

## **Fase 4 - Activitat 9.4: Creant un cluster de kubernetes. Desplegant una aplicació sobre un cluster de kubernetes.**

### **0- Identificació del grup i activitat:**

**Curs:** ASIX2

**Projecte:** GP2 DevOps i Cloud Computing

**Fase:** 4

**Activitat:** 9.4

**Grup:**

**Membres:**

### **1- Introducció i objectius de l'activitat 9.4**

- a) Lectura de les especificacions de l'activitat
- b) Creació d'un cluster de kubernetes
- c) Desplegant una aplicació sobre un cluster de kubernetes

### **2- Introducció d'alguns conceptes més sobre kubernetes**

- a) Què és un node **worker**?: És qualsevol node sobre el qual s'executa una aplicació per mitjà de contenidors que es trobaran dins de Pods. El **control plane** controla i monitoritza els nodes **workers** i els **Pods** que s'executen a cada **worker**.
- b) El **control plane** pot estar situat a un node o distribuït entre diversos nodes del cluster. Un node pot formar part del control plane o actuar únicament com a worker.
- c) Què és **minikube**?: Una implementació senzilla de kubernetes que requereix menys recursos, però només funciona sobre un host i no seria una opció acceptable per treballar en producció.
- d) Què és un **deployment** (desplegament)?: Per mitjà d'un **deployment** indicarem el número de Pods inicials sobre els quals s'executen els contenidors de l'aplicació, la imatge del contenidor a on es troba l'aplicació, el nom de l'aplicació i el port a través del qual accedirem a l'aplicació.
- e) Què és un **service** (servei)?: Per mitjà d'un **service** exposarem tots els Pods de l'aplicació com si fosin un únic servei de xarxa i també podrem fer balanceig de carrega. D'aquesta manera, farem disponible i accessible l'aplicació des de l'exterior del cluster com si fos una única aplicació sobre un únic servidor.

### **3- Creant un entorn amb els requeriments mínims per treballar amb minikube**

- a) Els requeriments mínims de **minikube** són: 2 CPUs, 2048MiB i 20GiB de disc dur.
- b) Fes un destroy de les màquines virtuals actuals creades amb l'activitat **gp1f4a9.2**. Fes una còpia de seguretat del fitxer **Vagrantfile** creat a l'activitat **gp1f4a9.2** de nom **gp1f4a9.2\_Vagrantfile**.
- c) Modifica el fitxer **Vagrantfile** de manera que es crearà només 1 màquina virtual amb els requeriments mínims de sistema.
- d) Modifica el fitxer **Vagrantfile** i esborra totes les línies d'instal·lació de **microk8s** (a l'activitat gp1f4a9.2 pots veure quines són aquestes línies) i afegeix aquestes línies per instal·lar **minikube**:

```
sudo apt-get install -y kubernetes-client
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
sudo dpkg -i minikube_latest_amd64.deb
exit
```

e) Inicia la màquina executant: **vagrant up**

f) Accedeix a la màquina via SSH i comprova que:

- **minikube** funciona executant: **minikube version**
- L'eina de gestió **kubectl** de **minikube** funciona executant: **kubectl version --client -o yaml**

#### **4- Testejant el funcionament de minikube. Creació d'un cluster d'un node, desplegament d'una aplicació i accés a l'aplicació.**

a) Crearem un cluster que utilitzi containers Docker com a base. Això és necessari perquè Docker és una implementació del programari de contenidors però no és l'única implementació disponible. Accedeix a **node1** i executa:

```
minikube start --driver docker
```

Això pot trigar més o menys temps en funció de la velocitat d'internet.

b) Comprova que ara ja tenim un cluster executant:

```
kubectl get node
```

i comprova que ara el teu node funciona com un **control-plane** del cluster.

c) Comprova que pots controlar i monitoritzar el cluster:

- Comprova l'estat del cluster: **minikube status**
- Comprova pot aturar el cluster: **minikube stop**
- Comprova pot iniciar el cluster: **minikube start**
- Comprova pot reiniciar el cluster: **minikube restart**
- Comprova pots eliminar el cluster: **minikube delete**

c) Esborra el cluster. Torna a crear-lo el cluster amb el nom **gp1f4a9**:

```
minikube start --driver docker -p gp1f4a9
```

d) Ara desplegarem una aplicació de prova. Executa les següents ordres per fer un **deployment** (desplegament) sobre **kubernetes** i crear un **service** (servei):

```
kubectl create deployment app1 --image nginx
```

```
kubectl expose deployment app1 --name app1-svc --type NodePort --port 80
```

e) Comprova de quina manera serà accessible aquest servei. Executa:

```
minikube service app1-svc --url -p gp1f4a9
```

f) Executa:

```
curl http://<ip>:<num_port>
```

a on **ip** i **port** són els valors mostrats. Comprova que pots veure la pàgina inicial de **nginx**.

g) Comprova:

- Els nodes del cluster: `kubectl get nodes`
- Els **Pods** del cluster : `kubectl get pods`
- Els **deployments** del cluster : `kubectl get deployments`
- Els **services** del cluster : `kubectl get services`

## **5- Creant un cluster amb 2 nodes**

a) Crea un cluster de **2 nodes** amb el nom **gp1f4a9.4**. Executa:

```
minikube start --driver docker --nodes 2 -p gp1f4a9.4
```

b) Comprova l'estat del cluster: `minikube profile list`

c) Comprova l'estat dels nodes:

```
kubectl get nodes
```

```
minikube status -p gp1f4a9.4
```

d) Afegeix el node **gp1f4a9-m02** com a worker:

```
kubectl label node gp1f4a9-m02 node-role.kubernetes.io/worker=worker
```

i torna comprovar l'estat del nodes.

e) Esborra el cluster **gp1f4a9.4**. Executa: `minikube delete -p gp1f4a9.4`

## **Lliurament de l'activitat - I (fins a l'apartat 4)**

a) Mostra els nodes, pods, deployments i services de **gp1f4a9**.

b) Demuestra que pots accedir a l'aplicació desplegada des de la màquina virtual amb **curl**.

c) Data límit per obtenir el 100% de la nota: **dilluns 27-11-22 a les 17.45**. A

## **Lliurament de l'activitat - II (apartat 5)**

a) Esborra el cluster **gp1f4a9**. Crea un nou cluster amb el mateix nom amb 2 nodes.

b) Fes que el 2n node sigui un worker.

c) Mostra l'estat del cluster i els nodes.

d) Data límit per obtenir el 100% de la nota: **dilluns 27-11-22 a les 17.45**.