



UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Kubernetes

Oscar Evanilson Gutiérrez Pérez

Carrera: Ingeniería en Computación

Código de estudiante: 219748308

4 de abril de 2022

¿Qué es Kubernetes?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa. Tiene un ecosistema grande y en rápido crecimiento. El soporte, las herramientas y los servicios para Kubernetes están ampliamente disponibles.

Google liberó el proyecto Kubernetes en el año 2014. Kubernetes se basa en la experiencia de Google corriendo aplicaciones en producción a gran escala por década y media, junto a las mejores ideas y prácticas de la comunidad.

¿Qué es Ingress?

Brevemente, Ingress se usa para exponer rutas HTTP y HTTPS desde el exterior a servicios dentro del Cluster. El tráfico entrante es controlado por las reglas que definimos en un archivo de configuración.

¿Qué es un LoadBalancer?

Cuando creamos un servicio del tipo LoadBalancer, automáticamente se aprovisiona un balanceador de carga externo al Cluster que nos proporciona una dirección IP a través de la que acceder a nuestros servicios.

Para conseguir eso, Kubernetes se sincroniza con la API del proveedor Cloud que estemos usando. Por cada servicio que creamos se aprovisiona un balanceador de carga diferente. Cuando el servicio es eliminado, el balanceador de carga que tiene asociado también es eliminado.

¿Qué es Rancher?

Rancher es una pila de software que se utiliza para gestionar clústeres Kubernetes. Se trata básicamente de un software que DevOps puede utilizar al adoptar el usuario de contenedores. Rancher incluye una distribución completa de Kubernetes, Docker Swarm y Apache Mesos, lo que facilita la gestión de clústeres de contenedores en cualquier plataforma de nube. Algunas de las empresas populares que utilizan

Rancher son: Alibaba travelers, Abeja, Trivago, UseInsider, Starbucks, Oxylabs, yousign, y muchas más.

Una de las ventajas significativas de Rancher es la capacidad de gestionar múltiples clústeres Kubernetes de forma simplificada. Ofrece una gestión simplificada de múltiples clústeres Kubernetes que pueden ser creados manualmente utilizando la distribución de Kubernetes de Ranchers llamada RKE (Rancher Kubernetes Engine) o importados en el panel de gestión del gestor de clústeres.

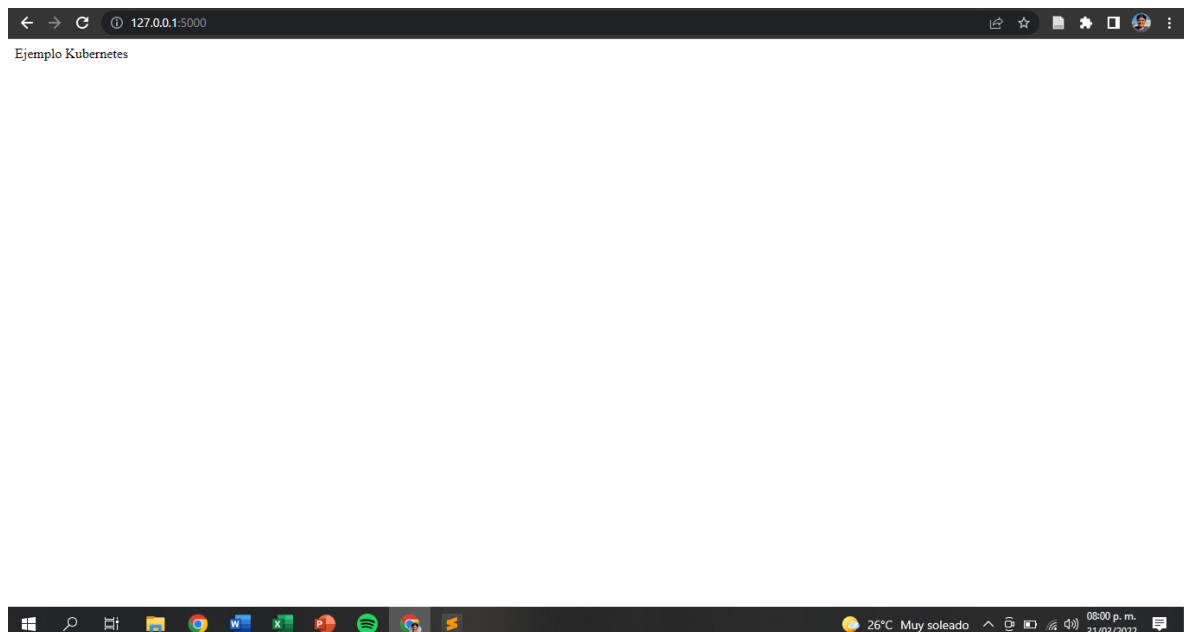
En este programa, utilicé el lenguaje de Python para hacer un ejemplo de un programa Python en Kubernetes, es un ejemplo sencillo donde se corre una aplicación en un servidor local, con ayuda de kubernetes.

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route("/")
5  def hello():
6      return "Ejemplo Kubernetes"
7
8  if __name__ == "__main__":
9      app.run(host='0.0.0.0')
```

En esta primera parte del ejemplo vemos el programa de Python, este archivo main tiene la función de crear un servidor local para nuestra aplicación, donde se verá escrito una frase, para esto utilicé la librería de Python llamada flask, esta librería es buena para desarrollo de aplicaciones o páginas web.

```
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Al correr el programa, se nos arroja esta información de que nuestra página web se está corriendo correctamente en el servidor local 5000.



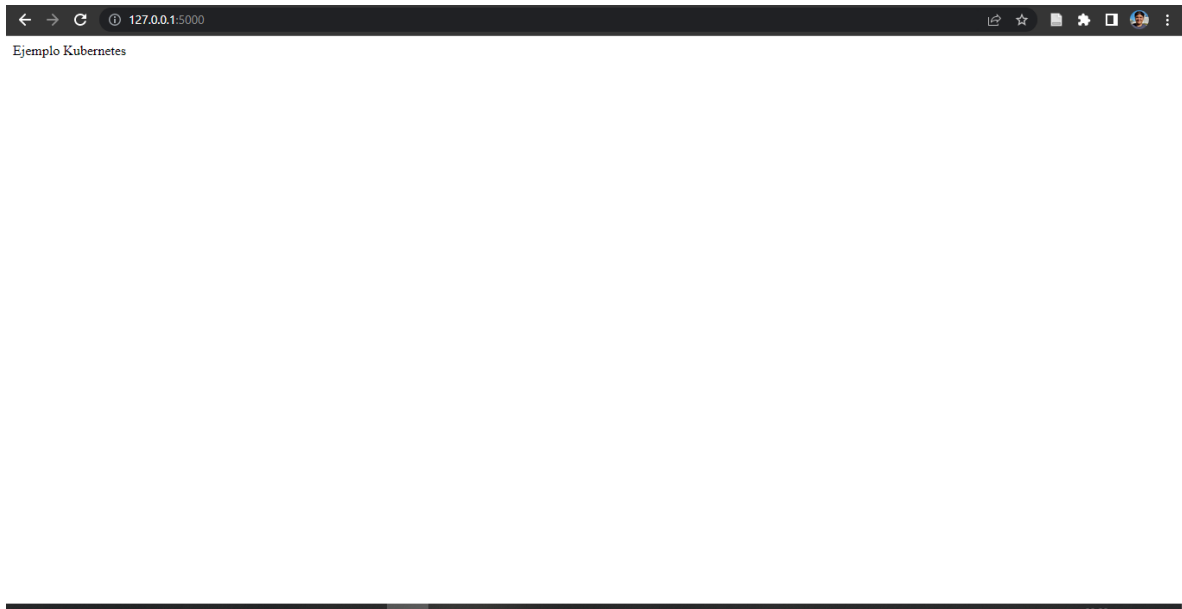
Ahora para comparar lo de esta semana con la pasada, crearemos un archivo Docker para correr nuestra aplicación, el código de este archivo es el siguiente:

```
1 FROM python:3.7
2
3 RUN mkdir /app
4 WORKDIR /app
5 ADD . /app/
6 RUN pip install -r requirements.txt
7
8 EXPOSE 5000
9 CMD ["python", "/app/main.py"]
```

Este archivo hará lo siguiente:

- Obtener la imagen base oficial de Python para la versión 3.7 de Docker Hub.
- En la imagen, crear un directorio llamado app.
- Establecer el directorio de trabajo en ese nuevo directorio de aplicaciones.
- Copiar el contenido del directorio local a esa nueva carpeta en la imagen.
- Ejecutar el instalador de pip (tal como lo hicimos antes) para incorporar los requisitos a la imagen.
- Informar a Docker que el contenedor escucha en el puerto 5000.
- Configurar el comando de inicio para usar cuando se inicie el contenedor.
- Crear una imagen

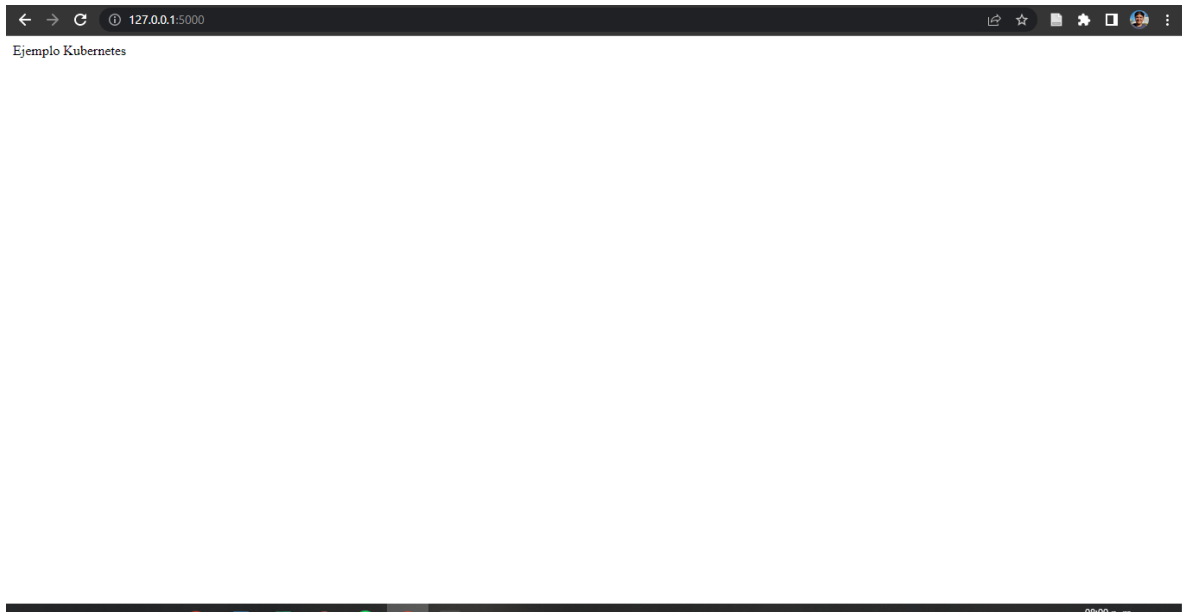
Y ahora solo debe de correrse el archivo con Docker.



Ahora para correr la aplicación con Kubernetes debemos crear el siguiente archivo:

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: hello-python-service
5  spec:
6    selector:
7      app: hello-python
8    ports:
9      - protocol: "TCP"
10      port: 5000
11      targetPort: 5000
12    type: LoadBalancer
13
14
15  ---
16  apiVersion: apps/v1beta1
17  kind: Deployment
18  metadata:
19    name: hello-python
20  spec:
21    replicas: 4
22    template:
23      metadata:
24        labels:
25          app: hello-python
26      spec:
27        containers:
28          - name: hello-python
29            image: hello-python:latest
30            imagePullPolicy: Never
31            ports:
32              - containerPort: 5000
```

Y si lo corremos vemos que también funciona nuestra aplicación.



Link de supositorio en GitHub para revisar el código:

<https://github.com/oscarevanilson/Computacion-Tolerante-a-Fallas>

Conclusiones

Este programa me resultó muy similar al que realizamos en la clase anterior, y gracias a eso pude comprender de mejor manera lo que debía de realizar en esta tarea asignada.

Me pareció muy interesante realizar la tarea de esta semana ya que en mi caso lo que hice fue correr una aplicación de manera local de tres formas, una desde lo nativo con flask, otra con lo que vimos la semana pasada que fue Docker y la última con lo que trata la actividad de esta semana que fue Kubernetes.

Creo que es muy similar el resultado final que el usuario puede llegar a observar al momento de utilizar alguna de estas tres formas para correr una aplicación, mínimo de manera local, pero creo que por lo que vi en internet, utilizar Docker o Kubernetes es mejor, ya que administra de mejor manera nuestra aplicación y algunos errores que pueden llegar a suceder.

Por último, espero que se avance el desarrollo de todas estas herramientas para que en un futuro sea más fácil de utilizar y de comprender, además de que se puedan hacer más cosas con estas y que pueda ayudar a muchos programadores en proyectos importantes o incluso a estudiantes como nosotros en sus tareas y proyectos.

Referencias

- ¿Qué es Kubernetes? (2022, 11 febrero). Kubernetes. Recuperado 31 de marzo de 2022, de <https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/>
- Pujol, J. (2021, 14 diciembre). Ingress en Kubernetes Desmitificado: ¿Qué lo diferencia de un NodePort o un LoadBalancer? Medium. Recuperado 31 de marzo de 2022, de <https://desarrollofront.medium.com/ingress-en-kubernetes-desmitificado-qu%C3%A9-lo-diferencia-de-un-nodeport-o-un-loadbalancer-b0cf060a6f8a>
- S. (2021, 4 mayo). Todo lo que necesita saber sobre Rancher: gestión de Kubernetes para empresas. SiXe Ingeniería. Recuperado 31 de marzo de 2022, de <https://sixe.es/noticias/suse-rancher-kubernetes-toda-la-informacion>
- Get started with Kubernetes (using Python). (2021, 25 mayo). Kubernetes. Recuperado 31 de marzo de 2022, de <https://kubernetes.io/blog/2019/07/23/get-started-with-kubernetes-using-python/>