

## **Fase 4 - Activitat 11.5: Connexió de Github amb Jenkins quan es fa un push del projecte ipcalc. Utilització de webhooks de Github.**

### **0- Identificació del grup i activitat:**

**Curs:** ASIX2

**Projecte:** GP2 DevOps i Cloud Computing

**Fase:** 4

**Activitat:** 11.4

**Grup:**

**Membres:**

### **1.- Objectiu de l'activitat**

L'objectiu d'aquesta activitat és la creació d'un projecte CI/CD de **Jenkins** per descarregar automàticament cada nova versió del codi de projecte **ipcalc** que es puja al seu dipòsit de **Github**. Per aconseguir aquest objectiu hem de:

- Configurar el projecte **ipcalc** de **Github** per treballar amb **Webhooks**.
- Crear i configurar un projecte de **Jenkins** que treballi amb el dipòsit de **Github** i amb **Webhooks**.
- Caldrà **ngrok** perquè **Github** i **Jenkins** es puguin comunicar.

### **2.- Abans de començar**

a) Posa en marxa les màquines virtuals **desenvolupador** i **pipelinecd**.

b) Comprova dins de la màquina **desenvolupador** que el dipòsit local del projecte **ipcalc** està sincronitzat amb dipòsit remot del projecte **ipcalc** dins de Github. Dins de màquina **desenvolupador** accedeix a la carpeta desenvolupaments/ipcalc i executa:

```
git diff origin/main -- [local-path]
```

i si el terminal NO mostra cap resultat, llavors és que estan sincronitzats.

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

```
ngrok http 8080
```

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

d) Torna a executar **vagrant ssh** des de la màquina **host** per tenir un segon terminal d'accés a la màquina virtual **pipelinecd**.

### **3.- Creació i configuració d'un nou projecte de Jenkins**

#### **3.1- Creació d'un projecte CI/CD de Jenkins**

a) Accedeix a la pàgina d'administració de **Jenkins**.

b) Selecciona **+ Item Nova** per crear un nou projecte CI/CD de **Jenkins**.

c) Escribeu a **Enter an item name** el nom del projecte: **ipcalc**.



d) Selecciona la opció **Freestyle project**

e) Fes clic a **OK** (a la part inferior de la pàgina) i es crearà el projecte.

### 3.2- Configuració inicial del projecte per establir connexió amb el dipòsit de Github

a) A la secció **General** selecciona:

- **Descripció** → CI/CD ipcalc
- **Github project** → **Project url** → Escriu la URL del teu dipòsit de Github amb el projecte **ipcalc**.

b) A la secció **Gestor del Codi Font** selecciona:



- **Git**
- **Repository URL** → Escriu la URL del teu dipòsit de Github.
- **Branches to build** → **Branch Specifier (blank for 'any')** → En blanc per treballar amb qualsevol branca

c) Al final de la pàgina de configuració prem el botó **Desa** per desar la configuració.

### 3.3- Comprovació de la sincronització manualment

a) Accedeix a **Dashboard** → **ipcacl**.

b) Selecciona l'opció **Construir ara**.

c) Comprova si l'operació ha estat un èxit. Dins de la secció **Build History** comprova que el primer **Build** identificat amb **#1** té la icona  i en cas contrari la icona .

d) Si tot ha anat bé, dins del Build identificat amb **#1** selecciona **Console Output** i comprova que efectivament s'ha sincronitzat amb el teu dipòsit de **Github** a la secció Cloning repository i que el procés a finalitzat **Finished: SUCCESS**. A on ha descarregat Jenkins els fitxers del projecte.

e) Accedeix a **Dashboard** → **ipcacl** → Espai de treball i comprova que s'han descarregat tots els fitxers de l'aplicació.

### 3.4- Configuració inicial del projecte ipcalc de Jenkins per sincronitzar-se automàticament amb Github

a) Accedeix al projecte CI/CD **ipcalc**. Selecciona l'opció **Configura**.

b) A la secció **Build Triggers** selecciona: **GitHub hook trigger for GITScm polling**

c) A la secció **Build Steps** selecciona: **Add build step** → **Execute shell**

d) A la secció **command** escriu el següent script:

```
ver=$(cat /var/lib/jenkins/workspace/ipcalc/README | grep Versió | cut -d " " -f 2)
echo "Versió $ver de l'aplicació ipcalc"
```

e) Selecciona **Desa** per desar la nova configuració.



f) Crea un **token API Secret** pel teu usuari de **Jenkins**:

- Selecciona **Dashboard** → **Gent** i fer clic al teu usuari de Jenkins.
- Selecciona **Configura**
- Selecciona **API Token** → **Add new token** → **Generate** i generar un **Token**
- Copia el valor generat per poder utilitzar-lo amb Github.

#### **4.- Configuració de Webhooks de Github**

a) Accedeix al teu dipòsit **ipcalc** de **Github**. A continuació accedeix a **Settings** → **Webhooks** i fes clic a **Add Webhook**.

b) A la secció **Payload** afegeix l'adreça URL pública que has obtingut amb **ngrok** i afegeix **/github-webhook/** a la URL. Per exemple si **ngrok** et dona **https://47ff-79-153-220-213.ngrok-free.app**, s'haurà d'escriure **https://47ff-79-153-220-213.ngrok-free.app/github-webhook/**

c) A la secció **Content type** selecciona: **applications/json**.

d) Afegeix el **Token** creat a l'apartat 3.4 dins de la secció **Secret**.

e) A la secció **SSL Verification** selecciona: **Enable SSL verification**.

f) A la secció **Which events would you like to trigger this webhook?** selecciona: **Just de push event**.

g) Habilita l'opció **Active**.

h) Fes clic a **Add Webhook**.

#### **5.- Comprovació de funcionament**

a) Entra a la carpeta **ipcalc** → **app** dins de la màquina **desenvolupador**. Modifica **index.php**. Afegeix a sota de la línia 11 una nova línia amb el següent contingut:

```
<p>Autor: asix2</p>
```

b) Modifica **README**. Fes que la darrera línia sigui: **Versió: 1.1**

c) Actualitza el dipòsit local i sincronitza-ho amb el remot. Torna a la carpeta **ipcalc** (que és a on tens el dipòsit Git local) i cxeuta:

```
git add *  
git commit -m "Versió 1.1 del projecte ipcalc"  
git push -u origin main
```

i comprova que s'ha pujat la nova versió del projecte a **ipcalc** de **Github**.

d) Accedeix a la màquina **pipelinecid** i comprova que s'ha descarregat la darrera versió de l'aplicació automàticament.

Si el resultat és l'esperat, llavors el projecte **ipcalc** de **Jenkins** dins de la màquina **pipelinecid** pot descarregar i treballar amb el codi de cada nova versió del projecte **ipcalc** desenvolupat des de la màquina **desenvolupador** que s'hagi pujat al dipòsit **ipcalc** de **Github**.



## **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 el teu nom complet.

**b)** Modifica **README**. Fes que la darrera línia sigui: **Versió: 1.2**

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

```
git add *  
git commit -m "Versió 1.2 del projecte gp2f4act12"  
git push -u origin main
```

**d)** Comprova que s'ha descarregat la nova versió del projecte dins del directori **/home/vagrant/projectes**.

**e)** Data límit per obtenir el **100%** de la nota: **dimarts 9-1-24** a les **17.45**.