

ICT & Infra S3 Supporting Services and Monitoring week 13

Class:	CB01
Student number:	4961854
Student name:	Heiko Morales

Introduction

Introduce basic monitoring using Prometheus so you may apply new skills.

Goals:

- Instrument an application using Prometheus
- Visualize data using Grafana

Please note that there are no strict requirements or recommendations on which platform you should run the components in this assignment. You can choose any OS and use Docker, if you find it more convenient.

Assignment 1. Setting up and Running Prometheus and Grafana

Difficulty: ★★★★★.

1. Visit the Prometheus website and download the most recent version of Prometheus for your OS. You might, for instance, consider running it not on your Windows laptop, but on a Linux EC2 in AWS, or [in a Docker](#)
 - a. These instructions [here](#) that were used in the demo are for Windows only, and they might be a bit outdated and incomplete, compared with the official manual
2. Run Prometheus with default config, explore startup log and then explore the Prometheus UI
 - a. Check Prometheus' own metrics at <http://localhost:9090/metrics>
 - b. Browse to the UI at <http://localhost:9090> and check the `_Status_` pages:
 - Configuration
 - Targets
 - Service Discovery
3. Explore metrics for Linux servers and a batch processing app
 - a. The Batch application used in the demo is located [here](#)
 - b. Follow the [instructions](#), used in the demo and provide the proofs below
4. Run one more instance of the batch processing app on a different host and reconfigure the Prometheus so that it scrapes data from that instance as well. Provide the new configuration file below, as well as updated list of targets from UI.
5. Install Grafana (or sign up for [Grafana Cloud](#)) and get it connected to your Prometheus server
 - a. explore the metrics and create a dashboard showing the metrics from your batch apps and the overall health of the systems you monitor

Provide below the explanations and the screenshots and upload the file to Canvas.

Solution:

We start by installing the two most important containers grafana and prometheus.

```
helko@helko: ~/dockercompose
GNU nano 4.8 docker-compose.yml Modified
version: '3'

volumes:
  prometheus-data:
    driver: local
  grafana-data:
    driver: local

services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    ports:
      - "9090:9090"
    volumes:
      - /etc/prometheus:/etc/prometheus
      - prometheus-data:/prometheus
    restart: unless-stopped
    command:
      - "--config.file=/etc/prometheus/prometheus.yml"

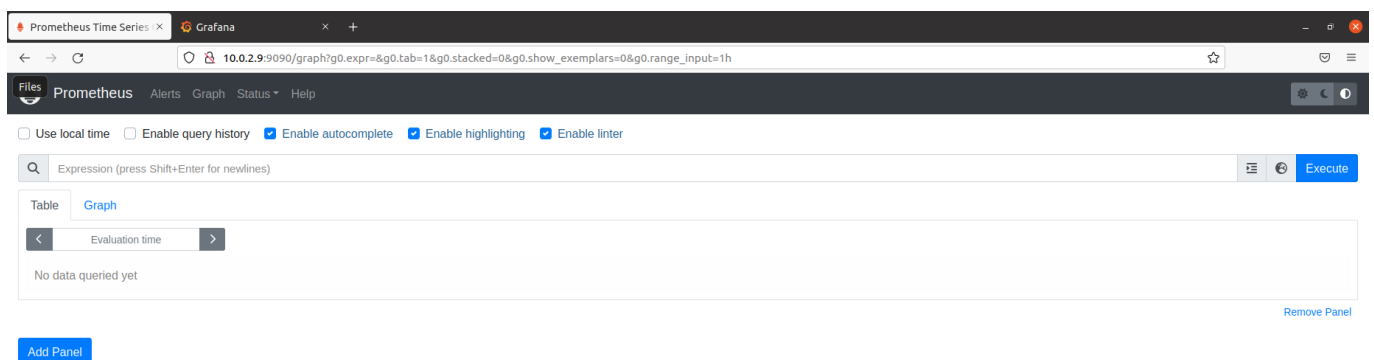
  grafana:
    image: grafana/grafana-oss:latest
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana-data:/var/lib/grafana
    restart: unless-stopped
```

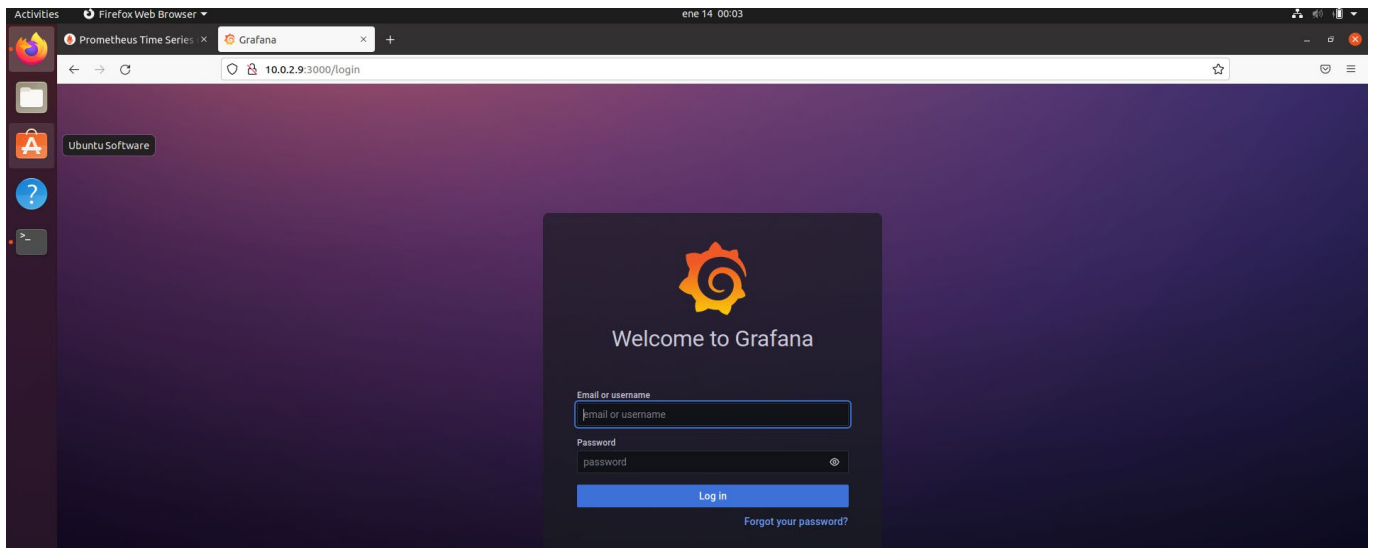
We install these in docker and see if they have been created

```
helko@helko:~/dockercompose$ sudo docker ps -a
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
6c85648bf08b	prom/prometheus:latest		"/bin/prometheus --c..."	19 seconds ago	Up 18 seconds	0.0.0.0:9090->9090/tcp,
:::9090->9090/tcp		prometheus				
b8a5de12cfbe	grafana/grafana-oss:latest		"/run.sh"	About a minute ago	Up About a minute	0.0.0.0:3000->3000/tcp,
:::3000->3000/tcp		grafana				

once created we can see if the web pages are deployed





Later we will go to the targets and we will see how only the localhost is monitored

Targets

AllUnhealthyCollapse All

Q

Filter by endpoint or labels

prometheus (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"job="prometheus"	1.859s ago	8.031ms	

We install the node exporter and cadvisor systems in docker compose. the first will provide us with data about the server hardware and the second with data about the docker containers.

```
node_exporter:
  image: quay.io/prometheus/node-exporter:latest
  container_name: node_exporter
  command:
    - '--path.rootfs=/host'
  pid: host
  restart: unless-stopped
  volumes:
    - '/:/host:ro,rslave'

cadvisor:
  image: google/cadvisor:latest
  container_name: cadvisor
  volumes:
    - '/:/rootfs:ro'
    - '/var/run:/var/run:ro'
    - '/sys:/sys:ro'
    - '/var/lib/docker:/var/lib/docker:ro'
    - '/dev/disk/:/dev/disk:ro'
  devices:
    - /dev/kmsg
```

Next, in the Prometheus configuration, let's add the two new systems just implemented.

```
heiko@heiko: ~/dockercompose
GNU nano 4.8 /etc/prometheus/prometheus.yml
global:
  scrape_interval:     15s # By default, scrape targets every 15 seconds.

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  # external_labels:
  #   monitor: 'codelab-monitor'

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: 'prometheus'
    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']

  # Example job for node_exporter
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['node_exporter:9100']

  # Example job for cadvisor
  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']
```

and as we see in the targets we are already monitoring more endpoints

Targets

AllUnhealthyCollapse All

Filter by endpoint or labels

cadvisor (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://cadvisor:8080/metrics	UP	instance="cadvisor:8080" job="cadvisor"	33.680s ago	367.288ms	

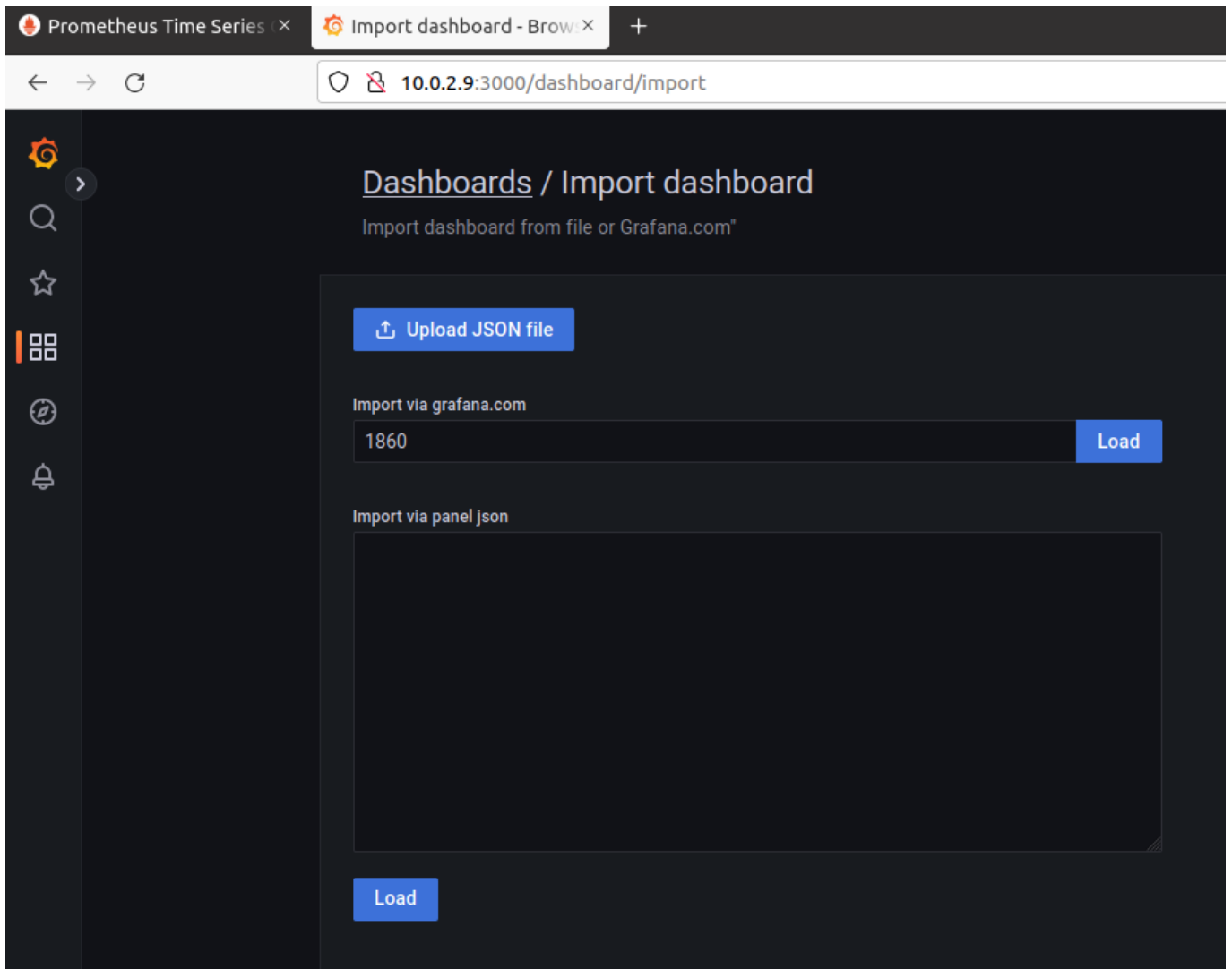
node_exporter (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://node_exporter:9100/metrics	UP	instance="node_exporter:9100" job="node_exporter"	32.431s ago	63.886ms	

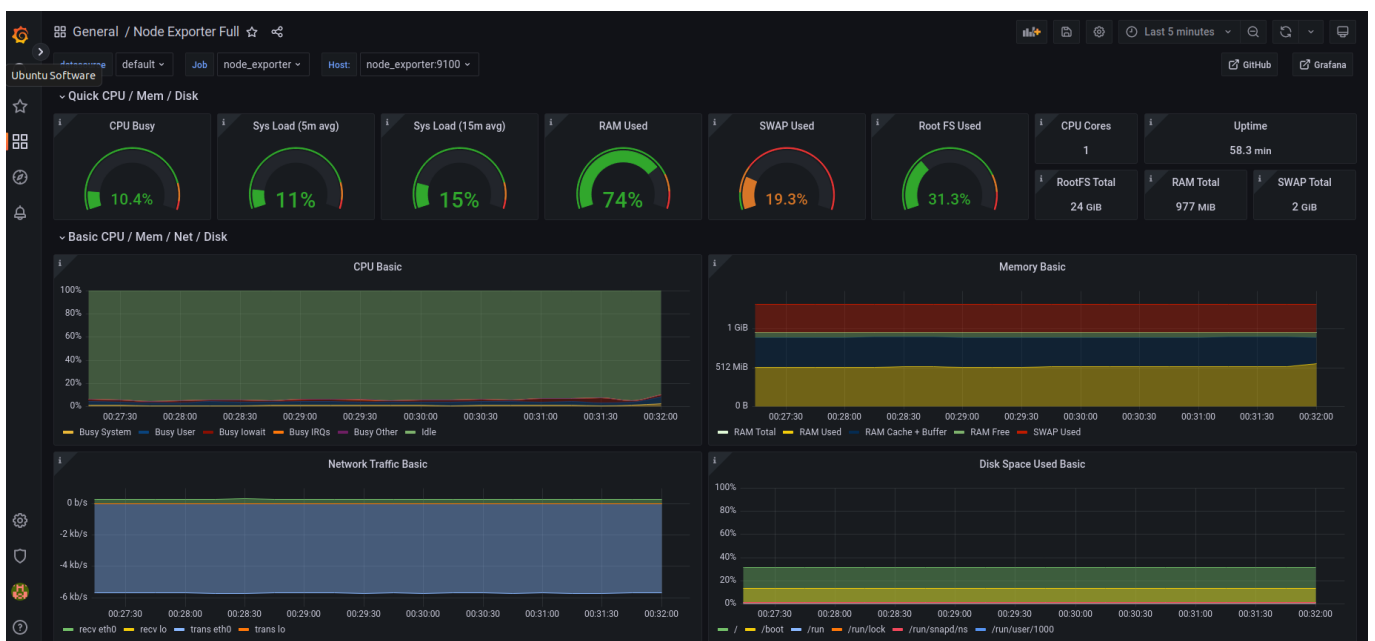
prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	25.412s ago	4.348ms	

now to visualize the data in a more visual way we import the node exporter dashboard



And we are already monitoring all the hardware of the device.



now to monitor the containers you just have to import your dashboard

Dashboards / Import dashboard

Import dashboard from file or Grafana.com*

Importing dashboard from Grafana.com

Published by

kokorinav

Updated on

2021-04-21 14:31:33

Options

Name

Cadvisor exporter

Folder

General

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used to uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

pMed7m0Mz

Change uid

Prometheus

Prometheus

Import

Cancel

and we already have available all the monitoring of the exposed services / containers.

