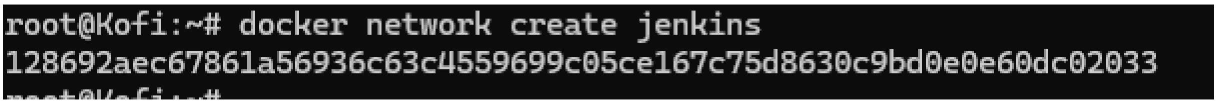
Primera parte

Create a bridge network in Docke

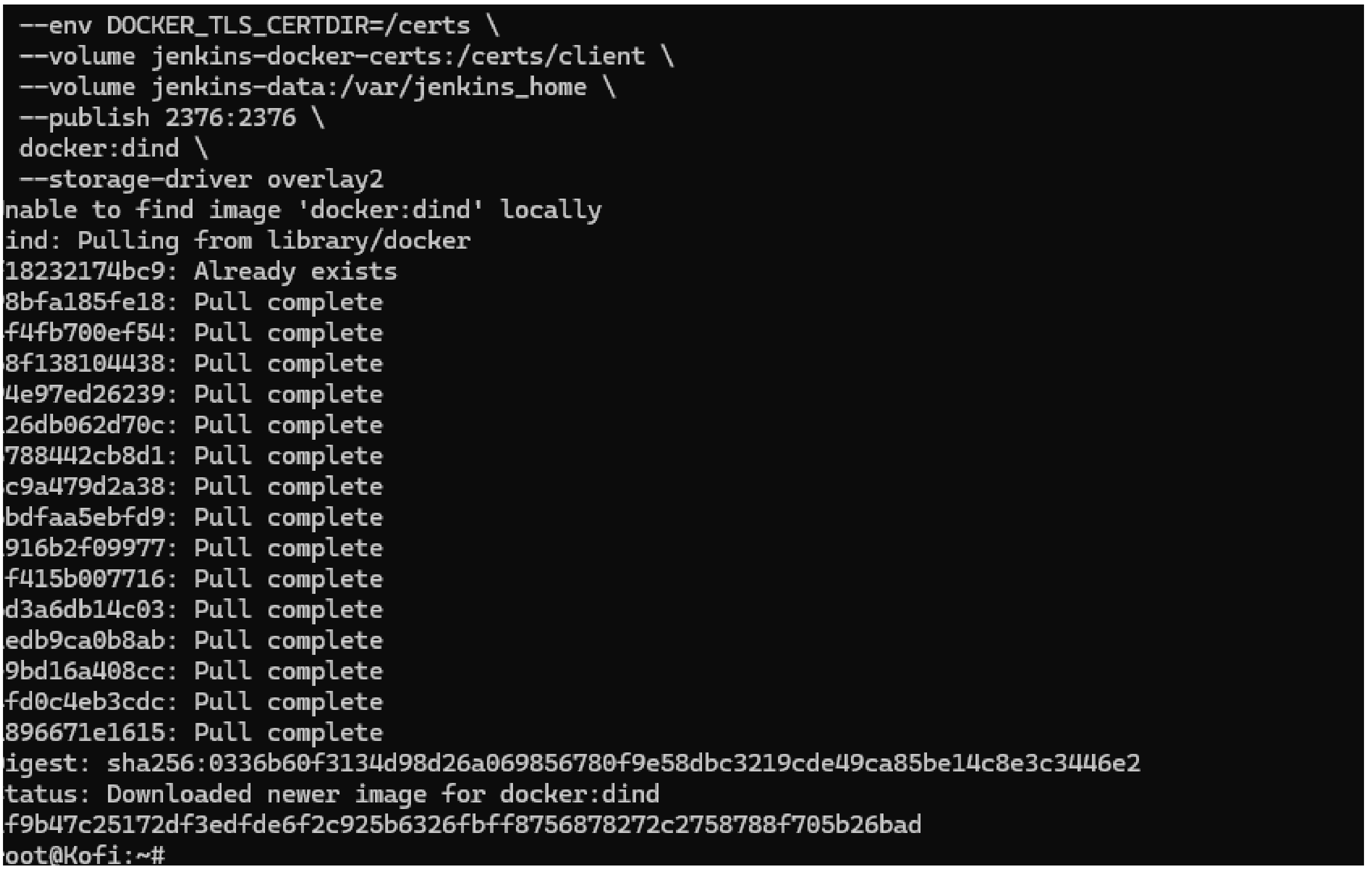
```
docker network create jenkins
```



```
root@Kofi:~# docker network create jenkins
128692aec67861a56936c63c4559699c05ce167c75d8630c9bd0e0e60dc02033
root@Kofi:~#
```

In order to execute Docker commands inside Jenkins nodes, download and run the docker:dind Docker image

```
docker run \
  --name jenkins-docker \
  --rm \
  --detach \
  --privileged \
  --network jenkins \
  --network-alias docker \
  --env DOCKER_TLS_CERTDIR=/certs \
  --volume jenkins-docker-certs:/certs/client \
  --volume jenkins-data:/var/jenkins_home \
  --publish 2376:2376 \
  docker:dind \
  --storage-driver overlay2
```



```
 --env DOCKER_TLS_CERTDIR=/certs \
 --volume jenkins-docker-certs:/certs/client \
 --volume jenkins-data:/var/jenkins_home \
 --publish 2376:2376 \
 docker:dind \
 --storage-driver overlay2
nable to find image 'docker:dind' locally
ind: Pulling from library/docker
18232174bc9: Already exists
8bfa185fe18: Pull complete
f4fb700ef54: Pull complete
8f138104438: Pull complete
4e97ed26239: Pull complete
26db062d70c: Pull complete
788442cb8d1: Pull complete
c9a479d2a38: Pull complete
bdfaa5ebfd9: Pull complete
916b2f09977: Pull complete
f415b007716: Pull complete
d3a6db14c03: Pull complete
edb9ca0b8ab: Pull complete
9bd16a408cc: Pull complete
fd0c4eb3cdc: Pull complete
896671e1615: Pull complete
igest: sha256:0336b60f3134d98d26a069856780f9e58dbc3219cde49ca85be14c8e3c3446e2
tatus: Downloaded newer image for docker:dind
f9b47c25172df3edfde6f2c925b6326fbff8756878272c2758788f705b26bad
oot@Kofi:~#
```

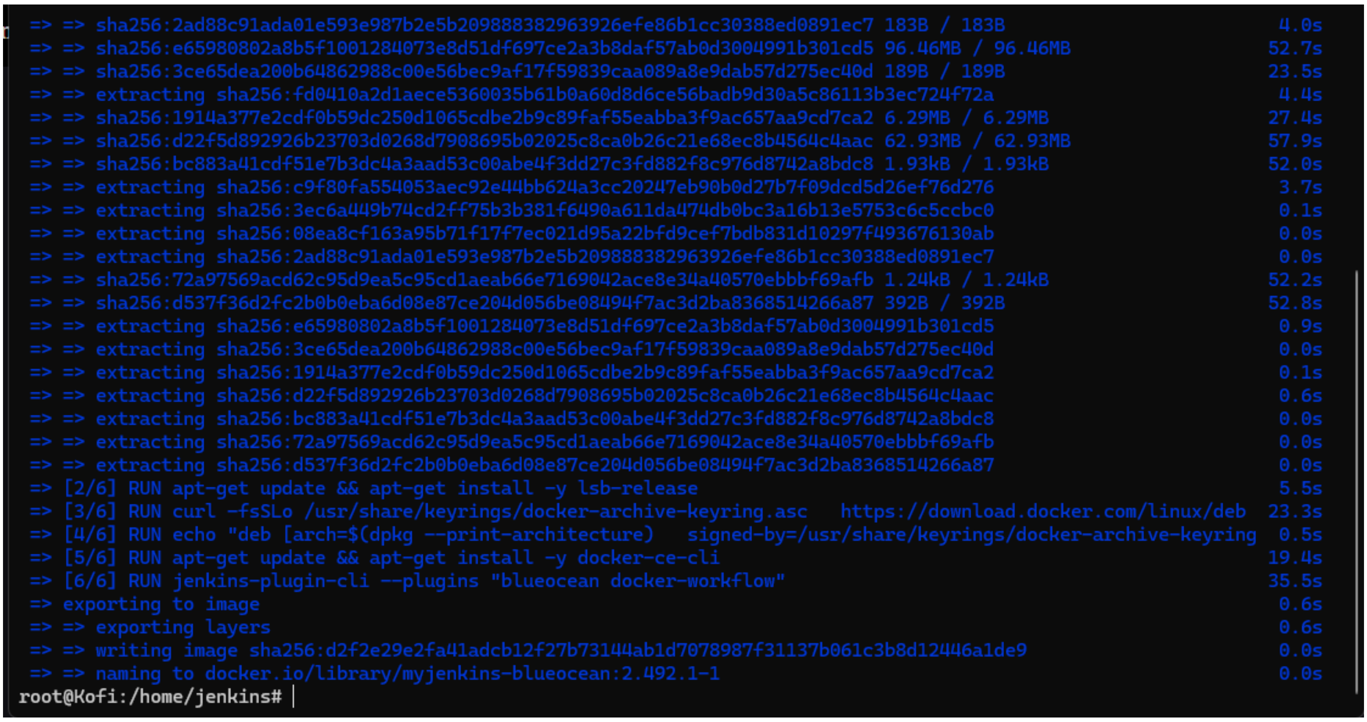
Customize the official Jenkins Docker image

```
cd ..
nano Dockerfile
```

```
    FROM jenkins/jenkins:2.492.1-jdk17
    USER root
    RUN apt-get update && apt-get install -y lsb-release
    RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
    https://download.docker.com/linux/debian/gpg
    RUN echo "deb [arch=$(dpkg --print-architecture) \
    signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
    https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
    RUN apt-get update && apt-get install -y docker-ce-cli
    USER jenkins
    RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"
```

Build a new docker image from this Dockerfile, and assign the image a meaningful name, such as "myjenkins-blueocean:2.492.1-1":

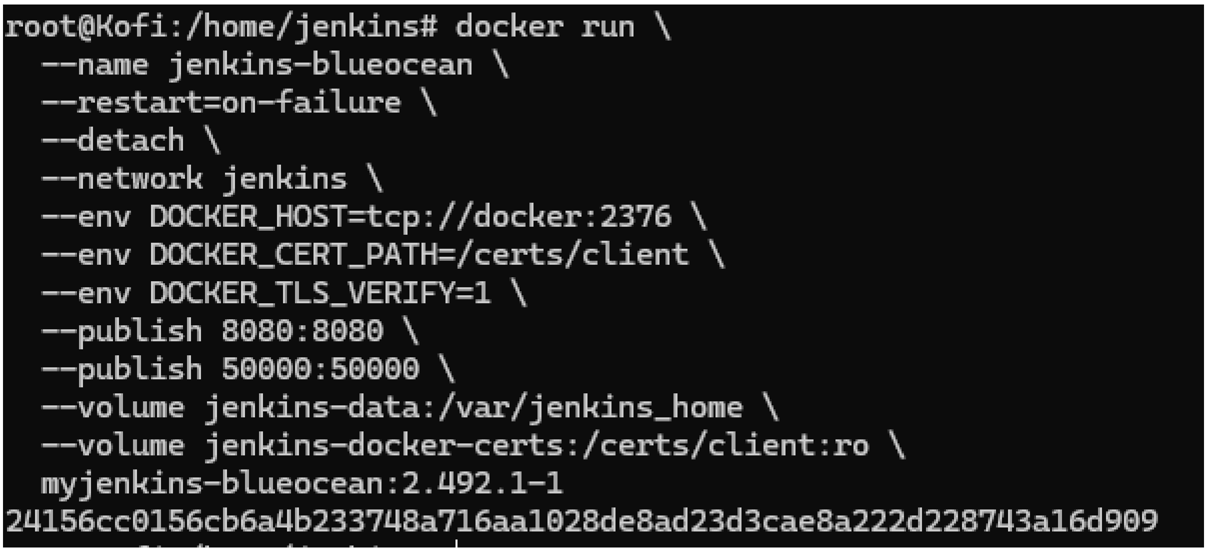
```
docker build -t myjenkins-blueocean:2.492.1-1 .
```



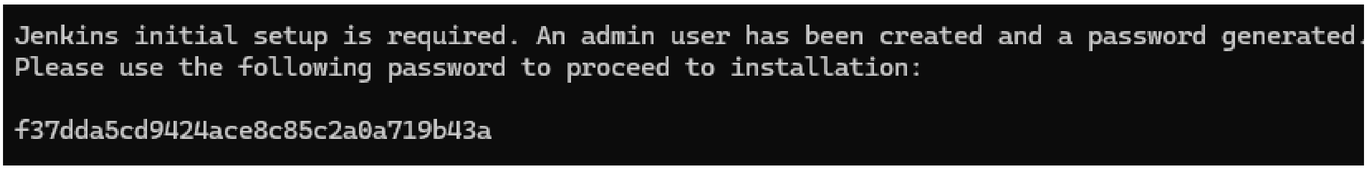
```
 => => sha256:2ad88c91ada01e593e987b2e5b209888382963926efe86b1cc30388ed0891ec7 183B / 183B                       4.0s
 => => sha256:e65980802a8b5f1001284073e8d51df697ce2a3b8daf57ab0d3004991b301cd5 96.46MB / 96.46MB                52.7s
 => => sha256:3ce65dea200b64862988c00e56bec9af17f59839caa089a8e9dab57d275ec40d 189B / 189B                      23.5s
 => => extracting sha256:fd0410a2d1aece5360035b61b0a60d8d6ce56badb9d30a5c86113b3ec724f72a                         4.4s
 => => sha256:1914a377e2cdf0b59dc250d1065cdbe2b9c89faf55eabba3f9ac657aa9cd7ca2 6.29MB / 6.29MB                  27.4s
 => => sha256:d22f5d892926b23703d0268d7908695b02025c8ca0b26c21e68ec8b4564c4aac 62.93MB / 62.93MB                57.9s
 => => sha256:bc883a41cdf51e7b3dc4a3aad53c00abe4f3dd27c3fd882f8c976d8742a8bdc8 1.93kB / 1.93kB                  52.0s
 => => extracting sha256:c9f80fa554053aec92e44bb624a3cc20247eb90b0d27b7f09dcd5d26ef76d276                         3.7s
 => => extracting sha256:3ec6a449b74cd2ff75b3b381f6490a611da474db0bc3a16b13e5753c6c5ccbc0                         0.1s
 => => extracting sha256:08ea8cf163a95b71f17f7ec021d95a22bfd9cef7bdb831d10297f493676130ab                         0.0s
 => => extracting sha256:2ad88c91ada01e593e987b2e5b209888382963926efe86b1cc30388ed0891ec7                         0.0s
 => => sha256:72a97569acd62c95d9ea5c95cd1aeab66e7169042ace8e34a40570ebbbf69afb 1.24kB / 1.24kB                  52.2s
 => => sha256:d537f36d2fc2b0b0eba6d08e87ce204d056be08494f7ac3d2ba8368514266a87 392B / 392B                      52.8s
 => => extracting sha256:e65980802a8b5f1001284073e8d51df697ce2a3b8daf57ab0d3004991b301cd5                         0.9s
 => => extracting sha256:3ce65dea200b64862988c00e56bec9af17f59839caa089a8e9dab57d275ec40d                         0.0s
 => => extracting sha256:1914a377e2cdf0b59dc250d1065cdbe2b9c89faf55eabba3f9ac657aa9cd7ca2                         0.1s
 => => extracting sha256:d22f5d892926b23703d0268d7908695b02025c8ca0b26c21e68ec8b4564c4aac                         0.6s
 => => extracting sha256:bc883a41cdf51e7b3dc4a3aad53c00abe4f3dd27c3fd882f8c976d8742a8bdc8                         0.0s
 => => extracting sha256:72a97569acd62c95d9ea5c95cd1aeab66e7169042ace8e34a40570ebbbf69afb                         0.0s
 => => extracting sha256:d537f36d2fc2b0b0eba6d08e87ce204d056be08494f7ac3d2ba8368514266a87                         0.0s
 => [2/6] RUN apt-get update && apt-get install -y lsb-release                                                    5.5s
 => [3/6] RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc   https://download.docker.com/linux/deb  23.3s
 => [4/6] RUN echo "deb [arch=$(dpkg --print-architecture)   signed-by=/usr/share/keyrings/docker-archive-keyring  0.5s
 => [5/6] RUN apt-get update && apt-get install -y docker-ce-cli                                                 19.4s
 => [6/6] RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"                                           35.5s
 => exporting to image                                                                                            0.6s
 => => exporting layers                                                                                           0.6s
 => => writing image sha256:d2f2e29e2fa41adcb12f27b73144ab1d7078987f31137b061c3b8d12446a1de9                      0.0s
 => => naming to docker.io/library/myjenkins-blueocean:2.492.1-1                                                  0.0s
root@Kofi:/home/jenkins# 
```

Run your own myjenkins-blueocean:2.492.1-1 image as a container in Docker

```
docker run \
  --name jenkins-blueocean \
  --restart=on-failure \
  --detach \
  --network jenkins \
  --env DOCKER_HOST=tcp://docker:2376 \
  --env DOCKER_CERT_PATH=/certs/client \
  --env DOCKER_TLS_VERIFY=1 \
  --publish 8080:8080 \
  --publish 50000:50000 \
  --volume jenkins-data:/var/jenkins_home \
  --volume jenkins-docker-certs:/certs/client:ro \
  myjenkins-blueocean:2.492.1-1
```



```
root@Kofi:/home/jenkins# docker run \
  --name jenkins-blueocean \
  --restart=on-failure \
  --detach \
  --network jenkins \
  --env DOCKER_HOST=tcp://docker:2376 \
  --env DOCKER_CERT_PATH=/certs/client \
  --env DOCKER_TLS_VERIFY=1 \
  --publish 8080:8080 \
  --publish 50000:50000 \
  --volume jenkins-data:/var/jenkins_home \
  --volume jenkins-docker-certs:/certs/client:ro \
  myjenkins-blueocean:2.492.1-1
24156cc0156cb6a4b233748a716aa1028de8ad23d3cae8a222d228743a16d909
```

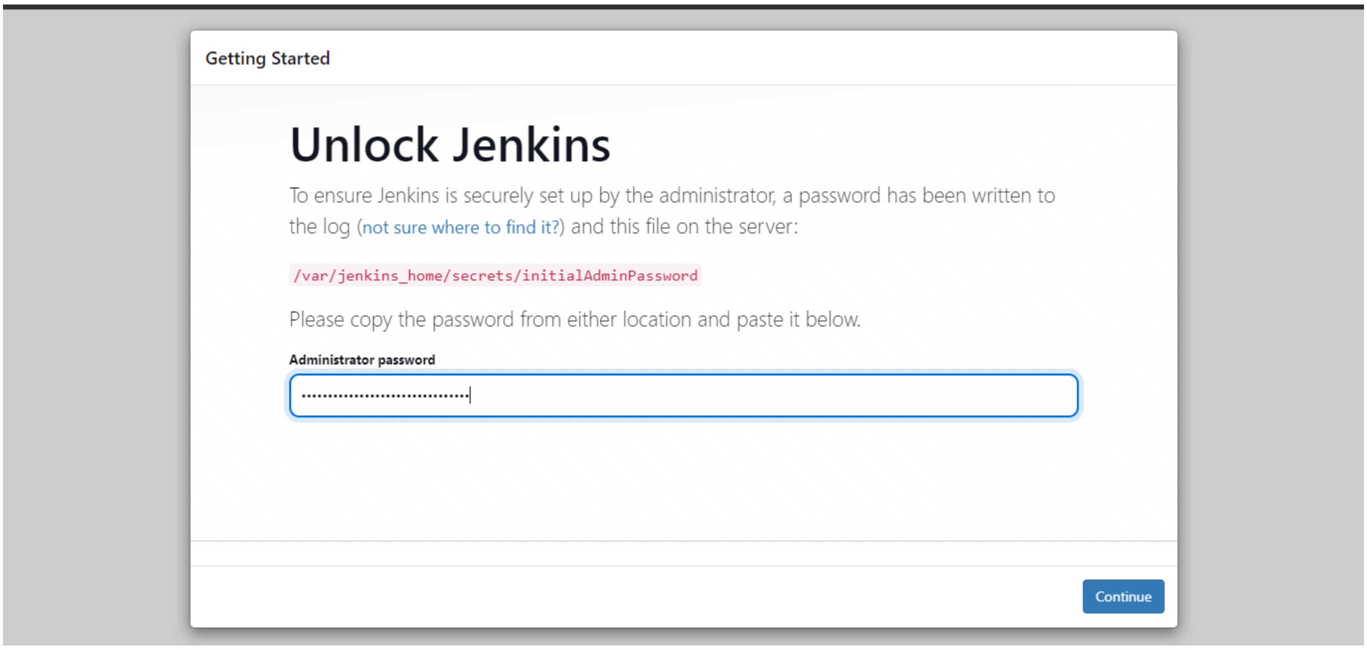


```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

f37dda5cd9424ace8c85c2a0a719b43a
```

```
docker logs jenkins-blueocean
```

```
752b699f61024c06aad14779353911dc
```

On the Unlock Jenkins page, paste this password into the Administrator password field and click Continue.



Getting Started

Unlock Jenkins

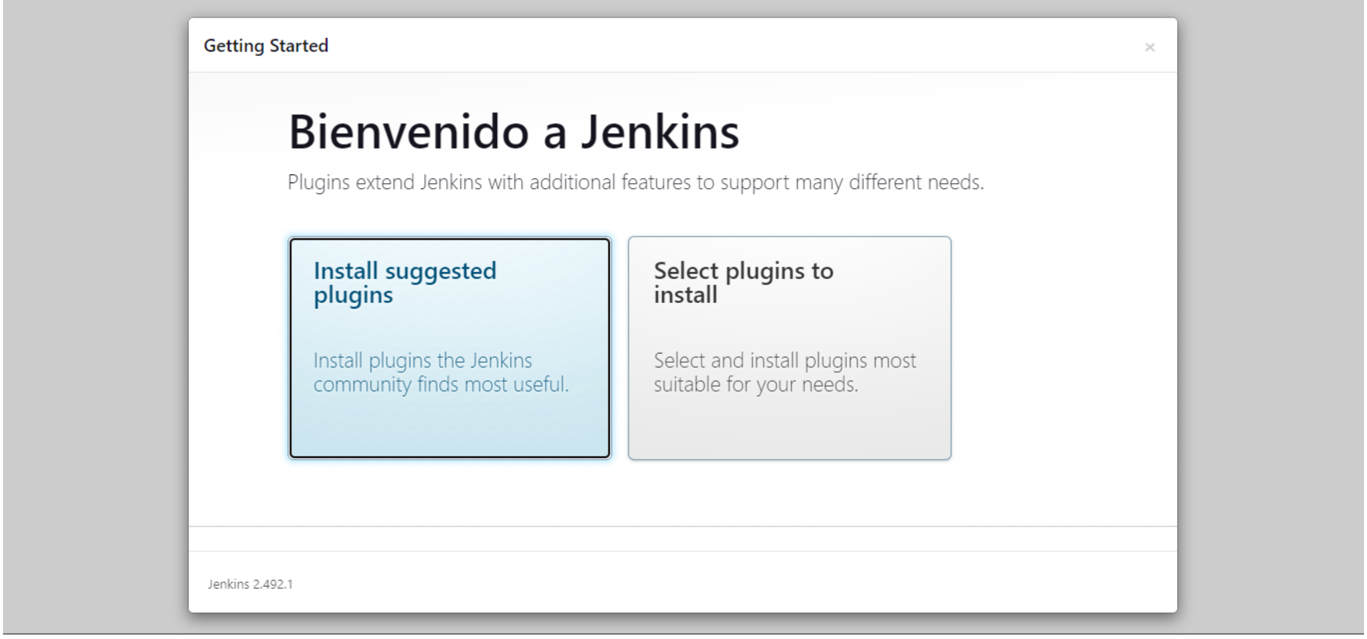
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/jenkins_home/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password: ································

After unlocking Jenkins, the Customize Jenkins page appears. Install suggested plugins



Getting Started

Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

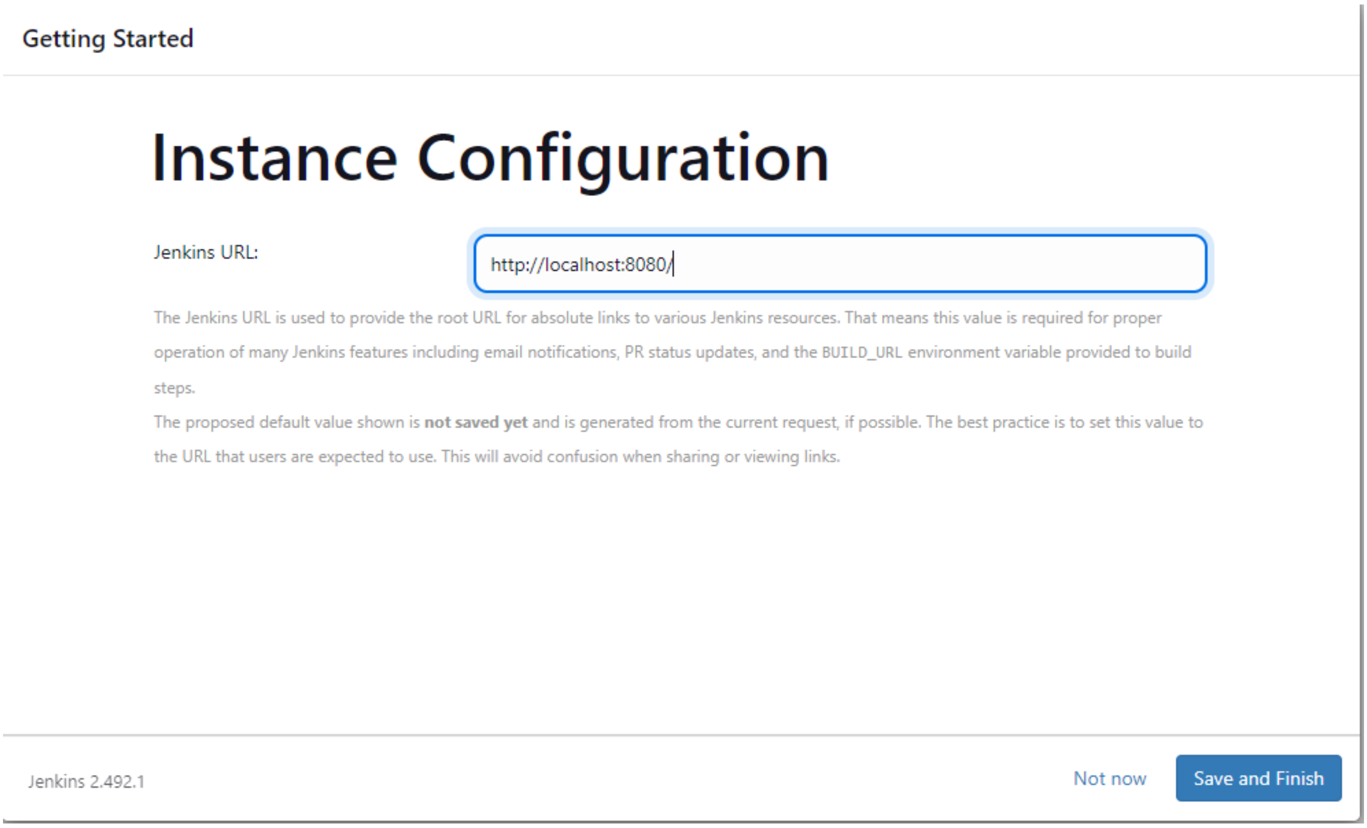
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.492.1

Skip



Getting Started

Instance Configuration

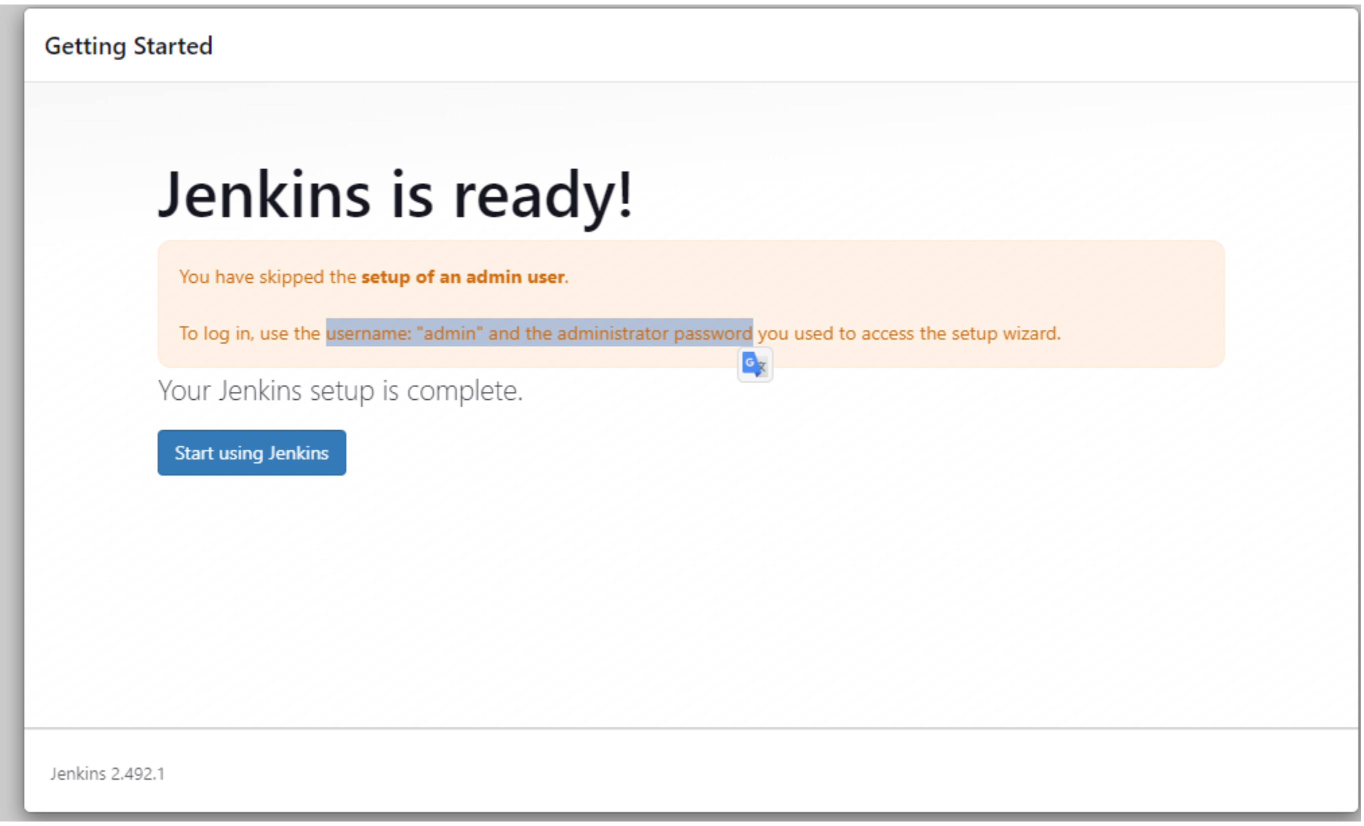
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.492.1

When the Jenkins is ready page appears, click Start using Jenkins.



Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

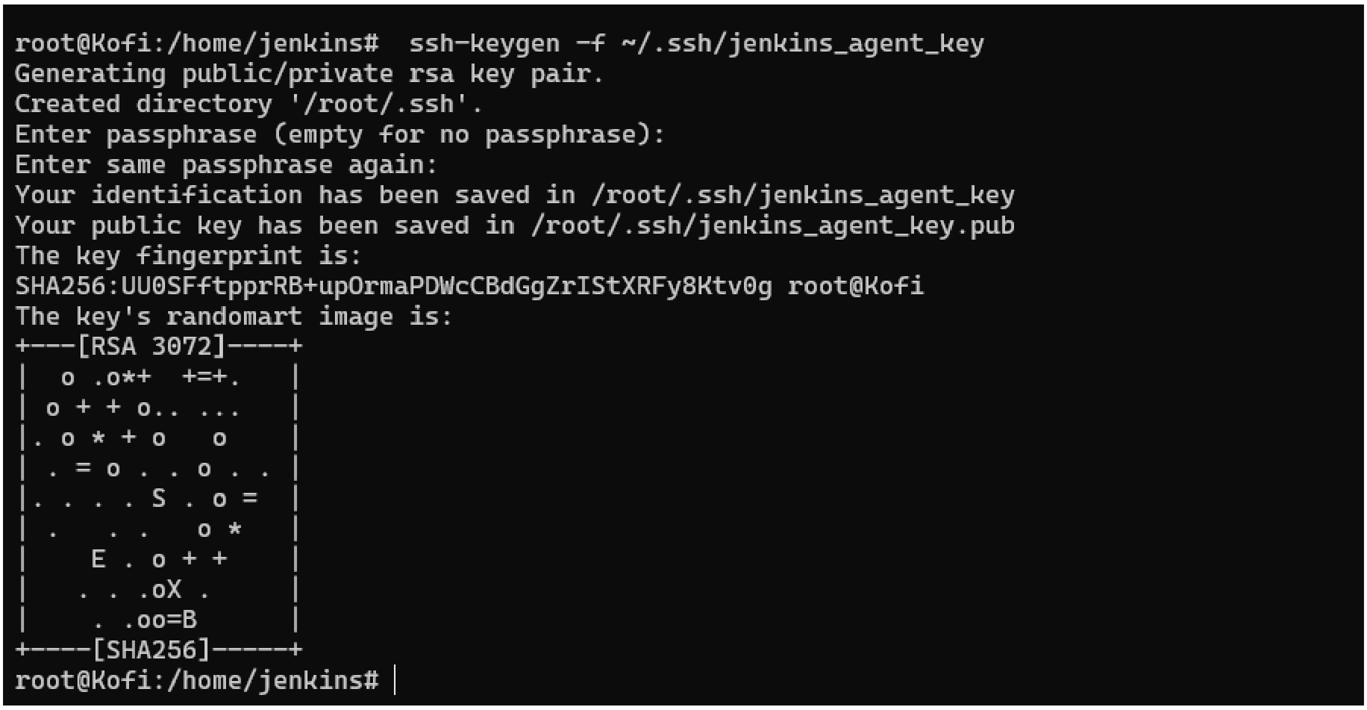
Jenkins 2.492.1

Segunda parte

Instalar un agente de Jenkins y conectarlo con el controlador de Jenkins (El agente y el controlador deben estar en la misma red).

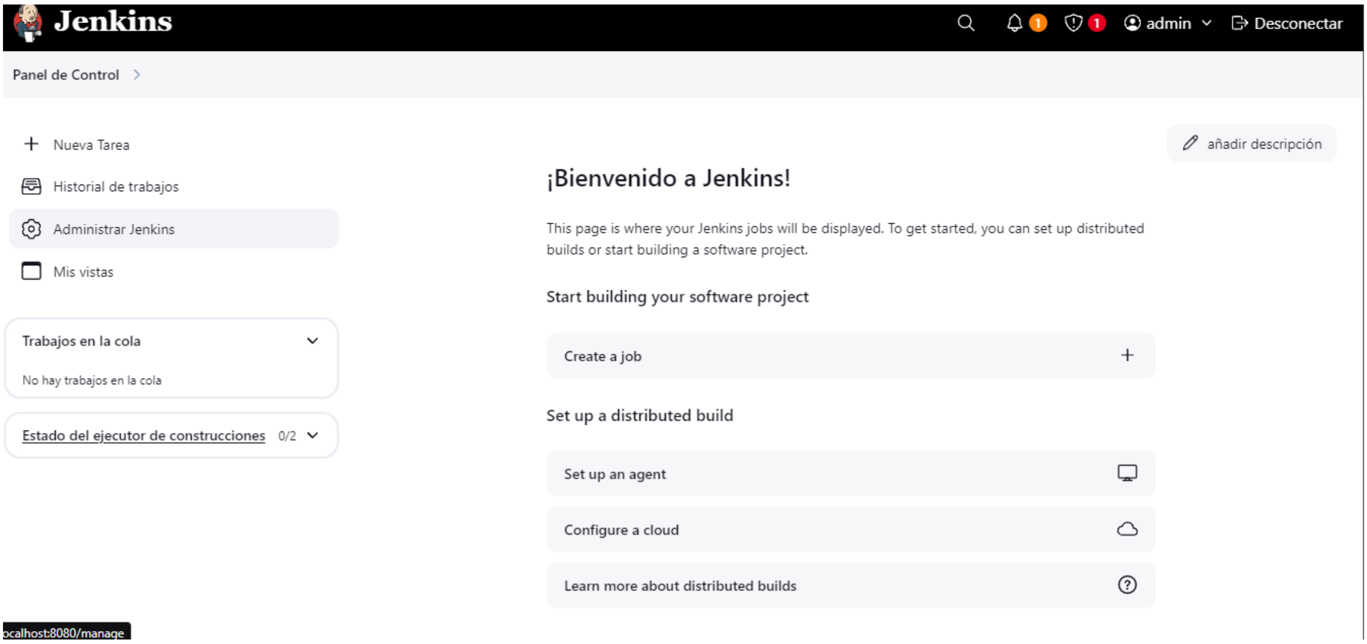
In a terminal window

```
ssh-keygen -f ~/.ssh/jenkins_agent_key
```



```
root@Kofi:/home/jenkins#  ssh-keygen -f ~/.ssh/jenkins_agent_key
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/jenkins_agent_key
Your public key has been saved in /root/.ssh/jenkins_agent_key.pub
The key fingerprint is:
SHA256:UU0SFftpprRB+upOrmaPDWcCBdGgZrIStXRFy8Ktv0g root@Kofi
The key's randomart image is:
+---[RSA 3072]----+
|  o .o*+  +=+.   |
| o + + o.. ...   |
|. o * + o   o    |
| . = o . . o . . |
|. . . . S . o =  |
| .   . .   o *   |
|    E . o + +    |
|   . . .oX .     |
|    . .oo=B      |
+----[SHA256]-----+
root@Kofi:/home/jenkins#
```

Go to your Jenkins dashboard



No hay trabajos en la cola

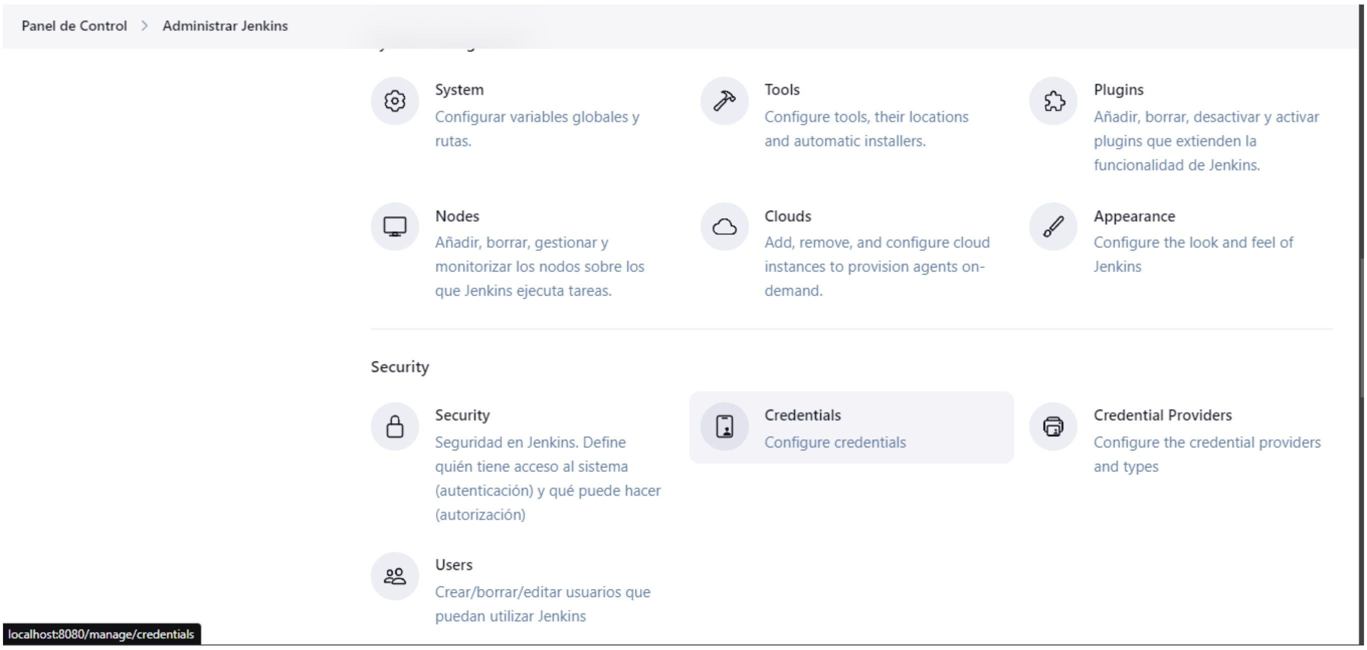
¡Bienvenido a Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Go to `Manage Jenkins` option in main menu and click on the Manage Credentials button



Panel de Control > Administrar Jenkins

System
Configurar variables globales y rutas.

Tools
Configure tools, their locations and automatic installers.

Plugins
Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.

Nodes
Añadir, borrar, gestionar y monitorizar los nodos sobre los que Jenkins ejecuta tareas.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

Security

Security
Seguridad en Jenkins. Define quién tiene acceso al sistema (autenticación) y qué puede hacer (autorización)

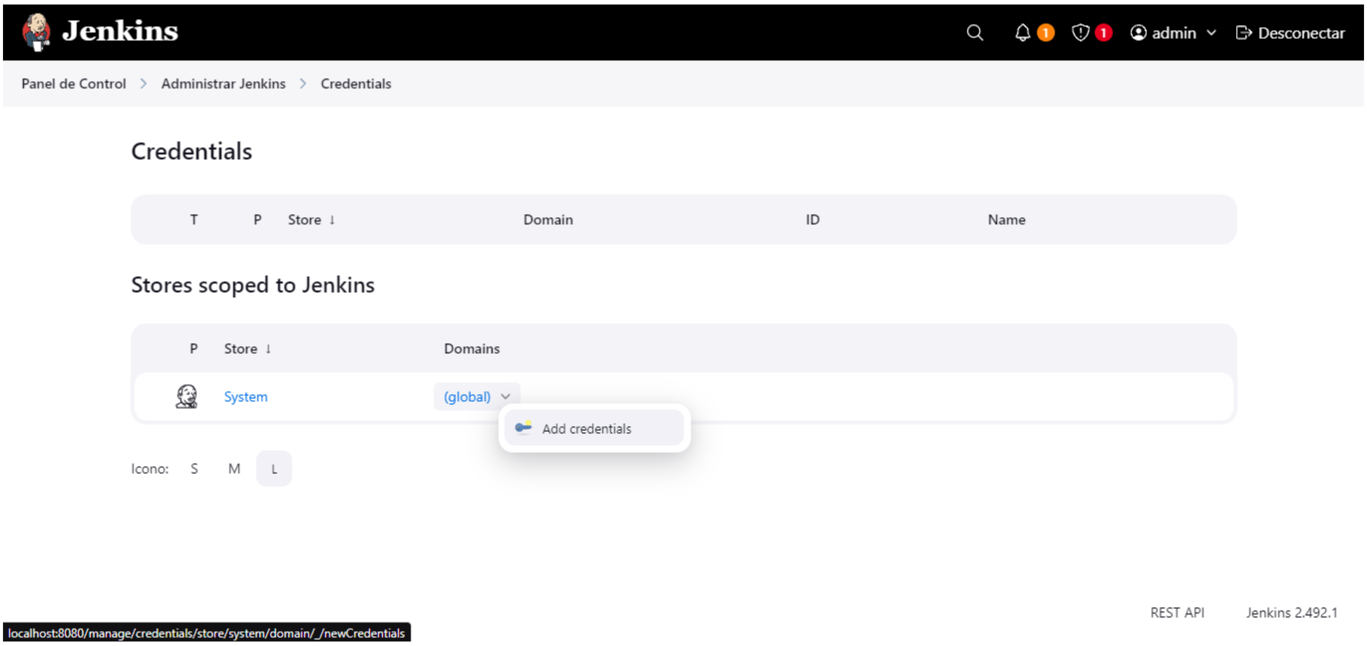
Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Crear/borrar/editar usuarios que puedan utilizar Jenkins

localhost:8080/manage/credentials

Select the drop option Add Credentials from the global item



Panel de Control > Administrar Jenkins > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

Add credentials

Icono: S M L

REST API Jenkins 2.492.1

localhost:8080/manage/credentials/store/system/domain/_/newCredentials

Fill in the form:

Kind: SSH Username with private key;

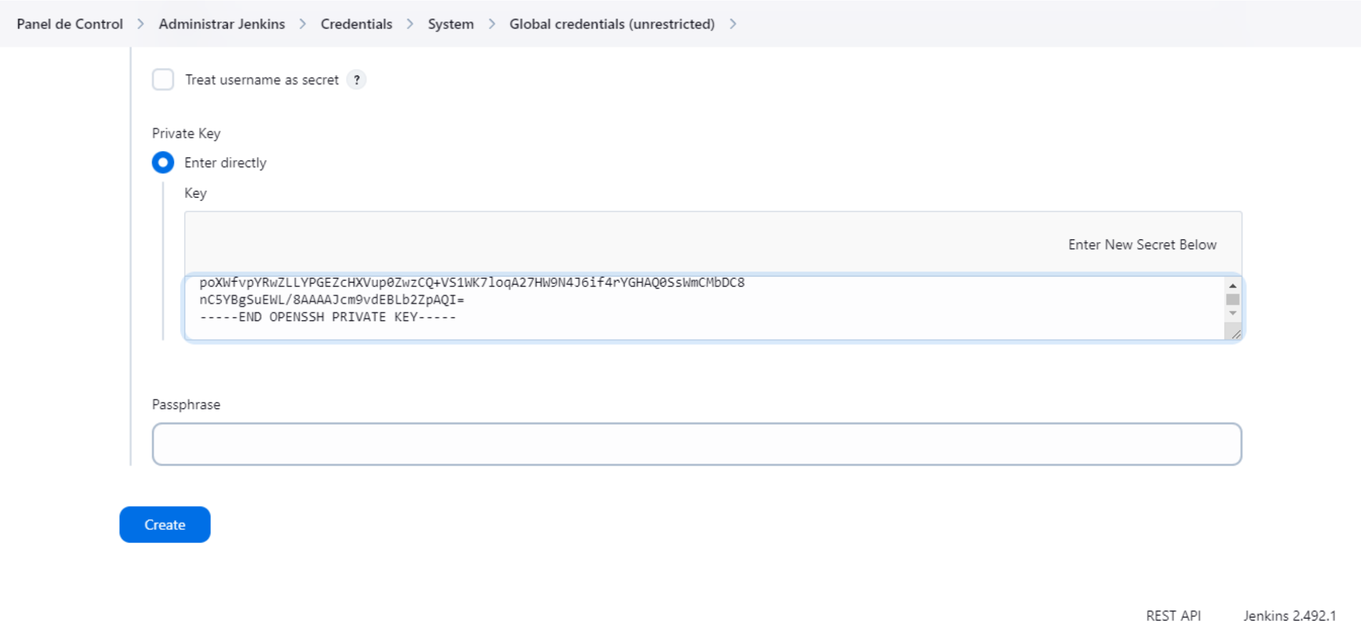
id: jenkins

description: The jenkins ssh key

username: jenkins

Private Key: select `Enter directly` and press the Add button to insert the content of your private key file at `~/.ssh/jenkins_agent_key`

Passphrase: fill your passphrase used to generate the SSH key pair (leave empty if you didn't use one at the previous step) and then press the `Create` button


Panel de Control > Administrar Jenkins > Credentials > System > Global credentials (unrestricted) >
REST API
Jenkins 2.492.1

□ Treat username as secret ?

Private Key

● Enter directly

Key

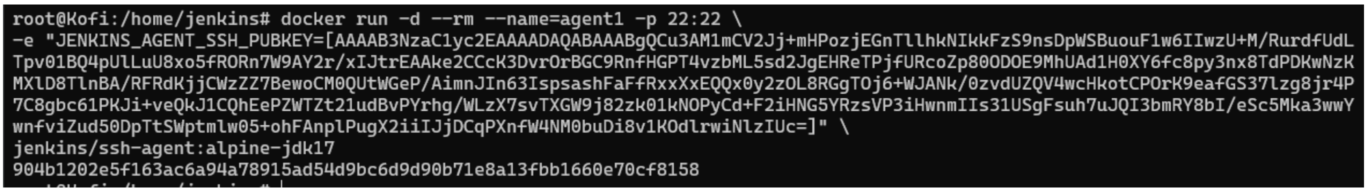
Enter New Secret Below

```
poXWfvpYRwZLLYPGEZcHXVup0ZwzCQ+VS1WK7loqA27HW9N4J6if4rYGHAQ0SsWmCMbDC8
nC5YBgSuEWL/8AAAAJcm9vdEBLb2ZpAQI=
-----END OPENSSH PRIVATE KEY-----
```

Passphrase

Here we will use the `docker-ssh-agent` image to create the agent containers.

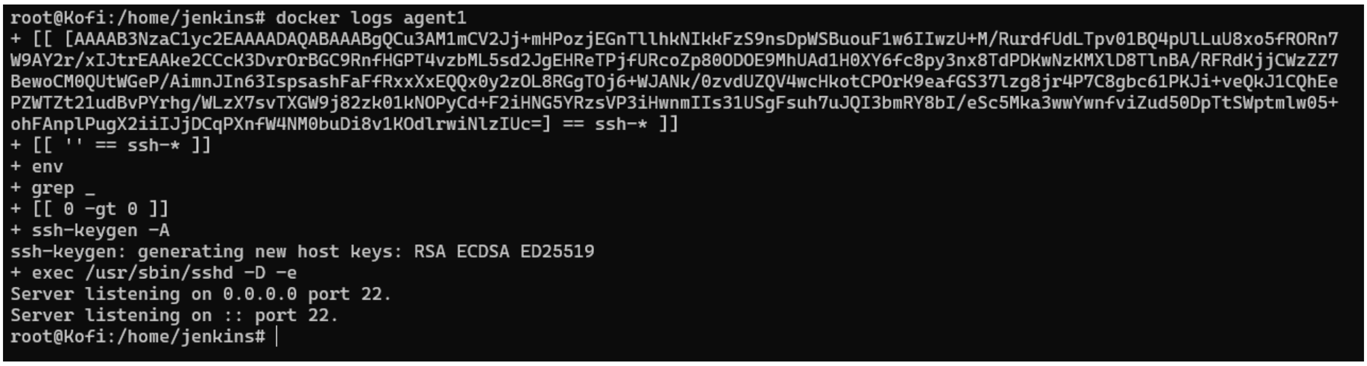
```
cat ~/.ssh/jenkins_agent_key.pub

docker run -d --rm --name=agent1 --network jenkins -p 4444:22 -e "JENKINS_AGENT_SSH_PUBKEY=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCv0lBlJ+6PMrgw5yQpsF4a/izRBOj2CQfb3jtrCNPeByVkZdHLxH5sE1y7dTQ+ZAvySy9b5ut1fC2S0HfuLlCCrVWhSB8YjRQNdIMKzx+p0FfjdR1GxJBGNqsygptuN8Kq5vfj9LX9InKt6wf//FStd0ECcga6qbC+yxekD3x1sRfWs3/sCTuzBuPLmplgH6HgwcwsdF4vlmx4bq8ejJpTjE8wmBmPsUSapKKxttX2HX+PYhSm3p4ldWzuYEu8dd98PEad9zLv9cSf05ulTOw1W86KYaSurZK8GWo1uAo8+eb5p/n/lQ+U/DBVTo3VSQlpFywbQhIsV2Ejh5RLluwxcvDFYTObZDClDS4PiyhUUsoRCPWbEpyFzsfshMOYsY3ZF5x7DY62GVUR6HbwhUBCJ1Kh1X9irYhiJmWNBlE99fUdpF0MMPyANcssCbR2yUBXrB2h440dpKTkFVrTmi7K+NSIolC7m9C8jQBJRwuD+/Iibq2VQL3p3vQ5THrvkYk=" jenkins/ssh-agent:alpine-jdk17
```



```
root@Kofi:/home/jenkins# docker run -d --rm --name=agent1 -p 22:22 \
-e "JENKINS_AGENT_SSH_PUBKEY=[AAAAB3NzaC1yc2EAAAADAQABAAABgQCu3AM1mCV2Jj+mHPozjEGnTllhkNIkkFzS9nsDpWSBuouF1w6IIwzU+M/RurdfUdL
Tpv01BQ4pUlLuU8xo5fRORn7W9AY2r/xIJtrEAAke2CCcK3DvrOrBGC9RnfHGPT4vzbML5sd2JgEHReTPjfURcoZp80ODOE9MhUAd1H0XY6fc8py3nx8TdPDKwNzK
MXlD8TlnBA/RFRdKjjCWzZZ7BewoCM0QUtWGeP/AimnJIn63IspsashFaFfRxxXxEQQx0y2zOL8RGgTOj6+WJANk/0zvdUZQV4wcHkotCPOrK9eafGS37lzg8jr4P
7C8gbc61PKJi+veQkJ1CQhEePZWTZt21udBvPYrhg/WLzX7svTXGW9j82zk01kNOPyCd+F2iHNG5YRzsVP3iHwnmIIs31USgFsuh7uJQI3bmRY8bI/eSc5Mka3wwY
wnfviZud50DpTtSWptmlw05+ohFAnplPugX2iiIJjDCqPXnfW4NM0buDi8v1KOdlrwiNlzIUc=]" \
jenkins/ssh-agent:alpine-jdk17
904b1202e5f163ac6a94a78915ad54d9bc6d9d90b71e8a13fbb1660e70cf8158
```

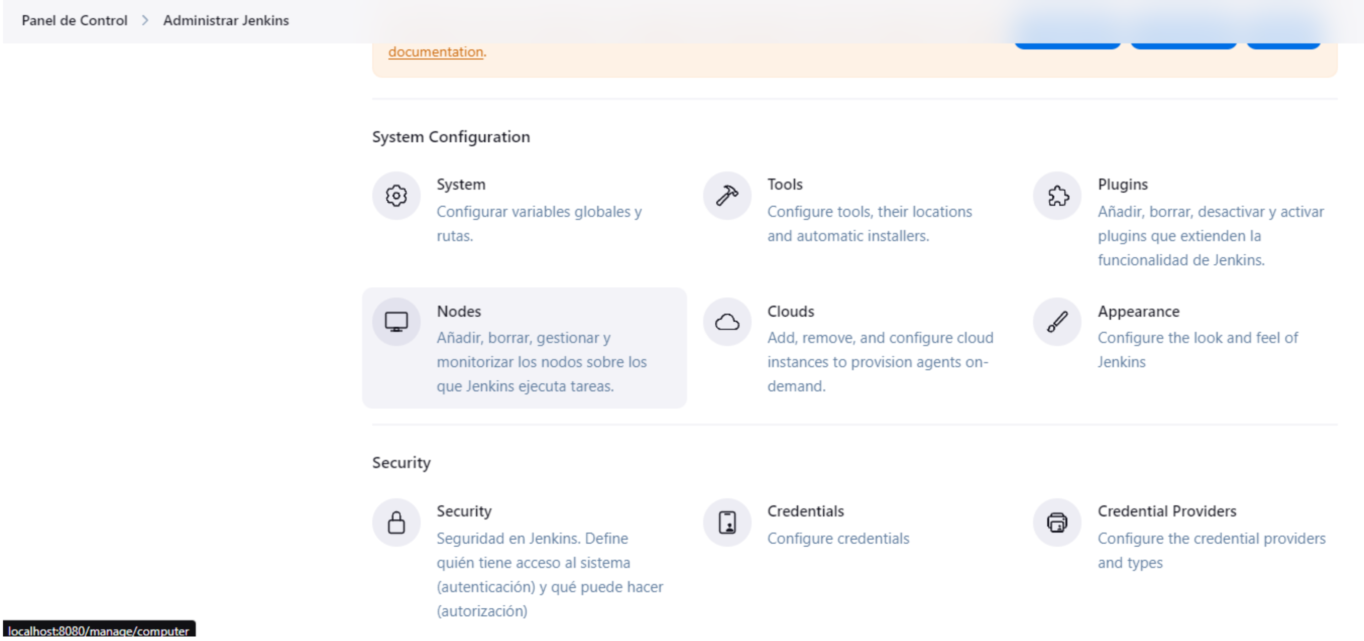
Verify the agent

```
docker logs agent1
```



```
root@Kofi:/home/jenkins# docker logs agent1
+ [[ [AAAAB3NzaC1yc2EAAAADAQABAAABgQCu3AM1mCV2Jj+mHPozjEGnTllhkNIkkFzS9nsDpWSBuouF1w6IIwzU+M/RurdfUdLTpv01BQ4pUlLuU8xo5fRORn7
W9AY2r/xIJtrEAAke2CCcK3DvrOrBGC9RnfHGPT4vzbML5sd2JgEHReTPjfURcoZp80ODOE9MhUAd1H0XY6fc8py3nx8TdPDKwNzKMXlD8TlnBA/RFRdKjjCWzZZ7
BewoCM0QUtWGeP/AimnJIn63IspsashFaFfRxxXxEQQx0y2zOL8RGgTOj6+WJANk/0zvdUZQV4wcHkotCPOrK9eafGS37lzg8jr4P7C8gbc61PKJi+veQkJ1CQhEe
PZWTZt21udBvPYrhg/WLzX7svTXGW9j82zk01kNOPyCd+F2iHNG5YRzsVP3iHwnmIIs31USgFsuh7uJQI3bmRY8bI/eSc5Mka3wwYwnfviZud50DpTtSWptmlw05+
ohFAnplPugX2iiIJjDCqPXnfW4NM0buDi8v1KOdlrwiNlzIUc=] == ssh-* ]]
+ [[ '' == ssh-* ]]
+ env
+ grep _
+ [[ 0 -gt 0 ]]
+ ssh-keygen -A
ssh-keygen: generating new host keys: RSA ECDSA ED25519
+ exec /usr/sbin/sshd -D -e
Server listening on 0.0.0.0 port 22.
Server listening on :: port 22.
root@Kofi:/home/jenkins# |
```

Ingresar a la configuración de Jenkins, Nodos



documentation.

System Configuration

System
Configurar variables globales y rutas.

Tools
Configure tools, their locations and automatic installers.

Plugins
Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.

Nodes
Añadir, borrar, gestionar y monitorizar los nodos sobre los que Jenkins ejecuta tareas.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

Security

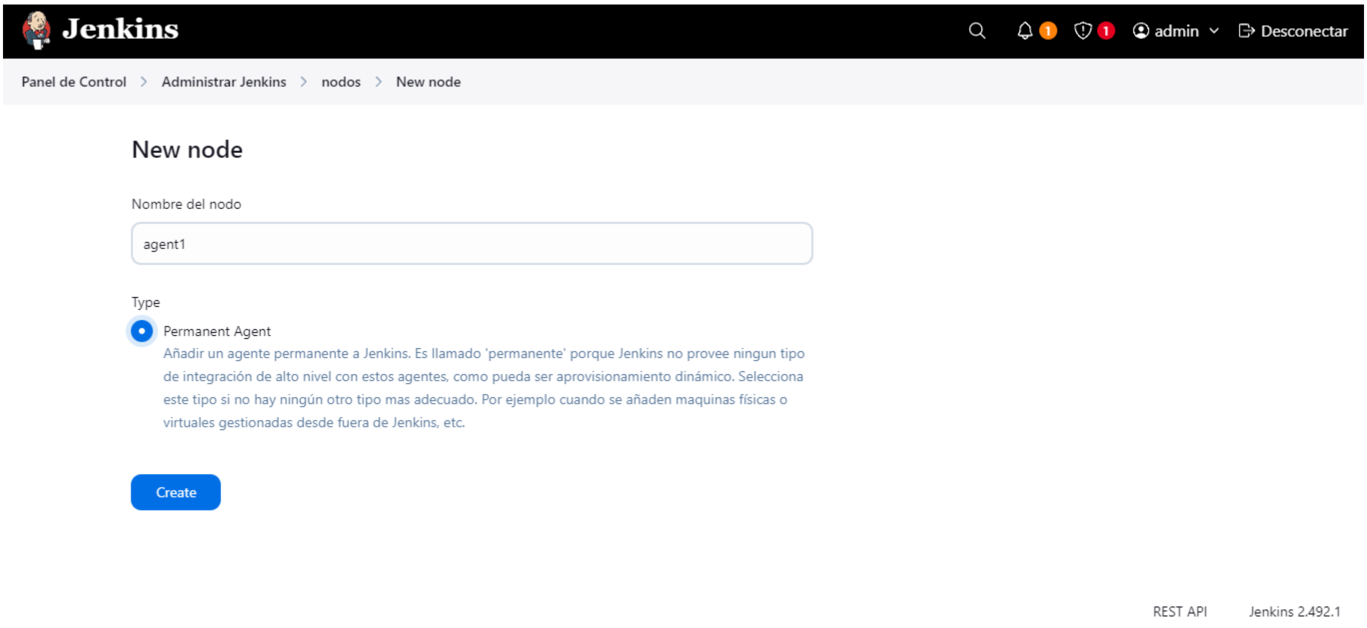
Security
Seguridad en Jenkins. Define quién tiene acceso al sistema (autenticación) y qué puede hacer (autorización).

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

localhost:8080/manage/computer

Crear un nuevo nodo que se lance al conectarlo al controlador.



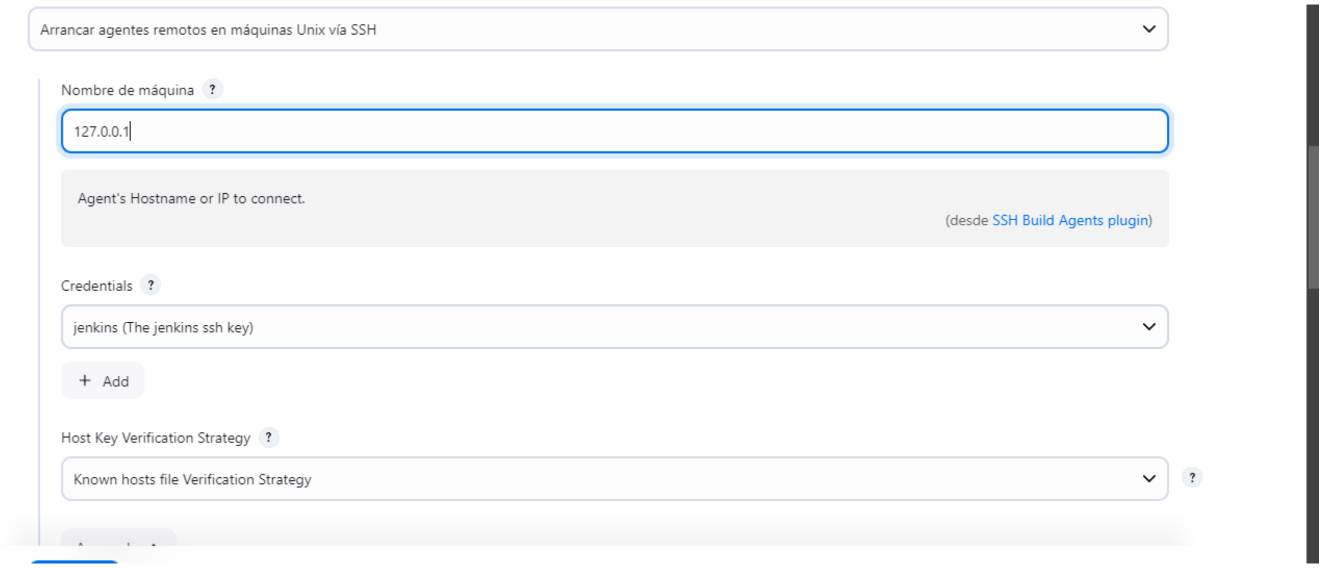
New node

Nombre del nodo

agent1

Type

● Permanent Agent
Añadir un agente permanente a Jenkins. Es llamado 'permanente' porque Jenkins no provee ningun tipo de integración de alto nivel con estos agentes, como pueda ser aprovisionamiento dinámico. Selecciona este tipo si no hay ningún otro tipo mas adecuado. Por ejemplo cuando se añaden maquinas físicas o virtuales gestionadas desde fuera de Jenkins, etc.



Arrancar agentes remotos en máquinas Unix vía SSH

Nombre de máquina ?

127.0.0.1

Agent's Hostname or IP to connect.

(desde SSH Build Agents plugin)

Credentials ?

jenkins (The jenkins ssh key)

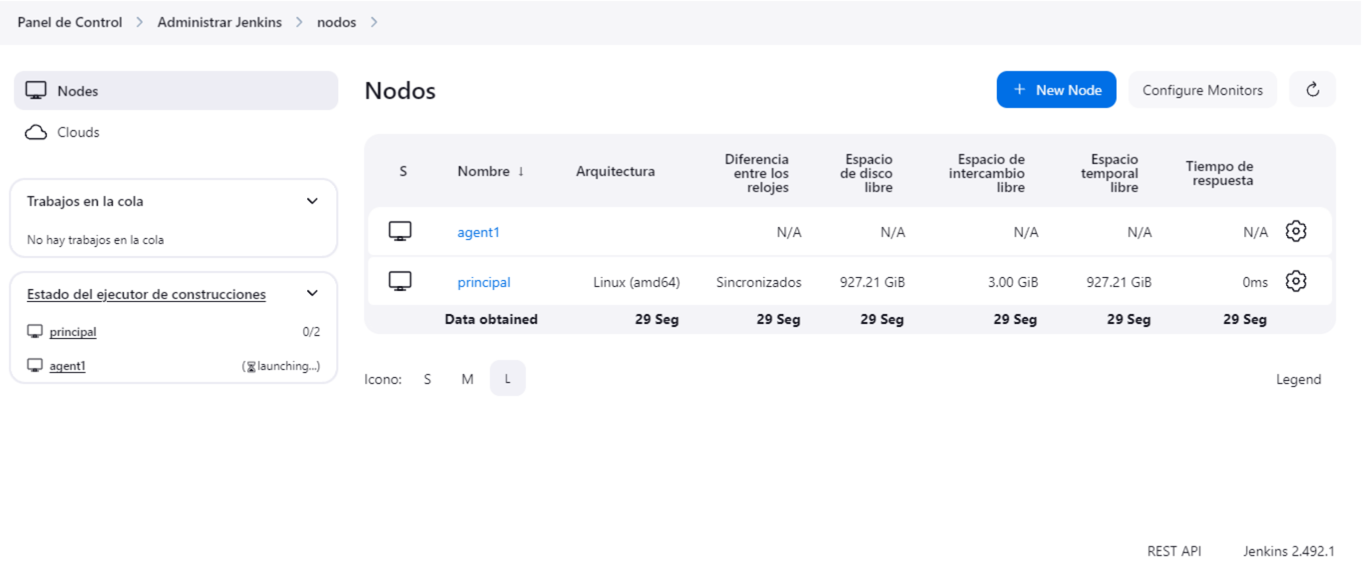
Host Key Verification Strategy ?

Known hosts file Verification Strategy

Correr el comando generado por Jenkins para iniciar y conectar el agente.

```
mkdir agent
```

Seleccionar el nodo



Nodos

Trabajos en la cola

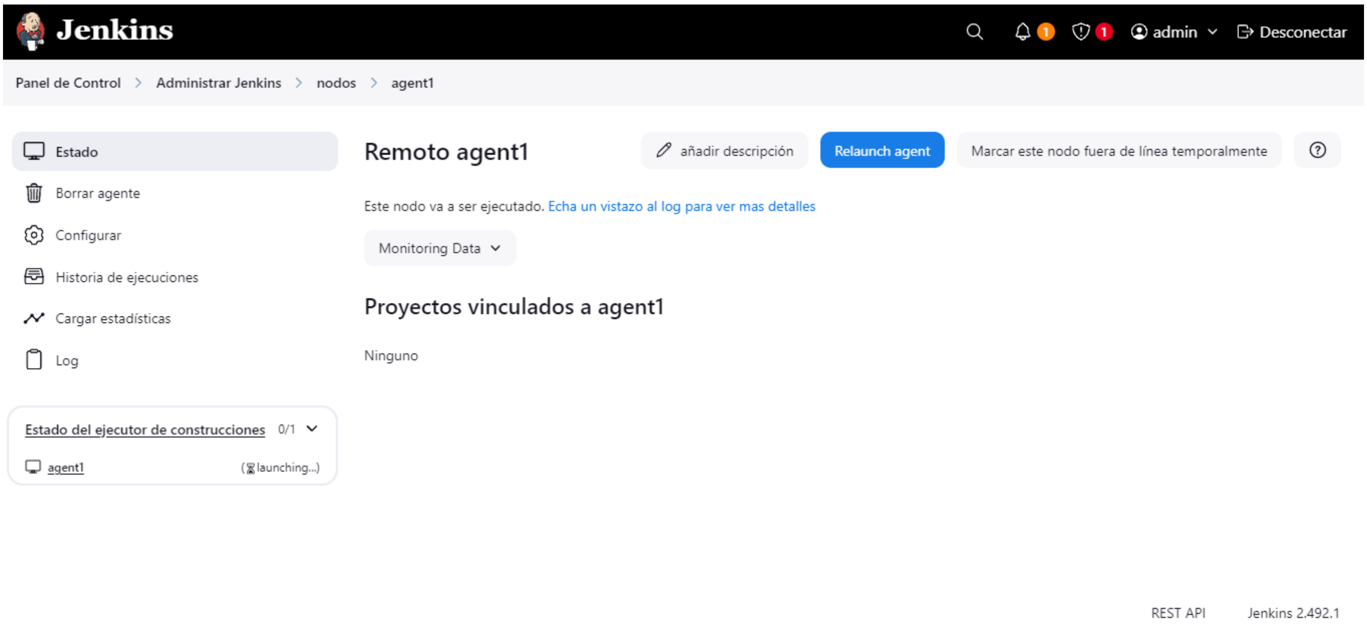
No hay trabajos en la cola

Estado del ejecutor de construcciones

principal	0/2
agent1	(launching...)

S	Nombre ↓	Arquitectura	Diferencia entre los relojes	Espacio de disco libre	Espacio de intercambio libre	Espacio temporal libre	Tiempo de respuesta
	agent1		N/A	N/A	N/A	N/A	N/A
	principal	Linux (amd64)	Sincronizados	927.21 GiB	3.00 GiB	927.21 GiB	0ms
	Data obtained	29 Seg	29 Seg	29 Seg	29 Seg	29 Seg	29 Seg

Relanzar agente



Remoto agent1

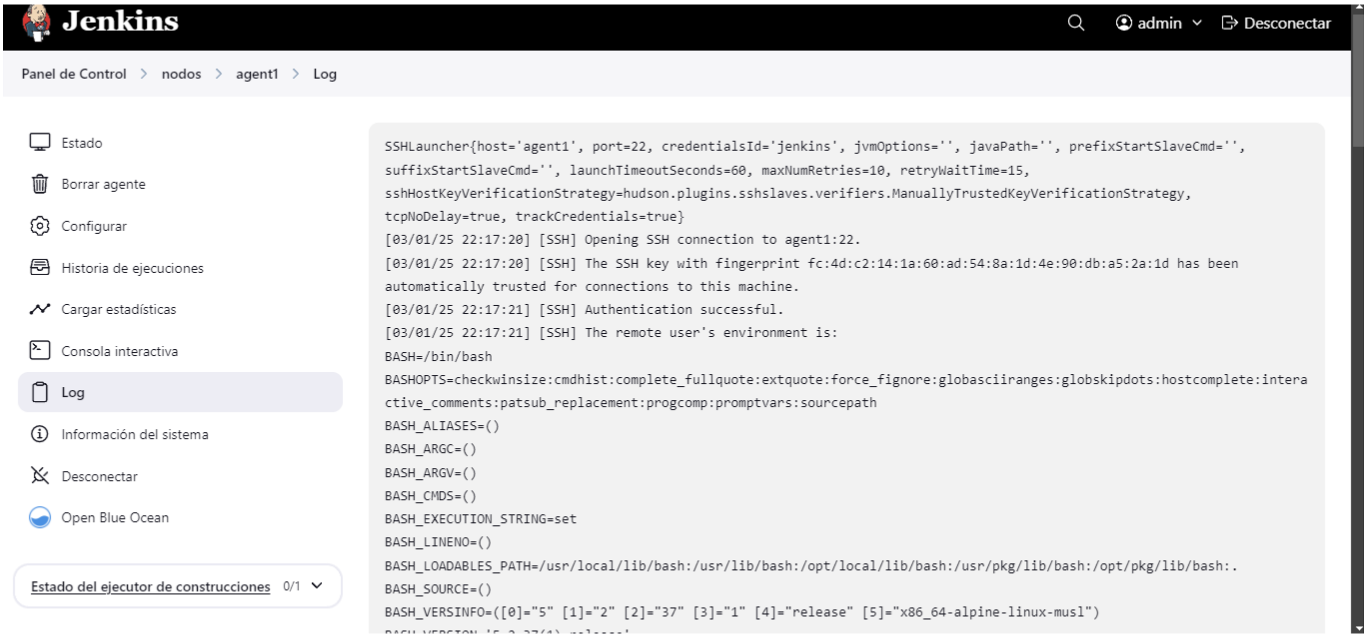
Este nodo va a ser ejecutado. Echa un vistazo al log para ver mas detalles

Proyectos vinculados a agent1

Ninguno

Si no funciona, podemos intentar pasar la clave directamente a jenkins

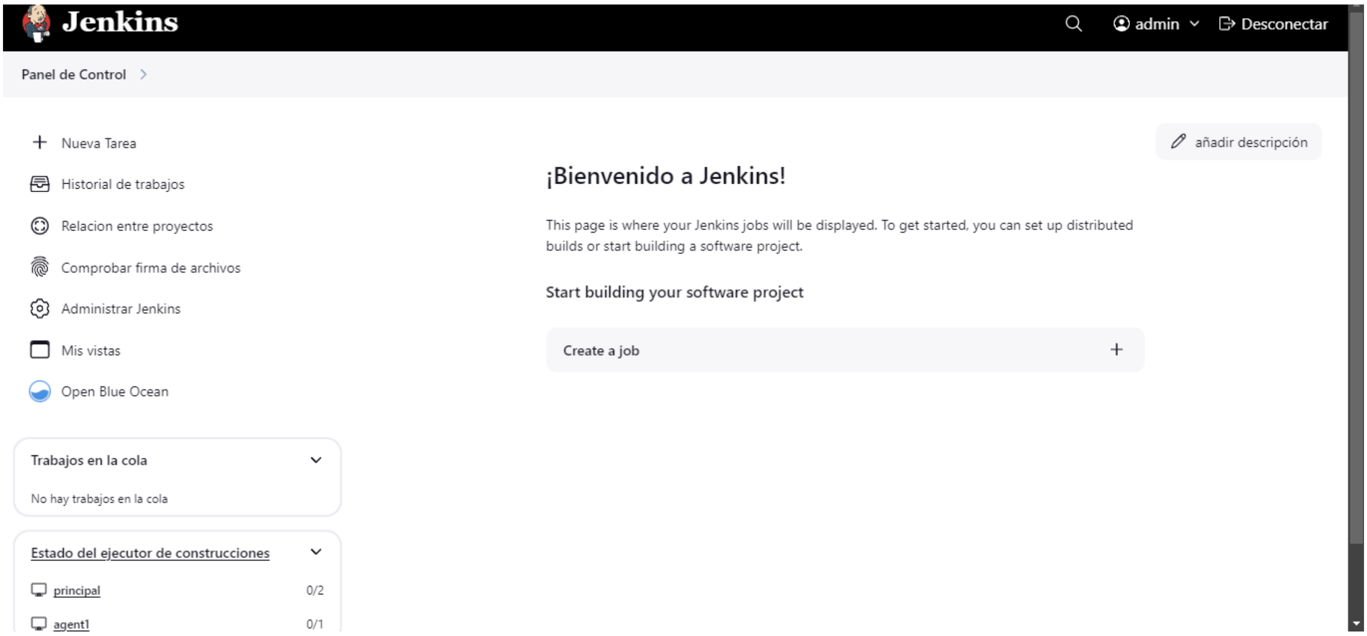
```
docker exec -it jenkins-blueocean sh -c "echo '-----BEGIN OPENSSH PRIVATE KEY-----
...' > /var/jenkins_home/.ssh/id_rsa"
docker exec -it jenkins-blueocean sh -c "chmod 600 /var/jenkins_home/.ssh/id_rsa"
```



```
SSHLauncher{host='agent1', port=22, credentialsId='jenkins', jvmOptions='', javaPath='', prefixStartSlaveCmd='',
suffixStartSlaveCmd='', launchTimeoutSeconds=60, maxNumRetries=10, retryWaitTime=15,
sshHostKeyVerificationStrategy=hudson.plugins.sshslaves.verifiers.ManuallyTrustedKeyVerificationStrategy,
tcpNoDelay=true, trackCredentials=true}
[03/01/25 22:17:20] [SSH] Opening SSH connection to agent1:22.
[03/01/25 22:17:20] [SSH] The SSH key with fingerprint fc:4d:c2:14:1a:60:ad:54:8a:1d:4e:90:db:a5:2a:1d has been
automatically trusted for connections to this machine.
[03/01/25 22:17:21] [SSH] Authentication successful.
[03/01/25 22:17:21] [SSH] The remote user's environment is:
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:extquote:force_fignore:globasciiranges:globskipdots:hostcomplete:intera
ctive_comments:patsub_replacement:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()
BASH_ARGV=()
BASH_CMDS=()
BASH_EXECUTION_STRING=set
BASH_LINENO=()
BASH_LOADABLES_PATH=/usr/local/lib/bash:/usr/lib/bash:/opt/local/lib/bash:/usr/pkg/lib/bash:/opt/pkg/lib/bash:.
BASH_SOURCE=()
BASH_VERSINFO=([0]="5" [1]="2" [2]="37" [3]="1" [4]="release" [5]="x86_64-alpine-linux-musl")
BASH_VERSION='5.2.37(1)-release'
```

Tercera parte

Crear un ítem de tipo pipeline en Jenkins y configurar el script usando una aplicación sencilla (e.j., https://github.com/jenkins-docs/simple-java-maven-app). Ejecutar el pipeline y revisar la consola.



¡Bienvenido a Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job +

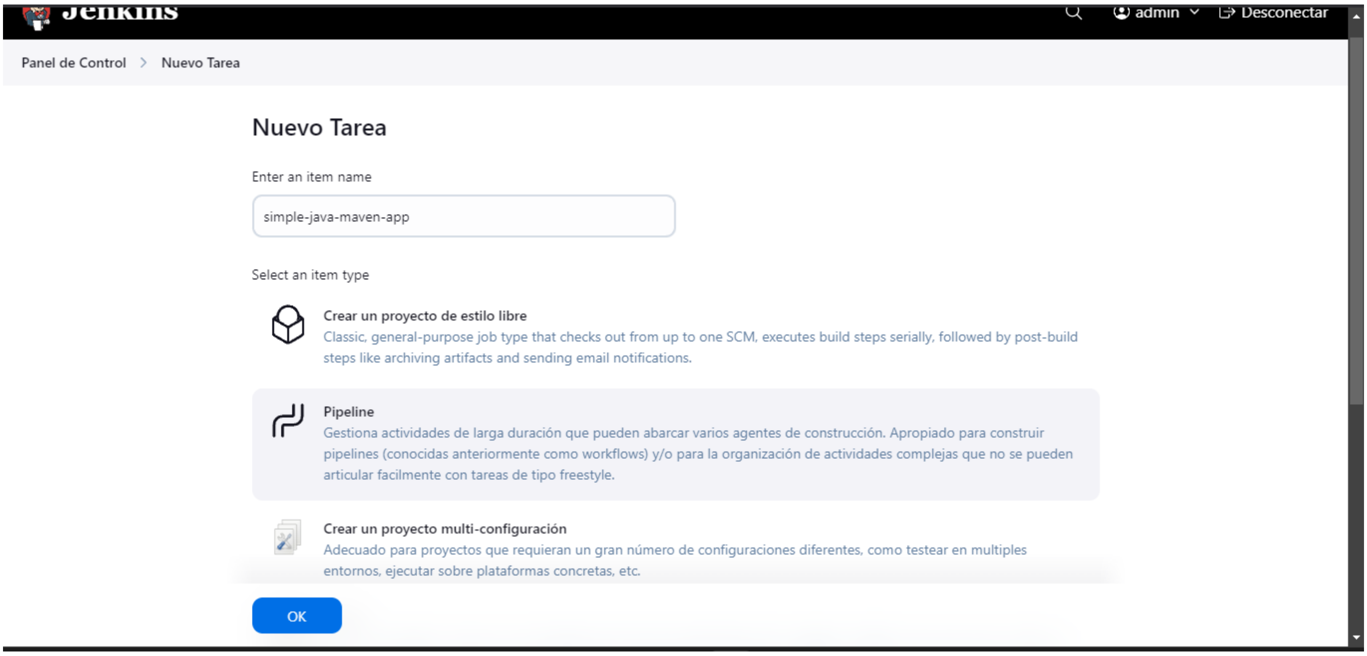
Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

principal	0/2
agent1	0/1

Crear tarea



Nuevo Tarea

Enter an item name

simple-java-maven-app

Select an item type

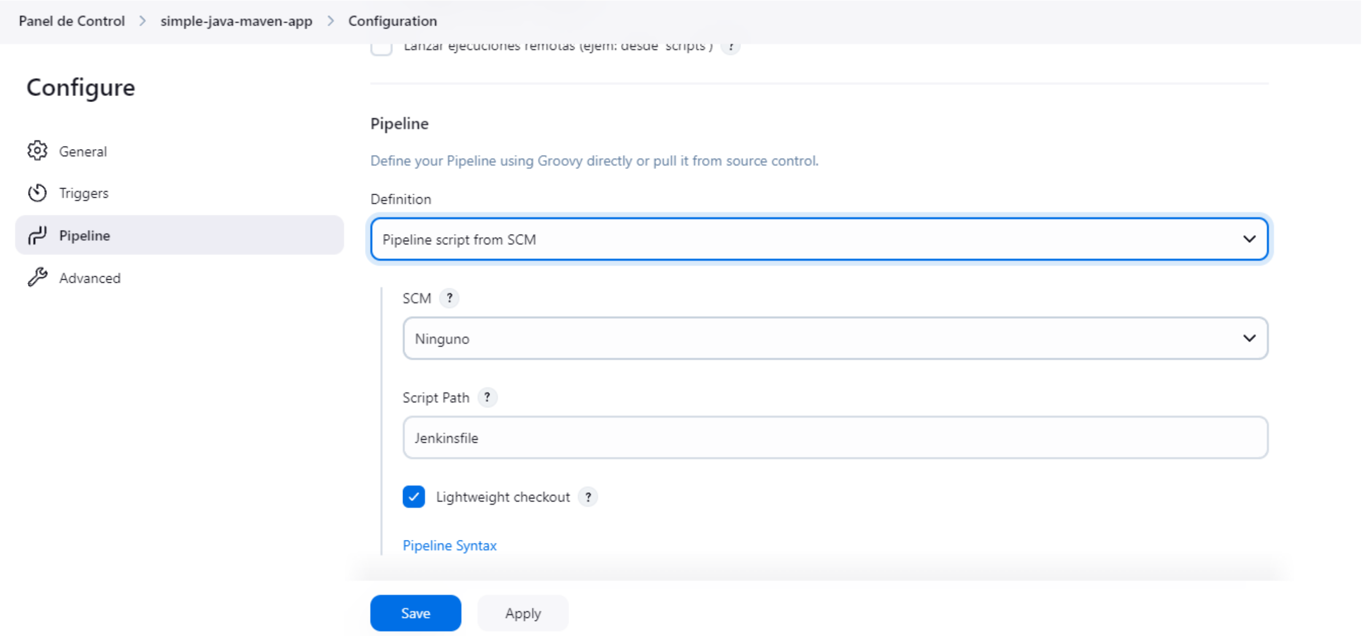
Crear un proyecto de estilo libre
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular facilmente con tareas de tipo freestyle.

Crear un proyecto multi-configuración
Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en multiples entornos, ejecutar sobre plataformas concretas, etc.

En la página de configuración del pipeline, desplázate hacia abajo hasta la sección "Pipeline".

En la opción "Definition", selecciona "Pipeline script from SCM".



Panel de Control > simple-java-maven-app > Configuration

Configure

□ Lanzar ejecuciones remotas (ejem: desde 'scripts')

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Ninguno

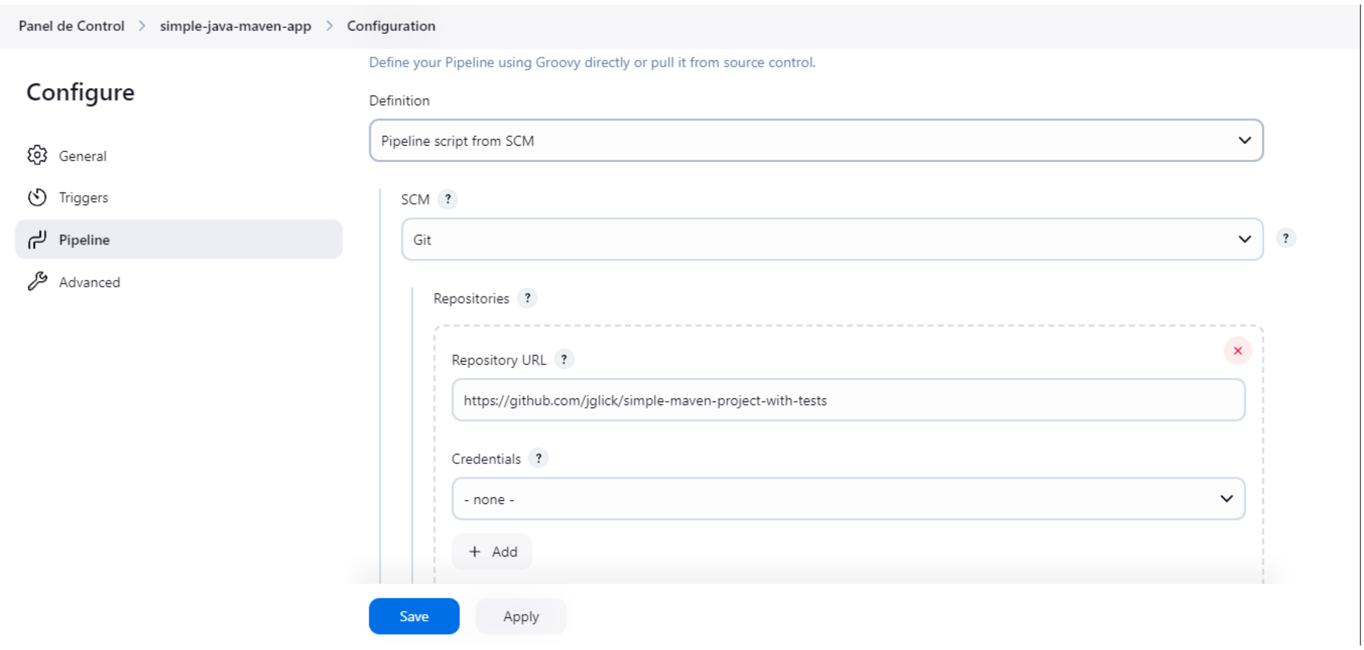
Script Path: Jenkinsfile

☑ Lightweight checkout

En "SCM", selecciona "Git".

En "Repository URL", ingresa la URL del repositorio de ejemplo:

Cambiar el path del Jenkinsfile si no esta en la raiz



Panel de Control > simple-java-maven-app > Configuration

Configure

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Git

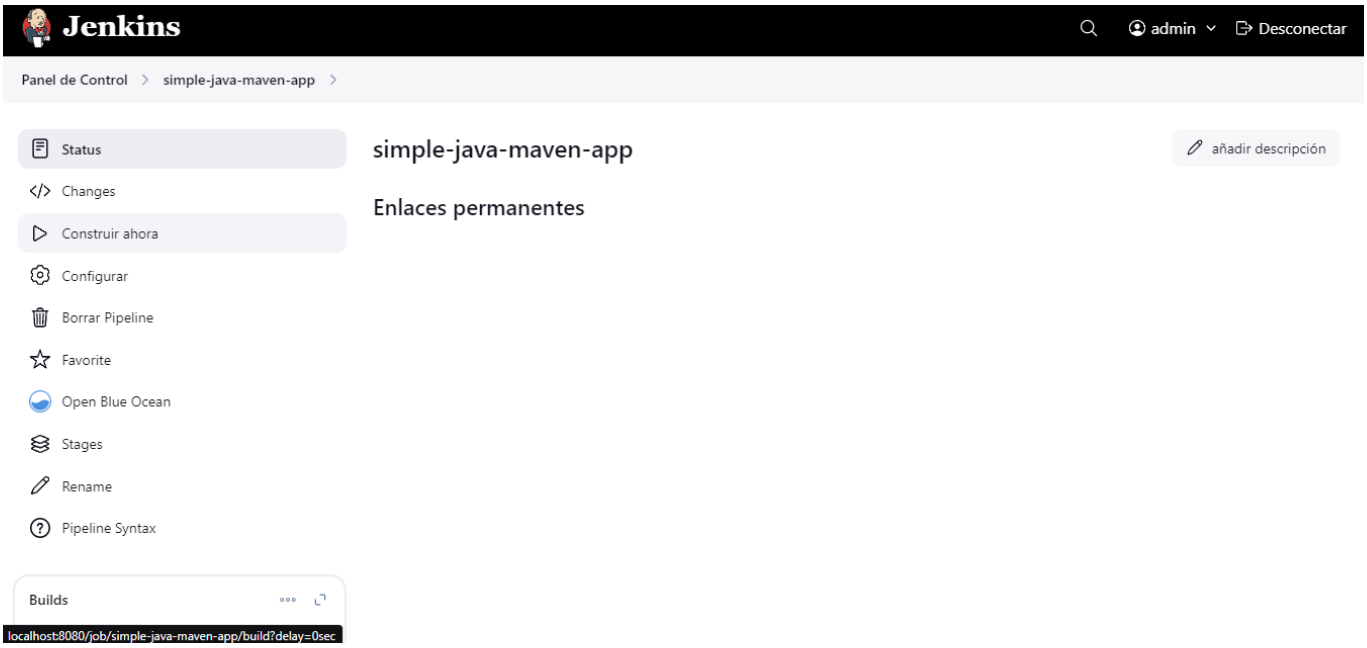
Repositories

Repository URL: https://github.com/jglick/simple-maven-project-with-tests

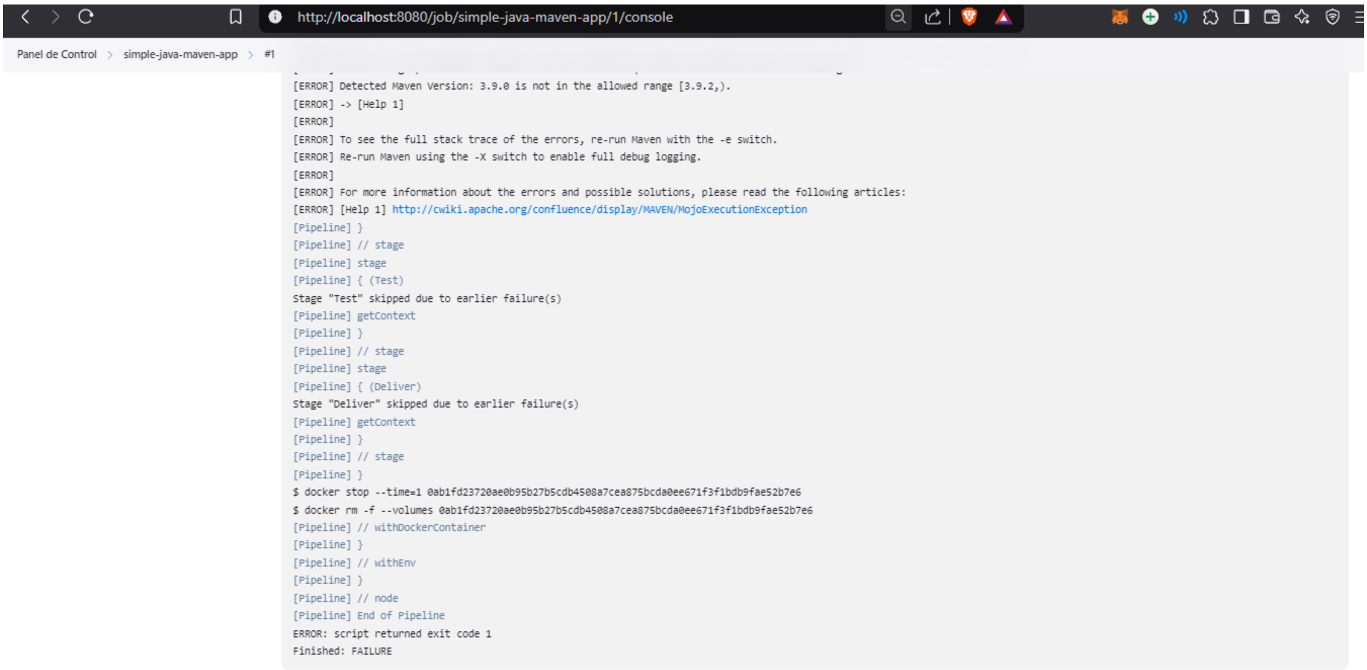
Credentials: - none -

Una vez guardado, serás redirigido a la página del pipeline.

Haz clic en "Construir ahora" (o "Build Now") para ejecutar el pipeline.



Mirar la consola


http://localhost:8080/job/simple-java-maven-app/1/console
Panel de Control > simple-java-maven-app > #1

```
[ERROR] Detected Maven Version: 3.9.0 is not in the allowed range [3.9.2,).
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
Stage "Test" skipped due to earlier failure(s)
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deliver)
Stage "Deliver" skipped due to earlier failure(s)
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
$ docker stop --time=1 0ab1fd23720ae0b95b27b5cdb4508a7cea875bcda0ee671f3f1bdb9fae52b7e6
$ docker rm -f --volumes 0ab1fd23720ae0b95b27b5cdb4508a7cea875bcda0ee671f3f1bdb9fae52b7e6
[Pipeline] // withDockerContainer
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

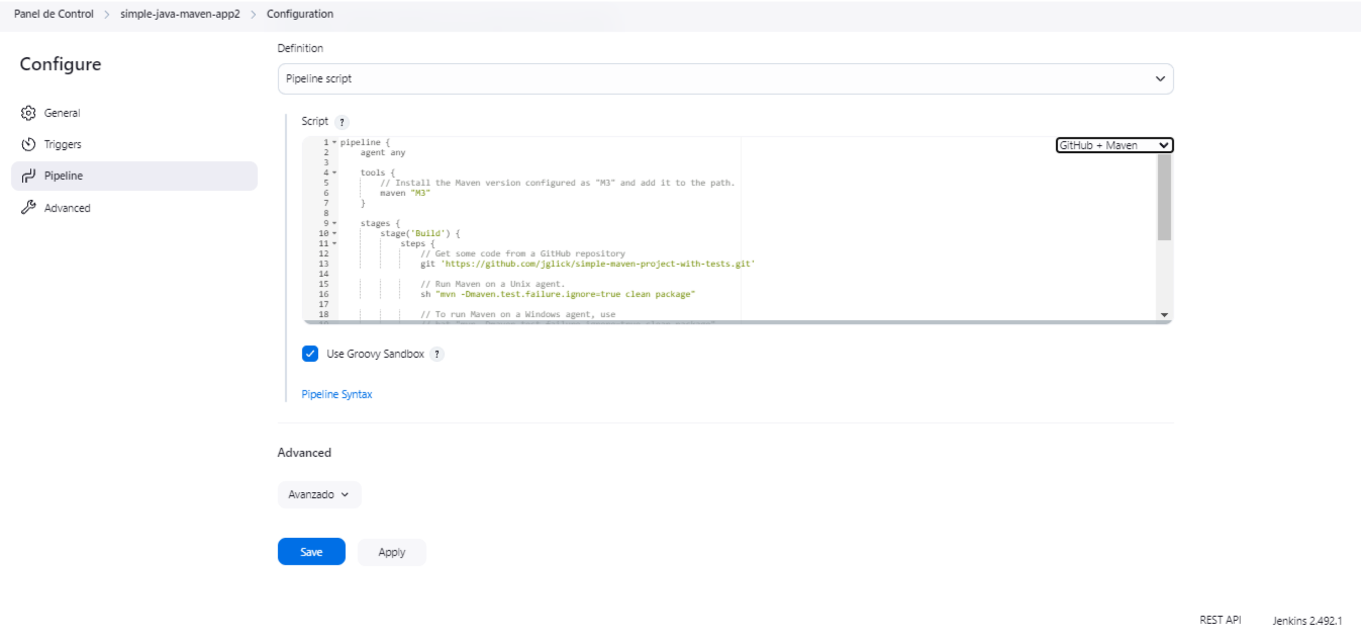
Da error porque el maven que esta jalando el jenkins no es el que pide en el pom.xml

Crear un ítem de tipo pipeline en Jenkins y configurar el script usando Github+Maven. Ejecutar el pipeline y revise la consola.

En la página de configuración del pipeline, desplázate hacia abajo hasta la sección "Pipeline".

En la opción "Definition", selecciona "Pipeline script".

Seleccionar Github+Maven

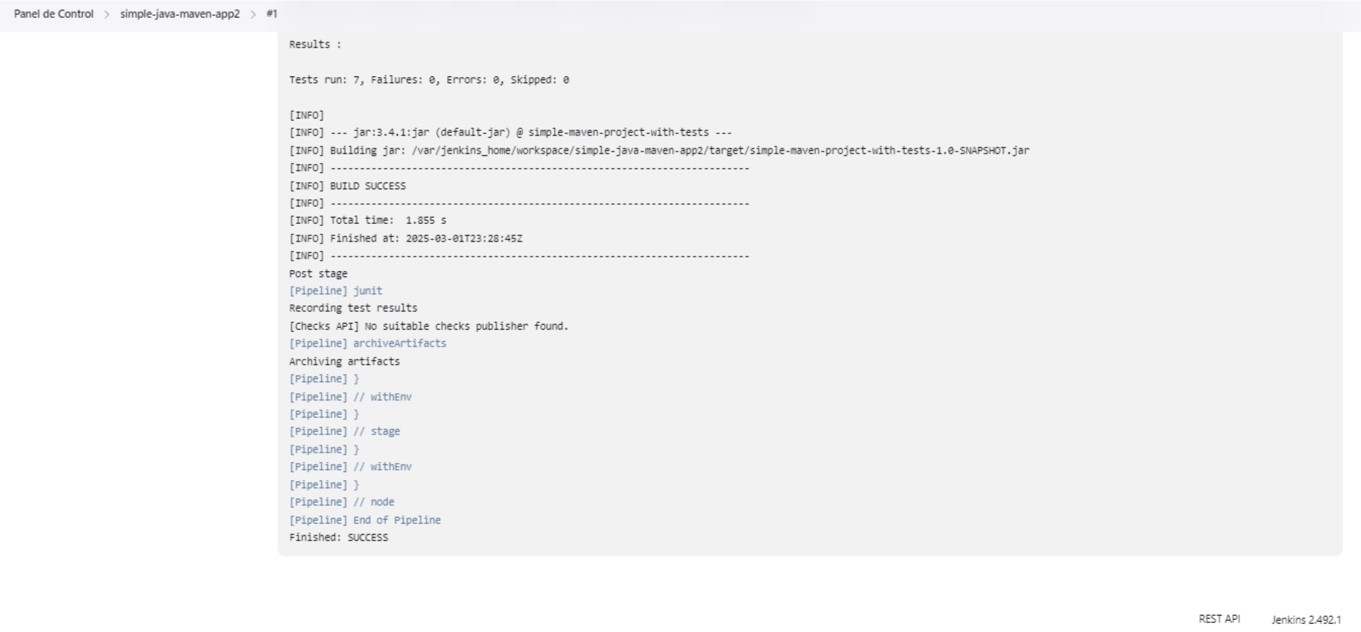


```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"

                // To run Maven on a Windows agent, use
```

Ejecutar el pipeline y revise la consola.

```
Started by user admin

Obtained jenkins/Jenkinsfile from git https://github.com/jenkins-docs/simple-java-maven-app
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins
 in /var/jenkins_home/workspace/simple-java-maven-app
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/simple-java-maven-app/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/jenkins-docs/simple-java-maven-app # timeout=10
Fetching upstream changes from https://github.com/jenkins-docs/simple-java-maven-app
 > git --version # timeout=10
 > git --version # 'git version 2.39.5'
 > git fetch --tags --force --progress -- https://github.com/jenkins-docs/simple-java-maven-app +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 05f17fbceabf683500da604bba18c6b8d7e3f109
(refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 05f17fbceabf683500da604bba18c6b8d7e3f109 # timeout=10
Commit message: "Merge pull request #1090 from jenkins-docs/dependabot/maven/org.apache.maven.plugins-maven-compiler-plugin-3.14.0"
```

```
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker inspect -f . maven:3.9.0
.
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withDockerContainer
Jenkins seems to be running inside container
536f8b7b9379fa4822662cd3365920932964ae59eebb82511539a529eb38485a
but /var/jenkins_home/workspace/simple-java-maven-app could not be found among []
but /var/jenkins_home/workspace/simple-java-maven-app@tmp could not be found among
[]
$ docker run -t -d -u 1000:1000 -v /root/.m2:/root/.m2 -w
/var/jenkins_home/workspace/simple-java-maven-app -v
/var/jenkins_home/workspace/simple-java-maven-
app:/var/jenkins_home/workspace/simple-java-maven-app:rw,z -v
/var/jenkins_home/workspace/simple-java-maven-
app@tmp:/var/jenkins_home/workspace/simple-java-maven-app@tmp:rw,z -e ******** -e
******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -
e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ********
-e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -e
******** -e ******** -e ******** -e ******** -e ******** -e ******** -e ******** -
e ******** maven:3.9.0 cat
$ docker top 0ab1fd23720ae0b95b27b5cdb4508a7cea875bcda0ee671f3f1bdb9fae52b7e6 -eo
pid,comm
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] sh
+ mvn -B -DskipTests clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.mycompany.app:my-app >----------------------
[INFO] Building my-app 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ my-app ---
[INFO]
[INFO] --- enforcer:3.5.0:enforce (enforce-maven) @ my-app ---
[INFO] Rule 1: org.apache.maven.enforcer.rules.version.RequireJavaVersion passed
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.548 s
[INFO] Finished at: 2025-03-01T23:25:20Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:3.5.0:enforce (enforce-maven) on project my-app:
[ERROR] Rule 0: org.apache.maven.enforcer.rules.version.RequireMavenVersion failed with message:
[ERROR] Detected Maven Version: 3.9.0 is not in the allowed range [3.9.2,).
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
Stage "Test" skipped due to earlier failure(s)
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deliver)
Stage "Deliver" skipped due to earlier failure(s)
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
$ docker stop --time=1 0ab1fd23720ae0b95b27b5cdb4508a7cea875bcda0ee671f3f1bdb9fae52b7e6
$ docker rm -f --volumes 0ab1fd23720ae0b95b27b5cdb4508a7cea875bcda0ee671f3f1bdb9fae52b7e6
[Pipeline] // withDockerContainer
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

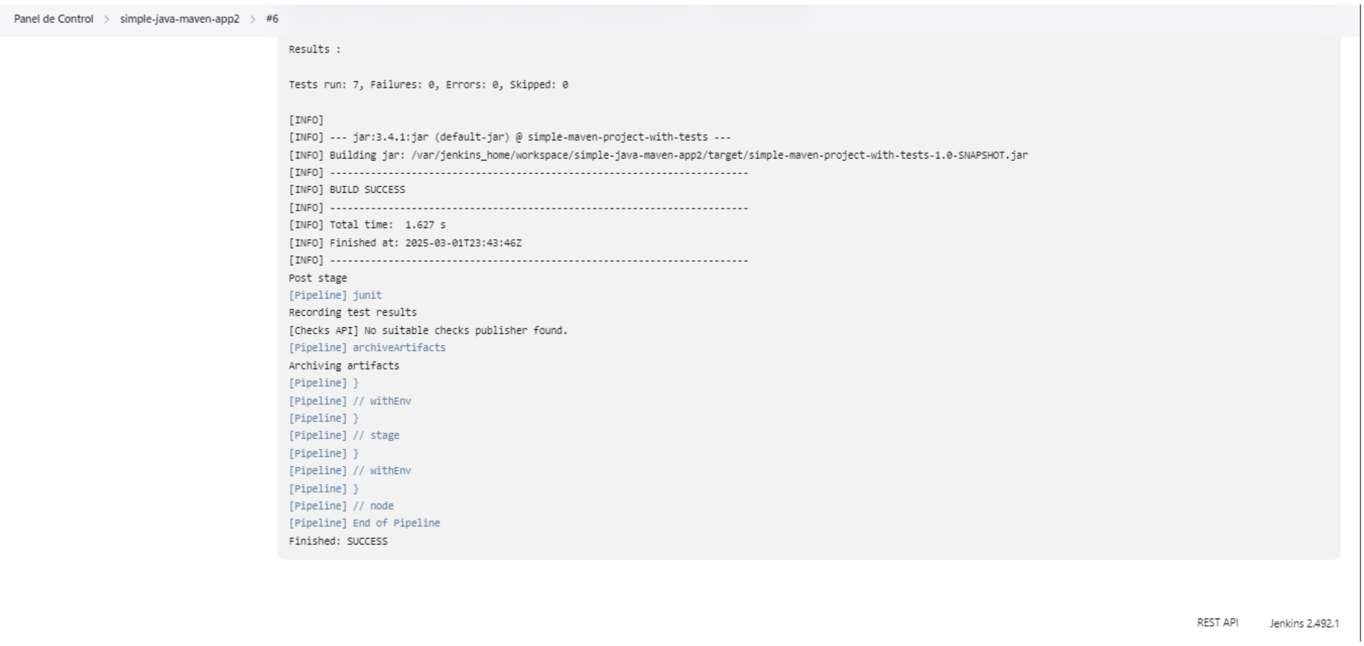


```
Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.855 s
[INFO] Finished at: 2025-03-01T23:28:45Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Modificar el script anterior para que verifique si el proyecto ya existe, en este caso elimine la carpeta y luego lo descargue de nuevo el proyecto.

```
// agregar estas stages al pipeline
stage('Check and Clean Workspace') {
            steps {
                script {
                    // Define the directory of the project
                    def projectDir = "simple-maven-project-with-tests"

                    // Check if the directory exists
                    if (fileExists(projectDir)) {
                        echo "Project directory exists. Deleting..."
                        // Delete the directory
                        sh "rm -rf ${projectDir}"
                    } else {
                        echo "Project directory does not exist. Proceeding to clone..."
                    }
                }
            }
        }

        stage('Clone Repository') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'
            }
        }
```

Ejecutar el pipeline y revise la consola.



```
Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.627 s
[INFO] Finished at: 2025-03-01T23:43:46Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```
Started by user admin

[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins
 in /var/jenkins_home/workspace/simple-java-maven-app2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Check and Clean Workspace)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] fileExists
[Pipeline] echo
Project directory does not exist. Proceeding to clone...
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] tool
```

```
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/simple-java-maven-
app2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/jglick/simple-maven-project-
with-tests.git # timeout=10
Fetching upstream changes from https://github.com/jglick/simple-maven-project-
with-tests.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.5'
 > git fetch --tags --force --progress -- https://github.com/jglick/simple-maven-
project-with-tests.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 9a5e9832814aa17fa7e514e5890fc6cf913d9a77
(refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
Commit message: "Creating a main class so there is something for `javadoc:javadoc`
to do"
 > git rev-list --no-walk 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ mvn -Dmaven.test.failure.ignore=true clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------< test:simple-maven-project-with-tests >----------------
[INFO] Building simple-maven-project-with-tests 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ simple-maven-project-with-tests ---
[INFO] Deleting /var/jenkins_home/workspace/simple-java-maven-app2/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ simple-maven-project-
with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-
```

```
java-maven-app2/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-java-maven-app2/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 3 source files with javac [debug target 1.8] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO] /var/jenkins_home/workspace/simple-java-maven-app2/src/test/java/test/Base.java: /var/jenkins_home/workspace/simple-java-maven-app2/src/test/java/test/Base.java uses or overrides a deprecated API.
[INFO] /var/jenkins_home/workspace/simple-java-maven-app2/src/test/java/test/Base.java: Recompile with -Xlint:deprecation for details.
[INFO]
[INFO] --- surefire:2.18.1:test (default-test) @ simple-maven-project-with-tests ---
[INFO] Surefire report directory: /var/jenkins_home/workspace/simple-java-maven-app2/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.SomeTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec - in test.SomeTest
Running test.OtherTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec - in test.OtherTest

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.627 s
[INFO] Finished at: 2025-03-01T23:43:46Z
[INFO] ------------------------------------------------------------------------
```

```
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Agregue un parámetro a Jenkins que permita escoger cual es la rama que se va a realizar la compilación, puede utilizar la rama declarative o testng disponibles en el proyecto de ejemplo

```
    parameters {
        choice(
            name: 'BRANCH',
            choices: ['declarative', 'testng'],
            description: 'Select the branch to build'
        )
    }
```

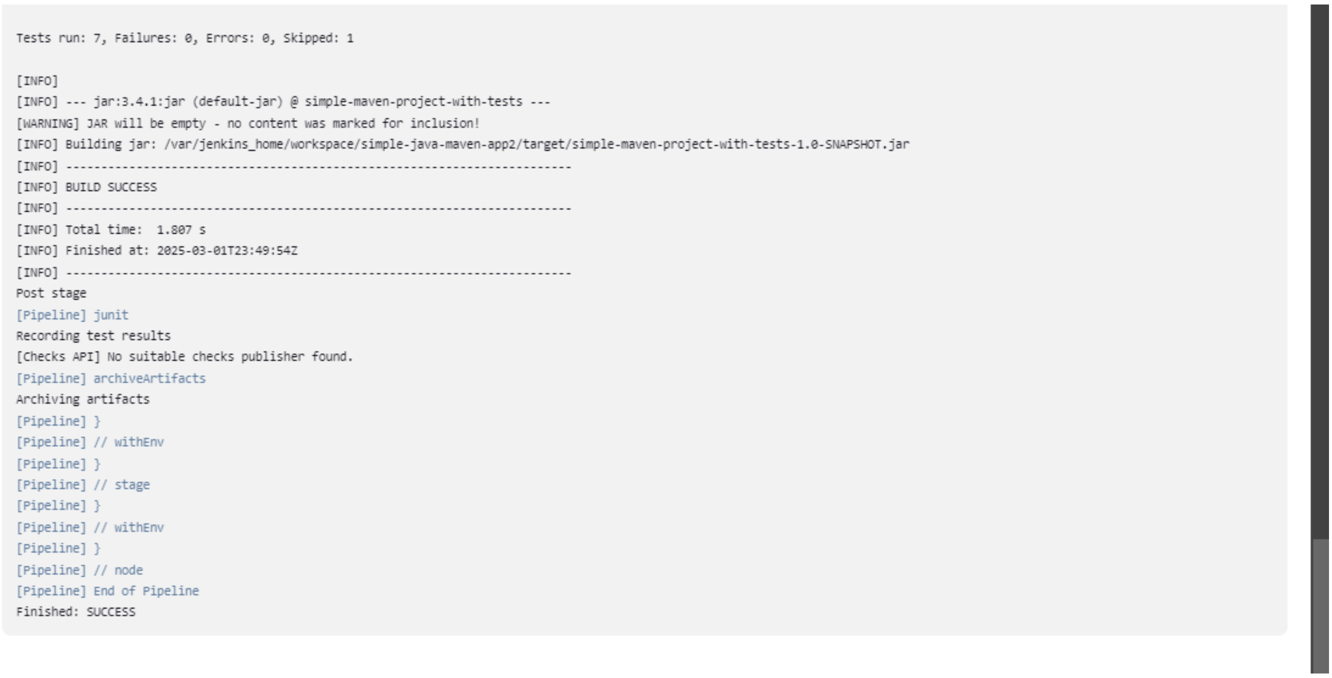
y en stages reemplazar

```
        stage('Clone Repository') {
            steps {
                // Get some code from a GitHub repository, using the selected branch
                git branch: "${params.BRANCH}", url: 'https://github.com/jglick/simple-maven-project-with-tests.git'
            }
        }
```

```
Ejecutar el pipeline y revise la consola.

Started by user admin

[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins
 in /var/jenkins_home/workspace/simple-java-maven-app2
```

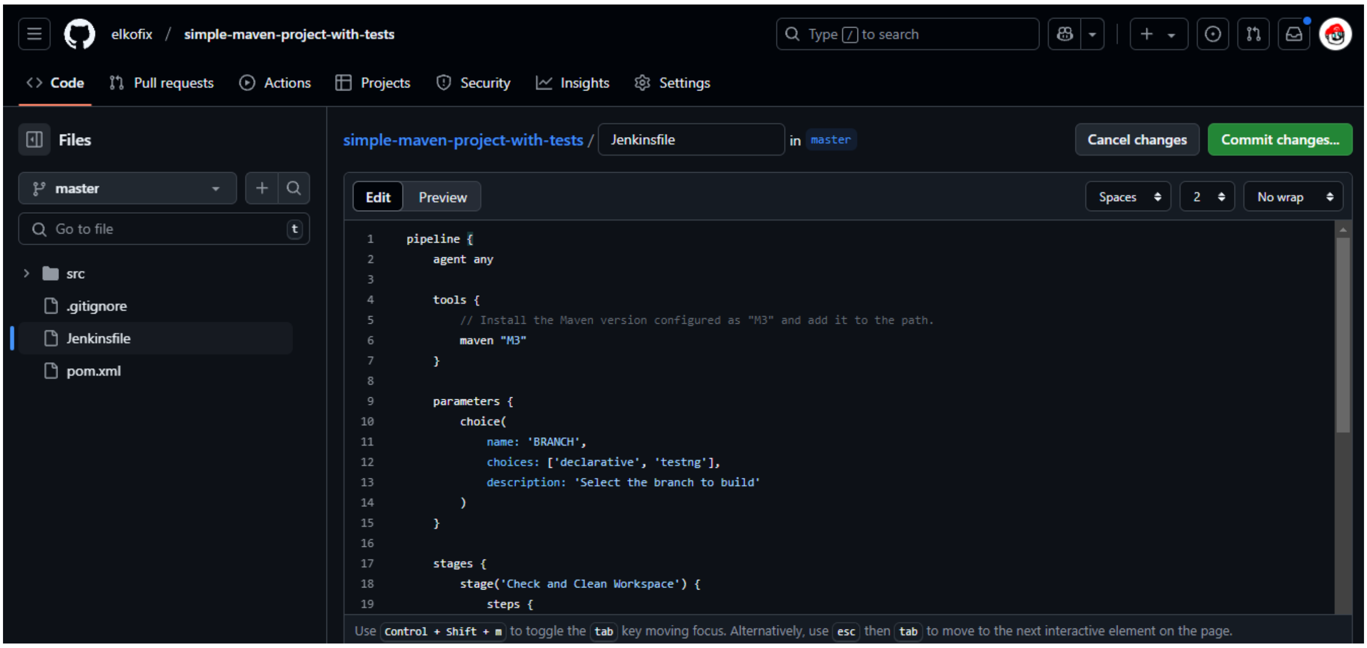
```
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Check and Clean Workspace)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] fileExists
[Pipeline] echo
Project directory does not exist. Proceeding to clone...
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/simple-java-maven-app2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/jglick/simple-maven-project-with-tests.git # timeout=10
Fetching upstream changes from https://github.com/jglick/simple-maven-project-with-tests.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.5'
 > git fetch --tags --force --progress -- https://github.com/jglick/simple-maven-project-with-tests.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/declarative^{commit} # timeout=10
Checking out Revision 88194bba5db3b3f5cd204e27c151b6788f896d9d (refs/remotes/origin/declarative)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D declarative # timeout=10
 > git checkout -b declarative 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
Commit message: "`label BUILD_TAG` deprecated"
 > git rev-list --no-walk 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ mvn -Dmaven.test.failure.ignore=true clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------< test:simple-maven-project-with-tests >----------------
[INFO] Building simple-maven-project-with-tests 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ simple-maven-project-with-tests ---
[INFO] Deleting /var/jenkins_home/workspace/simple-java-maven-app2/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-java-maven-app2/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ simple-maven-project-with-tests ---
[INFO] No sources to compile
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-java-maven-app2/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 3 source files with javac [debug target 1.7] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 7
[WARNING] source value 7 is obsolete and will be removed in a future release
[WARNING] target value 7 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- surefire:2.18.1:test (default-test) @ simple-maven-project-with-tests ---
[INFO] Surefire report directory: /var/jenkins_home/workspace/simple-java-maven-app2/target/surefire-reports
```

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.SomeTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 1, Time elapsed: 0.002 sec - in test.SomeTest
Running test.OtherTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec - in test.OtherTest

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 1

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.807 s
[INFO] Finished at: 2025-03-01T23:49:54Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



```
Tests run: 7, Failures: 0, Errors: 0, Skipped: 1

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.807 s
[INFO] Finished at: 2025-03-01T23:49:54Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Forkeamos el repo y agregamos nuestro pipeline en el Jenkinsfile



elkofix / simple-maven-project-with-tests
Type / to search
Code
Pull requests
Actions
Projects
Security
Insights
Settings
Files
master
Go to file
src
.gitignore
Jenkinsfile
pom.xml
simple-maven-project-with-tests / Jenkinsfile in master
Cancel changes
Commit changes...
Edit
Preview
Spaces
2
No wrap

```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    parameters {
        choice(
            name: 'BRANCH',
            choices: ['declarative', 'testng'],
            description: 'Select the branch to build'
        )
    }

    stages {
        stage('Check and Clean Workspace') {
            steps {
```

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Luego usamos la opcion de Pipeline script from SCM y con el link del repo

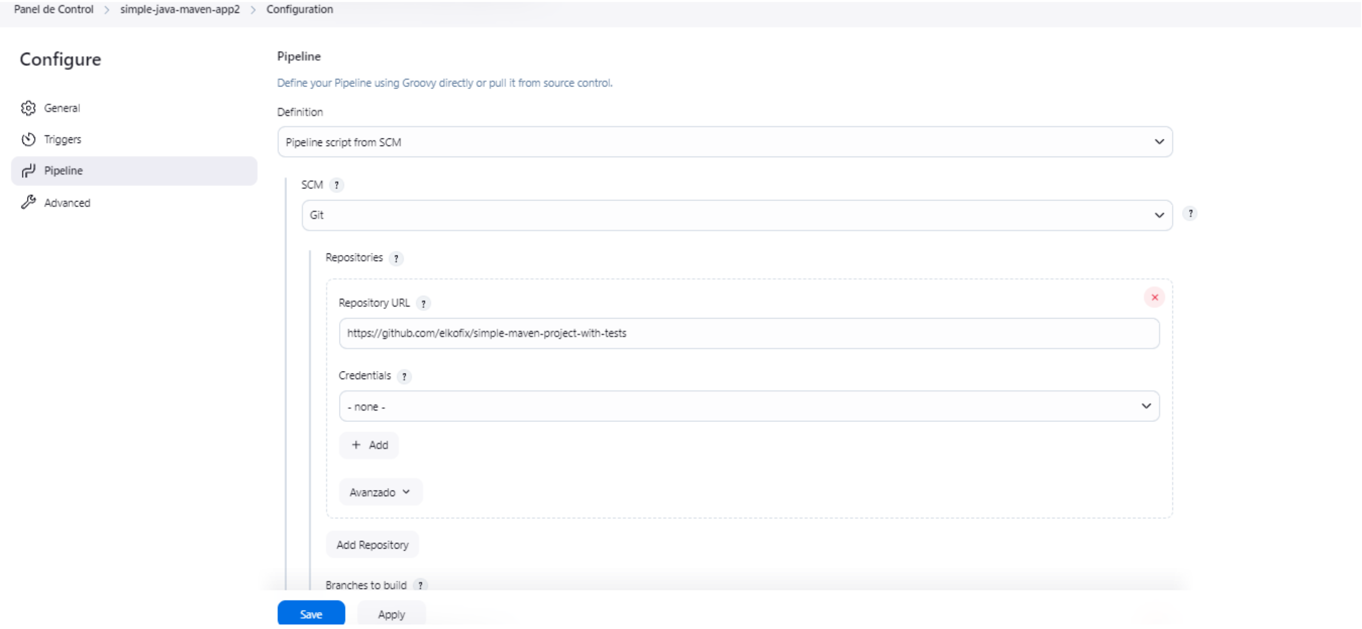
```
Started by user admin

Obtained Jenkinsfile from git https://github.com/elkofix/simple-maven-project-with-tests
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins
 in /var/jenkins_home/workspace/simple-java-maven-app2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
```

```
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/simple-java-maven-app2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/elkofix/simple-maven-project-with-tests # timeout=10
Fetching upstream changes from https://github.com/elkofix/simple-maven-project-with-tests
 > git --version # timeout=10
 > git --version # 'git version 2.39.5'
 > git fetch --tags --force --progress -- https://github.com/elkofix/simple-maven-project-with-tests +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision e687e649be5d96376437de41bed08e049397f0ba (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f e687e649be5d96376437de41bed08e049397f0ba # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Check and Clean Workspace)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] fileExists
[Pipeline] echo
Project directory does not exist. Proceeding to clone...
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] tool
[Pipeline] envVarsForTool
```

```
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/simple-java-maven-app2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/jglick/simple-maven-project-with-tests.git # timeout=10
Fetching upstream changes from https://github.com/jglick/simple-maven-project-with-tests.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.5'
 > git fetch --tags --force --progress -- https://github.com/jglick/simple-maven-project-with-tests.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/declarative^{commit} # timeout=10
Checking out Revision 88194bba5db3b3f5cd204e27c151b6788f896d9d (refs/remotes/origin/declarative)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D declarative # timeout=10
 > git checkout -b declarative 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
Commit message: "`label BUILD_TAG` deprecated"
 > git rev-list --no-walk 88194bba5db3b3f5cd204e27c151b6788f896d9d # timeout=10
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ mvn -Dmaven.test.failure.ignore=true clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------< test:simple-maven-project-with-tests >----------------
[INFO] Building simple-maven-project-with-tests 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ simple-maven-project-with-tests ---
[INFO] Deleting /var/jenkins_home/workspace/simple-java-maven-app2/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-java-maven-app2/src/main/resources
```

```
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ simple-maven-project-with-tests ---
[INFO] No sources to compile
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/jenkins_home/workspace/simple-java-maven-app2/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 3 source files with javac [debug target 1.7] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 7
[WARNING] source value 7 is obsolete and will be removed in a future release
[WARNING] target value 7 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- surefire:2.18.1:test (default-test) @ simple-maven-project-with-tests ---
[INFO] Surefire report directory: /var/jenkins_home/workspace/simple-java-maven-app2/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.SomeTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 2, Time elapsed: 0.003 sec - in test.SomeTest
Running test.OtherTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 1, Time elapsed: 0.001 sec - in test.OtherTest

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 3

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: /var/jenkins_home/workspace/simple-java-maven-app2/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.703 s
[INFO] Finished at: 2025-03-02T00:36:18Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
```

```
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/elkofix/simple-maven-project-with-tests

Credentials: - none -

Branches to build