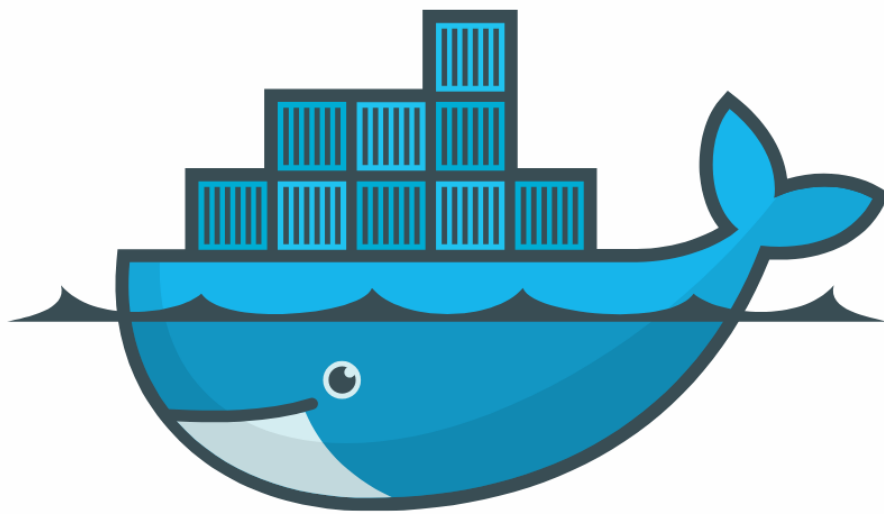


VOLÚMENES: PERSISTIENDO DATOS EN JENKINS



Ivana Sánchez Pérez

Introducción

Jenkins es un servidor de automatización open-source utilizado para integración continua (CI/CD).

La imagen oficial de Jenkins en Docker proporciona una forma rápida de ejecutar un servidor Jenkins en contenedores. Por defecto:

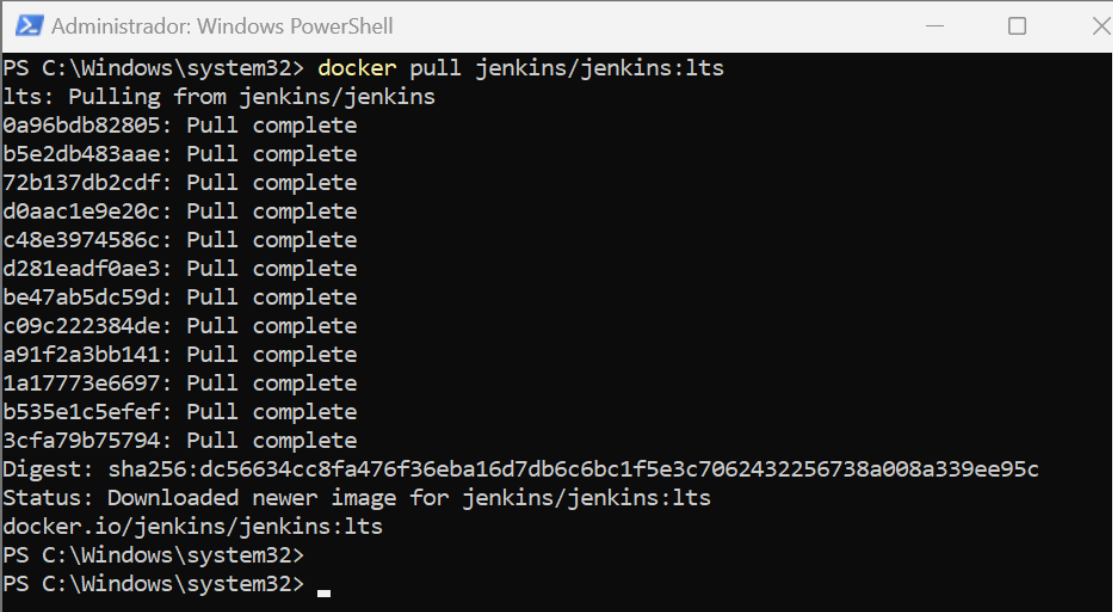
- Jenkins almacena sus datos en **/var/jenkins_home** dentro del contenedor.
- Jenkins expone dos puertos:
 - 8080 → Interfaz web.
 - 50000 → Conexiones de agentes.

Objetivo:

Ejecutar un contenedor de Jenkins con un volumen persistente para que sus configuraciones y proyectos no se pierdan si el contenedor se elimina.

Descargar la imagen oficial de Jenkins

Ejecutamos el comando **docker pull jenkins/jenkins:lts** en la PowerShell, que abriremos como administrador.



```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
0a96bdb82805: Pull complete
b5e2db483aae: Pull complete
72b137db2cdf: Pull complete
d0aac1e9e20c: Pull complete
c48e3974586c: Pull complete
d281eadf0ae3: Pull complete
be47ab5dc59d: Pull complete
c09c222384de: Pull complete
a91f2a3bb141: Pull complete
1a17773e6697: Pull complete
b535e1c5efef: Pull complete
3cfa79b75794: Pull complete
Digest: sha256:dc56634cc8fa476f36eba16d7db6c6bc1f5e3c7062432256738a008a339ee95c
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
PS C:\Windows\system32>
PS C:\Windows\system32>
```

Crear un contenedor con volumen persistente

Ejecutamos Jenkins y mapeamos un volumen en una ruta elegida para evitar pérdidas de datos.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker run -d --name jenkins_server `
>> -p 8080:8080 `
>> -p 50000:50000 `
>> -v C:\opt\jenkins_home:/var/jenkins_home `
>> jenkins/jenkins:lts
b856585cded3dd2725a1812749fbd1f73811cb7d5d57ab8bbebc10b3207b723f
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS
PORTS
b856585cded3   jenkins/jenkins:lts  "/usr/bin/tini -- /u..."  12 seconds ago  Up 11
seconds
0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp  jenkins_server
PS C:\Windows\system32>
```

Explicación del comando:

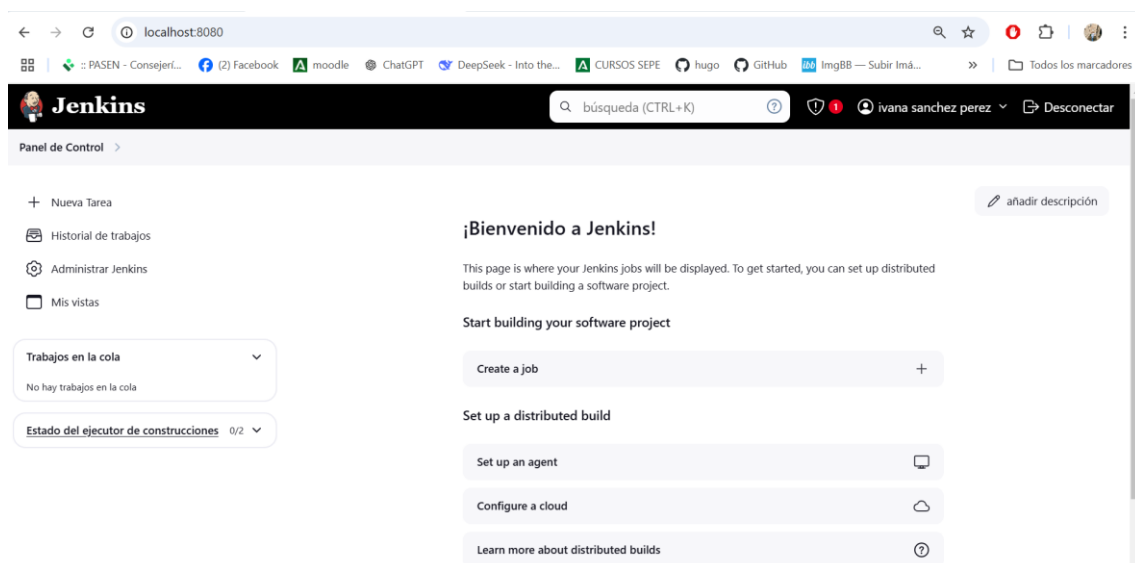
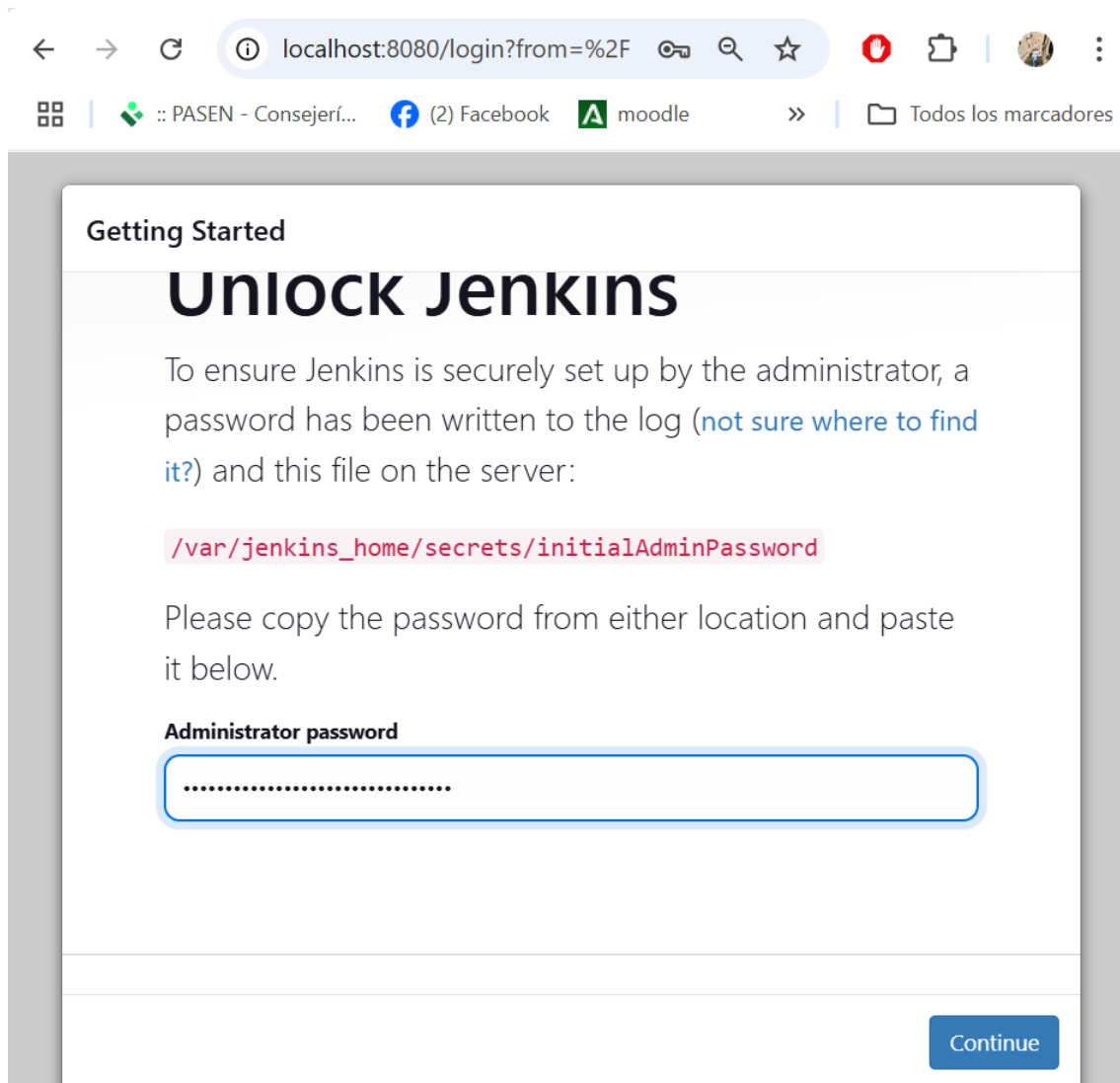
- **-d** → Ejecuta el contenedor en segundo plano.
- **--name jenkins_server** → Nombre del contenedor.
- **-p 8080:8080** → Expone la interfaz web en el puerto 8080.
- **-p 50000:50000** → Expone el puerto de comunicación para agentes.
- **-v C:\opt\jenkins_home:/var/jenkins_home** → Monta un volumen local en /var/jenkins_home, donde Jenkins guarda su configuración.
- **jenkins/jenkins:lts** → Usa la imagen oficial de Jenkins LTS.

Acceder a Jenkins y configurarlo

Para acceder a Jenkins por primera vez, necesitamos la clave de administrador generada automáticamente. La obtenemos con el comando **docker exec jenkins_server cat /var/jenkins_home/secrets/initialAdminPassword**

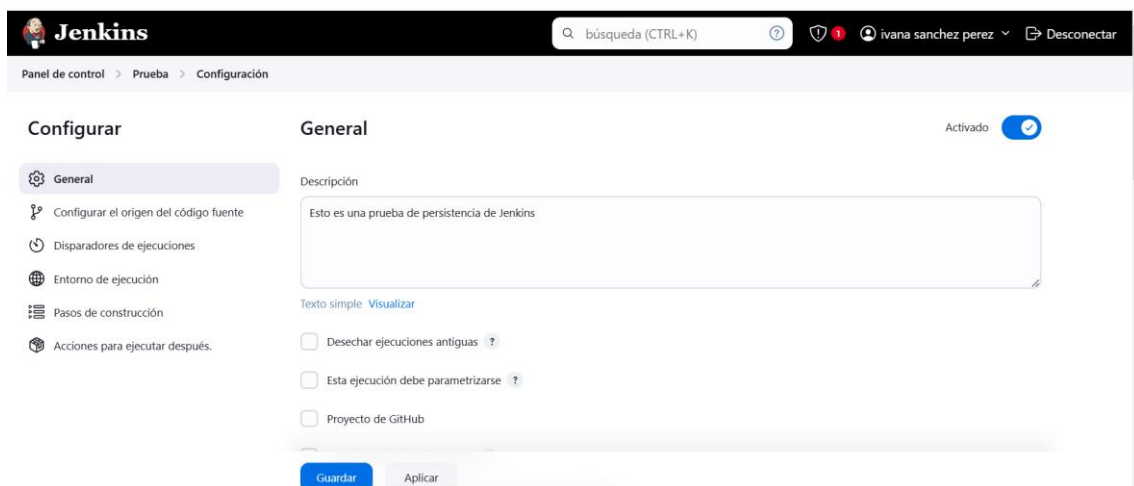
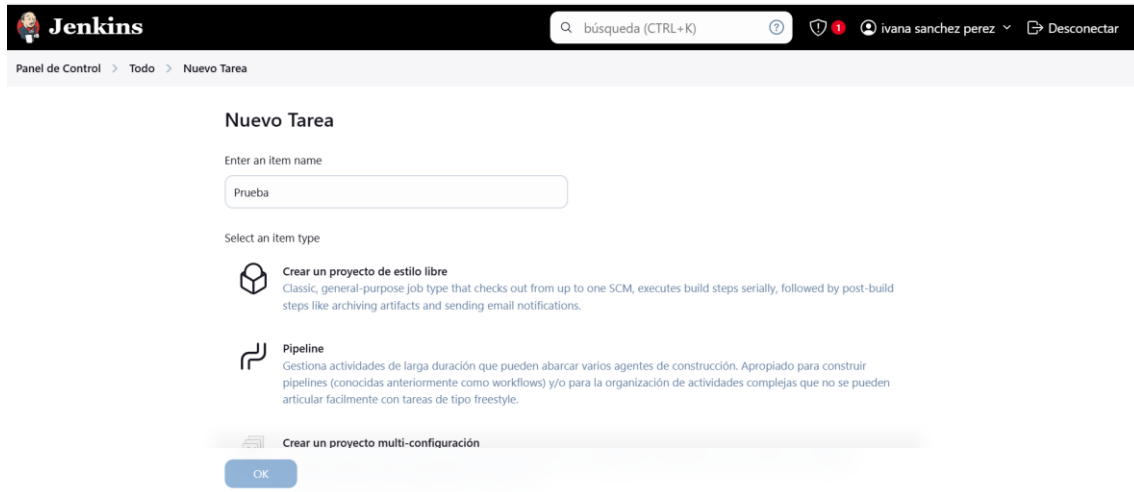
```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker exec jenkins_server cat /var/jenkins_home/secrets/initialAdminPassword
eb9421e3bbf4414db139a14afe62a2d4
PS C:\Windows\system32>
```

Copiamos la clave y accedemos a Jenkins en un navegador con <http://localhost:8080> y pegaremos la clave de administrador y seguiremos los pasos de configuración inicial.



Crear un Proyecto en Jenkins

En el panel de Jenkins hacemos clic en **Nueva tarea** y le damos un nombre, en mi caso **Prueba** → **Crear un proyecto de estilo libre** → **OK** y guardamos.



Comprobar la persistencia

Eliminamos el contenedor:

- **`docker stop jenkins_server`**
- **`docker rm jenkins_server`**

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker stop jenkins_server
jenkins_server
PS C:\Windows\system32> docker rm jenkins_server
jenkins_server
```

Si vamos a **C:\opt\jenkins_home**, seguirán estando los datos a pesar de que el contenedor no está corriendo.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Windows\system32> dir C:\opt\jenkins_home\

Directorio: C:\opt\jenkins_home

Mode                LastWriteTime         Length Name
----                -
da----             03/02/2025   0:04          .cache
d-----             03/02/2025   0:12        .groovy
d-----             03/02/2025   0:04         .java
d-----             03/02/2025   0:19         jobs
d-----             03/02/2025   0:12         logs
d-----             03/02/2025   0:12        plugins
da----             03/02/2025   0:13        secrets
d-----             03/02/2025   0:12        updates
d-----             03/02/2025   0:04     userContent
d-----             03/02/2025   0:13         users
d-----             03/02/2025   0:04         war
-a----             03/02/2025   0:04           0 .lastStarted
-a----             03/02/2025   0:04      1663 config.xml
-a----             03/02/2025   0:04       108 copy_reference_file.log
-a----             03/02/2025   0:04       156 hudson.model.UpdateCenter.xml
-a----             03/02/2025   0:12       370 hudson.plugins.git.GitTool.xml
-a----             03/02/2025   0:12     1680 identity.key.enc
-a----             03/02/2025   0:13          7 jenkins.install.InstallUtil.lastE
xecVersion
-a----             03/02/2025   0:13          7 jenkins.install.UpgradeWizard.sta
te
-a----             03/02/2025   0:13      179 jenkins.model.JenkinsLocationConf
iguration.xml
-a----             03/02/2025   0:04       171 jenkins.telemetry.Correlator.xml
-a----             03/02/2025   0:04     1037 nodeMonitors.xml
-a----             03/02/2025   0:22       258 queue.xml
-a----             03/02/2025   0:04        64 secret.key
-a----             03/02/2025   0:04          0 secret.key.not-so-secret
```

Recreamos el contenedor con el mismo volumen.

```
Administrador: Windows PowerShell

PS C:\Windows\system32> docker run -d --name jenkins_server `
>> -p 8080:8080 `
>> -p 50000:50000 `
>> -v C:\opt\jenkins_home:/var/jenkins_home `
>> jenkins/jenkins:lts
c271fc96e68913b7abccbd8c2d3d0a6b6919c5de9fdf7aefe9bfe0f35fbb1e14
PS C:\Windows\system32>
```

Verificamos que el proyecto creado sigue existiendo

