

Activitat sm5a3.1: Instal·lació, configuració, comunicació i avaluació d'aplicacions utilitzant arquitectures heterogènies – Part I

Objectius

Aquesta pràctica té com a primer objectiu, el desenvolupament, desplegament i accés a una aplicació web desplegades utilitzant contenidors Docker.

La primera part de la pràctica consistirà en desplegar i accedir a una aplicació utilitzant contenidors Docker. Per fer això:

- Caldrà instal·lar el programari de contenidors Docker en el vostre sistema.
- S'haurà de crear una imatge de l'aplicació.
- S'haurà de crear el contenidor.
- S'haurà d'accedir a l'aplicació des d'un altre equip de la xarxa amb arquitectura x86
- S'haurà d'accedir a l'aplicació des d'un altre equip de la xarxa amb arquitectura ARM

El segon objectiu és comunicar 2 equips d'arquitectures diferents, utilitzant el protocol MQTT. Per fer això, ens caldrà:

- Instal·lar el programari MQTT dins del nostre equip
- Instal·lar un programari per rebre missatge MQTT dins d'un equip client
- Fer proves de funcionament de la comunicació.

Part 1 – Desplegament d'una aplicació amb contenidors Docker

a) Instal·la el programari de contenidors Docker. Dins de l'activitat pj9f4a7.1 del projecte PJ9 – DevOps i Cloud Computing, en el punt 3.c hi ha les instruccions per instal·lar aquest programari.

b) Descarrega el codi de l'aplicació de càlcul de Llaunes de Begudes tal i com s'explica a l'activitat pj9f4a7.6 del projecte PJ9 – DevOps i Cloud Computing, en els punts 2.a i 2.b.

c) Desplega que l'aplicació utilitzant com a referència la informació que trobaràs a l'activitat pj9f4a7.6 del projecte PJ9 – DevOps i Cloud Computing, en el punt 3.

d) Comprova que l'aplicació funciona, utilitzant com a referència la informació que trobaràs a l'activitat pj9f4a7.6 del projecte PJ9 – DevOps i Cloud Computing, en el punt 4.

e) Crida'm per comprovar que l'aplicació funciona.

Part 2 – Comunicacions entre equips d'arquitectura heterogènia

a) Actualitza el sistema:

```
sudo apt update
sudo apt upgrade -y
```

b) Instal·la el programari:

```
sudo apt install mosquitto mosquitto-clients -y
```

c) Configura el servei perquè es posi en marxa durant l'arrancada del sistema:

```
sudo systemctl enable mosquitto
```

d) Configura el broker mosquito per poder escoltar per qualsevol adreça IP. Afegeix al final de `/etc/mosquitto/mosquitto.conf` el paràmetre `bind_address` amb el valor `0.0.0.0`. Després reinicia **mosquitto**.

e) Crea una contrasenya pel teu usuari grupXX a on XX és el teu número de grup. Executa:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd grupXX
```

f) Afegeix els següents paràmetres de configuració a `/etc/mosquitto/mosquitto.conf`:

```
allow_anonymous true <ip_address_del_teu_equip>
password_file /etc/mosquitto/passwd
```

Després reinicia **mosquitto**.

e) Comprova l'adreça IP de l'equip.

f) Instal·la **MyMQTT** dins del teu mòbil.

g) Configura **MyMQTT** per connectar-se amb l'adreça **IP** i **port** del servidor **mosquitto** del teu equip informàtic. Després subscriu-lo al tema "test/topic".

h) Des del servidor executa:

```
mosquitto_pub -h <ip_servidor> -t "test/topic" -m "Hola a tothom" -u <usuari> -P
<password>
```

i) Des de MyMQTT comprova que s'ha rebut el missatge "Hola a tothom"

Comprovacions

- a) Funcionament i control de l'aplicació CostLlaunaBegudes
- b) Enviament de missatges via **MQTT**
- e) Lliurament: **20-2-26**

ANNEX

Què és MQTT?

Definició

MQTT (Message Queuing Telemetry Transport) és un **protocol de missatgeria lleuger** dissenyat per a comunicació **publicador/subscriptor** en xarxes amb baixa amplada de banda, alta latència o entorns IoT.

- Inicialment creat per a dispositius amb recursos limitats i xarxes inestables.
- Permet enviar missatges petits i eficients entre dispositius connectats.

Principis bàsics

1. Broker

- És el servidor central que gestiona tots els missatges.
- Rep els missatges dels **publicadors** i els envia als **subscriptors** corresponents.

2. Publicador (Publisher)

- Dispositiu que envia missatges a un **topic** concret.

3. Subscriptor (Subscriber)

- Dispositiu que rep missatges d'un topic al que s'ha subscrit.

4. Topic

- Nom del canal de comunicació, jeràrquic, per exemple: casa/sala/temperatura.

Característiques clau

- **Lleuger i eficient** → ideal per IoT i xarxes limitades.
- **Protocol publicador/subscriptor** → els dispositius no necessiten coneixes's directament.
- **QoS (Quality of Service)** → tres nivells per garantir la entrega de missatges:
 - 0 → al màxim una vegada, sense confirmació
 - 1 → almenys una vegada
 - 2 → exactament una vegada
- **Retenció de missatges** → opcional, el broker pot guardar l'últim missatge enviat a un topic per nous subscriptors.
- **Keep Alive** → comprova que els clients segueixen connectats.

Ús típic

- **Domòtica:** sensors de temperatura, llums intel·ligents.
- **IoT industrial:** control i monitoratge de màquines.
- **Aplicacions mòbils:** notificacions en temps real.

Esquema