

□ Standard Operating Procedure (SOP)

🔗 Project Title: Full Stack DevOps – Cloud Deployment, Monitoring & CI/CD Automation

🏆 1 – Cloud Deployment

🎯 Objective

Deploy the full stack (frontend + backend + MySQL) on an AWS EC2 Ubuntu instance using Docker and Docker Compose with persistent MySQL storage.

□ Steps Followed

➤ Clone the Repository

```
git clone https://github.com/DhruvShah0612/fusionpact-devops-challenge.git  
cd fusionpact-devops-challenge
```

➤ Build and Run Containers

```
sudo docker-compose up -d --build
```

➤ Verify Running Containers

```
sudo docker ps
```

Frontend Dockerfile

```
FROM nginx:alpine  
  
# Remove default nginx html  
RUN rm -rf /usr/share/nginx/html/*  
  
# Copy everything in current folder (frontend/) to nginx html folder  
COPY . /usr/share/nginx/html/  
  
# Rename the file to index.html safely  
RUN [ -f /usr/share/nginx/html/Devops_Intern.html ] && mv /usr/share/nginx/html/Devops_Intern.html /usr/share/nginx/html/index.html || true  
  
EXPOSE 80  
  
CMD ["nginx", "-g", "daemon off;"]
```

Backend Dockerfile

```
FROM python:3.11-slim  
  
WORKDIR /app  
  
RUN apt-get update && apt-get install -y default-libmysqlclient-dev build-essential  
  
# Copy requirements (no 'backend/' here)  
COPY requirements.txt .  
  
RUN pip install --no-cache-dir -r requirements.txt  
  
# Copy app folder (no 'backend/' here)  
COPY app ./app  
  
EXPOSE 8000  
  
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Docker-compose.yml file

```
version: '3.9'

services:
  # -----
  # MySQL Database
  # -----
  db:
    image: mysql:8.0
    container_name: mysql-db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: fusionpact
      MYSQL_USER: fusionuser
      MYSQL_PASSWORD: fusionpass
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - fusionnet

  # -----
  # Backend - FastAPI
  # -----
  backend:
    build: ./backend
    container_name: fusion-backend
    restart: always
    environment:
      DATABASE_HOST: db
      DATABASE_USER: fusionuser
      DATABASE_PASSWORD: fusionpass
      DATABASE_NAME: fusionpact
      DATABASE_PORT: 3306
    depends_on:
      - db
    ports:
      - "8000:8000"
    networks:
      - fusionnet

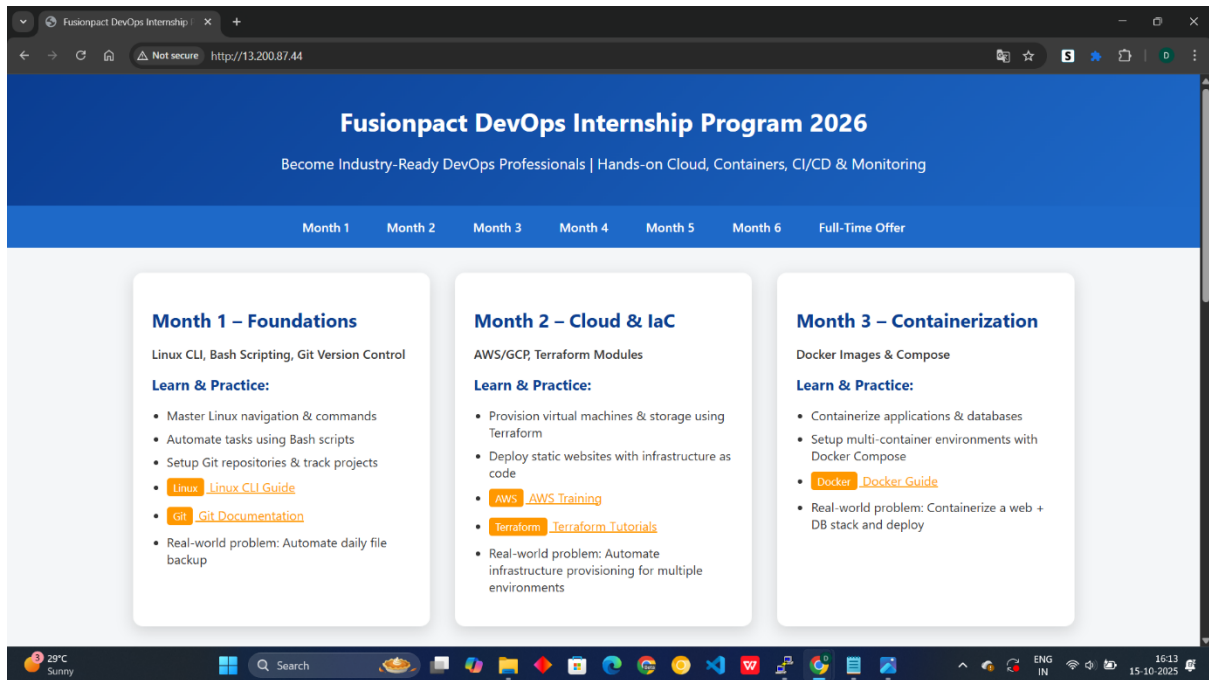
  # -----
  # Frontend - React/Vite
  # -----
  frontend:
    build: ./frontend
    container_name: fusion-frontend
    restart: always
    ports:
      - "80:80"
    depends_on:
      - backend
    networks:
      - fusionnet

# -----
# Named Volumes & Network
# -----
volumes:
  mysql_data:

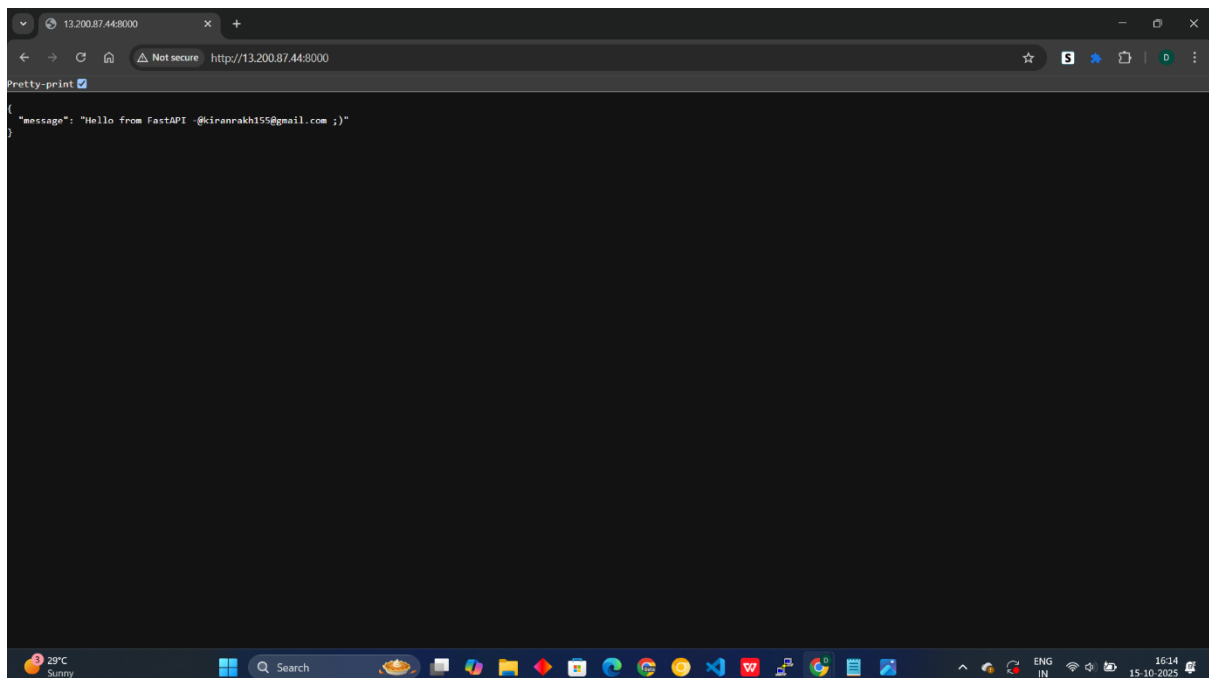
networks:
  fusionnet:
    driver: bridge
```

➤ **Access Services:**

Frontend → `http://<EC2-IP>/`



Backend → `http://<EC2-IP>:8000/`



MySQL Data Persistence Test

```
CREATE TABLE test_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50));
```

```
INSERT INTO test_table (name) VALUES ('Level1 Test');
```

```
SELECT * FROM test_table;
```

```
root@ip-172-31-33-166:/home/ubuntu/fusionpact-devops-challenge# docker exec -it mysql-db mysql
mysql -u fusionuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.43 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE fusionpact;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_fusionpact |
+-----+
| test_table           |
+-----+
1 row in set (0.01 sec)

mysql>
```

🏆 2 – Monitoring & Observability

🎯 Objective

Implement observability using Prometheus, Node Exporter, and Grafana for both application and infrastructure metrics.

❏ prometheus.yml

```
global:
  scrape_interval: 15s

scrape_configs:
  # Prometheus itself
  - job_name: 'prometheus'
    static_configs:
      - targets: ['prometheus:9090']

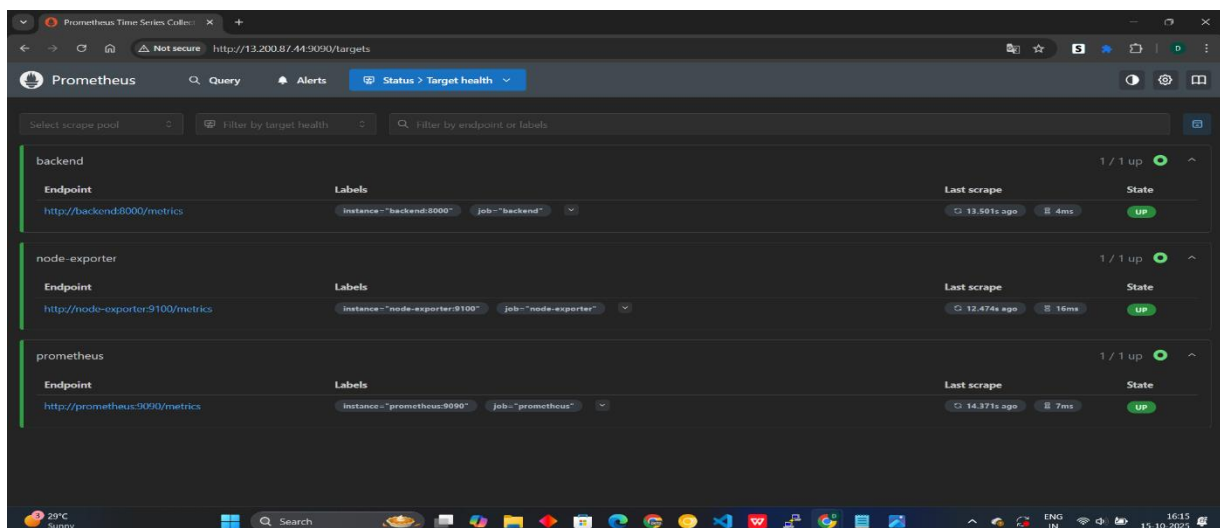
  # FastAPI backend
  - job_name: 'backend'
    static_configs:
      - targets: ['backend:8000']

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['node-exporter:9100']
```

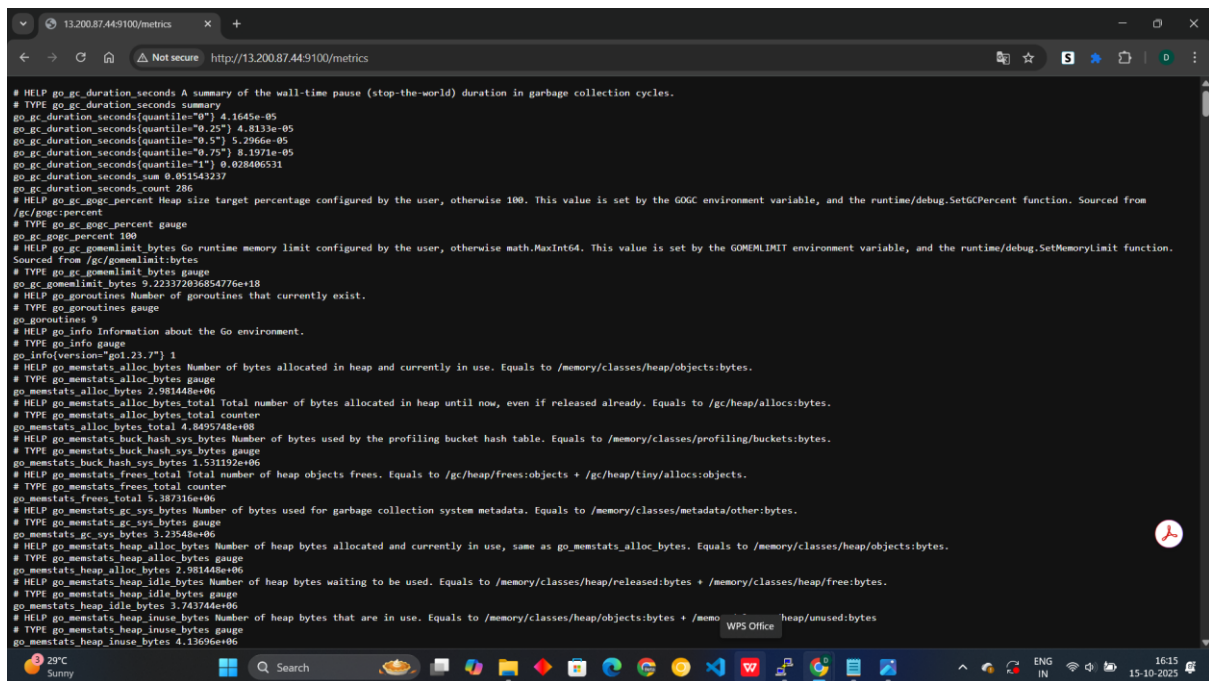
⚙️ Setup Commands

`docker-compose -f docker-compose.yml up -d`

Prometheus → `http://<EC2-IP>:9090`



Prometheus targets page (showing backend + node_exporter)

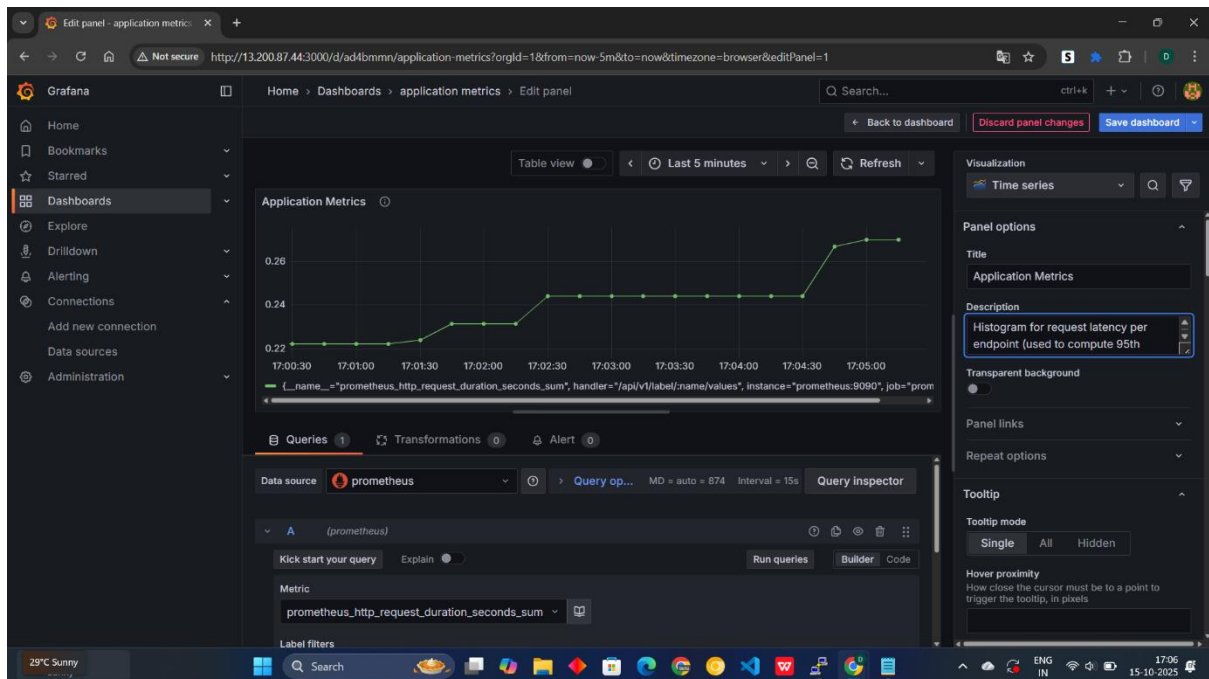
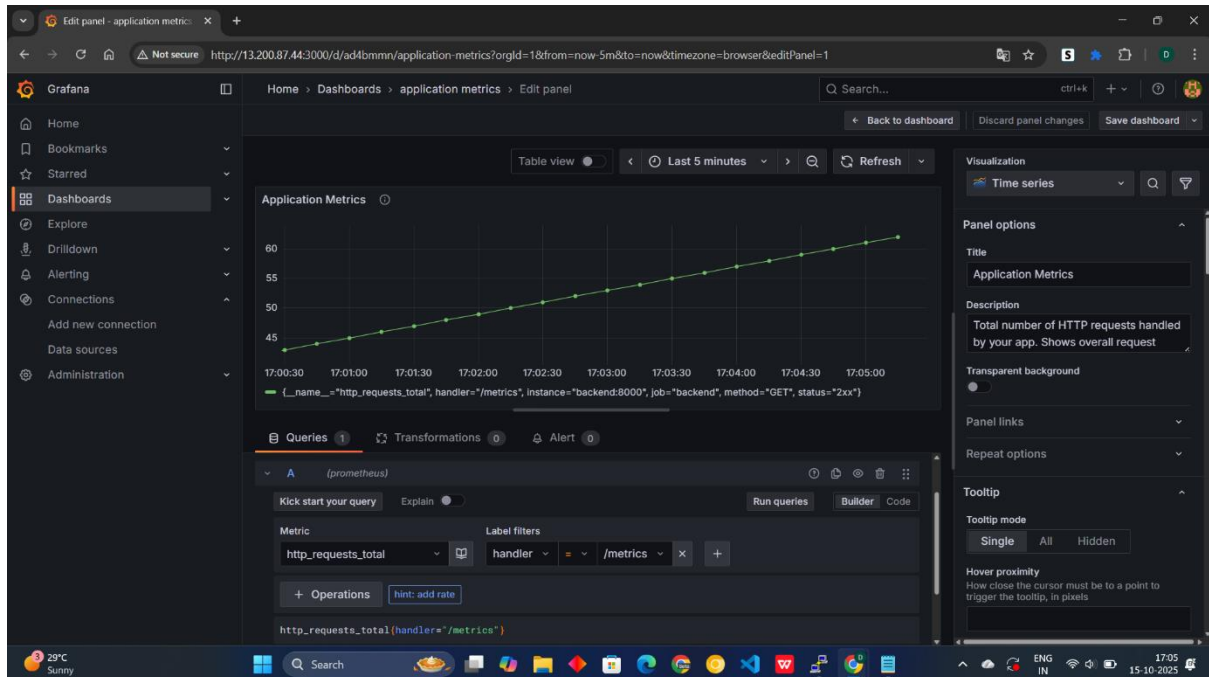


```
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 4.1645e-05
go_gc_duration_seconds{quantile="0.25"} 4.8133e-05
go_gc_duration_seconds{quantile="0.5"} 5.2966e-05
go_gc_duration_seconds{quantile="0.75"} 8.1971e-05
go_gc_duration_seconds{quantile="1"} 0.028406531
go_gc_duration_seconds_sum 0.051543237
go_gc_duration_seconds_count 286
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environment variable, and the runtime/debug.SetGCPercent function. Sourced from /gc/gogc:percent
# TYPE go_gc_gogc_percent gauge
go_gc_gogc_percent 100
# HELP go_gc_gomemlimit_bytes Go runtime memory limit configured by the user, otherwise math.MaxInt64. This value is set by the GOMEMLIMIT environment variable, and the runtime/debug.SetMemoryLimit function. Sourced from /gc/gomemlimit:bytes
# TYPE go_gc_gomemlimit_bytes gauge
go_gc_gomemlimit_bytes 9.223372036854776e+18
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 9
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version:"go1.23.7"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated in heap and currently in use. Equals to /memory/classes/heap/objects:bytes.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.981448e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated in heap until now, even if released already. Equals to /gc/heap/allocs:bytes.
# TYPE go_memstats_alloc_bytes_total gauge
go_memstats_alloc_bytes_total 4.8495748e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table. Equals to /memory/classes/profiling/buckets:bytes.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.531192e+06
# HELP go_memstats_frees_total Total number of heap objects freed. Equals to /gc/heap/frees:objects + /gc/heap/tiny/allocs:objects.
# TYPE go_memstats_frees_total gauge
go_memstats_frees_total 5.387316e+06
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata. Equals to /memory/classes/metadata/other:bytes.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 3.23548e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and currently in use, same as go_memstats_alloc_bytes. Equals to /memory/classes/heap/objects:bytes.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.981448e+06
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used. Equals to /memory/classes/heap/released:bytes + /memory/classes/heap/free:bytes.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 3.74374e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use. Equals to /memory/classes/heap/objects:bytes + /memory/classes/heap/unused:bytes
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 4.13696e+06
```

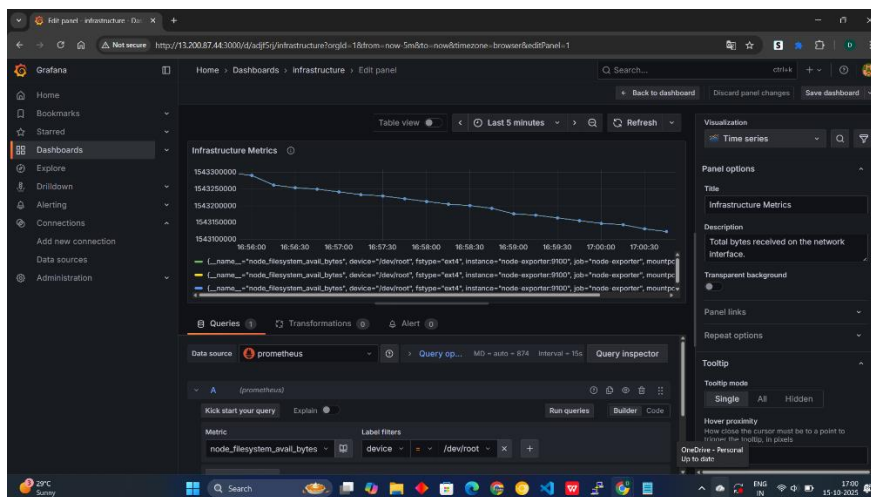
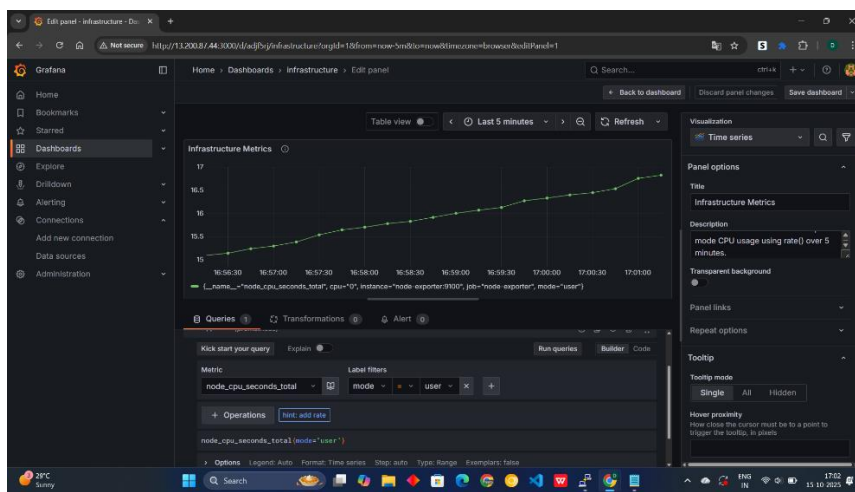
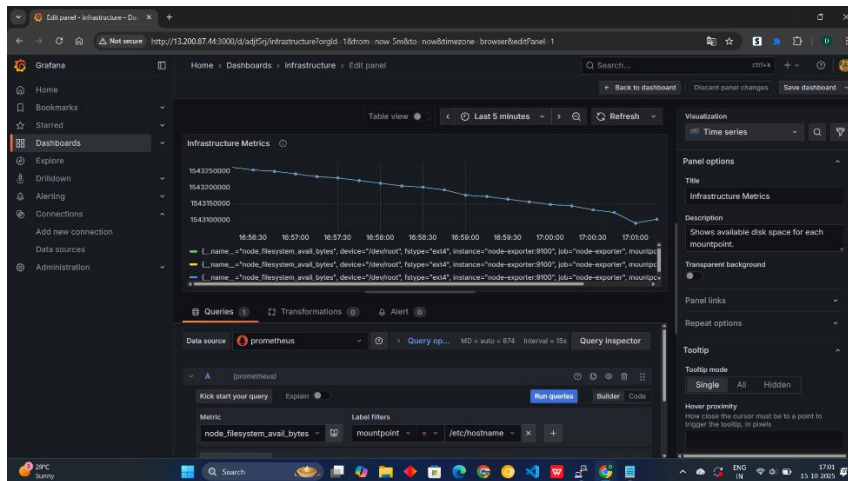
- **Grafana** → <http://<EC2-IP>:3000>
 - Username: admin
 - Password: admin
- **Add Prometheus as data source:**
<http://prometheus:9090>

Dashboards Created

Application Metrics



Infrastructure Metrics



3 – CI/CD Automation

Objective

Automate the build, push, and deploy stages using Jenkins with Docker integration on AWS EC2.

Final Jenkinsfile

```
pipeline {
  agent any
  environment {
    DOCKER_HUB_REPO = 'dhruvshah0612/fusionpact'
  }

  stages {
    stage('Checkout Code') {
      steps {
        git branch: 'main', credentialsId: 'github-https', url: 'https://github.com/DhruvShah0612/fusionpact-devops-challenge.git'
      }
    }

    stage('Docker Test') {
      steps {
        sh 'docker --version'
      }
    }
  }

  stage('Build Docker Image') {
    steps {
      sh 'docker build -t $DOCKER_HUB_REPO:${BUILD_NUMBER} .'
    }
  }

  stage('Push to Docker Hub') {
    steps {
      withCredentials([usernamePassword(credentialsId: 'dockerhub-creds', usernameVariable: 'USER', passwordVariable: 'PASS')]) {
        sh '''
        echo "$PASS" | docker login -u "$USER" --password-stdin
        docker push $DOCKER_HUB_REPO:${BUILD_NUMBER}
        '''
      }
    }
  }

  stage('Deploy') {
    steps {
      sh '''
      docker stop fusionpact-app || true
      docker rm fusionpact-app || true
      docker run -d --name fusionpact-app -p 8080:80 $DOCKER_HUB_REPO:${BUILD_NUMBER}
      '''
    }
  }
}

post {
  always {
    echo 'Cleaning up dangling images...'
    sh 'docker image prune -f || true'
  }
  success {
    echo '✅ Pipeline executed successfully!'
  }
  failure {
    echo '❌ Pipeline failed! Please check logs.'
  }
}
```

Jenkins Setup Command

```
docker run -d \
  --name jenkins-docker \
  -p 8080:8080 -p 50000:50000 \
  -v /home/ubuntu/jenkins_home:/var/jenkins_home \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -e DOCKER_GROUP_ID=$(getent group docker | cut -d: -f3) \
  jenkins/jenkins:lts
```

Jenkins Credentials

ID	Type	Description
github-https	GitHub Token	For code checkout
dockerhub-creds	Docker Hub Login	For image push
ec2-ssh-key	SSH Key (optional)	For remote deployment

Access

- Jenkins Web UI: <http://:8080>

This setup ensures a fully automated CI/CD pipeline from GitHub → Jenkins → Docker → EC2 deployment.