Clase 2

**Ejercicio 1**

- Vamos a crear un pod con un configmap con la documentacion oficial de google. Del ejemplo oficial:

https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/

- Creamos el configmap

kubectl create configmap special-config --from-literal=special.how=very

- Creamos el pod desde el archivo proveido en la carpeta googl

kubectl apply -f pod.yaml

- Vamos a ver si funciono

kubectl get pod

- Ahora, vamos a investigar adentro, con lo que ya vimos anteriormente

kubectl logs <>

kubectl describe pod <>

- Ahora, vamos a borrar todo lo hecho

kubectl delete pod <>

kubectl delete configmap <>

**Ejercicio 2**

- Ahora vamos a crear un pod desde cero, usar una registry privada y un configmap

- Vamos a crear un archivo index.html con un hola mundo y nuestro usuario

nano index.html ...

\*\*\* Hay un dockerfile de ejemplo, solo falta que le carguen el html, en la carpeta ejercicio

2.- Vamos a crear una imagen de docker con apache

docker build -t "<usuario docker hub>/<imagen>:01" .

3.- Vamos a loguearnos

docker login

4.- Vamos a enviar la imagen a docker hub

docker push <usuario docker hub>/<imagen>:01

5.- Vamos a iniciar minikube

minikube start

6.- Vamos a crear los secrest para que kubernetes pueda bajar la imagen que le vamos a enviar

kubectl create secret docker-registry regcred --docker-server=<your-registry-server> --docker-username=<your-name> --docker-password=<your-pword> --docker-email=<your-email>

7.- Vamos a crear el configmap con desde un archivo

kubectl create configmap game-config --from-file=game.properties

7.- Vamos a crear el pod con secret registry y el configmap

\*\*\* Esta el ejemplo adjunto, cambiar lo que corresponda.

8.- Vamos a exponer el pod

kubectl expose deployment <pod nombre> --type=NodePort --port=80

9.- Vamos a ver los datos de su creacion

kubectl describe pod <pod>

10.- Vamos a ver los logs del pod

kubectl logs <pod>

**Ejercicio 3**

- Ahora vamos a crear un típico job de seguridad, para esto vamos a usar un proyecto open llamado:

<https://github.com/ghcetraro/kube-bench>

- Para esto va a ser necesario tener instalardo git, sino lo tiene instalarlo con:

sudo apt-get install git git-core

- Nos clonamos el proyecto con:

git clone <https://github.com/ghcetraro/kube-bench>

- Buscamos un archivo que se llama jos.yaml y lo aplicamos

kubectl apply -f jobs.yaml

- Ahora vamos a buscar si se creo el pod y en que estado esta, con:

kubectl get pod | grepo kube

- Vamos a ver, si todo salio bien, que el pod esta creado pero apagado. Esto es porque es un job, corrio hizo su tarea y se apago esperando alguna accion nueva.

Este como es un pod de seguridad, podemos ver hacer un logs que hizo, vamos a hacer eso

kubectl logs <pod>

- Ahora vamos a buscar el job que lo genero con:

kubectl get jobs

- Si borramos el job, tambien se borra el pod, vamos a hacer eso:

kubectl delete jobs <nombre>

- Verficicamos que se borro el job

kubectl get jobs

- Y ahora vamos a verificar que se haya borrado el pod:

kubectl get pods