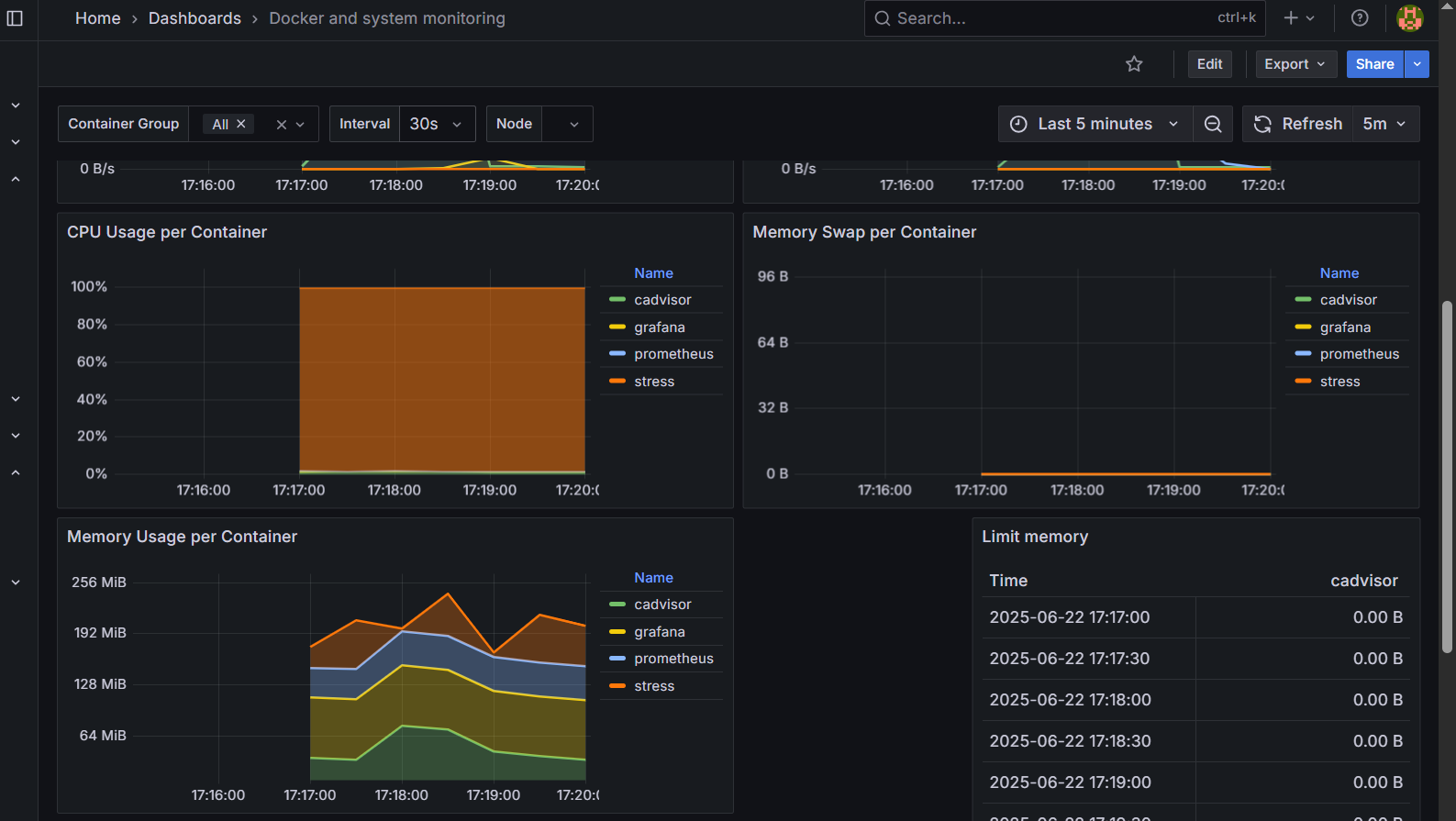
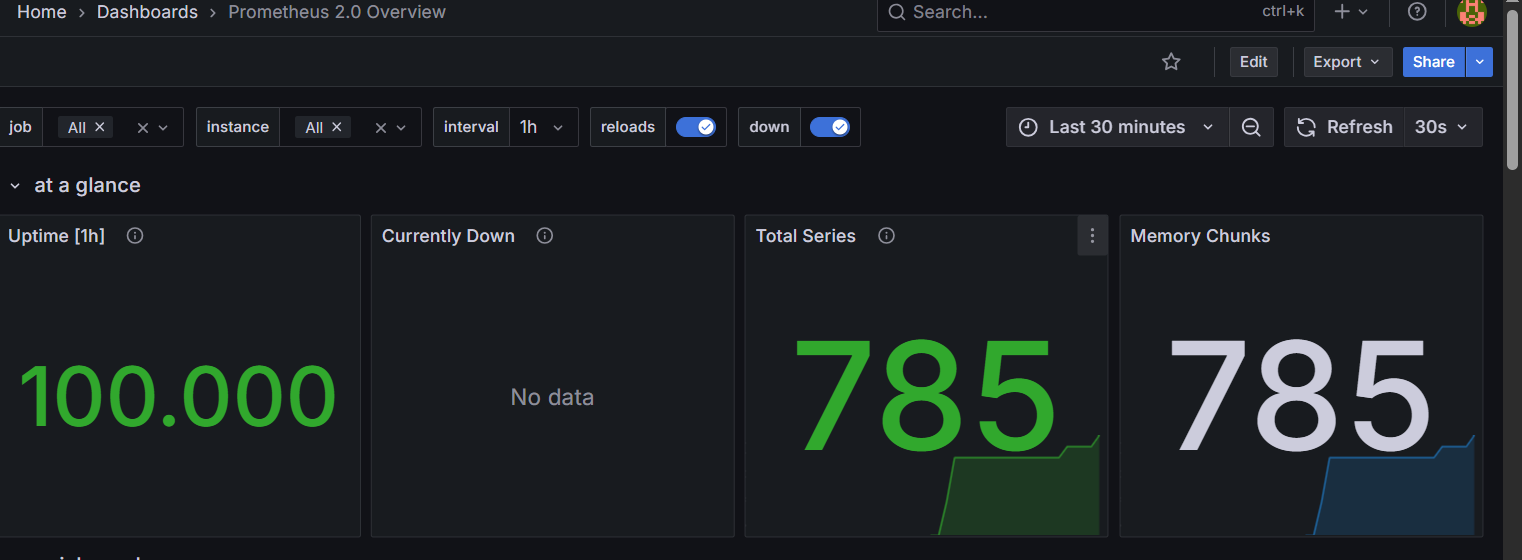
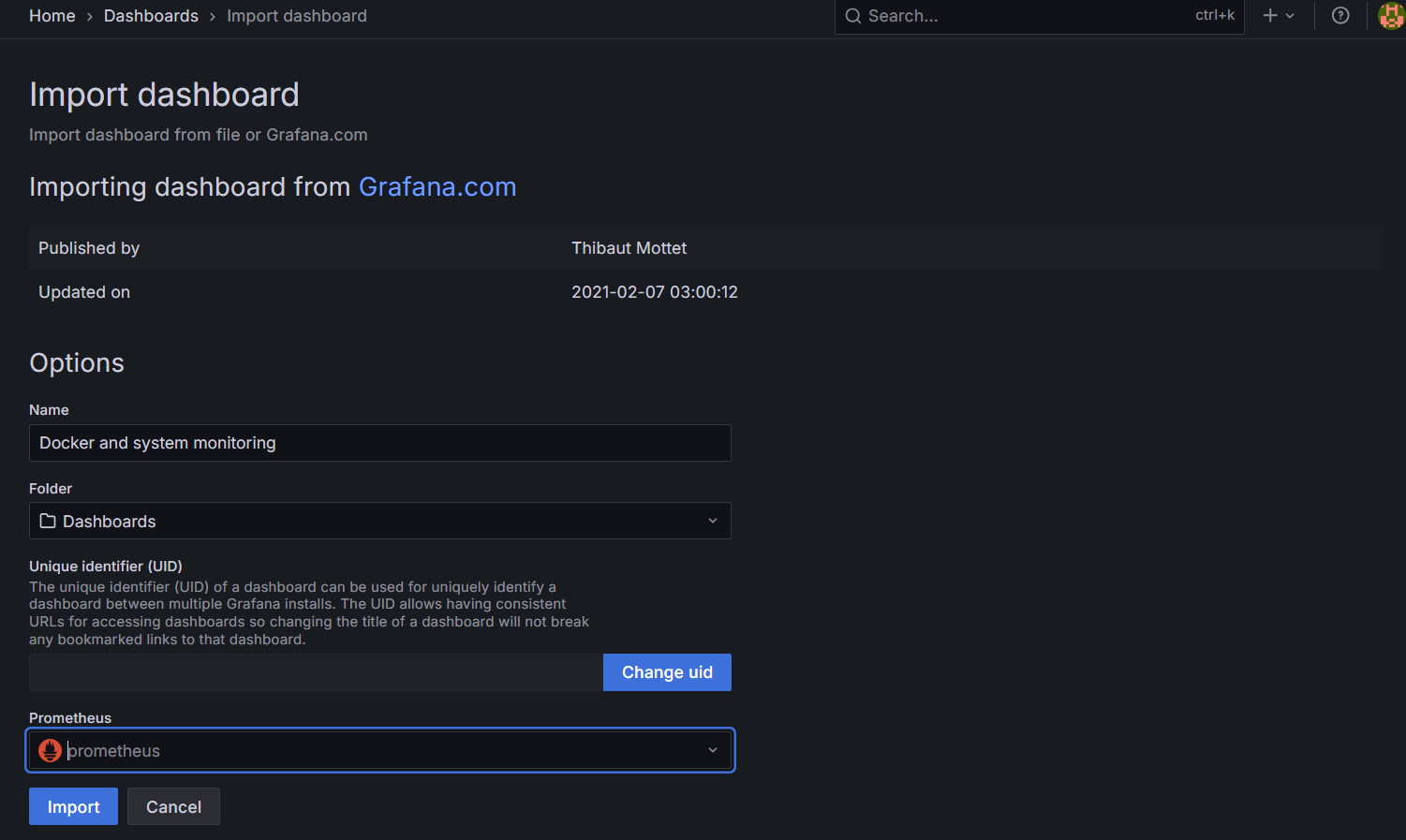
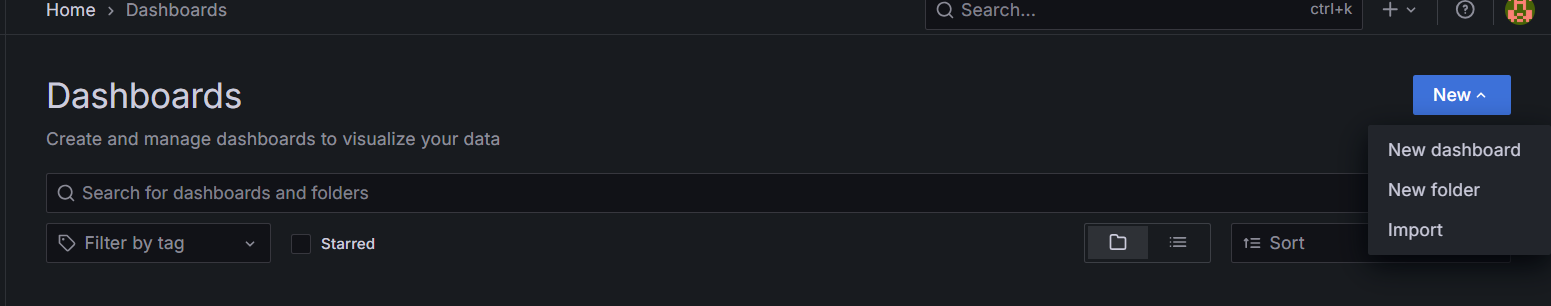
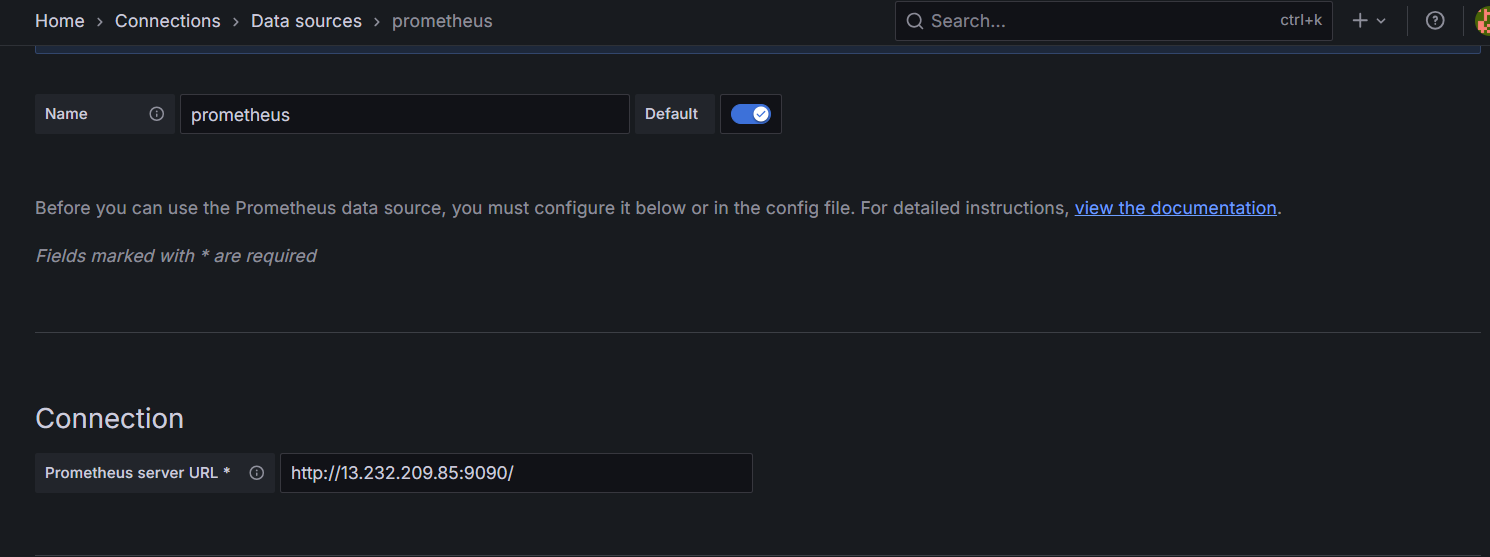
Docker implementation   
  
  
  
  




**Ways to Implement Monitoring for Docker Using Prometheus + Grafana  
1.** Individual Docker Containers (Manual Setup) .  
>Run Prometheus, Grafana, and cAdvisor as separate Docker containers.

>You define the Prometheus config file yourself.

>You run and link containers manually.

What is cAdvisor?

cAdvisor (Container Advisor) is a lightweight monitoring tool created by Google to collect, aggregate, process, and export resource usage and performance data about running containers.

Key Metrics Exposed by cAdvisor:

- CPU Usage

- Memory Usage

- Disk I/O

- Network I/O

- Container uptime and lifecycle stats  
  
 **Prometheus**: Pulls metrics from cAdvisor and stores them in a time-series database

 **cAdvisor**: Collects metrics from Docker containers

 **Grafana**: Reads data from Prometheus and displays it in dashboards  
  
Prerequisites

- EC2 instance with Docker installed

- Security Group with inbound rules open for:

- Port 9090 (Prometheus UI)

- Port 8080 (cAdvisor UI )

- Port 3000 (Grafana UI)

Step-by-Step Setup (With Explanations)

**Step 1: Create Docker Network**

docker network create monitoring

Docker containers are isolated by default. By creating a named network, containers like Prometheus and cAdvisor can discover and communicate with each other using container names instead of IP addresses.

**Step 2: Run cAdvisor as container**

docker run -d \

--name=cadvisor \

--network=monitoring \

-p 8080:8080 \

--volume=/:/rootfs:ro \

--volume=/var/run:/var/run:ro \

--volume=/sys:/sys:ro \

--volume=/var/lib/docker/:/var/lib/docker:ro \

gcr.io/cadvisor/cadvisor:latest

purpose

-name=cadvisor: Easy reference from Prometheus.

-network=monitoring`: So Prometheus can reach it by name.

- -p 8080:8080: Exposes the metrics and UI endpoint.

- The --volume flags mount important system paths so cAdvisor can collect real metrics about Docker containers and the host system.

**Step 3: Create Prometheus Config (`prometheus.yml`)**

mkdir -p ~/Prometheus  
vim Prometheus.yaml

global:

scrape\_interval: 15s

evaluation\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'cadvisor'

static\_configs:

- targets: ['cadvisor:8080']

- scrape\_interval: Defines how often Prometheus collects metrics (default is 1m, we use 15s for near real-time monitoring).

- job\_name: Logical name for the target group.

- targets: Points Prometheus to services to scrape. Since cAdvisor is on the same Docker network, it is reachable by container name (`cadvisor`).

**Step 4: Run Prometheus**

docker run -d \

--name=prometheus \

--network=monitoring \

-p 9090:9090 \

-v ~/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml \

prom/prometheus

- Prometheus reads the `prometheus.yml` configuration to know what targets to scrape.

- Uses the same `monitoring` network to reach `cadvisor`.

- Mounts the configuration file from your host to the container.

**Step 5: Run Grafana**

docker run -d \

--name=grafana \

--network=monitoring \

-p 3000:3000 \

grafana/grafana

- Grafana connects to Prometheus (via `http://prometheus:9090`) to read stored metrics.

- Runs on port 3000 and will be used for visual dashboards.

**Step 6: Access UIs**

| Prometheus | http://<EC2\_PUBLIC\_IP>:9090 | Explore and query metrics |

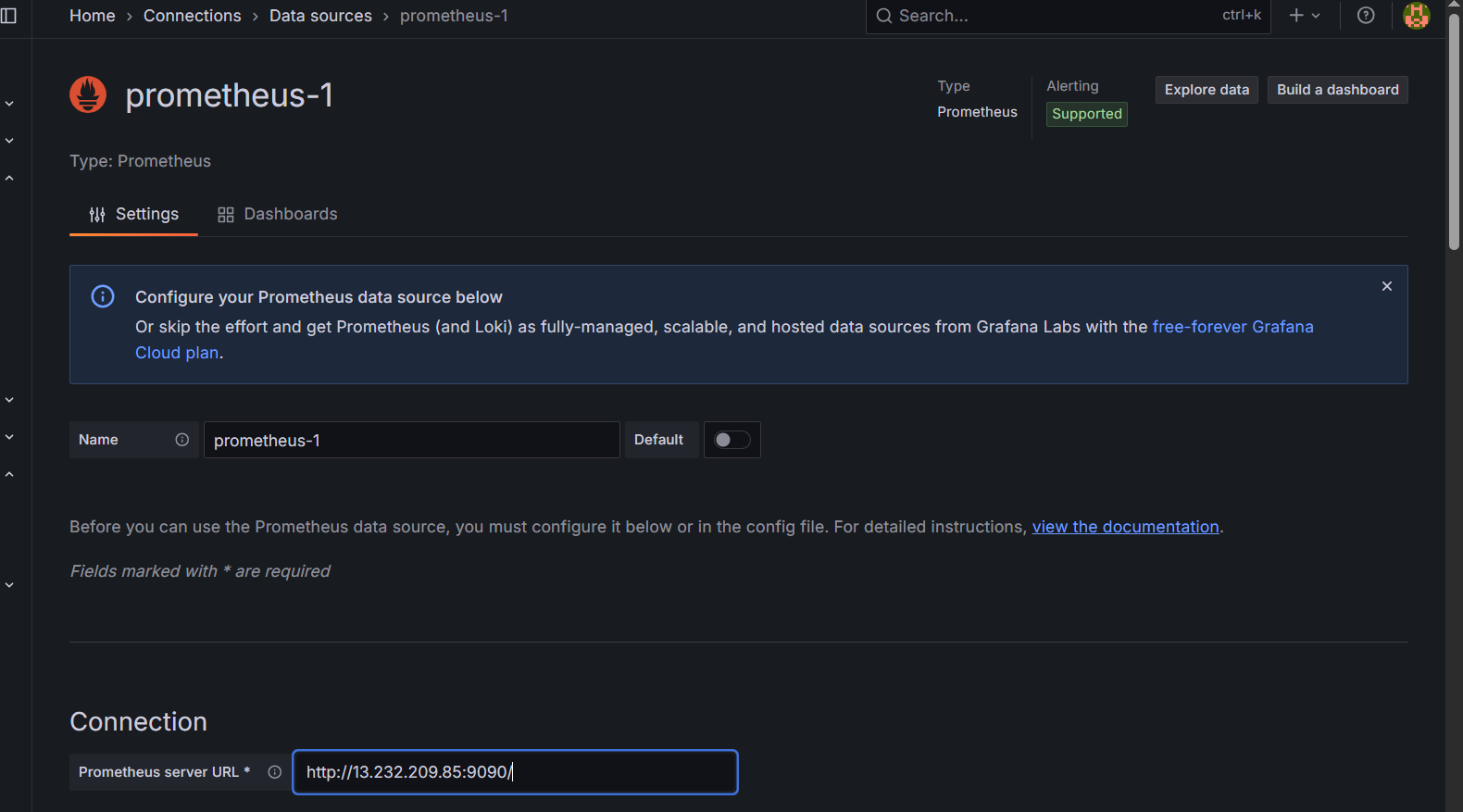
| Grafana | http://<EC2\_PUBLIC\_IP>:3000 | Build and view dashboards |

Step 7: Configure Grafana

1. Log into Grafana

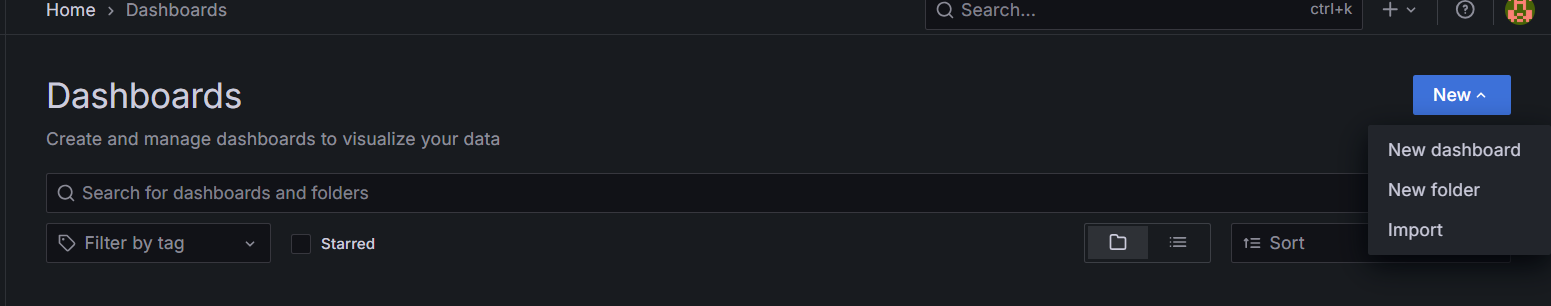
2. admin as password and Go to Configuration → Data Sources

3. Add Prometheus with URL: `http://prometheus:9090`

4. Save and Test  
  
  
Step 7 : Add Data Source   
1. Go to Connections > Data Sources > Add Data Source as Prometheus .  


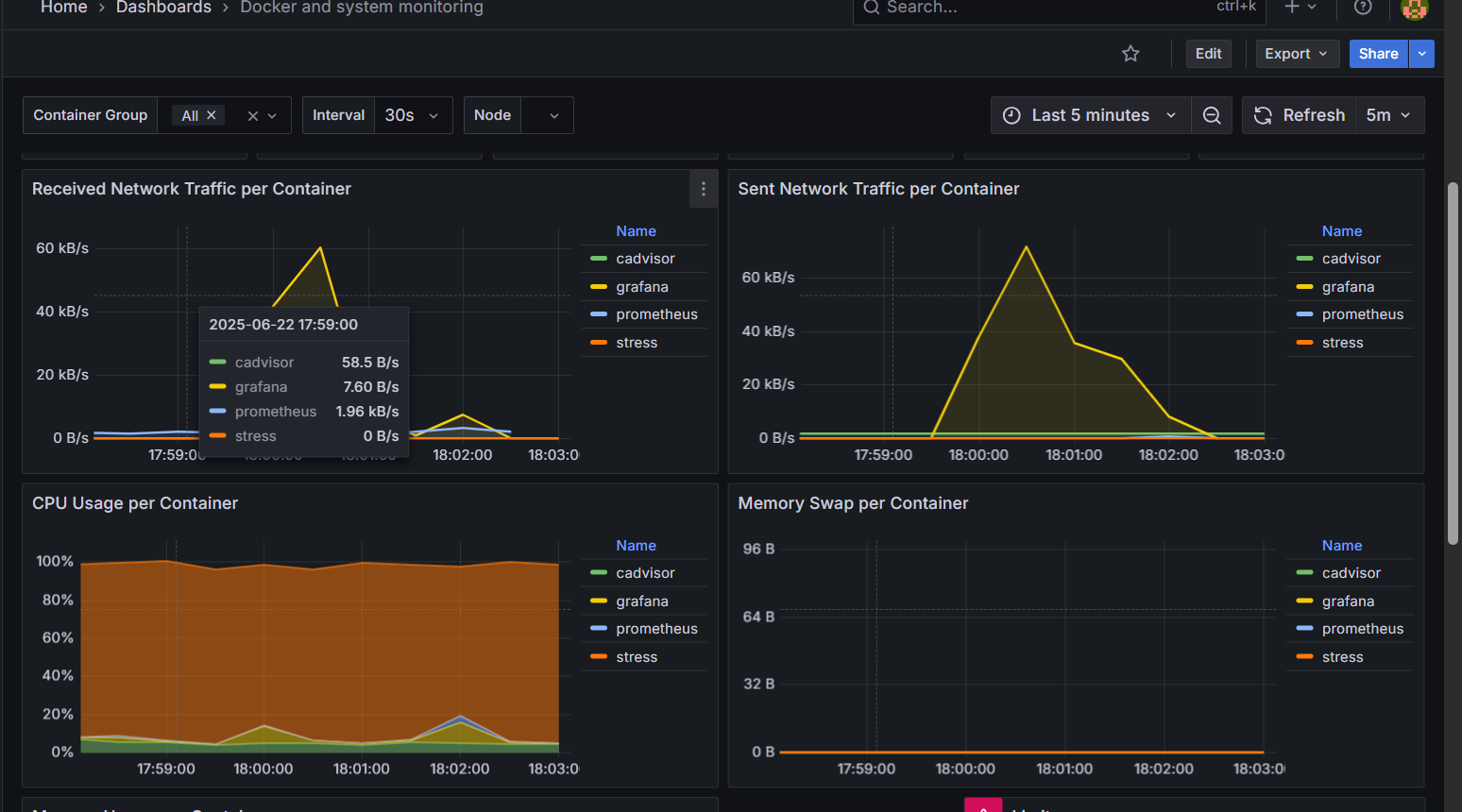
2. Enter the Prometheus and Click on Save & test .

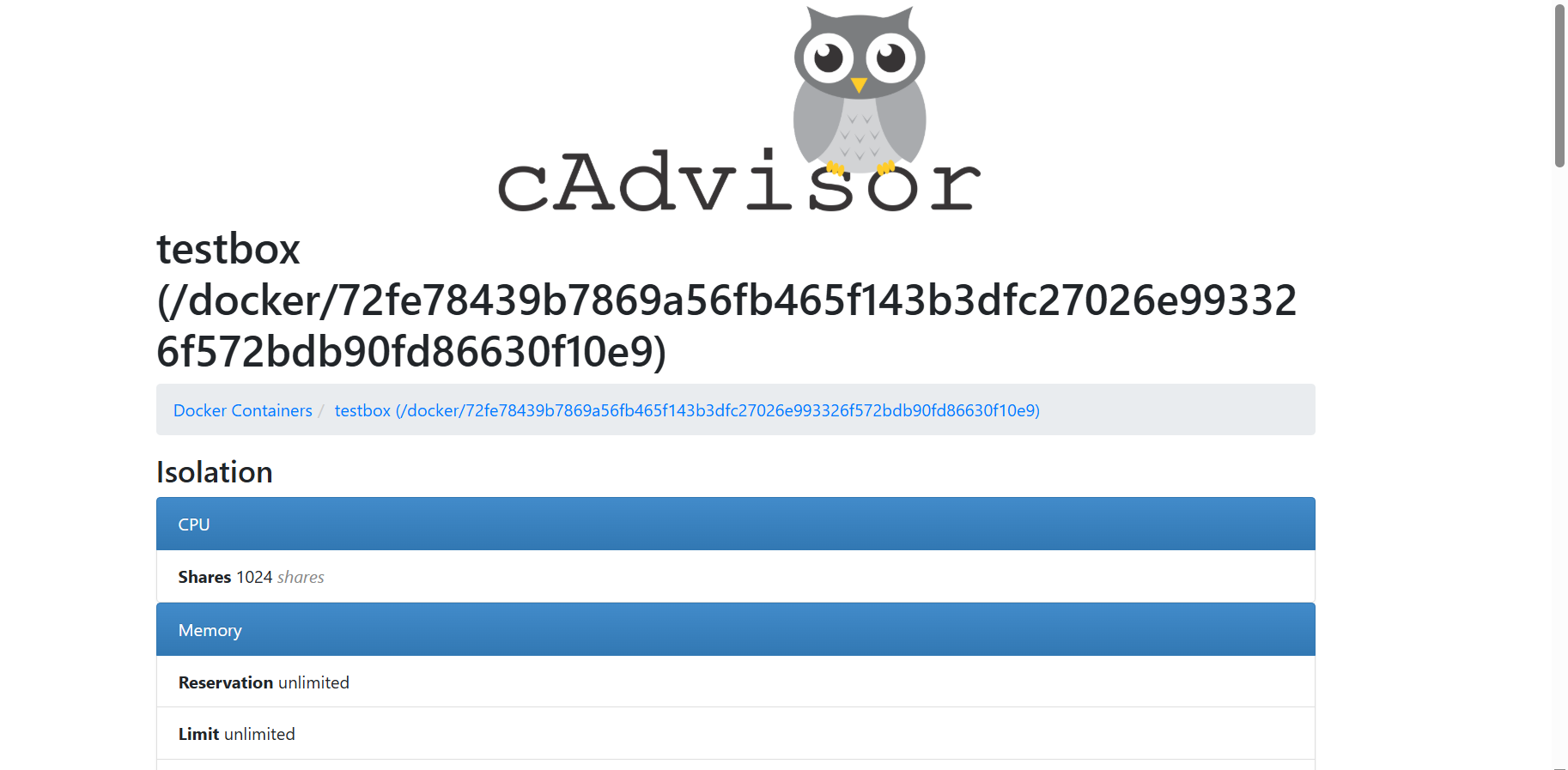
Step 8: Import Docker + cAdvisor Dashboard

1. Go to Dashboard > Import   


2. Use Dashboard ID: 893 for cadvisor

3. Select the Prometheus data source

4. Click Import .  




This community-maintained dashboard shows real-time CPU, memory, network, and filesystem metrics for Docker containers.

Step 9: Run a Sample Container for Metrics

docker run -d --name=stress progrium/stress --cpu 1 --vm 1 --vm-bytes 64M

Simulates CPU and memory load, so that cAdvisor and Grafana display meaningful charts.

You’ve now set up container-level monitoring using:

- cAdvisor (to collect metrics)

- Prometheus(to store metrics)

- Grafana(to visualize metrics)

**2. Docker Compose**Docker Compose uses a docker-compose.yml file to define and run multi-container Docker applications.

Step 1 : Create Project Directory

mkdir name of the directory   
  
Step 2 : Create Prometheus Config File   
> nano name of the dir/prometheus.yml .  
> Paste the below configuration   
global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['prometheus:9090']

- job\_name: 'cadvisor'

static\_configs:

- targets: ['cadvisor:8080']  
  
>Prometheus scrapes itself and cAdvisor every 15s.

>The target names (e.g. prometheus:9090) match container names in the same Docker network.  
  
Step 3 : Create Docker Compose File  
> nano docker-compose.yml   
version: '3'

services:

prometheus:

image: prom/prometheus

container\_name: prometheus

volumes:

- ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml

ports:

- "9090:9090"

networks:

- monitoring

cadvisor:

image: gcr.io/cadvisor/cadvisor:latest

container\_name: cadvisor

ports:

- "8080:8080"

volumes:

- /:/rootfs:ro

- /var/run:/var/run:ro

- /sys:/sys:ro

- /var/lib/docker/:/var/lib/docker:ro

networks:

- monitoring

grafana:

image: grafana/grafana

container\_name: grafana

ports:

- "3000:3000"

networks:

- monitoring

networks:

monitoring:

driver: bridge  
  
> All containers share a virtual network named monitoring.

> Prometheus mounts its config file.

>Grafana runs on port 3000.

>cAdvisor has access to host metrics via mounted volumes.  
  
Step 4: Start the Monitoring Stack  
> docker-compose up –d  
  
Step 5 : Access Interfaces

Prometheus : 9090  
Grafana 3000  
CAdvisor 8080  
  
Step 8 : Configure Grafana Data Source  
> Open Grafana in browser

>Go to Configuration → Data Sources

>Click Add data source

>Choose Prometheus

>Set URL to http://prometheus:9090

>Click Save and Test   
  
Step 6 : Import the Dashboards   
>Import the Dashboards as shown in first manual setup .  
  
**Detection Workflow**

>A new container is started (via Docker or Compose)

>Docker updates its internal data in /var/lib/docker and cgroup structures

>cAdvisor is continuously watching these paths

>cAdvisor detects the new container and begins collecting its metrics

>The new metrics are exposed at http://cadvisor:8080/metrics

>Prometheus scrapes these updated metrics

>Grafana dashboards update accordingly .