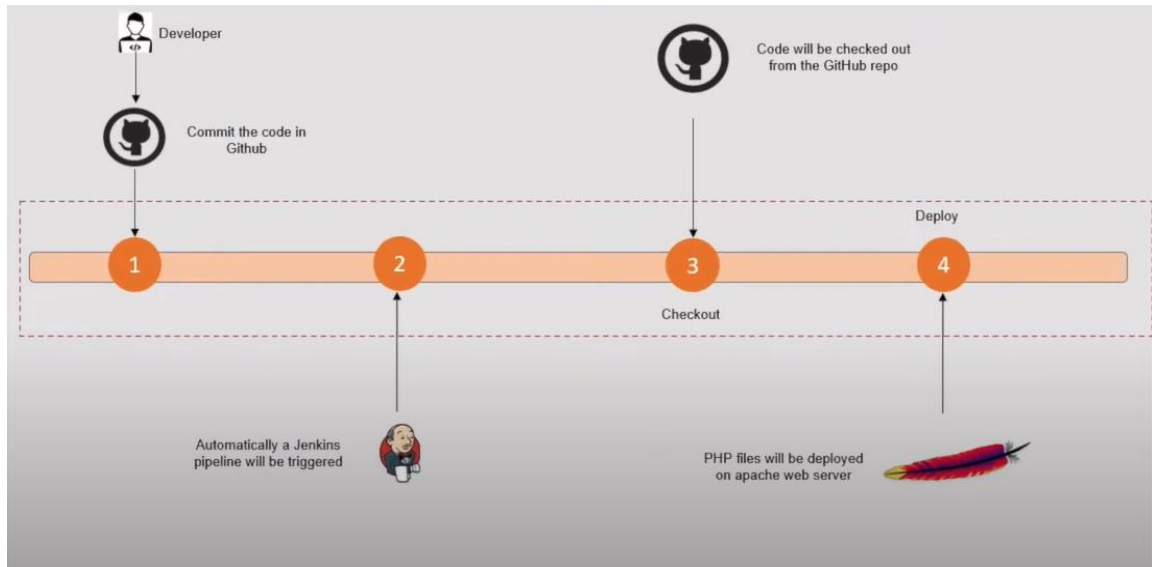


DESPLIEGUE CONTINUO CON JENKINS

Autor: FREDDY DANIEL ALCARAZO IBAÑEZ

Fecha: 12 de Abril del 2023.



Problemática a Resolver.

El despliegue manual de proyectos/aplicaciones web a producción requiere de intervención humana para realizar dicho proceso, este mismo consume tiempo y esta sujeto al error humano, lo desarrolladores desean que lo más pronto posible sus builds sean puestos en producción, sin embargo estos deben de pasar por revisiones antes de ponerlos en producción y en dicho proceso hay que completar varias etapas como ejemplo hacerle pruebas al software que ellos han creado.

Beneficios de Jenkins ante el problema

1. El beneficio principal del software Jenkins a esta problemática es que permite hacer deploy de manera automática a un proyecto web sin mucha intervención humana.
2. Reducción de errores
3. Acelerar el proceso de entregables
4. Reducir costos.

Despliegue del Pipeline en Jenkins y Códigos:

Para llevar a cabo este trabajo se instalo una máquina virtual Ubuntu con un servidor apache instalado y el servidor de Jenkins esta instalado en una maquina con el sistema operativo

Windows 10. El objetivo de este trabajo es hacer deploy de un proyecto web alojado en un repositorio github a un servidor de aplicaciones web como es el apache.

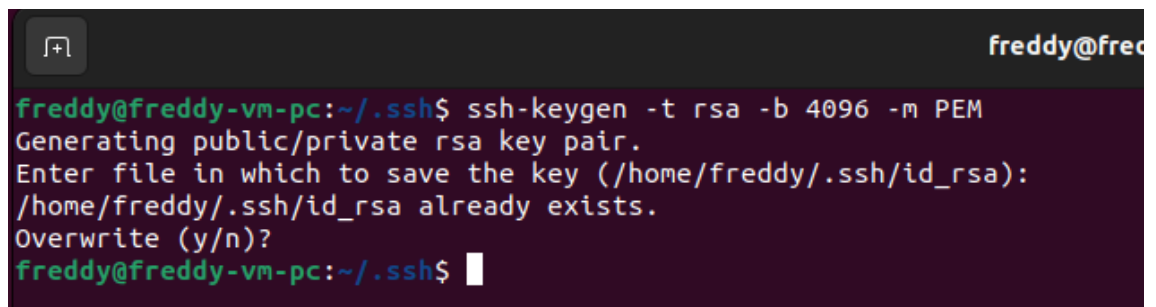
Pasos.

Configurar el servidor SSH de Ubuntu

Para poder realizar los pasos posteriores de este trabajo primero debemos de asegurarnos de tener instalado un servidor SSH en una maquina Ubuntu, luego debemos de generar las llaves para poder conectarse a dicho servidor y poder usarlo con Jenkins y hacer el deploy-

1. Crear llave ssh

```
$ ssh-keygen -t rsa -b 4096 -m PEM
```

A terminal window with a dark background. The prompt is 'freddy@freddy-vm-pc:~/.ssh\$'. The command 'ssh-keygen -t rsa -b 4096 -m PEM' is entered. The output shows the generation of a public/private RSA key pair, the file path '/home/freddy/.ssh/id_rsa', and a confirmation to overwrite the existing file. The prompt returns to 'freddy@freddy-vm-pc:~/.ssh\$' with a cursor.

```
freddy@freddy-vm-pc:~/.ssh$ ssh-keygen -t rsa -b 4096 -m PEM
Generating public/private rsa key pair.
Enter file in which to save the key (/home/freddy/.ssh/id_rsa):
/home/freddy/.ssh/id_rsa already exists.
Overwrite (y/n)?
freddy@freddy-vm-pc:~/.ssh$
```

Entrar al directorio de ssh:

```
$ cd ~/.ssh/
```

Mostrar clave privada creada previamente:

```
$ cat id_rsa
```

A terminal window showing the output of the 'cat id_rsa' command. It displays the RSA private key in PEM format, starting with '-----BEGIN RSA PRIVATE KEY-----' and ending with '-----END RSA PRIVATE KEY-----'. The key is a long string of alphanumeric characters.

```
freddy@freddy-vm-pc:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEKQIBAAKCAgEAXIwHQDTDX33Y4zSMM9m9D0oIvpgFt71xuRqejH3zP4cei9v
zqFdjs1koSe72uIxutG5ENVHGPYfHu63WJbvovTf46BR2U0RD+xOLmJ1TyB6c8J
waBbkaj79EdKPTVE34NhIUmMh3ZNoRnP4GHCNkIP6W0wzEVICKpqpTxAoE+u9nSC
esZILZrIIIt+/qz5YbbZm9bvU4HW0zDEOW5mVnAF59vQ3XDgKa1I8DXcZdNV02LnE
/Bs8U/ghq3aCFDs9uYLJ/TX3qm4+DwnHUXRruwugPpPaVvbGgC058I4YMYJikW6U
S3By00FEHtDVN++C7YcFjStfV5VDYtR0MFhiq6G+dEe3fr7Cbm8gfXmXV3E/AyWA
jwg9sCwfXUV8jhKwIRFVjagLQiUg6mIj4GZMBqvamKN2nJ12qYCYhDnaWyfBQcv
dossbhnTERPb8kwcDE0vDmtus1bi0/vVbPnNparUilvJCOT9B5nptsovVITV+arR
.....
MsDgk7hIaisAzWUlgJ0Hwo2QQJcx837grjiwNYUCftqUApFQZPL4IKq9W0DU6Vtd
R+TKl0ZL2H6U1KO/nc02aRGKF04nEVsITJl0NxIcCDEB0J11VZaNPakg4HCdUGHX
cBCxHnKM9b14cGzIADw5rrvlg/AWliXDPy46j/C7JG5q2Qd/WnueUQycRyIeRo0z
W9pEwkFb0tpsrotMfvjLGz1KkwJ1Y0SZi7NNvxHsHIOX6ANQSZkVQwDxm9qG
-----END RSA PRIVATE KEY-----
freddy@freddy-vm-pc:~/.ssh$
```

Agregar llave publica a las llaves permitidas:

```
$ cat id_rsa.pub >> authorized_keys
```

```
freddy@freddy-vm-pc:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
freddy@freddy-vm-pc:~/.ssh$ cat id_rsa.pub >> authorized_keys
freddy@freddy-vm-pc:~/.ssh$
```

Agregar algoritmo de conexión para que Jenkins a través del plugin “Publish over SSH” pueda conectarse:

PubkeyAcceptedAlgorithms +ssh-rsa

Para eso abrimos el archivo de configuración del servidor SSH:

\$ sudo nano /etc/ssh/sshd_config

Y agregamos:

```
# The strategy used for options in the default sshd_config file
# OpenSSH is to specify options with their default values in
# comments, but leave them commented. Uncommented options
# default value.

Include /etc/ssh/sshd_config.d/*.conf
PubkeyAcceptedAlgorithms +ssh-rsa

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Presionamos Ctrl+O para guardar y Ctrl+X para cerrar. Finalmente reiniciamos el servidor ssh:

\$ sudo systemctl restart sshd

2. Configurar Jenkins e Instalar plugin

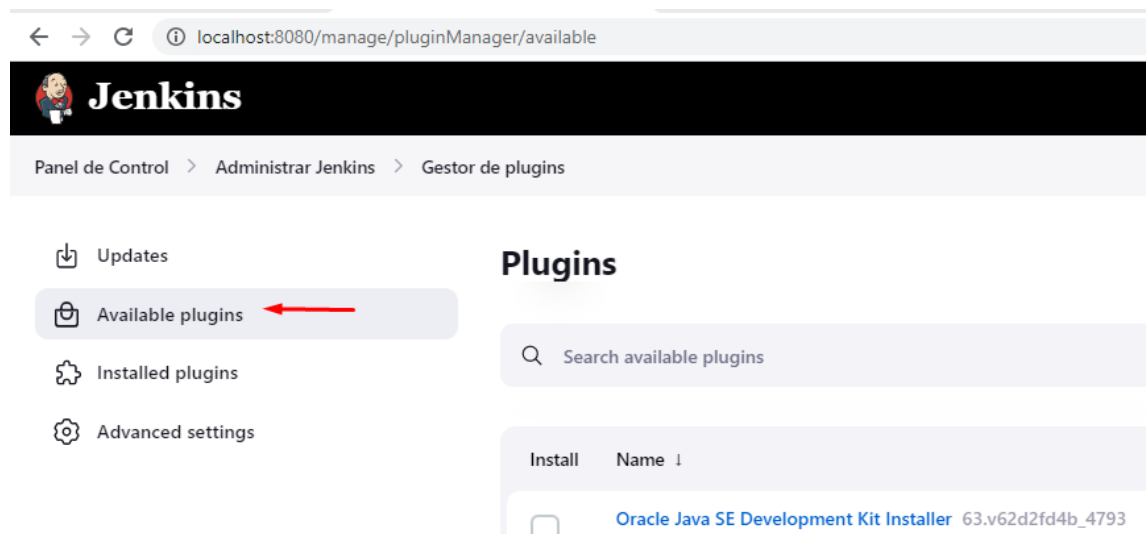
Ir a “Administrar Jenkins”:

The screenshot shows the Jenkins dashboard at localhost:8080. The left sidebar contains a menu with 'Administrar Jenkins' highlighted by a red arrow. The main area displays a table of jobs, with 'TareaCurso' listed under the 'Último Éxito' column. Below the table, there are sections for 'Trabajos en la cola' (No hay trabajos en la cola) and 'Estado del ejecutor de construcciones' (1 Inactivo, 2 Inactivo).

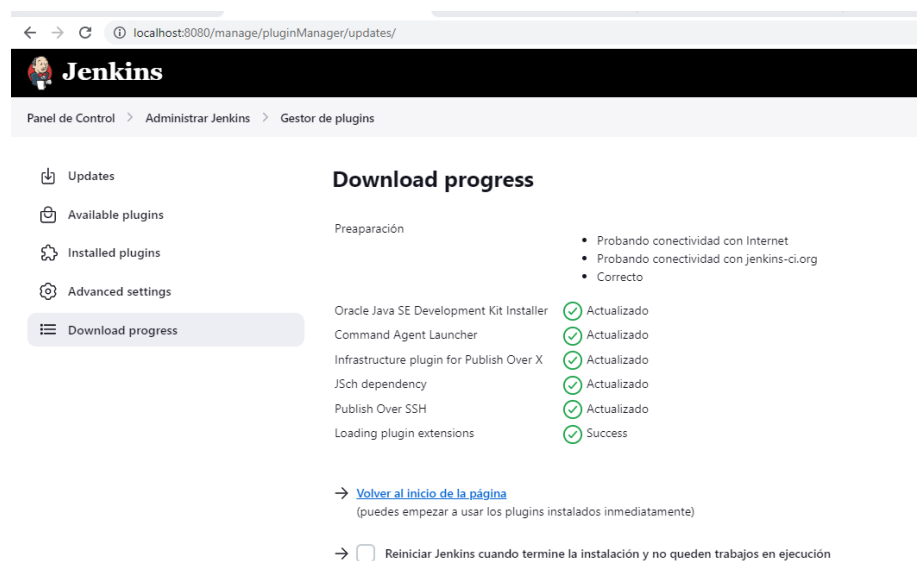
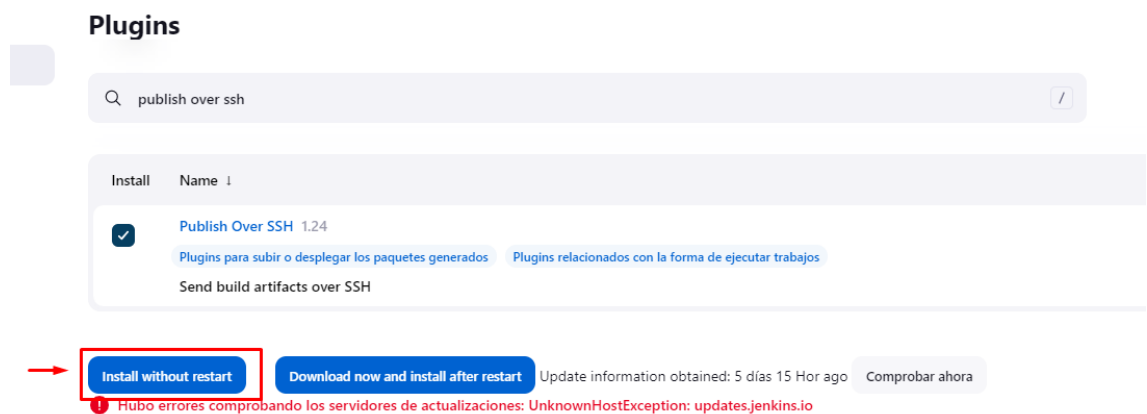
Luego click en administrar Plugins.

The screenshot shows the 'Administrar Jenkins' page at localhost:8080/manage/. The page has a yellow warning banner at the top. Below it, there are sections for 'System Configuration', 'Security', and 'Status Information'. In the 'System Configuration' section, the 'Administrar Plugins' option is highlighted with a red box and a red arrow. The description for 'Administrar Plugins' is: 'Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.'

Luego click en “Available Plugins”:



Luego buscar el plugin: **publish over ssh** e instalar sin reiniciar:



Luego de haber instalado el plugin volvemos a la página principal de Jenkins:

Luego click en “Administrar Jenkins”:

The screenshot shows the Jenkins main dashboard. The left sidebar contains links: Nueva Tarea, Personas, Historial de trabajos, **Administrar Jenkins** (highlighted with a red arrow), and Mis vistas. The main area displays a table of jobs with columns S, W, Nombre, and Último Éxito. Below the table, there are sections for 'Trabajos en la cola' (No hay trabajos en la cola) and 'Estado del ejecutor de construcciones' (1 Inactivo, 2 Inactivo).

Luego click en “Configurar el Sistema”:

The screenshot shows the 'Administrar Jenkins' page. At the top, there is a warning: 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation'. Below this, the 'System Configuration' section is highlighted with a red box and a red arrow. It contains two links: 'Configurar el Sistema' (Configurar variables globales y rutas.) and 'Global Tool Configuration' (Configure tools, their locations and automatic installers.).

Luego ubicamos la sección “Publish over SSH”:

Publish over SSH

Jenkins SSH Key ?

Passphrase ?

Path to key ?

Key ?

☐ Disable exec ?

SSH Servers

Añadir

Avanzado ▾

Guardar Apply

Ahora agregamos la llave ssh privada en el textarea “Key”:

```
freddy@freddy-vm-pc: ~/ssh
freddy@freddy-vm-pc: ~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
freddy@freddy-vm-pc: ~/ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEJKIBAAKCAgEAXIwLQhDHTDx33Y4z5NM9m9D0oIvpgFt71xuRqeJH3zP4ceL9v
zqFqJ51ko5e72uIxutGSENVHgPYFHu63WJbvovxvTf46BR2U0RD+XOLnJ1TyB6c8J
wa8BkaJ79EdKPTVE34NhIUmh32NoRnP4GHCnkIP6W0wzEVIcKpqtXAOE+u9n5C
eS2ILZrII+qz5YbbZn9bvu4Hw8ZDE0W5mVnAF59q3XDgKa1180XCZdNVoZLnE
/BS8U/ghe3JaCFDS9uYLJ/TX3qm+DmHuxRruuqppPaVbGgC05814WY31k6WU
538y00FEHtDVN++C7YcF35tFVSVDYtR0MfHlq6G+dEe3fr7Cbm8gFXmXV3E/AyWA
jwg9ScwFXUV9jhKwLRFVjagLQU6mIj4GZMBqvanKN2nJ12qYCYZVhDnakyFBQcv
dossbhnTERPb8kwcDEOyOntus1bJ0/vVbPnNparUllv3COT9B5ptpsovVITV+arR
kJM3E6IYVBvtzIEq3NayB66spLF/ZhZkISO+rP+J5QN3zVIXFZD18JGcP0wsJZQ
nUwms6GpWC1K7KZJbNKX90DUIb3dXrJJpX14GQaTL4xSUXfc3YuwfgULPYDk0h88
aZUzJHtHl9+JSKHuLLADJHrK0xdKkEPLzHPoTGL71d5SEsFDN9Inudu550CAwEA
AQKCAgAww20NexJN3UPALUULLC71LNm8sD99q4wKtqxCPcxtHPvGn050VpK4
OrLo1LM3/cFCgMtbjnfYCu7ux7eUdUA+H03G8xQUgInPFMcN5/IzC1GF10wpPw1E
4UG67M1Gcn7JNPacBRBBc4IvD0GHEBYkh2qRrRUjLfLIdLGsebdLKK+NA96N
IzG1hgoeQLbDVCf/AHR3V+fKkM1cTZ21AuyLD/JgvShT06Pb6MBJL51DzpuLx7
mVSARj/RfQ3LDuoJWUrwQw7YbZk0p6CGA4TG8PvyQl/Zn6cv+1QK/wALVOZBLde
JYHkcbenf0JgJlr6Wvh4sHAcUSUyonJ6N1Lsr7dJ8xIz+MELYZLxnFcmONVLxAc7
8chAd/efth3jEGiuJzhNLLyJj5odWCXRBtI9ttJ08/nTSRAxqvrZvt+8CKHERZQ
k3qsn4VzC/8uPcTx0CZlw1EajymxQdLjmiFhxJ54bsxz/jnJ4WshB4/ZgsKWRN
UdywKobZ28ZL8f0eqf1mAlLkqLWCKMYqarUNVR7rr73gEgFRFPLyIMG8f9n1
+epfZsUnGt9VjdSxjrVwau34hk15IeRikF7pb3htJdxM8chntxsuZJtReIX+nY63
KsmthK4z0Pi/1DvNvJ336GzGIQkDrZptfH2MB30fhXov3Yz4QKCAEA9IK7dMLK
gLYQfurbucTP89fIs1peC1M3sa5ZRTlafUBrWfJQgowe9gowd263DseDjpnDOPt
Bv+D2X2e3Hz5CsgLPyePcUYMDVGjjeYvAhKpZ1cAE57V456WprWHV8rJ2d4wUI
kGLRLSZVzKHDX3U6hDLmJ9UhuCnMaalUW6a63ehF8z+Rw/WwVqQE+aPyAoQ6pJ9G
o3561vzbVCCDap2K1XoYGLpLlALUgZpwjgGLTGGZNYFLxLtmLg4K88a133U7/l
PDrnQymtpTw5gobdgFmFZLCKSLAJj1RfCbL85e0Kc708zt25f0usCqpcp8
NQP8XYSYnU001wKCAQEAzbuwV4R7Ih4Pwu97/otVM14q49gP1cX8m0b397yJEYnt
QvL1CxyP8twqNnZWS1t0tvcjbrm83vqxTTUKRof0702P54uXg61x1QV17r1BtIU
lwk2as08tzy9ANoyoTWQlQY0aTns0b9nJ+zdhYgXc8y4c9g9b2ntppqk+qp58oRW
02j/jw/KaaJLMI0t/qn4EHx1RdL2ov5qbb4KNp0V9NBZAL0v3JgpX7rLL3+/Fje
VJDLfv4rD0u0gu0EXrw8bo4xX0r9FXjyq8Asa9xk1mNLGAvtkJjUqpW5S00ekr
PRRLbnrJxkuvPvc/+d0JbTPHg50wBt3gABlDCC7dWkCAQEAtQPVCFLfIUV8cAI
Ne06acAFHeXw4c9DPfScyBPJgC4sVpVw085VpX81QCV+1GeLKLIE3pCEgF3U
ACp9x7GwaLSfIU55qHAAV0xMCEGCQ1v438/oJfko+7VW6Sxv0LpHIAJkXdlv
0J+6b16aF3YHKt2rDmbsGLBcervDvk+hb70vT/nb/OG/+5SRwa13uz721vToah
```

-----BEGIN RSA PRIVATE KEY-----

Path to key ?

Key ?

```

QEIs1QyMIWZ+IYw8m2dwkndhri5aEJ0Pg/oHStt+n17X6y0Lf+3BqyZAd8j8nb0G
abu1iWkCAQBK7he+vijOf35kfA8H9T0xuUFLvHw+Y7qN3zoe238UyGgDMt58zIIM
/YzOSyl4kaYPoVEgwf8x9IQ8llsuMWmqSR6H/VpQJFc4a4W5TuqoUbkIJXWKrYH
0rJPiii8eCLNri8Hgfy7o8cUEXvZZuuvEsNvhsZ/4+HHRYsCXziq/pngx1YwHqL
IJ9UHRF45/S2JG4d44cG9DxU/dsAaqcCdimhwDgGmCTFACWsGFWxmLhZken7tZx+
VySNm975dldj+NwGaYFIUo0FWx8el/R4obpRLTM5yfiBbbKTOAtt4uNieLTM1ED
/rJS3P0rBzTInqrae1dgRGcwmPMc1AkfAolBAQDbimg3MQJWbMgO2Vbyh5PF+kH1
YuNjWSKFXUrV5YF5xMu/U6vEuNPrd56X/n5OvDjgJC9c2Mr4vUiAwz99xotM69oW
MsDgk7hlaisAzWUlgJ0Hwo2QQJcx837gRjiwNYUCftqUApFQZPi4IKg9W0DU6Vtd
R+TKI0Zi2H6U1KO/nC02aRGKFO4nEVsiTJl0NxlccDEBOJ11VZaNPakg4HCdUGHX
cBCxHnKM9b14cGzIADw5rrviG/AWiiXDPy46j/C7JG5q2Qd/WnueUQycRyleRo0z
W9pEwkFb0tpsrotMfjLgZ1KkwJ1Y0SZ7NNvxHsHIOX6ANQSZkvQwDxm9qG
-----END RSA PRIVATE KEY-----

```

☐ Disable exec ?

SSH Servers

Añadir

Avanzado ▾

Guardar Apply

Luego click en “Añadir” para agregar el servidor apache y rellenamos los campos:

SSH Servers

≡ SSH Server

Name ?

apacheServer-Ubuntu-vm

Hostname ?

192.168.18.50

Username ?

freddy

Remote Directory ?

/

Avanzado ▾

Hacemos un test de conexión:

192.168.18.50

Username ?
freddy

Remote Directory ?
/

Avanzado ▾

❗ Failed to connect or change directory

jenkins.plugins.publish_over.BapPublisherException: Failed to connect and initialize SSH connection. Message: [Failed to connect session for config [apacheServer-Ubuntu-vm]. Message [java.net.ConnectException: Connection refused: connect]]

Test Configuration

Añadir

Sin embargo, no se pudo realizar la conexión, a continuación, se configura el servidor SSH del servidor que tiene el servidor apache:

```
$ sudo nano /etc/ssh/sshd_config
```

Buscamos la instrucción “PasswordAuthentication” y descomentamos y le ponemos “yes”.

```
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware i
# some PAM modules and threads)
KbdInteractiveAuthentication no
```

Reiniciar el servidor ssh:

```
$ sudo systemctl restart sshd
```

Finalmente, la conexión de prueba se realizo correctamente:

SSH Server

Name [?]

apacheServer-Ubuntu-vm

Hostname [?]

192.168.18.50

Username [?]

freddy

Remote Directory [?]

/

Avanzado ▾

Success

Test Configuration

Añadir

Avanzado ▾

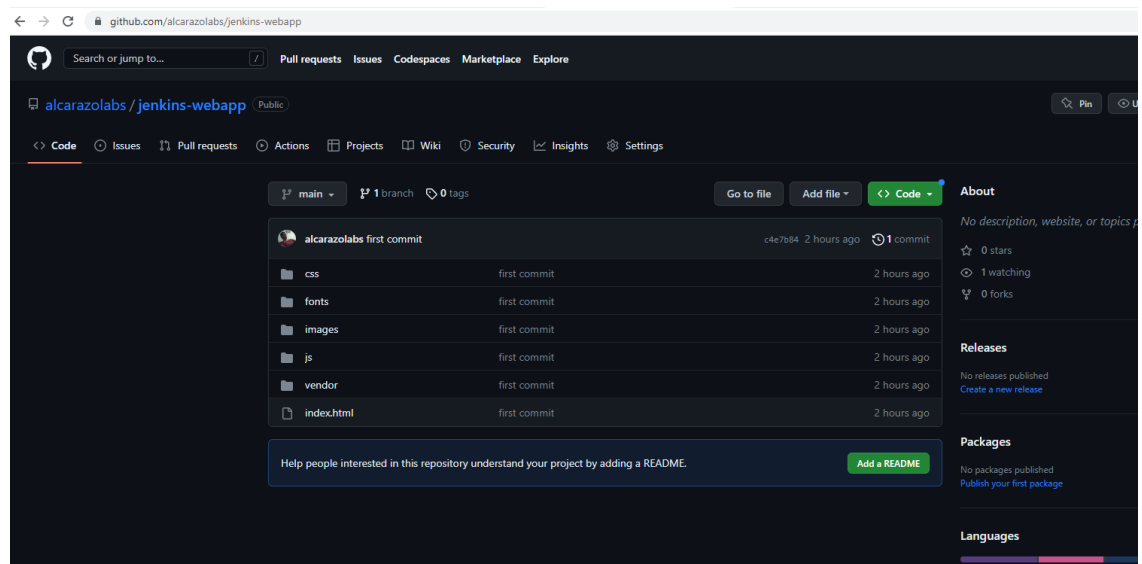
Guardar Apply

En caso de no haberse podido realizar la conexión revisamos los logs del servidor SSH y vemos que tipo de error esta lanzando y lo investigamos:

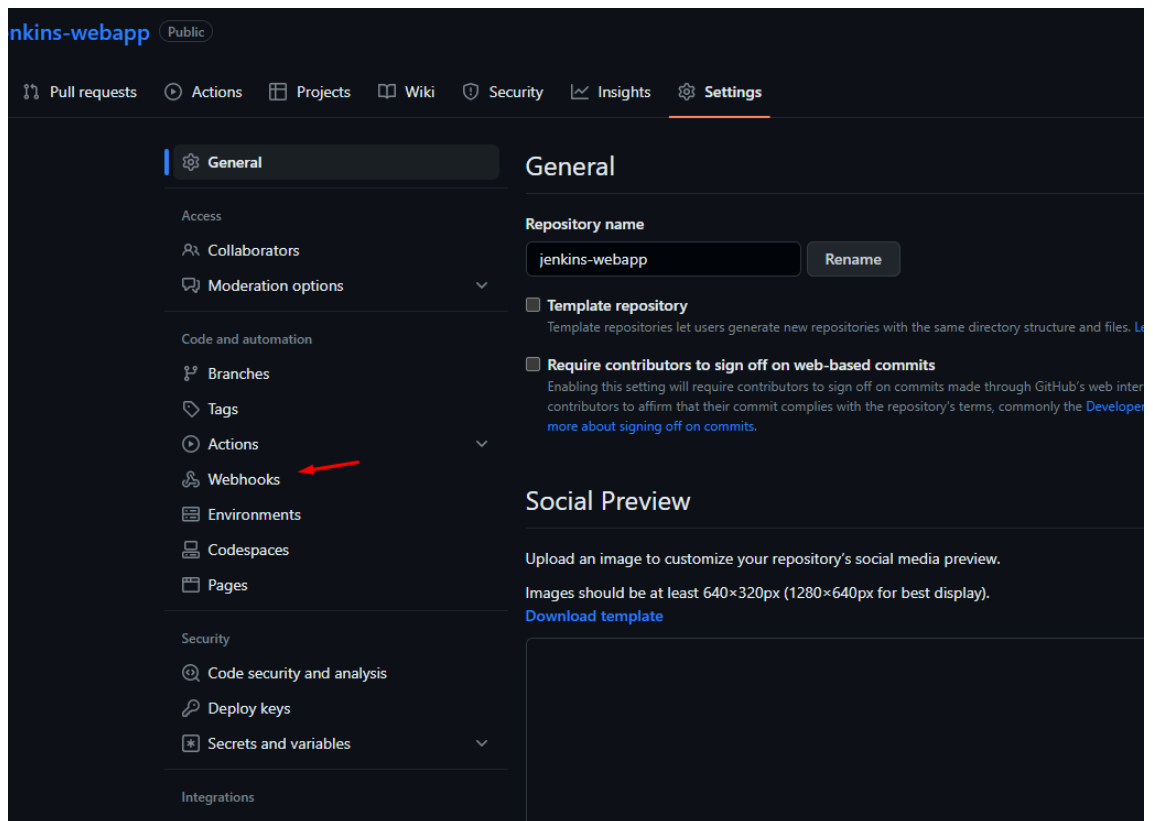
```
$ tail -n 500 /var/log/auth.log | grep 'sshd'
```

3. Configurar repositorio

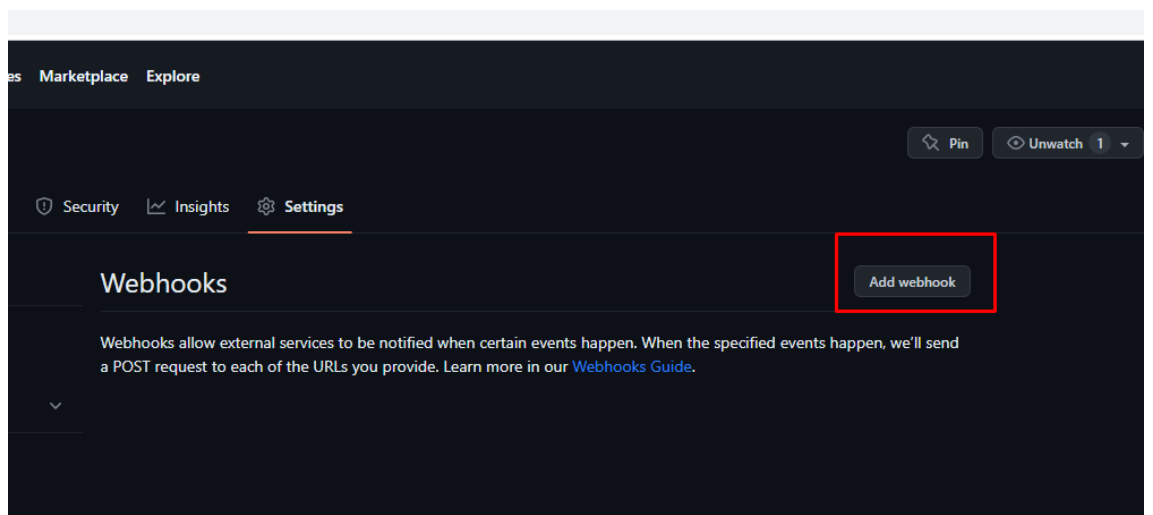
Accedemos a la página del repositorio alojado en github.com y click en settings:



Luego click en Webhooks:



Luego click en “Add webhook”:



Agregamos la dirección del servidor Jenkins más la palabra “github-webhook” al final:

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`http://192.168.18.42:8080/github-webhook/`

Content type

`application/x-www-form-urlencoded`

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ **Active**

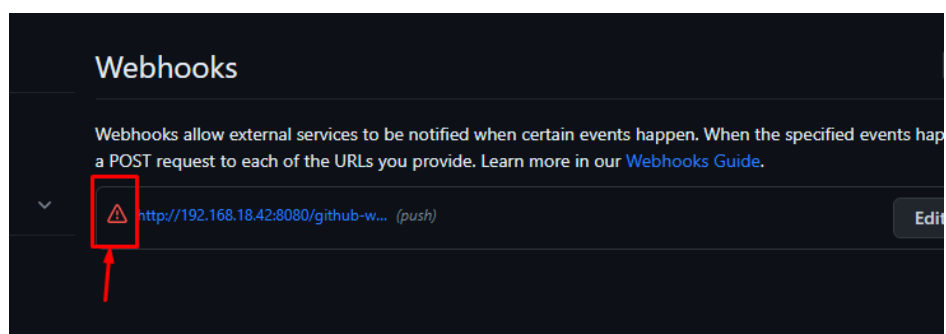
We will deliver event details when this hook is triggered.

Add webhook

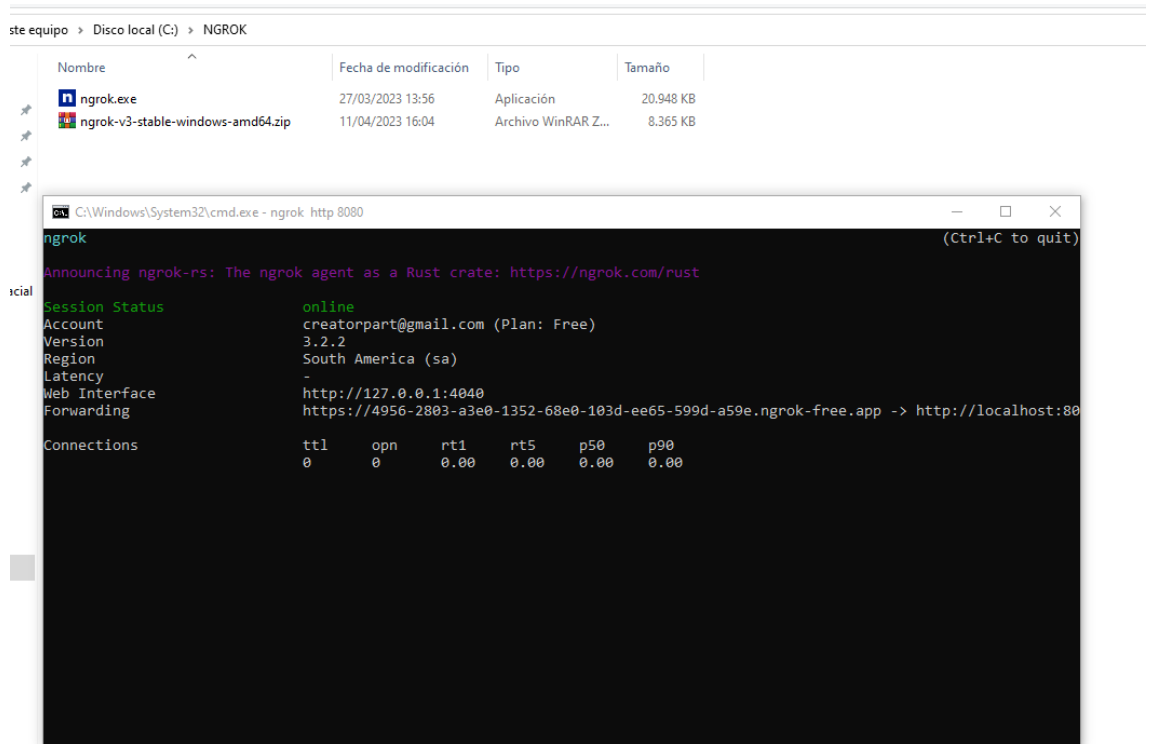
Si no funciona luego tendría que abrir puertos en el router y usar la ip publica ó en su lugar usar el servicio de ngrok para exponer al mundo el puerto 8080.

El content-type le ponemos "application/x-www-form-urlencoded" y el campo "secret" lo dejamos vacío. Además seleccionamos "Just push event" para que se haga deploy cada vez que haya cambios en el repositorio.

Sin embargo al darle "Add webhook", github no pudo conectarse:



Lo que haré es exponer el puerto 8080 de mi pc al mundo con el servicio ngrok.



Webhooks / Manage webhook

Settings

Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`https://4956-2803-a3e0-1352-68e0-103d-ee65-599d-a59e.ngrok-fr`

Content type

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

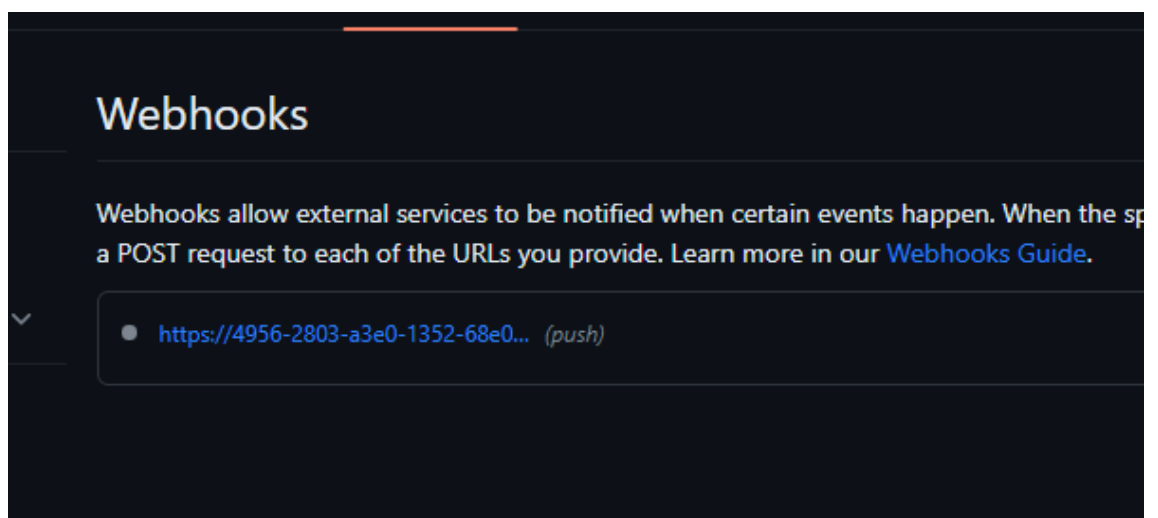
☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

Update webhook Delete webhook

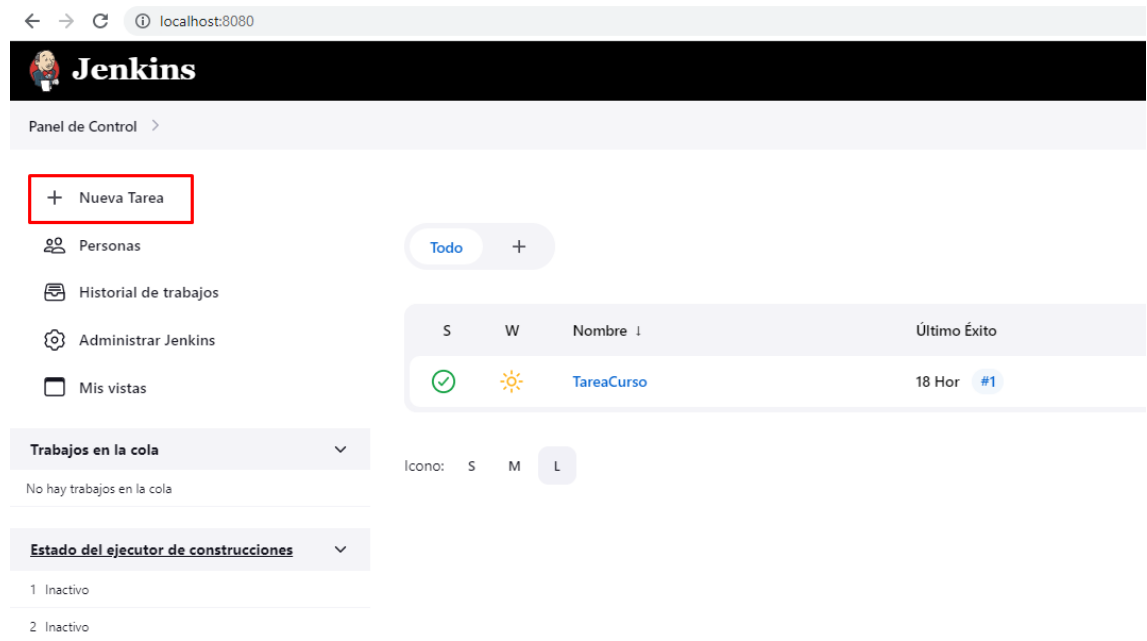
Como se ve en la imagen de arriba, ya se actualizo la dirección url del servidor Jenkins.



Ahora tenemos un punto en lugar de un símbolo de advertencia. Nota si sigue sin actualizarse eliminamos el webhook y volvemos a crear uno.

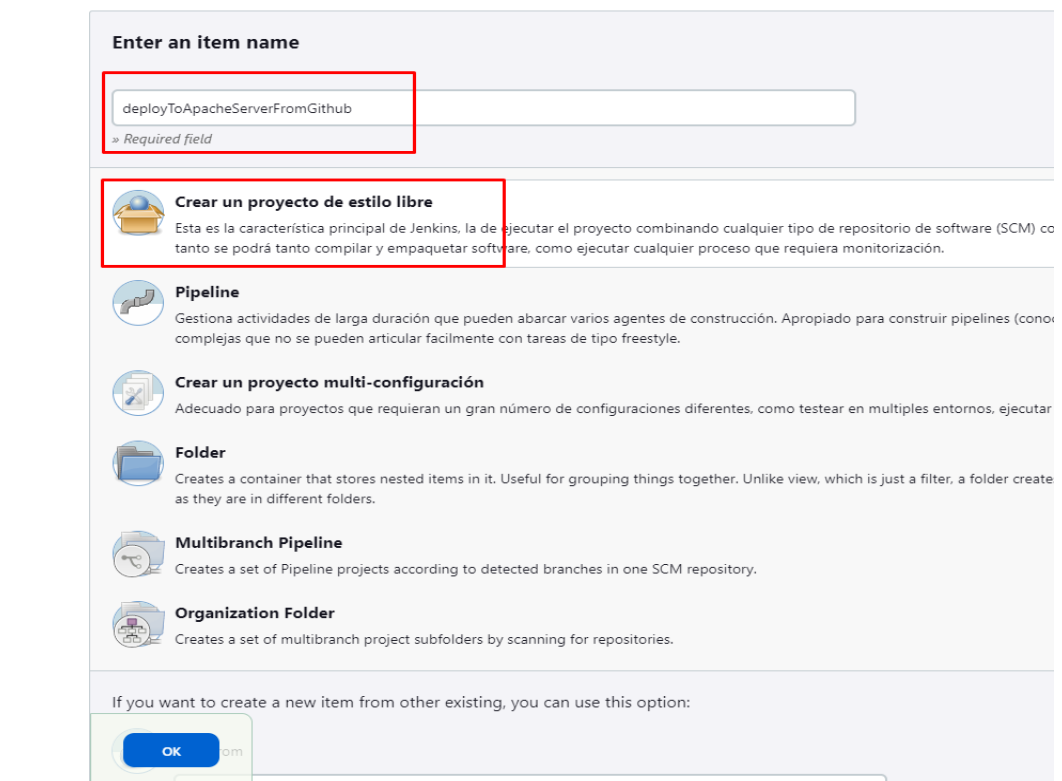
4. Crear proyecto Jenkins

Click en nueva tarea:



The screenshot shows the Jenkins web interface at localhost:8080. The 'Nueva Tarea' button is highlighted with a red box. The interface includes a sidebar with navigation options like 'Personas', 'Historial de trabajos', 'Administrar Jenkins', and 'Mis vistas'. A table displays a job named 'TareaCurso' with a status of 'Último Éxito' and a duration of '18 Hor'. Below the table, there are sections for 'Trabajos en la cola' (No hay trabajos en la cola) and 'Estado del ejecutor de construcciones' (2 Inactivo).

Le escribo un nombre al proyecto y se elige “CREAR UN PROYECTO DE ESTILO LIBRE”:



The screenshot shows the 'Enter an item name' dialog in Jenkins. The input field contains 'deployToApacheServerFromGithub'. Below the input field, the 'Crear un proyecto de estilo libre' option is highlighted with a red box. Other options listed include 'Pipeline', 'Crear un proyecto multi-configuración', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. At the bottom, there is an 'OK' button.

Indicar que es un proyecto github y ponemos la url del repositorio:

General

Descripción

Proyecto que hace deploy de un proyecto web a un servidor php.

[Plain text] [Visualizar](#)

☐ Desechar ejecuciones antiguas ?

☐ Esta ejecución debe parametrizarse ?

☒ GitHub project

Project url ?

<https://github.com/alcarazolabs/jenkins-webapp>

Más abajo indicamos que el origen del código fuente es un Git y luego indicamos la Branch en este caso es “main”:

Configurar el origen del código fuente

☐ Ninguno

☒ Git ?

Repositories ?

Repository URL ?

<https://github.com/alcarazolabs/jenkins-webapp.git>

Credentials ?

- none -

Add ▾

Avanzado ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

El ultimo paso es indicar en la sección “Build Steps” “Send files or execute commands over SSH”

Build Steps

Añadir un nuevo paso ^

Filter

- Ejecutar Ant
- Ejecutar línea de comandos (shell)
- Ejecutar tareas 'maven' de nivel superior
- Ejecutar un comando de Windows
- Invoke Gradle script
- Run with timeout
- Send files or execute commands over SSH**
- Set build status to "pending" on GitHub commit

Elegir el servidor al cual se hará el deploy e indicar el tipo de “Source files” en este caso se puso `**/*.*` si solo deseamos que se escriban archivos con extensión php sería `**/*.php` sin embargo en este caso le puse un asterisco al final para que escriba todo tipo de archivos .html, .js, css etc.

Send files or execute commands over SSH ?

SSH Publishers

SSH Server

Name ?

apacheServer-Ubuntu-vm

Avanzado ▾

Transfers

Transfer Set

Source files ?

`**/*.*`

Files to upload to a server.

The string is a comma separated list of includes for an Ant fileset eg. `'**/*.jar'` (see [Patterns](#) in the Ant manual).
The base directory for this fileset is the workspace.

(desde [Publish Over SSH](#))

Remove prefix ?

Guardar Apply

Luego se indica en que directorio del servidor destino se escribirán los archivos, en este caso como es un servidor apache, se indica que se escriba en la ruta `/var/www/html`

Remove prefix ?

Remote directory ?

/var/www/html/

Exec command ?

❗ Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Avanzado ▾

Add Transfer Set

Guardar Apply

Presionamos Guardar.

Para probar la activación del push modificamos algo en el repositorio o subimos el código al repositorio.

Finalmente hacemos click en “Construir Ahora”.

Panel de Control > deployToApacheServerFromGithub >

Estado Actual

- Cambios
- Zona de Trabajo
- Construir ahora
- Configurar
- Borrar Proyecto
- GitHub
- Rename

Proyecto deployToApacheServerFromGithub

Proyecto que hace deploy de un proyecto web a un servidor php.

Enlaces permanentes

- “Última ejecución (#2) hace 4 Min 18 Seg”
- “Última ejecución estable (#2) hace 4 Min 18 Seg”
- “Última ejecución correcta (#2) hace 4 Min 18 Seg”
- “Last completed build (#2) hace 4 Min 18 Seg”

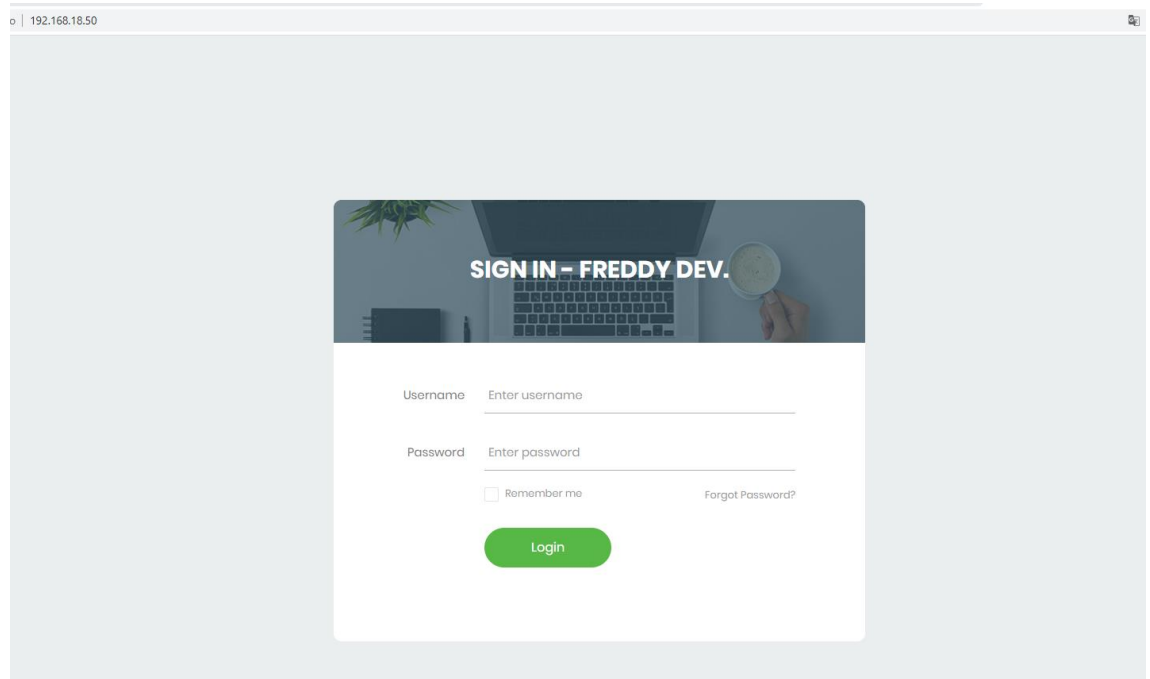
Historia de tareas **Tendencia** ▾

Filter builds... /

✓ #3	11 abr 2023 16:43
✓ #2	11 abr 2023 16:37
✓ #1	11 abr 2023 16:37

Atom feed Para Todos

Atom feed para los errores





Como se puede ver la tarea se ejecuto sin problemas:


← → ↻ ⓘ localhost:8080/job/deployToApacheServerFromGithub/3/


Jenkins


Panel de Control > deployToApacheServerFromGithub > #3


 **Estatus**


 Cambios


 Console Output


 Editar información de la ejecución


 Delete build '#3'


 Git Build Data

 Ejecución previa

 **Compilar #3 (11 abr 2023 16:43:15)**

 **Changes**
1. Update index.html ([commit: baec85c](#)) ([details](#) / [githubweb](#))

 Iniciado por el usuario [Freddy Alcarazo Ibañez](#)

 **Revision:** baec85cbd2bd5ab611eec34e4a5c5dc0377029a8
Repository: <https://github.com/alcarazolabs/jenkins-webapp.git>

- refs/remotes/origin/main

← → ↻ localhost:8080/job/deployToApacheServerFromGithub/3/console

Jenkins 🔍 búsqueda (CTRL+K)

Panel de Control > deployToApacheServerFromGithub > #3 > Salida de consola

📄 Estado

</> Cambios

📄 Console Output

📄 View as plain text

✎ Editar información de la ejecución

🗑 Delete build '#3'

🔗 Git Build Data

⏮ Ejecución previa

✓ **Salida de consola**

```
Lanzada por el usuario Freddy Alcarazo Ibañez
Running as SYSTEM
Ejecutando en el espacio de trabajo C:\ProgramData\Jenkins\workspace\deployToApacheServerFromGithub
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\deployToApacheServerFromGithub\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/alcarazolabs/jenkins-webapp.git # timeout=10
Fetching upstream changes from https://github.com/alcarazolabs/jenkins-webapp.git
> git.exe --version # timeout=10
> git --version # 'git version 2.38.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/alcarazolabs/jenkins-webapp.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision baec85cbd2bd5ab611eec34e4a5c5dc0377029a8 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f baec85cbd2bd5ab611eec34e4a5c5dc0377029a8 # timeout=10
Commit message: "Update index.html"
> git.exe rev-list --no-walk 353d257f26157ade8b397c5e957f569efdcfa572 # timeout=10
SSH: Connecting from host [DESKTOP-C880H39]
SSH: Connecting with configuration [apacheServer-Ubuntu-vm] ...
SSH: Disconnecting configuration [apacheServer-Ubuntu-vm] ...
SSH: Transferred 105 file(s)
Build step 'Send files or execute commands over SSH' changed build result to SUCCESS
Finished: SUCCESS
```

Deploy mediante “Declarative Pipeline”

Pasos:

Creamos una nueva tarea:

Enter an item name

deployPipelineToApacheServerFromGithub

Required field

Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

Pipeline

Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines complejas que no se pueden articular fácilmente con tareas de tipo freestyle.

Crear un proyecto multi-configuración

Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ej.

Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates items as they are in different folders.

Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

OK

☐ Desechar ejecuciones antiguas ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☐ Esta ejecución debe parametrizarse ?

☒ GitHub project

Project url ?

<https://github.com/alcarazolabs/jenkins-webapp>

Avanzado ▾

En la sección de Build Triggers elegimos “Github hook trigger for GITScm polling”

Build Triggers

- ☐ Construir tras otros proyectos ?
- ☐ Consultar repositorio (SCM) ?
- ☐ Ejecutar periódicamente ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Periodo de espera ?
- ☐ Lanzar ejecuciones remotas (ejem: desde 'scripts') ?

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Deploy-Github') {
6       steps {
7         echo 'Hello World'
8       }
9     }
10  }
11 }
12
```

- ☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Para copiar el código de que irá dentro de steps hacemos click en “Pipeline Syntax”

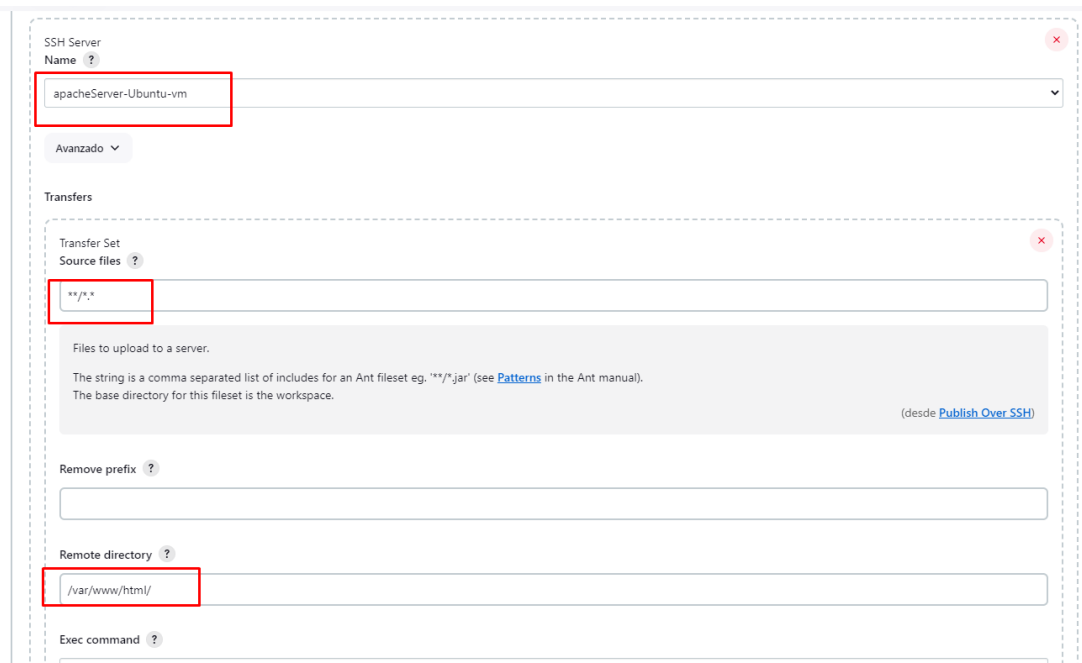
En la página que se a aperturado elegimos ahora el siguiente Sample step:

Steps

Sample Step

sshPublisher: Send build artifacts over SSH

Luego seleccionar el nombre del servidor a donde realizará el deploy, elegir el tipo de archivos a subir (.**) y el directorio a donde se subirán los archivos (var/www/html)



SSH Server

Name ?

apacheServer-Ubuntu-vm

Avanzado ▾

Transfers

Transfer Set ?

Source files ?

**/*.*

Files to upload to a server.

The string is a comma separated list of includes for an Ant fileset eg. "**/*jar" (see [Patterns](#) in the Ant manual).

The base directory for this fileset is the workspace.

(desde [Publish Over SSH](#))

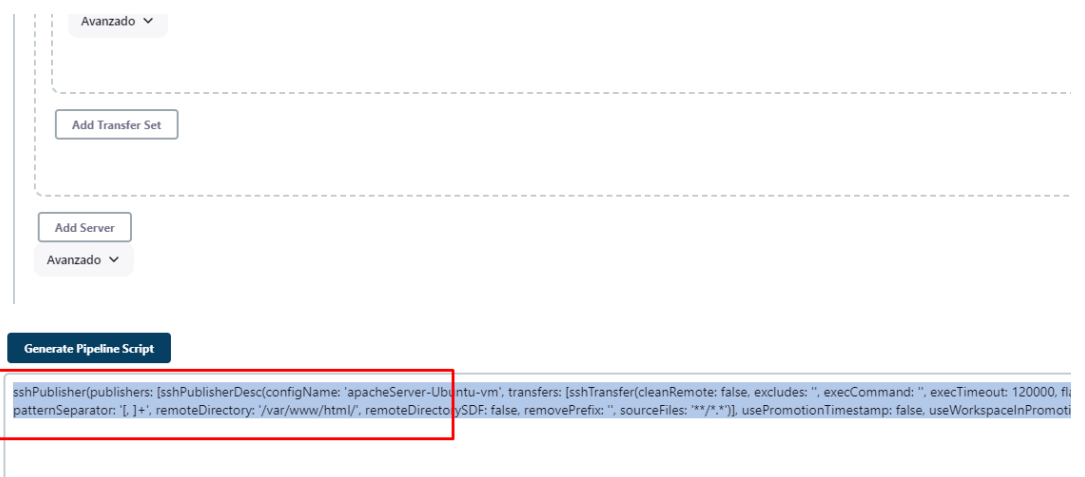
Remove prefix ?

Remote directory ?

/var/www/html/

Exec command ?

Más abajo presionamos el botón “Generar pipeline script” y copiamos.



Avanzado ▾

Add Transfer Set

Add Server

Avanzado ▾

Generate Pipeline Script

```
sshPublisher(publishers: [sshPublisherDesc(configName: 'apacheServer-Ubuntu-vm', transfers: [sshTransfer(cleanRemote: false, excludes: '', execCommand: '', execTimeout: 120000, fi
patternSeparator: '[, ]+', remoteDirectory: '/var/www/html/', remoteDirectorySDF: false, removePrefix: '', sourceFiles: '**/*.*')], usePromotionTimestamp: false, useWorkspaceInPromoti
```

Y pegamos dicho script en donde nos quedamos:

Pipeline script

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Deploy-Github') {
6       steps {
7         sshPublisher(publishers: [sshPublisherDesc(configName: 'apacheServer-Ubuntu-vm', transfers: [sshTransfer(cleanRemote: false, excludes
8       }
9     }
10  }
11 }
12 }
```

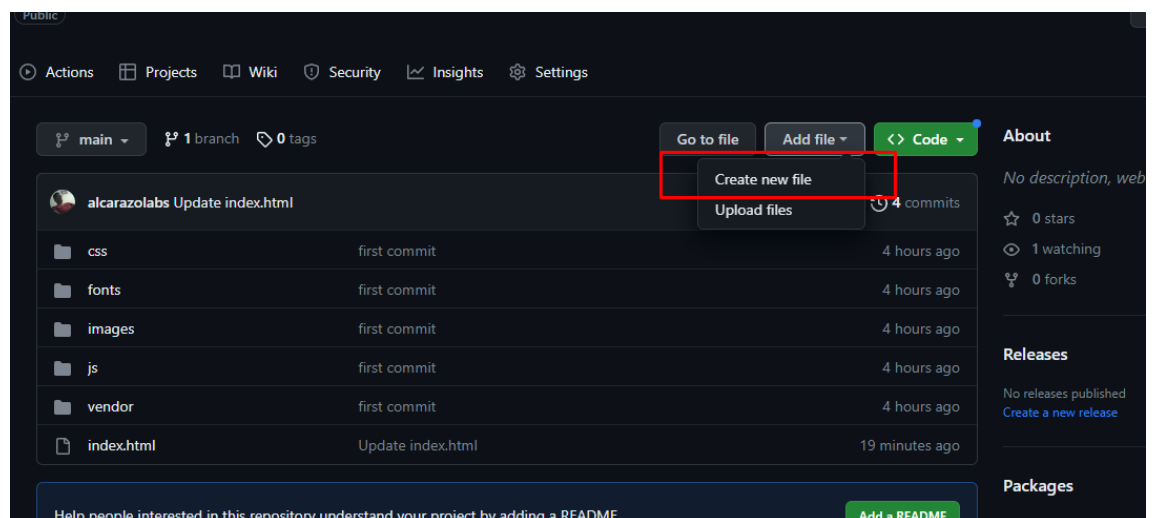
☒ Use Groovy Sandbox ?

Ahora copiamos todo este script:

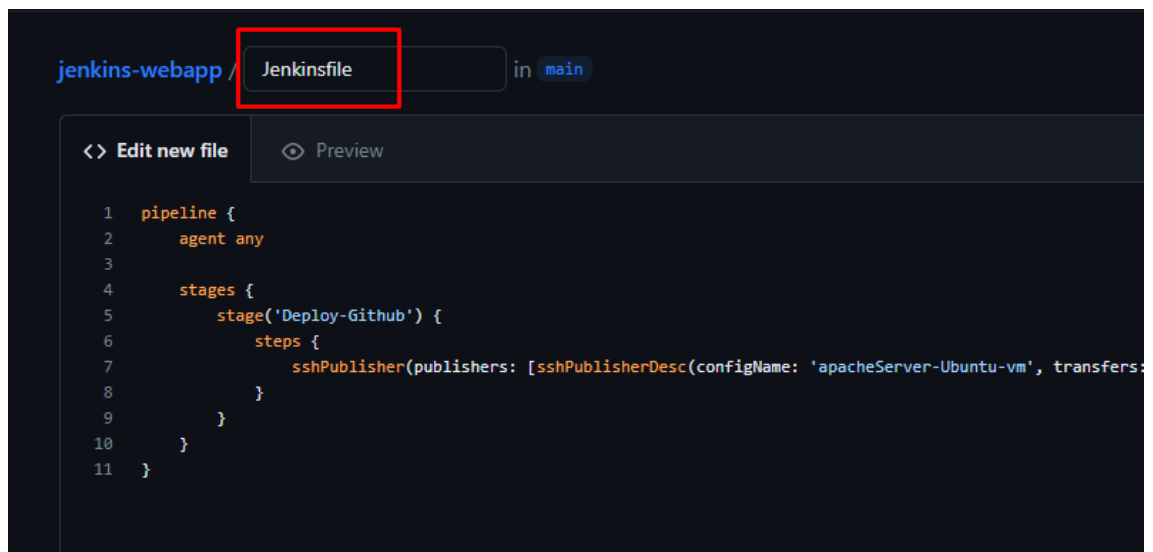
Pipeline script

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Deploy-Github') {
6       steps {
7         sshPublisher(publishers: [sshPublisherDesc(configName: 'apacheServer-Ubuntu-vm', transfers: [sshTransfer(cleanRemote: false, excludes
8       }
9     }
10  }
11 }
12 }
```

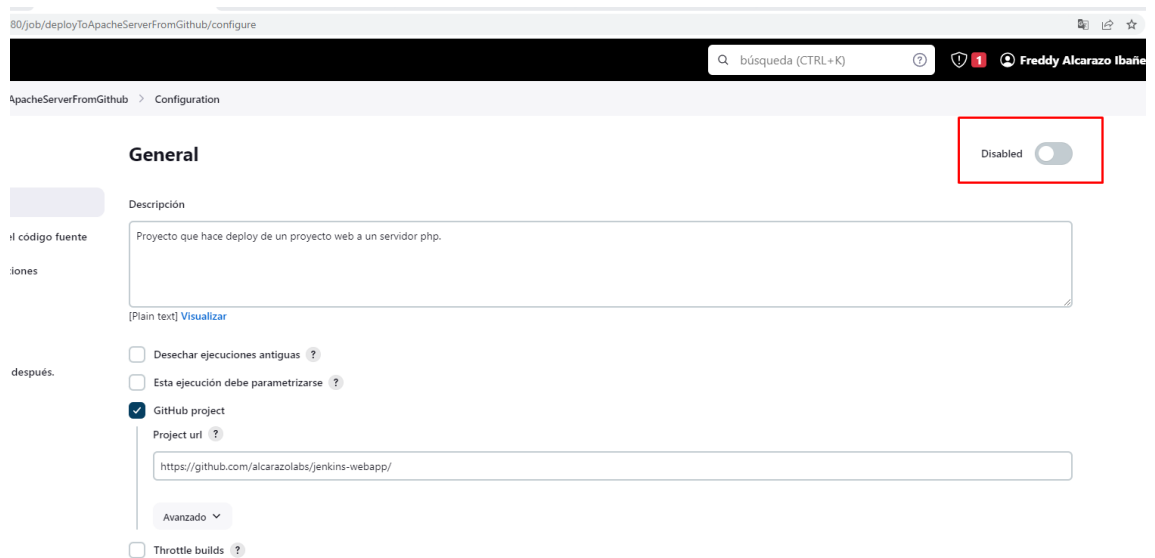
Nos vamos a la página del repositorio:



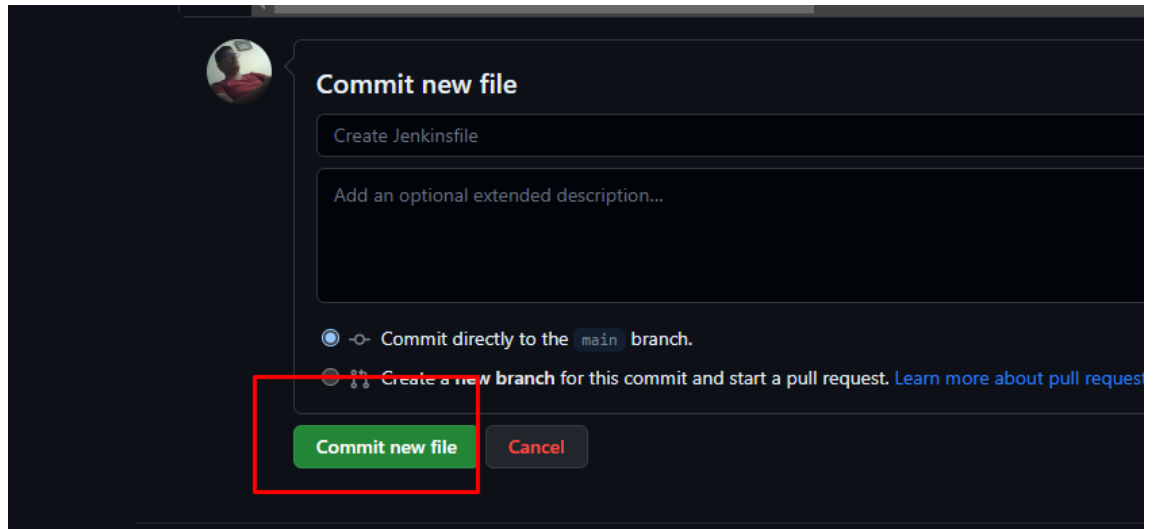
Nombramos a este archivo como “Jenkinsfile” sin extensión:



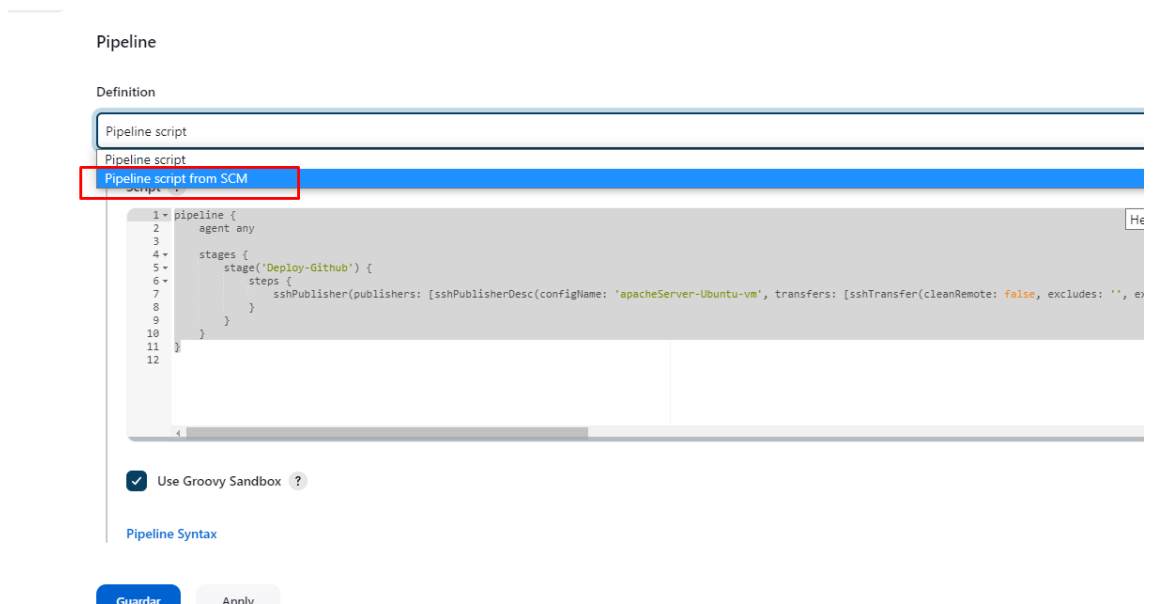
Antes de continuar, volvemos al dashboard de Jenkins y elegimos el proyecto creado anteriormente y lo deshabilitamos.



Ahora ya podemos hacer “commit” del nuevo archivo creado en la página del repositorio:



Luego cambios de “Pipeline script” a “Pipeline script from SCM”:



Ahora elegimos el tipo de SCM como “GIT” y luego pegamos la url del repositorio .git e indicamos la branch en este caso “main”.

verfrommainud / Configuration

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/alcarazolabs/jenkins-webapp.git

Credentials ?

- none -

Add

Avanzado

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Navegador del repositorio ?

Guardar Apply

Ahora tenemos que hacer unos cambios en el proyecto web. Luego hacer click en construir ahora a la tarea creada:

Status

</> Changes

▶ Construir ahora

⚙ Configurar

🗑 Borrar Pipeline

🔍 Full Stage View

🔄 GitHub

✎ Rename

❓ Pipeline Syntax

📄 GitHub Hook Log

☀ Historia de tareas

Tendencia ▼

🔍 Filter builds...

/

#2

11 abr 2023 17:16

#1

11 abr 2023 17:16

Atom feed Para Todos

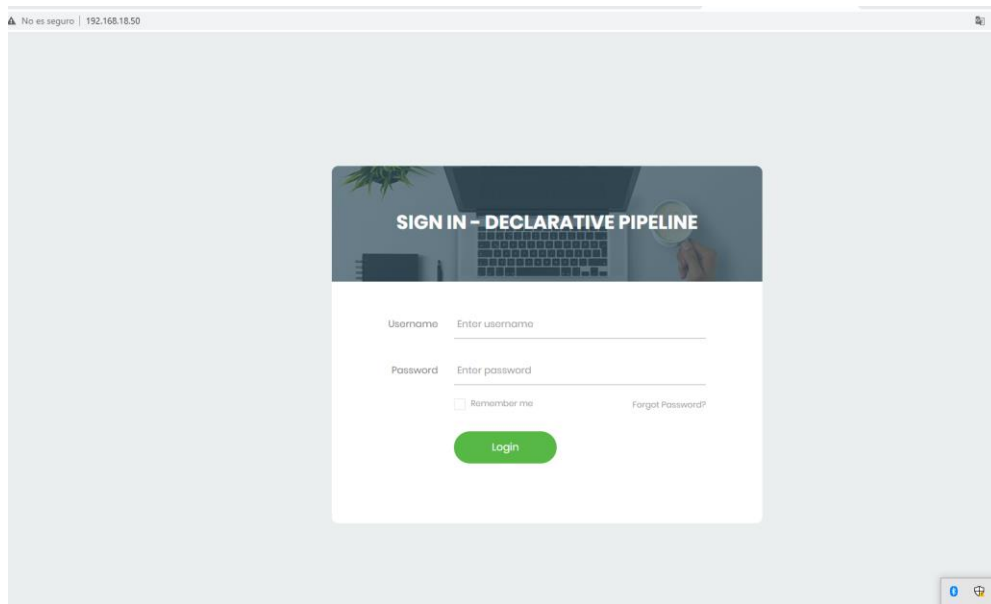
Atom feed para los errores

Pipeline deployPipelineToApacheServerFromGithub

Stage View

	Declarative: Checkout SCM	Deploy-Github
Average stage times: (Average full run time: ~5s)	2s	988ms
#2 abr 11 17:16 No Changes	2s	849ms
#1 abr 11 17:16 No Changes	2s	1s

Enlaces permanentes



Sim embargo esto debió de ser automático ahora vamos a intentar hacer un nuevo cambio y ver si la tarea se construye automáticamente:

Status

</> Changes

▶ Construir ahora

⚙ Configurar

🗑 Borrar Pipeline

🔍 Full Stage View

🐙 GitHub

✎ Rename

🔍 Pipeline Syntax

📄 GitHub Hook Log

Pipeline deployPipelineToApacheServerFromGithub

Stage View

	Declarative: Checkout SCM	Deploy-Github
Average stage times: (Average full run time: ~5s)	2s	988ms
#2 abr 11 17:16 No Changes	2s	849ms
#1 abr 11 17:16 No Changes	2s	1s

Enlaces permanentes

- "Última ejecución (#2) hace 1 Min 46 Seg"
- "Última ejecución estable (#2) hace 1 Min 46 Seg"
- "Última ejecución correcta (#2) hace 1 Min 46 Seg"
- "Last completed build (#2) hace 1 Min 46 Seg"

Historia de tareas

Tendencia ▼

Filter builds...

/

#3

(pending—En el periodo de gracia. Termina en 2.9 Seg)

#2

11 abr 2023 17:16

#1

11 abr 2023 17:16

Atom feed Para Todos

Atom feed para los errores

En el segundo intento la actualización fue automática no se necesitó construirla manualmente.

Conclusiones y Recomendaciones

Conclusiones:

- El orquestador Jenkins permitió hacer deploy de manera automática una aplicación web alojada en un repositorio github.
- Acelero el proceso de entrega de builds.
- Los deploys se realizan de manera organizada.
- Se sistematizo el despliegue de una aplicación web.

Recomendaciones

- Tener instalado Jenkins en un servidor en la nube.
- Configurar correctamente el servidor ssh para poder hacer la conexión con Jenkins.
- Trabajar en entornos Linux.