

## Fase 4 - Activitat 11.4: Actualització automàtica de l'aplicació disponible pels usuaris a la màquina de producció.

### 0- Identificació del grup i activitat:

**Curs:** ASIX2

**Projecte:** PJ9 DevOps i Cloud Computing

**Fase:** 4

**Activitat:** 11.4

**Grup/Individual:** Individual

**Membres:**

### 1.- Objectiu de l'activitat

L'objectiu d'aquesta activitat és que cada nova versió de l'aplicació pizzes desenvolupada dins de la màquina **desenvolupament** i que es puja a **Github** es desplegui automàticament sobre la màquina **producció** i per fer-la disponible immediatament als usuaris utilitzant **ngrok**, **Jenkins** i **Webhooks de Github**. Per poder fer aquesta darrera part aprofitarem que Jenkins permet executar ordres o scripts cada cop que es descarrega automàticament una nova versió de l'aplicació per posar en marxa un script que executarà les ordres necessàries per desplegar l'aplicació i fer-la disponible als usuaris.

### 2.- Abans de començar

a) Posa en marxa les màquines virtuals **desenvolupament** i **producció**.

b) Accedeix a la màquina **desenvolupament** i modifica el contingut de **pizza.php**. Fes que tingui el següent codi:

```
<html>
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="content-type">
    <title>FES LA TEVA PIZZA</title>
  </head>
  <body>
    <?php
      $preu = 8;
      $IVA=21;
      if (empty($_GET['ingr'])) {
        $pvp=((100+$IVA)/100)*$preu;
        echo "Has demanat només la pizza bàsica. El preu és $pvp € (IVA inclòs)<br>";
        exit(0);
      }
      else {
        $IVA=21;
        $pvp=((100+$IVA)/100)*($preu+count($_GET['ingr']));
        echo "Has demanat la pizza bàsica amb:<br> ";
        foreach ($_GET['ingr'] as $ingredient) {
          if ($ingredient) {
            echo "*" $ingredient<br>";
          }
        }
        echo "El preu és $pvp € (IVA inclòs)<br>";
        exit(0);
      }
      echo "<br>";
      echo "Autor: dacomio."<br>";
    ?>
  </body>
</html>
```

c) Dins de la màquina **producció** entra a la carpeta **ngrok** i executa:

```
ngrok http 8080
```

per aconseguir fer pública l'aplicació **Jenkins** de la teva màquina virtual.

d) Actualitza el dipòsit local de la màquina **desenvolupament** i sincronitza-ho amb el dipòsit remot a **Github**. Dins de la carpeta **pizzas** de la màquina **desenvolupament** executa:

```
git add *
git commit -m "Commit 4 de l'aplicatiu pizzas"
git push -u origin main
```

i comprova que s'ha pujat la nova versió de **pizzas** a **Github**.

e) Comprova que **Jenkins** descarrega la nova versió a la màquina **produccio**.

### **3- Modificació de Vagrantfile per afegir docker a les màquines desenvolupament i produccio**

a) Modifica la secció d'aprovisionament de les màquines per afegir el programari de contenidors. Fes que la secció d'aprovisionament sigui:

```
node.vm.provision "shell", inline: <<- SHELL
  sudo apt-get update -y
  sudo apt-get install -y net-tools whois aptitude git zip unzip curl
  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
SHELL
```

b) Executa: **vagrant provision**

c) Comprova que tant la màquina **produccio** com la màquina **desenvolupament** tenen el programari **docker** instal·lat i que l'usuari **vagrant** és membre del grup **docker**.

### **4- Crea un contenidor a la màquina produccio per desplegar i fer disponibles als usuaris els nous codis conforme es facin actualitzacions**

a) A la màquina **produccio** crea una carpeta de nom **projectes**. Dins de **projectes** crea una carpeta de nom **pizzas**. Dins de **pizzas** crea un fitxer de nom **pizza.html** amb aquest contingut:

```
<html>
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="content-type">
    <title>FES LA TEVA PIZZA</title>
  </head>
  <body>
    L'aplicació encara no està disponible. Aviat podreu personalitzar la vostra pizza!!!!!!
  </body>
</html>
```

b) Dins de la màquina **produccio** descarrega la imatge de contenidor **php:8.3-apache**.

c) Crea un contenidor de nom **pizzas** dins de la màquina **produccio** a partir de la imatge **php:8.3-apache**, que treballi en **2n terme**, permeti accedir a l'interior del contenidor via **bash**, que exporti el port **80** del contenidor al port **8000** de la màquina i que comparteixi com un volum la carpeta **/home/vagrant/projectes/pizzas** de la màquina amb la carpeta **/var/www/html** del contenidor.

d) Troba l'adreça IP de la màquina **produccio**.

e) Comprova des de la màquina física que pots veure l'estat actual de l'aplicació.

## 5- Configura la màquina producció per fer disponible l'actualització als usuaris automàticament

a) Primer has de configurar la màquina **produccio** amb els permisos adequats perquè **Jenkins** pugui crear, modificar, esborrar i llegir dins de la carpeta **pizzas** i els seus fitxers:

- A la màquina **produccio**, fes que la carpeta **pizzas** tingui permisos de:
  - **lectura/escritura/execució** pel propietari **vagrant**
  - **lectura/escritura/execució** pel **grup vagrant**
  - **lectura/execució** per **others**

Executa: `chmod 775 /home/vagrant/projectes/pizzas`

- A la màquina **produccio**, fes que la carpeta **pizza.html** tingui permisos de:
  - **lectura/escritura** pel propietari **vagrant**
  - **lectura/escritura** pel **grup vagrant**
  - **lectura** per **others**

Executa: `chmod 664 /home/vagrant/projectes/pizzas/pizza.html`

- A la màquina **produccio**, fes a l'usuari **jenkins** membre del grup **vagrant** executant:

`sudo gpasswd -a jenkins vagrant`

b) Ara, s'ha de crear un script de bash que executarà cada cop que **Jenkins** descarregui una nova actualització del codi de l'aplicació:

- A la màquina **produccio**, crea una carpeta de nom **scripts** dins de la carpeta **projectes** i fes que tingui permisos:
  - **lectura/escritura/execució** pel propietari **vagrant**
  - **lectura/escritura/execució** pel **grup vagrant**
  - sense permisos per **others**

Executa: `chmod 770 /home/vagrant/projectes/scripts`

- Dins de **scripts** crea un fitxer de nom **pizza.sh** amb aquest contingut:

```
#!/bin/bash
cp /var/lib/jenkins/workspace/pizzas/*.html /home/vagrant/projectes/pizzas
cp /var/lib/jenkins/workspace/pizzas/*.php /home/vagrant/projectes/pizzas
exit 0
```

i després fes que **pizza.sh** tingui permisos:

- **lectura/escritura/execució** pel propietari **vagrant**
- **lectura/escritura/execució** pel **grup vagrant**
- sense permisos per **others**

Executa: `chmod 770 /home/vagrant/projectes/scripts/pizzas.sh`

c) A finalment, haurem de configurar Jenkins per executar el script creat a l'apartat anterior cada cop que descarregui una nova actualització

- Accedeix al projecte **pizzas** de l'eina **Jenkins** de la màquina **produccio**, i després accedeix a **Build Steps** → **Add build step** → **Executa shell**. A continuació:
- Escribeu el nom i directori complet del script que has creat a l'apartat anterior: **/home/vagrant/projectes/scripts/pizza.sh**.
- Fes clic a **Desa**.

## 6- Proves de funcionament

a) Configura **ngrok** per poder exportar a l'exterior 2 ports. El port **8080** per **Jenkins** i el port **8000** per l'aplicació **pizzas**. Modifica el fitxer `/home/vagrant/.config/ngrok/ngrok.yml` i fes que tingui aquest contingut:

```
version: "3"
agent:
  authtoken: 2sEHgVBxiSftC7jGMb5krbjNz95_7uS2gGPxnmSxmAjcsEbKv
tunnels:
  first:
    addr: 8000
    proto: http
  second:
    addr: 8080
    proto: http
```

Recorda que dins d'un fitxer `.yml` o `.yaml` els espais han de ser sempre iguals.

b) Posa en marxa **ngrok** novament a la màquina **produccio** fer novament disponible públicament la teva eina **Jenkins** i l'aplicació **pizzas** executant:

```
ngrok start --all
```

c) Modifica **Webhooks** de **Github** per treballar amb la nova URL de **Jenkins**.

d) Dins de la màquina **desenvolupament** obre el fitxer **pizza.php** que es troba a la carpeta **pizzas** i afegeix just abans de `</body>` (que es troba cap el final del fitxer) el següent codi:

```
<a href="pizza.html">Retorna al formulari de selecció d'ingredients</a>
```

Modifica també les línies **13** i **25** de **pizza.php**. En comptes de `exit(0)` hauries de tenir `# exit(0)`

e) Puja la nova versió al dipòsit local i al remot:

```
git add *
git commit -m "Commit 5 de l'aplicatiu pizzas"
git push -u origin main
```

f) Si tota la cadena està configurada correctament i funciona sense problemes, passats uns instants comprova que s'ha actualitzat l'aplicació accedint a l'adreça URL donada per **ngrok** per accedir-hi. Recorda que la pàgina inicial de l'aplicació es diu `pizza.html`, de manera que si per exemple **ngrok** et dona la URL `https://945d-79-151-176-29.ngrok-free.app` llavors hauries d'accedir a l'aplicació utilitzant l'adreça `https://945d-79-151-176-29.ngrok-free.app/pizza.html`. Comprova que l'aplicació **pizzas** funciona escollint diversos ingredients i que hi ha el botó de tornada per tornar al formulari inicial un cop mostra el preu.

g) Dins de la màquina **desenvolupament**:

- Canvia dins de **pizza.php** el nom de l'autor i escriu el nom del teu usuari d Github.
- Canvia dins de **pizza.html** el nom de l'autor i escriu el nom del teu usuari d Github.
- Puja les modificacions al dipòsit local amb el comentari *"Commit 6 de l'aplicatiu pizzas"* i després puja-les a **Github**.

A continuació, comprova que les actualitzacions tenen efecte automàticat dins de la màquina **produccio** i de l'aplicació de selecció de pizzas.

## **Lliurament de l'activitat**

- a) Dins de la màquina **desenvolupament**, modifica **pizza.php** i fes que immediatament després del nom de l'autor surti la data i hora del lliurament afegint la instrucció: `echo date("l jS \of F Y h:i:s A") . "<br>";`
- b) Puja les modificacions al dipòsit local amb el comentari *"Commit 7 de l'aplicatiu pizzas"* i després puja-les a **Github**. Mostra dins de **Github** que la darrera actualització s'ha rebut correctament.
- c) Mostra dins dels **Webhooks** de **Github** que la darrera actualització s'ha enviat a **Jenkins** correctament.
- d) Mostra dins de **Jenkins** que la darrera actualització s'ha rebut correctament.
- e) Comprova que les actualitzacions tenen efecte automàticat dins de la màquina **produccio** i de l'aplicació de selecció de pizzas.
- f) Data límit per obtenir el **100%** de la nota: **dimecres 12-2-24** a les **21.00**.