Learning diary

Tasks:

- *Setup monitoring with Prometheus and Grafana*
- *Collect metrics from a Python WebApp (Flask)*
- *Visualize metrics in Grafana*

Setting up the project

Where are services running? - They are running locally.

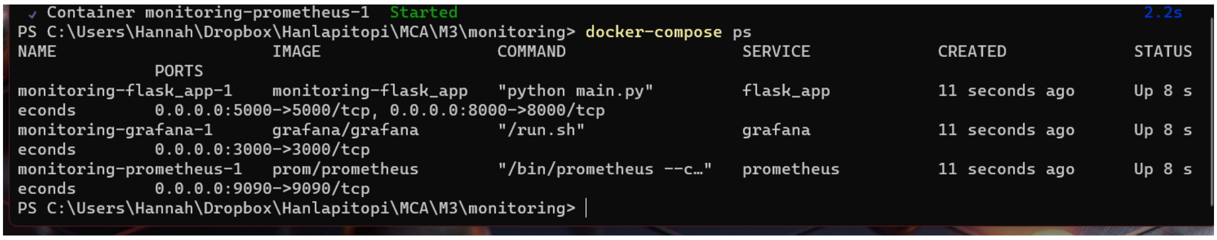
What will be the order of the services and which service will start first? Why? – Services: Prometheus, Grafana, webapp, since the Prometheus

Configs/Commands learnt: Yml includes config files, which reaches from the local environment to the docker container. In this file the metrics and pull requests are defined, which are portrayed in the dashboards.

Under environment: setting up variables for the container.

Prometheus yaml file pulls data from app service.

Setting up Python webapp



```
 ✓ Container monitoring-prometheus-1  Started                                                            2.2s
PS C:\Users\Hannah\Dropbox\Hanlapitopi\MCA\M3\monitoring> docker-compose ps
NAME                      IMAGE                  COMMAND                   SERVICE             CREATED            STATUS
              PORTS
monitoring-flask_app-1    monitoring-flask_app   "python main.py"          flask_app           11 seconds ago     Up 8 s
econds        0.0.0.0:5000->5000/tcp, 0.0.0.0:8000->8000/tcp
monitoring-grafana-1      grafana/grafana        "/run.sh"                 grafana             11 seconds ago     Up 8 s
econds        0.0.0.0:3000->3000/tcp
monitoring-prometheus-1   prom/prometheus        "/bin/prometheus --c…"    prometheus          11 seconds ago     Up 8 s
econds        0.0.0.0:9090->9090/tcp
PS C:\Users\Hannah\Dropbox\Hanlapitopi\MCA\M3\monitoring> 
```

What is the base image? – python:3.9-slim - Starting point for the container workflow, which creates all container images.

What is the working directory? – /app – WORKDIR sets that directory for the instructions followed.

What is the entry point command? – CDM [“python”, “main.py”] – used to configure the executable scripts after initiation.

Configs/Commands learnt: Flask webapp endpoint through port 5000. To pull data: using Prometheus-client library with two metrics: request processing time and number of requests. Collection for Prometheus through port 8000. Grafana: 3000, webapp: 5000, Prometheus: 9090

Setting up Grafana:

How does the main.py file connect to the graphs? – main.py uses Prometheus_client and Flask package: to listen to a port defined to get data and to enable the framework building of the webapp.

Through the port Prometheus collects the webapp metrics and Grafana helps visualizing them.

Main.py defines the metrics: Request_Time counts the time needed for connecting/processing and Request_count shows the number of connections to localhost:5000.