



Software Defined System (SDS) Report

Kinza Ghaffar
2309793

Task 1 - 10pts

Task 3.1 - 5-10pts

Task 3.2 - 10pts

Task 3.2 - 10pts

Task 1: Sending Formula 1 car data to cloud with Eclipse Kuksa

f1demo2val

First, I ran this command to build the image on Docker.

- `docker build -t docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest -f Dockerfile .`

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

F:\MS\2nd period\SDS\project\task1\f1demo2val>docker build -t docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest -f Dockerfile .
[+] Building 7.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> -> transferring dockerfile: 588B
=> [internal] load .dockerignore
=> -> transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.9.0-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/python:3.9.0-slim@sha256:de8d4a338fb815509de2046cea4ff48959c84aabc1b65cb41c5e2da1ac599b04
=> [internal] load build context
=> -> transferring context: 3.36kB
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY . /app
=> CACHED [4/4] RUN pip install --trusted-host pypi.python.org -r requirements.txt
=> exporting to image
=> -> exporting layers
=> -> writing image sha256:eac28ea6dd686ee0fd6092980bd8f280fa90a52e322a5e79a9042182d7bb12c
=> -> naming to docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest

What's Next?
  View a summary of image vulnerabilities and recommendations -> docker scout quickview

F:\MS\2nd period\SDS\project\task1\f1demo2val>
```

After, I executed this command to push the image on rahti registry.

- `docker push docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest`

```
F:\MS\2nd period\SDS\project\task1\f1demo2val>docker push docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest
The push refers to repository [docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val]
42bbbb1e5c27: Preparing
b2a68b8f5816: Preparing
9e367eb0c856: Preparing
b25b816984b3: Preparing
7f87da26703a: Preparing
889e9591fc68: Waiting
0d92fd747771: Waiting
f5600c6330da: Waiting
unauthorized: authentication required

F:\MS\2nd period\SDS\project\task1\f1demo2val>docker login -p Q2uRehmQk7qlAMUjq5i0UaHVM5858pABRh_WaXGxHE -u unused docker-registry.rahti.csc.fi
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded

F:\MS\2nd period\SDS\project\task1\f1demo2val>docker push docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val:latest
The push refers to repository [docker-registry.rahti.csc.fi/kinzaghaffar/f1demo2val]
42bbbb1e5c27: Layer already exists
b2a68b8f5816: Layer already exists
9e367eb0c856: Layer already exists
b25b816984b3: Layer already exists
7f87da26703a: Layer already exists
889e9591fc68: Layer already exists
0d92fd747771: Layer already exists
f5600c6330da: Layer already exists
latest: digest: sha256:238ed1ca0411bc05df9ffbb0a4fc752d926c904382c2a9743093799ca197611a size: 2000
```

val2mqtt

First, I ran this command to build the image on Docker.

- `docker build -t docker-registry.rahti.csc.fi/kinzaghaffar/val2mqtt:latest -f Dockerfile .`

```
F:\MS\2nd period\SDS\project\task1\fidemo2val>docker push docker-registry.rahti.csc.fi/kinzaghaffar/fidemo2val:latest
The push refers to repository [docker-registry.rahti.csc.fi/kinzaghaffar/fidemo2val]
42bbbb1e5c27: Layer already exists
b2a68b8f5816: Layer already exists
9e367eb0c856: Layer already exists
b25b816984b3: Layer already exists
7f87da26703a: Layer already exists
889e9591fc68: Layer already exists
0d92fd747771: Layer already exists
f5600c6330da: Layer already exists
[+] Building 1.25 (9/9) FINISHED
-> [internal] load build definition from Dockerfile
-> -> transferring dockerfile: 588B
-> [internal] load .dockerignore
-> -> transferring context: 2B
-> [internal] load metadata for docker.io/library/python:3.9.0-slim
-> [1/4] FROM docker.io/library/python:3.9.0-slim@sha256:de8d4a338fb815509de2046cea4ff48959c84aabc1b65cb41c5e2da1ac599b04
-> [internal] load build context
-> -> transferring context: 3.36kB
-> CACHED [2/4] WORKDIR /app
-> CACHED [3/4] COPY . /app
-> CACHED [4/4] RUN pip install --trusted-host pypi.python.org -r requirements.txt
-> exporting to image
-> -> exporting layers
-> -> writing image sha256:eac28ea6dd686ee00fd6092980bd8f280fa90a52e322a5e79a9042182d7bb12c
-> -> naming to docker-registry.rahti.csc.fi/kinzaghaffar/val2mqtt:latest

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
```

After, I executed this command to push the image on rahti registry.

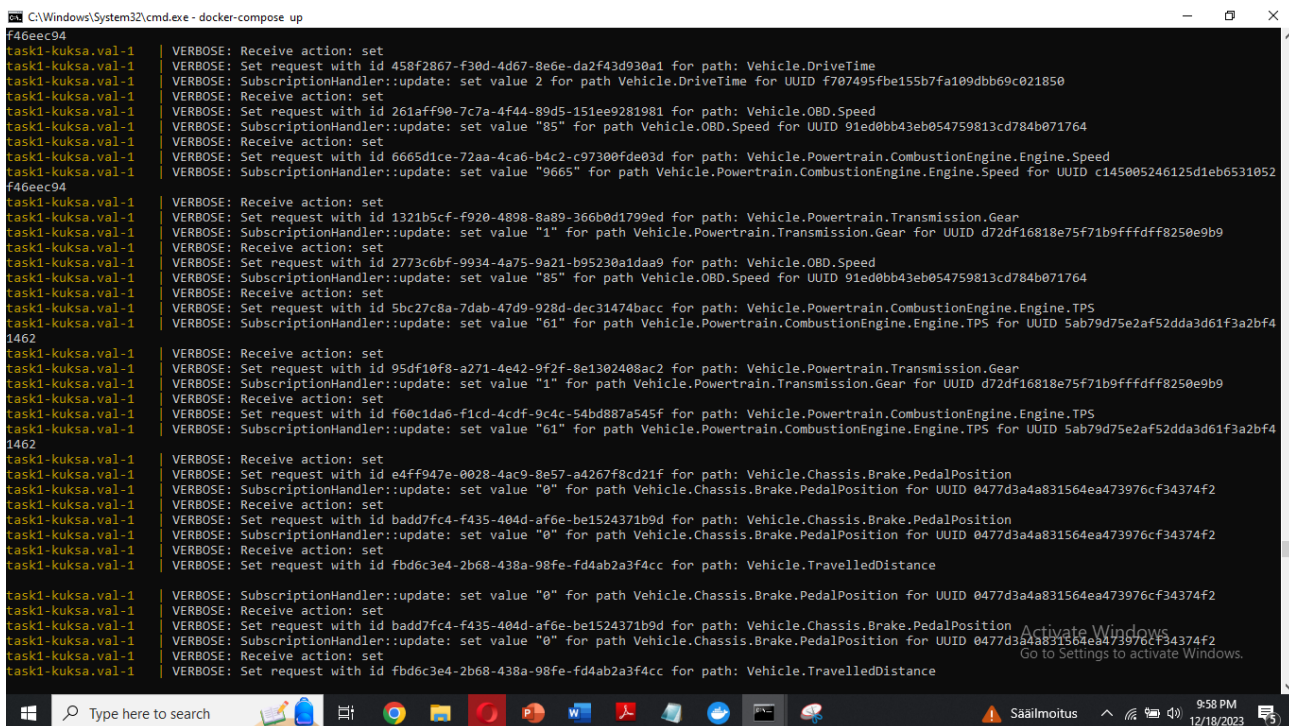
- `docker push docker-registry.rahti.csc.fi/kinzaghaffar/val2mqtt:latest`

```
F:\MS\2nd period\SDS\project\task1\fidemo2val>docker push docker-registry.rahti.csc.fi/kinzaghaffar/val2mqtt:latest
The push refers to repository [docker-registry.rahti.csc.fi/kinzaghaffar/val2mqtt]
42bbbb1e5c27: Layer already exists
b2a68b8f5816: Layer already exists
9e367eb0c856: Layer already exists
b25b816984b3: Layer already exists
7f87da26703a: Layer already exists
889e9591fc68: Layer already exists
0d92fd747771: Layer already exists
f5600c6330da: Layer already exists
latest: digest: sha256:238ed1c1a0411bc05df9ffb0a4fc752d926c904382c2a9743093799ca197611a size: 2000

F:\MS\2nd period\SDS\project\task1\fidemo2val>
```

Then I will run command to compose up the docker container

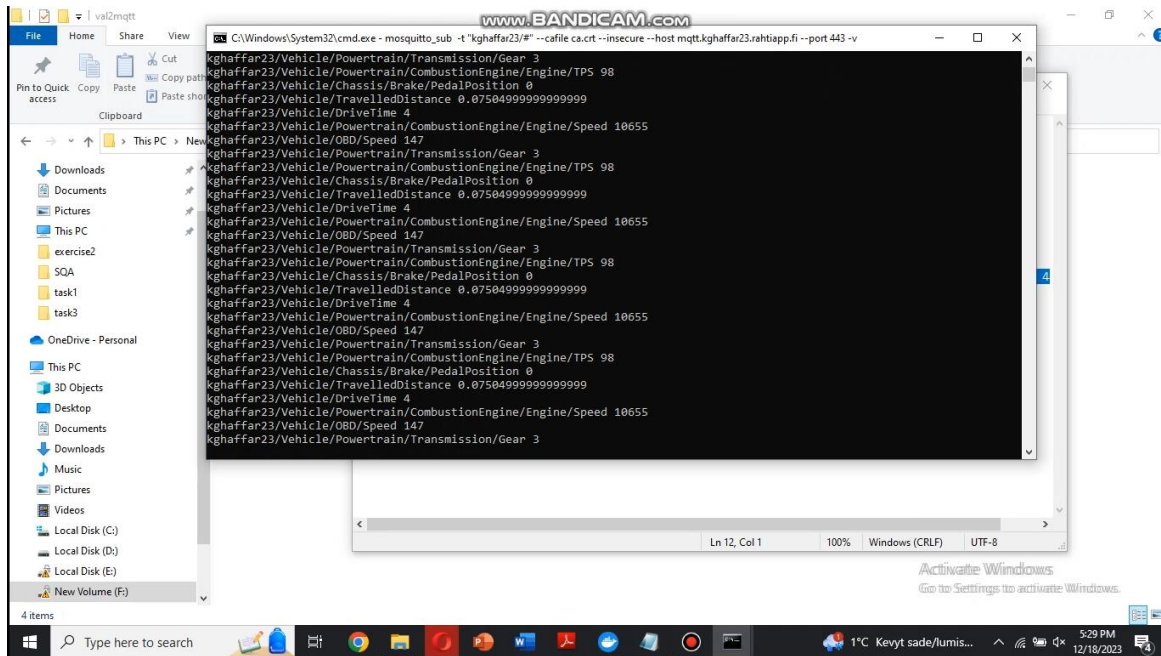
- `docker-compose up`



```
C:\Windows\System32\cmd.exe - docker-compose up
f46ec94
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 458f2867-f30d-4d67-8e6e-da2f43d930a1 for path: Vehicle.DriveTime
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "1" for path Vehicle.Powertrain.Transmission.Gear for UUID f707495f8155b7fa109dbb69c021850
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 261aff90-7c7a-4f44-89d5-151ee9281981 for path: Vehicle.OBD.Speed
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "85" for path Vehicle.OBD.Speed for UUID 91ed0bb43eb054759813cd784b071764
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 6665d1ce-72aa-4ca6-b4c2-c97300fde03d for path: Vehicle.Powertrain.CombustionEngine.Engine.Speed
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "9665" for path Vehicle.Powertrain.CombustionEngine.Engine.Speed for UUID c145005246125d1eb6531052
f46ec94
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 1321b5cf-f920-4898-8a89-366b0d1799ed for path: Vehicle.Powertrain.Transmission.Gear
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "1" for path Vehicle.Powertrain.Transmission.Gear for UUID d72df16818e75f71b9ffffdf8250e9b9
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 2773c6bf-9934-4a75-9a21-b95230a1daa9 for path: Vehicle.OBD.Speed
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "85" for path Vehicle.OBD.Speed for UUID 91ed0bb43eb054759813cd784b071764
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 5bc27c8a-7dab-47d9-928d-dec31474bacc for path: Vehicle.Powertrain.CombustionEngine.Engine.TPS
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "61" for path Vehicle.Powertrain.CombustionEngine.Engine.TPS for UUID 5ab79d75e2af52dda3d61f3a2bf4
1462
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id 95df10f8-a271-4e42-9f2f-8e1302408ac2 for path: Vehicle.Powertrain.Transmission.Gear
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "1" for path Vehicle.Powertrain.Transmission.Gear for UUID d72df16818e75f71b9ffffdf8250e9b9
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id f60c1da6-f1cd-4cdf-9c4c-54bd887a545f for path: Vehicle.Powertrain.CombustionEngine.Engine.TPS
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "61" for path Vehicle.Powertrain.CombustionEngine.Engine.TPS for UUID 5ab79d75e2af52dda3d61f3a2bf4
1462
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id e4ff947e-0028-4ac9-8e57-a4267f8cd21f for path: Vehicle.Chassis.Brake.PedalPosition
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "0" for path Vehicle.Chassis.Brake.PedalPosition for UUID 0477d3a4a831564ea473976cf34374f2
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id badd7fc4-f435-404d-af6e-be1524371b9d for path: Vehicle.Chassis.Brake.PedalPosition
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "0" for path Vehicle.Chassis.Brake.PedalPosition for UUID 0477d3a4a831564ea473976cf34374f2
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id fbd6c3e4-2b68-438a-98fe-fd4ab2a3f4cc for path: Vehicle.TravelledDistance
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "0" for path Vehicle.Chassis.Brake.PedalPosition for UUID 0477d3a4a831564ea473976cf34374f2
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id badd7fc4-f435-404d-af6e-be1524371b9d for path: Vehicle.Chassis.Brake.PedalPosition
task1-kuksa.val-1 | VERBOSE: SubscriptionHandler::update: set value "0" for path Vehicle.Chassis.Brake.PedalPosition for UUID 0477d3a4a831564ea473976cf34374f2
task1-kuksa.val-1 | VERBOSE: Receive action: set
task1-kuksa.val-1 | VERBOSE: Set request with id fbd6c3e4-2b68-438a-98fe-fd4ab2a3f4cc for path: Vehicle.TravelledDistance
```

And then the last command to pass information between them is

- `mosquitto_sub -t "kghaffar23/#" --cafile ca.crt --insecure --host mqtt.kghaffar23.rahtiapp.fi --port 443 -v`



Task 3: Monitoring cluster with Prometheus and Grafana

Task 3.1: Install Prometheus+Grafana

A: Install Prometheus+Grafana template by Rahi

Prometheus

Time-series data collection is a strong suit for Prometheus, an open-source monitoring and alerting toolset, which keeps track of system metrics and application performance.

Grafana

Grafana is an open-source platform for analytics and monitoring that provides an easy-to-use interface for building dashboards that can be customized. It easily connects with Prometheus and other data sources, giving teams access to strong visualization capabilities for information gathering and efficient system performance and status reporting. In contemporary IT environments, Prometheus and Grafana work together to provide broad observability.

Execution

First, I executed a command to deploy the “prometheus-grafana” downloaded template and got the link to both dashboards along with their login credentials.

- `oc apply -f prometheus-grafana.yaml`


```
C:\Windows\System32\cmd.exe

D:\check\test\kinza\ptask3\task3>
D:\check\test\kinza\ptask3\task3>oc apply -f prometheus-grafana.yaml
template.openshift.io/prometheus-grafana created

D:\check\test\kinza\ptask3\task3>
D:\check\test\kinza\ptask3\task3>oc apply -f mosquito_exporter.yaml
deployment.apps/mosquito-exporter created
service/mosquito-exporter-service created

D:\check\test\kinza\ptask3\task3>oc apply -f node-exporter.yaml
error: the path "node-exporter.yaml" does not exist

D:\check\test\kinza\ptask3\task3>oc apply -f node_exporter.yaml
deployment.apps/node-exporter created
service/node-exporter-service created

D:\check\test\kinza\ptask3\task3>

C:\Windows\System32\cmd.exe

You should soon be able to access Prometheus and Grafana through the project overview page.

Your Grafana admin credentials are:
  username: admin
  password: OcSR5f1S6h7Cokvt5TJklyykFMUxyX

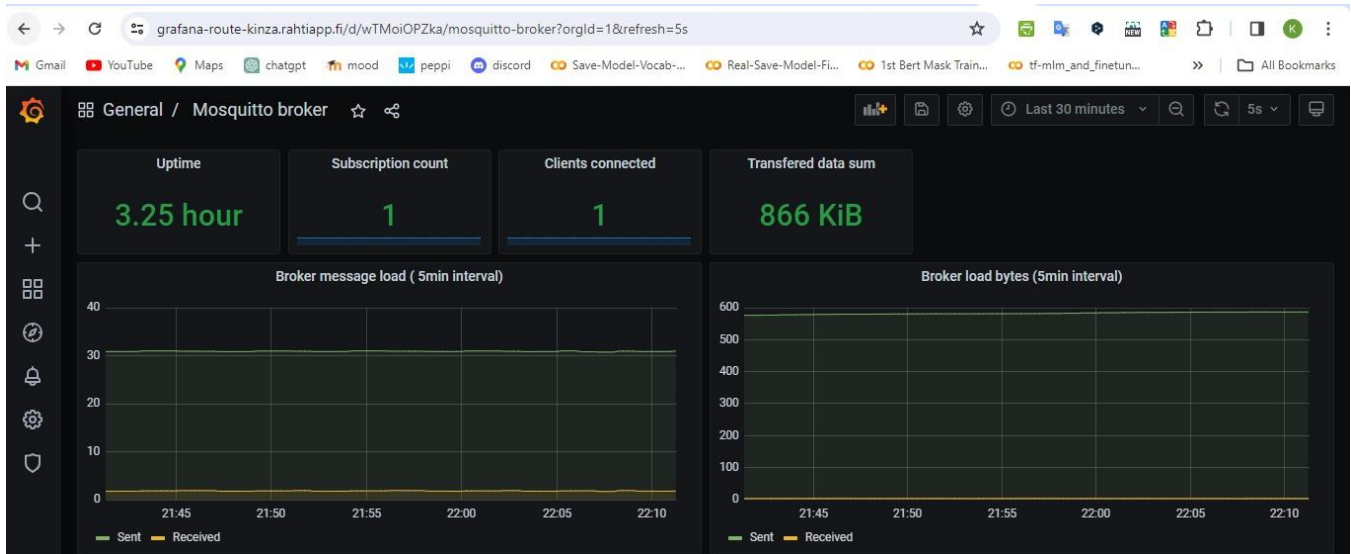
Your Prometheus basic auth credentials are:
  username: admin
  password: Q0he3lhan3Xdk7nnWvVd3odMTpRJCv

* With parameters:
  * Namespace=kinza
  * Prometheus image=prom/prometheus:v2.27.1
  * Storage retention time=15d
  * Prometheus volume size=10Gi
  * Prometheus memory limit=4G
  * Prometheus requested memory=4G
  * Prometheus basic auth username=admin
  * Prometheus basic auth password=Q0he3lhan3Xdk7nnWvVd3odMTpRJCv # generated
  * Grafana image=grafana/grafana:7.5.7
  * Grafana admin username=admin
  * Grafana admin password=OcSR5f1S6h7Cokvt5TJklyykFMUxyX # generated
  * Grafana volume size=100Mi
  * Grafana memory limit=1Gi
  * Grafana requested memory=512Mi
```

Task 3.2: Monitoring MQTT broker

Dashboard

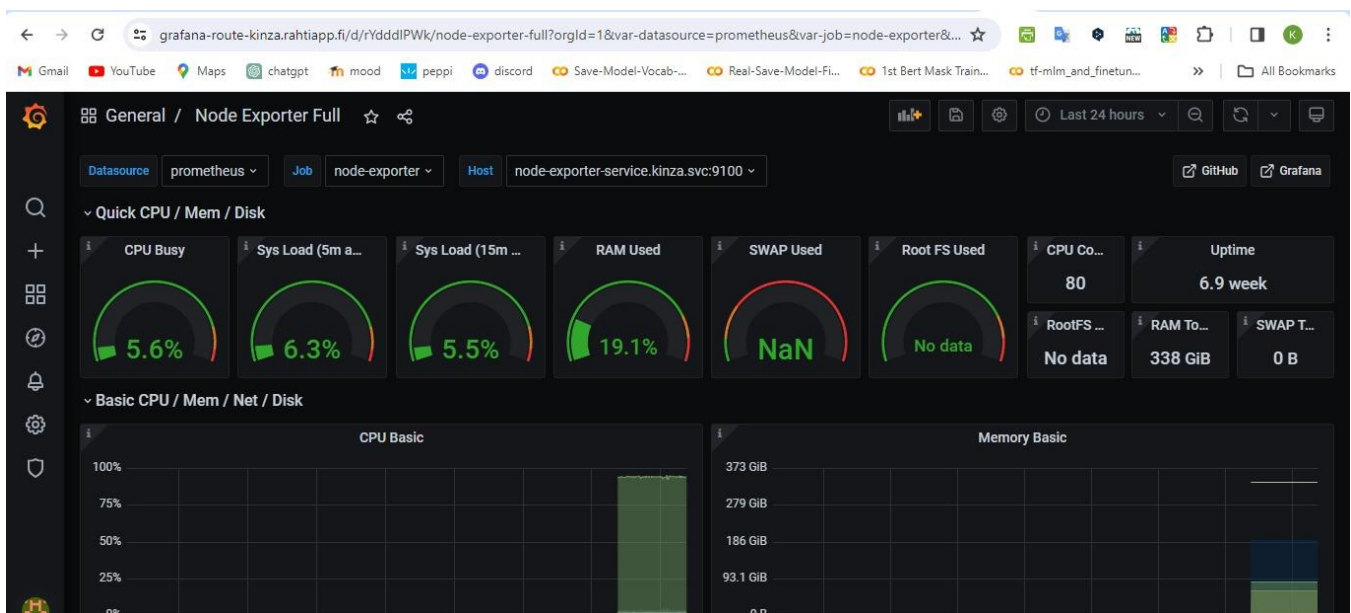
- **Username:** admin
- **Password:** OcSR5f1S6h7Cokvt5TJklyykFMUxyX
- Link to the dashboard: <https://grafana-route-kinza.rahtiapp.fi/d/wTMoiOPZka/mosquitto-broker?orgId=1&refresh=5s>



Task 3.3: Monitoring cluster with node-exporter

Dashboard

- **Username:** admin
- **Password:** Q0he3lhan3Xdk7nnWvVd3odMTpRJcV
- Link to the dashboard: <https://grafana-route-kinza.rahtiapp.fi/d/rYdddIPWk/node-exporter-full?orgId=1&var-datasource=prometheus&var-job=node-exporter&var-node=node-exporter-service.kinza.svc:9100&var-diskdevices=%5Ba-z%5D%2B%7Cnvme%5B0-9%5D%2Bn%5B0-9%5D%2B%7Cmmbblk%5B0-9%5D%2B>



C:\Windows\System32\cmd.exe

```
D:\check\test\kinza\ptask3\task3>oc new-app -f ./prometheus-grafana.yaml -p NAMESPACE=kinza  
--> Deploying template "kinza/prometheus-grafana" for "./prometheus-grafana.yaml" to project kinza
```

Prometheus + Grafana -----

Deploys Prometheus and Grafana for monitoring pods running in the same namespace. For more information on the use of this template, see <https://github.com/CSCfi/grafana-prometheus-template>.

Prometheus is configured by default to scrape metrics from applications running in pods in the same namespace. Grafana is configured by default with Prometheus as a datasource.

You need to add the following annotations to the pods you want to monitor:

- * prometheus.io/scrape: 'true'
- * prometheus.io/path: <path> if you need to scrape metrics from a path other than '/metrics'
- * prometheus.io/port: <port> if you need to use a port other than the pod's declared ports

You should soon be able to access Prometheus and Grafana through the project overview page.