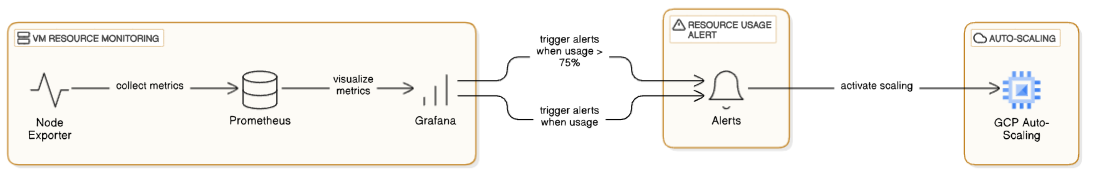
Architecture Design: VM Resource Monitoring and Auto-Scaling

# Overview

This architecture diagram illustrates the flow from local VM resource monitoring using Grafana and Node Exporter, to cloud-based auto-scaling when resource usage exceeds the threshold.  
  
1. Local VM with Node Exporter: Collects system metrics like CPU, memory, disk, and network usage and makes it available for Prometheus scraping.  
2. Grafana: Visualizes the collected metrics in real-time via dashboards, provides alerts when usage exceeds 75%.  
3. Alerts and Auto-Scaling: When Grafana triggers an alert (e.g., CPU > 75%), cloud auto-scaling policies are activated, and additional resources are provisioned in GCP.

# Architecture Design



# Components

- Grafana: Provides real-time monitoring, dashboard creation, and alerting.  
- Node Exporter: Collects system-level metrics and exposes them to Prometheus.  
- Prometheus: Scrapes data from Node Exporter and stores the metrics.  
- Cloud Auto-Scaling (GCP): Automatically scales resources based on resource usage metrics.

# Architecture Flow

1. VM Resource Monitoring:  
 - Node Exporter collects metrics and exposes them.  
 - Prometheus scrapes the metrics and stores them.  
 - Grafana visualizes the metrics in a user-friendly interface.  
  
2. Resource Usage Alert:  
 - Alerts are triggered in Grafana when resource usage exceeds the 75% threshold.  
   
3. Auto-Scaling:  
 - The cloud auto-scaling service adjusts resources in the cloud (GCP) based on the triggers from Grafana alerts.