***Weekly assessment***

**Setting up Prometheus and Grafana for monitoring and visualization involves several steps, including installation and configuration, metrics collection and visualization, and setting up alerts and notifications. Below, I'll explain these tasks with real-time configuration steps:**

**Step 1: Installation and Configuration**

**Prometheus Installation and Configuration:**

* Install Prometheus:

You can download and install Prometheus from the official website or use Docker for a quick setup.

docker run -d -p 9090:9090 --name prometheus prom/prometheus

* Configure Prometheus:

Create a prometheus.yml file to define your scrape targets (i.e., where Prometheus collects metrics from). Here's a simple example:

global:

scrape\_interval: 15s

scrape\_configs:

* + job\_name: 'my-app'

static\_configs:

* + targets: ['my-app-server:9090'] # Replace with your application's address
* Start Prometheus:

Start Prometheus with your configuration file.

* docker run -d -p 9090:9090 --name prometheus -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
* Grafana Installation and Configuration:

Install Grafana:

Similarly, you can download and install Grafana from the official website or use Docker.

docker run -d -p 3000:3000 --name grafana grafana/grafana

* Access Grafana UI:

Visit http://localhost:3000 and log in with the default credentials (admin/admin).

**Step 2: Metrics Collection and Visualization**

**Prometheus Metrics Collection:**

* In your application and infrastructure, expose metrics using Prometheus exposition format. You might use libraries like prometheus-client in your code.
* Prometheus will start collecting metrics from the specified scrape targets. You can verify this by visiting http://localhost:9090 and running queries.
* Grafana Visualization:

In Grafana, add Prometheus as a data source:

Go to Configuration > Data Sources > Add data source.

Choose Prometheus and configure the URL (http://localhost:9090).

Create a Dashboard:

* Go to Create > Dashboard.

Add panels to your dashboard and select Prometheus as the data source for each panel.

* Build Visualizations:

Customize panels to display the metrics you want to monitor (e.g., response time, CPU usage).

**Step 3: Alerts and Notifications**

**Prometheus Alerting Rules:**

* Create alerting rules in Prometheus. Edit the prometheus.yml file to define rules. Here's an example:

groups:

* + name: my-app-rules

rules:

* + alert: HighErrorRate

expr: rate(http\_requests\_total{job="my-app"}[5m]) > 0.1

for: 5m

labels:

severity: critical

annotations:

summary: "High error rate on my-app"

description: "High error rate detected for the past 5 minutes"

* Reload Prometheus to apply the new configuration:

docker kill -s HUP prometheus

* Alert Notifications:

In Grafana, go to Configuration > Notification channels and set up channels (e.g., email or Slack).

* Create alerting rules in Grafana:

Go to the dashboard you created.

Click on a panel, select "Edit," and go to the "Alert" tab.

Configure alert conditions, notifications, and thresholds.

Now, when a metric breaches the threshold, Prometheus will trigger an alert. Grafana will send notifications based on your alerting configuration.