

Fase 4 - Activitat 9.3: Orquestració de contenidors amb Kubernetes. Monitorització. Escalabilitat. Fault recovery

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 9.4

Grup/Individual: Individual

Membres/Alumne:

1- Objectius de l'activitat 9.3

- Lectura de les especificacions de l'activitat.
- Monitorització de l'aplicació desplegada a l'activitat pj9f4a9.2 amb l'eina Dashboard de de Kubernetes
- Escalar i descalar l'aplicació desplegada a l'activitat pj9f4a9.2 amb l'eina Dashboard de de Kubernetes

2- Eina Dashboard de Kubernetes

L'eina **Dashboard** és una eina web que forma part de **Kubernetes** i que permet entre moltes d'altres opcions:

- Monitoritzar nodes, pods, deployments, services i proxys.
- Pot esborrar o reiniciar deployments,pods,services i proxys
- Permet escalar/descalar pods.

3- Habilitació i accés al Dashboard de kubernetes per monitoritzar aplicacions

- Posa en marxa la màquina **node921** que vas crear a l'activitat **pj9f4act9.2**. Instal·la i habilita el plugin **Dashboard** de **Kubernetes** executant dins de la màquina virtual **node921**:

```
microk8s enable dashboard
```

- Executa dins de **node921** la següent ordre:

```
microk8s status --format short | grep dashboard
```

i comprova que el **Dashboard** ha estat habilitat executant. Hauria de sortir el missatge:

```
core/dashboard: enabled
```

- Dins del directori **ppv** crea un altre directori de nom **dashboard**. Accedeix a la carpeta creada i crea un token d'accés que serà posteriorment necessari per accedir al **Dashboard** executant:

```
microk8s kubectl create token default > token
```

- Per fer accessible des de la màquina virtual **node921** el **Dashboard** cal exposar el port intern **443/tcp** del cluster a un port lliure de la màquina virtual fent una redirecció de ports (port forwarding). Executa:

```
microk8s kubectl port-forward -n kube-system service/kubernetes-dashboard 10443:443 --address 0.0.0.0 &
```

i el resultat serà similar a:

```
[1] 42586
```

```
vagrant@k8s:~/dashboard$ Forwarding from 0.0.0.0:10443 -> 8443
```

e) Prem **Enter** per tonar treballar al terminal sense aturar el port forwarding. A continuació executa:

```
sudo netstat -atupn | grep kubectl | grep LISTEN
```

i comprova que **kubectl** ha obert el port **10443/tcp**.

f) Troba l'adreça IP de la màquina virtual **node921**. Executa: **ip -4 -br add show eth1**

g) Des del navegador **Firefox** de la teva **màquina física** estableix una connexió a **https://ip_màquina_node921:10443** a on **ip_màquina_node921** és l'adreça trobada a l'apartat anterior.

h) Accepta el risc potencial de seguretat anant a: *Avançat* → *Accepto el risc i vull continuar*.

i) Autenticat dins del **Dashboard**:

- Selecciona l'opció **Token** dins del formulari
- Escriu el token que vas crear a l'apartat **d)** a la secció **Enter token ***.
 - Si el token és vàlid i no ha expirat accediràs al **Dashboard**.
 - Si el token no val o ha expirat, crea un altre de la mateixa manera que ho vas fer a l'apartat **d)**.

4- Monitorització bàsica de l'estat de l'aplicació desplegada amb Kubernetes

a) Accedeix a Workloads → Deployment. Comprova que el Deployment **depl-ppv**:

- Funciona correctament
- Utilitza 3 Pods

b) Accedeix a Workloads → Pods. Comprova:

- Que tots els Pods de **depl-ppv** funcionen correctament.
- L'utilització total de CPU del conjunt de tots els Pods de **depl-ppv** expressat en cores.
- L'utilització total de la RAM per part del conjunt de Pods de **depl-ppv** expressat en MiB.

c) Accedeix a Service → Services. Comprova que el servei **serv-ppv** funciona correctament.

d) Accedeix a Service → Ingresses. Comprova que l'ingress **proxy-ppv** està activat.


e) Accedeix a Cluster → Nodes. Comprova que el node **node921** funciona correctament.

f) Accedeix a Cluster → Nodes → **node921**. Comprova:

- La quantitat de Pods en marxa. Tots són **Pods** de l'aplicació **PPV**?. Què són els altres **Pods**?
- La quantitat de cores de **node921** utilitzats pels **Pods** de l'aplicació **PPV**.
- La quantitat de memòria RAM de **node921** utilitzada pels **Pods** de l'aplicació **PPV**.

5- Escalament i desescalament manual dels Pods de les aplicacions

a) Escala l'aplicació desplegada amb **depl-ppv** manualment de manera que treballi amb **10** Pods:

- Accedeix a Workloads → Deployment.
- Selecciona Actions  de **depl-ppv**.
- Selecciona **Scale**.
- Demana treballar amb **10** repliques dins de la secció **Desired replicas**.
- Fes click a **Scale**.

b) Torna Workloads → Deployments i comprova que ara **depl-ppv** treballa amb **10** Pods.

c) Ara fes que **depl-ppv** treballa amb **2** Pods.

NOTA: L'escalament automàtic està més enllà del temari d'aquest projecte però si algú té interès en conèixer com funciona, pot consultar-ho aquí:

<https://livebook.manning.com/book/kubernetes-in-action/chapter-15/7>

En tot cas, com a mínim estaria bé saber que es pot auto-escalar a partir de:

- L'utilització de la CPU
- L'utilització de memòria RAM
- Mètriques personalitzades

6- Creació automàtica de nous Pods quan un dels que estan en execució deixa de funcionar

Accedeix a Workloads → Pods i a continuació:

- Comprova el codi identificador del primer Pod.
- Selecciona Actions → del primer Pod.
- Escull l'opció Delete
- Comprova que el pod deixa de funcionar i existir.
- Comprova que passats uns segons es posa en marxa un nou Pod amb un nou codi identificador.

Lliurament de l'activitat

- a) Accés a Dashboard.
- b) Mostra que depl-ppv, serv-ppv i els Pods de l'aplicació funcionen tots correctament.
- c) Visualitza l'utilització total de CPU i RAM del Cluster de Kubernetes.
- d) Escala a 15 la quantitat de Pods de l'aplicació de PPV.
- e) Atura i esborra un Pod de l'aplicació de PPV i comprova que automàticament es crea un altre.
- f) Què són els altres Pods de Kubernetes que no formen part de l'aplicació PPV?
- g) Data límit per obtenir el 100% de la nota: **dilluns 20-1-25 a les 17.45.**