

PRODUCTO 2. DESPLIEGUE EN ENTORNO DE PRODUCCIÓN

Devops y cloud computing



ADRIÁN JUGUERA AQUILINO
ERIC MOLINER PÉREZ

Índice

La tabla de contenido está vacía porque no estás utilizando los estilos de párrafo que deben aparecer en ella.

Conectarse al Servidor EC2 de AWS

- Iniciaremos sesión en nuestra cuenta de AWS después de registrarnos y lanzaremos una nueva instancia EC2, eligiendo una Amazon Machine Image (AMI), según nuestras preferencias, por ejemplo, **Ubuntu**.

Lanzar una instancia [Información](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Nombre y etiquetas [Información](#)

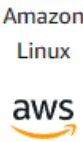

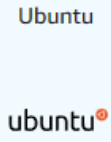




Nombre

[Agregar etiquetas adicionales](#)

▼ Application and OS Images (Amazon Machine Image) [Información](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

						 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 ami-0fe8bec493a81c7da (64-bit (x86)) / ami-0696e474aec8ce817 (64-bit (Arm)) Virtualización: hvm ENA enabled: true Tipo de dispositivo raíz: ebs	Apto para la capa gratuita ▼
--	------------------------------

- Configuramos un par de claves para poder acceder de forma segura a nuestra instancia.

Crear par de claves

✕

Nombre del par de claves
Con los pares de claves es posible conectarse a la instancia de forma segura.

Escriba el nombre del par de claves

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios iniciales ni finales.

Tipo de par de claves



☒ **RSA**
Par de claves pública y privada cifradas mediante RSA

☐ **ED25519**
Par de claves privadas y públicas cifradas ED25519

Formato de archivo de clave privada

☒ **.pem**
Para usar con OpenSSH

☐ **.ppk**
Para usar con PuTTY

 Cuando se le solicite, almacene la clave privada en un lugar seguro y accesible del equipo. **Lo necesitará más adelante para conectarse a la instancia.** [Más información](#) 

Cancelar

Crear par de claves

- Configuramos los ajustes de red permitiendo las conexiones ssh para poder interactuar con la instancia y también el tráfico http y https ya que vamos a configurar un servidor web.

▼ Configuraciones de red [Información](#)

Editar

Red [Información](#)

vpc-03a8f3a9faa1a79e1

Subred [Información](#)

Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública [Información](#)

Habilitar

Firewall (security groups) [Información](#)

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Cualquier lugar
0.0.0.0/0

☒ Permitir el tráfico de HTTPS desde Internet

Para configurar un punto de enlace, por ejemplo, al crear un servidor web

☒ Permitir el tráfico de HTTP desde Internet

Para configurar un punto de enlace, por ejemplo, al crear un servidor web

- Por último, revisamos el resumen y lanzaremos la instancia:

▼ Resumen

Número de instancias [Información](#)

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-0fe8bec493a81c7da

Virtual server type (instance type)

t3.micro

Firewall (security group)

Nuevo grupo de seguridad

Storage (volumes)

1 volume(s) - 9 GiB

Cancelar

Lanzar instancia

[Revisar comandos](#)

- Una vez lanzada, nos podemos conectar a ella mediante diversos métodos. En este caso va a ser a través de ssh.

```
ubuntu@ip-172-31-25-200: ~  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Wed Nov 15 21:10:06 UTC 2023  
  
System load:  0.0          Processes:            103  
Usage of /:   18.3% of 8.54GB Users logged in:        0  
Memory usage: 21%          IPv4 address for ens5: 172.31.25.200  
Swap usage:   0%  
  
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  
  
Last login: Wed Nov 15 21:06:34 2023 from 13.48.4.202  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-25-200:~$
```

Instalación de Jenkins

- En primer lugar, vamos a actualizar el sistema mediante la herramienta apt-get con los comandos **sudo apt-get update** y **sudo apt-get upgrade**.

A continuación seguiremos las instrucciones de la página oficial de Jenkins para instalaciones en Debian y Ubuntu que pueden consultarse en [este enlace](#).

- Una vez instalado, activamos el servicio y lo configuramos para que

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

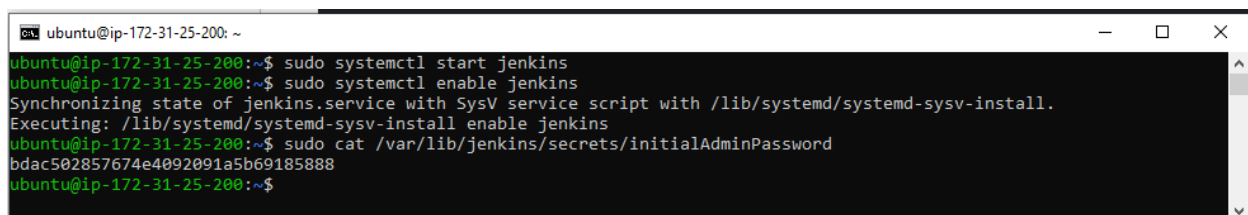
Update your local package index, then finally install Jenkins:

```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

The apt packages were signed using this key:

```
pub  rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
     63667EE74BBA1F0A08A698725BA31D57EF5975CA
uid                Jenkins Project
sub  rsa4096 2023-03-27 [E] [expires: 2026-03-26]
```

se inicie automáticamente con el sistema operativo. También aprovechamos para obtener la contraseña inicial, almacenada en la ruta */var/lib/jenkins/secrets/initialAdminPassword*.



```
ubuntu@ip-172-31-25-200: ~
ubuntu@ip-172-31-25-200:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-25-200:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-25-200:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
bdac502857674e4092091a5b69185888
ubuntu@ip-172-31-25-200:~$
```

- Antes de tratar de acceder al panel de administración de Jenkins, nos aseguramos de haber permitido el tráfico TCP por el puerto 8080, que es el que emplea Jenkins de forma predeterminada, y nos conectamos con la dirección ip pública de nuestra instancia.

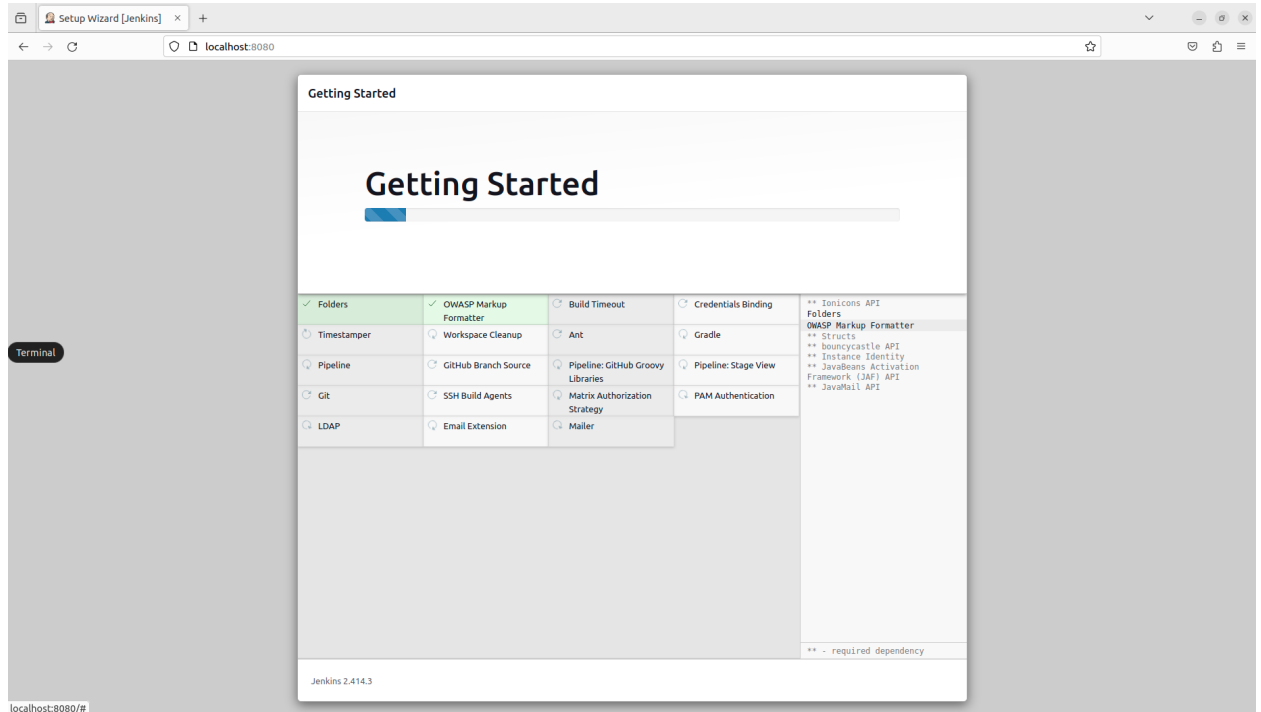
Reglas de entrada (4)

Search

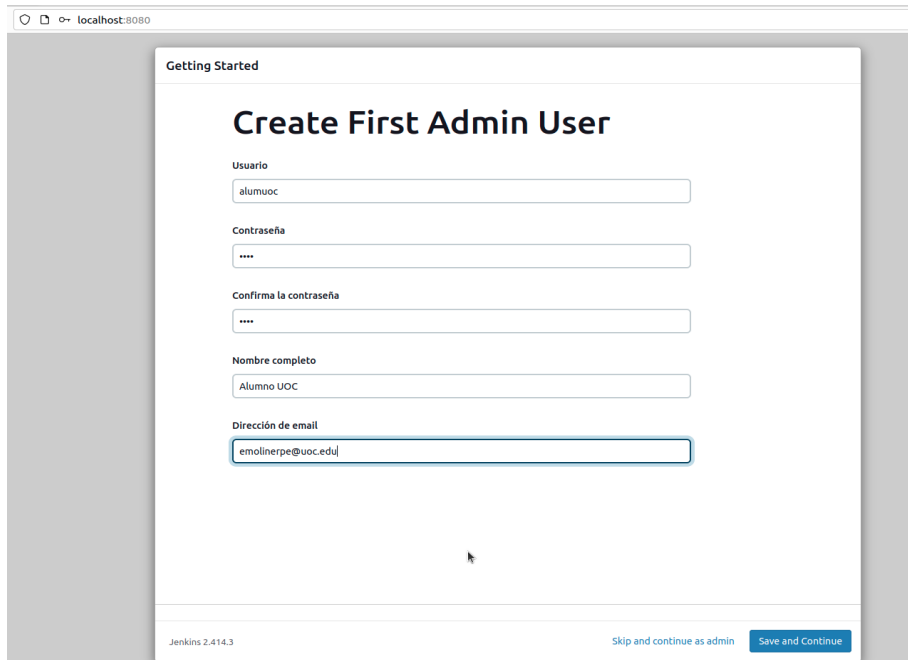
Administrar etiquetas Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción
<input type="checkbox"/>	-	sgr-0b2d0659b9f8bda...	IPv4	HTTP5	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-09807653201d2c...	IPv4	TCP personalizado	TCP	8080	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0d9d149371e36b...	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-009da699c997352...	IPv4	SSH	TCP	22	0.0.0.0/0	-

- Al entrar por primera vez nos pedirá la contraseña de administrador que hemos obtenido antes y nos preguntará si queremos instalar los plugins recomendados por la comunidad o seleccionarlos. Escogemos los seleccionamos por la comunidad, ya que entre ellos está el que nos va a permitir la integración con GitHub, y esperamos que termine de configurar.

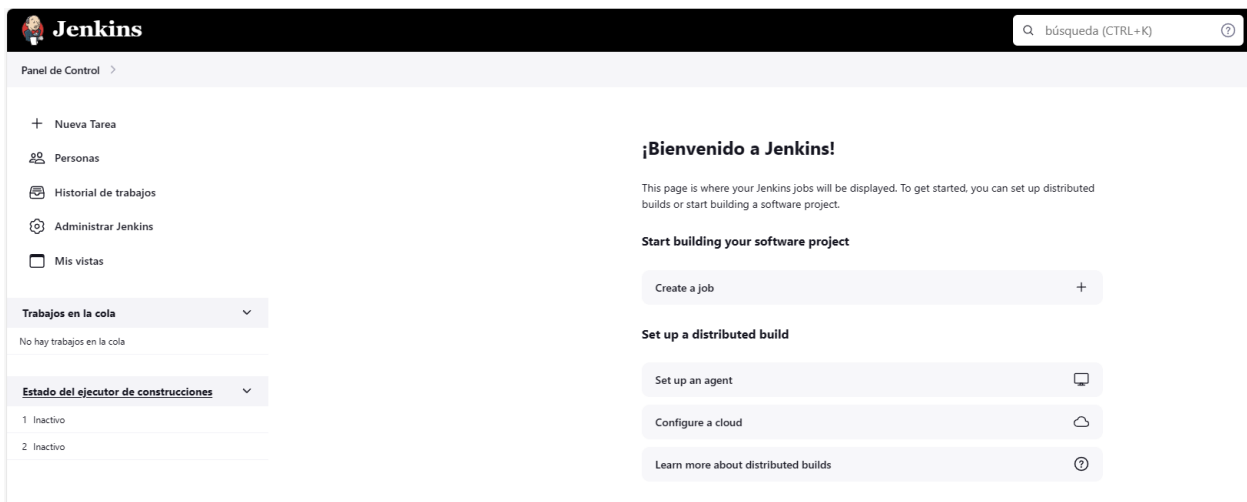


- Después nos pedirá crear una cuenta de administrador y ya podremos acceder al panel de control de Jenkins.



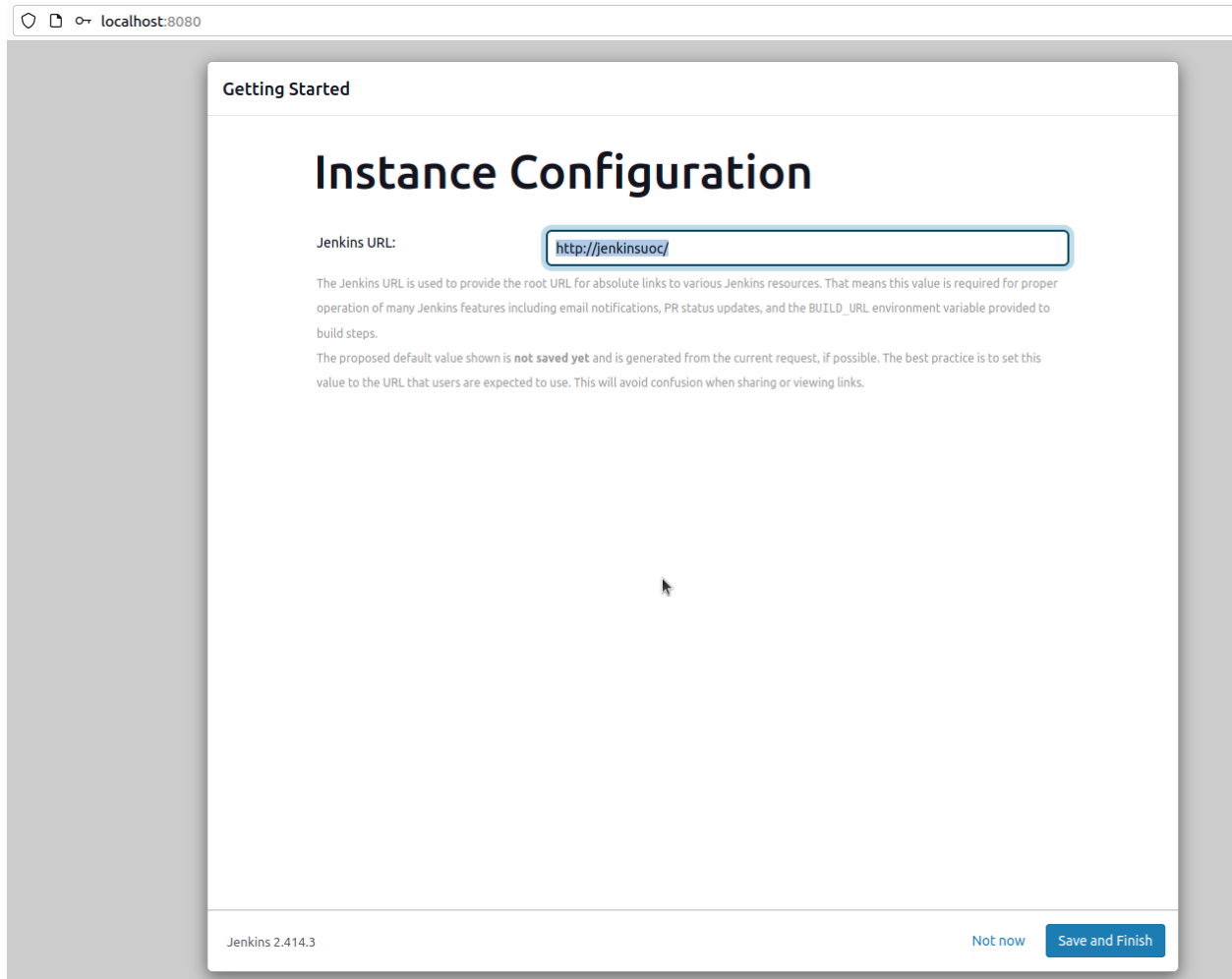
The screenshot shows the Jenkins 'Getting Started' page with the 'Create First Admin User' form. The form fields are: Usuario (alumuoc), Contraseña (masked with dots), Confirma la contraseña (masked with dots), Nombre completo (Alumno UOC), and Dirección de email (emolinerpe@uoc.edu). At the bottom, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.

- Una vez termine de instalarse configuraremos un user admin



The screenshot shows the Jenkins 'Panel de Control' (Dashboard) page. The left sidebar contains navigation links: Nueva Tarea, Personas, Historial de trabajos, Administrar Jenkins, and Mis vistas. The main content area displays '¡Bienvenido a Jenkins!' and provides instructions on how to start building software projects. It includes buttons for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

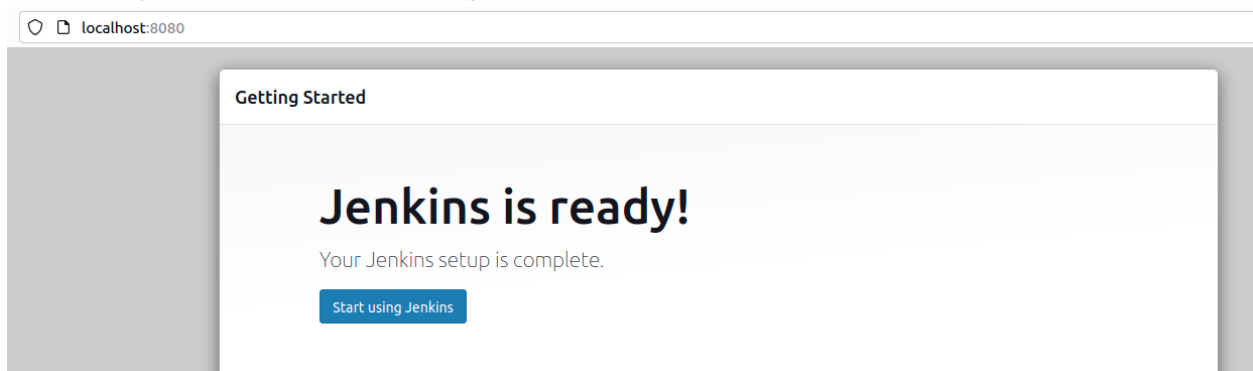
- Ahora configuraremos la URL de Jenkins: <http://jenkinsuoc/>



The screenshot shows the Jenkins 'Getting Started' page for 'Instance Configuration'. The browser address bar shows 'localhost:8080'. The page title is 'Getting Started' and the main heading is 'Instance Configuration'. Under 'Jenkins URL:', there is a text input field containing 'http://jenkinsuoc/'. Below the input field, there is explanatory text: 'The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps. The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.'

At the bottom left, it says 'Jenkins 2.414.3'. At the bottom right, there are two buttons: 'Not now' and 'Save and Finish'.

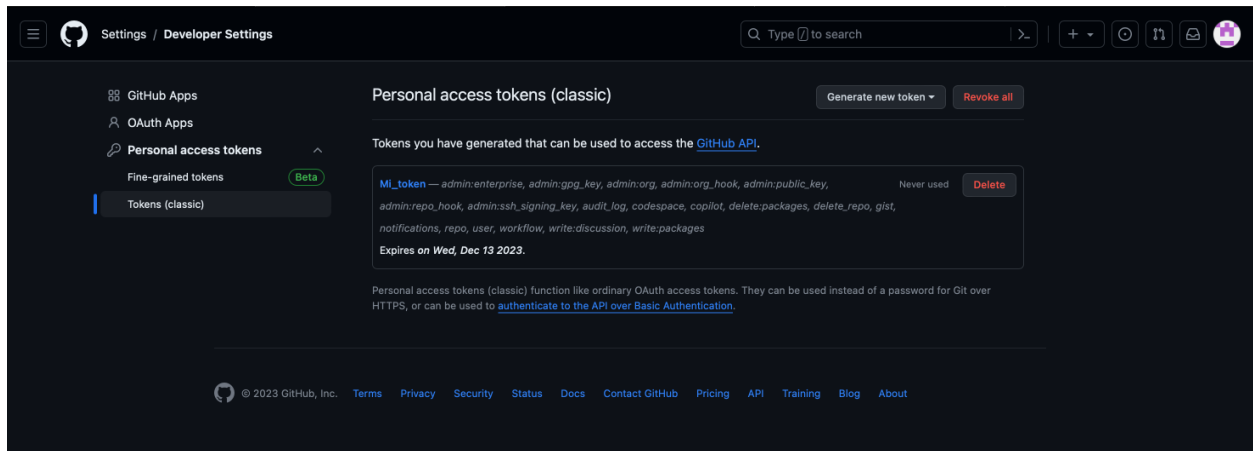
- Y ya tendríamos configurado Jenkins



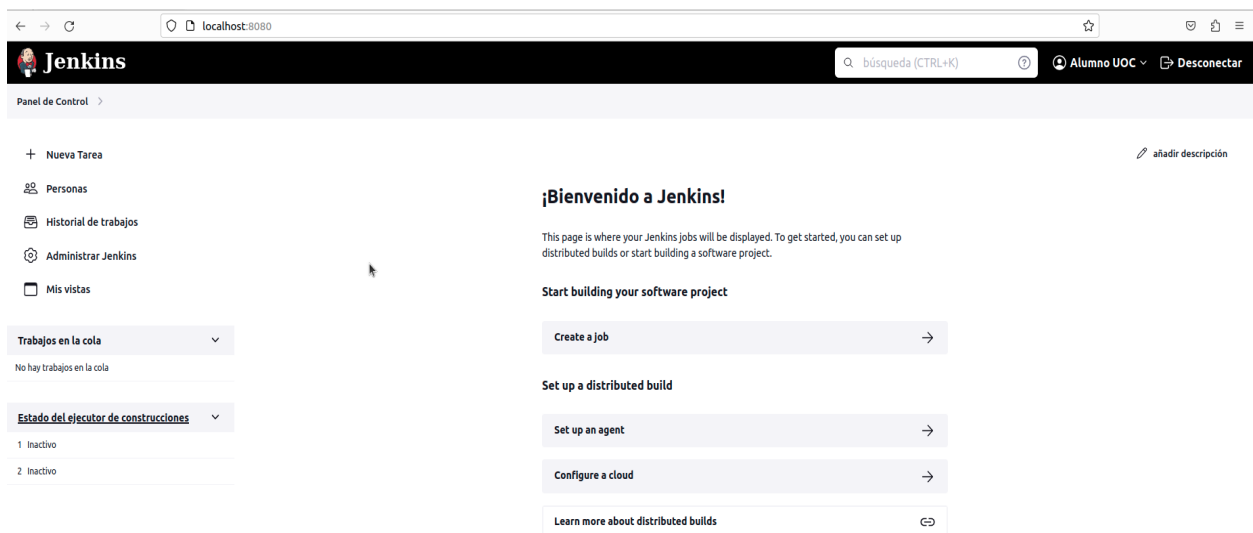
Configuración de Jenkins para despliegue automático desde GitHub:

Token Git Hub

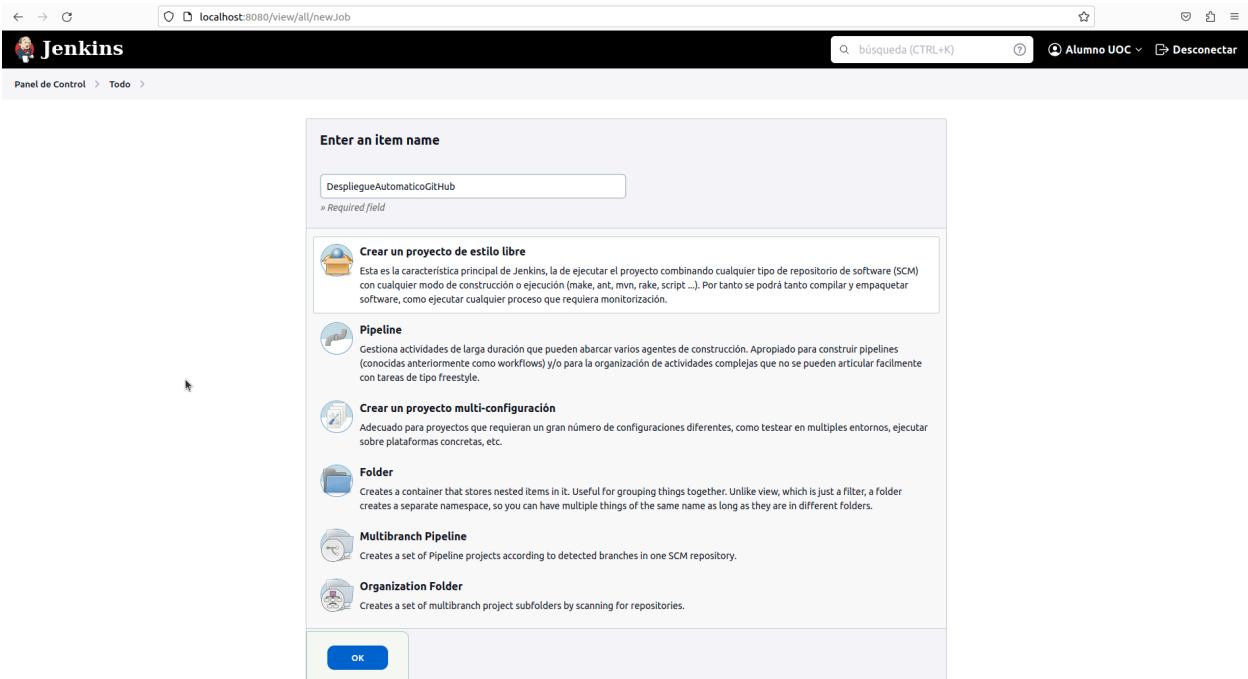
- Primero generamos el token de github.
- Para crear un token en git hub vamos a la configuración de nuestra cuenta, a developer setting, personal acces token y nos aparecer esta pestaña



- Aquí crearemos nuestro token, nos generará un código que posteriormente añadiremos en jenkins
- Una vez tenemos el token vamos a crear una tarea en Jenkins



- Escribiremos un nombre y seleccionamos crear un proyecto de estilo libre:



Enter an item name

DespliegueAutomaticoGitHub

* 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.

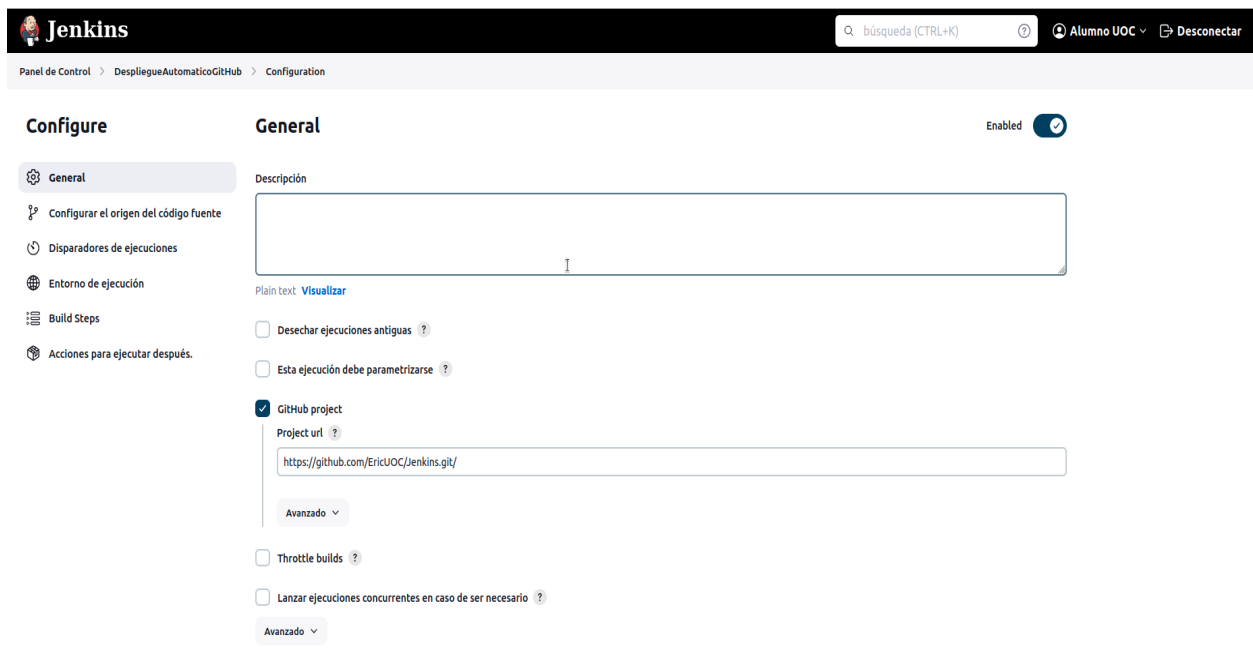
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

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

- Dentro de la configuración seleccionamos el proyecto de github



Configure

General

Configurar el origen del código fuente

Disparadores de ejecuciones

Entorno de ejecución

Build Steps

Acciones para ejecutar después.

General

Enabled

Descripción

Plain text: Visualizar

☐ Desechar ejecuciones antiguas ?

☐ Esta ejecución debe parametrizarse ?

☒ GitHub project

Project url ?

https://github.com/EricUOC/Jenkins.git

Avanzado

☐ Throttle builds ?

☐ Lanzar ejecuciones concurrentes en caso de ser necesario ?

Avanzado

- En origen de código seleccionamos nuestro repositorio de github, también tendremos que añadir nuestras credenciales

Configurar el origen del código fuente

☐ Ninguno

☒ Git ?

Repositories ?

Repository URL ? ✕

`https://ghp_0q4eGhr0NWkHuWk8SzruaZjTBQ1dcD3A9gNm@github.com/EricUOC/Jenkins.git`

Credentials ?

- none - ▼

Add ▼

Avanzado ▼

 **Jenkins** Q búsqueda (CTRL+K) ? Alumno UOC ▼ Desconectar

Panel de Control > Administrar Jenkins > Credenciales > System > Global credentials (unrestricted) >

New credentials

Kind

Secret text ▼

Scope ?

Global (Jenkins, nodes, items, all child items, etc) ▼

Secret

.....

ID ?

TokenGitHub

Description ?

Token de git hub

Create

- En disparador de ejecución seleccionamos GitHub hook trigger form GITScm polling

Disparadores de ejecuciones

- ☐ Lanzar ejecuciones remotas (ejem: desde 'scripts') ?
- ☐ Construir tras otros proyectos ?
- ☐ Consultar repositorio (SCM) ?
- ☐ Ejecutar periódicamente ?
- ☒ GitHub hook trigger for GITScm polling ?

Instalar Git

- Para que funcione necesitaremos instalar git en nuestro servidor:

```
eric@eric:~$ sudo apt install git
[sudo] contraseña para eric:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  git git-man liberror-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 4.203 kB de archivos.
Se utilizarán 20,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [Y/n] s
Des:1 http://es.ports.ubuntu.com/ubuntu-ports jammy/main arm64 liberror-perl all 0.17029-1 [26,5 kB]
Des:2 http://es.ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Des:3 http://es.ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 git arm64 1:2.34.1-1ubuntu1.10 [3.223 kB]
Descargados 4.203 kB en 1s (6.594 kB/s)
Seleccionando el paquete liberror-perl previamente no seleccionado.
(Leyendo la base de datos ... 173167 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../liberror-perl_0.17029-1_all.deb ...
Desempaquetando liberror-perl (0.17029-1) ...
Seleccionando el paquete git-man previamente no seleccionado.
Preparando para desempaquetar .../git-man_1%3a2.34.1-1ubuntu1.10_all.deb ...
Desempaquetando git-man (1:2.34.1-1ubuntu1.10) ...
Seleccionando el paquete git previamente no seleccionado.
Preparando para desempaquetar .../git_1%3a2.34.1-1ubuntu1.10_arm64.deb ...
Desempaquetando git (1:2.34.1-1ubuntu1.10) ...
Configurando liberror-perl (0.17029-1) ...
Configurando git-man (1:2.34.1-1ubuntu1.10) ...
Configurando git (1:2.34.1-1ubuntu1.10) ...
Procesando disparadores para man-db (2.10.2-1) ...
eric@eric:~$ git --version
git version 2.34.1
```

Minikube

- Minikube utiliza Docker para ejecutar los contenedores.

Instalación MiniKube

- Primero actualizamos el sistema con *sudo apt update*.
- Después ejecutamos *snap install kubectl*, como no podíamos instalarlo desde apt, hemos utilizado el repositorio de snap

```
eric@eric:~$ sudo apt-get update && sudo apt-get install -y kubectl
Obj:1 http://es.ports.ubuntu.com/ubuntu-ports jammy InRelease
Des:2 http://es.ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Obj:4 https://pkg.jenkins.io/debian-stable binary/ Release
Des:5 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Err:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg
  Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
Obj:7 http://es.ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Des:8 http://ports.ubuntu.com/ubuntu-ports jammy-proposed InRelease [270 kB]
Descargados 499 kB en 1s (872 kB/s)
Leyendo lista de paquetes... Hecho
W: Se produjo un error durante la verificación de las firmas. El repositorio no está actualizado y se utilizarán los ficheros de índice antiguos. Error de GPG: https://pkg.jenkins.io/debian-stable binary/
  Release: Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
W: Fallo al obtener https://pkg.jenkins.io/debian-stable/binary/Release.gpg Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
W: No se han podido descargar algunos archivos de índice, se han omitido, o se han utilizado unos antiguos en su lugar.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete kubectl
eric@eric:~$ sudo apt install kubectl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho

No hay un paquete apt "kubectl", pero hay un snap con ese nombre.
Intente «snap install kubectl»

E: No se ha podido localizar el paquete kubectl
Papelera : $ snao install kubectl
«snao» no encontrada. Quizá quiso decir:
  la orden «snap» del paquete deb «snapd (2.58+22.04.1)»
Pruebe con: sudo apt install <nombre del paquete deb>
eric@eric:~$ snap install kubectl
error: Esta revisión del snap "kubectl" se publicó usando el confinamiento clásico por lo cual
podría realizar cambios arbitrarios del sistema fuera de la caja de seguridad en el que los
snaps suelen estar confinados, lo cual podría suponer un riesgo para su sistema.

Si lo entiende y desea continuar repita la orden incluyendo --classic.
eric@eric:~$ snap install kubectl --classic
Se ha instalado kubectl 1.28.3 por Canonical✓
```

- Una vez instalado, verificaremos que esta bien instalado con *kubectl version --client*.

```
eric@eric:~$ kubectl version --client
Client Version: v1.28.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

- Seguido, descargaremos el binario de Minikube y lo cargaremos el sistema:

```
eric@eric:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-arm64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100 86.6M  100 86.6M    0     0  21.5M      0  0:00:04  0:00:04 --:--:-- 21.5M
```

- Otorga permisos de ejecución al binario con `chmod +x minikube-linux-arm64`:

```
eric@eric:~$ chmod +x minikube-linux-arm64
```

- Moveremos el binario a un directorio incluido en tu variable de entorno ``PATH``, por ejemplo:
`sudo mv minikube-linux-arm64 /usr/local/bin/minikube`

```
eric@eric:~$ sudo mv minikube-linux-arm64 /usr/local/bin/minikube
```

- Iniciamos minikube

```
eric@eric:~$ minikube start --driver=docker
🐤 minikube v1.32.0 en Ubuntu 22.04 (arm64)
🌟 Using the docker driver based on user configuration
👍 Using Docker driver with root privileges
🏠 Starting control plane node minikube in cluster minikube
📦 Pulling base image ...
📦 Descargando Kubernetes v1.28.3 ...
> preloaded-images-k8s-v18-v1...: 341.16 MiB / 341.16 MiB 100.00% 23.29 M
> gcr.io/k8s-minikube/kicbase...: 410.58 MiB / 410.58 MiB 100.00% 15.16 M
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
📦 Preparando Kubernetes v1.28.3 en Docker 24.0.7...
  ■ Generando certificados y llaves
  ■ Iniciando plano de control
  ■ Configurando reglas RBAC...
🔗 Configurando CNI bridge CNI ...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔍 Verifying Kubernetes components...
🌟 Complementos habilitados: storage-provisioner, default-storageclass
🏠 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- Verificamos que este bien instalado con: `minikube status`.
- También podemos verificarlo con el siguiente comando: `kubectl cluster-info`.

```
eric@eric:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

- Esto muestra la información del clúster, indicando que estás conectado a Minikube.

```
eric@eric:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```


Kubernetes

Instalar Docker

- Antes de instalar kubernetes necesitaremos instalar docker ya que es necesario para su funcionamiento, por ejemplo con:

```
sudo apt-get update
sudo apt-get install docker.io
docker --version
sudo systemctl enable docker
sudo systemctl status docker
udo systemctl start docker
```

Instalar Kubernetes

- Una vez instalado docker, procedemos a instalar kubernetes
- Para empezar añadimos la clave de Kubernetes en ambos nodos, esto lo hacemos con el siguiente comando:

```
**curl -s** https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
```

```
eric@eric:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

- Ahora añadimos el repositorio de Kubernetes para poder instalarlo en cada nodo:

```
sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
```

```

eric@eric:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Repositorio: «deb http://apt.kubernetes.io/ kubernetes-xenial main»
Descripción:
Archive for codename: kubernetes-xenial components: main
Más información: http://apt.kubernetes.io/
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-http_apt_kubernetes_io_-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-http_apt_kubernetes_io_-jammy.list
Obj:1 http://es.ports.ubuntu.com/ubuntu-ports jammy InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Des:3 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Obj:4 https://pkg.jenkins.io/debian-stable binary/ Release
Des:5 http://es.ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Obj:6 https://download.docker.com/linux/ubuntu jammy InRelease
Err:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg
  Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
Des:9 http://ports.ubuntu.com/ubuntu-ports jammy-proposed InRelease [270 kB]
Obj:10 http://es.ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Des:11 http://es.ports.ubuntu.com/ubuntu-ports jammy-updates/restricted arm64 Packages [726 kB]
Des:8 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8.993 B]
Ign:12 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 Packages
Des:12 http://apt.kubernetes.io kubernetes-xenial/main arm64 Packages [68,5 kB]
Descargados 1.303 kB en 1s (950 kB/s)
Leyendo lista de paquetes... Hecho
W: Se produjo un error durante la verificación de las firmas. El repositorio no está actualizado y se utilizarán los ficheros de índice antiguos. Error de GPG: https://pkg.jenkins.io/debian-stable binary/
  Release: Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
W: Fallo al obtener https://pkg.jenkins.io/debian-stable/binary/Release.gpg Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 5BA31D57EF5975CA
W: No se han podido descargar algunos archivos de índice, se han omitido, o se han utilizado unos antiguos en su lugar.

```

- El último paso en el proceso de instalación es instalar **Kubeadm** en ambos nodos a través del comando:
sudo apt install kubeadm.

```

eric@eric:~$ sudo apt install kubeadm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  conntrack cri-tools ebtables ethtool kubelet kubectl kubernetes-cni socat
Se instalarán los siguientes paquetes NUEVOS:
  conntrack cri-tools ebtables ethtool kubeadm kubectl kubelet kubernetes-cni socat
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 78,2 MB de archivos.
Se utilizarán 325 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.ports.ubuntu.com/ubuntu-ports jammy/main arm64 conntrack arm64 1:1.4.6-2build2 [32,4 kB]
Des:3 http://es.ports.ubuntu.com/ubuntu-ports jammy/main arm64 ebtables arm64 2.0.11-4build2 [85,4 kB]
Des:4 http://es.ports.ubuntu.com/ubuntu-ports jammy/main arm64 ethtool arm64 1:5.16-1 [208 kB]
Des:6 http://es.ports.ubuntu.com/ubuntu-ports jammy/main arm64 socat arm64 1.7.4.1-3ubuntu4 [348 kB]
Des:2 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 cri-tools arm64 1.26.0-00 [17,3 MB]
Des:5 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 kubernetes-cni arm64 1.2.0-00 [25,8 MB]
Des:7 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 kubelet arm64 1.28.2-00 [16,8 MB]
Des:8 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 kubectl arm64 1.28.2-00 [8.829 kB]
Des:9 https://packages.cloud.google.com/apt kubernetes-xenial/main arm64 kubeadm arm64 1.28.2-00 [8.793 kB]
Descargados 78,2 MB en 9s (8.716 kB/s)
Seleccionando el paquete conntrack previamente no seleccionado.
(Leyendo la base de datos ... 174441 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../0-conntrack_1%3a1.4.6-2build2_arm64.deb ...
Desempaquetando conntrack (1:1.4.6-2build2) ...
Seleccionando el paquete cri-tools previamente no seleccionado.
Preparando para desempaquetar .../1-cri-tools_1.26.0-00_arm64.deb ...
Desempaquetando cri-tools (1.26.0-00) ...
Seleccionando el paquete ebtables previamente no seleccionado.
Preparando para desempaquetar .../2-ebtables_2.0.11-4build2_arm64.deb ...
Desempaquetando ebtables (2.0.11-4build2) ...
Seleccionando el paquete ethtool previamente no seleccionado.
Preparando para desempaquetar .../3-ethtool_1%3a5.16-1_arm64.deb ...
Desempaquetando ethtool (1:5.16-1) ...
Seleccionando el paquete kubernetes-cni previamente no seleccionado.
Preparando para desempaquetar .../4-kubernetes-cni_1.2.0-00_arm64.deb ...
Desempaquetando kubernetes-cni (1.2.0-00) ...
Seleccionando el paquete socat previamente no seleccionado.
Preparando para desempaquetar .../5-socat_1.7.4.1-3ubuntu4_arm64.deb ...
Desempaquetando socat (1.7.4.1-3ubuntu4) ...
Seleccionando el paquete kubelet previamente no seleccionado.
Preparando para desempaquetar .../6-kubelet_1.28.2-00_arm64.deb ...
Desempaquetando kubelet (1.28.2-00) ...
Seleccionando el paquete kubectl previamente no seleccionado.
Preparando para desempaquetar .../7-kubectl_1.28.2-00_arm64.deb ...
Desempaquetando kubectl (1.28.2-00) ...
Seleccionando el paquete kubeadm previamente no seleccionado.
Preparando para desempaquetar .../8-kubeadm_1.28.2-00_arm64.deb ...
Desempaquetando kubeadm (1.28.2-00) ...
Configurando conntrack (1:1.4.6-2build2) ...
Configurando kubectl (1.28.2-00) ...
Configurando ebtables (2.0.11-4build2) ...
Configurando socat (1.7.4.1-3ubuntu4) ...
Configurando cri-tools (1.26.0-00) ...

```

- Verificamos la versión con: *kubeadm version*

```

eric@eric:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"28", GitVersion:"v1.28.2", GitCommit:"89a4ea3e1e4ddd7f7572286090359983e0387b2f", GitTreeState:"clean", BuildDate:"2023-09-13T09:34:32Z", GoVersion:"go1.20.8", Compiler:"gc", Platform:"linux/arm64"}

```