

Fase 4 - Activitat 10: Monitorització de sistemes utilitzant les eines Prometheus i Grafana desplegadas utilitzant contenidors Docker

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 10

Grup/Individual: Individual

Membres/Alumne:

1- Objectius de l'activitat 10

- a) Desplegament de les eines Prometheus i Grafana utilitzant contenidors Docker
- b) Monitorització bàsica del sistema: CPU, RAM, Disc i Xarxa.

2- Conceptes a desenvolupar dins d'aquesta activitat

- a) Algunes idees sobre monitorització:

- Monitoritzar el sistema consisteix en fer un seguiment continuat de l'estat dels recursos d'un equip i del seu sistema o del conjunt d'equips d'una xarxa.
- El propòsit de monitoritzar permet comprovar si el sistema funciona correctament, si està al límit i cal actualitzar-lo, comprovar si hi han anomalies que provoquen un mal funcionament d'algunes de les seves parts, millorar la configuració del sistema/aplicacions/serveis/maquinari, etc...
- Normalment es monitoritzen:
 - Servidors i equips clients (RAM, CPU, Disc, Xarxa...)
 - Serveis (Apache, SSH, DHCP, DNS, etc...)
 - Mòbils/Tauletes
 - Impressores
 - Hubs/Switch/Routers
 - SAIsEn general, qualsevol equip informàtic o qualsevol programa pot ser monitoritzat.
- Habitualment caldrà monitoritzar, com a mínim, l'utilització de la CPU i la quantitat de memòria RAM utilitzada i lliure. També és típic monitoritzar l'estat dels dispositius d'emmagatzematges i l'estat de les connexions amb la xarxa.
- Per monitoritzar s'utilitzen **mètriques**. Una mètrica són dades sobre l'utilització d'un determinat recurs. Per exemple, el % de memòria RAM lliure, el % de temps de CPU utilitzats, el número d'usuaris connectats a un sistema, el temps de carrega d'una pàgina web, les vegades que s'ha accedit a un fitxer d'una carpeta compartida, etc...
- Per monitoritzar el sistema hem de:
 - Recollir mètriques
 - Emmagatzemar mètriques
 - Serialitzar temporalment les dades
 - Visualitzar les mètriques amb gràfiques, taules i indicadors.
 - Configurar alertes que permetin donar un avís a l'administrador si les mètriques estan per sobre o per sota d'un determinat nivell. Aquestes alertes poden ser configurables.

- Dins d'una xarxa ens podem trobar que un equip s'encarregui de recollir, emmagatzemar i serialitzar les mètriques enviades per altres equips de la xarxa. També és normal que aquest equip ens doni les eines per visualitzar-les i s'encarregui de les alarmes.
- Per poder monitoritzar sistemes ens caldrà instal·lar dins dels equips el programari necessari per recollir, emmagatzemar, serialitzar i visualitzar mètriques, i configurar alertes.
- Dins d'una xarxa ens podem trobar que aquesta feina estigui repartida entre uns programes que només recullen mètriques dins de cada equip local i les envien a un altre equip que tindrà el programari necessari per recollir les dades de tota la xarxa, emmagatzemar-les, serialitzar-les, visualitzar-les i encarregar-se de gestionar les alertes.

b) Prometheus:

- És una eina de recollida, emmagatzematge, serialització i visualització de mètriques del sistema com per exemple la utilització de la CPU, la memòria RAM, l'espai de disc, l'accés a xarxa, etc.
- Aquesta eina també permet generar alarmes.
- Prometheus pot recollir mètriques d'altres sistemes informàtics o programes de la xarxa amb l'ajut d'un programari anomenat **exporter** necessari per recollir-les les mètriques i enviar-les.
- El sistema amb Prometheus també necessita el seu **exporter** per poder veure les pròpies mètriques.
- Permet exportar i compartir les dades a altres aplicacions especialitzades amb millor eines de visualització o anàlisi de les dades.

c) Grafana és una eina que permet visualitzar les dades recollides per **Prometheus** (o altres eines de monitorització similar) d'una manera que sigui fàcil de visualitzar per l'administrador. Permet la creació i personalització de Dashboards (taulers de control) que tinguin múltiples gràfiques de diversos formats, indicadors de tots tipus, i taules fetes a mida que permeten a l'administrador visualitzar d'una manera clara totes les dades necessàries per monitoritzar un sistema o una xarxa. També permet generar i configurar alertes del sistema o la xarxa.

d) És molt habitual combinar:

- Prometheus per recollir, emmagatzemar i serialitzar mètriques.
- Grafana per visualitzar mètriques i generar alertes.

3- Monitorització del sistema amb Grafana i Prometheus

3.1- Creació dels sistemes a monitoritzar

a) Crea una carpeta de nom **a10** dins de la carpeta **f4** que es troba a **pj9**. A continuació, dins de la carpeta **a10** crea una carpeta de nom **pj9f4a10**.

b) Dins de **pj9f4a10** copia el fitxer **Vagrantfile** amb que vas utilitzar per la pràctica **pj9f4a9.3** i fes els següents canvis de configuració:

- Utilitzarem el box **debian/bookworm64**.
- Treballarem amb **virtualbox** com a eina de virtualització (provider).
- El nom base del sistema i l'identificador dins de VirtualBox dels nodes serà **pj9f4a10**.
- El nom de domini dels nodes serà **clotfje.net**
- La RAM dels nodes serà **2048MB**
- Les CPUs assignades als nodes seran **1**
- Crearem **2** nodes
- Seleccionarem la targeta de xarxa WiFi a partir del seu identificador dins de VirtualBox.

c) Executa:

- **vagrant box update**
- **vagrant up**

i posa en marxa les màquines virtuals.

- d) Executa `vagrant status` i comprova s'han creat les màquines **pj9f4a101** i **pj9f4a102**.
- e) Accedeix a les màquina virtual **pj9f4a101**. Executa: `vagrant ssh pj9f4a101`
- f) Comprova dins de **pj9f4a101** que:
- L'usuari **vagrant** es membre del grup **docker**. Executa: `id vagrant`
 - El **docker** està disponible. Executa: `doker -v` (la versió hauria de er 27.4.1 o posterior).
 - Els nom de sistema del **pj9f4a101**. Executa: `hostname --fqdn`. El nom serà **pj9f4a101.clotfje.net**.
 - Comprova l'adreça IP de la interfície **eth1** executant: `ip addr show eth1`
- e) Surt de la màquina **pj9f4a101** i accedeix a les màquina virtual **pj9f4a102**. Executa: `vagrant ssh pj9f4a102`
- f) Comprova dins de **pj9f4a102** que:
- L'usuari **vagrant** es membre del grup **docker**. Executa: `id vagrant`
 - El **docker** està disponible. Executa: `doker -v` (la versió hauria de er 27.4.1 o posterior).
 - Els nom de sistema del **pj9f4a102**. Executa: `hostname --fqdn`. El nom serà **pj9f4a102 node921.clotfje.net**.
 - Comprova l'adreça IP de la interfície **eth1** executant: `ip addr show eth1`

3.2- Desplegament de les eines de monitorització Prometheus i Grafana amb contenidors Dockers sobre el sistema pj9f4a101. Visualització de mètriques bàsiques del sistema pj9f4a101

- a) Crea una carpeta de nom **monitoritza** dins de la màquina **pj9f4a101**. Accedeix a la carpeta **monitoritza** i des dins de la carpeta clona el dipòsit <https://github.com/stefanprodan/dockpromdins> de **Github**. Executa l'ordre:

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

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

- b) Accedeix a la carpeta **dockprom**. A continuació, desplega i posa en marxa les eines de monitorització **Grafana** i **Prometheus** per mitjà de contenidors. Executa:

```
docker compose up -d
```

- c) Comprova que els contenidors de **Prometheus**, **Grafana** estan funcionant. Executa:

```
docker ps -a | grep "prometheus" → per comprovar que el docker de prometheus funciona  
docker ps -a | grep "grafana" → per comprovar que el docker de grafana funciona
```

- d) Comprova també que el node exporter de la màquina està en marxa per poder visualitzar dades de la màquina **pj9f4a101**. Executa:

```
docker ps -a | grep "nodeexporter"
```

i comprova que està **nodeexporter** en marxa.

- e) Accedeix a **Grafana** des de la teva màquina física amb el navegador fent una connexió a l'adreça IP de la màquina **pj9f4a101** i al port **3000** que és per on escolta **Grafana**.

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

g) Accedeix a:

- *Dashboards* → *New* → *Import*
- A l'entrada **LOAD** indica que vols importar el dashboard 1860 i després fes clic a **LOAD**.
- A la següent pàgina, a l'entrada **Prometheus**, selecciona **Prometheus default** i fes clic a **IMPORT**
- Comprova que es carrega el Dashboard per veure mètriques del sistema local.
- Configura el Dashboard per veure les mètriques dels darrers 5 minuts i refrescar la pàgina cada 5 segons.

3.3- Desplegament de node exporter sobre el sistema pj9f4a102 i exportació de les seves mètriques

a) Crea una carpeta de nom **exporter** dins de la màquina **pj9f4a102**. Accedeix a la carpeta **exporter** i des dins de la carpeta clona descarrega una imatge per desplegar un contenidor amb **node exporter**. Executa l'ordre:

```
docker pull prom/node-exporter
```

b) Desplega un contenidor de node exporter. Executa:

```
docker run --name nodeexporter -i -t -d -p 9100:9100 prom/node-exporter
```

Ara **pj9f4a102** ja està exportant les seves mètriques.

3.4- Configura Prometheus de pj9f4a101 per recollir i emmagatzemar les mètriques de pj9f4a102

a) Troba l'adreça IP de **pj9f4a102**.

b) Accedeix a **pj9f4a101** i entra dins **monitoritza** → **dockprom** → **prometheus**.

c) Modifica el fitxer **prometheus.yml**. A la secció **scrape_configs** afegeix un nou **job** amb la següent configuració:

- **job_name** -> **pj9f4a102**
- **scrape_interval** -> **5 segons**
- **target** → Utilitza l'adreça **IP** de **pj9f4a102** i el port **9100**

d) Reinicia **Prometheus**. Torna a **monitoritza** → **dockprom** i executa:

```
docker compose restart
```

3.5- Visualització de mètriques bàsiques del sistema pf9f4a102

a) Accedeix a **Grafana** des de la teva màquina física amb el navegador fent una connexió a l'adreça IP de la màquina **pj9f4a101** i al port **3000** que és per on escolta **Grafana**.

b) Comprova que a la secció **Jobs** en tens **2**:

- **nodeexporter** → mètriques de **pj9f4a101**
- **pj9f4a102** → Mètriques de **pj9f4a102**

c) Comprova que pots visualitzar les mètriques de **pj9f4a101** i **pj9f4a102**. En tots dos casos, configura el Dashboard per veure les mètriques dels darrers 5 minuts i refrescar la pàgina cada 5 segons.

4- Proves d'estrès i monitorització dels sistema

a) Accedeix a **pj9f4a102** i instal·la el programa **stress**. Executa:

```
sudo apt-get install stress
```

b) Estresa la **CPU** del sistema **pj9f4a102** durant **15 segons** amb **4 processos** que fan càlculs matemàtics complexos. Executa:

```
sudo stress -c 4 -t 15
```

c) Comprova amb Grafana com s'incrementa el valor de **CPU Busy**, **Sys Load** i **Pressure** de la **CPU**.

d) Estresa l'ús de memòria del sistema **pj9f4a102** durant **20 segons** amb **1 procés** que fa ús intensiu de la memòria. Executa:

```
sudo stress -m 1 -t 20
```

e) Comprova amb **Grafana** com s'incrementa el valor de **RAM Used** i **Pressure** de la **RAM**.

Lliurament de l'activitat

a) Mostra els contenidors funcionant dins de **pj9f4a101** i **pj9f4a102**.

b) Accés per mitjà de **Grafana** als dashboards de **pj9f4a101** (mostrada com **nodeexporter**) i **pj9f4a102**.

c) Fes l'estrès de **CPU** de **pj9f4a102** indicat a la pràctica (apartat 4) i comprova que **Grafana** s'actualitza.

d) Fes l'estrès de **Memòria** de **pj9f4a102** indicat a la pràctica (apartat 4) i comprova que **Grafana** s'actualitza.

e) Data límit per obtenir el 100% de la nota: **dimarts 23-1-25** a les **17.45**.