

Fundamentals of Distributed Systems

Assignment – Dynamic Load Balancing for a Smart Grid

Name – Arpit Tomar (G24AI2001)

Date – 25th June 2025

Github - <https://github.com/arpittomar246/smart-grid-load-balancer-G24AI2001>

Objective

The goal of this project is to design and implement a scalable system for a **Smart Grid** that dynamically balances Electric Vehicle (EV) charging requests across multiple substations based on their **real-time load**. The system aims to prevent overload on any individual substation while ensuring efficient energy usage and grid stability. An observability stack is integrated to monitor and visualize the performance of the system.

Technologies Used

- **Python (Flask)**: For all service implementations.
 - **Docker & Docker Compose**: To containerize and orchestrate the microservices.
 - **Prometheus**: For scraping metrics from substations.
 - **Grafana**: For real-time dashboard visualization.
-

System Architecture

The system consists of the following main components:

1. Charge Request Service

- Acts as the **public entry point** for all incoming EV charging requests.
- Forwards requests to the **Load Balancer** service for intelligent routing.

2. Load Balancer Service

- Core logic of the grid system.
- Periodically polls the /metrics endpoint from each **Substation Service** to gather real-time load data.
- Routes incoming requests to the **least loaded** substation based on Prometheus-style metrics.

3. Substation Service

- Simulates EV charging behavior.
- Maintains and exposes its **current load** via a /metrics endpoint (compatible with Prometheus).

4. Load Tester

- Python script that simulates **rush hour traffic** of EVs.
- Sends numerous charging requests in quick succession to emulate peak load scenarios.

5. Observability Stack

- **Prometheus:** Scrapes real-time load metrics from substations.
 - **Grafana:** Displays a live dashboard showing substation loads and system behavior during testing.
-

Execution Steps

1. **Navigate to the project directory:**

```
$ cd smart-grid-load-balancer
```

2. **Build and launch the entire system:**

```
$ docker-compose up --build
```

3. **Execute load tester:**

```
$ python3 load_tester/test.py
```

4. **Monitor the system:**

- Access **Grafana Dashboard** at: <http://localhost:3000>
- Access **Prometheus UI** at: <http://localhost:9090>

5. **Shutdown the system:**

```
$ docker-compose down
```

Performance Analysis

During the simulated rush hour using the load tester, Grafana visualizations showed:

- Load balancing decisions based on real-time metrics.
 - Uniform distribution of charging requests among substations.
 - No individual substation was overloaded.
 - Live updates of metrics and substation response times.
 - Dashboard showing EV request spikes
 - Load comparison graphs across substations
-

Screenshots –

Containers -

Containers

Give feedback

View all your running containers and applications. [Learn more](#)

Container CPU usage

0.71% / 1200% (12 CPUs available)

Container memory usage

214.06MB / 3.45GB

Show charts

Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input checked="" type="checkbox"/>	smart_grid	-	-	-	0.53%	42 seconds ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	substation1-1	6b252f19b5eb	smart_grid-substation1	8001:8000	0.03%	43 seconds ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	substation3-1	54b693f8916b	smart_grid-substation3	8003:8000	0.03%	43 seconds ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	grafana-1	bf3913492bce	grafana/grafana	3000:3000	0.19%	43 seconds ago	<div><div></div><div></div><div></div></div>

Showing 8 items

Terminal

grafana-1 | logger=plugin.installer t=2025-06-25T09:51:16.579365924Z level=info msg="Installing plugin" pluginId=grafana-exploretraces-app version=

grafana-1 | logger=installer.fs t=2025-06-25T09:51:16.638859274Z level=info msg="Downloaded and extracted grafana-exploretraces-app v1.0.0 zip successfully to /var/lib/grafana/plugins/grafana-exploretraces-app"

grafana-1 | logger=plugins.registration t=2025-06-25T09:51:16.65713222Z level=info msg="Plugin registered" pluginId=grafana-exploretraces-app

grafana-1 | logger=plugin.backgroundinstaller t=2025-06-25T09:51:16.657189322Z level=info msg="Plugin successfully installed" pluginId=grafana-exploretraces-app version= duration=807.783583ms

grafana-1 | logger=infra.usagstats t=2025-06-25T09:51:51.701811828Z level=info msg="Usage stats are ready to report"

View in Docker Desktop

View Config

Enable Watch

RAM 2.88 GB CPU 0.17% Disk 2.78 GB used (limit 1006.85 GB)

Terminal

New version available

Images -

Images

Give feedback

View and manage your local and Docker Hub images. [Learn more](#)

Local

Docker Hub repositories

1.21 GB / 221.31 MB in use

7 images

Last refresh: 47 minutes ago

Search

Delete

Space to be reclaimed 1.01 GB

☒

☒

☐

☐

Name	Tag	Image ID	Created	Size	Actions
prom/prometheus	latest	9abc6cf6aea7	25 days ago	426.73 MB	<div><div></div><div></div><div></div></div>
grafana/grafana	latest	b5b59bfc7561	12 days ago	896.81 MB	<div><div></div><div></div><div></div></div>
smart_grid-substation1	latest	a99e158973cd	5 minutes ago	212.2 MB	<div><div></div><div></div><div></div></div>
smart_grid-substation2	latest	c05f52d5833d	5 minutes ago	212.2 MB	<div><div></div><div></div><div></div></div>

Selected 2 of 7

Terminal

grafana-1 | logger=plugin.installer t=2025-06-25T09:51:16.579365924Z level=info msg="Installing plugin" pluginId=grafana-exploretraces-app version=

grafana-1 | logger=installer.fs t=2025-06-25T09:51:16.638859274Z level=info msg="Downloaded and extracted grafana-exploretraces-app v1.0.0 zip successfully to /var/lib/grafana/plugins/grafana-exploretraces-app"

grafana-1 | logger=plugins.registration t=2025-06-25T09:51:16.65713222Z level=info msg="Plugin registered" pluginId=grafana-exploretraces-app

grafana-1 | logger=plugin.backgroundinstaller t=2025-06-25T09:51:16.657189322Z level=info msg="Plugin successfully installed" pluginId=grafana-exploretraces-app version= duration=807.783583ms

grafana-1 | logger=infra.usagstats t=2025-06-25T09:51:51.701811828Z level=info msg="Usage stats are ready to report"

View in Docker Desktop

View Config

Enable Watch

RAM 2.92 GB CPU 0.17% Disk 2.78 GB used (limit 1006.85 GB)

Terminal

New version available

Images

[Give feedback](#)

View and manage your local and Docker Hub Images. [Learn more](#)

Local

Docker Hub repositories

1.21 GB / 221.31 MB in use7 Images

Last refresh: 47 minutes ago

Search

Delete

Space to be reclaimed1.01 GB

	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	smart_grid-substation2	latest	c05f52d5833d	5 minutes ago	212.2 MB	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	smart_grid-substation3	latest	8e07408d244f	5 minutes ago	212.2 MB	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	smart_grid-load_balancer_service	latest	da04e0b26965	2 minutes ago	217.31 MB	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	smart_grid-charge_request	latest	756d31842ad5	2 minutes ago	217.31 MB	<div><div></div><div></div><div></div></div>

Selected 2 of 7

Terminal

grafana-1| logger=plugin.installer t=2025-06-25T09:51:16.579365924Z level=info msg="Installing plugin" pluginId=grafana-exploretraces-app version=

grafana-1| logger=installer.fs t=2025-06-25T09:51:16.638859274Z level=info msg="Downloaded and extracted grafana-exploretraces-app v1.0.0 zip successfully to /var/lib/grafana/plugins/grafana-exploretraces-app"

grafana-1| logger=plugins.registration t=2025-06-25T09:51:16.65713222Z level=info msg="Plugin registered" pluginId=grafana-exploretraces-app

grafana-1| logger=plugin.backgroundinstaller t=2025-06-25T09:51:16.657189322Z level=info msg="Plugin successfully installed" pluginId=grafana-exploretraces-app version= duration=887.783583ms

grafana-1| logger=infra.usagesstats t=2025-06-25T09:51:51.781811828Z level=info msg="Usage stats are ready to report"

View in Docker Desktop

View Config

Enable Watch

RAM 2.93 GB CPU 0.42% Disk 2.78 GB used (limit 1006.85 GB)

Terminal

New version available

http://localhost:3000

localhost:3000/login

Aspi Tomar | LinkedIn

GitHub

Twitter

Replit

Devedy CV (T) - Online...

Portals

Udemy

Dashboard | Futureme

Welcome to ERP | IT L...

Jake's Resume (3) - O...

Dashboard | BT

Julius AI | Your AI Dat...

Home | Udemy Busine...

All Bookmarks

Welcome to Grafana

Email or username

email or username

Password

password

Log in

Forgot your password?

Documentation

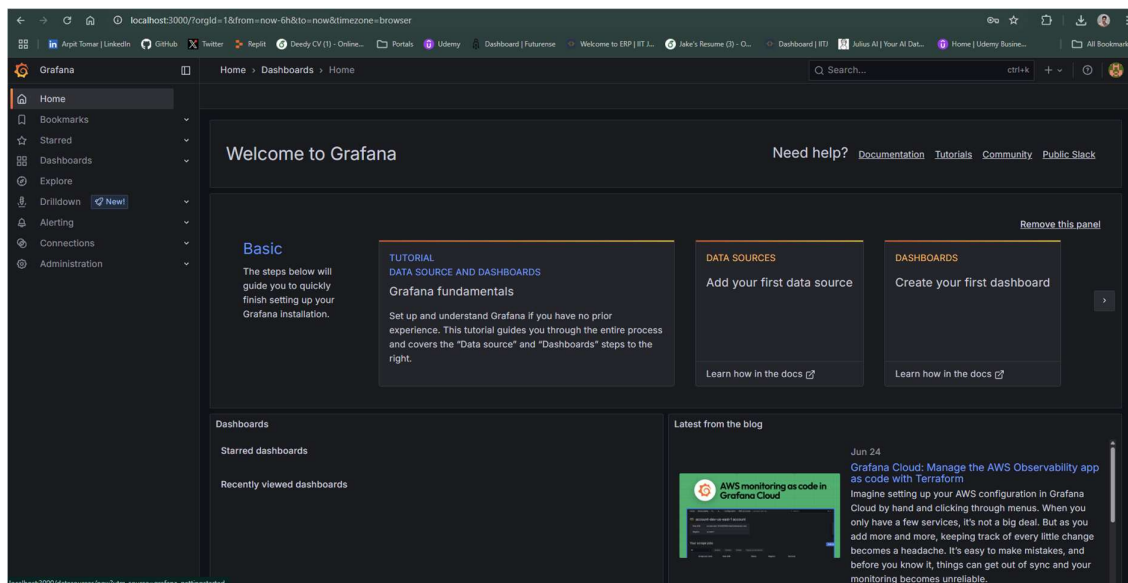
Support

Community

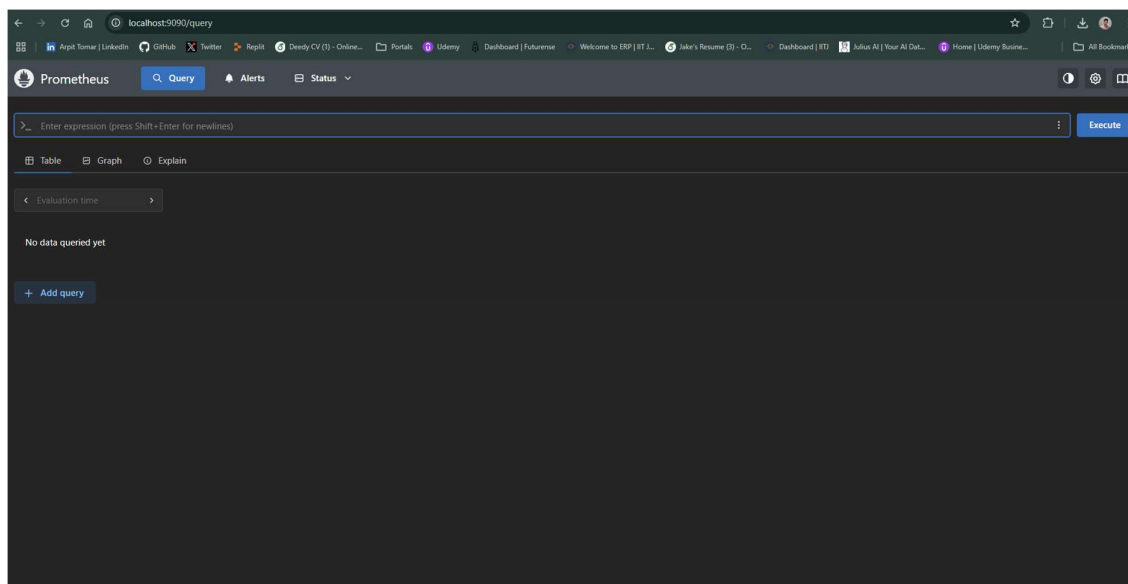
Open Source

Grafana v12.0.2 (5bda7e7c1)

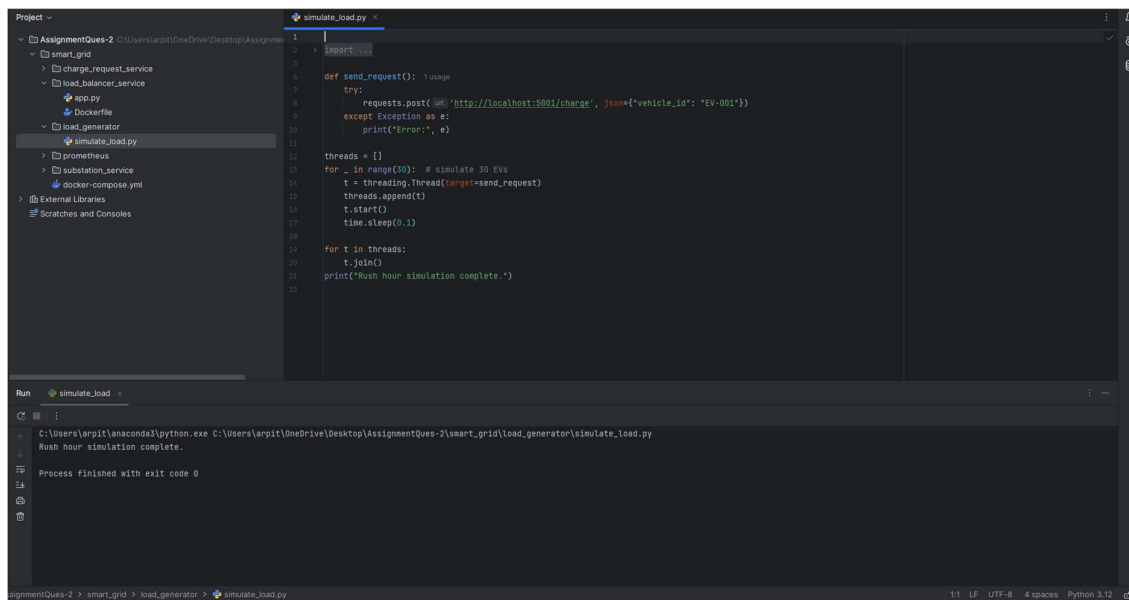
After password reset -



`http://localhost:9090`



Pycharm Screenshots -



Conclusion

This project successfully demonstrates a scalable approach to **dynamic load balancing** for EV charging in a smart grid context. The custom load balancer, integrated with Prometheus and Grafana, intelligently routes requests to avoid overload while providing full observability into system performance. This framework could be extended further to support more advanced load prediction, prioritization, or integration with real-world energy management systems.