

## Fase 4 - Activitat 11.4: CI/CD - Part IV: Preparant l'entorn de producció

### 0- Identificació del grup i activitat:

**Curs:** ASIX2

**Projecte:** GP2 DevOps i Cloud Computing

**Fase:** 4

**Activitat:** 11.4

**Grup:**

**Membres:**

### 1- Introducció i objectius de l'activitat 11.4

a) Lectura de les especificacions de l'activitat.

### 2.- Màquina pel desenvolupament d'aplicacions i fer proves d'integració continua

a) Dins de la carpeta de nom **gp1f4act11** de la teva màquina física una carpeta de nom **vm\_produccio**. Accedeix a **vm\_produccio** i crea el següent fitxer **Vagrantfile**:

```
IMATGE_BOX = "debian/bookworm64"
NOM_NODE = "produccio"
MEMORIA = 2048
CPUS = 2
TARGETA_XARXA = "xxxxxxxx"

Vagrant.configure("2") do |config|
  config.vm.box = IMATGE_BOX
  config.vm.hostname = NOM_NODE
  config.vm.network "public_network", bridge: TARGETA_XARXA
  config.vm.provider "virtualbox" do |v|
    v.name = NOM_NODE
    v.memory = MEMORIA
    v.cpus = CPUS
    v.customize ['modifyvm', :id, '--clipboard', 'bidirectional']
  end

  config.vm.provision "shell", inline: <<-SHELL
  sudo apt-get update -y
  sudo apt-get install -y net-tools
  sudo apt-get install -y whois
  sudo apt-get -y install apt-transport-https ca-certificates curl gnupg2 software-properties-common
  curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
  sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
  sudo apt-get update -y
  sudo apt-get -y install docker-ce docker-ce-cli containerd.io docker-compose
  sudo gpasswd -a vagrant docker
  exit
SHELL
end
```

A on hauràs de canviar **xxxxxxxx** pel nom de la teva targeta de xarxa dins del teu equip físic.

### 3.- Comprovant que es pot fer disponible de manera pública una aplicació desplegada amb Dockers sobre la màquina de producció local utilitzant

#### 3.1- Desplegant l'aplicació

a) Fes la configuració inicial de **git** dins de la màquina virtual **produccio**. Executa (sense sudo):

```
git config --global user.email "xxxxxxx" # "xxxxxxx" és el correu del teu compte de Github  
git config --global user.name "yyyyyyy" # "yyyyyyy" és el teu nom d'usuari de Github
```

b) Dins de la màquina virtual de **producció** clona el projecte **ipcalc** executant:

```
git clone https://github.com/globproj2/ipcalc.git
```

c) Entra dins de la carpeta **ipcalc** que s'ha creat dins de la teva màquina virtual i inicia l'aplicació executant l'ordre: **docker-compose up -d**

d) Comprova que s'ha creat la imatge de nom **ipcalc\_ipcalc:latest**, el contenidor **ipcalc\_ipcalc\_1** i el contenidor **ipcalc\_lb\_1**.

e) Comprova l'adreça IP de la interfície **eth1** de la màquina virtual.

g) Accedeix des de la màquina física a l'aplicació amb l'adreça IP trobada al punt anterior.

#### 3.2- Fes l'aplicació disponible públicament amb ngrok

a) Surt de la carpeta **ipcalc**. Crea una carpeta de nom **ngrok**, accedeix a la carpeta i executa la següent ordre per descarregar l'aplicació:

```
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
```

b) Descomprimeix i desempaqueta l'aplicació. Executa:

```
tar xfz ngrok-v3-stable-linux-amd64.tgz
```

c) Mou el program **ngrok** al directori **/usr/local/bin**. Executa: **sudo mv ngrok /usr/local/bin**

d) Comprova que funciona. Executa: **ngrok --version** (la versió hauria de ser la 3.1.0 o superior)

e) Dins del teu compte de **ngrok**, accedeix a la secció **Your Authtoken**.

f) Dins de la carpeta **ngrok** de la teva màquina **produccio** executa:

```
ngrok config add-authtoken xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

a on xxx representa el teu token.

g) La teva aplicació **ipcalc** escolta pel port **80/tcp** local de la màquina física, de manera que hauràs d'exposar aquest port a internet. Executa:

```
ngrok http 80
```

h) Accedeix des de la teva màquina física a l'aplicació **ipcalc** utilitzant l'adreça URL pública proporcionada per **ngrok**.

#### **4.- Configuració de l'accés la màquina virtual via SSH des de qualsevol altra màquina.**

a) Modifica el paràmetre **PasswordAuthentication** del fitxer **/etc/ssh/sshd\_config** de la màquina de **produccio** al valor **yes**. Reinicia el servei ssh executant: **sudo systemctl restart ssh**

b) Fes que el password de l'usuari **vagrant** de la màquina de **produccio** sigui **clotfje**.

c) Comprova que des de la màquina **pipelinecid** pots accedir a la màquina **produccio** via **SSH** amb el compte d'usuari **vagrant**.

#### **5.- Accés directe via SSH des de la màquina pipelinecid a la màquina de producció.**

a) Executa dins de la màquina **pipelinecid** l'ordre:

```
ssh-keygen -t rsa
```

i crea una clau pública per l'usuari **vagrant** de **pipelinecid**. El nom d'aquesta clau serà **id\_rsa.pub** i es desarà al directori **.ssh** de l'usuari. No ha de tenir passphrase. Per tant, a totes les preguntes s'ha de premer Enter.

b) Copia la clau pública de l'usuari **vagrant** de **pipelinecid** dins del directori **.ssh** de l'usuari **vagrant** de **produccio** executant:

```
scp id_rsa.pub vagrant@xxxxxxxx:~/ssh/authorized_keys2
```

a on **xxxxxxxx** és l'adreça IP de la màquina en **producció**.

c) Comprova que dins del **.ssh** de l'usuari **vagrant** de **produccio** s'ha creat el fitxer **authorized\_keys2**

d) Comprova que l'usuari **vagrant** de de la màquina **pipelinecid** potsaccedir via **SSH** al compte de l'usuari **vagrant** de la màquina en **produccio** via **SSH** directament sense que es demani la contrasenya perquè ara s'identifica amb la seva clau pública .

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

a) Accedeix a l'aplicació des de la màquina física a l'aplicació utilitzant la URL pública de **produccio**.

b) Comprovació d'accés via **SSH** des de la màquina **pipelinecid** a la màquina **produccio**.

c) Comprovació de l'accés directe via SSH des de la màquina **pipelinecid** a la màquina **produccio**.

d) Data límit per obtenir el 100% de la nota: **dimecres 9-1-24 a les 19.10**.