

Fase 4 - Activitat 11.7: Desplegant una nova versió de l'aplicació en el servidor en producció cada vegada que es canvia la versió a Github des de l'ordinador de desenvolupament

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: GP2 DevOps i Cloud Computing

Fase: 4

Activitat: 11.7

Grup:

Membres:

1.- Objectiu de l'activitat

- Modificació de la configuració del servidor **produccio** per acceptar modificacions des de l'eina **Jenkins** del servidor **pipelinecid**.
- Modificació del servidor **pipelinecid**:
 - L'usuari **jenkins** de **pipelinecid** ha de tenir permís per executar ordres sobre el servidor **produccio**.
 - L'eina **Jenkins** ha d'executar un script que actui sobre el servidor **produccio** i actualitzi la versió de l'aplicació cada cop que es puji una nova versió a **Github**.
- Proves de modificació de l'aplicació i comprovació que automàticament tindrem disponible la nova versió sobre el servidor.

2.- Modificacions previes dins de l'ordinador de desenvolupament

Afegeix al fitxer README de l'aplicació una nova línia de manera que el contingut de l'arxiu sigui ara aquest:

Activitat: gp1f4act11

Aplicació: ipv4

Versió: 1.2

Versió anterior: 1.1

Recorda que si has lliurat l'activitat gp1f4a11.5, en principi, la darrera versió és la 1.2 però si tens més versions, simplement indica la el número de la versió actual a **Versió** i el número de la anterior versió a **Versió anterior**.

3.- Modificacions dins de l'ordinador produccio

a) Assegura't que l'usuari vagrant de produccio té una contrasenya:

- Accedeix com a **root** → **sudo su -**
- Executa: **passwd vagrant**
- Fes que l'usuari **vagrant** de **produccio** tingui la contrasenya **clotfje**

b) Assegura't que el servidor SSH accepta la connexió remota:

- Edita **/etc/ssh/sshd_config**
- Modifica el paràmetre **PubkeyAuthentication** de manera que tingui el valor **yes** i esborra el caràcter **#** al principi per evitar que la línia a on es troba el paràmetre es consideri un comentari
- Reinicia el servidor: **sudo systemctl restart ssh**

4.- Configuració del servidor pipelinecid

- a) Fes que des del servidor **pipelinecid** es pugui fer un **ping** al servidor **produccio** amb el seu nom:
- Troba l'adreça IP del servidor **produccio**
 - Afegeix al fitxer **/etc/hosts** del servidor **pipelinecid** el **nom** i **adreça IP** del servidor **produccio**.
 - Comprova que des del servidor **pipelinecid** pots fer **ping produccio** i obtens una resposta
- b) Configura l'usuari **jenkins** de **pipelinecid** per poder accedir directament via SSH al servidor **produccio** amb clau pública **RSA**:
- Canvia a usuari **jenkis**. Executa: **sudo su - jenkins**
 - Com usuari **jenkins** crea una clau RSA. Executa: **ssh-keygen -t rsa** (recorda de premer Enter a totes les preguntes per acceptar les opcions per defecte).
 - Com usuari **jenkins** executa: **cd .ssh** per accedir a la carpeta de les seves claus.
 - Copia la clau **id_rsa.pub** de l'usuari **jenkins** de **pipelinecid** dins del directori **.ssh** de l'usuari **vagrant** de producció. Executa:
scp id_rsa.pub vagrant@produccio:~/.ssh/authorized_keys2
- c) Accedeix a l'eina **Jenkins** de **pipelinecid** i a continuació instal·la el plugin **Publish Over SSH**:
- Accedeix a Dashboard (Panel de control) → Configuració de Jenkins → Plugins
 - Selecciona Available plugins → busca el plugin **Publish Over SSH**
 - Selecciona el plugin **Publish Over SSH** i a continuació prem el botó **Install** per fer la instal·lació
- d) Configura el Plugin **Publish Over SSH**:
- Comprova que al directori **/var/lib/jenkins/.ssh** de **pipelinecid** pots troba el fitxer **id_rsa** a on es troba la parella de clau privada i pública de l'usuari **jenkins**.
 - Accedeix a Dashboard (Panel de control) → Configuració de Jenkins → System
 - Accedeix a Publish Over SSH → Path to key i escriu la localització del fitxer **id_rsa**, es a dir: **/var/lib/jenkins/.ssh/id_rsa**
 - Prem el botó Afegir a la secció SSH servers i escriu les dades del servidor **produccio**:
 - Name: **Produccio**
 - Hostname: **produccio**
 - Username: **vagrant**
 - Remote Directory: **/**
 - Finalment prem el botó Save i ara l'eina **Jenkins** de **pipelinecid** ja podrà executar ordres sobre el servidor **produccio** via **SSH**.
- e) Dins del servidor **pipelinecid** crea un directori de nom **scripts**. Dins del directori crea un fitxer de nom **gp1f4a11.7.sh** amb el següent contingut:

```
#!/bin/bash
ver=$(cat /var/lib/jenkins/workspace/ipcalc/README | grep "Versió:" | cut -d " " -f 2)
echo $ver
ver_ant=$(cat /var/lib/jenkins/workspace/ipcalc/README | grep "Versió anterior:" | cut -d " " -f 3)
echo $ver_ant
ssh vagrant@produccio mkdir -p projectes/ipcalc/$ver
scp -r /var/lib/jenkins/workspace/ipcalc vagrant@produccio:~/projectes/ipcalc/$ver
if [[ ! -z $ver_ant ]]
then
  comprova=$(ssh vagrant@produccio ls /home/vagrant/projectes/ipcalc | grep $ver_ant)
  if [[ $comprova != "" ]]
  then
    ssh vagrant@produccio docker-compose -f /home/vagrant/projectes/ipcalc/$ver_ant/ipcalc/docker-compose.yml down
  fi
fi
ssh vagrant@produccio docker-compose -f /home/vagrant/projectes/ipcalc/$ver/ipcalc/docker-compose.yml up -d
exit 0
```

i afegeix el permís d'execució: **chmod +x gp1f4a11.7.sh**



e) Accedeix al projecte `ipcalc` de l'eina **Jenkins** del servidor `pipelinecicd`, i modifica Build Steps → Executa shell. Esborra les ordres que vas escriure a l'activitat `gp1f4a11.5` i escriu ara el nom i directori complet del script que has creat a l'apartat anterior: `/home/vagrant/scripts/gp1f4a11.7.sh` i a continuació fes clic a **Desa**.

5.- Proves de funcionament

a) Posa en marxa **ngrok** a la màquina `pipelinecicd` fer disponible públicament la teva eina **Jenkins**. Comprova la nova URL assignada.

b) Posa en marxa **ngrok** a la màquina `producció` fer disponible públicament l'aplicació **ipcalc**. Comprova la nova URL assignada.

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

d) Dins de l'ordinador de **desenvolupament**:

- Entra a **desenvolupaments** → **ipcalc**. Modifica el fitxer **README** de manera que **Versió** valgui **1.3** i **Versió anterior** valgui **1.2**.
- Entra a **desenvolupaments** → **ipcalc** → **app**. Modifica el fitxer **index.php** i afegeix darrera de la línia 12 les següents 3 línies:

```
<?php
    echo date("l jS \of F Y h:i:s A") . "<br>";
?>
```
- Torna a **desenvolupaments** → **ipcalc** i executa:

```
git add *
git commit -m "Commit 1.3 de l'aplicació ipcalc"
git push -u origin main
```

e) Si tot està correctament configurat i ha funcionant sense problemes, passats uns segon comprova:

- Que dins de l'ordinador de **producció** tens un nou directori de nom `projectes->ipcalc->1.3`.
- Si executes `docker ps -a` en el servidor `producció` tens els dockers de l'aplicació.
- Si accedeixes a l'aplicació per mitjà de la seva URL de `ngrok` pots veure a la pàgina principal l'hora i data.

Lliurament de l'activitat

Entra a la carpeta `ipcalc` → `app` dins de la màquina `desenvolupador`. Modifica `index.php`. Fes que al paràgraf **Autor** el valor sigui: **asix2 - gp1f4a11 - ver 1.4**

b) Modifica **README**. Fes que:

- **Versió** sigui igual a **1.4**
- **Versió anterior** sigui igual a **1.3**

c) Actualitza el dipòsit local i sincronitza-ho amb el remot. Executa:

```
git add *
git commit -m "Versió 1.4 del projecte gp1f4act11"
git push -u origin main
```

d) Comprova que:

- Dins de l'ordinador de **producció** tens un nou directori de nom `projectes->ipcalc->1.4`.
- Si accedeixes a l'aplicació per mitjà de la seva URL de `ngrok` pots veure les modificacions.

e) Data límit per obtenir el **100%** de la nota: **dimecres 17-1-24** a les **21.00**.