

18-10-2023

# Servicios de Desarrollo AWS

Computo en la nube



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



Yadira Estrada Cervantes  
TECNM ITZ

## Índice

Introducción .....	2
Desarrollo.....	2
Crear Usuario de GitHub.....	2
Instalación de Git en Windows .....	6
Crear un Repositorio.....	14
Crear un Rol en IAM .....	17
Crear entorno en Elastic Beanstalk .....	19
Canalización de instancia CodePipeline.....	23
Actualización Constante.....	26
Eliminar Canalización .....	33
Eliminar Entorno .....	34
Eliminar Instancia .....	35
Eliminar Rol .....	36
Eliminar Repositorio.....	36
Conclusiones.....	38
Referencias .....	38

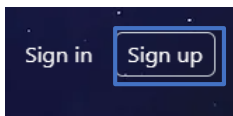
## Introducción

A continuación, se mostrará el proceso que se siguió para crear un entorno de desarrollo utilizando las tecnologías EC2, CodePipeline, Elastik Beanstalk, IAM y GitHub en AWS.

## Desarrollo

### Crear Usuario de GitHub

Entraremos a la [pagina de GitHub](#) donde daremos clic en la parte de “Sign Up”



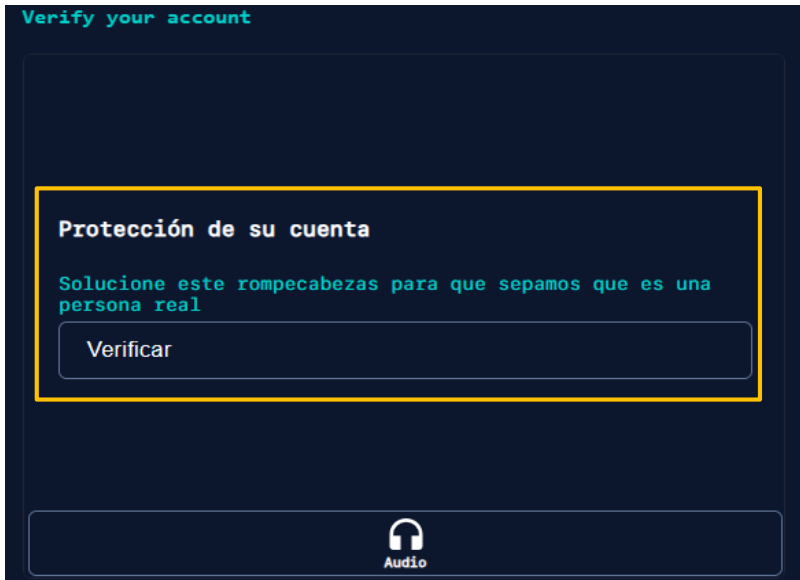
Introduciremos los siguientes datos:

- Email
- Contraseña
- Nombre de usuario
- Si deseamos recibir anuncios (en este caso no)

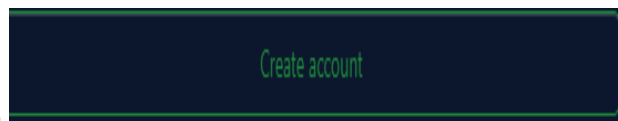
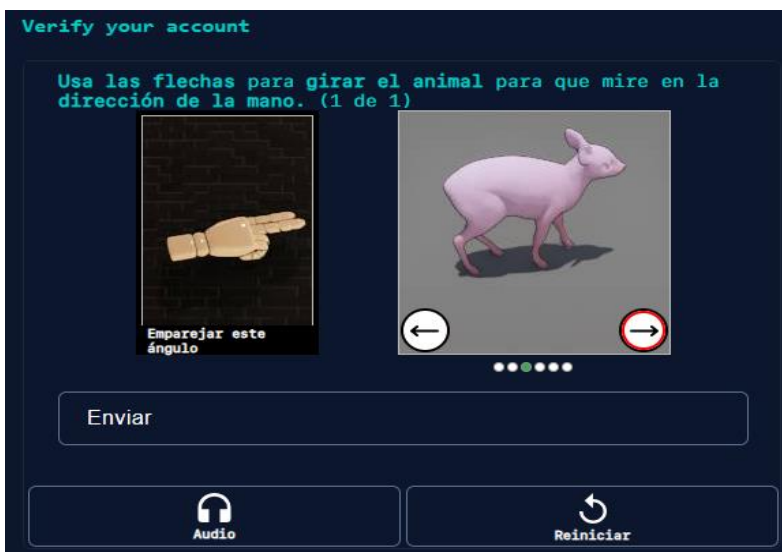
Proseguiremos dando clic en **continuar**.

A dark-themed screenshot of the GitHub sign-up form. The form contains several input fields, each with a checkmark and a blue border. The first field is "Enter your email\*" with the value "yadiraestrada@gmail.com". The second field is "Create a password\*" with a masked password "••••••••". The third field is "Enter a username\*" with the value "YadiraEstrada". Below these fields is a question: "Would you like to receive product updates and announcements via email?". The answer field contains "n" and is highlighted with a purple border. A "Continue" button is visible to the right of the answer field.

Introduciremos los siguientes datos: Nos pedirá que verifiquemos el correo establecido, para ello daremos clic en “Verificar”



Seguiremos las indicaciones que nos da GitHub



Una vez realizado el captcha daremos clic en

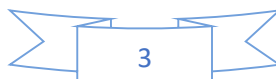
Daremos clic en [abrir GitHub](#) e introduciremos el [código](#) dado

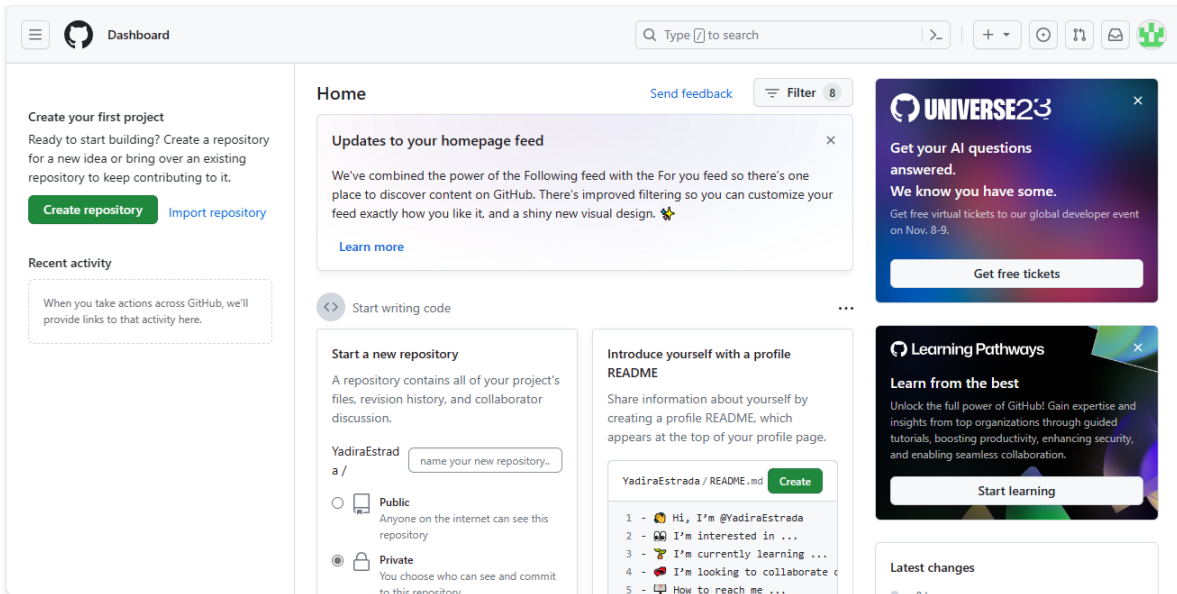
Continue signing up for GitHub by entering the code below:

23255935

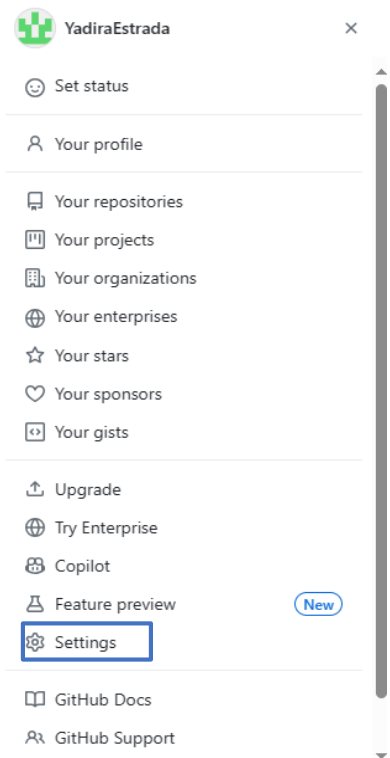
Open GitHub

Una vez realizado el Finalmente, podremos ver la página principal de GitHub





Nos iremos al perfil y daremos clic en [Configuraciones](#)



<> [Developer settings](#)

Bajaremos y buscaremos la opción de

Seleccionaremos la opción de **tokens** para **generar un token**

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Beta

Tokens (classic)

## Personal access tokens (classic)

Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Seleccionaremos el **token clasico**

Generate new token **Beta**

Fine-grained, repo-scoped

Generate new token (classic)

For general use

Pondremos **nombre** y **fecha** de expiración a consideración del usuario

### Note

tokenPrueba

What's this token for?

### Expiration \*

30 days

The token will expire on Fri, Nov 17 2023

Seleccionaremos todos los permisos

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input checked="" type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects
<input checked="" type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input checked="" type="checkbox"/> <b>admin:public_key</b>	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> <b>admin:repo_hook</b>	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks

**Generate token**

Finalizaremos dando clic en

Esto nos dará un token e cual copiaremos para usarlo después (lo oculto por seguridad)



## Instalación de Git en Windows

Entraremos a la [Página de instalación de Git](#) donde seleccionaremos la [arquitectura](#) de nuestra computadora Windows.

### Other Git for Windows downloads

Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

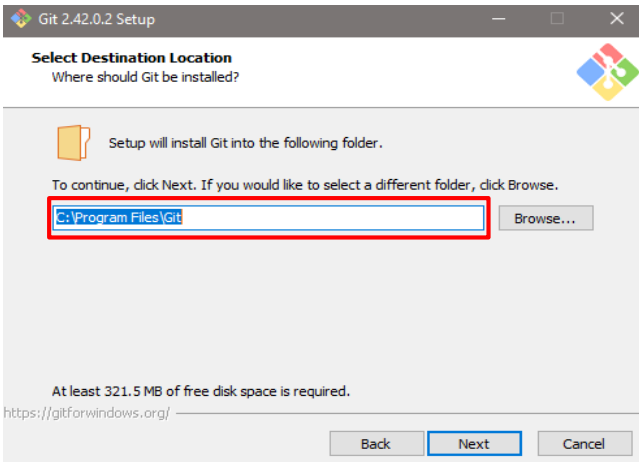


Veremos en la carpeta de descargas el archivo

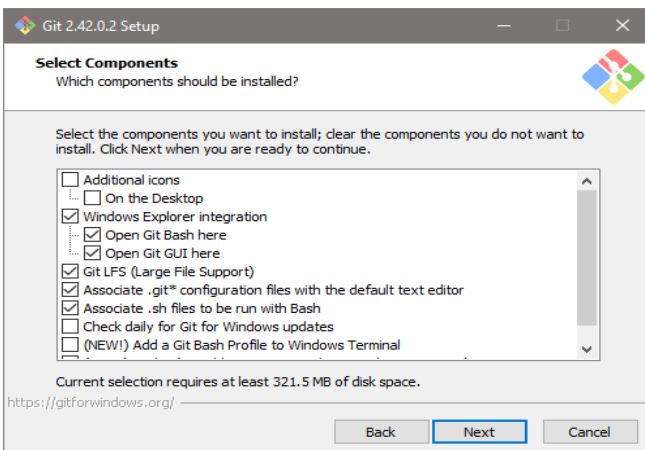
Al ejecutarlo, veremos la licencia, daremos clic en [next](#).



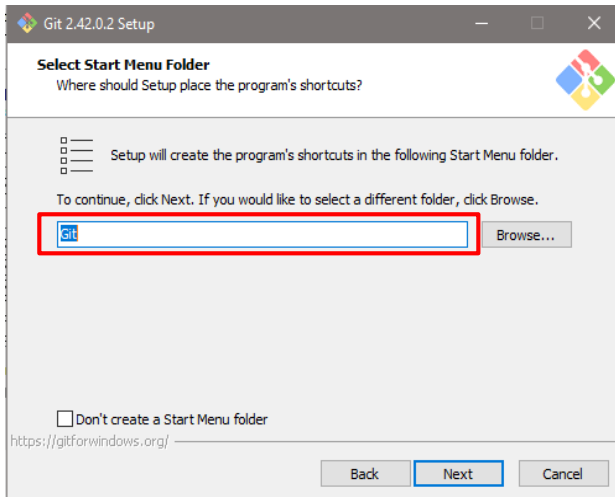
Seleccionaremos el **directorio** dónde se descargará, daremos clic en **next**.



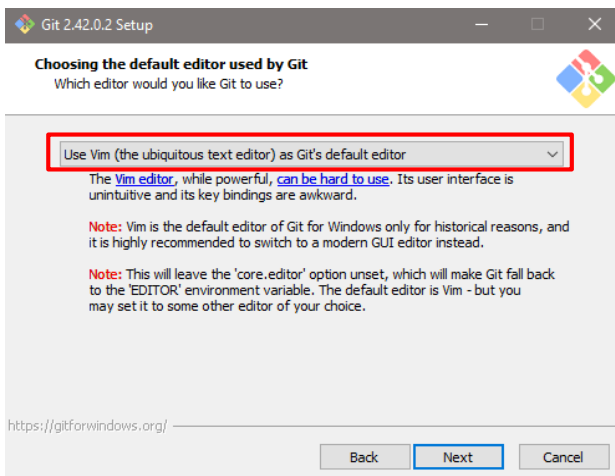
Veremos los diferentes componentes, lo dejaremos tal cual está; daremos clic en **next**.



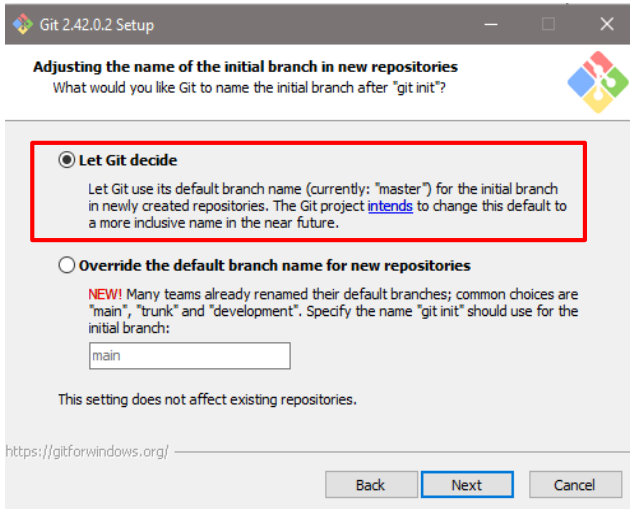
Veremos los diferentes menus de inicio, lo dejaremos en **git**; daremos clic en **next**.



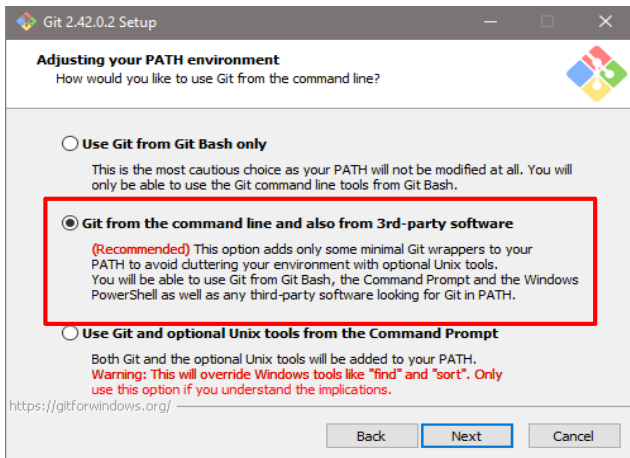
Veremos los diferentes editores, lo dejaremos en **Vim**; daremos clic en **next**.



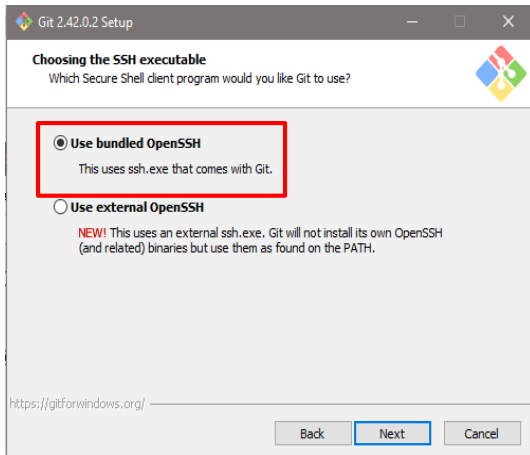
Veremos las diferentes formas de llamar a la branch principal (el personalizado o el de default), lo dejaremos en **el de default**; daremos clic en **next**.



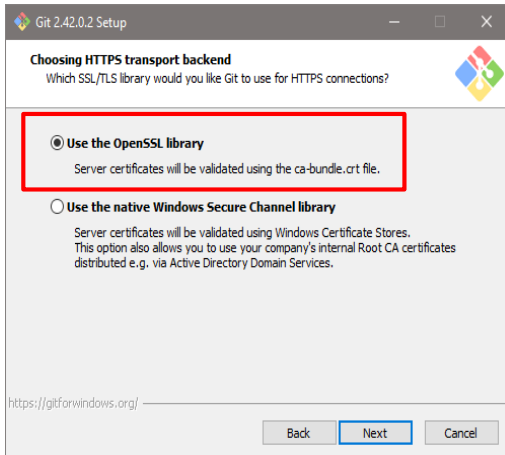
Veremos las diferentes formas de entorno (git bash, cmd, comand line), lo dejaremos en **el comand line**; daremos clic en **next**.



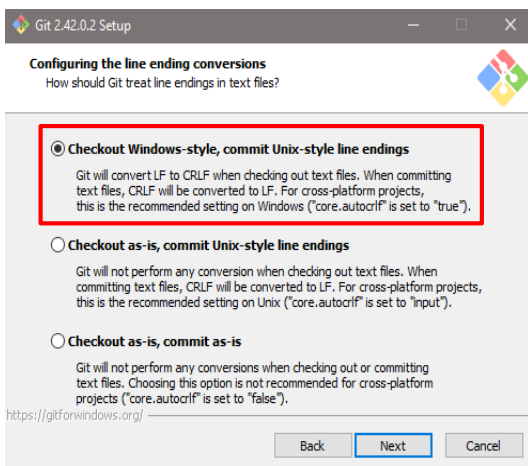
Nos preguntara que ssh deseamos usar, s una propia o de git, seleccionaremos la de **git**; daremos clic en **next**.



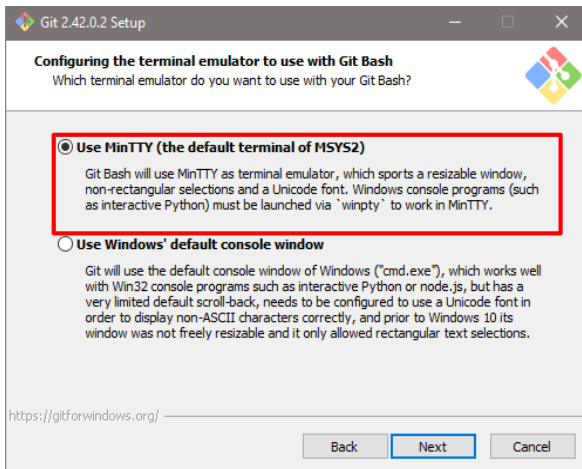
Nos preguntara por la librería que deseamos usar para http, si la de Windows u OpenSSL, seleccionaremos la de **OpenSSL**; daremos clic en **next**.



Nos preguntara por el tipo de cierre deseamos, seleccionaremos la de **Windows-style con el commit de unix**; daremos clic en **next**.



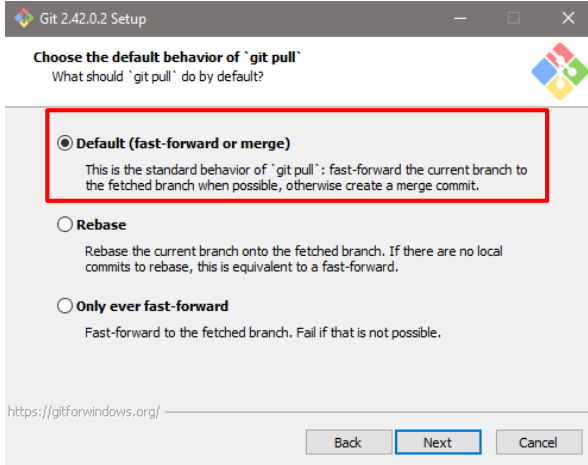
Nos preguntara por el tipo de cierre terminal, seleccionaremos la **minTTY**; daremos clic en **next**.



Nos preguntara por el tipo de pull que se desea hacer:

- Merge
- Rebase
- Fast-forward

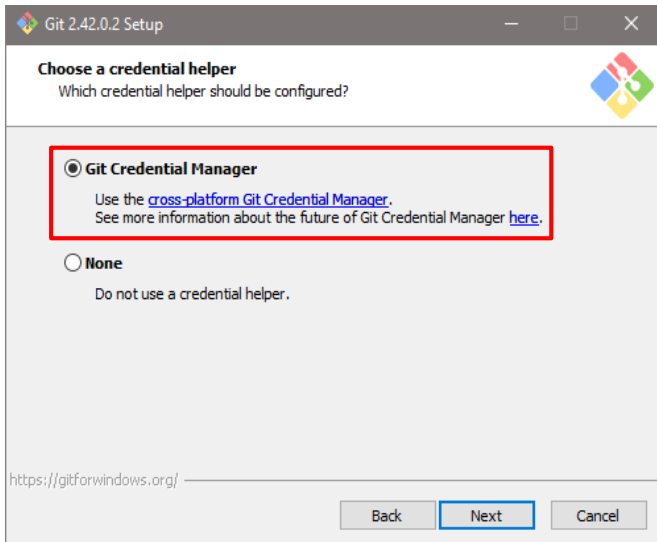
seleccionaremos la **merge**; daremos clic en **next**.



Nos preguntara por el manejador de credencial:

- Git credential manager
- Ninguno

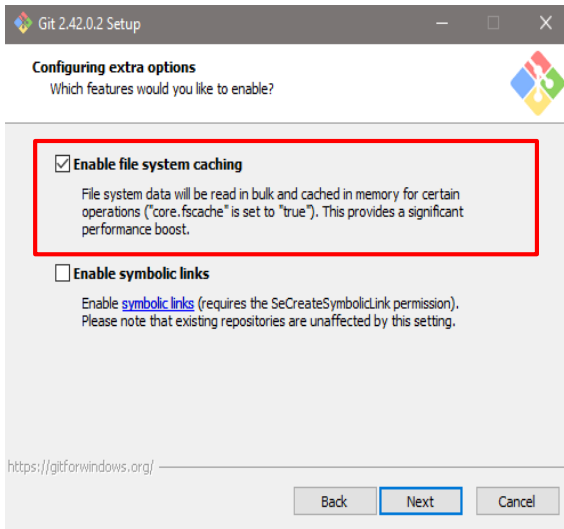
seleccionaremos la **manejador de git**; daremos clic en **next**.



Nos preguntara por el tipo de features de credencial:

- File System Catching
- Links Simbólicos

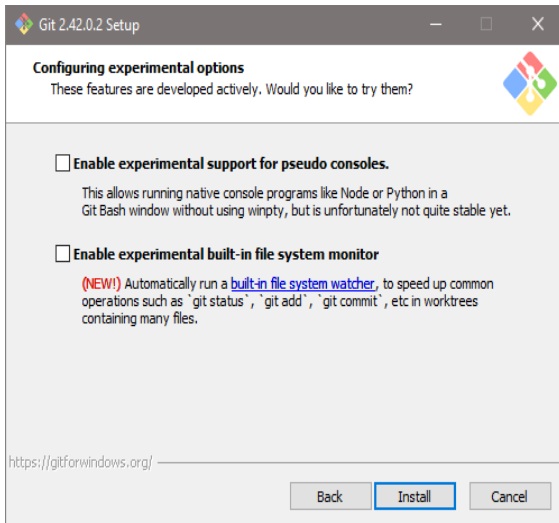
seleccionaremos la **File System Catching**; daremos clic en **next**.



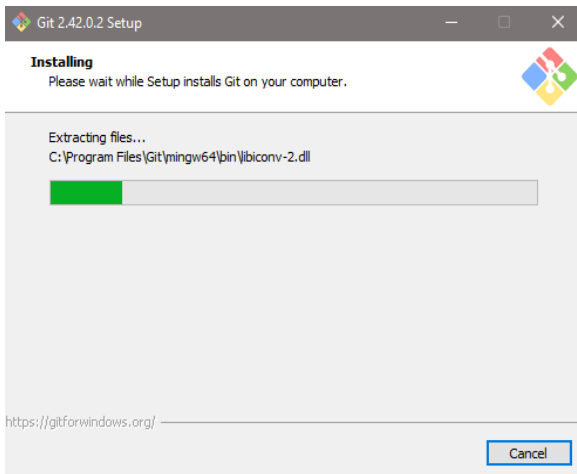
Nos preguntara por las configuraciones experimentales:

- Pseudo Consolas
- Monitor

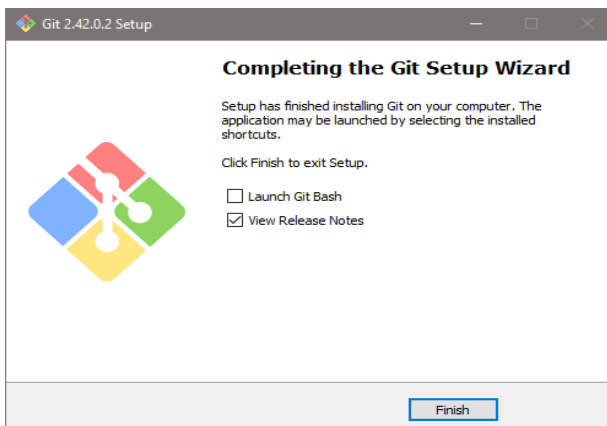
No seleccionaremos; daremos clic en **install**.



Esperaremos a que termine de instalar



Seleccionaremos **finish**.



Entraremos al git bash (buscándolo en el buscador de Windows) y escribiremos `git config --global user.name "nombre_usuario"` y `git config --global user.email email` para configurar de manera global al usuario.

```
LENOVO@DESKTOP-AJ5202D MINGW64 ~  
$ git config --global user.name "YadiraEstrada"  
LENOVO@DESKTOP-AJ5202D MINGW64 ~  
$ git config --global user.email yadraestrada@gmail.com
```

Podremos ver la configuración que acabamos de hacer con el comando `git config --list`

```
LENOVO@DESKTOP-AJ5202D MINGW64 ~  
$ git config --list  
diff.binary=cat  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fsck=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultBranch=master  
user.name=YadiraEstrada  
user.email=yadraestrada@gmail.com
```

## Crear un Repositorio

Irémpa a GitHub para cre un nuevo repositorio, dando clic en “[Crear Repositorio](#)”

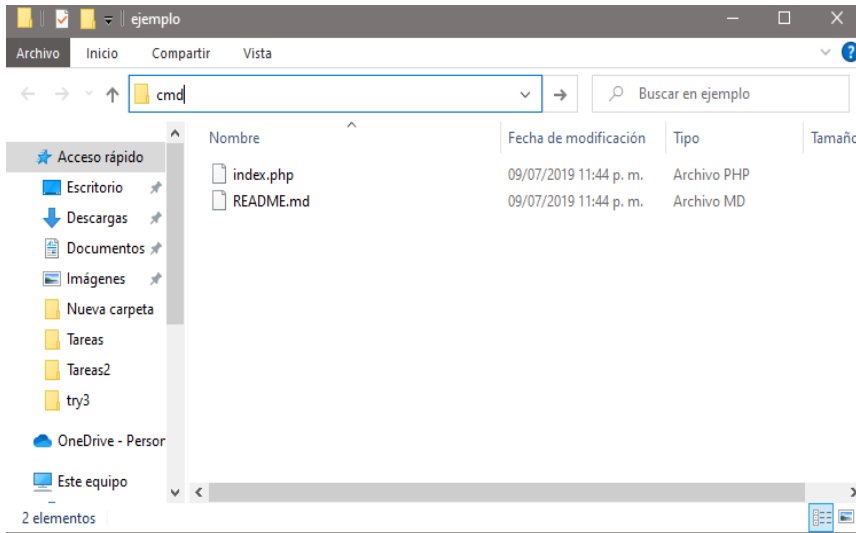
### Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository

Import repository

Posteriormente, abriremos la carpeta con el explorador de archivos y escribiremos `cmd` para abrir la línea de comandos.



En GitHub, introduciremos:

- El nombre del repositorio
- La descripción (opcional)
- Repositorio público o privado (el primero es más cuando se comparten librerías publicas y el segundo más para trabajo privado).

Required fields are marked with an asterisk (\*).

Owner \*  / Repository name \*   
✓ Prueba is available.

Great repository names are short and memorable. Need inspiration? How about [didactic-spoon](#) ?

Description (optional)

Public  
Anyone on the internet can see this repository. You choose who can commit.

Private  
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

Para continuar, daremos clic en

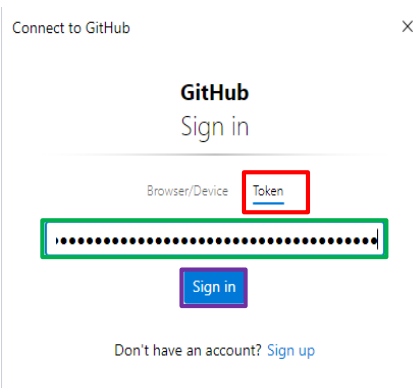
Nos dará dos maneras de crear el repositorio en GitHub, creando uno nuevo en local o subiendo uno existente, elegiremos la primera opción, copiando el primero código que se nos presenta. Para eso, daremos clic en **copiar**. Podemos explicar los comandos en:

- Echo "# Prueba" >> README.md – Crea el archivo readme
- git init – inicia el repositorio
- git add README.md –pide que añada el README al stage
- git commit –m "first commit" – pide que cree un commit con el comentario de first commit
- git remote add origin https://github.com.YadiraEstrada/Prueba.git- referencia al git en la nube
- git push –u origin main – sube los cambios al main

```
...or create a new repository on the command line

echo "# Prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/YadiraEstrada/Prueba.git
git push -u origin main
```

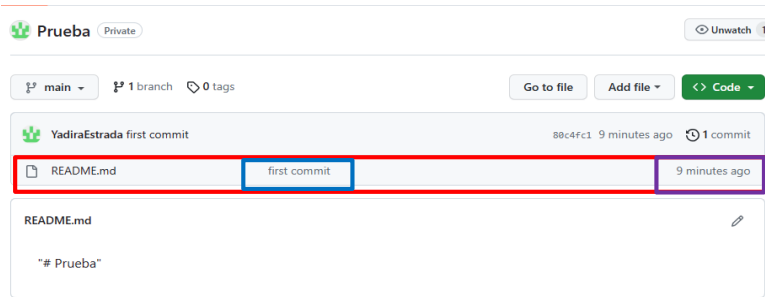
Al copiarla en el cmd nos abrirá una ventana para abrir la sesión de GitHub, para eso elegiremos la opción de **token**, donde pegaremos el **token** que habíamos guardado anteriormente. Para continuar daremos clic en **Sign In**.



Daremos enter para continuar, al final nos dirá que cambiara la rama principal a la **main**.

```
C:\Users\LENOVO\Desktop\ejemplo>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 228 bytes | 114.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/YadiraEstrada/Prueba.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
C:\Users\LENOVO\Desktop\ejemplo>
```

Si recargamos la página de GitHub podremos ver el **readme** que se subió, en el **commit** en el que se subió o modificó y hace cuánto **tiempo** se realizo



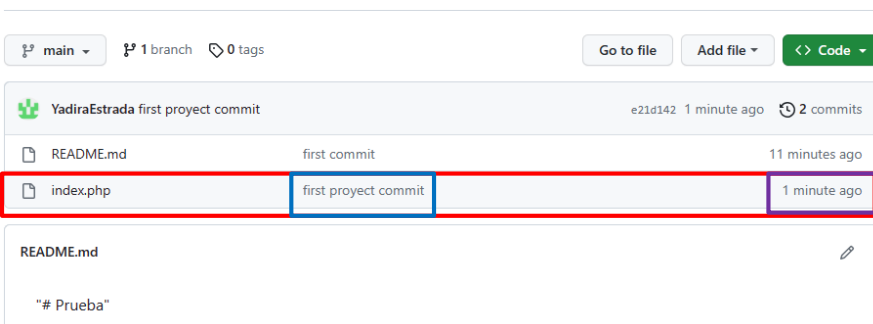
Para subir el resto de archivos con cambios es un **git add .**, posteriormente se crea el **commit** como ya se había enseñado antes y realizamos un **git push** para subir cambios locales al repositorio en la nube.

```
C:\Users\LENOVO\Desktop\ejemplo>git add .
warning: in the working copy of 'index.php', LF will be replaced by CRLF the next time Git touches it

C:\Users\LENOVO\Desktop\ejemplo>git commit -m "first project commit"
[main e21d142] first project commit
1 file changed, 157 insertions(+)
create mode 100644 index.php

C:\Users\LENOVO\Desktop\ejemplo>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 3.35 KiB | 1.68 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/YadiraEstrada/Prueba.git
 80c4fc1..e21d142  main -> main
```

Si recargamos la página de GitHub podremos ver el **archivo** que se subió, en el **commit** en el que se subió o modificó y hace cuánto **tiempo** se realizo



## Crear un Rol en IAM

Entraremos en el servicio de IAM mediante la consola con una sesión iniciada, buscaremos la sección de role y daremos clic en **Crear rol**

Posteriormente, elegiremos el tipo de confianza [de servicio de AWS](#), para que pueda utilizar los servicios de este (entre ellos EC2).

**Tipo de entidad de confianza**

- Servicio de AWS**  
Permita que servicios de AWS como EC2, Lambda u otros realicen acciones en esta cuenta.
- Cuenta de AWS**  
Permitir a las entidades de otras cuentas de AWS que le pertenezcan a usted o a un tercero realizar acciones en esta cuenta.
- Identidad web**  
Permite a los usuarios federados por el proveedor de identidad web externo especificado asumir este rol para realizar acciones en esta cuenta.
- Federación SAML 2.0**  
Permitir que los usuarios federados con SAML 2.0 a partir de un directorio corporativo realicen acciones en esta cuenta.
- Política de confianza personalizada**  
Cree una política de confianza personalizada para permitir que otras personas realicen acciones en esta cuenta.

En casos de uso elegiremos la opción de [EC2](#), donde nos permita [administrar instancias](#). Daremos clic en [siguiente](#) para continuar.

EC2

Elija un caso de uso para el servicio especificado.

Caso de uso

- EC2**  
Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**  
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**  
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**  
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**  
Allows EC2 Scheduled Instances to manage instances on your behalf.

Cancelar **Siguiente**

Para los permisos buscaremos el permiso [awselasticbeanstalkwebtier](#) y lo seleccionaremos. Daremos clic en [siguiente](#) para continuar.


### Agregar permisos Información

**Políticas de permisos (1/882)** Información

Elija una o varias políticas para adjuntarlas al nuevo rol.

Filtrar por Tipo

awselasticbeanstalkwebtier X Todos los tipos 1 coincidencia < 1 > ⚙️

<input checked="" type="checkbox"/>	Nombre de la política <a href="#">🔗</a>	Tipo	Descripción
<input checked="" type="checkbox"/>	 AWSElasticBeanstalk...	Administrada por AWS	Provide the instances in your web server ...

► Establecer límite de permisos: *opcional*

Cancelar Anterior **Siguiente**

Le pondremos el nombre al rol.

### Asignar nombre, revisar y crear

**Detalles del rol**

**Nombre del rol**  
Ingrese un nombre significativo para identificar a este rol.

test-rol

64 Caracteres máximos. Utilice caracteres alfanuméricos y '+ =, @, \_'.

**Descripción**  
Agregue una breve explicación para este rol.

Allows EC2 instances to call AWS services on your behalf.

1000 Caracteres máximos. Utilice caracteres alfanuméricos y '+ =, @, \_'.

Daremos clic en **Crear rol**

Podremos ver en el servicio de IAM en la parte de roles que se ha creado exitosamente.

[test-rol](#) Servicio de AWS: ec2 -

## Crear entorno en Elastic Beanstalk

Entraremos en el servicio de ElasticBeanstalk mediante la consola con una sesión iniciada,

buscaremos la sección de role y daremos clic en **Crear aplicación**

Para el nivel de entorno elegiremos la opción de [servidor web](#).

### Nivel de entorno [Información](#)

Amazon Elastic Beanstalk tiene dos tipos de niveles de entorno para admitir diferentes tipos de aplicaciones web.

Entorno de servidor web

Ejecute un sitio web, una aplicación web o una API web que atienda solicitudes HTTP. [Más información](#)

Entorno de trabajo

Ejecute una aplicación de proceso de trabajo que procese cargas de trabajo de ejecución prolongada bajo demanda o realice tareas de forma programada. [Más información](#)

Le pondremos **nombre** a la aplicación.

### Información de la aplicación [Información](#)

Nombre de aplicación

phprueba

La longitud máxima es de 100 caracteres.

► Etiquetas de aplicación (opcional)

Le dejaremos el **nombre del entorno** y el **dominio** lo dejaremos en la configuración de default.

### Información del entorno [Información](#)

Elija el nombre, el subdominio y la descripción del entorno. No se pueden cambiar más adelante.

Nombre del entorno

Phprueba-env

Debe tener entre 4 y 40 caracteres. El nombre solo puede contener letras, números y guiones. No puede comenzar ni terminar por un guion. Este nombre debe ser único dentro de una región de su cuenta.

Dominio

Deje en blanco el valor autogenerado

.us-west-1.elasticbeanstalk.com

Verificar  
disponibilidad

Descripción del entorno

Elegiremos el tipo de **plataforma administrada**, con **php** y las **configuraciones** de default,

## Plataforma [Información](#)

### Tipo de plataforma

- Plataforma administrada**  
Plataformas publicadas y mantenidas por Amazon Elastic Beanstalk. [Más información](#)
- Plataforma personalizada  
Plataformas creadas y de su propiedad. Esta opción no está disponible si no tiene plataformas.

### Plataforma

PHP

### Ramificación de la plataforma

PHP 8.2 running on 64bit Amazon Linux 2023

### Versión de la plataforma

4.0.1 (Recommended)

Dejaremos al código de [aplicación de ejemplo](#)

## Código de aplicación [Información](#)

- Aplicación de ejemplo**
- Existing version  
Application versions that you have uploaded.
- Cargar el código  
Cargue un paquete de código fuente desde su equipo o copie uno desde Amazon S3.

Para el tipo de instancia elegiremos la **única**, daremos clic en **siguiente** para continuar.

## Valores preestablecidos [Información](#)

Comience a partir de un elemento preestablecido que coincida con su caso de uso o elija una configuración personalizada para anular los valores recomendados y utilice los valores predeterminados del servicio.

### Elementos preestablecidos de configuración

- Instancia única (compatible con la capa gratuita)**
- Instancia única (mediante instancia de spot)
- Alta disponibilidad
- Alta disponibilidad (con instancias de spot y bajo demanda)
- Configuración personalizada

Cancelar

**Siguiente**

Para el rol de servicio elegiremos **crear uno nuevo**, seleccionaremos las políticas de **aws-elasticbeanstal-service-role** y pondremos el **rol que hemos creado en IAM**, daremos clic en **siguiente** para continuar.

**Rol de servicio**

Crear y utilizar un nuevo rol de servicio

Usar un rol de servicio existente

**Nombre del rol de servicio**  
 Ingrese el nombre de un rol de IAM que Elastic Beanstalk creará para asumir como rol de servicio. Beanstalk le adjuntará las políticas administradas necesarias.

aws-elasticbeanstalk-service-role

Ver los detalles de los permisos

**Par de claves de EC2**  
 Seleccione un par de claves de EC2 para iniciar sesión de forma segura en sus instancias de EC2. [Más información](#)

Elija un par de claves

**Perfil de instancia de EC2**  
 Elija un perfil de instancia de IAM con políticas administradas que permitan a las instancias de EC2 realizar las operaciones necesarias.

test-rol

Ver los detalles de los permisos

Cancelar Ir a revisión Anterior **Siguiente**

Elegiremos la **nube virtual privada** según nuestra conveniencia

**Nube virtual privada (VPC)**

**VPC**  
 Lance el entorno en una VPC personalizada en lugar de en la VPC predeterminada. Puede crearla junto a sus subredes en la consola de administración de la VPC. [Más información](#)

vpc-016130447c7cead58 | (172.31.0.0/16)

[Crear VPC personalizada](#)

Elegiremos la **subred de la instancia**

**Dirección IP pública**  
 Asigne una dirección IP pública a las instancias de Amazon EC2 del entorno.

Activado

**Subredes de instancia**

Q Filtrar subredes de instancia

<input type="checkbox"/>	Zona de disponibil...	Subred	CIDR	Nombre
<input checked="" type="checkbox"/>	us-west-1c	subnet-00ed4f2ad...	172.31.16.0/20	
<input type="checkbox"/>	us-west-1b	subnet-0deb977fe...	172.31.0.0/20	

Daremos clic en **Siguiente** para continuar.

**Grupos de seguridad de EC2**  
 Seleccione grupos de seguridad para controlar el tráfico.

Grupos de seguridad de EC2 (5)

🔍 Filtrar grupos de seguridad

<input type="checkbox"/>	Nombre del grupo	ID del grupo	Nombre
<input checked="" type="checkbox"/>	default	sg-0bbdaf8111bdc454d	
<input type="checkbox"/>	launch-wizard-1	sg-022634f4413c1f30a	
<input type="checkbox"/>	launch-wizard-2	sg-0e998aeb5c61977d5	
<input type="checkbox"/>	launch-wizard-3	sg-0150032effbbf13ba	
<input type="checkbox"/>	launch-wizard-4	sg-0ec8f8542f1bbf2a5	

Daremos clic en **Siguiente** para continuar.

Finalizaremos dando clic en **Enviar**

Esperaremos hasta ver el mensaje del **entorno lanzado correctamente**.

Entorno lanzado correctamente.

Elastic Beanstalk > Entornos > Phpprueba-env

Phpprueba-env Información

Acciones Cargar e implementar

**Información general del entorno**

Estado: ✔ Ok

ID del entorno: e-ym6e25bd9

Dominio: Phpprueba-env.eba-ufdbbrp2.us-west-1.elasticbeanstalk.com

Nombre de aplicación: phpprueba

**Plataforma** Cambiar la versión

Plataforma: PHP 8.2 running on 64bit Amazon Linux 2023/4.0.1

Ejecución de la versión: -

Estado de la plataforma: ✔ Supported

Podremos ver la instancia en EC2

Phpprueba-env i-047812747bf1a28d0 ✔ En ejecución

## Canalización de instancia CodePipeline

Entraremos en el servicio de CodePipeline mediante la consola con una sesión iniciada, buscaremos

la sección de role y daremos clic en **Crear la canalización**

Le pondremos **nombre a la canalización**, además de **crear un nuevo rol**, con el **nombre predeterminado**. Para continuar daremos clic en **siguiente**.

**Nombre de la canalización**  
Escriba el nombre de la canalización. No podrá editarlo después de su creación.

test-canaliza

No más de 100 caracteres

**Rol de servicio**

Nuevo rol de servicio  
Cree un rol de servicio en su cuenta.

Rol de servicio existente  
Elija un rol de servicio existente de su cuenta.

**Nombre del rol**

AWSCodePipelineServiceRole-us-west-1-test-canaliza

Escriba el nombre del rol de servicio.

Permitir que AWS CodePipeline cree un rol de servicio para que pueda utilizarse con esta nueva canalización

► **Configuración avanzada**

Cancelar **Siguiente**

Seleccionaremos **GitHub** como proveedor de origen en su versión 1 y daremos clic en **conectar**

**Proveedor del origen**  
Aquí es donde almacenó sus artefactos de entrada para la canalización. Elija el proveedor y luego proporcione los detalles de conexión.

GitHub (versión 1)

Conceda acceso a AWS CodePipeline a su repositorio de GitHub. Esto permite a AWS CodePipeline cargar confirmaciones de GitHub a su canalización.

Conectando

Nos abrirá una ventana donde damos clic en **“Authorize aws-codesuite”**

Authorize AWS CodePipeline (N. California)

AWS CodePipeline (N. California) by aws-codesuite wants to access your YadiraEstrada account

Repository webhooks and services  
Admin access

Repositories  
Public and private

Cancel **Authorize aws-codesuite**

Authorizing will redirect to  
<https://us-west-1.console.aws.amazon.com>

Posteriormente daremos clic en **confirmar**

**Procesando la solicitud OAuth**

Elija Confirmar para completar la conexión

Escogeremos el **repositorio antes creado**, con la **ramificación principal main** y utilizaremos el **WebHook de git** para la canalización. Posteriormente daremos clic en **siguiente**

Repositorio

Ramificación

Cambiar las opciones de detección  
Elija un modo de detección para iniciar la canalización de forma automática cuando se produce un cambio en el código fuente.

Webhooks de GitHub (recomendado)  
Utilizar webhooks en GitHub para iniciar mi canalización de forma automática cuando se produce un cambio

AWS CodePipeline  
Utilizar AWS CodePipeline para comprobar periódicamente si se han producido cambios

Nos saltaremos la opción de compilación dando clic en **“omitir la etapa de compilacion”**

**Compilación - opcional**

Proveedor de compilación  
Esta es la herramienta de su proyecto de compilación. Proporcione detalles del artefacto de compilación, como el sistema operativo, el archivo de especificación de compilación y los nombres de los archivos de salida.

Posteriormente daremos clic en **omitir** para confirmar

**Omitir la etapa compilación**

La canalización no incluirá una etapa compilación. ¿Está seguro de que desea omitir esta etapa?

Pasaremos a la parte de la implementación, donde elegiremos a **Elastic Beanstalk como proveedor**, a la **región de california** (por ser la más cercana), a la **aplicación que creamos** antes junto con su **nombre de entorno de default**. Posteriormente daremos clic en **siguiente**

**Implementar**

**Proveedor de implementación**  
Elija cómo implementar en las instancias. Seleccione el proveedor y luego proporcione los detalles de configuración para ese proveedor.

AWS Elastic Beanstalk

**Región**  
EE.UU. Oeste (Norte de California)

**Nombre de la aplicación**  
Elija una aplicación que ya haya creado en la consola de AWS Elastic Beanstalk. O bien, cree una aplicación en la consola de AWS Elastic Beanstalk y luego vuelva a esta tarea.

phpprueba

**Nombre del entorno**  
Elija un entorno que ya haya creado en la consola de AWS Elastic Beanstalk. O bien, cree un entorno en la consola de AWS Elastic Beanstalk y luego vuelva a esta tarea.

Phpprueba-env

Cancelar Anterior **Siguiente**

Para finalizar daremos clic en **Crear la canalización**

Podremos ver a creado correctamente

Herramientas para desarrolladores > CodePipeline > Canalizaciones > test-canaliza

**test-canaliza**

Notificar Editar Detener la ejecución Clonar la canalización **Cambio de versión**

**Source** Realizado correctamente  
ID de ejecución de la canalización: 6eca647a-495b-4a00-bb0f-f944c9066dbd

Source  
GitHub (versión 1)  
Realizado correctamente  
- hace 7 minutos  
e21d142a  
e21d142a Source: first project commit

Y que se ha actualizado correctamente

**Deploy** Realizado correctamente  
ID de ejecución de la canalización: 6eca647a-495b-4a00-bb0f-f944c9066dbd

Deploy  
AWS Elastic Beanstalk  
Realizado correctamente  
- hace 6 minutos  
e21d142a Source: first project commit

## Actualización Constante

Entraremos al index.php de ejemplo que creamos, escribiremos en el código php para que imprima, como se muestra a continuación.

```
<?php
echo("funciona")
?>
```

Añadiremos los cambios al **stage**, crearemos un **commit** con el comentario de "index modificado" y lo **subiremos** a GitHub. Podemos visualizar que **se ha subido**.

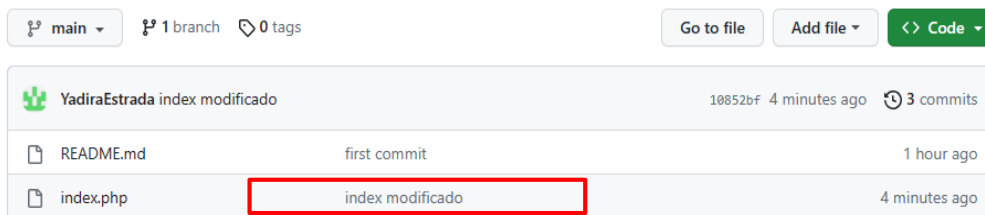
```
C:\Users\LENOVO\Desktop\ejemplo>git add .
warning: in the working copy of 'index.php', LF will be replaced by CRLF the next time Git touches it

C:\Users\LENOVO\Desktop\ejemplo>git commit -m "index modificado"
[main 10852bf] index modificado
 1 file changed, 2 insertions(+), 142 deletions(-)

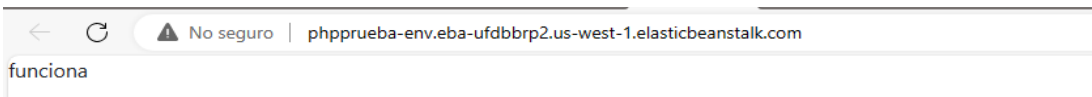
C:\Users\LENOVO\Desktop\ejemplo>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 374 bytes | 187.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/YadiraEstrada/Prueba.git
 e21d142..10852bf main -> main

C:\Users\LENOVO\Desktop\ejemplo>
```

Si vamos a la página de GitHub, en el repositorio, podremos ver que se ha subido el **commit**



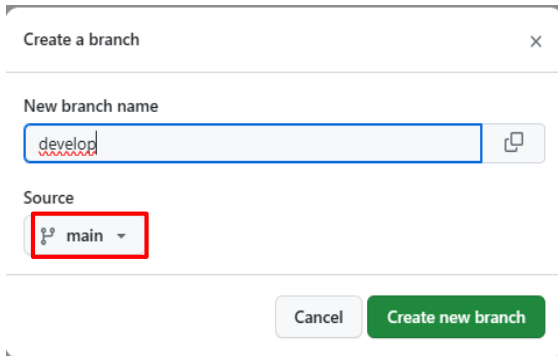
Si vamos al dominio de la página, podremos ver que se ha generado el cambio exitosamente.



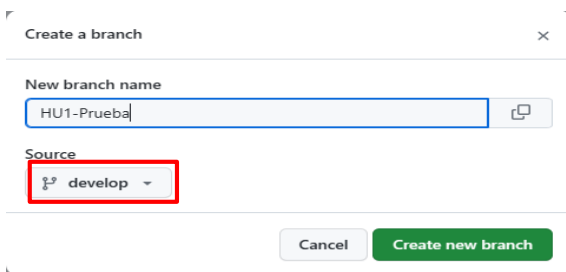
Pero así no es como se maneja git, no es darle el uso adecuado, para no generar errores en la página principal se crea la rama de **develop**, donde se van juntando los cambios que se van generando, para ello iremos a GitHub y, aunque se puede crear la rama en local con el comando `git checkout -b nombre-rama` es más recomendable crearlo desde GitHub en caso de estar trabajando en un equipo de desarrollo y para evitar problemas de choque de versiones.

Iremos a las branches y daremos clic en **New branch**

Le daremos **nombre a la branch** y la crearemos desde la **main**, posteriormente le daremos clic en **crear nueva rama**.



Pero eso no es suficiente, debemos crear branches según los cambios que vamos a hacer, por ejemplo, por tareas, en este caso se manejará por historias de usuario, crearemos una branch para una de estas. Le daremos **nombre a la branch** y la crearemos desde la **develop**, posteriormente le daremos clic en **crear nueva rama**.



Haremos un **git pull** para que nos baje **los cambios de las branch** al repositorio local. Cambiaremos a la rama donde trabajaremos, en este caso **HU1-Prueba**

```
C:\Users\LENOVO\Desktop\ejemplo>git pull
From https://github.com/YadiraEstrada/Prueba
* [new branch]   HU1-Prueba -> origin/HU1-Prueba
* [new branch]   develop  -> origin/develop
Already up to date.

C:\Users\LENOVO\Desktop\ejemplo>git switch HU1-Prueba
Switched to a new branch 'HU1-Prueba'
branch 'HU1-Prueba' set up to track 'origin/HU1-Prueba'.

C:\Users\LENOVO\Desktop\ejemplo>
```

Escribiremos código html en el index.php para que muestre y lo guardaremos.

**Nota:** antes de generar cambios siempre se debe de hacer un git pull en caso de que alguien mas hizo cambios en la misma rama, de lo contrario dará varios problemas de versiones.

```

<?php
echo("funciona")
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Prueba</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://miromannino.github.io/Justified-Gallery/bower_components/justified-gallery/dist/css/
  <link rel="stylesheet" href="https://miromannino.github.io/Justified-Gallery/bower_components/colorbox/example3/colorbox.
</head>
<body>
<h1>Funciona</h1>
</body>
</html>

```

subiremos los cambios con `git add .`, `git commit -m ""` y `git push` como se mostró anteriormente podremos ver los cambios generados (los commits, branch y mergeos) con el comando `git log --graph`

```

C:\Users\LENOVO\Desktop\ejemplo>git log --graph
* commit 80119c30d7bdcf1e1c55a350476d590d41c2c9f1 (HEAD -> HU1-Prueba, origin/HU1-Prueba)
  Author: YadiraEstrada <yadraestrada@gmail.com>
  Date: Thu Oct 19 01:13:38 2023 -0600

    HU1-Prueba_Nuevo cambio

* commit 10852bfea855b8650116e2e1867a39d1ffed33a2 (origin/main, origin/develop, main)
  Author: YadiraEstrada <yadraestrada@gmail.com>
  Date: Thu Oct 19 00:57:38 2023 -0600

    index modificado

* commit e21d142a048f46477581efb14fa54ceda63c51c5
  Author: YadiraEstrada <yadraestrada@gmail.com>
  Date: Wed Oct 18 23:38:45 2023 -0600

    first proyect commit

* commit 80c4fc1b5a76c08e0ead1b5e972661ac598e9c02
  Author: YadiraEstrada <yadraestrada@gmail.com>
  Date: Wed Oct 18 23:28:06 2023 -0600

    first commit

```

Para hacer el mergeo, primero haremos el `git pull` tanto de `HU1-Prueba` como de `develop`, una vez verificado que no tienen ningún problema iremos a GitHub

**Nota:** se recomienda hacer los merge desde GitHub para evitar problemas de versiones, pero en comando seria con `git merge nombre_de_la_rama` desde la rama a la cual se le combinará.

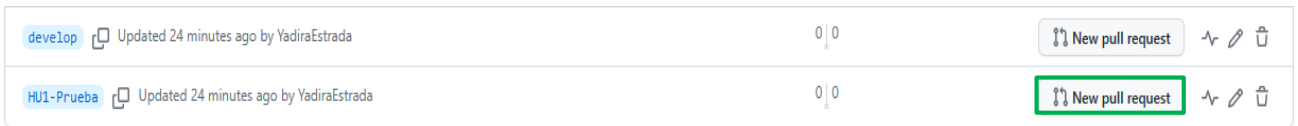
```

C:\Users\LENOVO\Desktop\ejemplo>git pull
Already up to date.

C:\Users\LENOVO\Desktop\ejemplo>git pull origin develop
From https://github.com/YadiraEstrada/Prueba
 * branch      develop      -> FETCH_HEAD
Already up to date.

```

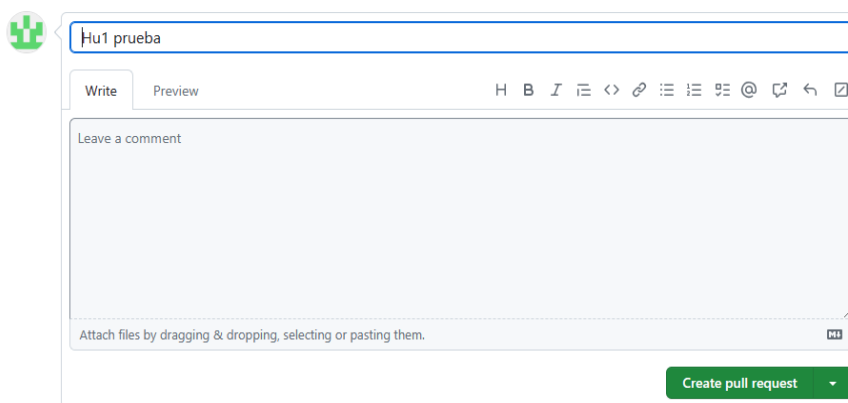
Nos iremos a las ramas del repositorio en GitHub y generaremos un **pull request**



Le indicaremos que haremos merge de la branch **HU1-Prueba** a **develop**



Le pondremos nombre al **merge** y crearemos el **pull request**



Normalmente se revisa entre varios programadores, aquí quien aprobará el pull request debe de ser

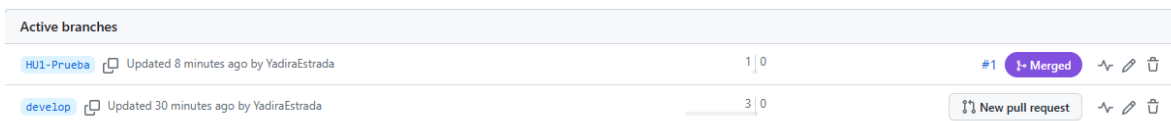
quien lo reviso, esto se logra dando clic en **Merge pull request** para posteriormente dar clic en **Confirm merge**

Veremos que ha sido mergeada

Hu1 prueba #1

**Merged** YadiraEstrada merged 2 commits into **main** from **HU1-Prueba** now

Veremos que nos marca la rama como **mergeada**



En consola bajaremos los cambios al repositorio local con un **git pull**

```
C:\Users\LENOVO\Desktop\ejemplo>git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 626 bytes | 208.00 KiB/s, done.
From https://github.com/YadiraEstrada/Prueba
  10852bf..46c7eaa  main      -> origin/main
Already up to date.

C:\Users\LENOVO\Desktop\ejemplo>
```

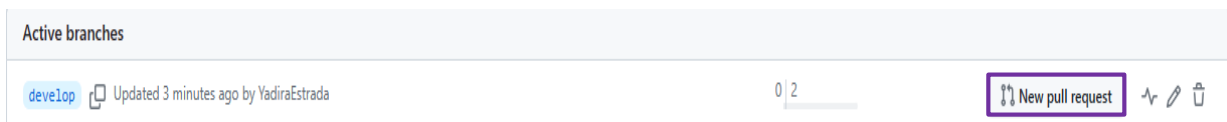
Cambiaremos a la rama **develop** y también bajaremos los cambios de **develop** y de **main** por si alguien subió cambios (man solo se usa cuando el sistema está completo, los cambios de **develop** se suben una vez que esté revisado, validado y aceptado el sistema).

```
C:\Users\LENOVO\Desktop\ejemplo>git switch develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

C:\Users\LENOVO\Desktop\ejemplo>git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 627 bytes | 15.00 KiB/s, done.
From https://github.com/YadiraEstrada/Prueba
  10852bf..55deec7  develop -> origin/develop
Updating 10852bf..55deec7
Fast-forward
 index.php | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

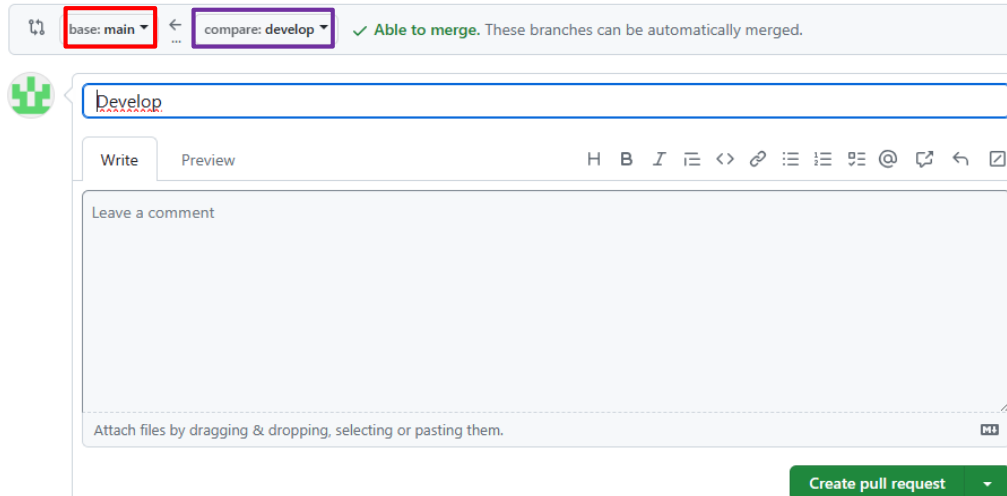
C:\Users\LENOVO\Desktop\ejemplo>git pull origin main
From https://github.com/YadiraEstrada/Prueba
 * branch      main      -> FETCH_HEAD
Already up to date.
```

Proseguiremos a hacer el **pull request**.



Le indicaremos que haremos merge de la branch **develop** a **main**

Le pondremos nombre al **merge** y crearemos el **pull request**



Normalmente se revisa entre varios programadores, aquí quien aprobará el pull request debe de ser quien lo reviso, esto se logra dando clic en **Merge pull request** para posteriormente dar clic en **Confirm merge**

Volveremos al dominio y veremos que se ha subido el cambio

funciona  
**Funciona!!**

Ademas de que podemos ver la actualización en CodePepiline



Si volvemos a ejecutar el `git log --graph` podremos observar los cambios.



```
C:\Users\LENOVO\Desktop\ejemplo>git log --graph
* commit 55deec766f1d292782368a734571a0c3561d0ea4 (HEAD -> develop, origin/develop)
Merge: 10852bf 9d0e207
Author: YadiraEstrada <148421054+YadiraEstrada@users.noreply.github.com>
Date: Thu Oct 19 01:59:17 2023 -0600

Merge pull request #2 from YadiraEstrada/HU1-Prueba

HU1 prueba

* commit 9d0e20788aefff1378734afca7719109d74fe83d (origin/HU1-Prueba, HU1-Prueba)
Author: YadiraEstrada <yadraestrada@gmail.com>
Date: Thu Oct 19 01:56:25 2023 -0600

que jale

* commit a8eeb0997156970f7b7b6714853efbe40d6a28ae (origin/main, main)
Author: YadiraEstrada <yadraestrada@gmail.com>
Date: Thu Oct 19 01:19:43 2023 -0600

HU1-Prueba_Nuevo cambio Index

* commit 80119c30d7bdcf1e1c55a350476d590d41c2c9f1
Author: YadiraEstrada <yadraestrada@gmail.com>
Date: Thu Oct 19 01:13:38 2023 -0600

HU1-Prueba_Nuevo cambio
```

## Eliminar Canalización

Iremos a las canalizaciones y seleccionaremos la que se **desea eliminar**, posteriormente daremos clic en “**Eliminar Canalización**”

The screenshot shows the 'Canalizaciones' (Pipelines) page in Azure DevOps. At the top, there are buttons for 'Notificar', 'Ver el historial', 'Cambio de versión', and 'Eliminar la canalización'. Below these is a search bar and a table of pipelines. The table has columns for 'Nombre', 'Ejecución más reciente', 'Revisiones de código fuente más recientes', and 'Última ejecución'. One pipeline, 'test-canaliza', is highlighted with a red box. It shows a status of 'Realizado correctamente' and was executed 'hace 3 minutos'. The source is 'Merge pull request #3 from YadiraEstrada/develop'.

Nos pedirá que escribamos la palabra **delete**, lo haremos para después dar clic en el botón “**eliminar**”

**Eliminar test-canaliza?** ✕

Para confirmar la eliminación, escriba *delete* en el campo.

delete

**Eliminando recursos de detección de cambios**  
 Los siguientes recursos se utilizaron para detectar los cambios de origen y se eliminarán.

Tipo de cambio	Detalles
Eliminar	Webhook testcanaliza--Source--YadiraEstradaPrueba--1712008

No se necesitan actualizaciones de recursos para este cambio de la acción de origen.

Cancelar Eliminar

## Eliminar Entorno

Iremos a los entornos y seleccionaremos la que se **desea eliminar**, posteriormente daremos clic en “**Acciones**” y en “**Terminar entorno**”

Elastic Beanstalk > Environments

**Entornos (1)** [Información](#) 🔄 Creación de entorno

🔍 Filtrar entornos

Nombre del entorno	Estado	Nombr...	Plataf...
<span style="color: #007bff;">Phpprueba-env</span>	✔ Ok	phpprueba	PHP 8...

**Acciones** ▲

- Cargar configuración
- Guardar configuración
- Intercambiar dominio del entorno
- Clonar entorno
- Anular la operación actual
- Reiniciar los servidores de aplicaciones
- Volver a crear el entorno
- Terminar entorno
- Restaurar entorno

Nos pedirá que escribamos el **nombre del entorno**, lo haremos para después dar clic en el botón “**terminar**”

**Confirmar la terminación del entorno** ✕

¿Terminar Phpprueba-env permanentemente? Esta acción no se puede deshacer.

- Nivel: WebServer
- Plataforma: PHP 8.2 running on 64bit Amazon Linux 2023
- Versión: code-pipeline-1697702681788-38f5d3fd9edf658a6286b948dd89a433d5095c6b
- Última modificación: Octubre 19, 2023 02:05:01 (UTC-6)

**⚠ La terminación de este entorno también terminará sus recursos asociados.**

- URL:** Phpprueba-env.eba-ufdbbrp2.us-west-1.elasticbeanstalk.com se liberará.
- Recursos adicionales:** también se terminarán todos los recursos asociados a este entorno.

Escriba el nombre del entorno que quiera confirmar:

## Eliminar Instancia

Iremos a las instancias y seleccionaremos la que se **desea eliminar**, posteriormente daremos clic en “Estado de la instancia” y en “Terminar instancia”

**Instancias (1/1)** Información 🔄 Conectar Estado de la instancia ▲ Acciones ▼ Lanzar instancias ▼

🔍 Buscar instancia por atributo o etiqueta (case-sensitive)

Estado de la instancia = running ✕ Quitar los filtros

Name	ID de la instancia	Estado de la i...	Acción ...	Estado de la ...	Zona de dispon.
Phpprueba-env	i-047812747bf1a28d0	En ejecución	Terminar instancia	Terminando	Sin alarmas + us-west-1c

Nos advertirá de las consecuencias, daremos clic en el botón “terminar”

**¿Terminar instancia?** ✕

**⚠** En una instancia respaldada por EBS, la acción predeterminada se aplica al volumen de EBS raíz que se eliminará cuando se termine la instancia. El almacenamiento en las unidades locales se perderá.

¿Está seguro de que desea terminar estas instancias?

ID de la instancia	Protección de terminación
i-047812747bf1a28d0 (Phpprueba-env)	<span style="color: green;">✔ Disabled</span>

**Limpiar los recursos asociados**

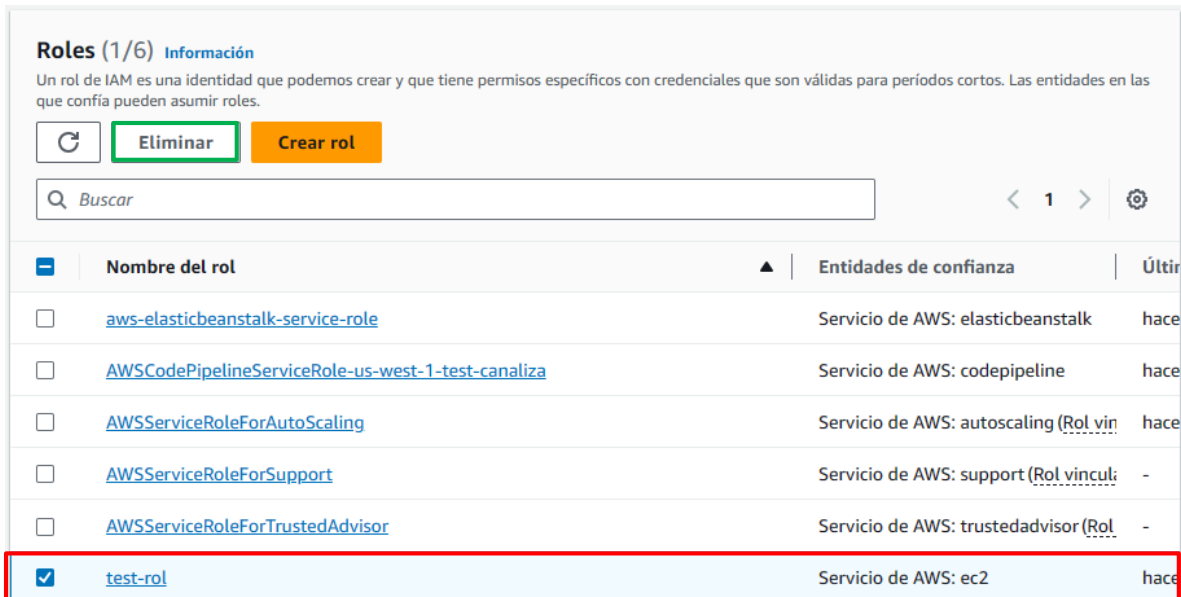
Los recursos asociados pueden generar costos cuando se terminen estas instancias.

▶ **Liberar IP elásticas asociadas**

Para confirmar que desea terminar las instancias, elija el botón de terminar que aparece a continuación. Las instancias que tengan la protección de terminación habilitada no se terminarán. La terminación de la instancia no se puede deshacer.

## Eliminar Rol

Iremos a las instancias y seleccionaremos la que se **desea eliminar**, posteriormente daremos clic en “**Eliminar**”

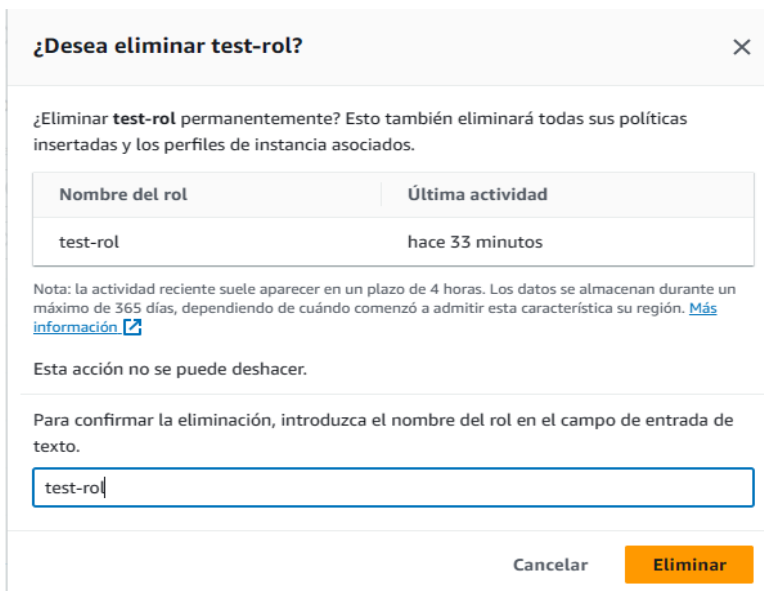


**Roles (1/6) Información**  
Un rol de IAM es una identidad que podemos crear y que tiene permisos específicos con credenciales que son válidas para períodos cortos. Las entidades en las que confía pueden asumir roles.

Buscar

<input type="checkbox"/>	Nombre del rol	Entidades de confianza	Últi
<input type="checkbox"/>	<a href="#">aws-elasticbeanstalk-service-role</a>	Servicio de AWS: elasticbeanstalk	hace
<input type="checkbox"/>	<a href="#">AWSCodePipelineServiceRole-us-west-1-test-canaliza</a>	Servicio de AWS: codepipeline	hace
<input type="checkbox"/>	<a href="#">AWSServiceRoleForAutoScaling</a>	Servicio de AWS: autoscaling (Rol vincu	hace
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	Servicio de AWS: support (Rol vincul	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	Servicio de AWS: trustedadvisor (Rol	-
<input checked="" type="checkbox"/>	<a href="#">test-rol</a>	Servicio de AWS: ec2	hace

Nos pedirá que escribamos el **nombre del rol**, lo haremos para después dar clic en el botón “**terminar**”



**¿Desea eliminar test-rol?**

¿Eliminar **test-rol** permanentemente? Esto también eliminará todas sus políticas insertadas y los perfiles de instancia asociados.

Nombre del rol	Última actividad
test-rol	hace 33 minutos

Nota: la actividad reciente suele aparecer en un plazo de 4 horas. Los datos se almacenan durante un máximo de 365 días, dependiendo de cuándo comenzó a admitir esta característica su región. [Más información](#)

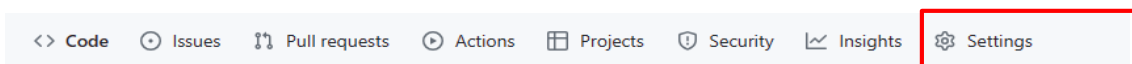
Esta acción no se puede deshacer.

Para confirmar la eliminación, introduzca el nombre del rol en el campo de entrada de texto.

test-rol

## Eliminar Repositorio

Nos iremos a la **configuración** del repositorio



<> Code Issues Pull requests Actions Projects Security Insights **Settings**

Iremos hasta abajo y buscaremos la opcion de **eliminar este repositorio**

Delete this repository

Once you delete a repository, there is no going back. Please be certain.

Delete this repository

Posteriormente, daremos clic en **deseo eliminar este repositorio**.

Delete YadiraEstrada/Prueba ×




**YadiraEstrada/Prueba**

☆ 0 stars 👁 1 watcher

I want to delete this repository


Sleccionaremos la opcion de **“He leído y entiendo las consecuencias”**

Delete YadiraEstrada/Prueba ×



**YadiraEstrada/Prueba**

☆ 0 stars 👁 1 watcher

 Unexpected bad things will happen if you don't read this!

- This will permanently delete the YadiraEstrada/Prueba repository, wiki, issues, comments, packages, secrets, workflow runs, and remove all collaborator associations.
- This will not change your billing plan. If you want to downgrade, you can do so in your Billing Settings.

I have read and understand these effects

Nos perira que escribamos el **nombre del repositorio** y eligimos la opcion de **“eliminar este repositorio”**

Delete YadiraEstrada/Prueba ×



**YadiraEstrada/Prueba**

☆ 0 stars 👁 1 watcher


To confirm, type "YadiraEstrada/Prueba" in the box below

YadiraEstrada/Prueba

Delete this repository

Nos perdira que pongamos nuestra [contraseña](#), la introduiremos y daremos clic en [confirmar](#)

Confirm access

 Signed in as @YadiraEstrada

Password

[Forgot password?](#)

**Confirm**

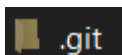
Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

Veremos que no hay ningun repositorio

Find a repository... Type ▾ Language ▾ Sort ▾ New

YadiraEstrada doesn't have any public repositories yet.

Localmente se borra eliminando la carpeta oculta `.git` que se encuentra en el directorio del proyecto local



## Conclusiones

Es sencillo preparar un entorno de desarrollo en la nube para una entrega continua más eficiente y un trabajo en equipo más unificado, teniendo en cuenta las restricciones de costos, pues aquí no se vio nada de bases de datos debido a esa razón.

## Referencias

*GitHub: Let's build from here.* (s/f).

(S/f-a). Amazon.com. Recuperado el 24 de octubre de 2023, de <https://aws.amazon.com/es/codepipeline/>

(S/f-b). Amazon.com. Recuperado el 24 de octubre de 2023, de [https://aws.amazon.com/es/elasticbeanstalk/?gclid=CjwKCAjws9ipBhB1EiwAccEi1By9xgc0dAGAQU7-UiaqXUYPRsx7oA8HreVSjbN1sVnjvcn8yKVSQRoCMKAQAvD\\_BwE&trk=ebd7e8c6-e243-44b0-be75-e840d9daf700&sc\\_channel=ps&ef\\_id=CjwKCAjws9ipBhB1EiwAccEi1By9xgc0dAGAQU7-UiaqXUYPRsx7oA8HreVSjbN1sVnjvcn8yKVSQRoCMKAQAvD\\_BwE:G:s&s\\_kwid=AL!4422!3!651612322691!e!!g!!elastic%20beanstalk!19836370531!146688820483](https://aws.amazon.com/es/elasticbeanstalk/?gclid=CjwKCAjws9ipBhB1EiwAccEi1By9xgc0dAGAQU7-UiaqXUYPRsx7oA8HreVSjbN1sVnjvcn8yKVSQRoCMKAQAvD_BwE&trk=ebd7e8c6-e243-44b0-be75-e840d9daf700&sc_channel=ps&ef_id=CjwKCAjws9ipBhB1EiwAccEi1By9xgc0dAGAQU7-UiaqXUYPRsx7oA8HreVSjbN1sVnjvcn8yKVSQRoCMKAQAvD_BwE:G:s&s_kwid=AL!4422!3!651612322691!e!!g!!elastic%20beanstalk!19836370531!146688820483)

(S/f-c). Amazon.com. Recuperado el 24 de octubre de 2023, de [https://aws.amazon.com/es/free/?gclid=CjwKCAjws9ipBhB1EiwAccEi1ND81dSoRMXGQglGeP7qjYc0J3DhUg6dAbes0ek1vzMdoDZv2P5ghhoCCgsQAvD\\_BwE&trk=ce272e14-75e2-489c-9e0a-5baac4b78cd1&sc\\_channel=ps&ef\\_id=CjwKCAjws9ipBhB1EiwAccEi1ND81dSoRMXGQglGeP7qjYc0J3DhUg6dAbes0ek1vzMdoDZv2P5ghhoCCgsQAvD\\_BwE:G:s&s\\_kwid=AL!4422!3!646879397643!e!!g!!ec2!19636894305!146667380878&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\\*all&awsf.Free%20Tier%20Categories=\\*all](https://aws.amazon.com/es/free/?gclid=CjwKCAjws9ipBhB1EiwAccEi1ND81dSoRMXGQglGeP7qjYc0J3DhUg6dAbes0ek1vzMdoDZv2P5ghhoCCgsQAvD_BwE&trk=ce272e14-75e2-489c-9e0a-5baac4b78cd1&sc_channel=ps&ef_id=CjwKCAjws9ipBhB1EiwAccEi1ND81dSoRMXGQglGeP7qjYc0J3DhUg6dAbes0ek1vzMdoDZv2P5ghhoCCgsQAvD_BwE:G:s&s_kwid=AL!4422!3!646879397643!e!!g!!ec2!19636894305!146667380878&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all)

(S/f-d). Amazon.com. Recuperado el 24 de octubre de 2023, de <https://aws.amazon.com/es/iam/>