

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<>”

```
update complete: *happy helming!*
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

```
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins$ rm -rf ./templates/*
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins$
```

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.yaml 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

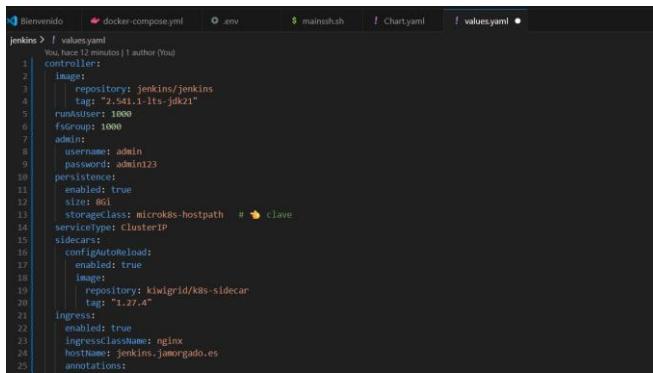
```
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins$ cd templates/
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins/templates$ ls
Chart.yaml  charts  templates  values.yaml
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins/templates$ cd ..
jiahaoLIN@vmi2812274:~/jiahaoHLC/docker/jenkins$ helm dependency update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jenkins" chart repository
Update complete. #Happy Helm-ing!
Saving 1 charts
Downloading jenkins from repo https://charts.jenkins.io
Deleting outdated charts
194.163.147.140  You, hace 4 minutos  JiaHaoLin69 (Hace 4 minutos)  Lín. 1, Col.
```

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

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

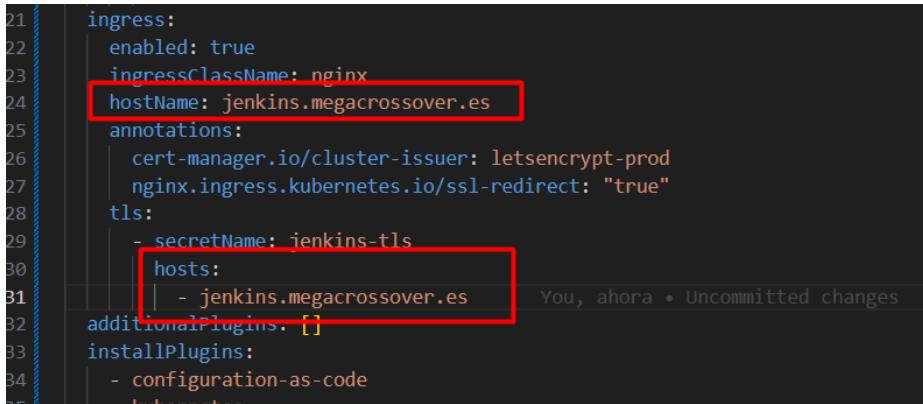
```
● jiahaolin@vmi2812274:~/jiahaohlc/docker/jenkins$ tar -xzvf ./charts/jenkins-5.8.7.tgz
jenkins/Chart.yaml
jenkins/values.yaml
jenkins/templates/NOTES.txt
jenkins/templates/_helpers.tpl
jenkins/templates/auto-reload-config.yaml
jenkins/templates/config-init-scripts.yaml
jenkins/templates/config.yaml
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.yaml (copiado desde el repositorio de morgado)



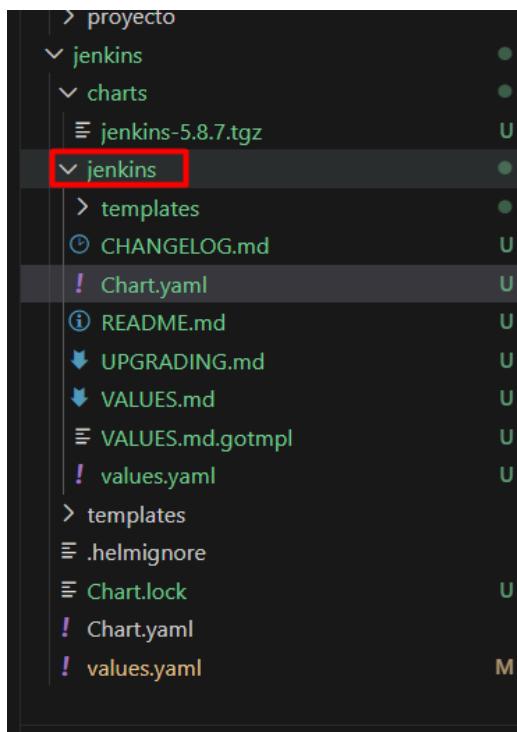
```
jenkins > [values.yaml]
You have 17 minutes | 1 author (Visu)
1 controller:
2   image:
3     repository: jenkins/jenkins
4     tag: "2.54.1-lts-jdk21"
5   runAsUser: 1000
6   fsGroup: 1000
7   admin:
8     username: admin
9     password: admin123
10    persistence:
11      enabled: true
12      size: 8G
13      storageClass: microk8s-hostpath # 🌐 clave
14      serviceType: ClusterIP
15    sidecars:
16      - configToReload:
17        enabled: true
18        image:
19          repository: 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
35     - kubernetes
```

Dentro de este directorio



Para levantar esto nos metemos al directorio jenkins y ejecutamos

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

```
jiahao@vmi2812274:~/jiahao/hc/docker/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
  https://jenkinsci.github.io/jenkinsci-image/
```

Comprobamos con ver el pods (kubectl get pods -n jenkins -w)

Y vemos que furula con el comando de mai lova

```
jiahao@vmi2812274:~/jiahao/hc/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

The screenshot shows a list of DNS entries for the domain megacrossover.es. A modal window titled 'Añadir entrada DNS' (Add DNS entry) is open, displaying a success message: 'La entrada jenkins.megacrossover.es se ha añadido correctamente' (The entry jenkins.megacrossover.es has been added correctly). The modal also contains a copy icon. The table below lists other entries:

Entrada DNS	Tipo	Valor	Modificar
1761846866523._domainkey.megacrossover.es	TXT	ZZH1j03ZeZodO6JrRTH48GfPTI4567JCrL+SFK... (long hex string)	
autoconfig.megacrossover.es	CNAME	autoconfig.serviciodecorreo.es	
autodiscover.megacrossover.es	CNAME	autodiscover.serviciodecorreo.es	
control.megacrossover.es	CNAME	pdc.servidoresdns.net	
megacrossover.es	SPF	v=spf1 include:_spf.serviciodecorreo.es ~all	
megacrossover.es	A	194.163.147.140	
megacrossover.es	MX 10	mx.serviciodecorreo.es	

Accedemos a la página web

The screenshot shows a web browser with the URL 'jenkins.megacrossover.es/login?from=%2F'. The page features a large background image of a smiling cartoon character in a tuxedo. On the right side, there is a 'Sign in to Jenkins' form with fields for 'Username' and 'Contraseña', a 'Keep me signed in' checkbox, and a blue 'Sign in' button.

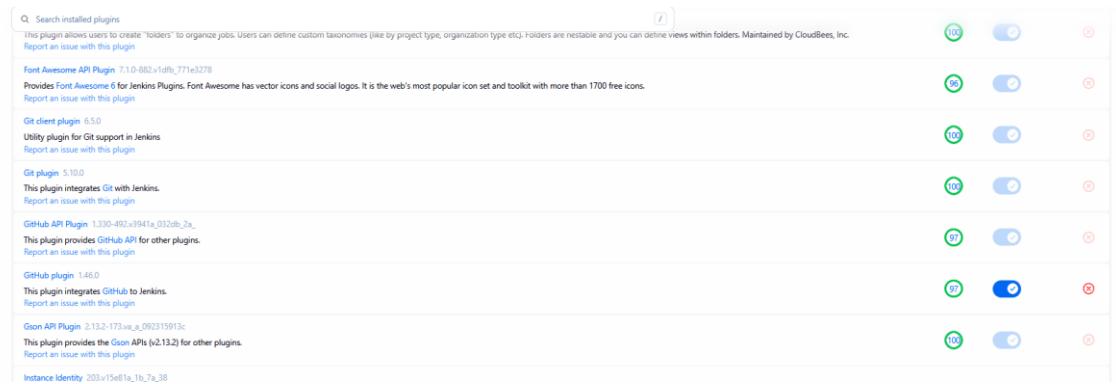
Y estamos dentro de la página web

The screenshot shows the Jenkins dashboard at 'jenkins.megacrossover.es'. The left sidebar includes links for 'Nueva Tarea' and 'Historial de trabajos'. The main area displays a '¡Bienvenido a Jenkins!' message and a summary of job status: 'Trabajos en la cola: No hay trabajos en la cola' and 'Estado del ejecutor de construcciones: 0/0'. There is a 'Create a job' button and a '+' icon.

Entramos en administración



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



Q Search installed plugins

This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.

Report an issue with this plugin

Font Awesome API Plugin 7.1.0-882.v1dfb_771e3278

Provides Font Awesome 5 for Jenkins Plugins. Font Awesome has vector icons and social logos. It is the web's most popular icon set and toolkit with more than 1700 free icons.

Report an issue with this plugin

Git client plugin 6.5.0

Utility plugin for Git support in Jenkins

Report an issue with this plugin

Git plugin 5.10.0

This plugin integrates Git with Jenkins.

Report an issue with this plugin

Github API Plugin 1.330-492.v3941a_032db_2a

This plugin provides GitHub API for other plugins.

Report an issue with this plugin

GitHub plugin 1.46.0

This plugin integrates GitHub to Jenkins.

Report an issue with this plugin

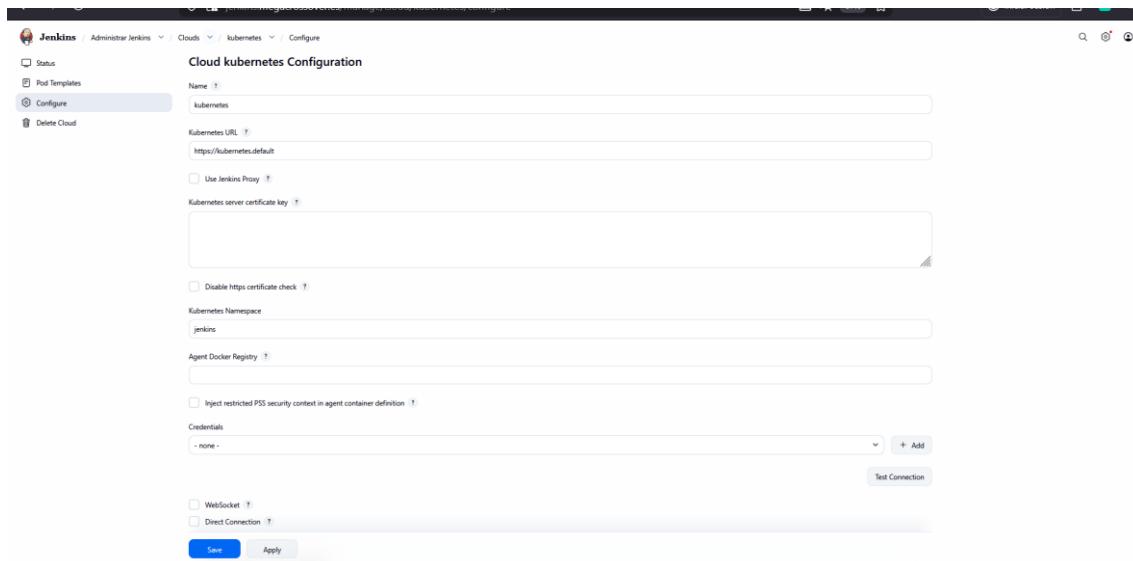
Gson API Plugin 2.12.2-173.vca_a_092315913c

This plugin provides the Gson APIs (v2.13.2) for other plugins.

Report an issue with this plugin

Instance Identity 203v15e81a_1b_7a_38

En CLOUD kubernetes tenemos la información del kubernetes



Jenkins / Administrar Jenkins / Clouds / Kubernetes / Configure

Status

Pod Templates

Configure

Delete Cloud

Cloud kubernetes Configuration

Name: kubernetes

Kubernetes URL: https://kubernetes.default

Use Jenkins Proxy

Kubernetes server certificate key

Disable https certificate check

Kubernetes Namespace: jenkins

Agent Docker Registry:

Inject restricted PSS security context in agent container definition

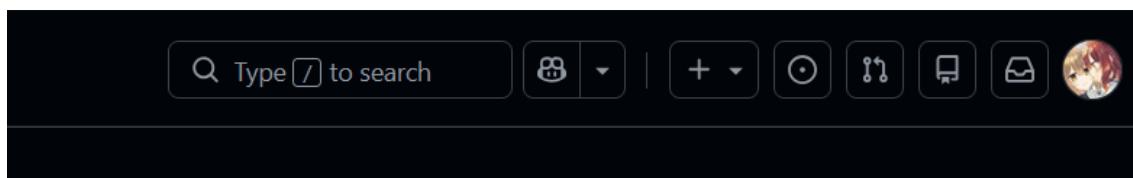
Credentials: - none -

Test Connection

Save Apply

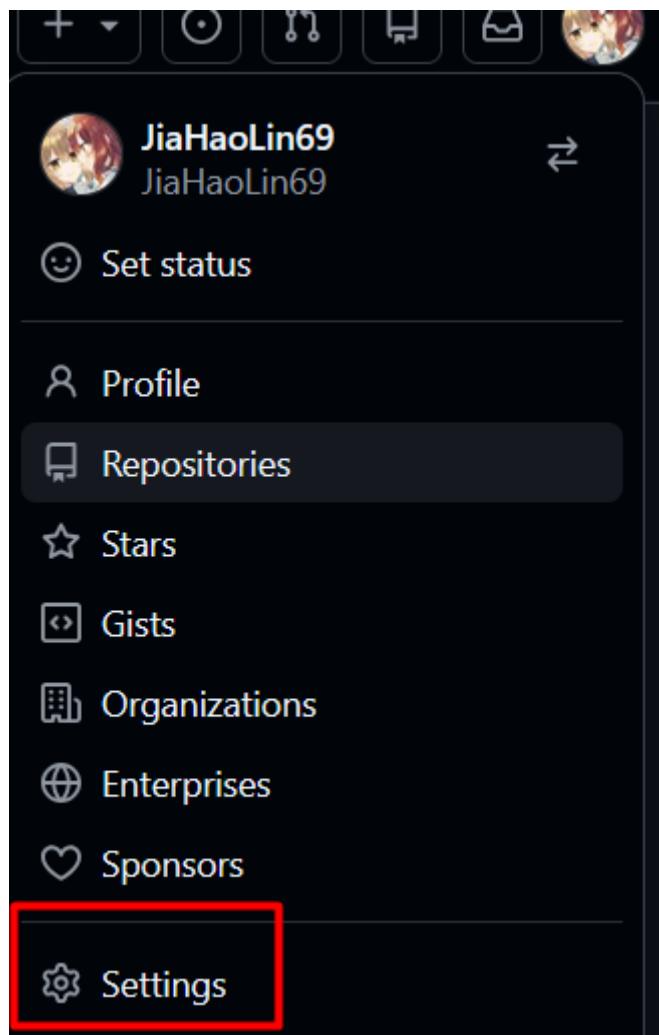
Ahora vamos a crear las credenciales en Jenkins, PARA ELLO, tenemos que generar un token.

Accedemos a nuestro GitHub

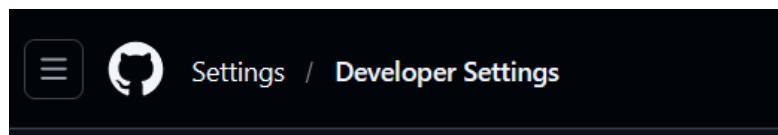


Q Type ⌂ to search

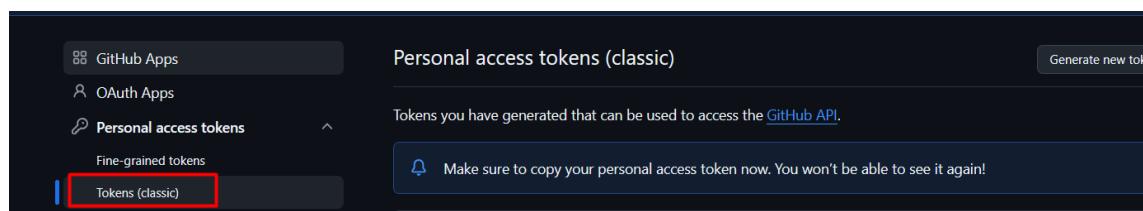
Entramos en la configuración de usuario de github



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

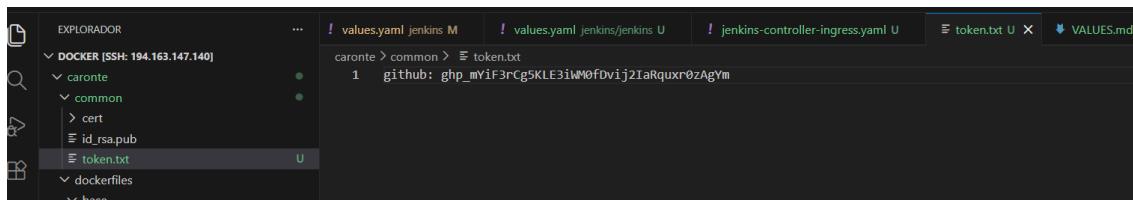
7 days (Feb 25, 2026) ▾
The token will expire on the selected date

Select scopes

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

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

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



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

The screenshot shows the Jenkins 'Add Username with password' credential configuration screen. The form includes fields for Scope (set to 'Global'), Username ('JiaHaoLin69'), Password (represented by a series of dots), ID ('github-cred'), and Description (empty). A 'Treat username as secret' checkbox is also present. A blue 'Create' button is at the bottom.

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

The screenshot shows the 'Personal access tokens / New access token' page. On the left, there's a sidebar with options like Home, Hub, Build Cloud, Hardened Images, Scout, Testcontainers Cloud, Docker Desktop, Settings (which is selected), Account information, Email, Password, 2FA, Personal access tokens (which is also selected), Connected accounts, and Convert. The main area has fields for 'Access token description' (set to 'Jenkins'), 'Expiration date' (set to 'None'), and 'Access permissions' (set to 'Read & Write'). Below these are 'Cancel' and 'Generate' buttons.

Lo copiamos en token.txt

The screenshot shows a terminal window with three tabs: 'values.yaml jenkins M', 'values.yaml jenkins/jenkins U', and 'jenkins-controller-ingress.ya'. The first tab contains a file with two lines: 'github: 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' and 'docker_hub: dckr_pat_IqIOztkWW0e-3aJvjadLRLatPqs'. The second tab shows the same content. The third tab is partially visible.

Y creamos la credencial en Jenkins

The screenshot shows the 'Add Username with password' dialog. It has fields for 'Scope' (set to 'Global (Jenkins, nodes, items, all child items, etc)'), 'Username' (set to 'jiahaolin69'), 'Treat username as secret' (unchecked), 'Password' (a redacted string), 'ID' (set to 'docker-cred'), and 'Description' (an empty field). At the bottom is a large blue 'Create' button.

Global

Credentials that should be available everywhere.

+ Add Credentials

github-cred	...
jenkins-cred	...

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 *

Content type *

Secret

Content type *

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](#).

<input checked="" type="radio"/> https://jenkins.megacrossover.es/git... (push)	Edit	Delete
This hook has never been triggered.		

Entramos y vamos al apartado de recent deliveries

The screenshot shows the Jenkins Webhooks / Manage webhook interface. At the top, there are two tabs: 'Settings' and 'Recent Deliveries'. The 'Recent Deliveries' tab is selected, highlighted by a blue border. Below the tabs, a single delivery entry is listed. The entry includes a small icon of a box with a gear, the ID '295fbab0-0cbc-11f1-9ef2-e4e5a7332490', the word 'ping', the timestamp '2026-02-18 12:22:52', and three dots for more options.

Creamos una nueva tarea

The screenshot shows the Jenkins home page. At the top, there is a Jenkins logo and the word 'Jenkins'. Below the logo, there are two main buttons: a red-bordered 'Nueva Tarea' button and a 'Historial de trabajos' button. A horizontal bar is visible above the buttons.

Creamos un folder

The screenshot shows the 'Nuevo Tarea' creation dialog. At the top, there is a text input field labeled 'Enter an item name' containing the text 'caronte', which is also highlighted with a red box. Below the input field, there is a section titled 'Select an item type' with several options:

- Pipeline**: Describes activities of long duration that can involve multiple construction agents. Appropriate for building pipelines (formerly known as workflows) and/or for organizing complex activities that cannot be easily articulated with freestyle tasks.
- Crear un proyecto de estilo libre**: Describes a 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**: Describes a container that stores nested items. 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. This option is also highlighted with a red box.
- Multibranch Pipeline**: Describes a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Describes a set of multibranch project subfolders by scanning for repositories.

At the bottom of the dialog, there is a blue 'OK' button.

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

The screenshot shows the Jenkins 'caronte' dashboard. On the left, there's a sidebar with options like Status, Configure, New Item, Delete Folder, Rename, Credentials, and Trabajos en la cola. The main area shows a table with three items:

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		Artesa	N/D	N/D	N/D
		Bautander	N/D	N/D	N/D
		portolio	N/D	N/D	N/D

At the bottom, there's a footer with icons for S, M, and L.

Creamos una pipeline

The screenshot shows the Jenkins web interface for the 'portfolio' folder under the 'caronte' user. The top navigation bar includes the Jenkins logo, the user name 'caronte', and the current path '/portfolio'. On the left, a sidebar lists various management options: Status, Configure, New Item, Delete Folder, Historial de trabajos, Move, Rename, and Credentials. The main content area displays the 'portfolio' folder icon and the message 'Full folder name: caronte/portfolio'. A note states 'This folder is empty'. At the bottom right, there is a 'Create a job' button with a '+' sign. Two dropdown menus at the bottom provide filtering for 'Trabajos en la cola' (No hay trabajos en la cola) and 'Estado del ejecutor de construcciones' (0/0).