Project Report: Smart grid EV charging request load balancing

Objective: To design and build a scalable system for a Smart Grid that dynamically balances Electric
Vehicle (EV) charging requests across multiple substations based on their real-time load,
complete with a comprehensive observability stack

Architecture:

Technology Stack: Python (Flask, Requests, Prometheus Client), Docker, Docker Compose, Prometheus, Grafana.

Component Descriptions:

Charge Request Service:
API for EVs to submit charging requests

Load Balancer:
Routes requests to the least-loaded substation.

Substation Service:
Simulates charging and exposes Prometheus metrics

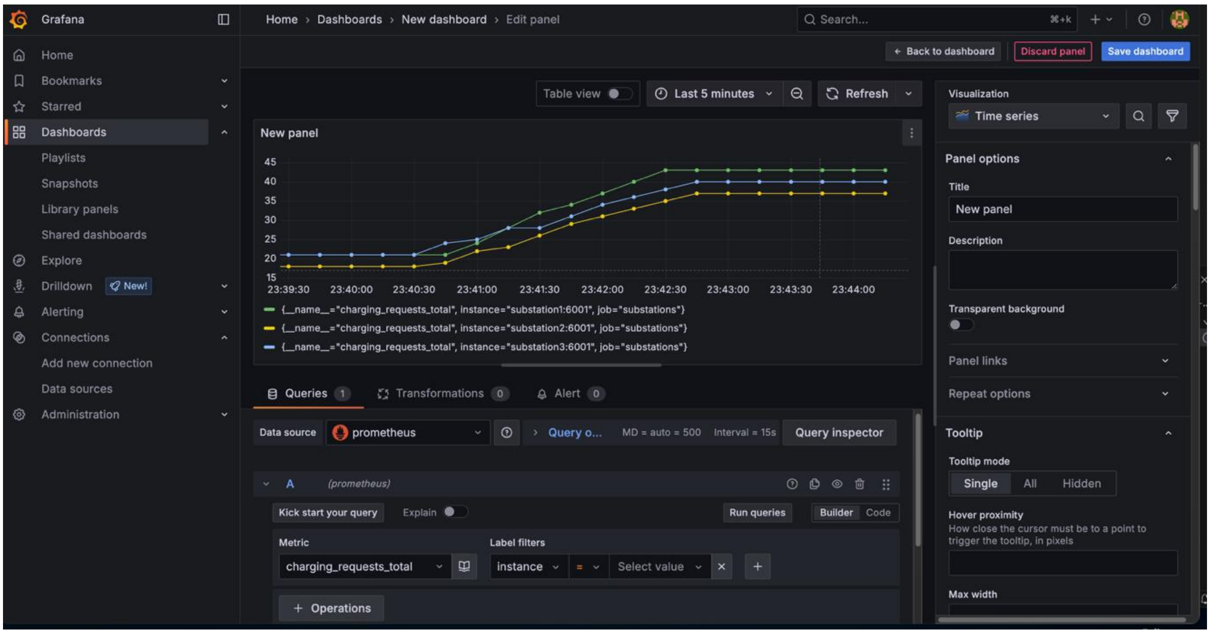
Monitoring Stack:
Prometheus (scrapes metrics) + Grafana (visualization)

Data Flow

EV->ChargeRequest->LoadBalancer->Substation->Prometheus->Grafana

Performance Analysis:

Grafana Screenshots



{__name__="charging_requests_total", instance="substation1:6001", job="substations"}

{__name__="charging_requests_total", instance="substation2:6001", job="substations"}

{__name__="charging_requests_total", instance="substation3:6001", job="substations"}

Data source: prometheus

Query o... MD = auto = 500 Interval = 15s

A (prometheus)

Explain

Metric: charging_requests_total

Label filters: instance = Select value

Visualization: Time series

Title: New panel

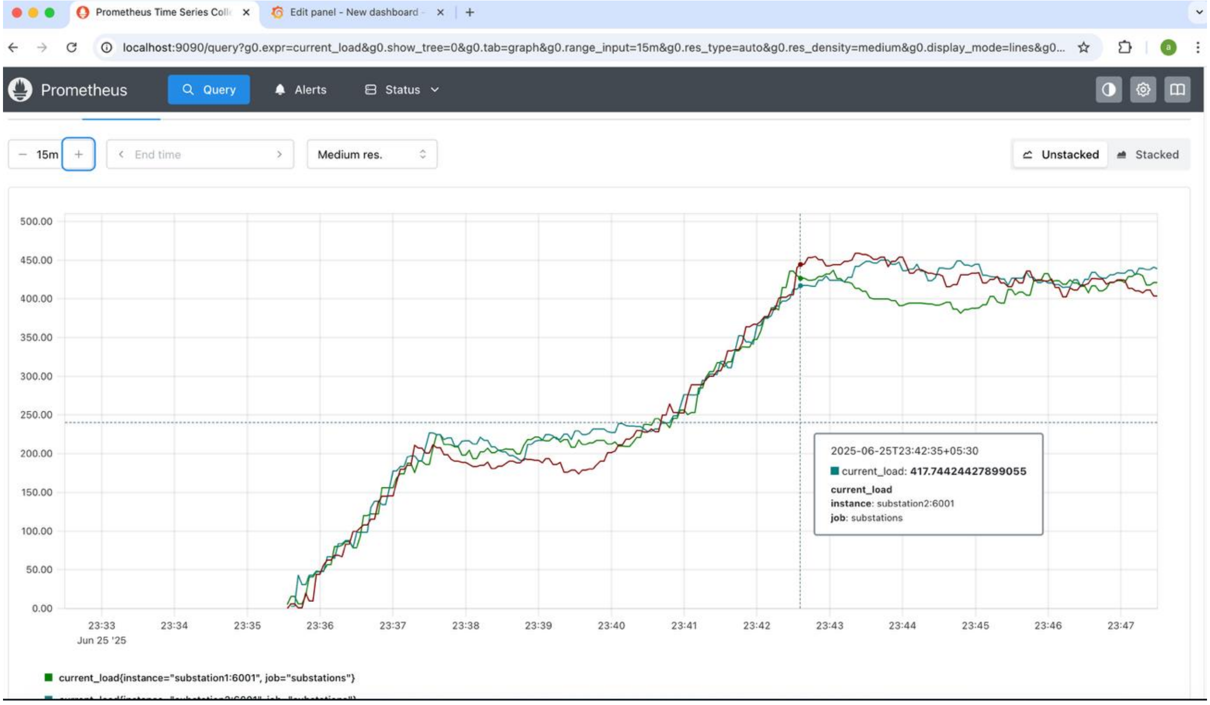
Description:

Transparent background

Tooltip mode: Single All Hidden

Hover proximity: How close the cursor must be to a point to trigger the tooltip, in pixels

Prometheus screenshot



localhost:9090/query?g0.expr=current_load&g0.show_tree=0&g0.tab=graph&g0.range_input=15m&g0.res_type=auto&g0.res_density=medium&g0.display_mode=lines&g0...

2025-06-25T23:42:35+05:30

current_load: 417.74424427899055

current_load

instance: substation2:6001

job: substations

current_load{instance="substation1:6001", job="substations"}

Analysis:

The load is almost equally distributed among all participating nodes.