Document Report: VM Resource Monitoring and Auto-Scaling

# Objective

The goal of this project is to monitor the resource usage (CPU, memory, disk, and network) of a local VM and set up an auto-scaling mechanism that triggers resource scaling to a public cloud (e.g., AWS, GCP, or Azure) when usage exceeds 75%. This document provides a step-by-step guide for setting up Grafana, Node Exporter, creating dashboards for monitoring, configuring alerts, and setting up auto-scaling.

# Step-by-Step Guide

## 1. Install Grafana

To install Grafana on your VM:  
  
1. Update and Install Grafana:  
 Use the following commands to install Grafana via the `pkg` command.  
  
 sudo pkg update  
 sudo pkg install -y grafana8  
  
2. Start Grafana:  
 Enable and start the Grafana service.  
  
 sudo service grafana start  
  
3. Access Grafana:  
 Open a web browser and navigate to http://<VM\_IP>:3000.  
 Username: admin  
 Password: admin (You will be prompted to change the password after the first login).

## 2. Install Node Exporter

1. Download and Install Node Exporter:  
 Install Node Exporter by following these steps:  
  
 wget https://github.com/prometheus/node\_exporter/releases/download/v1.3.1/node\_exporter-1.3.1.linux-amd64.tar.gz  
 tar -xvzf node\_exporter-1.3.1.linux-amd64.tar.gz  
 sudo mv node\_exporter-1.3.1.linux-amd64/node\_exporter /usr/local/bin/  
  
2. Run Node Exporter:  
 Start Node Exporter to expose system metrics for Prometheus scraping.  
  
 nohup /usr/local/bin/node\_exporter --web.disable-tls --web.enable-http2=false &  
  
 Verify by navigating to http://<VM\_IP>:9100/metrics.

## 3. Configure Prometheus in Grafana

1. Add Prometheus Data Source:  
 - Go to Configuration → Data Sources in Grafana.  
 - Select Prometheus.  
 - Set the URL to http://<VM\_IP>:9090.  
 - Click Save & Test.

## 4. Create Grafana Dashboards

1. Create New Dashboard:  
 - In Grafana, navigate to Create → Dashboard.  
 - Add panels to display key metrics:  
 - CPU Usage: Query avg(rate(node\_cpu\_seconds\_total{mode="idle"}[5m])) by (instance).  
 - Memory Usage: Query (node\_memory\_MemTotal\_bytes - node\_memory\_MemFree\_bytes) / node\_memory\_MemTotal\_bytes \* 100.  
 - Disk Usage: Query node\_filesystem\_avail\_bytes.  
  
2. Save the Dashboard:  
 Save the dashboard for future use.

## 5. Configure Alerts in Grafana

1. Set Alert Conditions:  
 On each panel, set up alerts for when resource usage exceeds 75%, for example, CPU usage.  
  
2. Add Notification Channels:  
 Configure notification channels (e.g., Email, Slack) under Alerting → Notification Channels.

## 6. Set Up Auto-Scaling

1. Configure Cloud Auto-Scaling:  
 - Create an Auto Scaling Group in AWS, Azure, or GCP.  
 - Set a scaling policy that triggers scaling when CPU or Memory exceeds 75%.  
 - Use the cloud-specific API to automate scaling based on resource usage.  
  
2. Trigger Auto-Scaling from Grafana Alerts:  
 Use cloud API or manual methods to trigger auto-scaling actions when alerts are received.

# Conclusion

By following the steps above, you can effectively monitor system resources using Grafana and Node Exporter, configure alerts based on resource usage, and set up auto-scaling in a public cloud to handle increased resource demand.