

## **Assignment Report**

**Project Title :** Dynamic Load Balancing for a Smart Grid using Microservices

**Student Name:** Nitin Kumar

**Registration ID:** G24AI2056

**Course:** Fundamentals of Distributed Systems

**Submission Date:** June 25, 2025

**Github Repository:** <https://github.com/Nitin5499/Dynamic-Load-Balancing-for-a-Smart-Grid>

---

### **Project Overview**

The objective of this assignment is to simulate a dynamic load balancing mechanism for a smart grid system. By leveraging a microservices-based architecture, we aim to distribute charging loads among substations efficiently. Key technologies include Docker, Prometheus, Grafana, and Flask for service orchestration, monitoring, and visualization.

### **Problem Statement**

In a distributed environment, operations can occur concurrently on different nodes. Since there's no global clock, it becomes difficult to determine the correct order in which operations should be applied. This can lead to inconsistencies in data when updates are propagated without regard to their causal relationships.

The goal of this project was to simulate such a system and build a solution that:

- Maintains data consistency across nodes.
- Respects the causal order of operations.
- Buffers and reorders updates, when necessary, especially under delayed network conditions.

### **Service Descriptions**

#### **charge\_request\_service**

Handles incoming vehicle charging requests. This microservice simulates demand generation from electric vehicles that need charging.

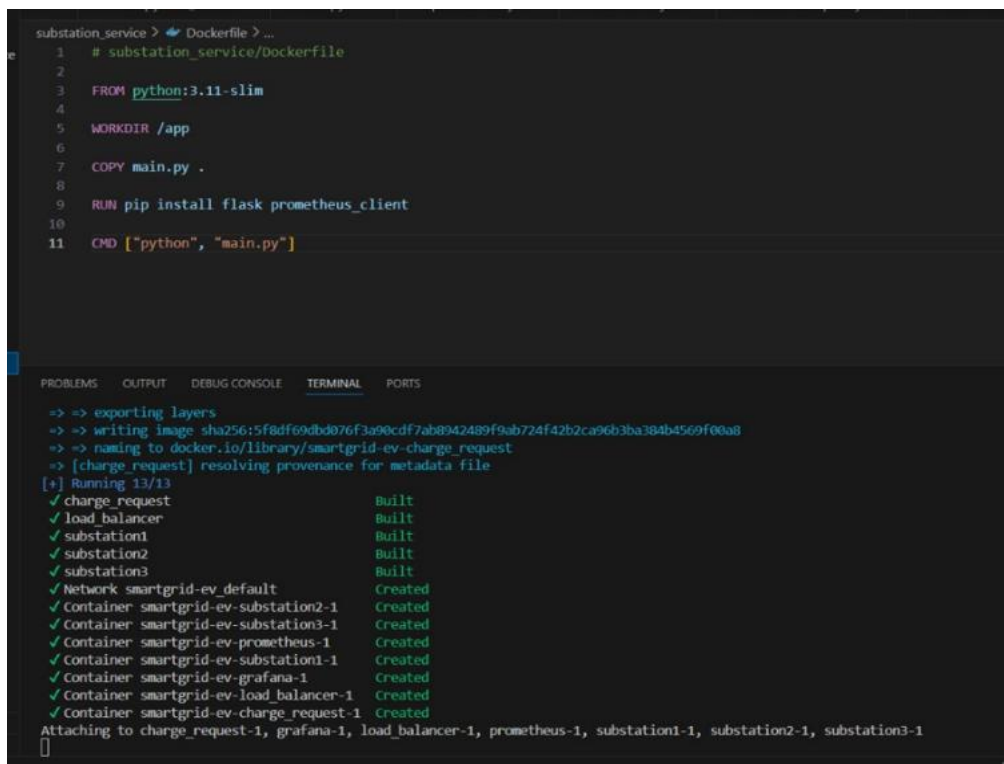
#### **substation\_service**

Simulates multiple substations that respond to charging requests. Each substation has a load capacity, and responses depend on the current load status.

## load\_balancer/main.py

The load balancer is the core component that intelligently assigns incoming charging requests to the substation with the lowest load. It uses basic logic to fetch current load values from each substation before assignment.

## Screenshots



```
substation_service > Dockerfile > ...
1 # substation_service/Dockerfile
2
3 FROM python:3.11-slim
4
5 WORKDIR /app
6
7 COPY main.py .
8
9 RUN pip install flask prometheus_client
10
11 CMD ["python", "main.py"]

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=> => exporting layers
=> => writing image sha256:5f8df69dbd076f3a90cdf7ab0942489f9ab724f42b2ca96b3ba384b4569f00a8
=> => naming to docker.io/library/smartgrid-ev-charge_request
=> [charge_request] resolving provenance for metadata file
[*] Running 13/13
✓ charge_request Built
✓ load_balancer Built
✓ substation1 Built
✓ substation2 Built
✓ substation3 Built
✓ Network smartgrid-ev_default Created
✓ Container smartgrid-ev-substation2-1 Created
✓ Container smartgrid-ev-substation3-1 Created
✓ Container smartgrid-ev-prometheus-1 Created
✓ Container smartgrid-ev-substation1-1 Created
✓ Container smartgrid-ev-grafana-1 Created
✓ Container smartgrid-ev-load_balancer-1 Created
✓ Container smartgrid-ev-charge_request-1 Created
Attaching to charge_request-1, grafana-1, load_balancer-1, prometheus-1, substation1-1, substation2-1, substation3-1
```

```
substation_service > Dockerfile > ...
1 # substation_service/Dockerfile
2
3 FROM python:3.11-slim
4
5 WORKDIR /app
6
7 COPY main.py .
8
9 RUN pip install flask prometheus_client
10
11 CMD ["python", "main.py"]
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
substation2-1	172.21.0.3 - - [19/Jun/2025 17:37:36]	"GET /metrics HTTP/1.1"	404 -	
substation1-1	172.21.0.3 - - [19/Jun/2025 17:37:37]	"GET /metrics HTTP/1.1"	404 -	
substation1-1	172.21.0.6 - - [19/Jun/2025 17:37:39]	"GET /metrics HTTP/1.1"	404 -	
substation2-1	172.21.0.6 - - [19/Jun/2025 17:37:39]	"GET /metrics HTTP/1.1"	404 -	
substation3-1	172.21.0.6 - - [19/Jun/2025 17:37:39]	"GET /metrics HTTP/1.1"	404 -	
substation3-1	172.21.0.3 - - [19/Jun/2025 17:37:41]	"GET /metrics HTTP/1.1"	404 -	
substation2-1	172.21.0.3 - - [19/Jun/2025 17:37:41]	"GET /metrics HTTP/1.1"	404 -	
substation1-1	172.21.0.3 - - [19/Jun/2025 17:37:42]	"GET /metrics HTTP/1.1"	404 -	
substation1-1	172.21.0.6 - - [19/Jun/2025 17:37:44]	"GET /metrics HTTP/1.1"	404 -	
substation2-1	172.21.0.6 - - [19/Jun/2025 17:37:44]	"GET /metrics HTTP/1.1"	404 -	

## Monitoring Setup

### Prometheus Configuration

Prometheus is configured to scrape metrics from all services. We'll need a `prometheus.yml` file under a `/prometheus/` directory. This file defines the targets to monitor and scraping intervals.

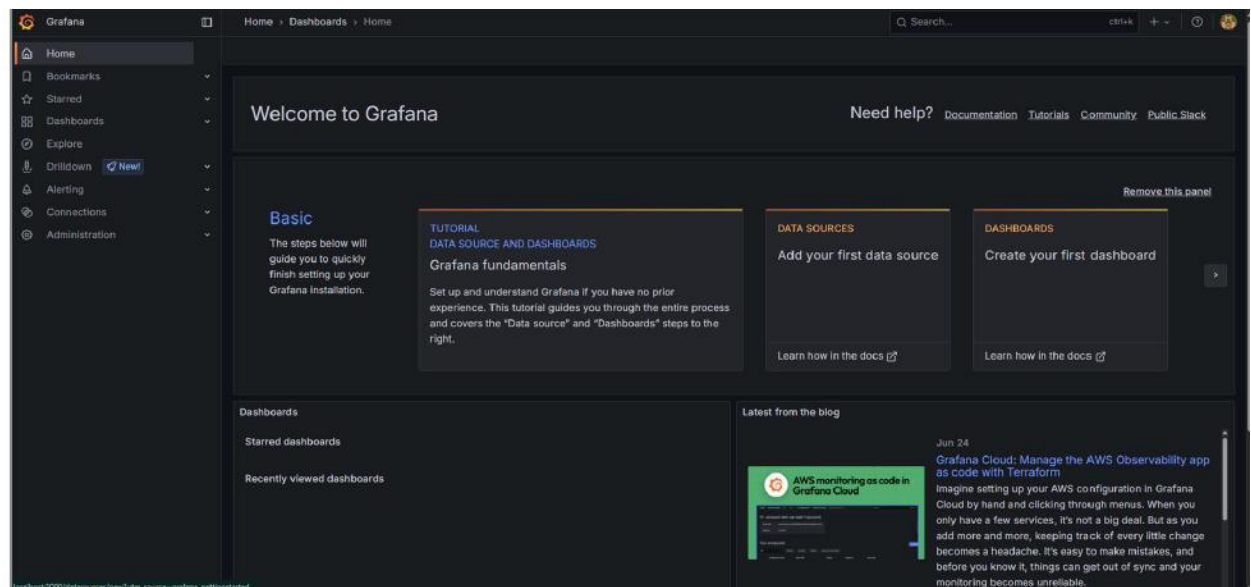
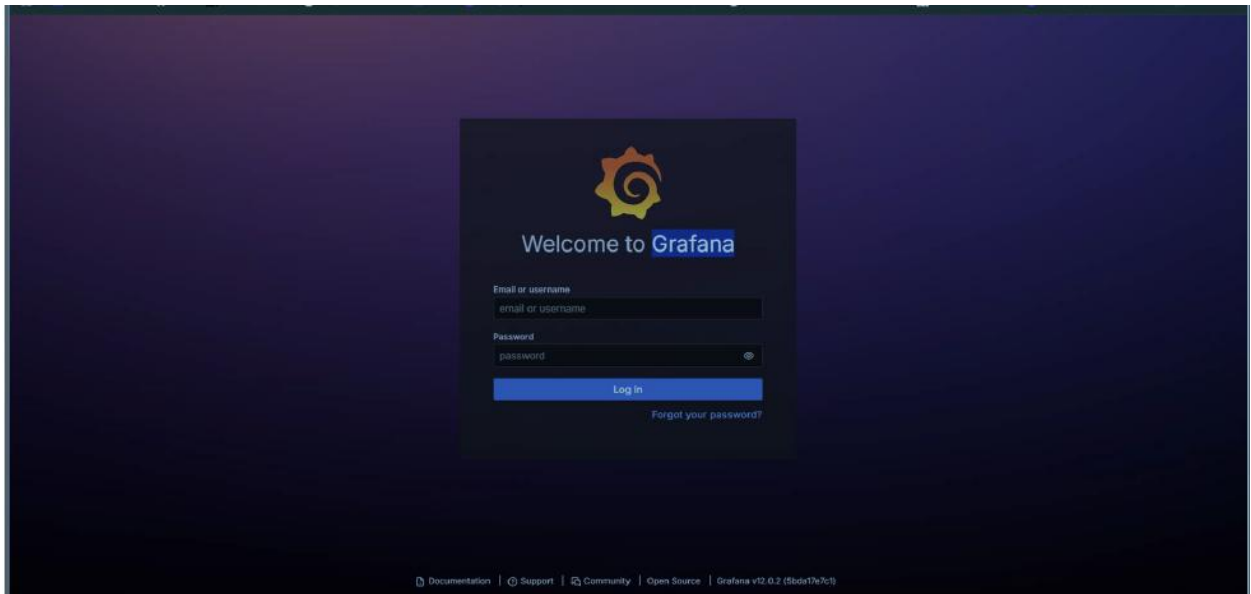
### Load Testing

A simple load tester script is included to simulate multiple EV charging requests. It sends parallel requests to test how the load balancer distributes them across substations.

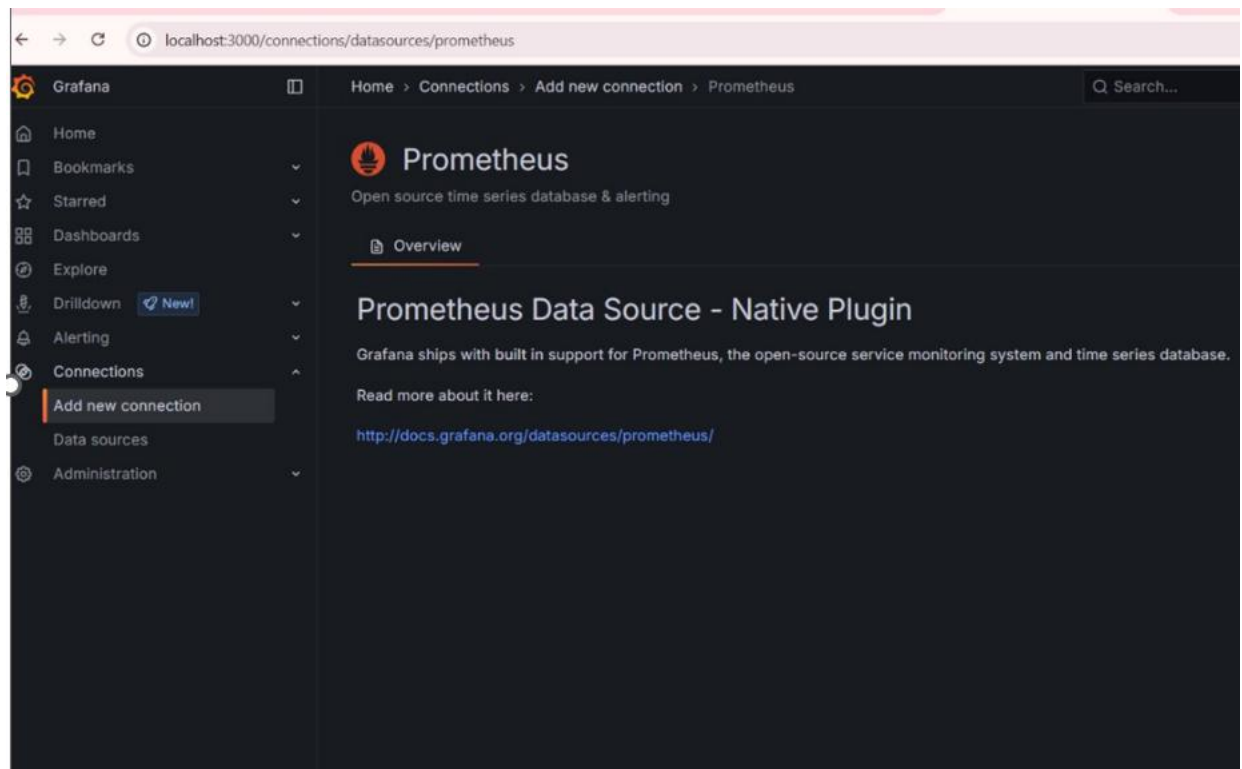
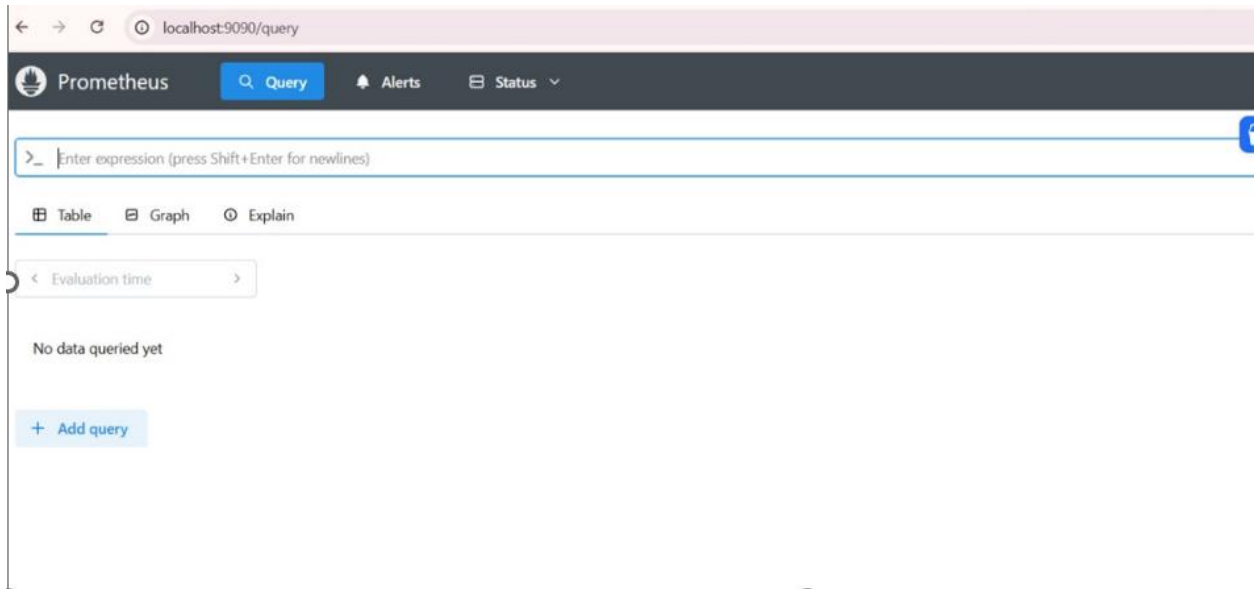
### Docker & System Execution

All components are containerized using Docker. A `docker-compose.yml` file handles orchestration.

- Grafana: <http://localhost:3000>



Prometheus: <http://localhost:9090>



## Load Testing

```
src > node.py > VectorClock > __init__
3 import time
4
5 # ----- VectorClock Class -----
6
7 class VectorClock:
8     def __init__(self, node_id, all_nodes):
9         self.clock = {nid: 0 for nid in all_nodes}
10        self.node_id = node_id
11
12        def increment(self):
13            self.clock[self.node_id] += 1
14
15        def update(self, received_clock):
16            for node, val in received_clock.items():
17                self.clock[node] = max(self.clock.get(node, 0), val)
18
19        def is causally ready(self, received_clock, sender_id):
20
21 node1 | * Running on http://127.0.0.1:5000
22 node1 | * Running on http://172.21.0.2:5000
23 node1 | Press CTRL+C to quit
24 node3 | [node3] Node started with clock: {'node1': 0, 'node2': 0, 'node3': 0}
25 node3 | * Serving Flask app 'node'
26 node3 | * Debug mode: off
27 node3 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
28 node3 | * Running on all addresses (0.0.0.0)
29 node3 | * Running on http://127.0.0.1:5000
30 node3 | * Running on http://172.21.0.3:5000
31 node3 | Press CTRL+C to quit
32 node1 | 172.21.0.1 - - [18/Jun/2025 17:05:20] "POST /put HTTP/1.1" 200 -
33 node3 | 172.21.0.1 - - [18/Jun/2025 17:05:21] "POST /replicate HTTP/1.1" 200 -
34 node2 | 172.21.0.1 - - [18/Jun/2025 17:05:21] "POST /put HTTP/1.1" 200 -
35 node3 | 172.21.0.1 - - [18/Jun/2025 17:05:22] "POST /replicate HTTP/1.1" 200 -
36 node3 | 172.21.0.1 - - [18/Jun/2025 17:05:25] "GET /get?key=x HTTP/1.1" 200 -
37
38 * History restored
39 View in Docker Desktop View Config Enable Watch
40 PS E:\vector-clock-kv-store> docker-compose down
```

```
3 services:
4   charge_request:
5     ports:
6       - 5002:5002
7     depends_on:
8       - load_balancer
9
10    # Load Balancer
11    > Run Service
12    load_balancer:
13      build: ./load_balancer
14      ports:
15        - 5002:5002
16      depends_on:
17        - substation1
18        - substation2
19        - substation3
20
21 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
22
23 => => exporting layers 1.3s
24 => => writing image sha256:f01033f86b250ebb84d77e34ebacbea5a3321660021be8775bdb9303740bffe 0.0s
25 => => naming to docker.io/library/smartgrid-ev-charge_request 0.1s
26 => [charge_request] resolving provenance for metadata file 0.1s
27 [v] Running 13/13
28 ✓ charge_request Built 0.0s
29 ✓ load_balancer Built 0.0s
30 ✓ substation1 Built 0.0s
31 ✓ substation2 Built 0.0s
32 ✓ substation3 Built 0.0s
33 ✓ Network smartgrid-ev_default Created 0.4s
34 ✓ Container smartgrid-ev-prometheus-1 Created 2.7s
35 ✓ Container smartgrid-ev-substation3-1 Created 2.7s
36 ✓ Container smartgrid-ev-substation2-1 Created 2.7s
37 ✓ Container smartgrid-ev-substation1-1 Created 2.7s
38 ✓ Container smartgrid-ev-grafana-1 Created 3.4s
39 ✓ Container smartgrid-ev-load_balancer-1 Created 3.4s
40 ✓ Container smartgrid-ev-charge_request-1 Created 0.4s
41 Attaching to charge_request-1, grafana-1, load_balancer-1, prometheus-1, substation1-1, substation2-1, substation3-1
```

