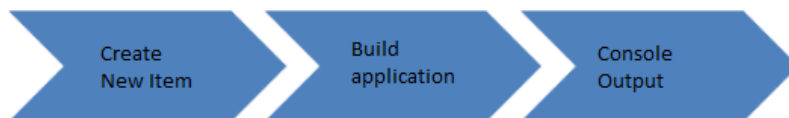


Empleos de Laboratorio Jenkins

¿Qué es un proyecto de Jenkins Free Style?

Jenkins Freestyle Project es un trabajo, script o canalización de compilación repetible que contiene pasos y acciones posteriores a la compilación. Es un trabajo o tarea mejorada que puede abarcar múltiples operaciones. Le permite configurar activadores de compilación y ofrece seguridad basada en proyectos para su proyecto Jenkins. También ofrece complementos para ayudarle a crear pasos y acciones posteriores a la creación.

Los tipos de acciones que pueden realizarse en un paso de compilación de Jenkins o en una acción posterior a la compilación son bastante limitados. Hay muchos complementos estándar disponibles dentro de un proyecto Jenkins Freestyle para ayudarle a superar este problema.



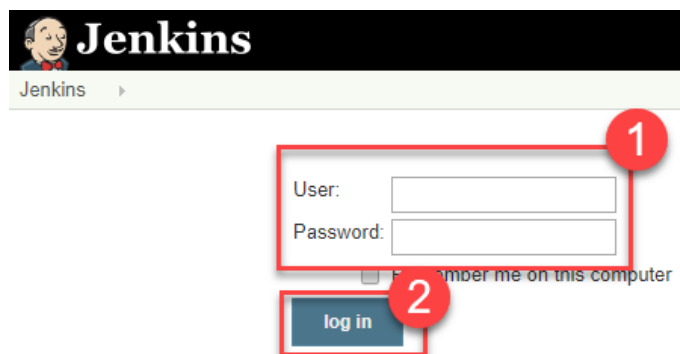
Cómo crear un trabajo en Jenkins

Cómo crear un trabajo de nueva compilación en Jenkins

El trabajo de compilación de estilo libre es una opción muy flexible y fácil de usar. Puedes usarlo para cualquier tipo de proyecto; es fácil de configurar y muchas de sus opciones aparecen en otros trabajos de compilación. A continuación se muestra un proceso paso a paso para crear un trabajo en Jenkins .

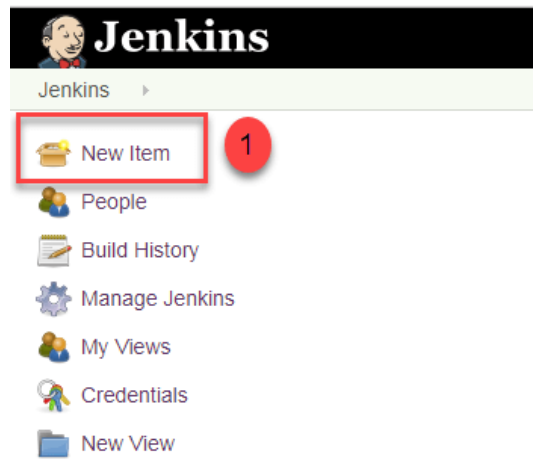
Paso 1) Iniciar sesión en Jenkins

Para crear un trabajo de estilo libre de Jenkins, inicie sesión en su panel de Jenkins visitando la ruta de instalación de Jenkins. Por lo general, estará alojado en localhost en <http://localhost:8080> Si ha instalado Jenkins en otra ruta, use la URL adecuada para acceder a su panel como se muestra en el siguiente ejemplo de creación de trabajo de Jenkins.



Paso 2) Crear nuevo artículo

Haga clic en " **Nuevo artículo** " en la parte superior izquierda de su panel.



Paso 3) Ingrese los detalles del artículo

En la siguiente pantalla,

1. Ingrese el nombre del elemento que desea crear. Usaremos "Hola mundo" para esta demostración.
2. Seleccionar proyecto de estilo libre
3. Haga clic en Aceptar

The image shows the 'Enter an item name' dialog box in Jenkins. At the top, there is a text input field containing 'Hello World', which is highlighted with a red rectangle and a red circle with the number '1'. Below the input field is a list of project types. The 'Freestyle project' option is highlighted with a red rectangle and a red circle with the number '2'. At the bottom of the dialog, there is an 'OK' button, which is highlighted with a red rectangle and a red circle with the number '3'. The dialog box also includes a 'Required field' label next to the input field.

Paso 4) Ingrese los detalles del proyecto

Ingrese los detalles del proyecto que desea probar.

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Description: Hello world java test program

[Plain text] [Preview](#)

- ☐ Discard old builds
- ☐ GitHub project
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

[Advanced...](#)

Paso 5) Ingrese la URL del repositorio

En Administración de código fuente, ingrese la URL de su repositorio. Tenemos un repositorio de pruebas ubicado en <https://github.com/kriru/firstJava.git>

Source Code Management | General | Build Triggers | Build Environment | Build | Post-build Actions

☐ None
☒ Git

Repositories

Repository URL: https://github.com/kriru/firstJava.git

Credentials: - none - [Add](#)

[Advanced...](#)
[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'): */master

[Add Branch](#)

También es posible que utilice un repositorio local.

Si su repositorio de GitHub es privado, Jenkins primero validará sus credenciales de inicio de sesión con GitHub y solo luego extraerá el código fuente de su repositorio de GitHub.

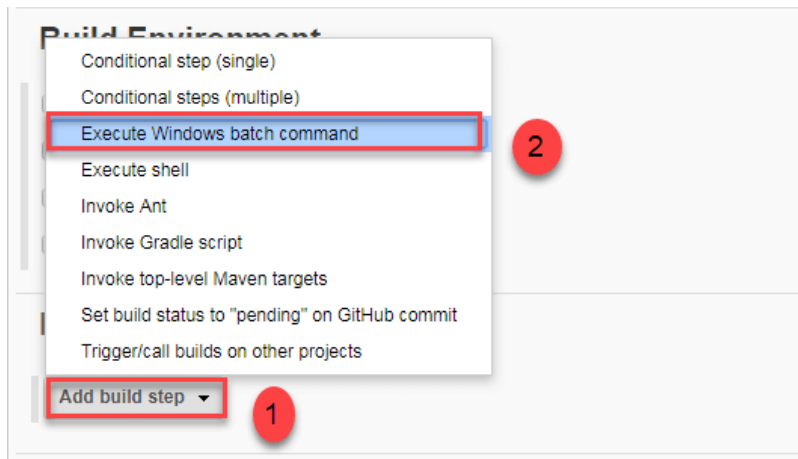
Paso 6) Modificar la configuración

Ahora que ha proporcionado todos los detalles, es hora de crear el código. Modifique la configuración en la sección **de compilación** para compilar el código en el momento que desee. Incluso puedes programar la compilación para que se realice periódicamente, en horarios establecidos.

En **construcción**,

1. Haga clic en " **Agregar paso de compilación** ".

2. Haga clic en " **Ejecutar comando por lotes de Windows** " y agregue los comandos que desee ejecutar durante el proceso de compilación.

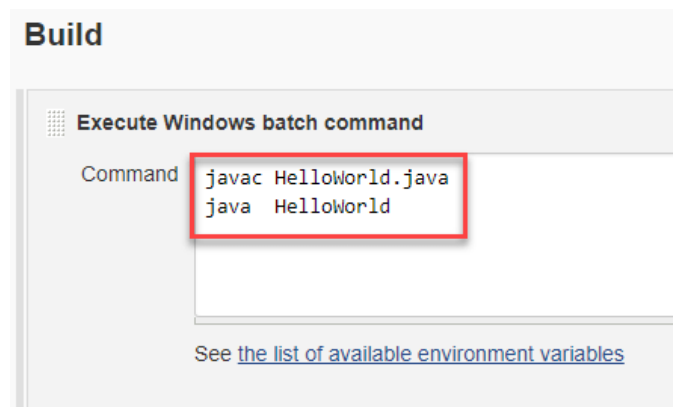


Aquí, agregue los comandos de Java para compilar el código de Java.

Agregó los siguientes comandos de Windows:

```
javac HolaMundo.java
```

```
Hola Mundo
```



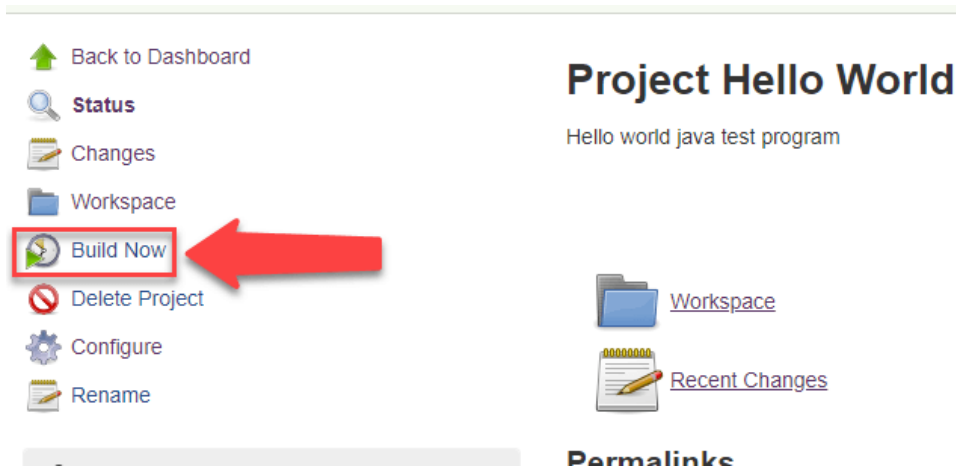
Paso 7) Guarde el proyecto

Cuando hayas introducido todos los datos,

1. Haga clic en **Aplicar**
2. **Guarde** el proyecto.

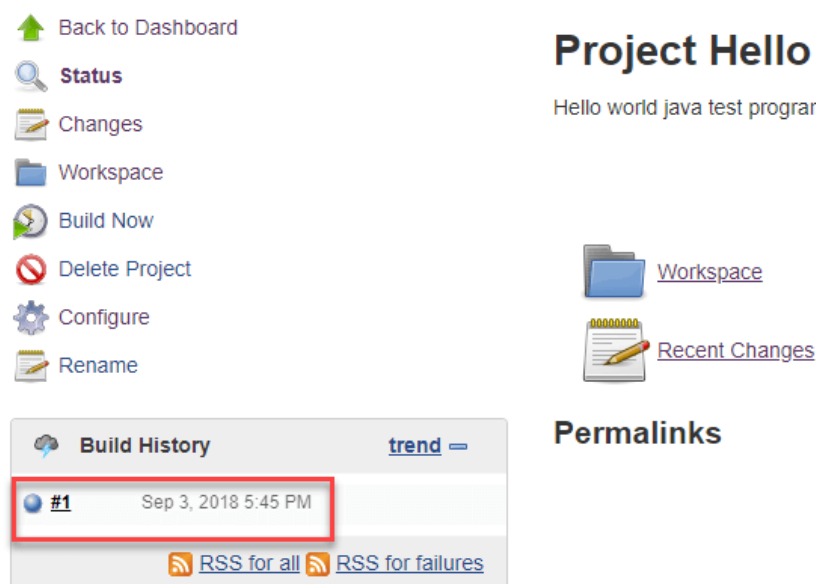
Paso 8) Construir el código fuente

Ahora, en la pantalla principal, haga clic en el botón **Construir ahora** en el lado izquierdo para compilar el código fuente.



Paso 9) Verifique el estado

Después de hacer clic en **Construir ahora**, puede ver el estado de la compilación que se ejecuta en **Historial de compilación**.



Paso 10) Ver la salida de la consola

Haga clic en el **número de compilación** y luego haga clic en **la salida de la consola** para ver el estado de la compilación que ejecuta. Debería mostrarle un mensaje de éxito, siempre que haya seguido la configuración correctamente, como se muestra en el siguiente ejemplo de creación de un nuevo trabajo de Jenkins.

Jenkins > Hello World > #1

- Back to Project
- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete Build
- Next Build

Console Output

```
Started by user The_Guru99
Building in workspace C:\Program Files (x86)\Jenkins\workspace\Hello World
Cloning the remote Git repository
Cloning repository https://github.com/kriru/firstJava.git
> git.exe init C:\Program Files (x86)\Jenkins\workspace\Hello World # timeout=10
Fetching upstream changes from https://github.com/kriru/firstJava.git
> git.exe --version # timeout=10
> git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +refs
> git.exe config remote.origin.url https://github.com/kriru/firstJava.git # ti
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.url https://github.com/kriru/firstJava.git # ti
Fetching upstream changes from https://github.com/kriru/firstJava.git
> git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +refs
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
> git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
> git.exe rev-parse "origin/master^{commit}" # timeout=10

C:\Program Files (x86)\Jenkins\workspace\Hello World>javac HelloWorld.java

C:\Program Files (x86)\Jenkins\workspace\Hello World>java HelloWorld
Hello World

Finished: SUCCESS
```

En resumen, hemos ejecutado un programa HelloWorld alojado en GitHub. Jenkin extrae el código del repositorio remoto y lo compila continuamente con la frecuencia que usted define.