

Tarea 23: Creacion de los Pipeline en los distintos lenguajes



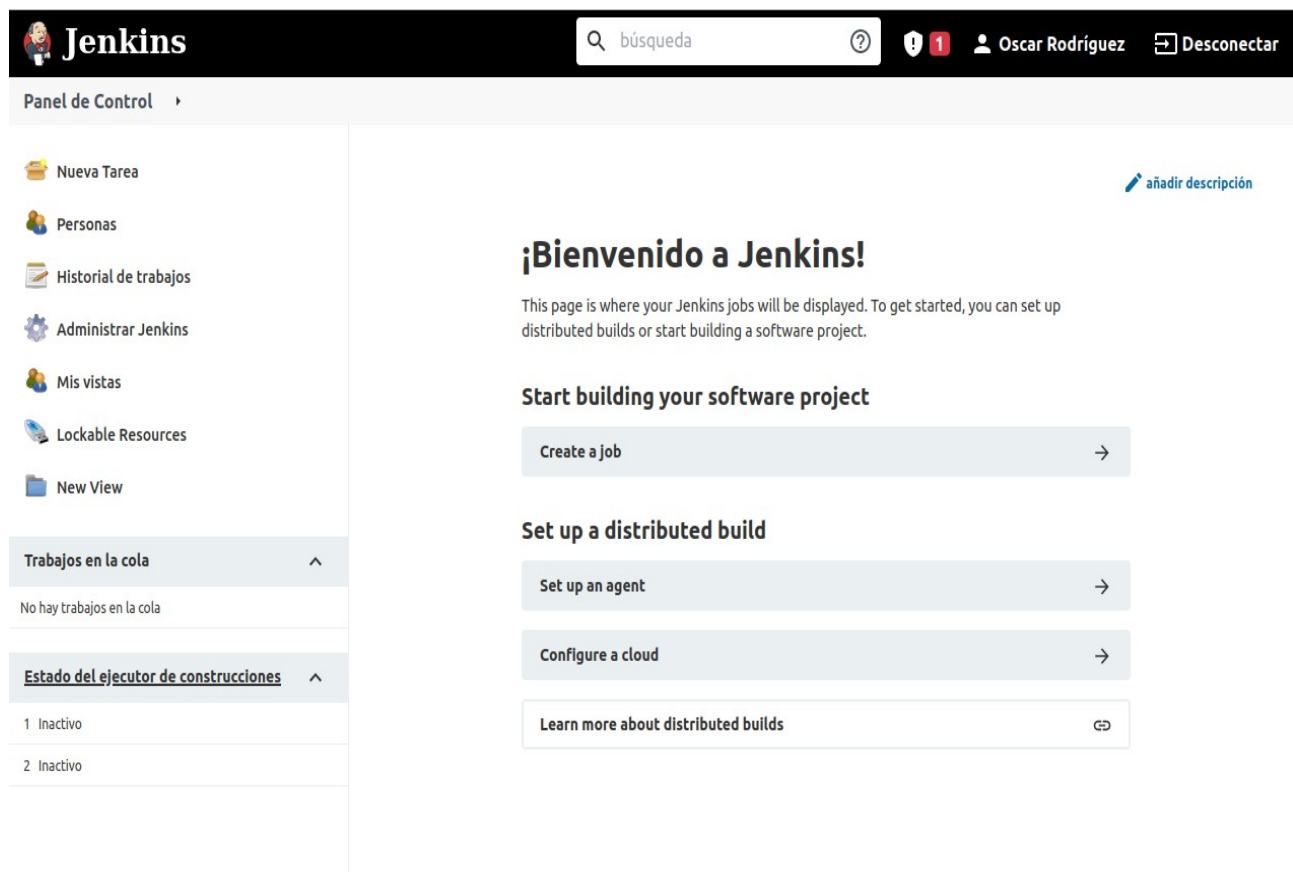
Jenkins

Índice

1. Creación de un Pipeline en Java
2. Creación de un Pipeline en PHP
3. Creación de un Pipeline en Phyton

1. Creación de un Pipeline en Java

Primero abrimos Jenkins en el menú principal y seleccionamos nueva tarea



The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, a help icon, a notification bell with a red '1', the user name 'Oscar Rodríguez', and a 'Desconectar' button. Below the header is a 'Panel de Control' section with a list of links: 'Nueva Tarea', 'Personas', 'Historial de trabajos', 'Administrar Jenkins', 'Mis vistas', 'Lockable Resources', and 'New View'. The main content area is divided into two columns. The left column has a section 'Trabajos en la cola' (Jobs in queue) which is currently empty, and a section 'Estado del ejecutor de construcciones' (Builder status) showing two 'Inactivo' (Inactive) entries. The right column features a large '¡Bienvenido a Jenkins!' (Welcome to Jenkins!) message, followed by instructions on how to get started. Below this, there are three main action buttons: 'Create a job', 'Set up a distributed build', and 'Configure a cloud', each with a right-pointing arrow. At the bottom of the right column is a link 'Learn more about distributed builds' with an external link icon.

Jenkins

búsqueda

Oscar Rodríguez Desconectar

Panel de Control

Nueva Tarea

Personas

Historial de trabajos

Administrar Jenkins

Mis vistas

Lockable Resources

New View

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

1 Inactivo

2 Inactivo

añadir descripción

¡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 →

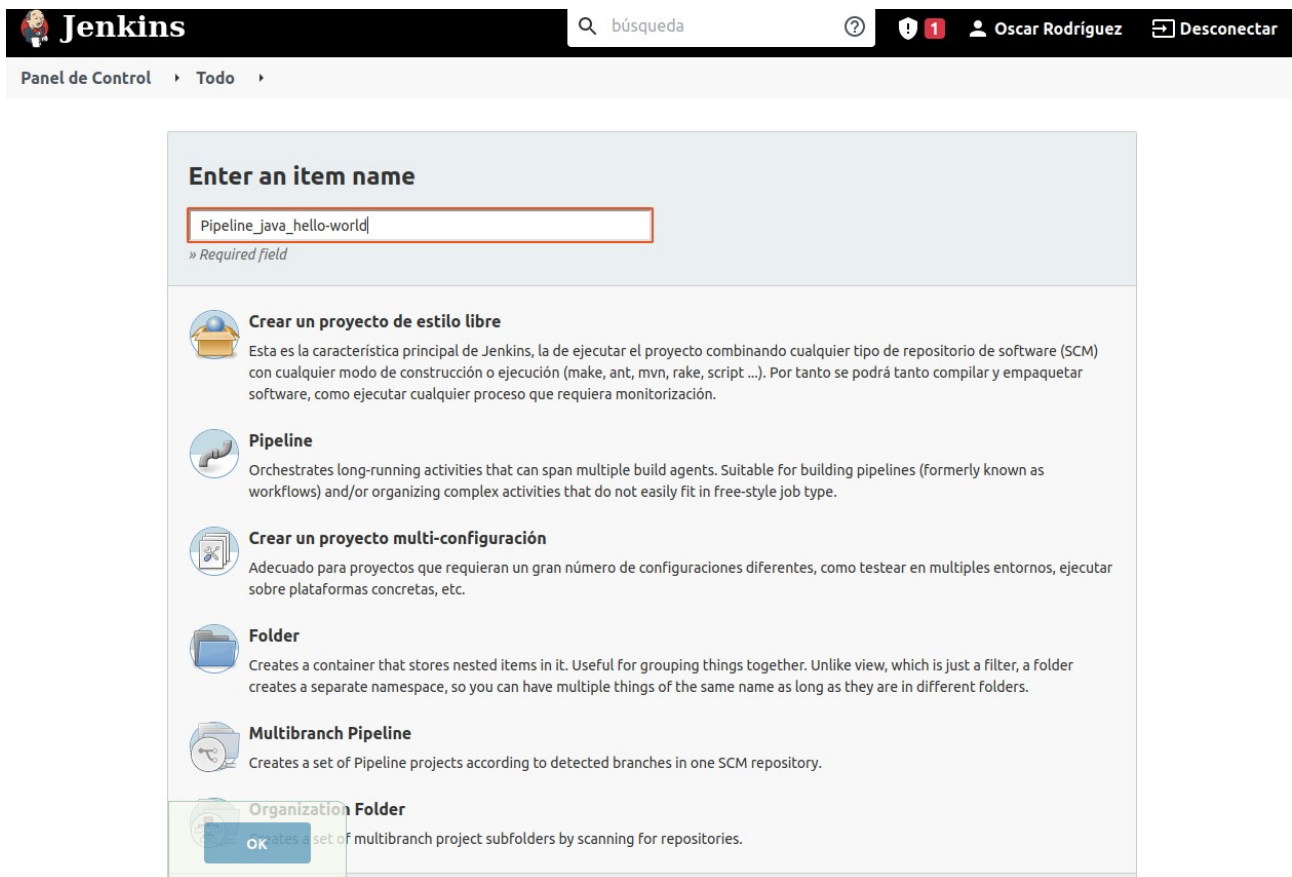
Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

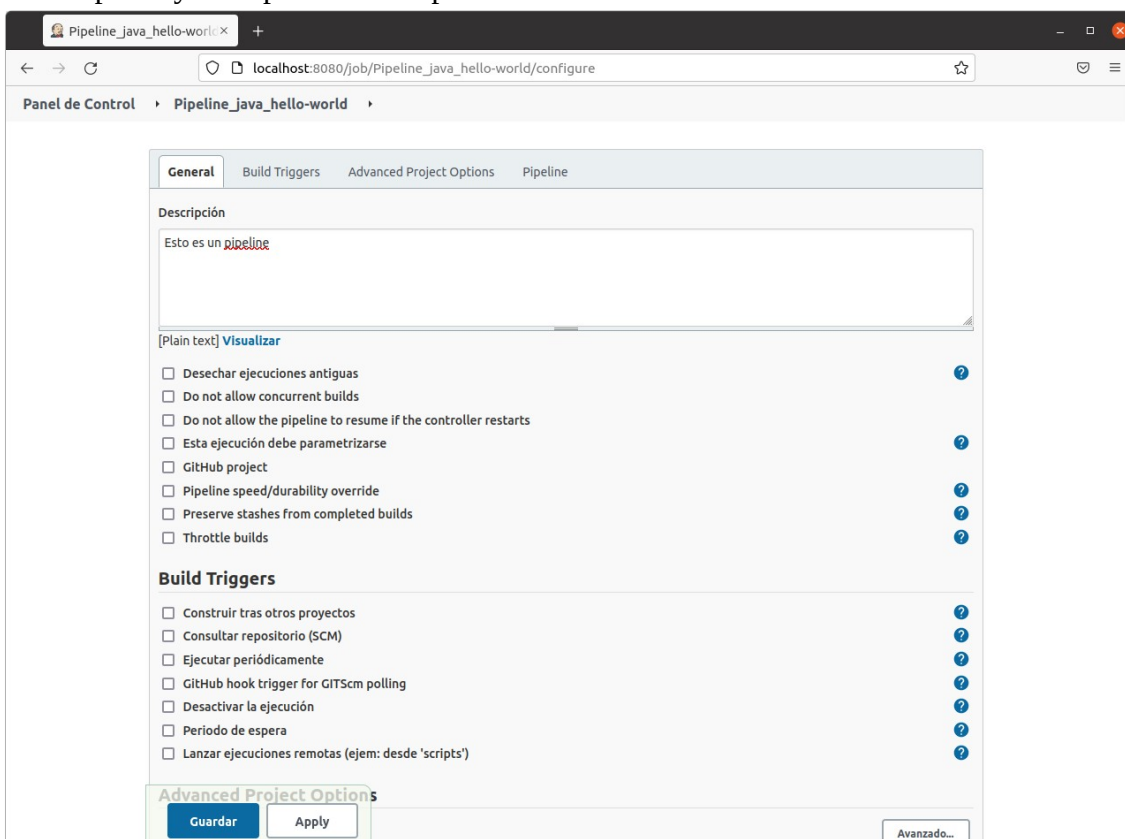
Después escribiremos el nombre del Pipeline, en este caso crearemos un Pipeline en java



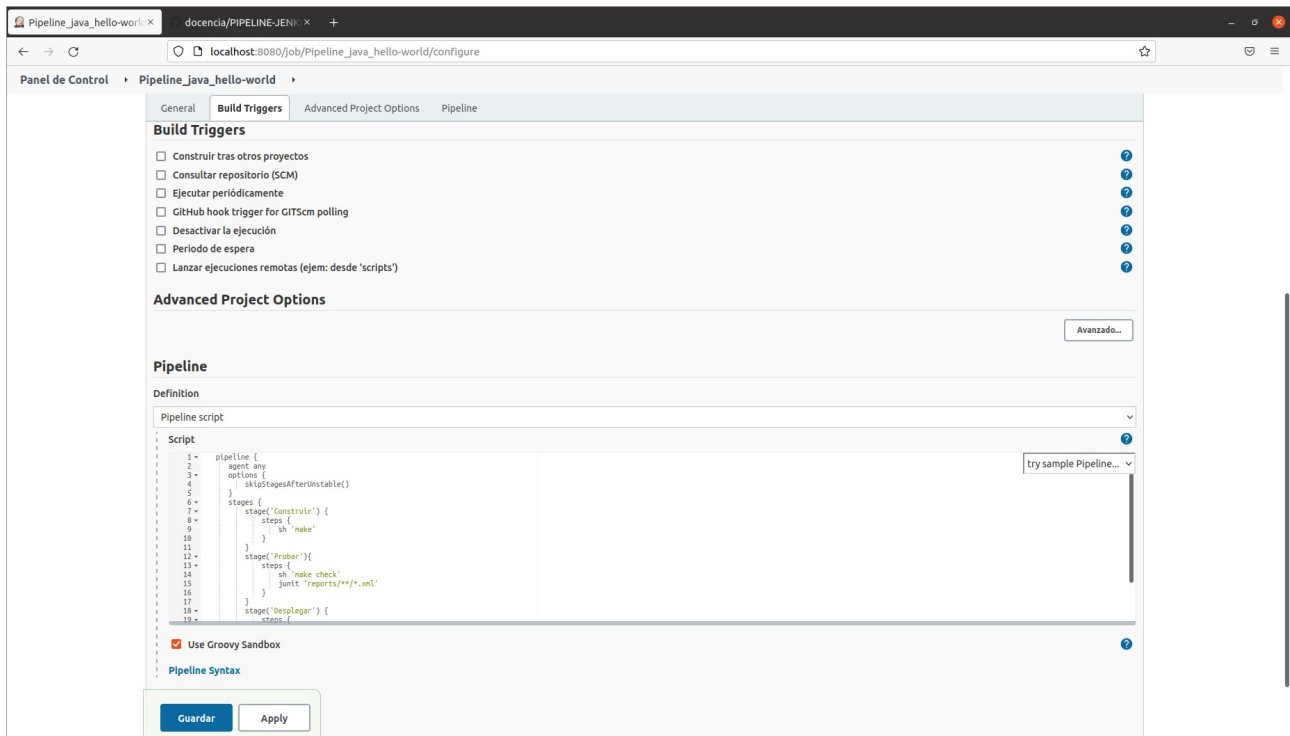
The image shows the Jenkins 'Enter an item name' dialog box. At the top, there's a search bar with the text 'búsqueda'. Below it, the 'Panel de Control' and 'Todo' links are visible. The main section is titled 'Enter an item name' and contains a text input field with the value 'Pipeline_java_hello-world'. Below the input field, there's a link '» Required field'. The dialog also lists several options for creating a new item:

- 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**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- 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.

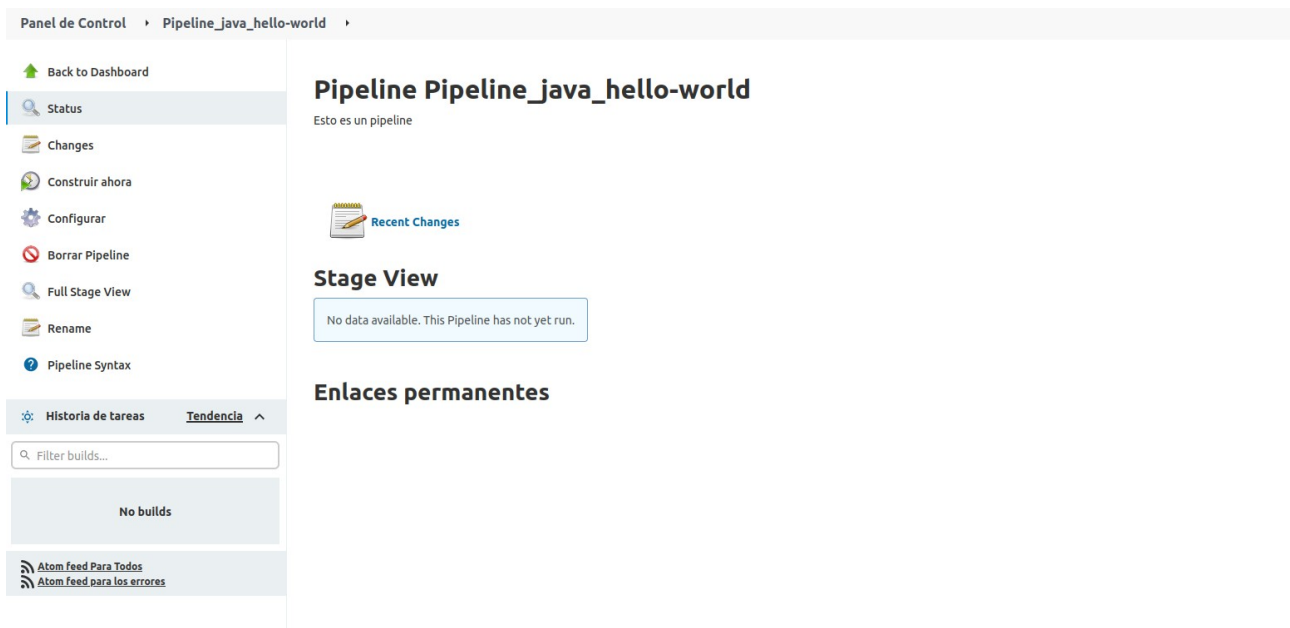
Ahora vamos a configurar nuestro nuevo Pipeline, para ello Primero introduciremos una descripción y en Pipeline el Script.



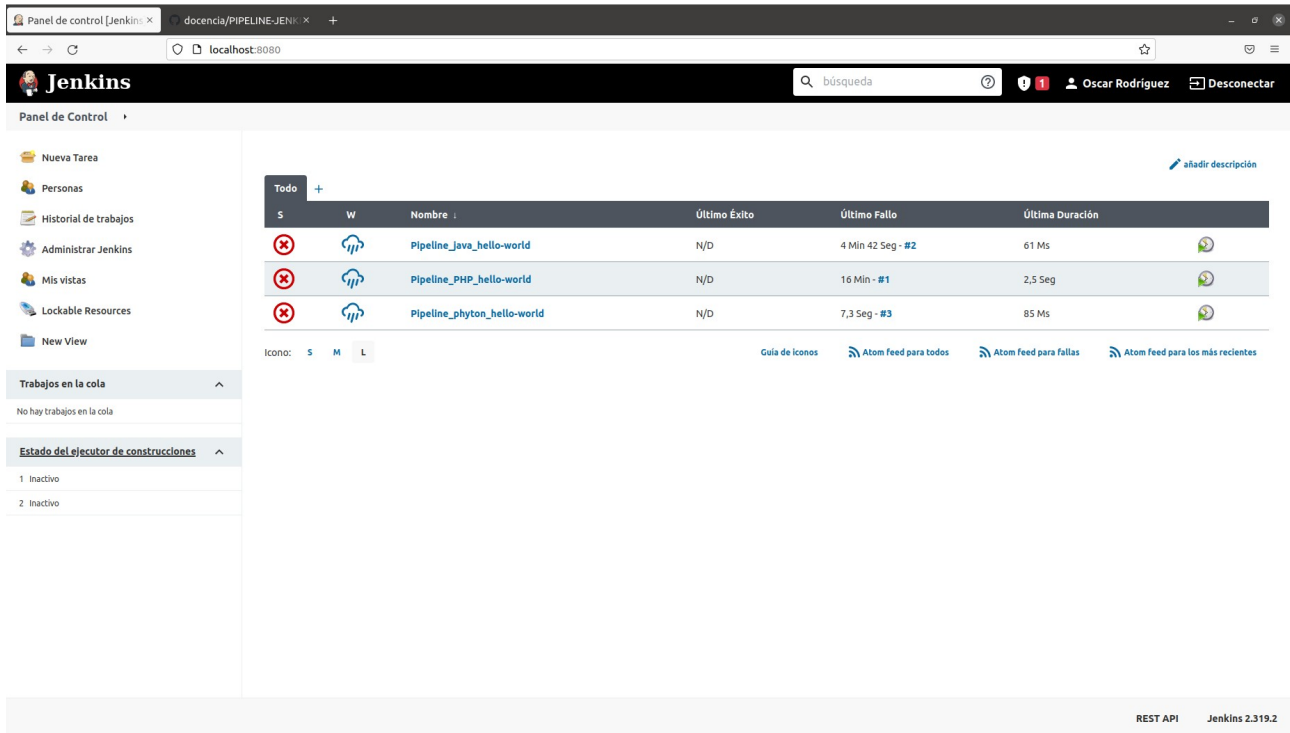
The image shows the Jenkins configuration page for the 'Pipeline_java_hello-world' job. The browser address bar shows 'localhost:8080/job/Pipeline_java_hello-world/configure'. The page has a breadcrumb trail: 'Panel de Control > Pipeline_java_hello-world >'. The configuration is divided into several tabs: 'General', 'Build Triggers', 'Advanced Project Options', and 'Pipeline'. The 'General' tab is active, showing a 'Descripción' field with the text 'Esto es un pipeline'. Below the description field, there's a 'Visualizar' button. A list of checkboxes is shown, including 'Desechar ejecuciones antiguas', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'Esta ejecución debe parametrizarse', 'GitHub project', 'Pipeline speed/durability override', 'Preserve stashes from completed builds', and 'Throttle builds'. The 'Build Triggers' section includes checkboxes for 'Construir tras otros proyectos', 'Consultar repositorio (SCM)', 'Ejecutar periódicamente', 'GitHub hook trigger for GITScm polling', 'Desactivar la ejecución', 'Periodo de espera', and 'Lanzar ejecuciones remotas (ejem: desde 'scripts')'. At the bottom, there are buttons for 'Guardar', 'Apply', and 'Avanzado...'. The 'Advanced Project Options' section is partially visible at the bottom.



Después de aplicar los cambios construiremos nuestro Pipeline en ‘Construir Ahora’



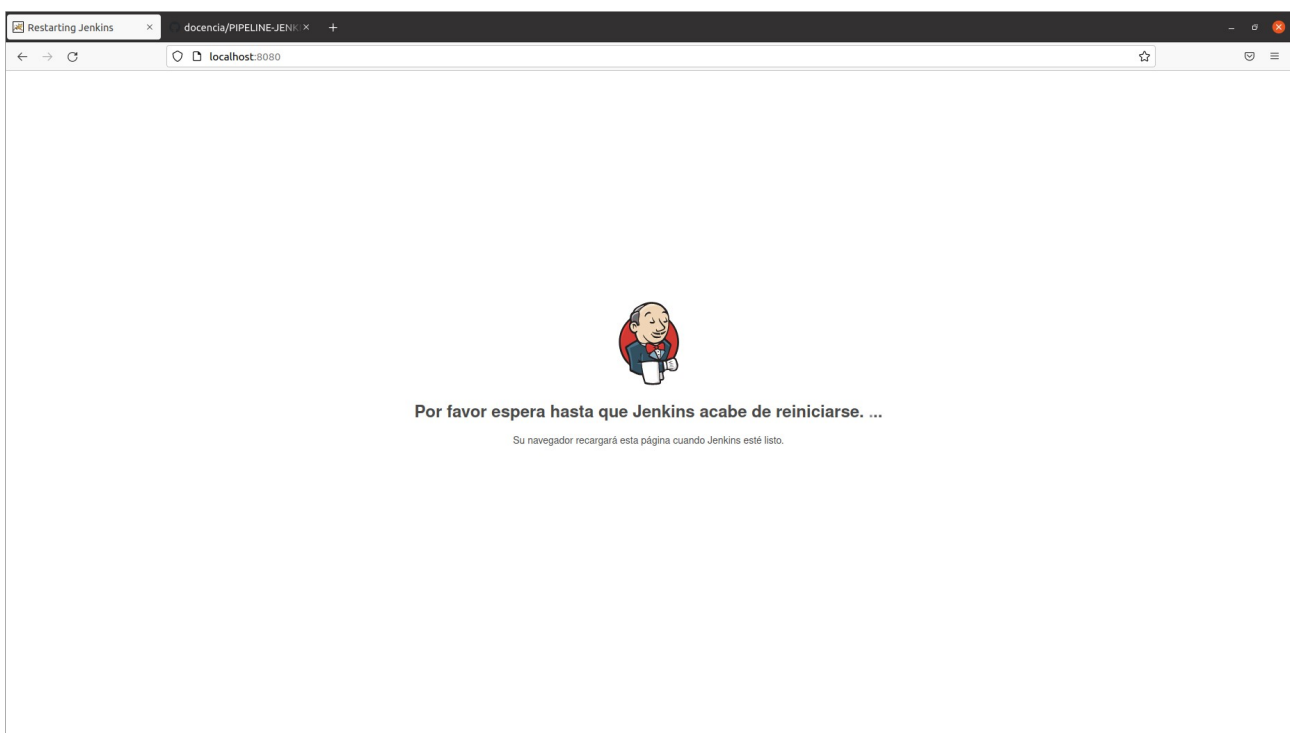
Nos dará un error debido a los plugins, para solucionarlo accederemos a administrar los plugins e instalaremos los plugins Docker common y Docker Pipeline. Cuando se termine de instalar reiniciaremos Jenkins



The screenshot shows the Jenkins web interface at localhost:8080. The left sidebar contains navigation links like 'Nueva Tarea', 'Personas', 'Historial de trabajos', 'Administrar Jenkins', 'Mis vistas', 'Lockable Resources', and 'New View'. The main area displays a table of build jobs under the 'Todo' filter. Three jobs are listed, all with a failed status (red 'X' icon):

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
✗	🔧	Pipeline_java_hello-world	N/D	4 Min 42 Seg - #2	61 Ms
✗	🔧	Pipeline_PHP_hello-world	N/D	16 Min - #1	2,5 Seg
✗	🔧	Pipeline_python_hello-world	N/D	7,3 Seg - #3	85 Ms

Below the table, there are links for 'Gula de iconos', 'Atom feed para todos', 'Atom feed para fallas', and 'Atom feed para los más recientes'. At the bottom right, it says 'REST API' and 'Jenkins 2.319.2'.




The screenshot shows the Jenkins 'Restarting Jenkins' screen. It features the Jenkins logo (a cartoon man in a suit) and the text: 'Por favor espera hasta que Jenkins acabe de reiniciarse. ...'. Below this, a smaller line of text says: 'Su navegador recargará esta página cuando Jenkins esté listo.' The browser's address bar shows 'localhost:8080'.

Una vez actualizado construiremos de nuevo el Pipeline y nos volverá a dar error, esta vez nos dirá que /var/run/docker.sock no tiene permisos entonces vamos a dárselos con el siguiente comando.

```
usuario@ubuntu-20:/var$ sudo chmod 777 run
```

Ahora que ya hemos resuelto los errores vamos construir el Pipeline y como es de esperar nos mostrara una salida correcta de que se ha ejecutado sin problemas.


Salida de consola

Lanzada por el usuario **Oscar Rodríguez**

```
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Pipeline_java_hello-world
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] isUnix
[Pipeline] sh
+ docker inspect -f , maven:3.8.4-openjdk-11-slim

Error: No such object: maven:3.8.4-openjdk-11-slim
[Pipeline] withEnv
[Pipeline] {
[Pipeline] isUnix
[Pipeline] sh
+ docker pull maven:3.8.4-openjdk-11-slim
3.8.4-openjdk-11-slim: Pulling from library/maven
a2abf6c4d29d: Pulling fs layer
2bbde5250315: Pulling fs layer
202a34e7968e: Pulling fs layer
8c484b17211c: Pulling fs layer
abf6a4acf793: Pulling fs layer
ccee672dcde: Pulling fs layer
bfff8cddbdc5e: Pulling fs layer
2cbd6f67a49f: Pulling fs layer
abf6a4acf793: Waiting
ccee672dcde: Waiting
bfff8cddbdc5e: Waiting
8c484b17211c: Waiting
2cbd6f67a49f: Waiting
202a34e7968e: Verifying Checksum
202a34e7968e: Download complete
2bbde5250315: Verifying Checksum
2bbde5250315: Download complete
abf6a4acf793: Verifying Checksum
abf6a4acf793: Download complete
ccee672dcde: Verifying Checksum
ccee672dcde: Download complete
```

Una vez terminada la ejecución comprobaremos que la salida es correcta y mostraremos el estado de la misma.

[añadir descripción](#)

Todo +						
S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración	
✓		Pipeline_java_hello-world	7 Min 14 Seg - #9	22 Min - #8	5 Min 6 Seg	
✗		Pipeline_PHP_hello-world	N/D	40 Min - #3	0,1 Seg	
✗		Pipeline_phyton_hello-world	N/D	40 Min - #5	35 Ms	

Icono: S M L

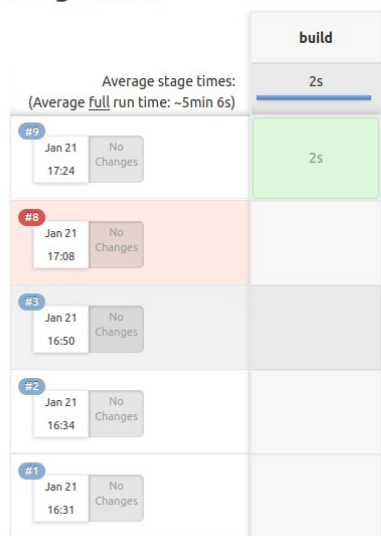
[Guía de iconos](#) [Atom feed para todos](#) [Atom feed para fallas](#) [Atom feed para los más recientes](#)

Pipeline Pipeline_java_hello-world

Esto es un pipeline



Stage View



2. Creacion de un Pipeline en PHP

Habiendo creado el anterior Pipeline, haremos el mismo proceso sin embargo cambiaremos nuestro script y titulo utilizado para PHP. Una vez ejecutado todo comprobaremos que la salida es correcta y el estado sera el siguiente

Pipeline Pipeline_PHP_hello-world

Esto es un pipeline en php



Recent Changes

Stage View



3. Creación de un Pipeline en Phyton

Volvemos a repetir el proceso pero esta vez con phyton

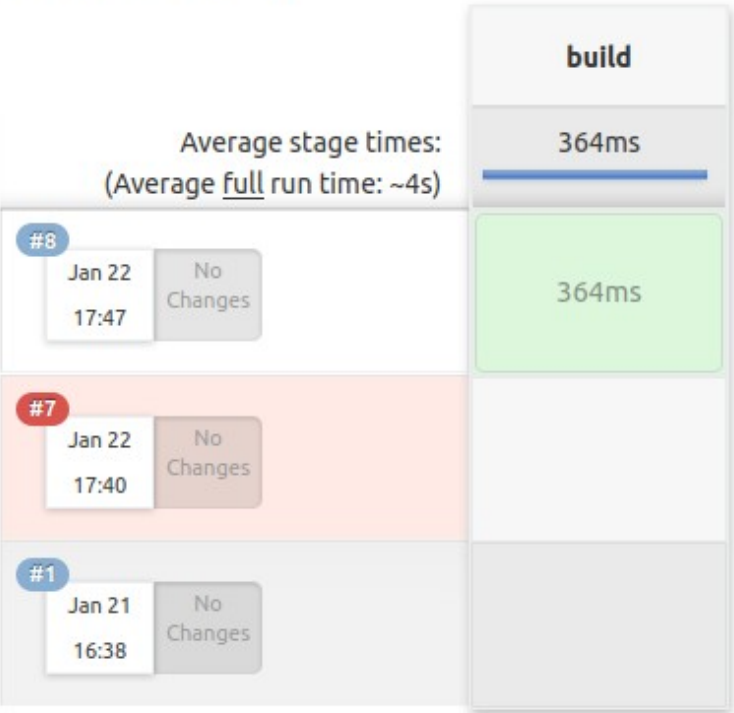
Pipeline Pipeline_phyton_hello-world

Esto es un pipeline en phyton



Recent Changes

Stage View



Una vez construidos todos los Pipeline nos quedaremos con el siguiente resultado

[añadir descripción](#)

Todo	S	W	Nombre	Último Éxito	Último Fallo	Última Duración	
	✓	🔗	Pipeline_Java_hello-world	1 día 0 Hor - #9	1 día 0 Hor - #8	5 Min 6 Seg	🕒
	✓	🔗	Pipeline_PHP_hello-world	1 Min 31 Seg - #8	2 Min 43 Seg - #7	7,1 Seg	🕒
	✓	🔗	Pipeline_phyton_hello-world	32 Seg - #8	6 Min 55 Seg - #7	4,6 Seg	🕒

