

Fase 4 - Activitat 11: 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: 11

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 **pj9f4a11** per fer aquesta activitat.

b) Dins de **pj9f4a11** copia el fitxer **Vagrantfile** que pots trobar aquí:

<https://raw.githubusercontent.com/asix2pj9/pj9f4a11/refs/heads/main/Vagrantfile>

i fes els següents canvis de configuració:

- Fes que el nom de sistema de la primera màquina sigui: **Operacions**
- Fes que el nom de sistema de la segona màquina sigui: **Monitoritza**
- La RAM assignada a tots dos equips serà de **2048MB**
- Les CPUs assignades a tots dos equips seran **2**
- Tots dos equips treballaran amb **adaptador pont** i amb la targeta **WiFi** de l'equip físic.

c) Posa en marxa les màquines virtuals **Operacions** i **Monitoritza**.

- d) Accedeix a les màquina virtual **Operacions** i comprova que:
- L'usuari **vagrant** es membre del grup **docker**. Executa: `id vagrant`
 - El **docker** està disponible. Executa: `docker -v` (la versió hauria de ser 20.10.0 o posterior).
 - Els nom de sistema executant: `hostname --fqdn`.
 - Comprova l'adreça IP de la interfície **eth1** executant: `ip addr show eth1`
- e) Surt de la màquina **Operacions** i accedeix a les màquina virtual **Monitoritza**. Comprova que:
- L'usuari **vagrant** es membre del grup **docker**. Executa: `id vagrant`
 - El **docker** està disponible. Executa: `docker -v` (la versió hauria de ser 20.10.0 o posterior).
 - Els nom de sistema executant: `hostname --fqdn`
 - 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 Monitoritza. Visualització de mètriques locals del sistema Monitoritza

- a) Accedeix a la màquina **Monitoritza** i clona el dipòsit que trobarà a la següent URL de Github:

<https://github.com/stefanprodan/dockprom>

- b) Comprova que es crea una carpeta 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 compose ps -a | grep prometheus` → per comprovar que el docker de prometheus funciona
`docker compose 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 **monitoritzacio**. Executa:

`docker compose 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 **Monitoritza** 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 amb ID **1860** i després fes clic a **LOAD**.
- A la pàgina següent fes clic a **IMPORT**.
- Comprova que es carrega un **Dashboard** que permet veure mètriques del sistema local.
- Configura el **Dashboard** per veure les mètriques dels darrers **5 min.** i refrescar la pàgina cada **5s**.

- h) Comprova que pots veure les mètriques locals bàsiques (CPU, RAM, SWAP, utilització del disk, trànsit de xarxa, etc.....) del sistema **Monitoritza**.

3.3- Desplegament de node exporter sobre Operacions i exportació de les seves mètriques

a) Accedeix a la màquina **Operacions**. Descarrega una imatge per poder desplegar un contenidor amb **node exporter** que permetrà enviar **mètriques** de la màquina **Operacions** a la màquina **Monitoritza**. Executa:

```
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
```

c) Comprueba que s'ha creat el contenidor **nodeexporter** i que està funcionant. Si és així, ara el sistema **Operacions** ja està exportant les seves mètriques i el sistema **Monitoritza** pot recollir-les amb **Prometheus** i mostrar-les amb **Grafana**.

3.4- Configura Prometheus de Monitoritza per recollir i emmagatzemar les mètriques d'Operacions

a) Accedeix a la màquina **Monitoritza** i entra dins **dockprom** → **prometheus**.

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

- **job_name** -> **metOperacions**
- **scrape_interval** -> **5 segons**
- **target** → Utilitza l'adreça IP de la màquina **Operacions** i el port **9100**

c) Reinicia **Prometheus**. Per fer-ho, has de tornar a **dockprom** i executar: **docker compose restart**

3.5- Visualització de mètriques bàsiques del sistema Operacions

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

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

- **nodeexporter** → mètriques de la màquina **Monitoritza**
- **metOperacions** → Mètriques de la màquina **Operacions**

c) Comprova que pots visualitzar les mètriques de **Monitoritza** i **Operacions**. 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 **Operacions** i instal·la el programa **stress**. Executa:

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

b) Estresa la **CPU** del sistema **Operacions** durant **15 segons** amb **4 processos** que fan càlculs matemàtics complexes. 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** del sistema **Operacions**.

d) Estresa l'ús de memòria del sistema **Operacions** durant **30 segons** amb 2 **processos** que fan un ús intensiu de la memòria. Executa:

```
sudo stress -m 2 -t 30
```

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

Lliurament de l'activitat

a) Accés per mitjà de **Grafana** al **Dashboards** de **Monitoritza** (mostrada com **nodeexporter**) i comprovació del seu estat actual d'utilització de RAM, CPU, Disc i Xarxa.

b) Accés per mitjà de **Grafana** al **Dashboards** d'**Operacions** (mostrada com a **metOperacions**) i comprovació del seu estat actual d'utilització de RAM, CPU, Disc i Xarxa.

c) Fes l'estrès de **CPU** de la màquina **Operacions** indicat a la pràctica i comprova que **Grafana** s'actualitza.

d) Fes l'estrès de **Memòria** de la màquina **Operacions** indicat a la pràctica i comprova que **Grafana** s'actualitza.

d) Data límit per obtenir el 100% de la nota: **dimarts 2-2-26** fins a les **19.10**.