

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 2
Sección A



Manual Técnico - Prometheus y Grafana

GRUPO 4

Repositorio	https://github.com/Yaiza81/SO2_1S2022_P_G4
-------------	---

Manual Técnico:

Introducción:

En el siguiente escrito se tiene la finalidad de proporcionar al usuario la lógica utilizada para realizar el monitoreo de la máquina virtual por medio de Prometheus y Grafana.

Este manual no está dedicado a enseñar las herramientas utilizadas para contenerizar y orquestar la aplicación. Para conocer la guía de la correcta utilización del programa se debe consultar el “Manual Técnico - Docker y Kubernetes”.

Objetivo del manual:

Se pretende mostrar la lógica empleada y el funcionamiento de las herramientas Prometheus y Grafana para monitorear el estado de la arquitectura de la aplicación. Se describe la manera en la que se trabajó adjuntando capturas de pantalla donde corresponda para una fácil comprensión por parte del lector.

Manual Técnico:

Prometheus:

Prometheus es una solución de monitoreo de código abierto para recopilar y agregar métricas como datos de series de tiempo. Prometheus está diseñado para rastrear el sistema general, la salud y el comportamiento del sistema.

Se utilizó docker Compose para facilitar la instalación de Prometheus. La composición del archivo respectivo fue la siguiente

```
version:
'2.1'

networks:
  monitoring:
    driver: bridge

volumes:
  prometheus_data: {}
  grafana_data: {}

services:
  prometheus:
    image: prom/prometheus:v2.20.1
    container_name: prometheus
    volumes:
      - ./prometheus:/etc/prometheus
      - prometheus_data:/prometheus
    ports:
      - 9090:9090
    expose:
      - 9090
    networks:
      - monitoring

grafana:
```

```

    image: grafana/grafana:7.1.5
    container_name: grafana
    volumes:
      - grafana_data:/var/lib/grafana

-
  ./grafana/provisioning:/etc/grafana/provisioning
  environment:
    - GF_AUTH_DISABLE_LOGIN_FORM=true
    - GF_AUTH_ANONYMOUS_ENABLED=true
    - GF_AUTH_ANONYMOUS_ORG_ROLE=Admin
  ports:
    - 3000:3000
  expose:
    - 3000
  networks:
    - monitoring
node-application-monitoring-app:
  build:
    context: node-application-monitoring-app
  ports:
    - 8080:8080
  expose:
    - 8080
  networks:
    - monitoring

```

Eso incluye las distintas versiones de cada software a utilizar, así como los puertos en donde se estuvieron desplegando cada una.

Tanto para Prometheus como Grafana se utilizaron archivos .yaml para configuraciones más específicas de cada uno. La esencia de todo el funcionamiento se encuentra dentro de un servidor montado en Nodejs.

El cual, tiene una estructura como la siguiente imagen:

```
const client = require('prom-client');
const express = require('express');
const app = express();
const { histogram } = require('./histogram');
const { gauge } = require('./gauge');
const { summary } = require('./summary');
const { counter } = require('./counter');

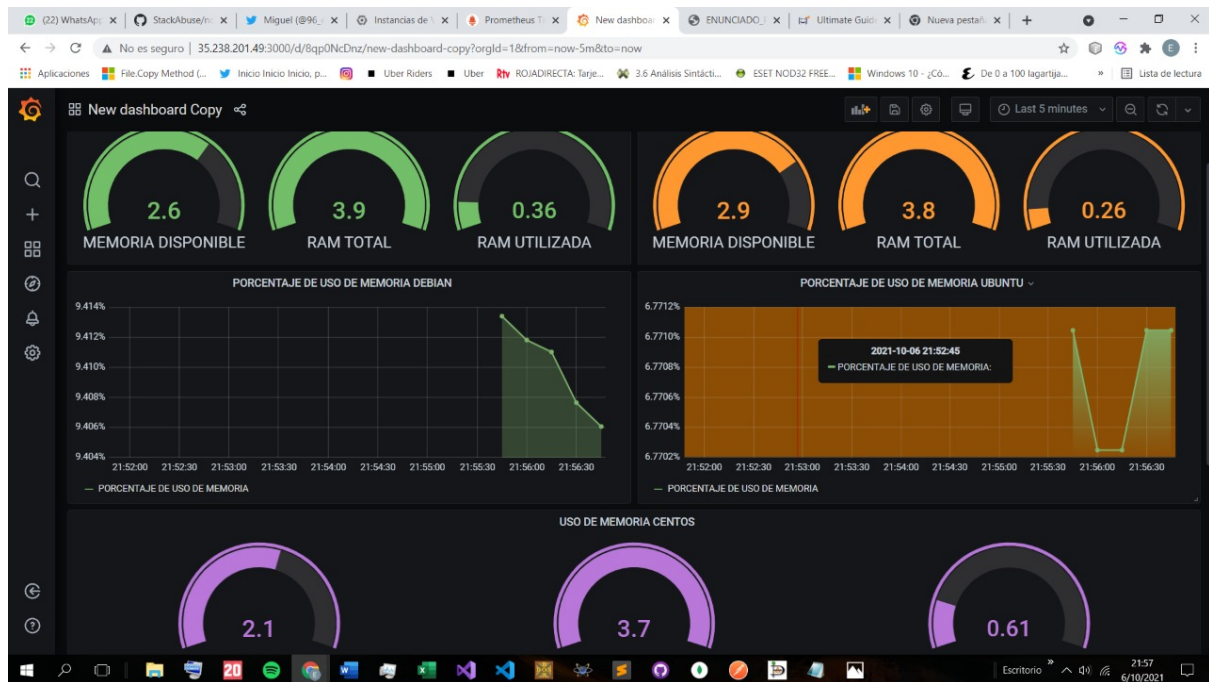
// Create a Registry to register the metrics
const register = new client.Registry();

// Add a default metrics and enable the collection of it
client.collectDefaultMetrics({
  app: 'node-application-monitoring-app',
  prefix: 'node_',
  timeout: 10000,
  gcDurationBuckets: [0.001, 0.01, 0.1, 1, 2, 5], // These are the default buckets.
  register
});

// Create a histogram metric
const httpRequestDurationMicroseconds = new client.Histogram({
  name: 'http_request_duration_seconds',
  help: 'Duration of HTTP requests in seconds',
  labelNames: ['method', 'route', 'code'],
  buckets: [0.1, 0.3, 0.5, 0.7, 1, 3, 5, 7, 10] // 0.1 to 10 seconds
});
```

Grafana:

Es una solución de código abierto que sirve para ejecutar análisis de datos, extraer métricas que dan sentido ante enormes cantidades de datos y monitorear aplicaciones y recursos hardware con la ayuda de atractivos paneles de control personalizables..



Grafana únicamente es diseño del Dashboard para presentar las estadísticas del monitoreo. No requiere ningún tipo de programación.

Descripción de herramientas utilizadas:

➤ Prometheus:

Es un conjunto de herramientas de monitoreo y alerta de sistemas de código abierto creado originalmente en SoundCloud. Es un proyecto independiente de código abierto y se mantiene independientemente de cualquier empresa.

Prometheus recopila y almacena sus métricas como datos de series temporales, es decir, la información de las métricas se almacena con la marca de tiempo en la que se registró, junto con pares clave-valor opcionales denominados etiquetas.



➤ Grafana:

Es una solución de código abierto que sirve para ejecutar análisis de datos, extraer métricas que dan sentido ante enormes cantidades de datos y monitorear aplicaciones y recursos hardware con la ayuda de atractivos paneles de control personalizables.

