

## **Fase 4 - Activitat 10: Monitorització del sistema i d'aplicacions desplegadas sobre kubernetes**

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

**Curs:** ASIX2

**Projecte:** GP2 DevOps i Cloud Computing

**Fase:** 4

**Activitat:** 10

**Grup:**

**Membres:**

### **1- Objectius de l'activitat 10**

- a) Lectura de les especificacions de l'activitat.
- b) Monitorització del funcionament de les aplicacions desplegadas a l'activitat 9.5 amb l'eina Dashboard de de Kubernetes
- c) Escalabilitat i monitorització del cluster
- d) Monitorització del sistema amb eina desplegadas utilitzant docker.

### **2- Conceptes a desenvolupar dins d'aquesta activitat**

a) L'eina **Dashboard** és una eina web que forma part de **Kubernetes** i que permet:

- Monitoritzar els nodes, pods, deployments i services del cluster.
- Gestionar el cluster:
  - Pot esborrar o reiniciar deployments/pods/services
  - Pod modificar/assignar/eliminar nodes.
  - Permet escalar/desescalar pods
  - Reassignar pods entre nodes.
  - etc...

b) **Prometheus** és una eina de recollida de mètriques del sistema com per exemple la utilització de la CPU, la memòria RAM, l'espai de disc, l'accés a xarxa, etc..Aquestes dades són emmagatzemades dins d'una base de dades gestionada per l'aplicació i poden ser exportades a altres aplicacions per poder ser visualitzades de forma còmoda per un administrador del sistema.

c) **Grafana** és una eina que permet visualitzar les dades recollides per **Prometheus** d'una manera que sigui fàcil de visualitzar per l'administrador. Permet la creació de diversos tipus de gràfiques i la creació de taulers de control fets a mida per visualitzar dins d'una única pantalla totes les gràfiques que volguem visualitzar.

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

a) Posa en marxa la màquina **k8s** que vas crear a l'activitat **gp1f4act9.5**. Instal·la i habilita **Dashboard** de **Kubernetes** executant dins de la màquina virtual **k8s**: `microk8s enable dashboard`

b) Executa `microk8s status --format short | grep dashboard` i comprova que el **Dashboard** ha estat habilitat executant. Hauria de sortir el missatge: `core/dashboard: enabled`

c) Crea un directori de nom **dashboard**. Accedeix a la carpeta creada i dins de la carpeta crea un token d'accés que serà posteriorment necessari per accedir al **Dashboard** de **kubernetes** executant:

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

**d)** Per fer accessible des de la màquina virtual **k8s** el **Dashboard** de **kubernetes** 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:

```
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
```

i comprova que **kubectl** ha creat un procés que es comporta com un servidor que està en estat **LISTEN** i escoltant per l'adreça i port tcp **0.0.0.0:10443**.

**f)** Troba l'adreça IP de la màquina virtual **k8s**. Executa: **ip -4 -br add show eth1**

**g)** Des del navegador **Firefox** de la teva **màquina física** estableix una connexió a **https://ip\_màquina\_k8s:10443** a on **ip\_màquina\_k8s** és l'adreça trobada a l'apartat anterior. Accepta el risc potencial de seguretat anant a: *Avançat* → *Accepto el risc i vull continuar*.

**h) Autenticat dins del Dashboard:**

- Selecciona l'opció **Token** dins del formulari
- Escriu el token que vas trobar a l'apartat **d)** a la secció **Enter token \***.
  - Si el token és vàlid i no ha expirat accediràs al **Dashboard** de **kubernetes**.
  - 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 les aplicacions desplegades sobre el cluster de Kubernetes**

**a)** Accedeix a Workloads → Deployment. Comprova que els 2 deployments funcionen correctament i que el número de Pods en marxa és correcte per cada deployment.

**b)** Accedeix a Workloads → Pods. Comprova:

- La quantitat total de Pods i que tots funcionen correctament.
- L'utilització total de CPU del conjunt de tots els Pods expressat en cores.
- L'utilització total de la RAM per part del conjunt de Pods expressat en MiB.
- Quina aplicació utilitza Pods que requereixen més memòria RAM?

**c)** Accedeix a Service → Services. Comprova:


- Que el servei **serv-calc** funciona correctament
- Que el servei **serv-llaunes** funciona correctament

**d)** Accedeix a Cluster → Node. Comprova que el node **k8s** funciona correctament

**e)** Accedeix a Cluster → Node → **k8s**. Comprova:

- El % d'utilització dels cores del node **k8s**.
- El % d'utilització de RAM del node **k8s**

## 4- Escalament i desescalament manual dels Pods de les aplicacions

- a) Escala l'aplicació desplegada via **dep-llaunes** manualment de manera que treballi amb 20 Pods:
- Accedeix a Workloads → Deployment.
  - Selecciona Actions  → Scale de l'aplicació dep-llaunes.
  - Demana treballar amb 20 repliques dins de la secció Desired replicas.
- b) Torna Workloads → Deployments i comprova que ara dep-llaunes treballa amb 20 Pods.
- c) Ara fes que les 2 aplicacions treballin només amb 2 Pods.

**NOTA:** L'escalament automàtic està més enllà del temari d'aquest Global Project 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

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

Accedeix a Workloads → Pods i a continuació:

- Selecciona Actions  → del primer de l'aplicació llaunes i comprova el seu codi hash
- Escull l'opció Delete
- Comprova que el pod deixa de funcionar.
- Comprova que passats uns segons es posa en marxa un nou Pod amb un nom comú igual però un codi hash diferent.

## 6- Monitorització del sistema amb Grafana i Prometheus

a) Crea una carpeta de nom **monitoritza** dins de la màquina **k8s**. Accedeix a la carpeta **monitoritza** i des dins de la carpeta clona el dipòsit <https://github.com/stefanprodan/dockprom> de **GitHub**. Executa l'ordre:

```
git clone https://github.com/stefanprodan/dockprom
```

i comprova es crea una carpeta de nom **dockprom**.

b) Accedeix a la carpeta **dockprom** i posa en marxa l'aplicació de monitorització del sistema basada en **docker** utilitzant les eines **Grafana** i **Prometheus**. Executa:

```
docker compose up -d
```

c) Accedeix a l'aplicació des de la teva màquina física amb el navegador fent una connexió a l'adreça IP de la màquina **k8s** i al port **3000** que és per on escolta **Grafana**.

d) Valida't dins a l'aplicació utilitzant el nom d'usuari **admin** i la contrasenya **admin**. Escribeu un nou password per l'aplicació (proposta: **fjeclot**)

- e) Accedeix al menú de la dreta i selecciona Administration → Data sources → Prometheus → clic a default → Save & test. Ara **Prometheus** és la font de mètriques per defecte de **Grafana**.
- f) Selecciona Dashboard → General → Docker Host
- g) Comprova que tens accés a les mètriques principals del sistema (CPU, RAM, .....

### **Lliurament de l'activitat**

- a) Accés a Dashboard.
- b) Visualitza l'utilització total de CPU i RAM del cluster de kubernetes.
- c) Escala a 3 la quantitat de Pods de les dues aplicacions desplegades sobre kubernetes.
- d) Atura i esborra el primer Pod de l'aplicació calculadora i comprova que automàticament es crea un altre.
- e) Mostra les mètriques d'utilització del sistema amb Grafana i Prometheus.
- f) Data límit per obtenir el 100% de la nota: **dimarts 5-12-22 a les 20.05**.