

Crear instancia de Jenkins en EC2

Lanzar una instancia [Información](#)

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas [Información](#)

Nombre [Agregar etiquetas adicionales](#)

▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon) [Información](#) [Lanzar una instancia](#)

Una AMI posee el sistema operativo, el servidor de aplicaciones y las aplicaciones de la instancia. Si a continuación no ve una AMI adecuada, utilice el campo de búsqueda o elija **Buscar más AMI**.

[Recientes](#) [Inicio rápido](#)

 Ubuntu [ubuntu®](#)

 Amazon Linux  macOS  Windows  Red Hat  SUSE Linux  Debian

 [Buscar más AMI](#)
Inclusión de AMI de AWS, Marketplace y la comunidad

Imágenes de máquina de Amazon (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type [Apto para la capa gratuita](#)

ami-0ef0fafba270833fc (64 bits (x86)) / ami-0e92357cce1e2ffe9 (64 bits (Arm))
Virtualización: hvm Activado para ENA: true Tipo de dispositivo raíz: ebs

Descripción

Arquitectura [64 bits \(x86\)](#) **ID de AMI** ami-0ef0fafba270833fc **Fecha de publicación** 2025-10-15 **Nombre de usuario** [ubuntu](#) [Proveedor verificado](#)

Tipo de instancia [Información](#) | [Obtener asesoramiento](#)

Tipo de instancia [t3.medium](#)

t3.medium
Familia: t3 2 vCPU 4 GiB Memoria Generación actual: true
Bajo demanda Windows base precios: 0.064 USD por hora Bajo demanda Linux base precios: 0.0456 USD por hora
Bajo demanda Ubuntu Pro base precios: 0.0491 USD por hora
Bajo demanda RHEL base precios: 0.0744 USD por hora Bajo demanda SUSE base precios: 0.1019 USD por hora

Todas las generaciones [Comparar tipos de instancias](#)

[Se aplican costos adicionales a las AMI con software preinstalado](#)

Par de claves (inicio de sesión) [Información](#)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio [ejemplo-cicd-key](#) [Crear un nuevo par de claves](#)

Configuraciones de red [Información](#)

VPC : **obligatorio** | [Información](#)

vpc-01a8c3c181ec3d071 172.31.0.0/16 [\(predeterminado\)](#)

Subred | [Información](#)

Ponemos esas dos reglas en nuestras reglas de entrada.

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y _-./()#@+=;&!\$*

Descripción - obligatorio | [Información](#)

launch-wizard-9 created 2026-01-03T17:42:01.870Z

Reglas de grupos de seguridad de entrada

▼ Regla del grupo de seguridad 1 (TCP, 22, 79.116.129.114/32) [Eliminar](#)

Tipo	Protocolo	Intervalo de puertos
ssh	TCP	22

Tipo de origen | [Información](#)

Mi IP

Nombre | [Información](#)

Agregue CIDR, lista de prefijos o grupo de s

por ejemplo, SSH para Admin Desktop

79.116.129.114/32 [X](#)

▼ Regla del grupo de seguridad 2 (TCP, 8080, 79.116.129.114/32) [Eliminar](#)

Tipo	Protocolo	Intervalo de puertos
TCP personalizado	TCP	8080

Tipo de origen | [Información](#)

Mi IP

Nombre | [Información](#)

Agregue CIDR, lista de prefijos o grupo de s

por ejemplo, SSH para Admin Desktop

79.116.129.114/32 [X](#)

[Agregar regla del grupo de seguridad](#)

▼ **Configurar almacenamiento** [Información](#) [Avanzado](#)

Configurar almacenamiento [Información](#) [Avanzado](#)

1x 50 GiB gp3 Volumen raíz, 3000 IOPS, No cifrado

ⓘ Los clientes que cumplen los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS [X](#)

[Agregar un nuevo volumen](#)

La AMI seleccionada contiene volúmenes de almacén de instancias; sin embargo, la instancia no permite dichos volúmenes. Por lo que no se podrá obtener acceso a ninguno de estos volúmenes de la AMI desde la instancia

ⓘ Haga clic en actualizar para ver la información de la copia de seguridad Las etiquetas que asigne determinan si alguna política de Data Lifecycle Manager realizará una copia de seguridad de la instancia. [Actualizar](#)

0 x sistemas de archivos [Editar](#)

► **Detalles avanzados** [Información](#)

Ahí vemos ya nuestra IP pública, por ahora es dinámica, si paramos y arrancamos la instancia, cambiaria. Por ahora lo dejamos así.

i-09e651cb902d869e6 (jenkins-ec2)

Detalles | Estado y alarmas | Monitoreo | Seguridad | Redes | Almacenamiento | Etiquetas

Resumen de instancia [Información](#)

ID de la instancia [i-09e651cb902d869e6](#)

Dirección IPv4 pública [54.74.153.59 | dirección abierta](#)

Dirección IPv6 -

Estado de la instancia [Instancias](#) En ejecución

Tipo de nombre de anfitrión Nombre de IP: ip-172-31-31-238.eu-west-1.compute.internal

Responder al nombre DNS de recurso privado IPv4 (A)

Nombre DNS de IP privada (solo IPv4) [Ip-172-31-31-238.eu-west-1.compute.internal](#)

Tipo de instancia [t3.medium](#)

Direcciones IPv4 privadas [172.31.31.238](#)

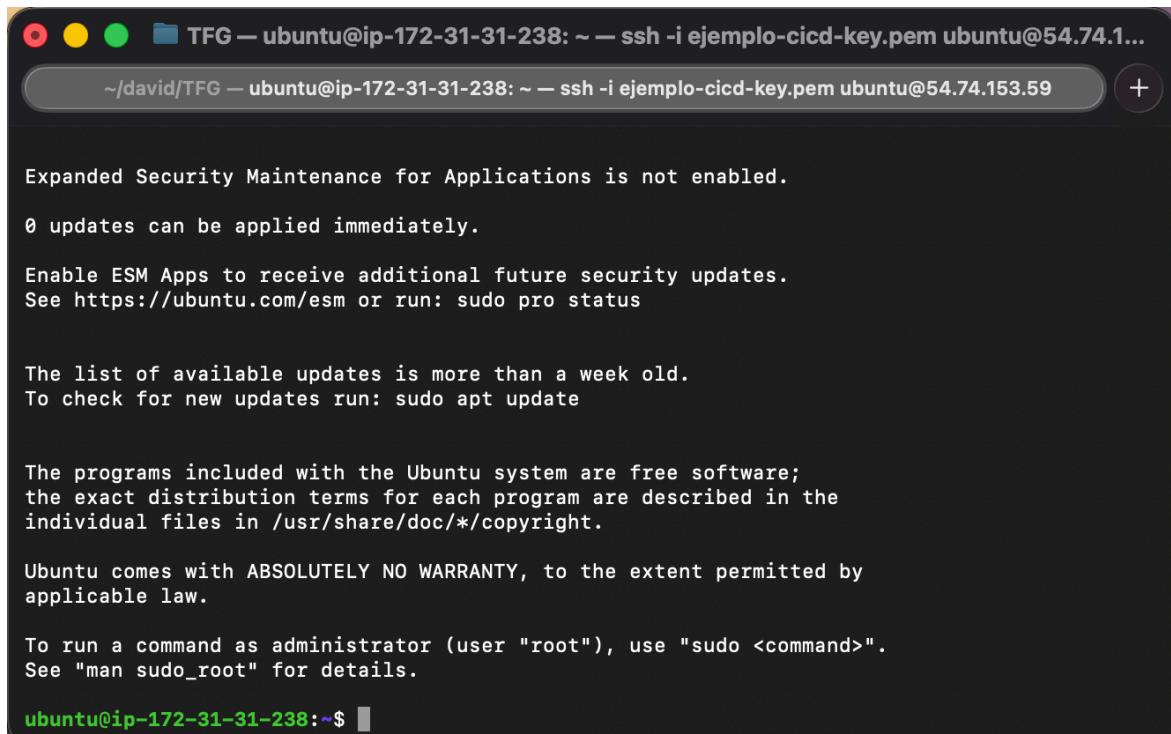
DNS público [ec2-54-74-153-59.eu-west-1.compute.amazonaws.com | dirección abierta](#)

Direcciones IP elásticas -

Entramos a nuestra consola EC2 desde nuestra terminal.

```
~/david/TFG — ubuntu@ip-172-31-41-174: ~ -- zsh — 92x24
(base) david@MacBook-Pro-de-David TFG % ssh -i ejemplo-cicd-key.pem ubuntu@54.74.153.59
```

Ya la tenemos arrancada.



The screenshot shows a terminal window with two tabs. The active tab displays system status information:

```
TFG — ubuntu@ip-172-31-31-238: ~ — ssh -i ejemplo-cicd-key.pem ubuntu@54.74.1...
~/david/TFG — ubuntu@ip-172-31-31-238: ~ — ssh -i ejemplo-cicd-key.pem ubuntu@54.74.153.59 +
```

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-238:~\$ █

Actualizamos el sistema:

```
sudo apt update && sudo apt -y upgrade
```

Instalamos Docker:

```
curl -fsSL https://get.docker.com | sudo sh
```

```
sudo usermod -aG docker ubuntu
```

```
newgrp Docker
```

comprobamos si se ha instalado:

```
docker versión
```

levantamos Jenkins desde Docker:

```
docker volume create jenkins_home
```

```
docker run -d --name jenkins \
-p 8080:8080 -p 50000:50000 \
-v jenkins_home:/var/jenkins_home \
--restart unless-stopped \
jenkins/jenkins:lts
```

si ejecutamos este comando:

```
docker ps
```

y ves algo asi, es que lo tienes bien:

```
[ubuntu@ip-172-31-31-238:~$ docker ps
CONTAINER ID IMAGE COMMAND      CREATED STATUS
PORTS NAMES
c7e22a96b06b jenkins/jenkins:lts "/usr/bin/tini -- /u..." About a minute ago Up About
a minute 0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp, 0.0.0.0:50000->50000/tcp, [::]:50000
->50000/tcp jenkins
ubuntu@ip-172-31-31-238:~$ ]
```

Ahora vamos a ejecutar estos comandos:

Construir una imagen de Jenkins con docker-cli:

```
mkdir -p ~/jenkins-docker && cd ~/jenkins-docker
```

```
nano Dockerfile
```

```
[ docker:x:yyy:ubuntu
ubuntu@ip-172-31-31-238:~$ mkdir -p ~/jenkins-docker && cd ~/jenkins-docker
[nano Dockerfile ]
```

Pegamos esto:

```
FROM jenkins/jenkins:lts
```

```
USER root
```

```
RUN apt-get update && apt-get install -y \
```

```
    ca-certificates \
```

```
    curl \
```

```
    gnupg \
```

```
    lsb-release \
```

```
    python3 \
```

```
    python3-venv \
```

```
    python3-pip \
```

```
&& apt-get clean \
```

```
&& rm -rf /var/lib/apt/lists/*
```

```
RUN install -m 0755 -d /etc/apt/keyrings \
```

```
&& curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg \
```

```
&& chmod a+r /etc/apt/keyrings/docker.gpg \
```

```
&& echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" \
```

```
> /etc/apt/sources.list.d/docker.list \
```

```
&& apt-get update \
```

```
&& apt-get install -y docker-ce-cli \
```

```
&& apt-get clean \
```

```
&& rm -rf /var/lib/apt/lists/*
```

```
USER jenkins
```

Guardamos y cerramos el fichero.

Ahora vamos a construirlo:

```
docker build -t jenkins-docker:lts .
```

```
[root@jenkins-docker ~]# docker build -t jenkins-docker:lts .
[+] Building 19.8s (6/6) FINISHED
   docker:default
```

Parar y borrar el Jenkins actual:

```
docker stop jenkins
```

```
docker rm Jenkins
```

```
[root@jenkins-docker ~]# docker stop jenkins
[root@jenkins-docker ~]# docker rm jenkins
jenkins
jenkins
```

Levantar Jenkins con docker.sock y group-add 999:

```
docker run -d \
```

```
--name jenkins \
```

```
-p 8080:8080 -p 50000:50000 \
```

```
-v jenkins_home:/var/jenkins_home \
```

```
-v /var/run/docker.sock:/var/run/docker.sock \
```

```
--group-add 999 \
```

```
--restart unless-stopped \
```

```
jenkins-docker:lts
```

```
[root@jenkins-docker ~]# docker run -d \
--name jenkins \
-p 8080:8080 -p 50000:50000 \
-v jenkins_home:/var/jenkins_home \
-v /var/run/docker.sock:/var/run/docker.sock \
--group-add 999 \
--restart unless-stopped \
[jenkins-docker:lts
d184d92ce9fd46a9128acfd00bfe270bcf1b0fe0241986832f623f9418993262]
```

Verificamos:

```
docker exec -it jenkins docker version
```

```
docker exec -it jenkins docker ps
```

```
ubuntu@ip-172-31-31-238:~/jenkins-docker$ docker exec -it jenkins docker version
docker exec -it jenkins docker ps
Client: Docker Engine - Community
[+] Version:          29.1.3
    API version:      1.52
    Go version:       go1.25.5
    Git commit:       f52814d
    Built:            Fri Dec 12 14:49:42 2025
    OS/Arch:          linux/amd64
    Context:          default

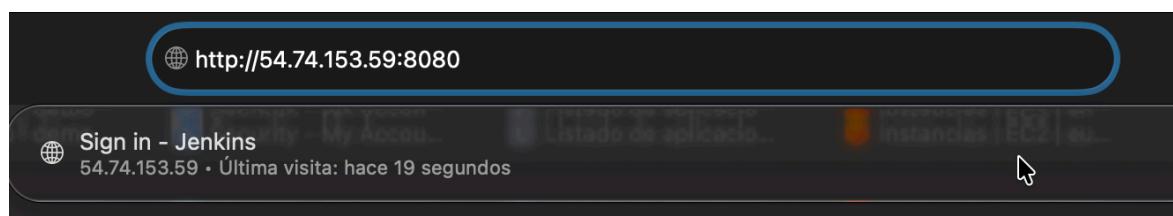
Server: Docker Engine - Community
Engine:
[+] Version:          29.1.3
    API version:      1.52 (minimum version 1.44)
    Go version:       go1.25.5
    Git commit:       fbf3ed2
    Built:            Fri Dec 12 14:49:37 2025
    OS/Arch:          linux/amd64
    Experimental:    false
    containerd:
        Version:        v2.2.1
        GitCommit:      dea7da592f5d1d2b7755e3a161be07f43fad8f75
    runc:
[+] Version:          0.14.0
    Ports:
        NAMES
d184d92ce9fd   jenkins-docker:lts   "/usr/bin/tini -- /u..."   12 seconds ago   Up 11 seconds
ds   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp, 0.0.0.0:50000->50000/tcp, [::]:50000->50000/tcp
0000/tcp   jenkins
ubuntu@ip-172-31-31-238:~/jenkins-docker$
```

Ya tenemos configurado nuestro Docker, vamos ya con la parte de jenkins

Sacamos la password inicial:

```
docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
```

Una vez que tenemos la password, abrimos nuestro navegador y ponemos la ip_publica de la instancia y el puerto.



Escribimos la password que hemos recuperado por terminal

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

*****|

[Continue](#)

Pulsamos a “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 2.528.3

Comienza la instalación.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** JAXB ** SnakeYAML API ** Jakarta Activation API ** Jakarta XML Binding API ** JSON API ** Jackson 2 API ** Pipeline: Supporting APIs ** Plugin Utilities API ** Font Awesome API ** Bootstrap 5 API ** JQuery3 API ** ECharts API ** Prism API ** Display URL API ** Checks API ** JUnit ** Matrix Project ** Resource Disposer Workspace Cleanup Ant ** OkHttp ** Apache HttpComponents Client 4.x API ** Pipeline: Job Gradle ** Pipeline: Milestone Step ** - required dependency
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ LDAP	
⌚ Email Extension	⌚ Mailer	⌚ Dark Theme		

Jenkins 2.528.3

Creamos el usuario root

Getting Started

Create First Admin User

Usuario	<input type="text" value="admin"/>
Contraseña	<input type="password" value="admin"/>
Confirma la contraseña	<input type="password" value="admin"/>
Nombre completo	<input type="text" value="administrador"/>
Dirección de email	<input type="text" value=""/>

Jenkins 2.526.3

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

Aquí lo dejamos como esta y damos a finish.

The screenshot shows the Jenkins 'Instance Configuration' page. At the top left, it says 'Getting Started'. The main title is 'Instance Configuration'. Below that, there's a 'Jenkins URL:' field containing 'http://54.74.153.59:8080/'. A note explains that this URL is required for proper Jenkins operation and should be set to the URL users expect. At the bottom right, there are 'Not now' and 'Save and Finish' buttons.

Pulsamos a la ruleta de ajuste que vemos arriba a la derecha

The screenshot shows the Jenkins dashboard. It features a 'Nueva Tarea' button, a 'Historial de trabajos' link, and sections for 'Trabajos en la cola' (empty) and 'Estado del ejecutor de construcciones' (0/2). The main area displays a 'Bienvenido a Jenkins!' message, a 'Start building your software project' button, and a 'Create a job' button. Below these are sections for 'Set up a distributed build' (with 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds' buttons), a search bar, and Jenkins version information.

Damos a plugins

The screenshot shows the Jenkins System Configuration interface. It includes sections for System, Tools, Clouds, Plugins, and Appearance. The 'Plugins' section is highlighted with a red box.

- System:** Configurar variables globales y rutas.
- Tools:** Configure tools, their locations and automatic installers.
- Clouds:** Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins:** Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins. (This section is highlighted with a red box.)
- Appearance:** Configure the look and feel of Jenkins.

Damos a available plugins y en el navegador buscamos, marcamos e instalamos:

1. **SSH Agent**
2. **Credentials Binding**

Si alguno no aparece, posiblemente ya este instalado.

The screenshots show the Jenkins Plugins management interface.

Screenshot 1: The 'Available plugins' tab is selected. A red arrow points from the 'Available plugins' tab to the search bar. The search bar contains the text 'ssh a'. The results table lists several plugins:

Install	Name	Released	Health
<input type="checkbox"/>	Pipeline Graph Analysis 245.v88f03631a_b_21	Hace 3 Mes 25 días	97
<input type="checkbox"/>	Library plugins (for use by other plugins)	Provides a REST API to access pipeline and pipeline run data.	
<input type="checkbox"/>	PAM Authentication 1.12	Hace 10 Mes	91
<input type="checkbox"/>	Security		
<input type="checkbox"/>	JavaMail API 1.6.2-11		

Screenshot 2: The search bar now contains 'ssh a'. The 'SSH Agent' plugin is selected for installation, indicated by a checked checkbox. A red circle highlights the 'Install' button. A tooltip message says: 'Este plugin está en adopción! Buscamos nuevos mantenedores. Visita nuestro sitio de la iniciativa Adopte un plugin para obtener más información.'

Una vez instalados, reiniciamos Jenkins.



Ahora vamos a esta ruta en nuestro Jenkins:

Administrar Jenkins → Credentials → System → Global credentials (unrestricted) → Add Credentials

Rellenamos los campos así y pegamos nuestro contenido de nuestro .pem que hemos creado en nuestra instancia de AWS.

Jenkins / Administrar Jenkins / Credentials / System / Global credentials (unrestrict... 🔍 ⚙️ 🌐

New credentials

Kind

Scope

ID

Description

Username

Treat username as secret ?

Private Key Enter directly

Create

The screenshot shows the Jenkins Global credentials (unrestricted) configuration page. A private key has been entered directly into the 'Key' field, which contains a long RSA PRIVATE KEY string. The 'Enter directly' radio button is selected under 'Private Key'. There is also a 'Passphrase' field below the key entry.

Para crear el resto de las credenciales en un futuro, las vamos a hacer de la siguiente forma:

Pulsamos a “add credentials”

The screenshot shows the Global credentials (unrestricted) list. A blue button labeled '+ Add Credentials' is highlighted with a red box. Below it, a note says 'Credentials that should be available irrespective of domain specification to requirements matching.'

Y luego:

New credentials

The screenshot shows the 'New credentials' configuration form. It includes fields for 'Kind' (set to 'Secret text'), 'Scope' (set to 'Global (Jenkins, nodes, items, all child items, etc)'), 'Secret' (containing the value 'valor de la credencial'), 'ID' (containing the value 'nombre de la variables'), and 'Description' (empty). A 'Create' button is at the bottom.

Creamos la nueva tarea y seleccionamos esto

The screenshot shows the Jenkins 'Nuevo Tarea' (New Item) creation interface. The 'Enter an item name' field contains 'multiPipeline'. Below it, the 'Select an item type' section lists four options:

- 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 fácilmente con tareas de tipo freestyle.
- Crear un proyecto multi-configuration**: Adecuado para proyectos que requieren un gran número de configuraciones diferentes, como testear en múltiples 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.

The 'Multibranch Pipeline' option is highlighted with a red box. At the bottom right is a blue 'OK' button.

The screenshot shows the 'Configuration' page for the 'multiPipeline' job. The 'General' tab is selected. The 'Display Name' field is set to 'multiPipeline'. The 'Enabled' switch is turned on. The left sidebar shows other configuration tabs: General, Branch Sources, Build Configuration, Scan Multibranch Pipeline Triggers, Orphaned Item Strategy, Appearance, Health metrics, and Properties.

The screenshot shows the 'Branch Sources' configuration page for the 'multiPipeline' job. The 'Git' source is selected. The 'Project Repository' field is set to 'https://gitlab.com/davidtome97/jenkinsjava.git'. The 'Behaviours' section contains 'Discover branches' and 'Discover tags', both of which are highlighted with a red box. The 'Property strategy' dropdown is set to 'All branches get the same properties'. At the bottom are 'Save' and 'Apply' buttons.

Jenkins / multiPipeline / Configuration

Configuration

- General
- Branch Sources
- Build Configuration**
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Build Configuration

Mode: by Jenkinsfile

Script Path: Jenkinsfile

Scan Multibranch Pipeline Triggers

Periodically if not otherwise run

Interval: 5 minutes

Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

Save **Apply**

Jenkins / multiPipeline / Configuration

Configuration

- General
- Branch Sources
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy**
- Appearance
- Health metrics
- Properties

Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

Abort builds

Discard old items

Days to keep old items: if not empty, old items are only kept up to this number of days
7

Max # of old items to keep: if not empty, only up to this number of old items are kept
20

Appearance

Icon: Metadata Folder Icon

Save **Apply**