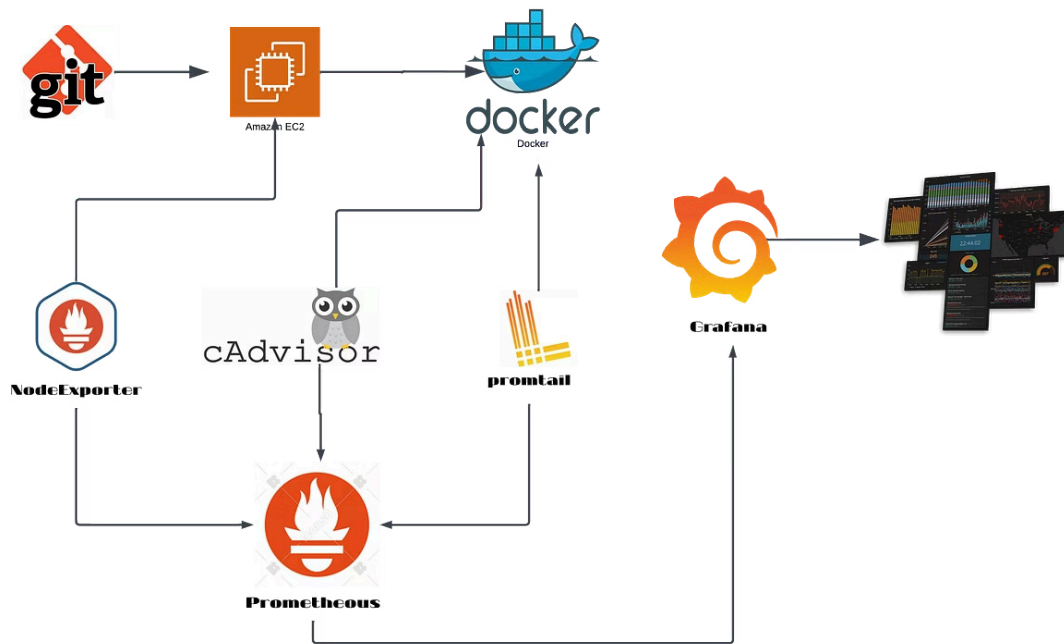


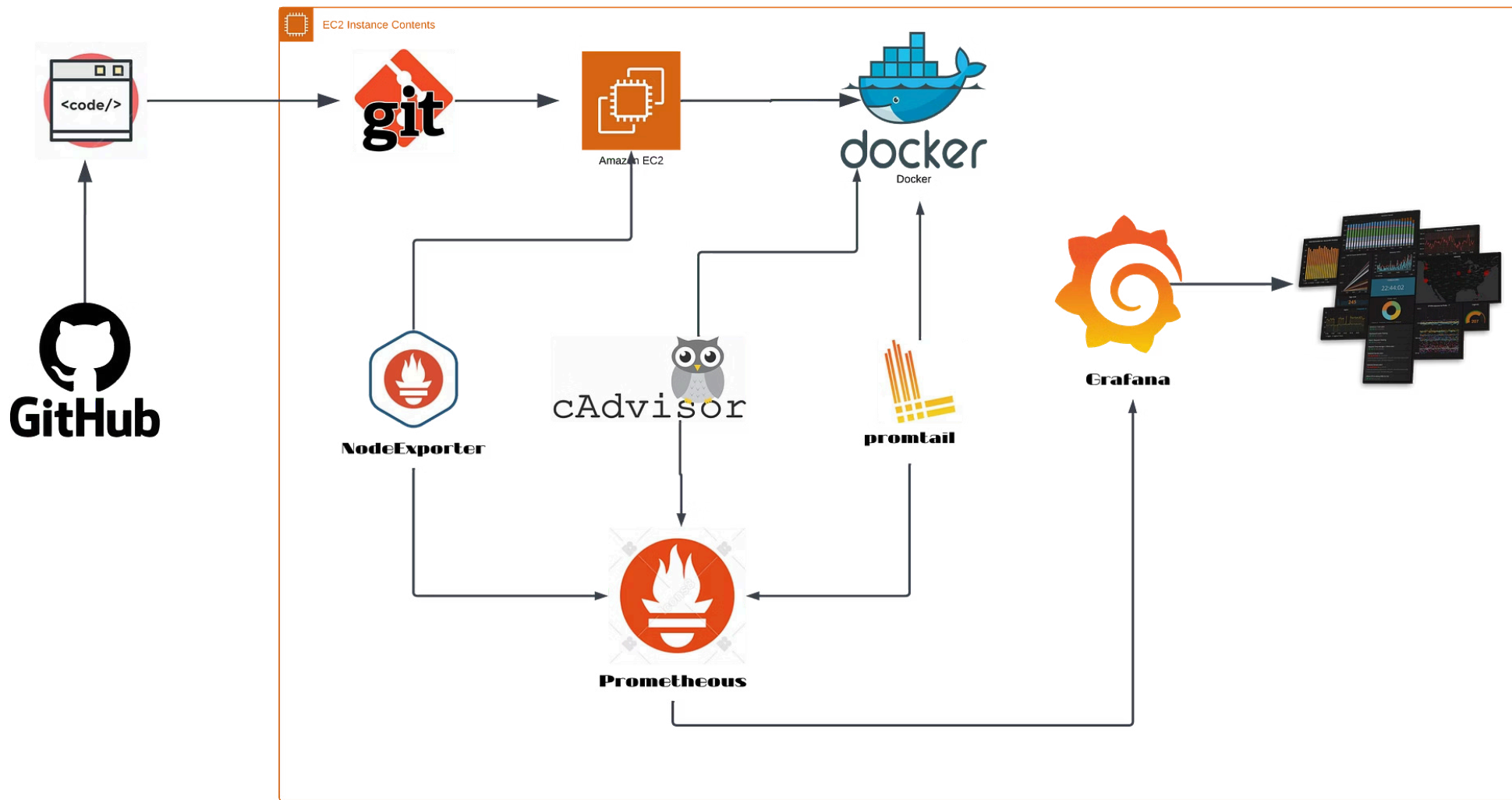
Assignment Building a Powerful Observability Stack for Dockerized Applications: A Complete Guide



- **Docker & Docker Compose:** Containerization and orchestration.
- **Prometheus:** Metrics collection and monitoring.
- **Grafana:** Data visualization and dashboard creation.
- **cAdvisor:** Container resource monitoring.
- **Node Exporter:** Hardware and OS metrics exporter.
- **Notes App:** A custom service to demonstrate monitoring.

Author : Abhisheke Kumar

Date : 27-06-2025



Assignment Project Objective:

The goal is to set up a complete **Observability Stack** for Dockerized applications using the following tools:

- **Grafana** for data visualization.
- **Prometheus** as a time-series database and metric collector.
- **Node Exporter** for system-level metrics collection.
- **cAdvisor** for container resource usage monitoring.
- **Loki** for log aggregation.
- **Promtail** for log shipping.

This integrated solution will provide comprehensive monitoring of system and container metrics, application logs, and data visualization in Grafana.

Step-by-Step Implementation:

Step 1: Create a Virtual Server

- Provision a virtual server with:
 - **4GB RAM**
 - **2 vCPUs**

Step 2: Install Docker and Docker Compose

1. Update and upgrade system packages: `apt-get update apt-get upgrade`
2. Install Docker: `apt-get install docker.io`
3. Add your user to the Docker group: `usermod -aG docker $USER && newgrp docker`
4. Install Docker Compose: `apt-get install docker-compose`
5. Clone the repository: `git clone https://github.com/cdrAbhi/Obserbility-Prometheus-and-Grafana-.git`

Step 3: Build the Application Docker Image

- Create a **Dockerfile** to build the image for your application:

```
# Build Stage
FROM python:3.9 AS builder
WORKDIR /app
COPY requirements.txt .
RUN apt-get update && apt-get install -y gcc libffi-dev libssl-dev python3-dev \
    && python -m venv /venv \
    && . /venv/bin/activate \
    && pip install --no-cache-dir -r requirements.txt \
    && apt-get remove --purge -y gcc libffi-dev libssl-dev python3-dev \
    && apt-get autoremove -y \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

# Final Stage
FROM python:3.9-alpine
WORKDIR /app
COPY --from=builder /venv /venv
COPY . .
ENV PATH="/venv/bin:$PATH"
EXPOSE 8000
CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

Step 4: Create a Docker Compose Configuration

- Create a `docker-compose.yml` file to orchestrate the application and observability tools:

```
version: '3.8'
volumes:
  notesapp: {}
  prometheus-vol: {}
  grafana-vol: {}

networks:
  web-network:
    driver: bridge

services:
  notes-web:
    build:
      context: ./notes-app
    container_name: "notes-app"
    ports:
      - "8000:8000"
    volumes:
      - "notesapp"
    networks:
      - web-network
```

Step 5: Set Up Prometheus and Grafana

1. Download the Prometheus configuration file: `wget`

```
https://raw.githubusercontent.com/prometheus/prometheus/main/documentation/examples/prometheus.yml
```

2. Add Prometheus to Docker Compose:

```
prometheus:
  image: prom/prometheus:latest
  container_name: prometheus
  ports:
    - "9090:9090"
  volumes:
    - prometheus-vol:/prometheus
    - ./prometheus.yml:/etc/prometheus/prometheus.yml:ro
  networks:
    - web-network
  restart: unless-stopped
  depends_on:
    - node-exporter
    - cadvisor
```

3. Add Grafana to Docker Compose:

```
grafana:
  image: grafana/grafana-enterprise
  container_name: grafana
  ports:
    - "3000:3000"
  volumes:
    - grafana-vol:/var/lib/grafana
    - ./data/grafana/provisioning/datasources:/etc/grafana/provisioning/datasources
    - ./data/grafana/provisioning/dashboards:/etc/grafana/provisioning/dashboards
  networks:
    - web-network
  restart: unless-stopped
```

Step 6: Set Up Node Exporter and cAdvisor

- **Node Exporter** configuration:

```
node-exporter:
  image: prom/node-exporter:latest
  container_name: node-exporter
  ports:
    - "9100:9100"
  networks:
    - web-network
  restart: unless-stopped
```

- **cAdvisor** configuration:

```
cadvisor:
  image: gcr.io/cadvisor/cadvisor:latest
  container_name: cadvisor
  ports:
    - "8080:8080"
  volumes:
    - /:/rootfs:ro
    - /var/run:/var/run:rw
    - /sys:/sys:ro
    - /var/lib/docker:/var/lib/docker:ro
  networks:
    - web-network
  restart: unless-stopped
```

- Add Node Exporter and cAdvisor targets to `prometheus.yml`:

```
scrape_configs:
  - job_name: "Docker-CaAdvisor"
    static_configs:
      - targets: ["cadvisor:8080"]
  - job_name: "Node-exporter"
    static_configs:
      - targets: ["node-exporter:9100"]
```


Step 7: Set Up Loki and Promtail

- **Loki** configuration in Docker Compose:

```
loki:
  image: grafana/loki:latest
  container_name: loki
  ports:
    - "3100:3100"
  volumes:
    - ./loki-config.yml:/etc/loki/local-config.yml
  networks:
    - web-network
  restart: unless-stopped
```

- **Promtail** configuration in Docker Compose:

```
promtail:
  image: grafana/promtail:latest
  container_name: promtail
  volumes:
    - /var/log:/var/log
    - ./promtail-config.yaml:/etc/promtail/config.yml
  networks:
    - web-network
  restart: unless-stopped
```

- Add Loki target to `prometheus.yml`:

```
scrape_configs:
  - job_name: "Loki"
    static_configs:
      - targets: ["loki:3100"]
```

Step 8: Start the Observability Stack

- Run Docker Compose: `docker-compose up -d`

Step 9: Configure Grafana

Access Grafana at `http://<your-server-ip>:3000` and set up the following:

1. **Add Datasources** for Prometheus and Loki.
2. **Create Dashboards** for visualization.
 - You can either:
 - Manually create dashboards.
 - Import pre-configured dashboard templates.

```
version: '3.8'

volumes:
  notesapp: {}
  prometheus-vol: {}
  grafana-vol: {}

networks:
  web-network:
    driver: bridge

services:
  notes-web:
    build:
      context: ./notes-app
    container_name: "notes-app"
    ports:
      - "8000:8000"
    volumes:
      - "notesapp"
    networks:
      - web-network
  grafana:
    image: grafana/grafana-enterprise
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana-vol:/var/lib/grafana
      - ./data/grafana/provisioning/datasources:/etc/grafana/provisioning/datasources
      - ./data/grafana/provisioning/dashboards:/etc/grafana/provisioning/dashboards
    networks:
      - web-network
    restart: unless-stopped

  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    ports:
      - "9090:9090"
    volumes:
      - prometheus-vol:/prometheus      #This line mounts the prometheus.yml file from your local filesystem
      - ./prometheus.yml:/etc/prometheus/prometheus.yml:ro #This line mounts the prometheus.yml file from your local
filesystem (or project directory) into the /etc/prometheus/prometheus.yml path inside the container.Prometheus requires a
configuration file to know which targets to scrape for metrics
    networks:
      - web-network
    restart: unless-stopped
    depends_on:
      - node-exporter
      - cadvisor

  node-exporter:
    image: prom/node-exporter:latest
    container_name: node-exporter
    restart: unless-stopped
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
    command:
      - '--path.procfs=/host/proc'
      - '--path.rootfs=/rootfs'
      - '--path.sysfs=/host/sys'
      - '--collector.filesystem.mount-points-exclude=^/(sys|proc|dev|host|etc)($$|/)'
    expose:
      - 9100
    ports:
      - "9100:9100"
    networks:
      - web-network

  cadvisor:
    image: gcr.io/cadvisor/cadvisor:latest
    container_name: cadvisor
    ports:
      - "8080:8080"
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker:/var/lib/docker:ro
    networks:
      - web-network
    restart: unless-stopped

  loki:
    image: grafana/loki:latest
    container_name: loki
    ports:
      - "3100:3100"
    volumes:
      - ./loki-config.yml:/etc/loki/local-config.yml
    networks:
      - web-network
    restart: unless-stopped

  promtail:
    image: grafana/promtail:latest
    container_name: promtail
    volumes:
      - /var/log:/var/log
      - ./promtail-config.yaml:/etc/promtail/config.yml
    networks:
      - web-network
    restart: unless-stopped
```

observed_timestamp_rfc3339	severity_text	instrumentation_scope	body (excerpt)
2025-06-27T10:15:26.123+05:30	INFO	noteapp.auth	User login successful — userId='u_8732' ip='192.168.0.18' auth_method='JWT'
2025-06-27T10:15:27.004+05:30	INFO	noteapp.api	Note created — noteId='n_5471' userId='u_8732' title='Meeting Notes' created_at='2025-06-27T10:15:27Z'
2025-06-27T10:15:28.219+05:30	DEBUG	noteapp.db	Executed SQL: SELECT * FROM notes WHERE user_id='u_8732' — duration=23ms
2025-06-27T10:15:29.011+05:30	INFO	noteapp.analytics	Note sync event — noteId='n_5471' syncType='cloud_backup' syncStatus='success'
2025-06-27T10:15:30.544+05:30	WARN	noteapp.api	Slow response detected — endpoint='/api/notes/share' method='POST' latency=841ms userId='u_2378'
2025-06-27T10:15:31.882+05:30	ERROR	noteapp.notifications	Push notification failed — deviceToken='xyz123' error='Device unreachable' event='reminder:note_due'

Logs

2025-06-27 16:09:52 {"body":"vYTC@XcpF85J@hV5zPjha@2QtvfsfQlleTXrn/3jo80=1747224592", "attributes" : {"obs_ts" : 1747224602}}

2025-06-27 16:09:54 {"body":"kfaĵ/waHGbt+v20poLkLqtDfvYAOIMueiVvYPUuiE9Z=1747224594", "attributes" : {"obs_ts" : 1747224594}}

2025-06-27 16:09:55 {"body":"RY6qHtVc1NTFHZQ7nDakBjhWXCfKEcI1j8TFh/uqlQ=1747224595", "attributes" : {"obs_ts" : 1747224595}}

2025-06-27 16:09:56 {"body":"kxmTDHEMMsxe23Vq2a0fc9EPtRFqZb10is+DRjzQUC8=1747224596", "attributes" : {"obs_ts" : 1747224596}}

2025-06-27 16:09:57 {"body":"f92SuVaNW9+u2MpN8RZk8dDMXG9te1OrN@A6RBFKU5S=1747224597", "attributes" : {"obs_ts" : 1747224607}}

2025-06-27 16:09:58 {"body":"D4qLMApxeDuJvZRaUOVAdhQfR2W8CLBbRqPzEyDeAm4=1747224598", "attributes" : {"obs_ts" : 1747224598}}

2025-06-27 16:10:00 {"body":"VzzMo6Y7VxBdawfXSd5dYIkY2+Nps63kZEstj1uWJ4Q=1747224600", "attributes" : {"obs_ts" : 1747224600}}

2025-06-27 16:10:01 {"body":"Pwp7RWB8ICwB5y305/RelbwfN6LWA8YKF5krZR2MtjA=1747224601", "attributes" : {"obs_ts" : 1747224601}}

2025-06-27 16:10:02 {"body":"IY2QSEjdatu7pD3/9DM2sJS@mvCwwUpew3VGmV/dIfc=1747224602", "attributes" : {"obs_ts" : 1747224612}}

2025-06-27 16:10:03 {"body":"HwlgsJxRWE7VIA8sg+eMZte7YsuuX7JkBxquZTxy5ps=1747224603", "attributes" : {"obs_ts" : 1747224603}}

2025-06-27 16:10:05 {"body":"nUf/wfNndDtaEoder+jtKjBnxkkxTGawGrcefXXHbza=1747224605", "attributes" : {"obs_ts" : 1747224605}}

2025-06-27 16:10:06 {"body":"thlLuFaODbklie8@xxZ8HVDUNGBinCSilUtlEnplKXg=1747224606", "attributes" : {"obs_ts" : 1747224606}}

2025-06-27 16:10:07 {"body":"T4xCXjYogN8Rc90c28IvEixXKgct6vv19VTu2fuQFC4=1747224607", "attributes" : {"obs_ts" : 1747224617}}

2025-06-27 16:10:08 {"body":"Q60e2T55wVUvwLBdMR4ityX+3mWABBXM2imjWoeulH@=1747224608", "attributes" : {"obs_ts" : 1747224608}}

2025-06-27 16:10:09 {"body":"RRFDesJ8A+ZdnxJZoyLuf6TAJWBHeKPB/GbNVIryhUQ=1747224609", "attributes" : {"obs_ts" : 1747224609}}

2025-06-27 16:10:11 {"body":"ByW+N/S4Mf6CGnyw354LmjYD+1v4TuCZfg+uy@keYiQ=1747224611", "attributes" : {"obs_ts" : 1747224611}}

2025-06-27 16:10:12 {"body":"vw104KfQV100ufOpaLUQm3BCcwu3ujawhj78gUZmo3A=1747224612", "attributes" : {"obs_ts" : 1747224622}}

2025-06-27 16:10:13 {"body":"+x9161IeWFyk5q9xFVUCHdXSz2qSIHPCkO20qeowHKQ=1747224613", "attributes" : {"obs_ts" : 1747224613}}

2025-06-27 16:10:14 {"body":"yBrvv2KKfbF@4Q3aU6fMlyhWmM1qY /altkshNvYY3I=1747224614", "attributes" : {"obs_ts" : 1747224614}}

2025-06-27 16:10:16 {"body":"wkWJiHtdtnG1h17g+1MjNkfxzE2//p80Cq543w@Etpo=1747224616", "attributes" : {"obs_ts" : 1747224616}}

2025-06-27 16:10:17 {"body":"EwkGIXJDyOWMU6H8po3CD6 fwW10z65Xu3+gu6dBPjB=1747224617", "attributes" : {"obs_ts" : 1747224627}}

2025-06-27 16:10:18 {"body":"c1M8mcqlaPyzoe6ch8fe2wx4Ybknlijth+f/BxTVXHI=1747224618", "attributes" : {"obs_ts" : 1747224618}}

2025-06-27 16:10:19 {"body":"ZjtjJ47Xmyp7RVrIwRSHNqcbHh8s1CcEx6ke6FSHyM4=1747224619", "attributes" : {"obs_ts" : 1747224619}}

2025-06-27 16:10:21 {"body":"B1QbkgSpTSEG@PyeL1WSkTmq5T2P64+mC8XDFoYD+g=1747224621", "attributes" : {"obs_ts" : 1747224621}}

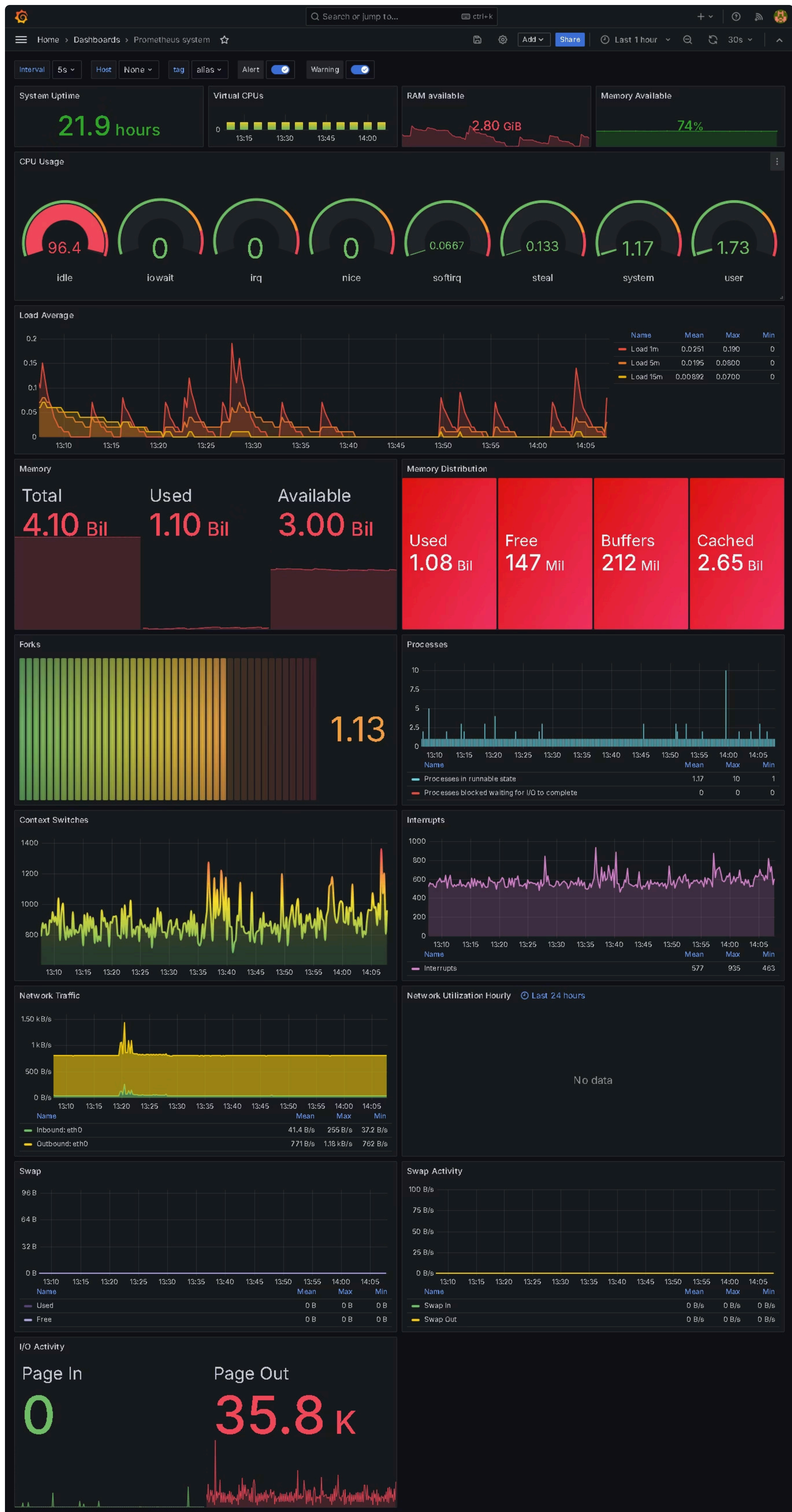
2025-06-27 16:10:23 {"body":"9KvnIdnLmqGZfeyqqSnw@/cdueh1sYdQ/1+qMmXf1xM=1747224623", "attributes" : {"obs_ts" : 1747224623}}

2025-06-27 16:10:24 {"body":"u8tZzaXAiJQnmWhhYh8@RRZNPazW/K2rd/bhVLpzV7Y=1747224624", "attributes" : {"obs_ts" : 1747224625}}

2025-06-27 16:10:26 {"body":"9hbuo5w1NDdX7iMAD9bwaZPS1pW@4TAKDXMJDo/E9E@=1747224626", "attributes" : {"obs_ts" : 1747224626}}

2025-06-27 16:10:28 {"body":"H7B5c2i76qLYxz9HJLASLnOfmWE1Jc3BC1djvGMNPjg=1747224628", "attributes" : {"obs_ts" : 1747224628}}

Result :



Thank you ! ❤️

