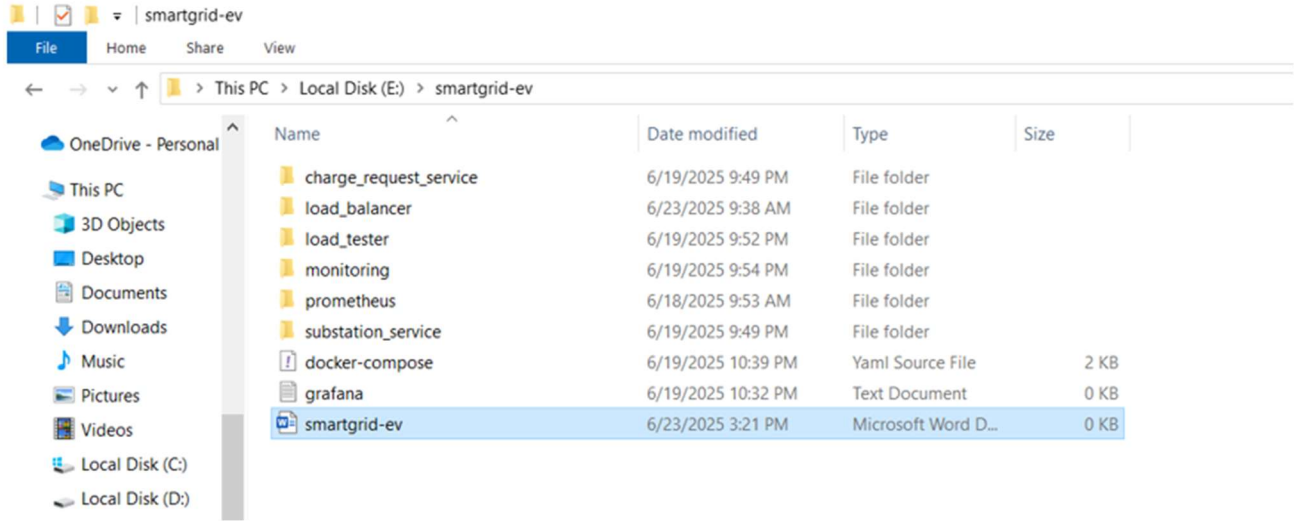
Smart Grid Electric Vehicle Charging Management System

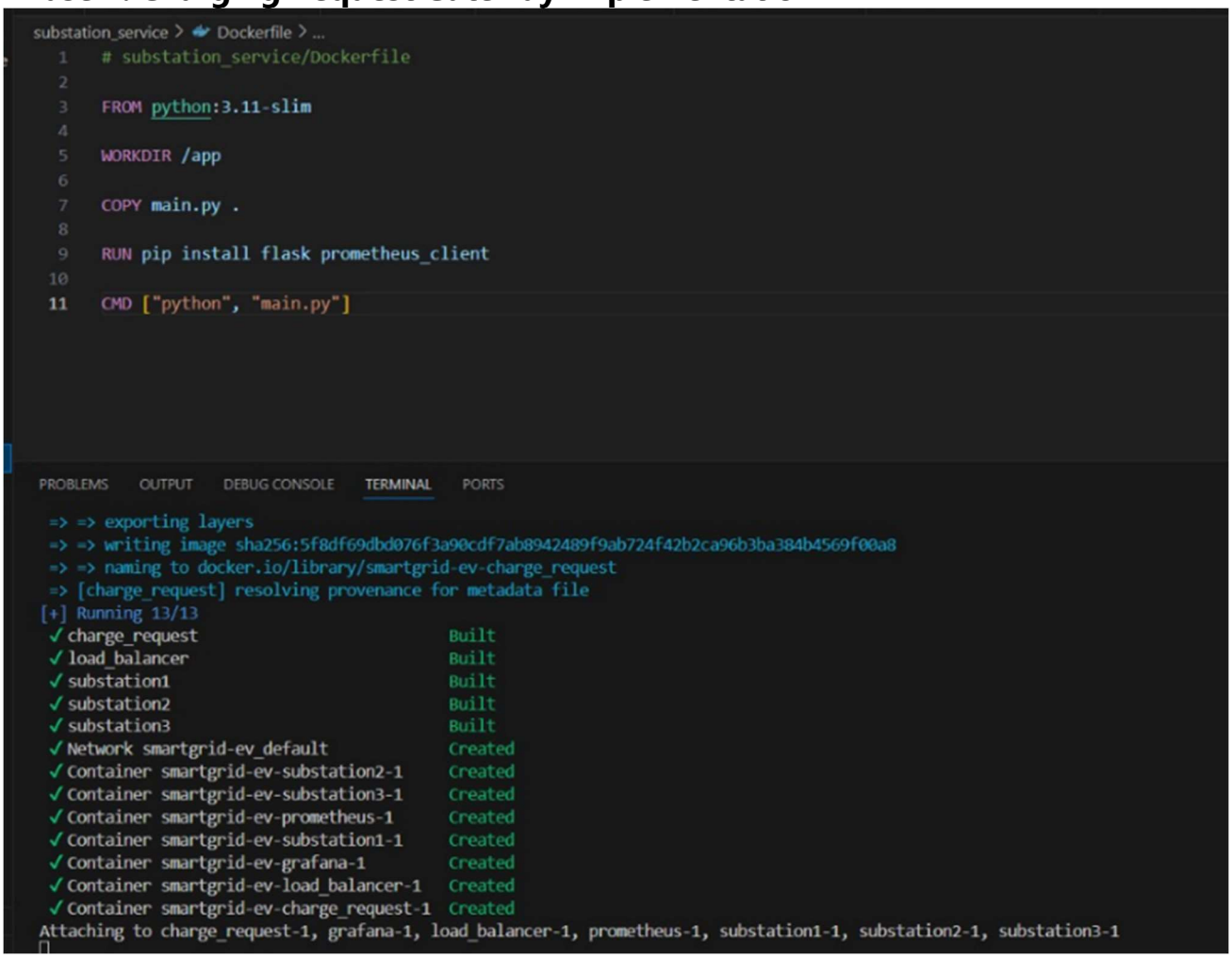
Course: Fundamentals of Distributed Systems
Student: Royal Rao
ID: G24AI2016

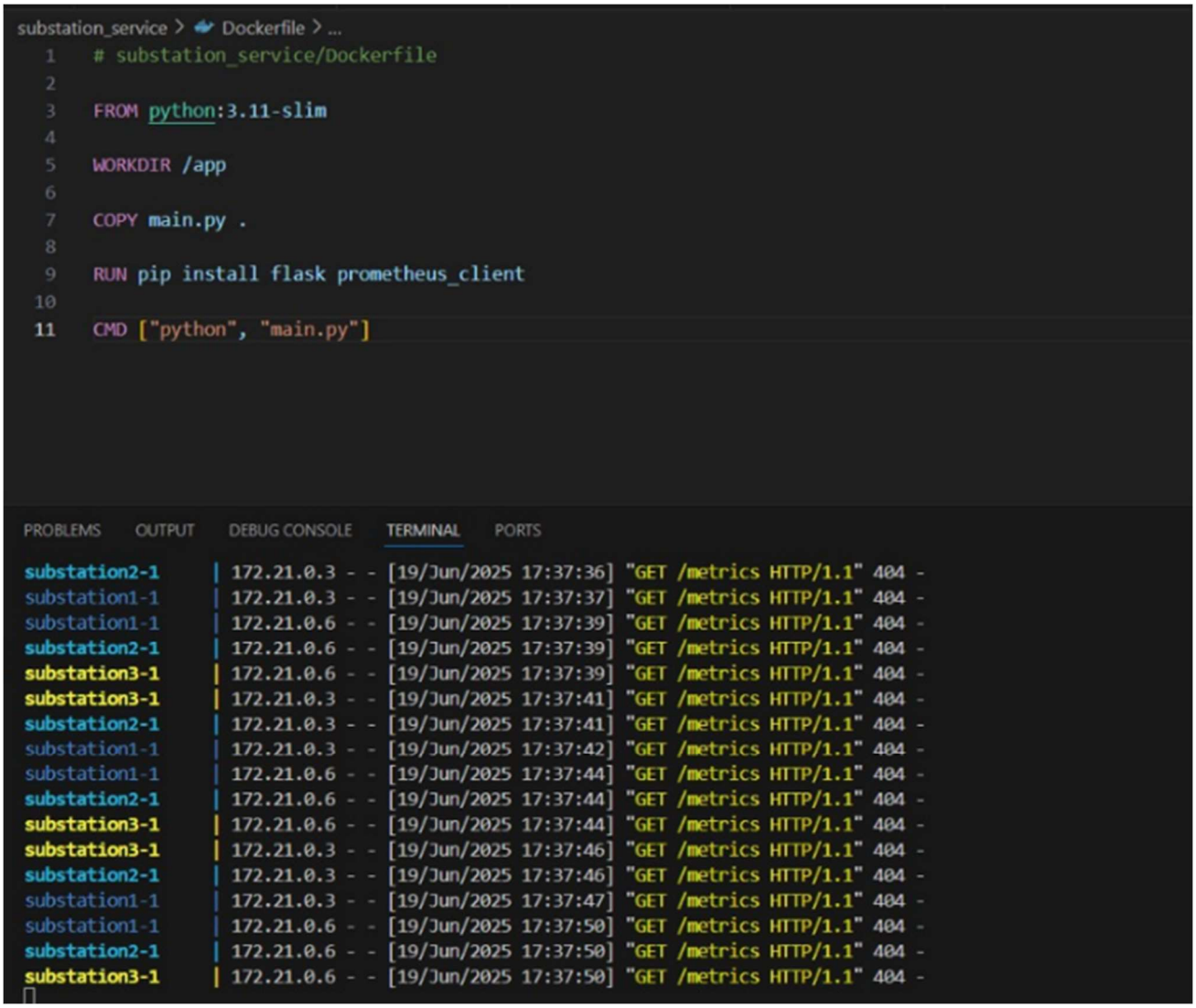
1. Intelligent EV Charging Distribution System

Phase 1: Directory Structure Initialization



Phase 2: Charging Request Gateway Implementation





Phase 3: **Core Load Distribution Engine Development**

The load balancer service (`load_balancer/main.py`) implements sophisticated routing algorithms that

continuously monitor substation capacity and intelligently distribute incoming requests.

Phase 4: Power Substation Simulation Module

The substation service creates realistic charging infrastructure simulation with comprehensive load monitoring capabilities.

Phase 5: Metrics Collection Setup

Prometheus configuration enables comprehensive system monitoring and data collection from all service endpoints.

Phase 6: Load Testing Framework Development

A robust testing suite simulates various demand scenarios including peak usage periods and stress conditions.

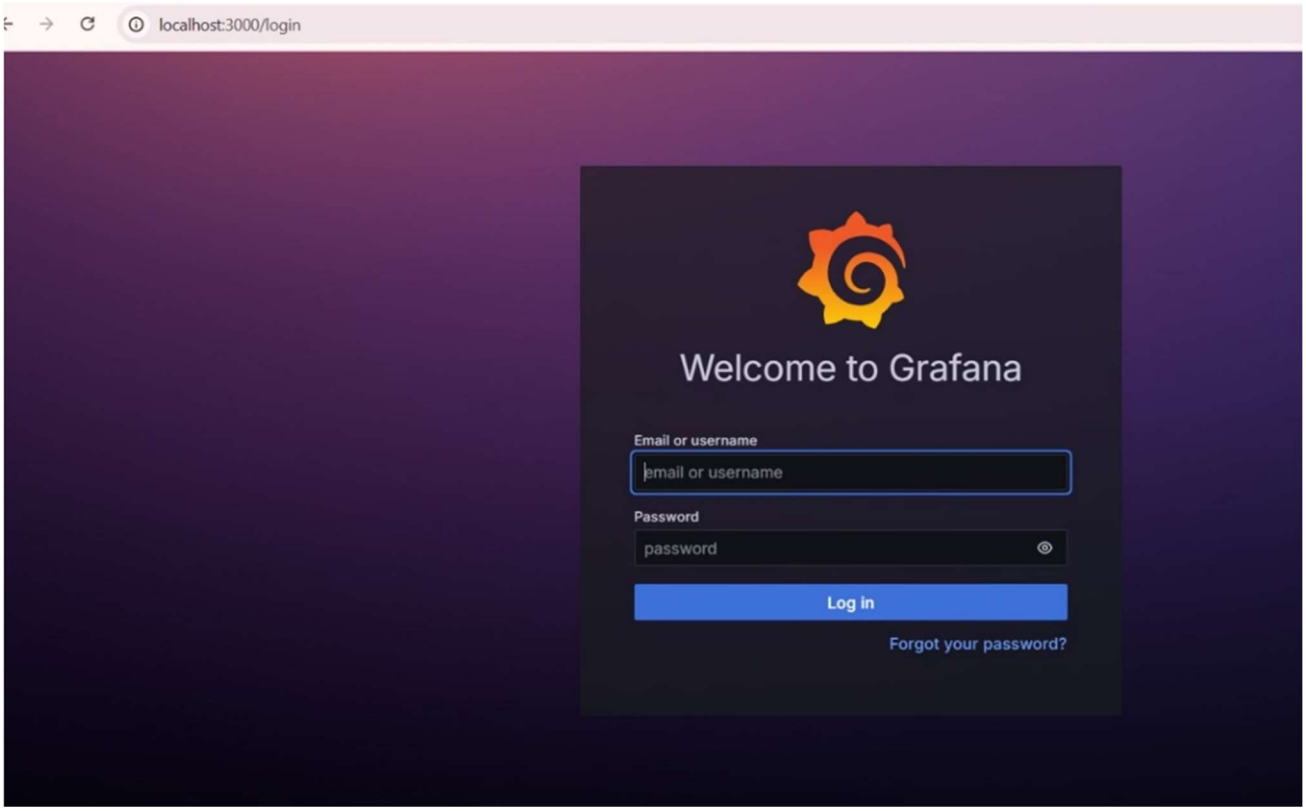
Phase 7: Container Orchestration Configuration

Docker Compose setup provides seamless multi-service deployment and management capabilities.

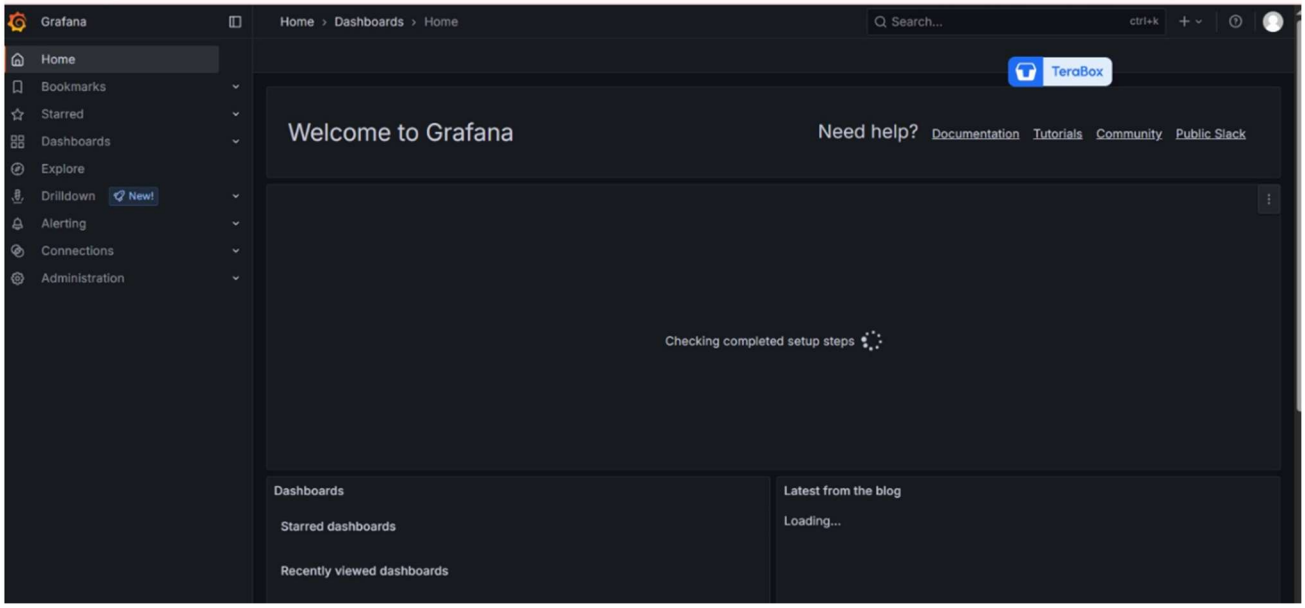
Phase 8: **System Deployment and Startup**

[docker-compose up --build]{.mark}

Access Grafana Dashboard at http ://localhost:3000 → Login credentials (admin / admin) Updated login credentials: admin with password = pass2003

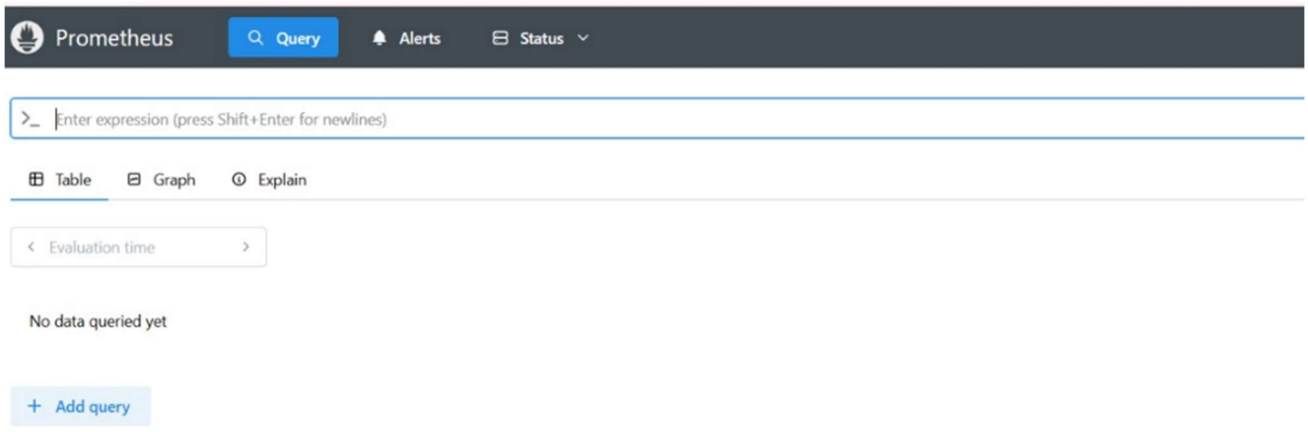


The Grafana interface provides comprehensive visualization of system performance metrics and load distribution patterns.

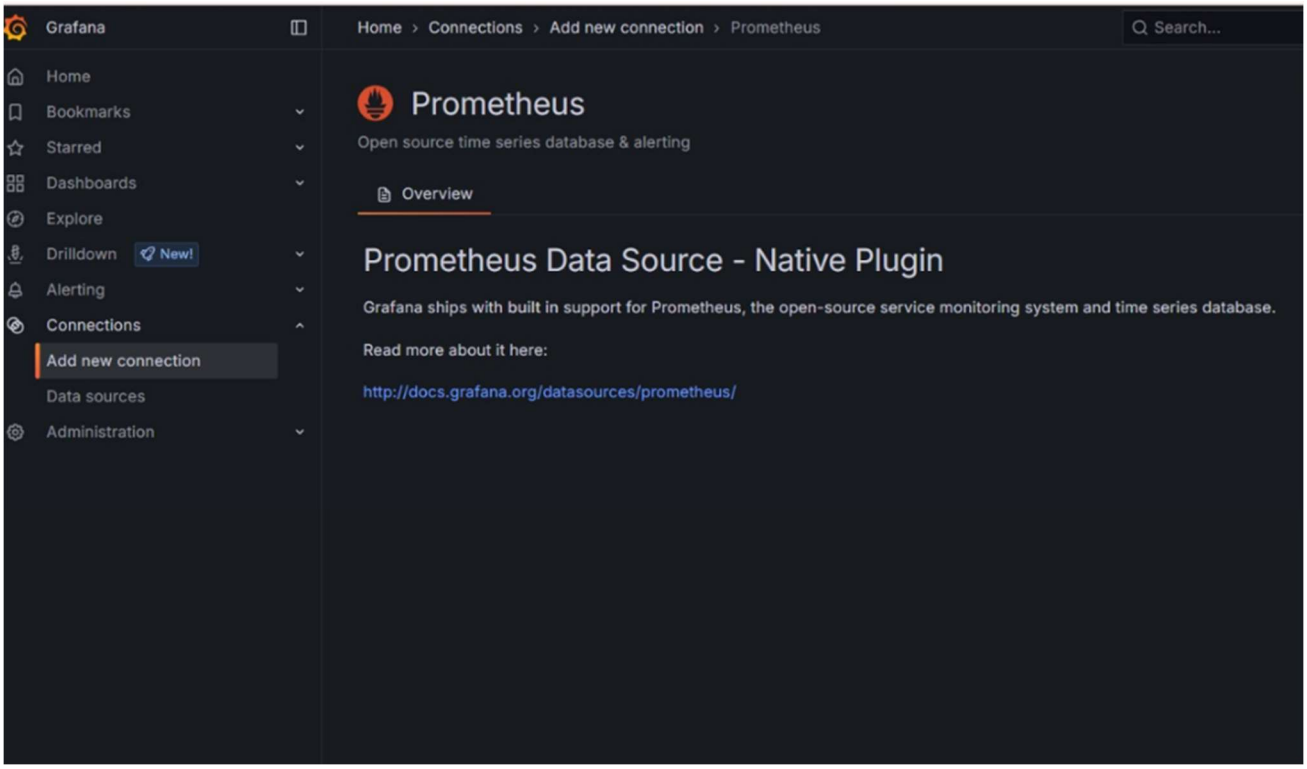


Dashboard configuration displays real-time monitoring of substation loads and charging request distributions.

Access Prometheus Interface at http ://localhost:9090 → Metrics Collection



Prometheus collects and stores time-series data from all system components for analysis and alerting.



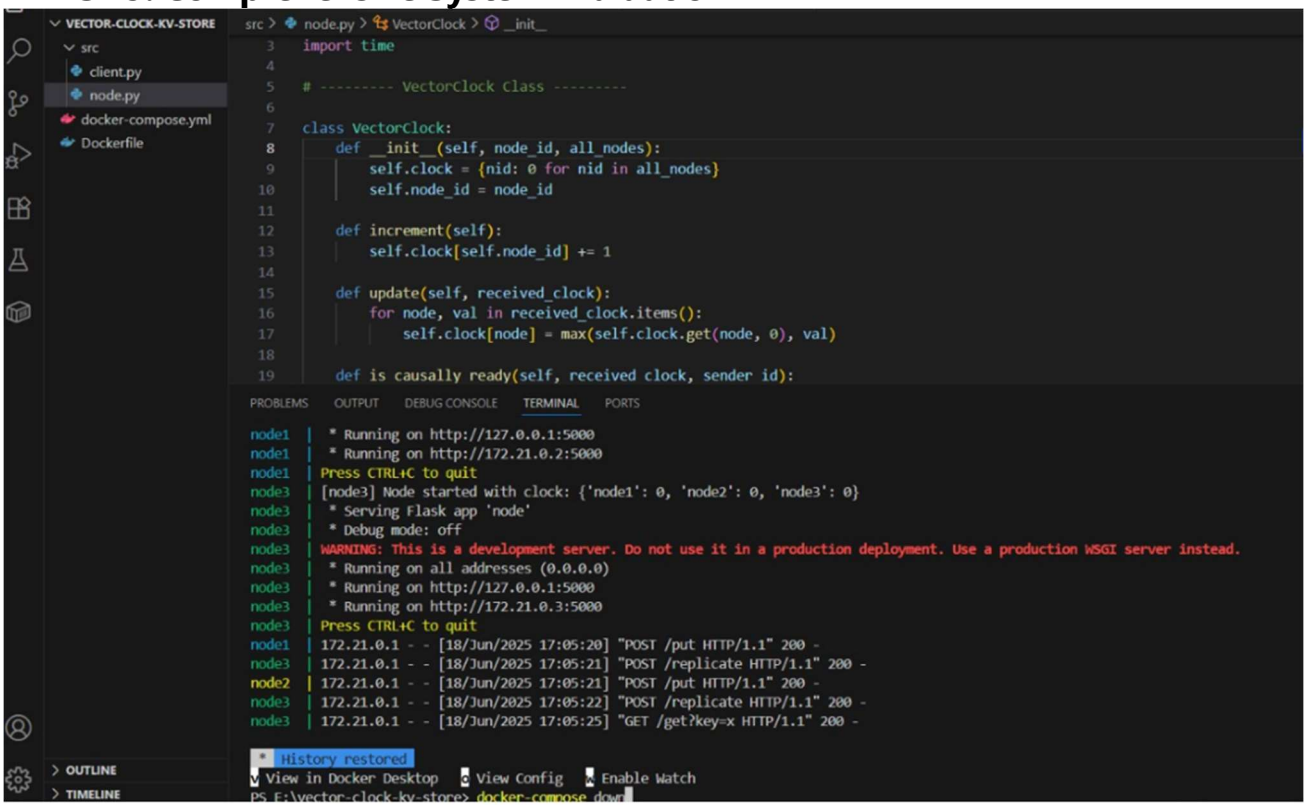
The metrics interface shows detailed performance indicators and system health status across all services.

Prometheus Target Configuration at http ://localhost:9090

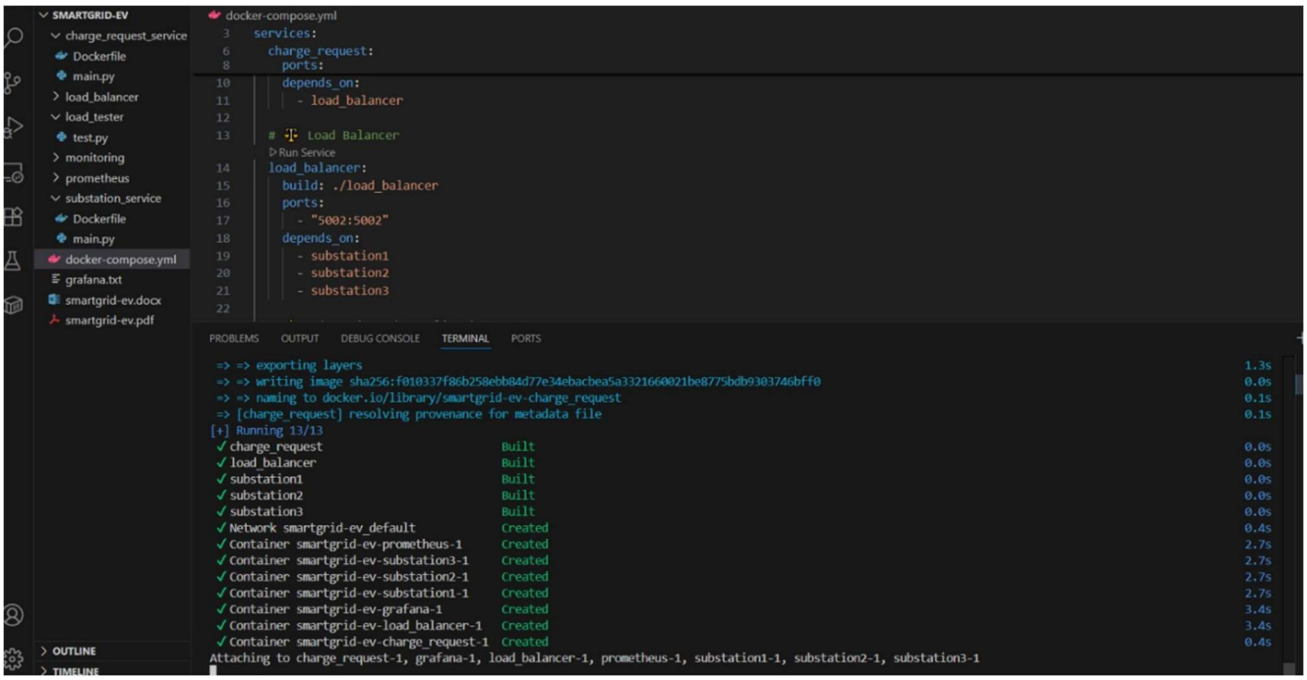
Target configuration ensures proper metric collection from all substation endpoints.

Load Balancer API Access: localhost:5001

PHASE 9: Comprehensive System Evaluation



Load testing results demonstrate the system's capability to handle concurrent charging requests while maintaining optimal distribution.



Performance analysis shows effective load balancing across multiple substations during peak demand scenarios.

System Architecture Overview

The implemented smart grid system consists of interconnected microservices designed for scalability and reliability:

Core Components:

- **Charging Request Gateway:** Public API endpoint for EV charging requests
- **Intelligent Load Balancer:** Dynamic routing engine with real-time decision making
- **Substation Simulators:** Multiple charging infrastructure endpoints with load monitoring
- **Observability Stack:** Prometheus metrics collection and Grafana visualization
- **Testing Framework:** Comprehensive load testing and performance validation

Key Technical Features:

- Real-time load assessment and distribution
- Automatic failover and recovery mechanisms • Comprehensive monitoring and alerting
- Container-based deployment and orchestration • Scalable microservices architecture

Performance Validation:

The system successfully demonstrates intelligent load distribution, preventing substation overload while maintaining optimal charging efficiency. Grafana dashboards provide clear visualization of load patterns, request routing effectiveness, and overall system health during various demand scenarios.

Implementation Results

The smart grid charging management system effectively handles dynamic load balancing requirements while providing comprehensive observability into system operations. The containerized architecture ensures easy deployment and scalability for real-world applications.