

Fase 4 - Activitat 9.4: Desplegament automàtic d'aplicacions sobre màquina de producció. Treballant amb Jenkins, zrok, Github, Tokens de zrok, Webhooks de Github i scripts del bash

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 9.4

Grup/Individual: Individual

Membres:

1.- Objectiu de l'activitat

L'objectiu d'aquesta activitat és que cada nova versió de l'aplicació pizzas desenvolupada dins de la màquina **desenvolupament** que es puja a **Github** es desplegui automàticament sobre la màquina **produccio** per fer-la disponible immediatament als usuaris.

En caldrà:

- Instal·lar **Apache** i **PHP** dins de les màquines de **produccio** i **desenvolupament**.
- Utilitzar **ngrok**, **Jenkins** i **Webhooks de Github** per poder sincronitzar els 2 equips.
- Aprofitar que **Jenkins** permet també executar **scripts del bash** cada cop que es fa una descarrega automàtica per posar en marxa un **script** que desplegarà les noves versions de l'aplicació sobre la màquina **produccio** per fer-les immediatament disponibles als usuaris.

2- Modificació de Vagrantfile per afegir Apache2 i PHP a les màquines desenvolupament i produccio

a) Modifica la secció **provision** del fitxer **Vagrantfile** per afegir el programari Apache2 i PHP dins de les màquines **desenvolupament** i **produccio**. Fes que el contingut sigui el següent:

```
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 apache2 apache2-doc php8.2 libapache2-mod-php
  sudo gpasswd -a vagrant www-data
  sudo chown -R www-data:www-data
  sudo chmod 770 /var/www/html
SHELL
```

b) Executa:

- **vagrant provision** si ja ten en marxa les màquines virtuals
- **vagrant up --provision** si les màquines virtuals estan aturades i has d'iniciar-les.

c) Inicia i accedeix a la màquina **produccio** i **desenvolupament**, i comprova que totes dues tenen **apache2** i **php** instal·lats, que l'usuari **vagrant** és membre del grup **www-data**, que els permisos de **/var/www/html** són **rw-rw-r--**, que el propietari de **/var/www/html** és **www-data** i que el grup de **/var/www/html** és **www-data**.

3- Configuració i preparació de la màquina produccio

a) Primer has de configurar la màquina **produccio** amb els permisos adequats perquè **Jenkins** pugui crear, modificar, esborrar i llegir dins de la carpeta **/var/www/html** i els seus fitxers. Per això, només cal afegir l'usari **jenkins** al grup **www-data** i **sudo** executant:

```
sudo gpasswd -a jenkins www-data  
sudo gpasswd -a jenkins sudo
```

i a continuació reinicia **Jenkins**:

```
sudo systemctl restart jenkins
```

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. Per fer això:

- A la màquina **produccio**, crea una carpeta de nom **scripts**.
- Dins de **scripts** crea un fitxer de nom **pizzas.sh** amb aquest contingut:

```
#!/bin/bash  
cp /var/lib/jenkins/workspace/pizzas/pizza.html /var/www/html  
cp /var/lib/jenkins/workspace/pizzas/pizza.php /var/www/html  
exit 0
```

- Fes que **pizzas.sh** sigui executable amb l'ordre: **chmod 770 /home/vagrant/scripts/pizzas.sh**

c) Finalment, accedeix a **Jenkins** i configura el projecte **pizzas** de manera que executi el script **pizzas.sh** cada cop que descarregui una nova actualització des de Github:

- Accedeix al projecte **pizzas** de l'eina **Jenkins** de la màquina **produccio**, i després l'eina de configuració del projecte que està al menú de l'esquerra. A continuació, accedeix a **Build Steps**, i selecciona **Add build step** → **Execute shell**.
- A continuació, escriu el nom i directori complet del script que has creat a l'apartat anterior: **/home/vagrant/scripts/pizzas.sh**.
- Fes clic a **Desa (Save)**.

d) A continuació, exposa públicament l'aplicació **Jenkins** de la màquina **produccio**. Executa:

```
zrok enable token_zrok (a on has de canviar token_zrok pel teu token de zrok)  
zrok share public 8080
```

e) Actualitza (si et cal) l'adreça URL dins de **Webhooks** de **Github**. Només has de canviar el valor **Payload URL** al principi i fer clic al botó **Update webhook** al final.

e) Des d'un altre terminal, torna a entrar a la màquina **produccio** i exposa públicament **Apache2** per poder més veure l'aplicació més endavant. En el segon terminal, només cal executar:

```
zrok share public 80
```

Apunta la **URL pública** donada per **zrok** poder utilitzar-la a l'apartat **5** de l'activitat.

f) Ja tenim preparada la màquina produccio. Ara per ara, l'aplicació encara no està penjada i no està disponible. Estarà disponible després del següent apartat.

4.- Actualització de l'aplicació en la màquina desenvolupament. Pujada a Github.

a) Accedeix a la màquina **desenvolupament** i modifica el contingut de **pizza.php**. Esborra el contingut anterior i 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>";
      }
      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>";
      }
      echo "<br>";
      echo "Autor: xxyyzz"<br>";
    ?>
  </body>
</html>
```

A on **xxyyzz** és el teu nom d'usuari de **Github**.

b) 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 3 de l'aplicatiu pizzas"
git push -u origin main
```

c) Comprova que s'han pujat les noves versions de **pizza.html** i **pizza.php** al teu projecte **pizzas** de **Github**.

5- Comprovació de la descarrega automàtica de l'actualització en la màquina producció des de Github. Comprovació del desplegament automàtic de l'actualització sobre l'Apache de la màquina producció.

a) Fes les següents comprovacions per assegurar-te que la nova versió s'ha descarregat automàticament a la màquina **produccio**:

- Accedeix a a **Settings** → **Webhooks** del projecte **pizzas** de **Github**. A continuació refresca la pàgina i comprova que el **Webhook** que havies creat a l'apartat 3 té un tick verd ✓ i el missatge **Last delivery was successful**.
- Accedeix a la web d'administració de l'eina **Jenkins** de la màquina **produccio** utilitzant l'adreça **URL pública** proporcionada per **zrok**.
- Entra dins del projecte **pizzas** i refresca la pàgina. Comprova que s'ha creat un nou **Build** que abans no existia i que tu **NO** has posat en marxa manualment.
- Entra dins del **Build** que s'ha creat automàticament i accedeix a la secció **Console Output**. Comprova a la informació mostrada per **Jenkins** que a la primera línia, surt un missatge informant que el **Build** va ser iniciat per un **push** del teu usuari de **Github**.
- Torna a pàgina principal del projecte **pizzas** de **Jenkins**, i comprova dins de la secció **Espai de treball** que els fitxers **pizzas.html** i **pizza.php** tenen el mateix codi que els fitxers **pizzas.html** i **pizza.php** dins de la màquina **desenvolupament**.
- Comprova que l'hora donada per **Jenkins** d'actualització dels fitxers dins de la màquina **produccio** coincideix amb l'hora donada per **Github** de pujada del codi.

b) Fes la següent comprovació per assegurar-te que la nova versió s'ha desplegat automàticament a la màquina **produccio**:

- Accedeix a la carpeta **/var/www/html** i comprova que **pizza.html** s'ha copiat correctament i que és la darrera versió.
- Accedeix a la carpeta **/var/www/html** i comprova que **pizza.php** s'ha copiat correctament i que és la darrera versió.

c) Fes la següent comprovació per assegurar-te que la nova versió funciona correctament dins de la màquina **produccio**:

- Accedeix a URL pública de l'aplicació tenint en compte que el fitxer d'inici és **pizza.html**.
- Comprova que funciona i que el valor de la pizza és **12,1 €** si selecciones 2 ingredients, i que a més a més, surt la llista d'ingredients extres.

Lliurament de l'activitat

a) Amb tot el sistema preparat, dins de la màquina desenvolupament, modifica la línia de **pizza.html**:

Nom de l'administrador/desenvolupador: **dacomo**

i escriu ara:

Nom de l'administrador/desenvolupador: **xyyyz del pj9 (DevOps i Cloud Computing)**

a on **xyyyz** és el teu nom d'usuari de **Github**.

b) Afegeix abans de </body> de **pizza.php** la línia: Torna a pizza.html

c) Puja les modificacions al **dipòsit local** amb el comentari *"Commit final de l'aplicatiu pizzas"* i després puja-les a **Github**.

d) Comprova que les actualitzacions tenen efecte automàticat dins de l'aplicació que es troba a la màquina **produccio**.

e) Comprova que l'aplicació funciona.

f) Data límit per obtenir el **100%** de la nota: **dilluns 26-1-26** a les **19.10**.