KUBERNETES EJEMPLO

Instalaciones

Instalaciones

- Lo primero que se debe realizar es la instalación de Kubectl. Los pasos adecuados a cada sistema operativo se encuentran en la siguiente liga:

Instalar y Configurar kubectl | Kubernetes

- Después es necesario instalar VirtualBox, para poder ejecutar la máquina virtual de Minikube. La descarga del software está en la siguiente liga:

Descargas – Oracle VM VirtualBox

- Para finalizar se debe instalar Minikube, los pasos para la instalación en los diferentes sistemas operativos se encuentran en la siguiente y última liga:

Instalar Minikube | Kubernetes

- De esta manera ya estarían listas las herramientas que serán necesarias para poder llevar a cabo la ejecución de el ejemplo.

Abrir interfaz visual

- Antes que nada se debe iniciar minikube. Para poder ejecutar la máquina virtual de minikube se ejecuta el comando:

```
minikube start
```

- Ya realizado esto se debe abrir la insterfaz. Para poder abrir la interfaz visual se realiza con el siguiente comando:

```
minikube dashboard
```

- Una vez realizado esto, ya se tiene perfectamente configurado kubernetes en nuestra máquina local.
- Ahora es necesario hablar sobre unos conceptos...

Conceptos básicos

- **Namespaces:**
  Los namespaces (espacios de nombres) en Kubernetes permiten establecer un nivel adicional de separación entre los contenedores que comparten los recursos de un clúster.

- Podrían denominarse entornos de trabajo. (desarrollo, pruebas, producción)

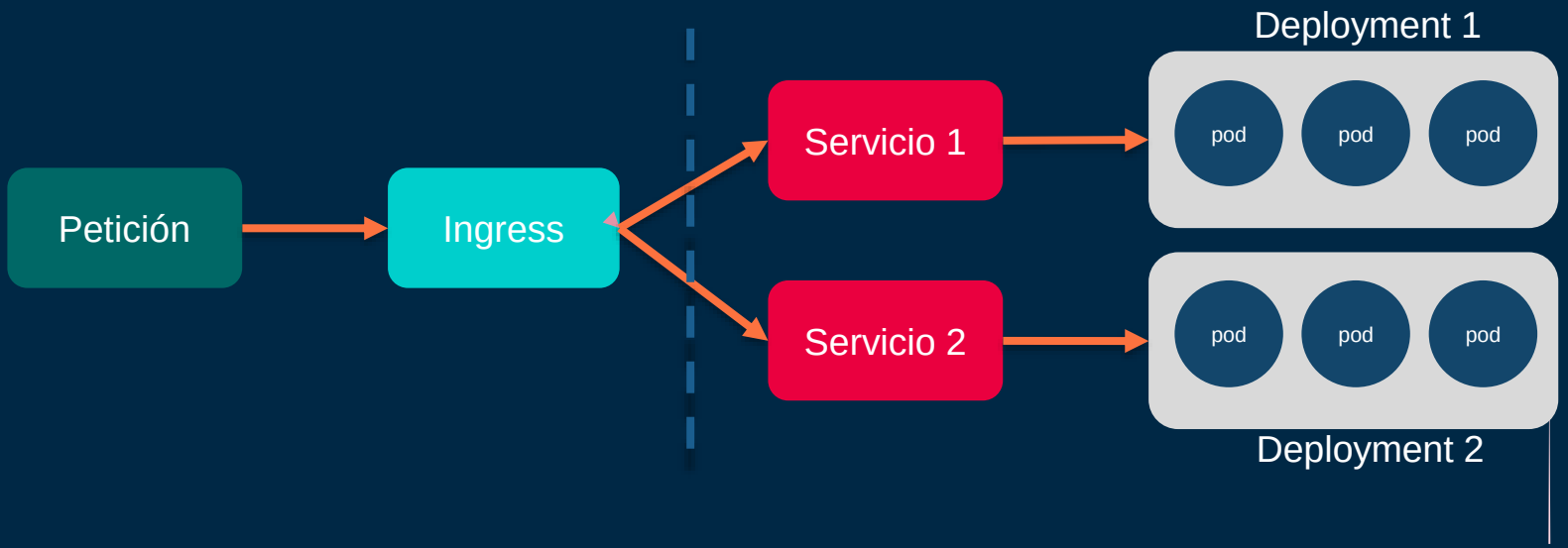
- **Pods:** Los pods son los objetos más pequeños y básicos que se pueden implementar en Kubernetes. Un pod representa una instancia única de un proceso en ejecución en un clúster.
  Los pods contienen uno o más contenedores, como los contenedores de Docker.

Conceptos básicos

- **Deployment:** Un controlador de Deployment proporciona actualizaciones declarativas para los Pods y los ReplicaSets. Cuando describes el estado deseado en un objeto Deployment, el controlador del Deployment se encarga de cambiar el estado actual al estado deseado de forma controlada.
- Se encarga de crear pods y replicas para esos pods
- **Services:** Los servicios (services) nos permiten acceder a nuestras aplicaciones.
- **Ingress**: Por último tenemos a ingress, este recurso nos permite acceder a servicios a través de HTTP(S) y el tráfico se controla utilizando un conjunto de reglas que tú defines.

¿Cómo se relacionan estos componentes?

- El siguiente diagrama muestra la relación real de los componentes descritos anteriormente:

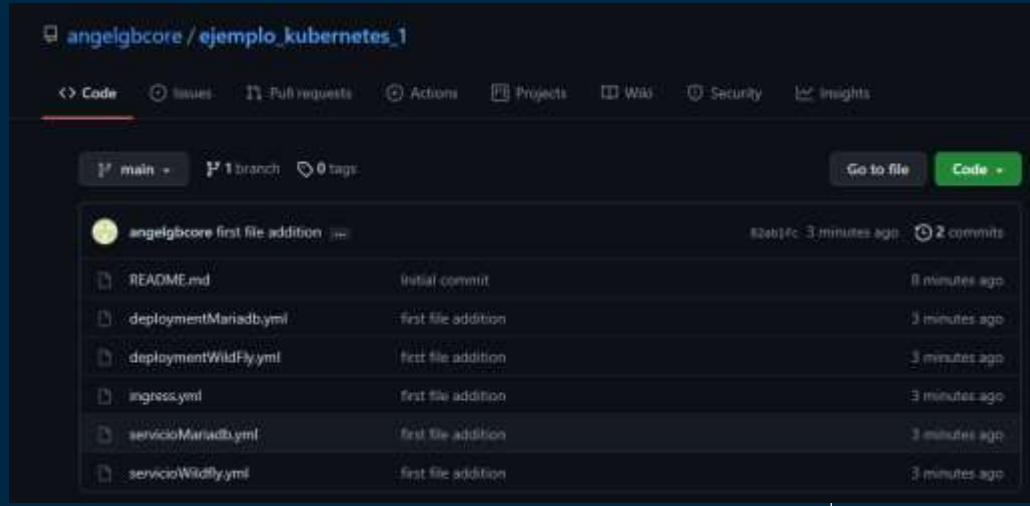


- La documentación oficial:
Kubernetes Documentation | Kubernetes

Manos a la obra

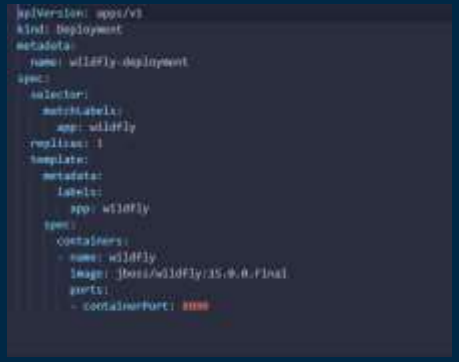
- Para empezar a trabajar, se deben tener los archivos de configuración en tipos de archivo con extensión .yml que se encuentran en el siguiente repositorio:

https://github.com/angelgbcore/ejemplo_kubernetes_1.git

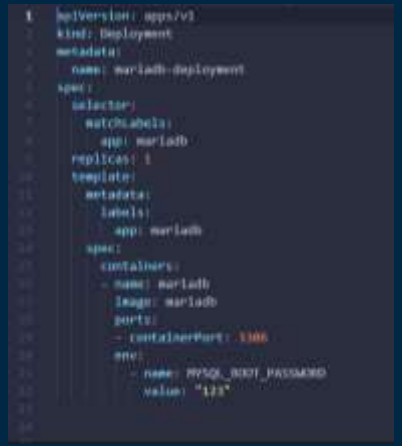


Comandos

Para empezar, debemos de posicionarnos en la carpeta donde clonaron el repositorio, para poder ejecutar los respectivos comandos que aparecen debajo de cada imagen:



Kubectl apply –f deployWildfly.yml



Kubectl apply –f deployMariadb.yml

```yaml
kind: Service
apiVersion: v1
metadata:
  name: wildfly-service
spec:
  selector:
    app: wildfly
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
```

Kubectl apply –f servicioWildfly.yml

```yaml
kind: Service
apiVersion: v1
metadata:
  name: mariadb-service
spec:
  selector:
    app: mariadb
  ports:
  - protocol: TCP
    port: 3306
    targetPort: 3306
```

Kubectl apply –f servicioMariadb.yml

Cabe mencionar que primero se ejecutan los comandos de los deploys y después los de servicios

Ingress

- Para terminar la configuración de componentes se necesita el ingress.
- Para poder ejecutar ingress es necesario tener que ejecutar los comandos:

`minikube addons enable ingress`
(este es para activar el la funcionalidad del ingress)

`kubectl apply –f ingress.yml`
(este ya es para aplicar la configuración del ingress)

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: wildfly-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /
        backend:
          serviceName: wildfly-service
          servicePort: 8080
```

Revisando…

- Ya terminados los pasos anteriores con éxito, debemos de revisar que realmente se esté ejecutando el servicio WildFly que hasta ahora hemos creado.

- Ejecutamos el comando siguiente para conocer la dirección IP de la máquina virtual minikube para poder abrir el servicio de wildfly en el navegador:

```
minikube ip
```

- Ingresamos por último la dirección al navegador

Visualizar en el navegador el servicio WildFly

- Hasta ahora ya se puede observar de manera gráfica los resultados del ejemplo.
- Internamente sucede tal y como se plasma en el diagrama presentado con anterioridad