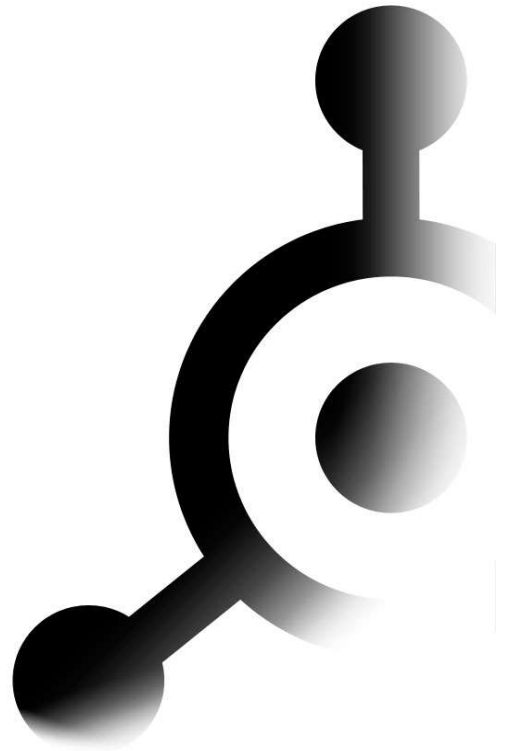




Instalación Jenkins



Control de versiones

Histórico

Versión	Fecha Creación	Creador	Fecha Revisión	Revisor	Fecha Aprobación	Aprobador
1.0	22/08/22	Antonio Lledó		Miguel Sánchez		
1.1	23/01/23	Antonio Lledó		Miguel Sánchez		

Cambios

Versión	Índice	Cambios respecto a la última versión
1	0	Versión inicial
1	1	Añadido método de instalación WAR

Tabla de Contenido

1. Introducción.	4
2. Instalación a partir de archivo WAR	4
3. Puesta en marcha rápida de un servidor Jenkins mediante Docker	4
4. Instalación	4
Instalación de Docker en UBUNTU 22.04 LTS	4
Configurar Jenkins en Docker para usar Docker	5
Dockerfile	5
Construir la imagen	6
Lanzar contenedor con Jenkins	7
Post-installation setup wizard	7
Prueba de Docker in Docker con Jenkins	9
Limpiar el entorno	12
5. Agregar nodo Docker a Jenkins	12
Contenedor Jenkins	12
Contenedor Nodo	12
Usar nube.	16
6. Usando docker-compose	16
Instalación docker-compose	16
Archivo docker-compose.yaml	17
Operaciones con docker-compose	18
7. Solución de problemas	19
Configurando un alias de host para SSH	19
Redirección de puertos con SSH (localhost.io)	20

1. Introducción.

El objetivo de este tema es preparar un servidor Jenkins

2. Instalación a partir de archivo WAR

<https://www.jenkins.io/doc/book/installing/war-file/>

java -jar jenkins.war

3. Puesta en marcha rápida de un servidor Jenkins mediante Docker

- **Instala sed, git, docker y docker-compose en tu sistemas**
- Descargar proyecto de [git@github.com:alledova/bootcamp-lifecycle.git](https://github.com/alledova/bootcamp-lifecycle.git)
Necesitarás esta llave privada [lifecycle.pem](#). ¿No sabes qué hacer con esta llave privada? pues igual la puesta en marcha rápida no es para tí. ¿Aún crees que sí? Mira cómo [configurar un alias de host para SSH](#), en este caso para github.com. Descarga rápida lifecycle.pem:

```
$ curl -L "https://drive.google.com/uc?export=download&id=1H8J1nr9rr61m4e2A0Xr_ceJsJ7ADP1L7" -o lifecycle.pem
```
- Cambia al directorio del proyecto y ejecuta el script `jenkins-bootcamp.sh`
- Accede mediante el navegador a la dirección <http://localhost:8080>
- Sigue las instrucciones del punto
📖 Bootcamp Devops lifeCycle - Jenkins Docker in Docker , para configurar Jenkins.

4. Instalación

Como base de la instalación se toma una máquina virtual en VirtualBox con Ubuntu 22.04 LTS
Instalación de escritorio mínima en español.

Usuario administrador: **lifecycle**

Por facilidad de acceso se habilita el acceso SSH mediante par de claves rsa.:

Crear par de claves rsa, en nuestro equipo de trabajo, o en el base de instalación

```
$ ssh-keygen -t rsa -f lifecycle
```

Obtendremos el par de claves rsa en `$HOME/.ssh`:

- La clave privada, que usaremos para identificarnos en las conexiones SSH. **lifecycle**
- La clave pública, que hay que añadir a `/home/lifecycle/.ssh/authorized_keys` de equipo base de instalación. **lifecycle.pub**

Nota. SSH es muy sensible a los permisos establecer 644 para `authorized_keys` y 600 para `lifecycle`. Ejemplo:

```
$ chmod 644 $HOME/.ssh/authorized_keys
```

Para que sudo no pida password en el equipo base añadir a `/etc/sudoers.d` un archivo, nombra el archivo por ejemplo `lifecycle` ,con el siguiente contenido:

```
# No sudo password for user lifecycle
lifecycle ALL=(ALL) NOPASSWD: ALL
```

Instalación de Docker en UBUNTU 22.04 LTS

Siempre en el 'equipo base de instalación'

Add Docker's official GPG key:

```
$ sudo mkdir -p /etc/apt/keyrings
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Set up the Docker's repository:

```
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker Engine:

```
$ sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-compose
```

Configure Docker to start on boot:

```
$ sudo systemctl enable docker.service  
$ sudo systemctl enable containerd.service
```

Post-installation steps for Linux:

```
$ sudo usermod -aG docker lifecycle  
$ newgrp docker
```

Test install:

```
$ docker run hello-world  
$ curl --unix-socket /var/run/docker.sock http://localhost/version
```

Configurar Jenkins en Docker para usar Docker

Más sobre esta solución en:

<https://jpetazzo.github.io/2015/09/03/do-not-use-docker-in-docker-for-ci/>

Creemos el directorio **jenkins_home** para asegurar la persistencia de datos en **\$HOME**

```
$ mkdir $HOME/jenkins_home
```

NOTA.- En el caso de no haber seguido los puntos anteriores y el equipo base es distinto hay que revisar los permisos del directorio recién creado. Si nuestro usuario no tiene UID 1000 / GID 1000, el UID del usuario del contenedor Docker que lanzará a su vez los contenedores si tendrá ese UID, para evitar problemas de acceso a archivos debemos hacerlos coincidir. El GID del grupo docker también debe coincidir entre anfitrión e invitado. Por supuesto el usuario del sistema base ha de pertenecer al grupo docker.

Tomar nota del GID del grupo docker y del UID y del GID del usuario de nuestro equipo 'base de instalación' para hacerlos coincidir en la imagen docker del paso siguiente. Para ello ejecutar el comando id

```
$ id  
uid=1000(lifecycle) gid=1000(lifecycle) grupos=1000(lifecycle),...,998(docker)
```

Revisar los permisos en el sistema base del socket de Docker. Deben ser, usuario root, grupo docker y permiso de lectura y escritura para el usuario y el grupo

```
$ ls -l /var/run/docker.sock  
srw-rw---- 1 root docker 0 ago 13 11:59 /var/run/docker.sock
```

En caso contrario establecemos esos permisos

```
$ sudo chown root:docker /var/run/docker.sock  
$ sudo chmod 660 /var/run/docker.sock
```

Dockerfile

Creemos un archivo dockerfile, lo nombramos 'dockefile', para personalizar la imagen docker de la que partimos jenkins/jenkins:lts-jdk11. Cambia el gid del grupo *docker* por el obtenido al ejecutar

el comando *id* en el paso anterior, en rojo, uid de usuario, gid del grupo principal de usuario y gid de grupo docker

```
FROM jenkins/jenkins:lts-jdk11

ARG gid_docker=999
ARG uid_user=1001
ARG gid_user=1001

USER root

RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSL /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 docker-ce-cli
containerd.io docker-compose-plugin build-essential curl vim ssh netcat
RUN systemctl enable docker.service
RUN systemctl enable containerd.service
RUN groupmod -g ${gid_docker} docker # gid grupo docker
RUN usermod -aG docker jenkins

RUN curl -fsSL https://deb.nodesource.com/setup_18.x | bash -
RUN apt-get install -y nodejs

USER jenkins
RUN jenkins-plugin-cli --plugins "docker-workflow:521.v1a_a_dd2073b_2e
docker-plugin:1.2.9 docker-build-step:2.8"
USER root
RUN usermod -u $uid_user jenkins # uid usuario equipo base
RUN groupmod -g $gid_user jenkins # gid grupo principal usuario base
RUN chown -R jenkins:jenkins /usr/share/jenkins/
RUN chown -R jenkins:jenkins /var/jenkins_home/
USER jenkins
```

Construir la imagen

```
$ docker build -f dockerfile -t myjenkins .
```

Lanzar contenedor con Jenkins

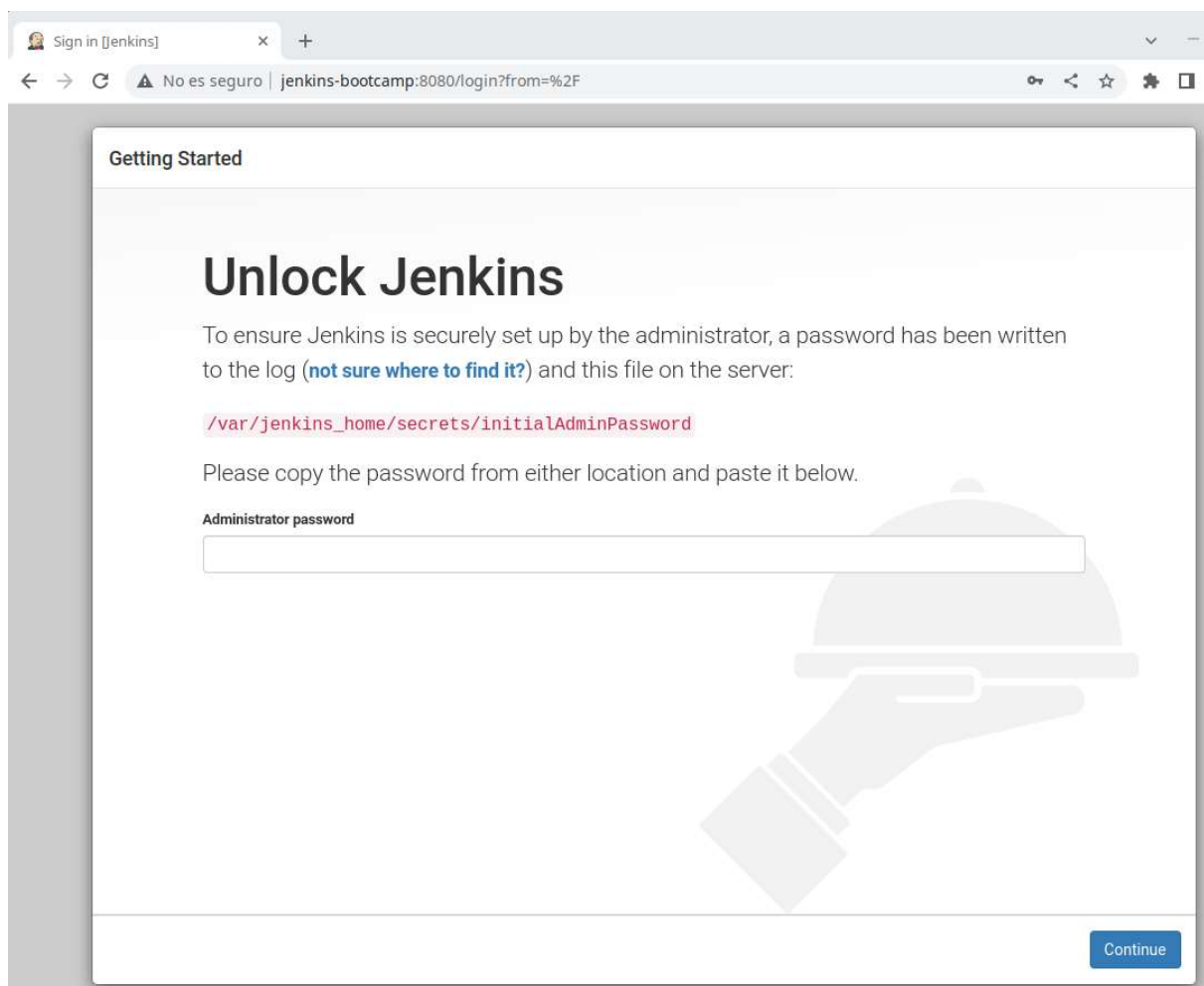
```
$ docker run --name jenkins-bootcamp -d -p 8080:8080 -p 50000:50000 -v  
$HOME/jenkins_home:/var/jenkins_home -v  
/var/run/docker.sock:/var/run/docker.sock myjenkins
```

Post-installation setup wizard

Accedemos a la interfaz web de Jenkins en la dirección IP de nuestro equipo base de instalación puerto 8080. Si nuestro base de instalación es una MV con interfaz de red en modo puente se recomienda por facilidad de acceso realizar un NAT de puerto desde VirtualBox, o el hypervisor utilizado

Seguimos los pasos del asistente de configuración.

<https://www.jenkins.io/doc/book/installing/docker/#setup-wizard>



Para conocer el password cat sobre el archivo indicado. A través del contenedor

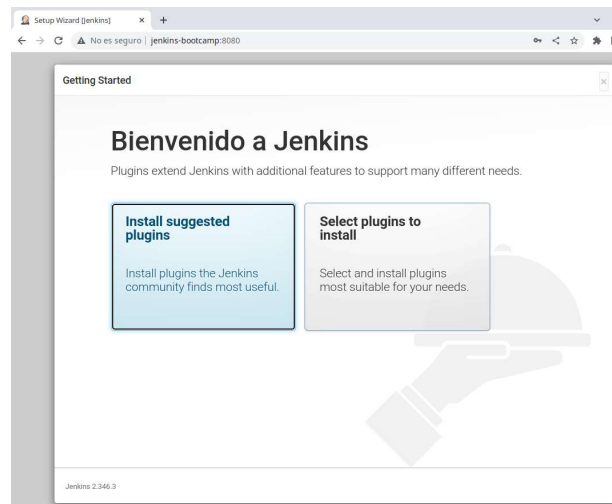
```
$ docker exec jenkins-bootcamp cat /var/jenkins_home/secrets/initialAdminPassword
```

o a través del volumen compartido

```
$ cat $HOME/jenkins_home/secrets/initialAdminPassword
```

o a través del registro de la imagen

En el siguiente paso indicamos 'Install suggested plugins'



Introducimos los datos del que será nuestro usuario administrador

Getting Started

Create First Admin User

Usuario:

Contraseña:

Confirma la contraseña:

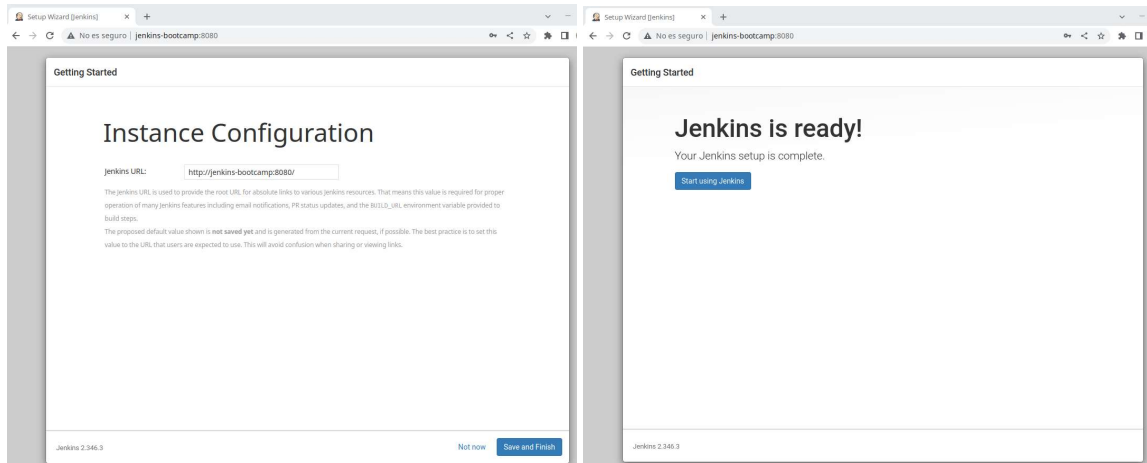
Nombre completo:

Dirección de email:

Jenkins 2.346.3

[Skip and continue as admin](#) [Save and Continue](#)

Por último indicamos la URL de acceso a Jenkins



TRUCO.- Editando el archivo `/etc/hosts` de nuestro equipo base y añadiendo una entrada que apunte a `127.0.0.1` podemos configurar el nombre deseado a la url de acceso a la consola Jenkins `127.0.0.1 localhost jenkins-bootcamp`

Prueba de Docker in Docker con Jenkins


Para probar nuestro contenedor vamos a 'Nueva tarea' Le damos un nombre y seleccionamos el tipo de proyecto Pipeline.




Enter an item name

whalesay


» 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) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por 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 (conocidas anteriormente como workflows) y/o para la organización de actividades 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 multiples entornos, ejecutar sobre plataformas concretas, etc.


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 a separate namespace, so you can have multiple things of the same name as long 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.

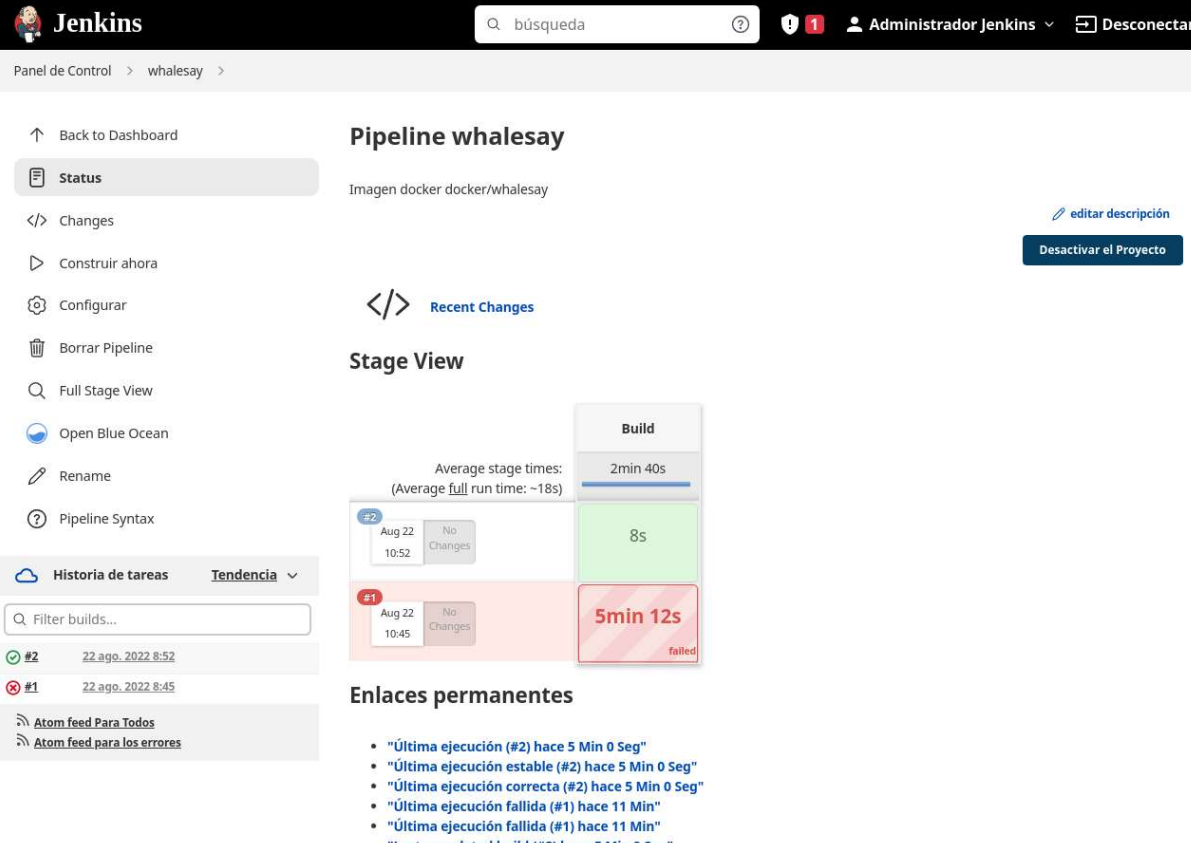
OK

Opcionalmente introducimos una descripción y el código del pipeline, que se muestra a continuación, en la sección Pipeline en Script

```
pipeline {
  agent any
  stages {
    stage('Build') {
      agent {
        docker {
          image 'docker/whalesay'
        }
      }
      steps {
        sh 'cowsay Hola Bootcamp Devops lifeCycle'
      }
    }
  }
}
```

Guardamos

Siguiente paso 'Construir ahora', en el menú de la izquierda



Jenkins búsqueda 1 Administrador Jenkins Desconectar

Panel de Control > whalesay >

Back to Dashboard

Status

Changes

Construir ahora

Configurar

Borrar Pipeline

Full Stage View

Open Blue Ocean

Rename

Pipeline Syntax

Historia de tareas Tendencia

Filter builds...

#2 22 ago. 2022 8:52

#1 22 ago. 2022 8:45

Atom feed Para Todos

Atom feed para los errores

Pipeline whalesay

Imagen docker docker/whalesay

editar descripción

Desactivar el Proyecto

Recent Changes

Stage View

Average stage times:
(Average full run time: ~18s)

Build

2min 40s

8s


5min 12s failed


Enlaces permanentes


- Última ejecución (#2) hace 5 Min 0 Seg"
- Última ejecución estable (#2) hace 5 Min 0 Seg"
- Última ejecución correcta (#2) hace 5 Min 0 Seg"
- Última ejecución fallida (#1) hace 11 Min"
- Última ejecución fallida (#1) hace 11 Min"
- Last completed build (#2) hace 5 Min 0 Seg"


Y realizamos el seguimiento de la tarea, si aún no se ve aparecerá en un breve instante en la parte inferior del menú de la izquierda. En la imagen se aprecian dos tareas, una con un resultado correcto y otra con una finalización con errores.


Pulsando sobre ellas veremos los detalles de la misma, podemos seguir la evolución de la tarea mientras se ejecuta en 'Console Output'

Jenkins



1


Administrador Jenkins


Desconectar


Panel de Control


> whalesay


> #2


Back to Project


Status


Changes


Console Output


Edit Build Information


Delete build "#2"


Open Blue Ocean


Restart from Stage


Replay

Pipeline Steps

Workspaces


Previous Build

Build #2 (22 ago. 2022 8:52:18)


Iniciado por el usuario [Administrador Jenkins](#)

Conservar esta ejecución para siempre

Started 9 Min 34 Seg ago
Took 18 Seg

añadir descripción

Registro de la tarea con una finalización correcta.


Jenkins

Panel de Control > whalesay > #2

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build #2

Open Blue Ocean


Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build


Salida de consola

Started by user [Administrador Jenkins](#)

```

[Pipeline] Start of Pipeline
[Pipeline] node

Running on Jenkins in /var/jenkins_home/workspace/whalesay
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] node

Running on Jenkins in /var/jenkins_home/workspace/whalesay@2
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker inspect -f . docker/whalesay
.
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withDockerContainer
Jenkins seems to be running inside container 1cd78de976b49840460a90d5a6e7ac817802f601e86d68500f6061a69246c3c2
$ docker run -t -d -u 1000:1000 -w /var/jenkins_home/workspace/whalesay@2 --volumes-from 1cd78de976b49840460a90d5a6e7ac817802f601e86d68500f6061a69246c3c2
***** .e ***** .e ***** .e ***** .e ***** .e ***** .e ***** .e ***** .e *****
$ docker top 6aaf2429682c79160b4d300ff30597e2ab600af89d8b271106727bef6c369817 -eo pid,comm
[Pipeline] {
[Pipeline] sh
+ cowsay Hola Bootcamp Devops LifeCycle

< Hola Bootcamp Devops LifeCycle >
-----
      \
       \
        \
         ##
        ## ##
       ## ## ##
      /-----\
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
    \           /
     \         /
      \       /
       \     /
        \   /
         \ /
          o
         / \
        /   \
       /     \
      /       \
     /         \
    /           \
   /             \
  /               \
 /                 \
/                   \
\                   /
 \                 /
  \               /
   \             /
```

Limpiar el entorno

Detenemos y eliminamos el contenedor generado, no el directorio de datos

```
$ docker stop jenkins-bootcamp  
$ docker rm jenkins-bootcamp
```

En el caso de querer limpiar todo ejecutamos

```
$ docker system prune
```

5. Agregar nodo Docker a Jenkins

Agregaremos un nuevo nodo a Jenkins, este nuevo nodo también será un contenedor Docker.

Para ello los dos contenedores han de verse a nivel de red. El primer paso será crear una red con Docker en la que ejecutaremos los dos contenedores.

```
$ docker network create jnetwork
```

Creamos el directorio que albergará el volumen de trabajo compartido entre nodos

```
$ mkdir $HOME/workspace  
$ chmod 777 $HOME/workspace
```

Contenedor Jenkins

Para el contenedor Jenkins reutilizamos la imagen construida anteriormente, la ejecutamos con dos nuevos parámetros, uno para indicar la red y otro el volumen de trabajo

```
$ docker run --name jenkins-bootcamp -d -p 8080:8080 -p 50000:50000 -v  
$HOME/jenkins_home:/var/jenkins_home -v $HOME/workspace:/root/workspace -v  
/var/run/docker.sock:/var/run/docker.sock --cgroupns host --network jnetwork  
--network-alias jenkins myjenkins
```

Contenedor Nodo

Para el nodo partiremos de la imagen [docker/docker](#), que personalizamos con el siguiente archivo *dockerfile.nodo*

```
FROM docker:dind
ENV SSH_PUBLIC_KEY="<INYECTAR AQUÍ EL CONTENIDO DE LA CLAVE PÚBLICA lifecycle.pub>"
ENV RUTA="/root"
RUN apk update
RUN apk add --update unzip curl wget shadow

#RUN groupmod -g 998 docker
RUN mkdir $RUTA/workspace && chmod 777 $RUTA/workspace

# Opcional, instalamos utilidades que necesitaremos más adelante
RUN apk add ansible aws-cli

RUN apk add nodejs-current npm

#Install JDK11 and Groovy
RUN apk add openjdk11
RUN wget
https://www.apache.org/dyn/closer.lua/groovy/4.0.4/distribution/apache-
groovy-binary-4.0.4.zip?action=download -O
$RUTA/apache-groovy-binary-4.0.4.zip && unzip
$RUTA/apache-groovy-binary-4.0.4.zip -d $RUTA/ && rm
$RUTA/apache-groovy-binary-4.0.4.zip
COPY ./java.sh /etc/profile.d/java.sh

#Enable ssh login
RUN apk add --update --no-cache openssh
RUN echo 'PasswordAuthentication yes' >> /etc/ssh/sshd_config
RUN echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config
RUN echo -n 'root:zeus2022' | chpasswd
RUN mkdir -p $RUTA/.ssh && chmod 755 $RUTA/.ssh && touch
$RUTA/.ssh/authorized_keys && chmod 644 $RUTA/.ssh/authorized_keys
RUN echo $SSH_PUBLIC_KEY >> $RUTA/.ssh/authorized_keys
COPY ./entrypoint.sh $RUTA/entrypoint.sh
RUN chmod +x $RUTA/entrypoint.sh

ENTRYPOINT ["/root/entrypoint.sh"]
```

Como se puede ver examinando el contenido del archivo dockerfile.nodo para construir nuestra imagen personalizada necesitaremos los siguientes archivos:

java.sh para establecer las variables de entorno relacionadas con Java y Groovy

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk
export GROOVY_HOME=/root/groovy-4.0.4
export PATH=$GROOVY_HOME/bin:$PATH
```

entrypoint.sh script encargado de lanzar el servicio SSHD en el nodo. Jenkin y nuestro nodo se comunicaran mediante este protocolo

```
#!/bin/sh
ssh-keygen -A
/usr/sbin/sshd -D -e "$@"
```

Con estos dos archivos ya podemos construir la imagen de nuestro nodo personalizada.

```
$ docker image build -f dockerfile.nodo . -t dockernodo
```

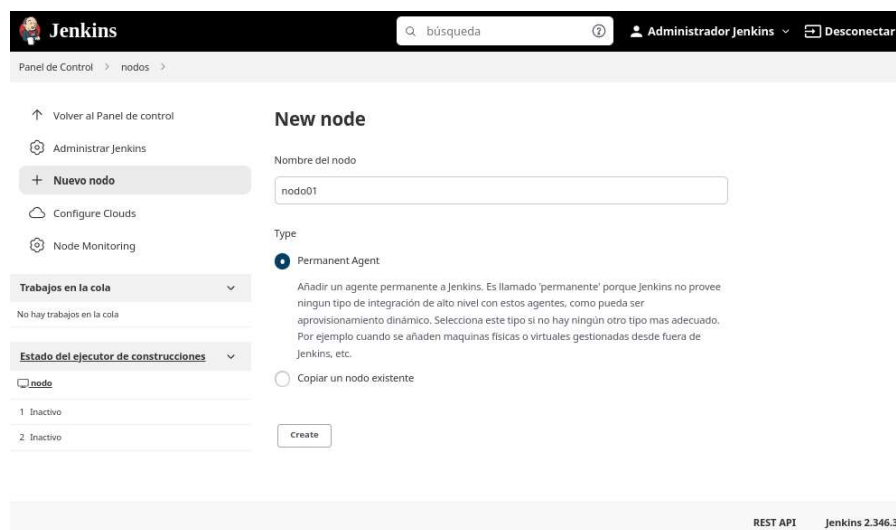
Y lanzamos el contenedor del nodo

```
$ docker run --name nodo01 -d -p 2022:22 -v $HOME/workspace:/root/workspace -v
/var/run/docker.sock:/var/run/docker.sock --network jnetwork --network-alias
nodo dockernodo
```


Es posible que no se resuelva el nombre DNS de nuestro contenedor nodo, 'nodo', en el contenedor Jenkins, para evitar este problema tomamos nota de su dirección IP

```
$ docker exec nodo01 ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
55: eth0@if56: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    state UP
    link/ether 02:42:ac:13:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.21.0.2/16 brd 172.19.255.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Ya estamos en disposición de agregar el nuevo nodo a Jenkins. Para ello accedemos mediante navegador a la consola de administración de Jenkins y en 'Administrar Jenkins' -> 'Administrar nodos' -> 'Nuevo nodo'



Introducimos nombre del nodo y seleccionamos 'Permanet agent' pulsamos sobre 'create'


Jenkins

Administrador Jenkins

Panel de Control > nodos >

↑ Volver al Panel de control
 ⚙ Administrar Jenkins
 + Nuevo nodo
 ☁ Configure Clouds
 📊 Node Monitoring

Trabajos en la cola
 No hay trabajos en la cola
Estado del ejecutor de construcciones
 📦 **nodo**
 1 Inactivo
 2 Inactivo

Nombre

Descripción

Number of executors

Directorio raíz remoto

Etiquetas

Usar

Metodo de ejecución

Nombre de máquina

Credentials

 + Add

Host Key Verification Strategy

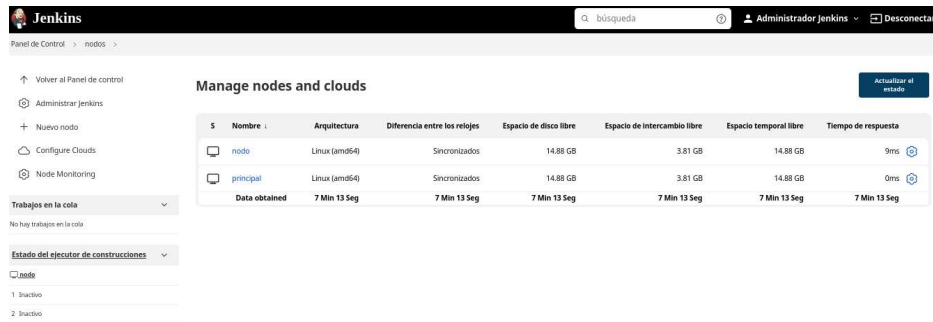
Avanzado...

Disponibilidad

Propiedades del nodo
☐ Disable deferred wipeout on this node
☐ Localización de herramientas
☐ Variables de entorno
 Guardar

- **Nombre:** nodo01
- **Number of executors:** 2
- **Método de ejecución:** Arrancar agentes remotos en máquinas Unix via SSH
- **Nombre de la máquina:** La IP obtenida anteriormente.
- **Credentials:** Creamos unas credenciales nuevas. Lo ideal es usar la clave privada lifecycle creada al principio, Si da problemas cambiar a usuario/contraseña. root/<Establecida en el dockerfile.nodo>

No es necesario introducir ningún valor más. Pulsamos sobre Guardar



Editamos nuestro nodo principal, pulsando sobre la rueda dentada a la derecha, y establecemos el número de ejecutores a 0. Guardar los cambios. De ese modo desaparece la alerta de seguridad y nos aseguramos que las tareas se ejecutarán en el nuevo nodo.

Para probarlo, 'Construir ahora' en nuestro proyecto de pruebas, whalesay

Usar nube.

Una forma distinta de llegar al mismo punto es con la opción de agregar nube, para ello las imágenes de los contenedores y la forma de lanzarlas son un poco distintas, consultar los siguientes enlaces:

1. [Docker](#)
2. [Run Jenkins in a Docker container — part 1](#)
3. [ipetazzo/Using Docker-in-Docker for your CI or testing environment? Think twice.](#)

6. Usando docker-compose

docker-compose nos permite configurar y gestionar los dos contenedores descritos de una forma sencilla y conjunta. El único condicionante es la opción `--cgroups` que regula la forma de acceder a los tablespaces

El servicio Docker tiene establecido el valor de dicha opción a `"host"` por defecto `--default-cgroupns-mode "host"`. Si estuviera establecida a `"private"` para usar *docker-compose* debemos intentar cambiar el valor en la unidad de servicio *systemd* de Docker y reiniciar el servicio

```
--default-cgroupns-mode "host"
```

Si no podemos cambiar la unidad de servicio habrá que ejecutar los contenedores como se ha descrito anteriormente, añadiendo la opción `--cgroups host` y no podremos usar *docker-compose*, aún no tiene soporte para gestionar esta opción.

Instalación docker-compose

Usando apt

```
$ sudo apt install docker-compose
```

Si nuestra distribución Ubuntu es anterior a la 22.04 LTS *docker-compose* puede dar problemas de versión seguir este procedimiento para actualizarlo

Eliminar la versión anterior si se instaló con apt, apt-get

```
$ sudo apt-get remove docker-compose
```

Eliminar la versión anterior si se instaló manualmente

```
$ ruta=$(whereis -b docker-compose) && sudo rm ${ruta#docker-compose: }
```

Instalación manual

```
$ DESTINATION=/usr/bin/docker-compose
```



```
$ sudo curl -L
https://github.com/docker/compose/releases/download/${VERSION}/docker-compose-$(
uname -s)-$(uname -m) -o $DESTINATION
$ sudo chmod 755 $DESTINATION
```

Archivo docker-compose.yaml

Una vez instalado d Operaciones con docker-compose

Construir y lanzar los contenedores en background

```
$ docker-compose up -d --build
```

Quitar el modificador -d si queremos lanzarlos y ver el resultado en consola.

Consultar registro

```
$ docker-compose compose logs
```

Detener los contenedores

```
$ docker-compose down
```

7. Despliegue de Jenkins en Minikube mediante Helm.

Paso seguidos:

- Instalar Minikube: desde binario o paquete de la distribución

```
sudo apt install ./minikube_latest_amd64.deb
```

-Lanzar minikube

```
minikube --memory 4096 --cpus 2 start --vm-driver=virtualbox -p jenkins
```

- minikube status -p jenkins

- minikube dashboard

- brew install kubectl

- brew install kubernetes-helm

- kubectl get ns

- kubectl create -f minikube/jenkins-namespace.yaml

- kubectl create -f minikube/jenkins-volume.yaml

- minikube mount ./data:/data/jenkins-volume

- helm init

- helm repo add jenkins https://charts.jenkins.io

- helm repo update

- helm search repo jenkins

- helm pull jenkins/jenkins

Instalación

- helm show values jenkins/jenkins > values.yaml

- serviceType: ClusterIP → NodePort (línea 129)

- nodePort: 32000 (descomentar línea ~146)

- volumes [] →

volumes: (línea ~647)

- type: HostPath

hostPath: /var/run/docker.sock

mountPath: /var/run/docker.sock

- storageClass: → jenkins-pv (línea 818)

- helm upgrade --install jenkins jenkins/jenkins -n jenkins -f helm/jenkins-values.yaml --namespace

jenkins-project

- helm list --namespace jenkins-project

8. Solución de problemas

El problema más pertinaz encontrado ha sido **Jenkins process is stuck**, las siguientes recomendaciones han servido de ayuda.

This error means the Jenkins process is stuck on some command. Some suggestions:

- ☐ Upgrade all of your plugins and re-try.
- ☐ Make sure you've the right number of executors and jobs aren't stuck in the queue.
- ☐ If you're pulling the image (not your local), try adding alwaysPull true (next line to image).
- ☐ When using agent inside stage, remove the outer agent. See: JENKINS-63449.
- ☐ Execute `org.jenkinsci.plugins.durabletask.BourneShellScript.LAUNCH_DIAGNOSTICS=true` in Jenkins's Script Console to debug.
- ☐ When the process is stuck, SSH to Jenkins VM and run `docker ps` to see which command is running.
- ☐ Run `docker ps -a` to see the latest failed runs. In my case it tried to run `cat` next to custom CMD command set by container (e.g. `ansible-playbook cat`), which was the invalid command. The `cat` command is used by design. To change entrypoint, please read JENKINS-51307.
- ☐ If your container is still running, you can login to your Docker container by `docker exec -it -u $(docker ps -ql) bash` and run `ps wuax` to see what's doing.
- ☐ Try removing some global variables (could be a bug), see: parallel jobs not starting with docker workflow.

Ha sido de especial ayuda ejecutar:

```
org.jenkinsci.plugins.durabletask.BourneShellScript.LAUNCH_DIAGNOSTICS=true
```

en la consola de scripts de Jenkins, la ejecución del comando ha permitido mostrar errores que antes permanecían ocultos, determinando que la ejecución del pipeline *whalesay* se detenía al no haber correspondencia al workspace entre servidor y nodo. Para solucionar el problema se ha preparado un volumen para el workspace entre ambos contenedores.

Los **problemas derivados de los permisos de archivos** se corrigen si establecemos correctamente UID y GID en el archivo *dockerfile* de la imagen Jenkins. El único requisito previo es crear el directorio `$HOME/jenkins_home` antes de lanzar los contenedores.

Configurando un alias de host para SSH

Para solucionar problemas de identificación con clave privada al acceder a servidores remotos mediante SSH o a repositorios GIT con protocolo SSH, podemos establecer alias de host remotos

indicando sus credenciales de acceso en el archivo *config* en el directorio *.ssh* de nuestro directorio de usuario. `$HOME/.ssh/config`, ejemplo

```
Host github.com
  Hostname github.com
  IdentityFile ~/.ssh/lifecycle.pem
  IdentitiesOnly yes
```

No olvidar copiar la clave privada en la ruta indicada en *IdentityFile* y establecerle permisos de archivo 400.

Con `Host github.com` establecemos un alias al host indicado en `Hostname`, en el ejemplo al dejar los valores igual no establecemos ningún alias. `IdentityFile` indica la ruta a la clave privada. Con `IdentitiesOnly yes` cambiamos el comportamiento de SSH, al iniciar una identificación renuncia a establecer la conexión probando primero con la clave privada por defecto, usualmente *id_rsa*

Redirección de puertos con SSH (localhost.io)