

Prometheus Assignment

- Export the metrics (like request per second, memory usage, cpu usage etc) in the existing mini project given to Interns
- Install Prometheus and Grafana using Docker (with docker-compose)
- Configure prometheus (scrape configs) such way that it can scrape the metrics from default metric path of the application job
- Validate the entire configuration to check if the data is coming or not in Prometheus UI
- Create the Dashboards in Grafana on top of the metrics exported by adding the Prometheus as a Datasource.

prometheus-project/

```
|— app/
|   |— Dockerfile
|   |— requirements.txt
|   |— your_flask_app.py
|— prometheus.yml
|— docker-compose.yml
```

docker-compose.yml

services:

app:

build:

context: ./app

ports:

- "8080:8080"

prometheus:

image: prom/prometheus

ports:

- "9091:9090"

volumes:

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

grafana:

image: grafana/grafana

ports:

- "3001:3000"

prometheus.yml

```
scrape_interval: 15s
scrape_configs:
  - job_name: 'flask_app'
    static_configs:
      - targets: ['app:8080']
```

app/Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "your_flask_app.py"]
```

app/requirements.txt

```
s.txt
Flask
prometheus_flask_exporter
```

app/your_flask_app.py

```
from flask import Flask
from prometheus_flask_exporter import PrometheusMetrics
```

```
app = Flask(__name__)
metrics = PrometheusMetrics(app)
```

```
@app.route('/')
def hello():
    return 'Hello, World!'
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

```

sigmoid@ssigmoid-ThinkPad-T450:~$ cd Desktop/
sigmoid@ssigmoid-ThinkPad-T450:~/Desktop$ cd prometheus_project/
sigmoid@ssigmoid-ThinkPad-T450:~/Desktop/prometheus_project$ docker-compose up -d
Creating network "prometheus_project_default" with the default driver
Creating prometheus_project_app_1      ... done
Creating prometheus_project_prometheus_1 ... done
Creating prometheus_project_grafana_1  ... done

```

