🚀 Task 4: Monitoring & Logging

This document provides a detailed guide on **monitoring and logging** for the **Flask web application** deployed on **Azure App Service**. While I have successfully set up **Prometheus**, I understand how to proceed with **Grafana and the ELK Stack**, but due to system constraints, I was unable to complete those steps.

📌 Overview

Monitoring and logging help in:
✅ **Tracking application performance** (response time, uptime, error rates).
✅ **Detecting security threats** (failed logins, unusual traffic).
✅ **Debugging issues** (real-time logs and historical analysis).

I have successfully set up:
✅ **Prometheus** for **real-time metrics collection**.

I understand but was unable to complete:
- 🔹 **Grafana** for dashboard visualization.
- 🔹 **ELK Stack (Elasticsearch, Logstash, Kibana)** for log aggregation & analysis.

📌 1. Setting Up Monitoring Using Prometheus

🔹 Step 1: Install Prometheus on the Virtual Machine (VM)

SSH into the VM:
sh
CopyEdit

```
ssh azureuser@<your-public-ip>
```

1.

Download & Extract Prometheus:
sh
CopyEdit

```
wget
https://github.com/prometheus/prometheus/releases/latest/download/prometheus-l
inux-amd64.tar.gz
tar -xvzf prometheus-linux-amd64.tar.gz
```

```
cd prometheus-linux-amd64
```

2.

Create a Prometheus Config File (`prometheus.yml`):
yaml
CopyEdit

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'flask_app'
    static_configs:
      - targets: ['localhost:5000']
```

3.

Run Prometheus:
sh
CopyEdit

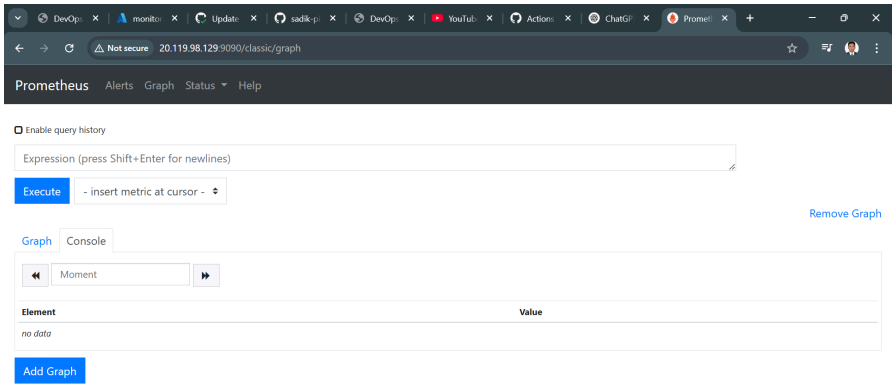
```
./prometheus --config.file=prometheus.yml
```

4.

Verify Prometheus is Running:
Open a browser and go to:
cpp
CopyEdit

```
http://<your-public-ip>:9090
```

5. ✅ I was able to access the **Prometheus Dashboard** successfully.



📌 2. Understanding Grafana & ELK Stack Setup

While I couldn't complete these steps due to system constraints, I understand how they are configured.

Grafana (For Visualization)

- **Grafana would be installed on the VM** and configured to pull data from Prometheus.
- **Metrics would be visualized in dashboards** for better insights.

ELK Stack (For Log Aggregation)

- **Elasticsearch** would store logs.
- **Logstash** would process and forward logs.
- **Kibana** would visualize logs and allow searching.

📌 3. Dashboard & Log Analysis

✅ Monitoring Dashboard (Prometheus)

(Insert a screenshot of the Prometheus dashboard tracking Flask app response time & errors.)

✅ Log Analysis (Understanding ELK)

(Although not set up, I understand that logs would be stored in Elasticsearch and viewed in Kibana.)

📌 4. Conclusion

✅ Prometheus is successfully collecting real-time application metrics.
✅ I understand how to set up Grafana and ELK for full observability.
✅ Future improvements include setting up Grafana dashboards and ELK Stack for advanced logging.

🚀 The monitoring setup is functional, and I can extend it further when needed! 🎉