Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 2
2do Semestre 2021
Ing. Cesar Batz
Aux. Brayan Chinchilla

Práctica 2

Objetivos

- Poner en practica los conocimientos adquiridos en el laboratorio sobre virtualización a través de maquinas virtuales y contenedores.
- Emplear formas de monitoreo de sistemas operativos.
- Desarrollar aplicaciones utilizando virtualización ligera.
- Desplegar un sistema utilizando un orquestador de contenedores.

Descripción

Se debe desplegar una pequeña aplicación web. Haciendo uso de tecnologías como Docker y Kubernetes, se busca agilizar el escalamiento y despliegue de la aplicación. Adicionalmente, se busca implementar tecnologías de monitorización para tener un panorama claro del estado de los recursos que consume nuestra aplicación.

Virtualización Ligera

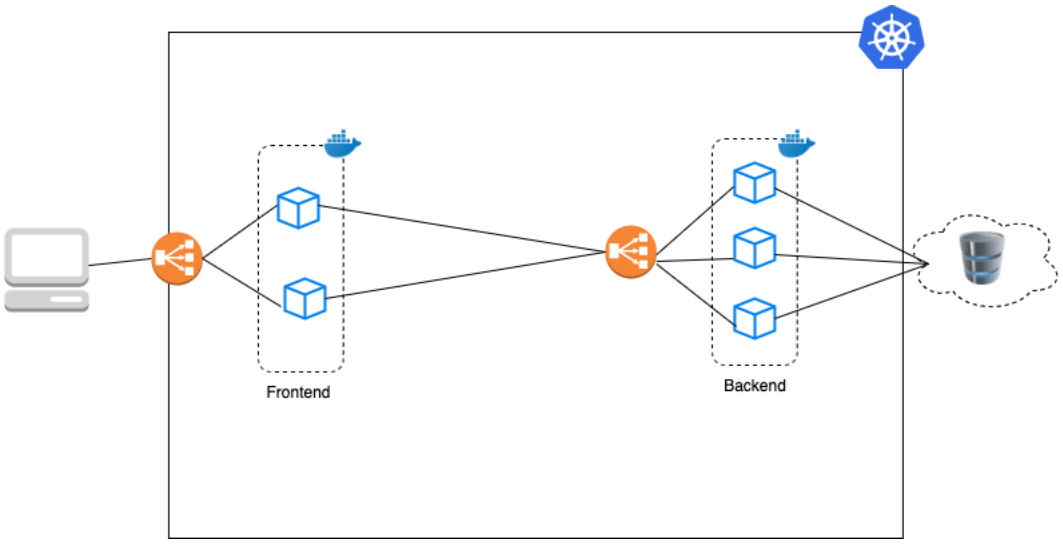
La arquitectura del sistema debe estar conformada por: Frontend, Backend y Base de datos. Estas partes deben comunicarse entre sí. No debe ser un sistema grande ni complejo, simplemente debe de reflejar el uso de los tres distintos componentes mencionados. El estudiante es libre de elegir cuáles tecnologías y lenguajes utilizar en todas las partes, por ejemplo: React, Angular, Nodejs, Java, Python, Nginx, MongoDB, SQL, Google Cloud, AWS, etc. Queda a discreción del estudiante qué sistema implementar, algunos ejemplos de son: pequeño e-commerce, página de películas o videojuegos, versión sencilla de una red social.

Lo importante será utilizar contenedores Docker para desarrollar los componentes previamente mencionados. Tanto backend como frontend deben estar contenerizados para permitir su escalamiento y administración. Por otro lado, la base de datos no necesariamente debe estar contenerizada, sino puede existir como una entidad externa.

Orquestación

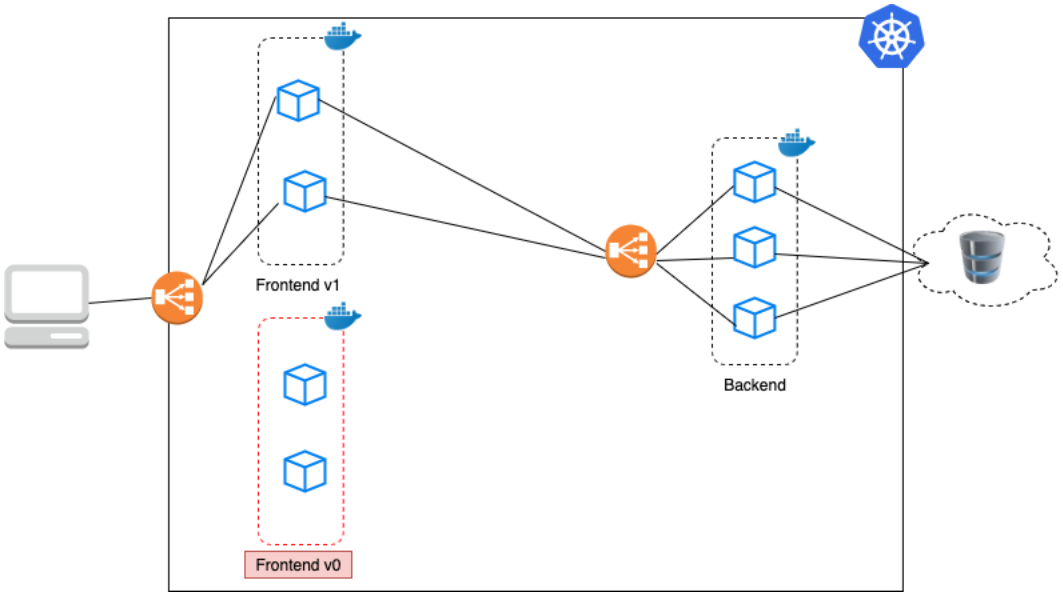
Para desplegar y administrar los contenedores se utilizará Kubernetes como orquestador. El tamaño del clúster y los recursos asignados al mismo quedan a discreción del estudiante, solo se necesita que tenga capacidad para soportar una carga liviana.

Inicialmente, el frontend debe tener una replica de 2 contenedores sirviendo la página web y exponer el servicio públicamente a través de un balanceador de carga. Asimismo, el backend debe tener una replica de 3 contenedores, pero estos deben ser accesibles únicamente por el frontend a través de comunicación privada dentro del clúster. Por otra parte, la base de datos no se alojará dentro del clúster de Kubernetes sino como una entidad externa alojada en la nube, queda a discreción del estudiante desplegar su propia base de datos o utilizar un servicio administrado de base de datos como Mongo DB Atlas.



Es obligatorio que se utilicen archivos YAML para especificar el estado inicial deseado del sistema y ejecutar las acciones para crear y levantar la arquitectura.

Una de las ventajas de utilizar un orquestador de contenedores es la facilidad de desplegar actualizaciones. Durante la calificación, se espera que el estudiante pueda desplegar cambios a su frontend, y tras un despliegue exitoso simular un rollback de dichos cambios. La política de actualización deberá ser RollingUpdate con un tiempo de espera entre lanzamiento de contenedores de 20 segundos. Para agilizar este requisito, se requiere que el estudiante tenga lista una imagen adicional de su frontend con cambios fáciles de apreciar. Se recomienda realizar pruebas de este proceso previamente.



Mientras la tecnología de orquestación se requiere que sea Kubernetes, no existe ninguna restricción en cuanto a los servicios de nube utilizados para montar Kubernetes como GKS o EKS, ni tampoco existe restricción en cuanto al registro de imágenes utilizado como GCR, o ECR.

También se calificará que se tengan conocimientos de cómo funcionan los contendores, el orquestador de Kubernetes, las partes del archivo YAML, así como los comandos que se utilizan para interactuar con el orquestador de contenedores.

Monitorización

Al momento de levantar servidores, el monitoreo es una de las cuestiones más importantes que se deben tomar en cuenta, debido a que es necesario saber el estado en que se encuentra nuestra arquitectura.

Prometheus es una base de series de tiempo y un sistema de monitoreo y alertas. Las series de tiempo almacenan datos ordenados cronológicamente, midiendo variables a lo largo del tiempo y las bases de datos enfocadas a series de tiempo son especialmente eficientes en almacenar y consultar estos datos. Grafana es la plataforma de análisis para métricas, que le permite consultar, visualizar, alertar y comprender los datos, sin importar dónde estén almacenadas.

En uno de los nodos del clúster de Kubernetes que se ha instalado, se debe de contar con Prometheus y Grafana para poder realizar el monitoreo. Se deben realizar las configuraciones necesarias o emplear las plantillas adecuadas para poder monitorear cosas elementales como:

- Estado de CPU
- Estado de RAM
- Tráfico de red

Entregables:

- Código fuente de las aplicaciones.
- Archivos de configuración de Docker y Kubernetes.
- Manual técnico sobre la instalación y configuración de Prometheus y Grafana.
- Manual técnico sobre la configuración y despliegue del sistema con Docker y Kubernetes.

Consideraciones

- La practica se realiza en los grupos de 2 o 3 integrantes establecidos.
- Los archivos de configuración de Kubernetes deben ser escritos por el estudiante y no configurados por algún software.
- Cualquier copia total o parcial será reportada a la Escuela de Sistemas para que proceda como corresponde.

Penalizaciones

- 30% de la nota si se emplea un software para generar los archivos de configuración.

Forma de Entrega

- Mediante UEDI subiendo el archivo .zip: [SO2]Practica2_<no_grupo>
- Subiendo todo a un repositorio de GitHub el cuál debe ser privado con el nombre: practica2_grupo<no_grupo>_so2. Se deberá agregar al usuario: *brayan-chinchilla*.

Fecha de Entrega:

Jueves 11 de noviembre de 2021 hasta las 17:00