

# JENKINS

Es un webhooks dentro de nuestro repositorio de GitHub, esto lo que hace es que cada vez que hagamos un push con github se haga una compilación con el Jenkins ejecutando el DockerFile con el Docker compose y al compilarla te lo suben a Docker hub.

Tenemos que crear un directorio ci-cd para Jenkins.

Lo que despliega esto es el jenkinsfile. Esta versión de Jenkins que usaremos servirá para generar PODs automáticos para dejar libre el Jenkinsfile original. Cuando acaba de ejecutar los Jenkins file, se eliminan.

Ahora ejecutamos los siguientes comandos en la VPS para añadir el repo de Helm

**“Helm repo add Jenkins <https://charts.jenkins.io>”**

```
jiahaolin@vmi2812274:~/jiahaohlc/docker/caronte/proyecto/personal/mi-portfolio-helm$ helm repo add jenkins https://charts.jenkins.io
"jenkins" has been added to your repositories
```

Y actualizamos los repositorios de helm

**“un helm repo update”**

```
jiahaolin@vmi2812274:~/jiahaohlc/docker/caronte/proyecto/personal/mi-portfolio-helm$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jenkins" chart repository
Update Complete. #Happy Helming!#
```

Creamos un namespace

**“kubectl create namespace <<nombre>>”**

```
jiahaolin@vmi2812274:~/jiahaohlc/docker/caronte/proyecto/personal/mi-portfolio-helm$ kubectl create namespace jenkins
namespace/jenkins created
```

Creamos una carpeta en Docker y iniciamos un helm

**“helm create Jenkins”**

```
jiahaolin@vmi2812274:~/jiahaohlc/docker/caronte$ cd ..
jiahaolin@vmi2812274:~/jiahaohlc/docker$ mkdir jenkins
jiahaolin@vmi2812274:~/jiahaohlc/docker$ helm create jenkins
Creating jenkins
```

Borramos todo el contenido del directorio templates

```
jenkins$ rm -rf ./templates/*
```

Lo subimos a github para hacer un pull para llevar los archivos en nuestro equipo local ya que no trabajaremos directamente desde la VPS.

Modificamos el chart.yml con los siguientes contenidos

```
apiVersion: v2
name: jenkins
description: Jenkins CI wrapper chart
type: application
version: 0.1.0

dependencies:
- name: jenkins
  version: 5.8.7
  repository: https://charts.jenkins.io
```

Actualizamos la dependencias

## Helm dependency update

```
jenkins$ cd templates/
jenkins$ ls
Chart.yaml  charts  templates  values.yaml
jenkins$ cd ..
jenkins$ helm dependency update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jenkins" chart repository
Update Complete. ✎ Happy Helming! ✎
Saving 1 charts
Downloading jenkins from repo https://charts.jenkins.io
Deleting outdated charts
jenkins$
```

Y si descomprimos el fichero que nos ha dado dará los siguientes ficheros

**Tar -xzvf ./charts/Jenkins-5.8.7.tgz**

```
jiahaojin@vmi2812274:~/jiahaojin/docker/jenkins$ tar -xzf ./charts/jenkins-5.8.7.tgz
jenkins/Chart.yaml
jenkins/values.yaml
jenkins/templates/NOTES.txt
jenkins/templates/_helpers.tpl
jenkins/templates/auto-reload-config.yaml
jenkins/templates/config-init-scripts.yaml
jenkins/templates/config.yaml
jenkins/templates/deprecation.yaml
jenkins/templates/home-pvc.yaml
jenkins/templates/jcasc-config.yaml
jenkins/templates/jenkins-agent-svc.yaml
jenkins/templates/jenkins-aws-security-group-policies.yaml
jenkins/templates/jenkins-controller-alerting-rules.yaml
jenkins/templates/jenkins-controller-backendconfig.yaml
jenkins/templates/jenkins-controller-ingress.yaml
jenkins/templates/jenkins-controller-networkpolicy.yaml
jenkins/templates/jenkins-controller-pdb.yaml
jenkins/templates/jenkins-controller-podmonitor.yaml
jenkins/templates/jenkins-controller-route.yaml
jenkins/templates/jenkins-controller-secondary-ingress.yaml
jenkins/templates/jenkins-controller-servicemonitor.yaml
jenkins/templates/jenkins-controller-statefulset.yaml
jenkins/templates/jenkins-controller-svc.yaml
jenkins/templates/rbac.yaml
jenkins/templates/secret-additional.yaml
jenkins/templates/secret-claims.yaml
jenkins/templates/secret-https-jks.yaml
jenkins/templates/secret.yaml
```

Ahora realizamos las añadimos las siguientes variables de entorno en el archivo values.yml (copiado desde el repositorio de morgado)

```
jenkins > | values.yaml
1 controller:
2   image:
3     repository: jenkins/jenkins
4     tag: "2.541.1-lts-jdk21"
5     runtime: 1000
6     fsGroup: 1000
7   admin:
8     username: admin
9     password: admin123
10  persistence:
11    enabled: true
12    size: 8Gi
13    storageClass: microk8s-hostpath # clave
14  serviceType: clusterip
15  sidecars:
16    configAutoload:
17      enabled: true
18      image:
19        repository: k1u1grid/k8s-sidecar
20        tag: "1.27.4"
21  ingress:
22    enabled: true
23    ingressClassName: nginx
24    hostname: jenkins.jamorgado.es
25    annotations:
```

Y modificamos los siguientes parámetros

```
21 ingress:
22   enabled: true
23   ingressClassName: nginx
24   hostname: jenkins.megacrossover.es
25   annotations:
26     cert-manager.io/cluster-issuer: letsencrypt-prod
27     nginx.ingress.kubernetes.io/ssl-redirect: "true"
28   tls:
29     - secretName: jenkins-tls
30     hosts:
31       - jenkins.megacrossover.es
32 additionalPlugins: []
33 installPlugins:
34   - configuration-as-code
```

```

> projecto
└─ jenkins
   └─ charts
      └─ jenkins-5.8.7.tgz
      └─ jenkins
         └─ templates
            CHANGELOG.md
            ! Chart.yaml
            ! README.md
            ! UPGRADING.md
            ! VALUES.md
            VALUES.md.gotmpl
            ! values.yaml
         templates
         .helmignore
         Chart.lock
         ! Chart.yaml
         ! values.yaml

```

## "helm install jenkins jenkins/jenkins -n kenkins -f values.yaml"

```
jiahao@vm1212274: /jenkins$ helm install jenkins jenkins/jenkins -n jenkins -f values.yaml
NAME: jenkins
LAST DEPLOYED: Tue Feb 17 13:34:06 2026
NAMESPACE: jenkins
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
NOTES:
1. Get your 'admin' user password by running:
   kubectl exec --namespace jenkins -it svc/jenkins -c jenkins -- /bin/cat /run/secrets/additional/chart-admin-password && echo
2. Visit https://jenkins.megacrossover.es

3. Login with the password from step 1 and the username: admin
4. Configure security realm and authorization strategy
5. Use Jenkins Configuration as Code by specifying configScripts in your values.yaml file, see documentation: https://jenkins.megacrossover.es/configuration-as-code and examples: https://github.com/jenkinsci/configuration-as-code-plugin/tree/master/demos

For more information on running Jenkins on Kubernetes, visit:
https://cloud.google.com/solutions/jenkins-on-container-engine

For more information about Jenkins Configuration as Code, visit:
https://jenkins.io/projects/jcasc/

NOTE: Consider using a custom image with pre-installed plugins
```

Y vemos que furula con el comando de mai lova

```
jiahaolin@vmi2812274:~/jiahaohc/docker/jenkins$ kubectl get pods -n jenkins -w
```

NAME	READY	STATUS	RESTARTS	AGE
cm-acme-http-solver-qszhx	1/1	Running	0	70s
jenkins-0	0/2	Init:1/2	0	73s
jenkins-0	0/2	PodInitializing	0	74s
jenkins-0	1/2	Running	0	76s

## Añadimos una entrada DNS en nuestro dominio

Eliminar

Entrada DNS

Tipo

Valor

Modificar

<input type="checkbox"/>	176184686523...domainkey.megacrossover.es	TXT	ZZH2KP72e3O4OkhRTH48GfPTt4567JcL-SRK...aBefFC0nqTAjltcvWdy8T8UfMR2NBu3p67hdj8Z0z4+V0/h3x193MULksObFv5Q5MQJDgyluJk4JDQ9HbffGA1K2z3w0zROZYDhUWWMYmmvCM2SEzQjDA	
<input type="checkbox"/>	autoconfig.megacrossover.es	CNAME	autoconfig.serviciodecorreo.es	
<input type="checkbox"/>	autodiscover.megacrossover.es	CNAME	autodiscover.serviciodecorreo.es	
<input type="checkbox"/>	control.megacrossover.es	CNAME	pdcservidoresdns.net	
<input type="checkbox"/>	megacrossover.es	SPF	v=spf1 include:_spf.serviciodecorreo.es ~all	
<input type="checkbox"/>	megacrossover.es	A	194.163.147.140	
<input type="checkbox"/>	megacrossover.es	MX 10	mx.serviciodecorreo.es	

Añadir entrada DNS

La entrada jenkins.megacrossover.es se ha añadido correctamente

## Accedemos a la página web

jenkins.megacrossover.es/login?from=%2F

### Sign in to Jenkins

Username

Contraseña

☐ Keep me signed in

Sign in

## Y estamos dentro de la página web

Jenkins

+ Nueva Tarea

Historial de trabajos

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones 0/0

¡Bienvenido a Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

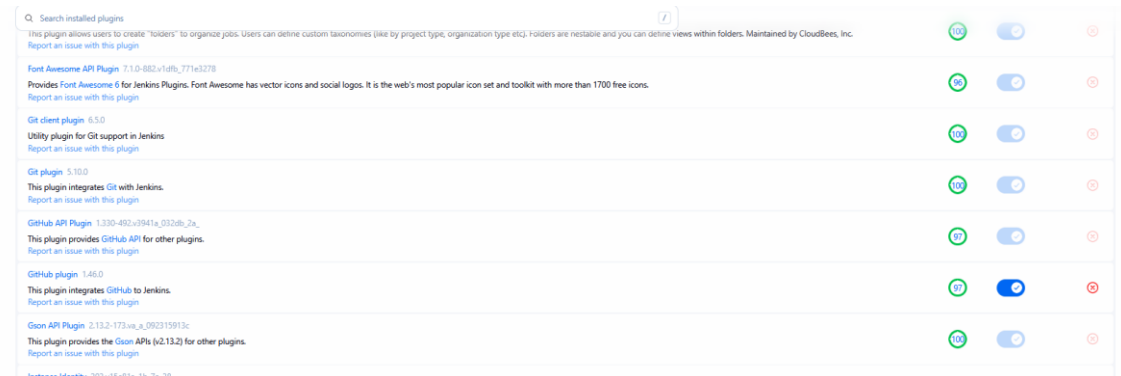
Start building your software project

Create a job

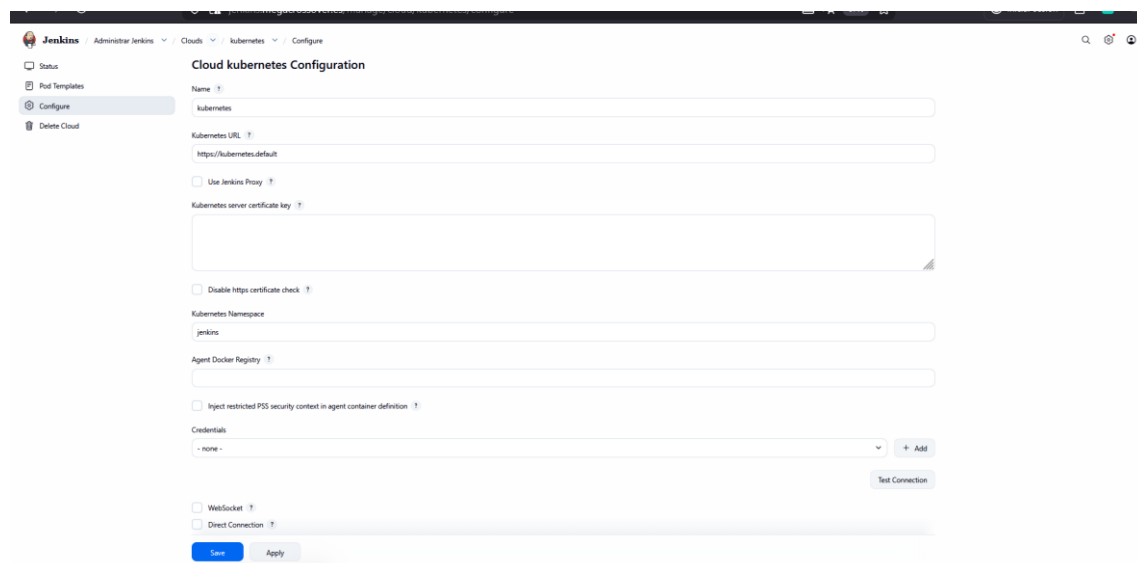
## Entramos en administración



Y vemos los plugins que tenemos activado que en este caso veremos el de github

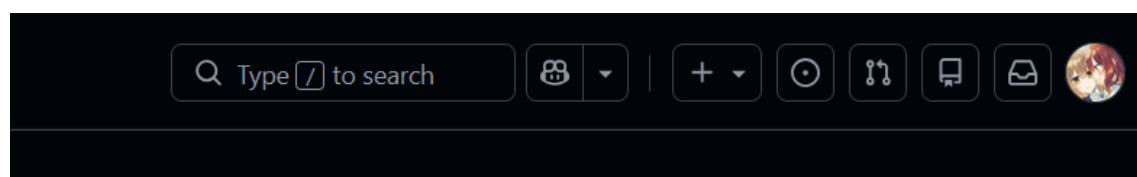


En CLOUD kubernetes tenemos la información del kubernetes

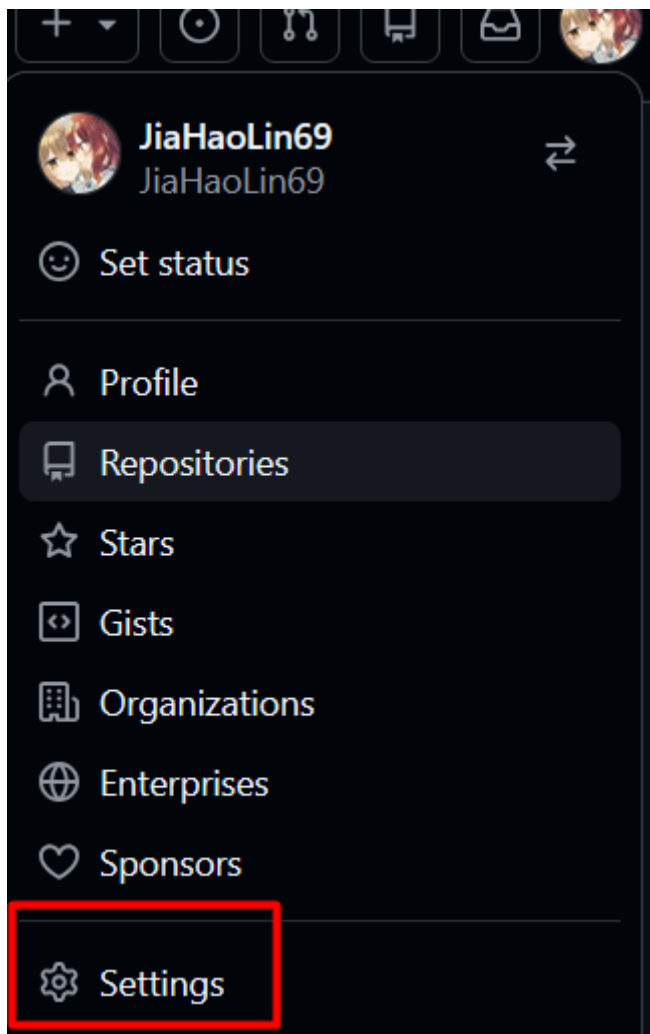


Ahora vamos a crear las credenciales en Jenkins, PARA ELLO, tenemos que generar un token.

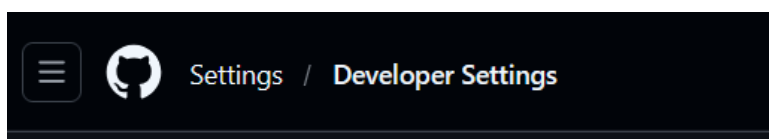
Accedemos a nuestro GitHub



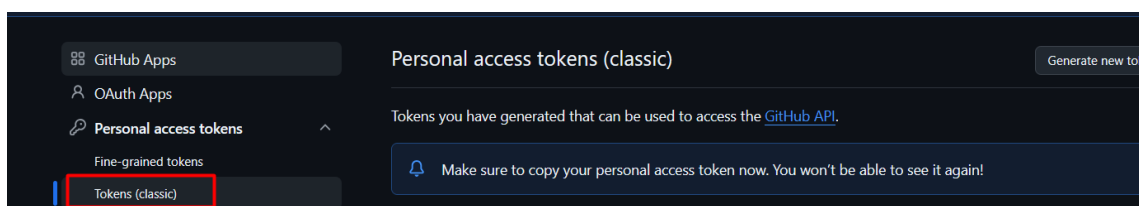
Entramos en la configuración de usuario de github



Vamos al developer settings



Generamos un token classic



## New personal access token (classic)

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](#).

**Note**

Jenkins

What's this token for?

**Expiration**

7 days (Feb 25, 2026)

The token will expire on the selected date

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<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 type="checkbox"/> workflow	Update GitHub Action workflows

<input checked="" type="checkbox"/> <b>admin:repo_hook</b>	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks

Creamos en la carpeta common un token.txt con el token de nuestro github recién creado

```
caronte > common > token.txt
1 github: ghp_mYiF3rCg5KLE3iW0fDvij2IaRquxr0zAgYm
```

The screenshot shows a terminal window with a file explorer on the left. The file explorer shows a directory structure with 'common' containing 'token.txt'. The terminal shows the command 'token.txt' being executed, resulting in a file with one line of content: '1 github: ghp\_mYiF3rCg5KLE3iW0fDvij2IaRquxr0zAgYm'.



Una vez generado el token creamos las credenciales, en password ponemos la credencial

<

Add Username with password

×

Scope ?

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

Username

JiaHaoLin69

☐ Treat username as secret ?

Password

.....

ID ?

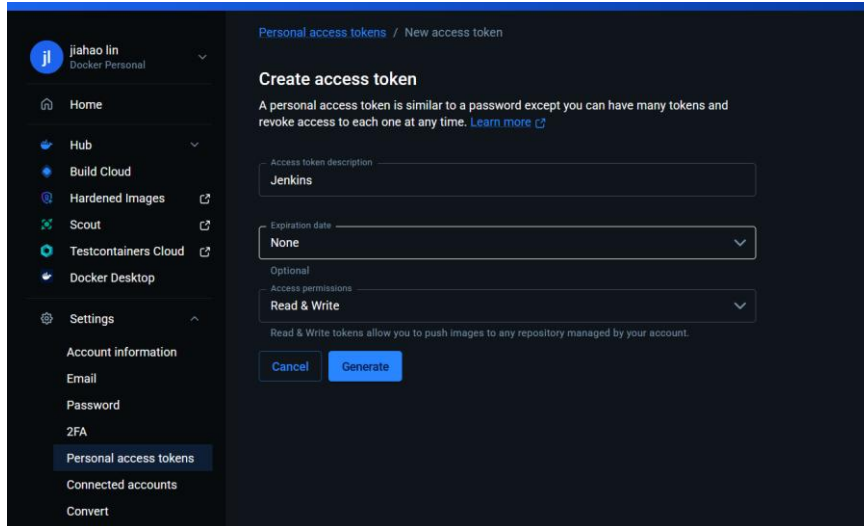
github-cred

Description ?

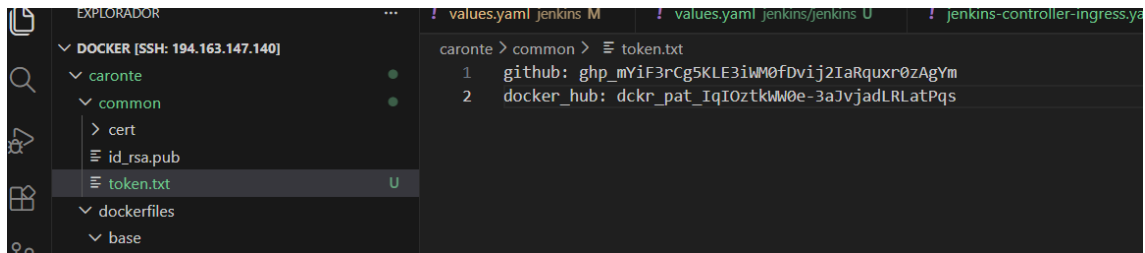
Create

## Docker HUB

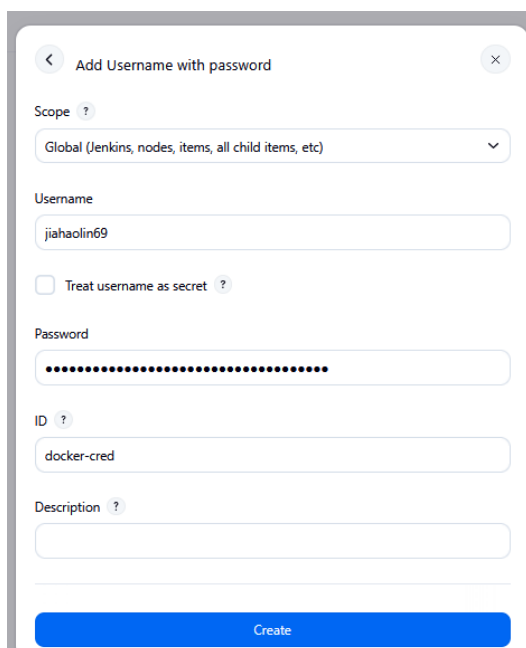
Usuario → settings → personal Access token

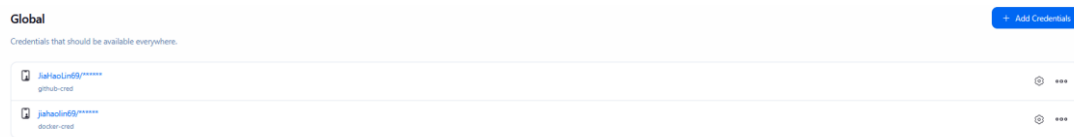


Lo copiamos en token.txt



Y creamos la credencial en Jenkins

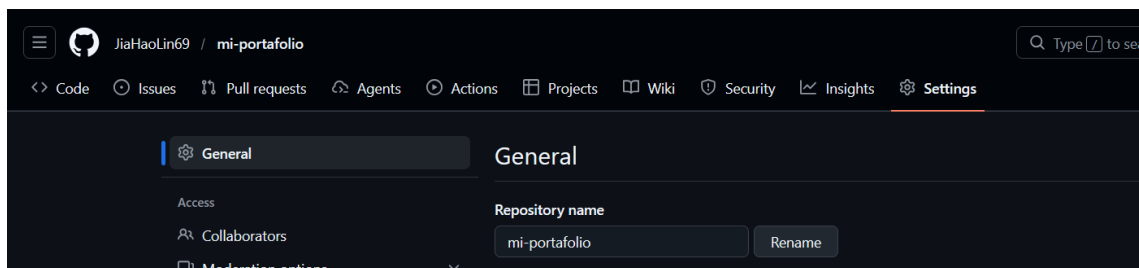




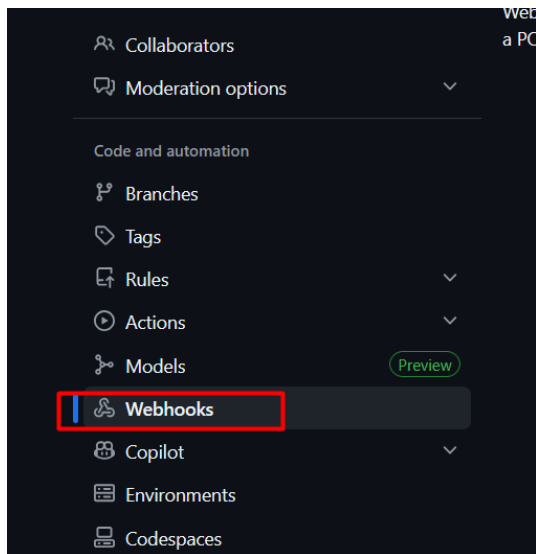
## IMPORTANTE

Ahora si reventamos un POD y esta configuración no se pierde, porque se crea una réplica de manera automática en el que se guarda estas configuraciones

Una vez hecho esto vamos al repositorio de portfolio. Pinchamos en setting



Vamos al webhook



Añadimos un webhook (es una especie de canal que sirve para conectar el repositorio de github con jenkins)

**Webhooks / Add webhook**

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, etc). More information can be found in [our developer documentation](#).

**Payload URL \***

`https://jenkins.megacrossover.es/github-webhooks/`

**Content type \***

`application/x-www-form-urlencoded`

**Secret**

**Content type \***

`application/x-www-form-urlencoded`

**Secret**

**SSL verification**

By default, we verify SSL certificates when delivering payloads.

☒ **Enable SSL verification** ☐ **Disable (not recommended)**

**Which events would you like to trigger this webhook?**

☒ Just the push event.

☐ Send me **everything**.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

**Add webhook**

Y lo añadimos

**Webhooks** **Add webhook**

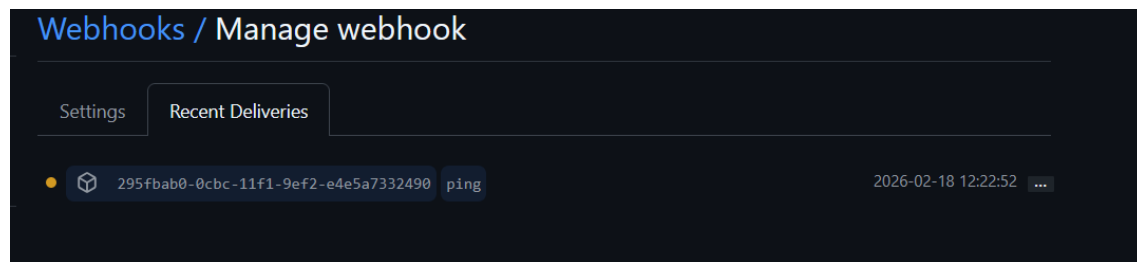
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

- `https://jenkins.megacrossover.es/git...` (*push*)

This hook has never been triggered.

**Edit** **Delete**

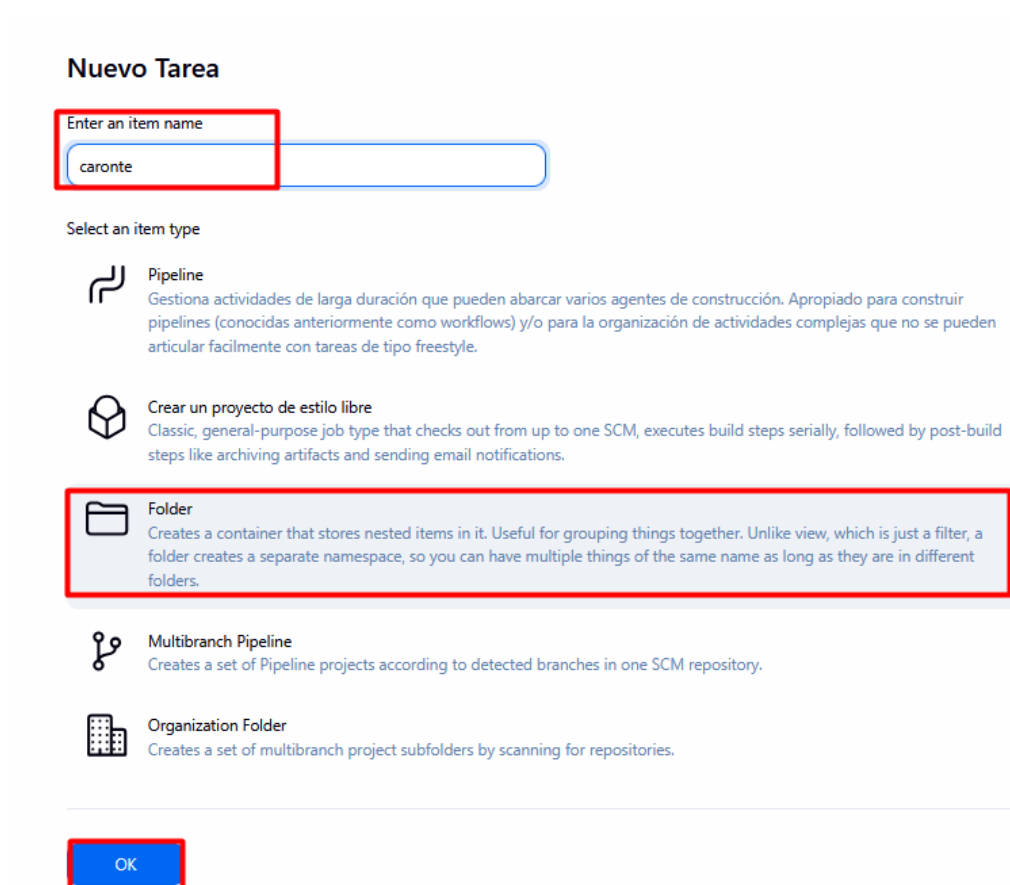
Entramos y vamos al apartado de recent deliveries



Creamos una nueva tarea



Creamos un folder




## Creamos otra carpeta dentro de Caronte

### Nuevo Tarea


Enter an item name

Artesa


Select an item type




**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 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.



**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.


If you want to create a new item from other existing, you can use this option:

Copy from

Type to autocomplete

OK

## Creamos estos 3

 Jenkins / caronte

Status

Configure

+ New Item

Delete Folder

Historial de trabajos

Rename

Credentials




Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones 0/0


caronte

All +

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		Artesa	N/D	N/D	N/D
		Buandander	N/D	N/D	N/D
		portfolio	N/D	N/D	N/D

Iconos: S M L

# Creamos una pipeline

 Jenkins

caronte

portfolio

Status

Configure

+ New Item

Delete Folder

Historial de trabajos

Move

Rename

Credentials

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

0/0

portfolio

Full folder name: caronte/portfolio

All

This folder is empty

Create a job