

Server Apache & GitLab - CI/CD

1 - Ejecutamos el siguiente comando para correr nuestro contenedor GitLab

`docker run -d gitlab/gitlab-ce`

```
matl@matl-virtual-machine:~$ docker run -d gitlab/gitlab-ce
3e44dd25ad6c379bbe2152045d49057f19779b83a998f50f1552dec4ccf01121
matl@matl-virtual-machine:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
3e44dd25ad6c   gitlab/gitlab-ce  "/assets/wrapper"       7 seconds ago  Up 5 seconds (health: starting)  22/tcp, 80/tcp, 443/tcp           pensive_ishizaka
matl@matl-virtual-machine:~$
```

Verificamos que se está ejecutando con el comando:

`docker ps`

2 - Una vez en ejecución utilizamos el siguiente comando para poder ingresar al contenedor y realizar algunas verificaciones:

`docker exec -it <contenedor ID> bash`

Para obtener la IP de Gitlab : `hostname -i`

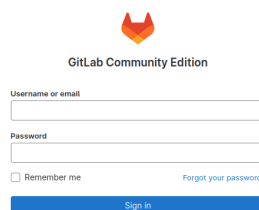
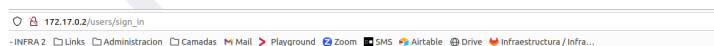
Para obtener la clave root de Gitlab: `cat /etc/gitlab/initial_root_password`

```
root@3e44dd25ad6c:/# hostname -i
172.17.0.2
root@3e44dd25ad6c:/# cat /etc/gitlab/initial_root_password
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initial_root_password']' setting in 'gitlab.rb', it was provided before database was seeded for the first time (usually, the first reconfigure run).
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com/ee/security/reset_user_password.html#reset-your-root-password.
Password: 956casEdKpIShGqyIC35TCHP6rwo5CLc+0ShwCMwNwB=
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
root@3e44dd25ad6c:/#
```

Podemos verificar que ingresando a la ip que nos arrojó el comando anterior, podemos ingresar a nuestro servidor. Rellenamos los campos:

Username: `root`

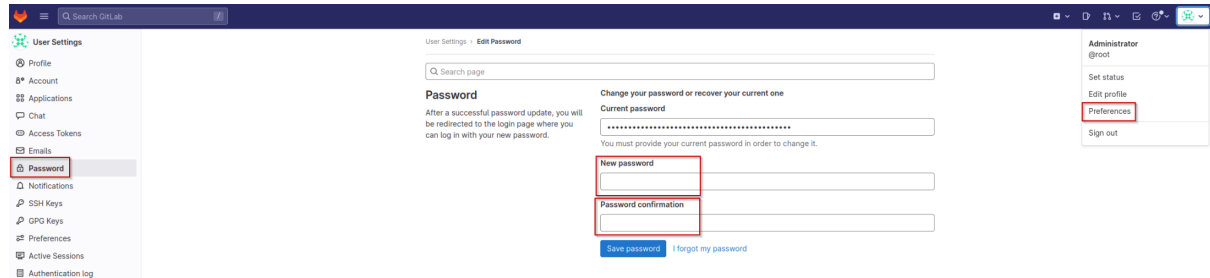
Password: *el que obtuvimos anteriormente con el comando*



Recomiendo que cambien dicho password por uno más simple en caso de que lo vayamos usar a este ejercicio de prueba (para una mayor comodidad) desde la siguiente ruta:

Perfil>Preferencias>Password

Adjunto captura del lugar exacto



Una vez modificada la clave, nos va a desloguear y volvemos a colocar las credenciales nuevas.

3 - Creamos un nuevo proyecto desde el botón  y seteamos los parámetros según nuestros requisitos. Pongo ejemplo los que utilicé yo:



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

New project > Create blank project

Project name

my-web

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

http://172.17.0.2/ root

Project slug

my-web

Want to organize several dependent projects under the same namespace? [Create a group](#).

Visibility Level [?](#)

☒ Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

☐ Internal

The project can be accessed by any logged in user except external users.

☐ Public

The project can be accessed without any authentication.

Project Configuration

☐ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project Cancel

Nos va a llevar a una página muy similar a la siguiente, en donde vamos a tener nuestro proyecto (vacío de momento).

Administrator > my-web

Project 'my-web' was successfully created.

my-web
Project ID: 2

Invite your team
Add members to this project and start collaborating with your team.
[Invite members](#)

The repository for this project is empty
You can get started by cloning the repository or start adding files to it with one of the following options.

[Clone](#) [Upload File](#) [New file](#) [Add README](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#)
[Add Wiki](#) [Configure Integrations](#)

Command line instructions
You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Administrator"
git config --global user.email "admin@example.com"
```

Create a new repository

```
git clone http://3e44dd25ad6c/root/my-web.git
cd my-web
git switch -c main
touch README.md
git add README.md
git commit -m "add README"
git push -u origin main
```

Push an existing folder

```
cd existing_folder
git init --initial-branch=main
git remote add origin http://3e44dd25ad6c/root/my-web.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

Push an existing Git repository

Guardamos el link para poder clonar nuestro repositorio más adelante.

[Clone](#) [Upload File](#) [New file](#)

Clone with SSH
git@3e44dd25ad6c:root/my-web.git

Clone with HTTP
http://3e44dd25ad6c/root/my-web

Open in your IDE

- Visual Studio Code (SSH)
- Visual Studio Code (HTTPS)
- IntelliJ IDEA (SSH)
- IntelliJ IDEA (HTTPS)

En mi caso es : `http://3e44dd25ad6c/root/my-web.git`

Clonamos nuestro repo localmente y hacemos un push a Gitlab

Usamos el siguiente comando para clonar:

git clone http://172.17.0.2/root/my-web.git (revisar que modificamos el clone y le colocamos la ip de nuestro servidor de Gitlab)

Colocamos nuestras credenciales

```
matl@matl-virtual-machine: ~/proyecto
root@29e3c0f41102: /
matl@matl-virtual-machine:~/proyecto$ git clone http://172.17.0.2/root/my-web.git
Cloning into 'my-web'...
Username for 'http://172.17.0.2': root
Password for 'http://root@172.17.0.2':
warning: You appear to have cloned an empty repository.
matl@matl-virtual-machine:~/proyecto$
```

En mi caso voy a subir a dicho directorio de donde clonamos nuestro repositorio un template HTML5

```
matl@matl-virtual-machine: ~/proyecto/my-web$ cd ..
matl@matl-virtual-machine:~/proyecto$ ls
ice-cream  my-web
matl@matl-virtual-machine:~/proyecto$ cp -r ./ice-cream/* my-web/
```

Se ve de la siguiente manera:

```
matl@matl-virtual-machine: ~/proyecto/my-web$ ls -l
total 232
-rw-rw-r-- 1 mati mati 15732 mar 14 14:59 about.html
-rw-rw-r-- 1 mati mati 10764 mar 14 14:59 contact.html
drwxrwxr-x 2 mati mati 4096 mar 14 14:59 css
-rw-rw-r-- 1 mati mati 12391 mar 14 14:59 gallery.html
-rw-rw-r-- 1 mati mati 88419 mar 14 14:59 ice-cream-shop-website-template.jpg
drwxrwxr-x 2 mati mati 4096 mar 14 14:59 img
-rw-rw-r-- 1 mati mati 34819 mar 14 14:59 index.html
drwxrwxr-x 2 mati mati 4096 mar 14 14:59 js
drwxrwxr-x 8 mati mati 4096 mar 14 14:59 lib
-rw-rw-r-- 1 mati mati 1456 mar 14 14:59 LICENSE.txt
drwxrwxr-x 2 mati mati 4096 mar 14 14:59 mail
-rw-rw-r-- 1 mati mati 15867 mar 14 14:59 product.html
-rw-rw-r-- 1 mati mati 551 mar 14 14:59 READ-ME.txt
drwxrwxr-x 3 mati mati 4096 mar 14 14:59 scss
-rw-rw-r-- 1 mati mati 14805 mar 14 14:59 service.html
matl@matl-virtual-machine:~/proyecto/my-web$
```

Ahora si realizo un git status me figuran todos los archivos que he copiado dentro de la carpeta.

```
mati@mati-virtual-machin
root@29e3c0f41102: /
On branch main
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    LICENSE.txt
    README.txt
    about.html
    contact.html
    css/
    gallery.html
    ice-cream-shop-website-template.jpg
    img/
    index.html
    js/
    lib/
    mail/
    product.html
    scss/
    service.html

nothing added to commit but untracked files present (use "git add" to track)
mati@mati-virtual-machine:~/proyecto/my-web$
```

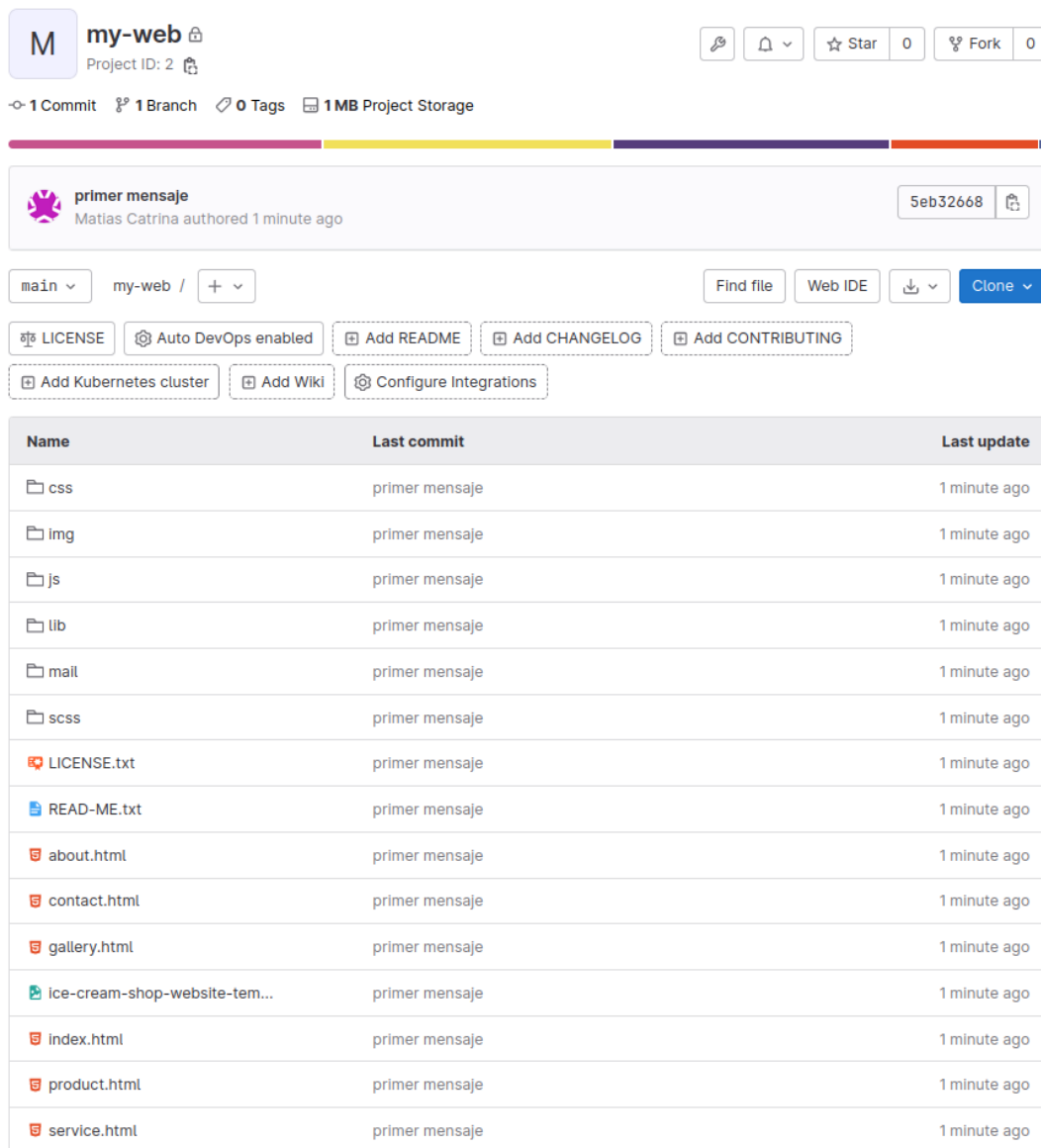
Procedemos a subirlos a nuestro gitlab con los siguientes comandos:

- `git add .`
- `git commit -m "Mensaje commit"`
- `git push`

Nuestro push quedaria algo asi:

```
mati@
root@29e3c0f41102: /
mati@mati-virtual-machine:~/proyecto/my-web$ git push
Username for 'http://172.17.0.2': root
Password for 'http://root@172.17.0.2':
Enumerating objects: 183, done.
Counting objects: 100% (183/183), done.
Delta compression using up to 4 threads
Compressing objects: 100% (178/178), done.
Writing objects: 100% (183/183), 1011.83 KiB | 22.48 MiB/s, done.
Total 183 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
To http://172.17.0.2/root/my-web.git
 * [new branch]      main -> main
mati@mati-virtual-machine:~/proyecto/my-web$
```

Si corroboramos en nuestro GitLab debería figurar lo siguiente:



The screenshot shows a GitLab repository page for a project named 'my-web'. At the top, there's a header with the project name, a lock icon, and statistics: 1 Commit, 1 Branch, 0 Tags, and 1 MB Project Storage. Below this is a commit message 'primer mensaje' by 'Matias Catrina' from 1 minute ago. The file browser shows a directory structure with folders like 'css', 'img', 'js', 'lib', 'mail', 'scss' and files like 'LICENSE.txt', 'READ-ME.txt', 'about.html', 'contact.html', 'gallery.html', 'ice-cream-shop-website-tem...', 'index.html', 'product.html', and 'service.html'. All files show they were last updated '1 minute ago'.

4 - Levantamos nuestro segundo contenedor que en este caso va a ser un Apache de la siguiente manera:

`docker run -dit ubuntu:latest`

```
mat@mat-virtual-machine: ~$ docker run -dit ubuntu:latest
27347b32f1374e3fd0ceafa17662a38f30b4ffe1201d226d6efc038dee624e05
mat@mat-virtual-machine: ~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS                               NAMES
27347b32f137   ubuntu:latest  "/bin/bash"             4 seconds ago Up 3 seconds       22/tcp, 80/tcp, 443/tcp            eloquent_williamson
8e44dd25ad6c   gitlab/gitlab-ce  "/assets/wrapper"       21 minutes ago Up 21 minutes (healthy)                pensive_ishizaka
mat@mat-virtual-machine: ~$
```

5 - Creamos una nueva red en Docker con el comando

`docker network create <nombre de nuestra red>`

```
mat@mat-virtual-machine:~$ docker network create red1
b9fc308f1a7ba071a95dc9c6153f2bd5a4b500504f243dbb73ed8114d88ec56a
mat@mat-virtual-machine:~$ docker network ls
NETWORK ID        NAME        DRIVER        SCOPE
1943ae68128b      bridge      bridge        local
ffe443daabcd      host        host          local
d3e4ed64eb73      none        null          local
b9fc308f1a7b      red1        bridge        local
mat@mat-virtual-machine:~$
```

Repasemos un poco nuestra infraestructura actual

- 1 Contenedor ejecutando GitLab.
- 1 Contenedor ejecutando Apache.
- 1 Red en Docker.

Al tener 2 contenedores y entre ellos se van a comunicar, vamos a realizar la conexión a una misma red a través del siguiente comando:

docker network connect <nombre de red que creamos> <container id>

```
mat@mat-virtual-machine:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
29e3c0f41102  ubuntu:latest  "/bin/bash"             6 minutes ago Up 6 minutes                                     sleepy_galileo
3e44dd25ad6c   gitlab/gitlab-ce  "/assets/wrapper"       31 minutes ago Up 31 minutes (healthy)  22/tcp, 80/tcp, 443/tcp  pensive_ishizaka
mat@mat-virtual-machine:~$ docker network ls
NETWORK ID        NAME        DRIVER        SCOPE
1943ae68128b      bridge      bridge        local
ffe443daabcd      host        host          local
d3e4ed64eb73      none        null          local
b9fc308f1a7b      red1        bridge        local
mat@mat-virtual-machine:~$ docker network connect red1 29e3c0f41102
mat@mat-virtual-machine:~$ docker network connect red1 3e44dd25ad6c
```

Podemos verificar que nuestro contenedor ya tiene seteado la nueva red usando:

docker inspect <container id>

6 - Vamos a ingresar a nuestro contenedor de Apache para poder realizar algunas modificaciones de la siguiente manera:

```
docker exec -it 29e3c0f41102 bash
```

Luego instalamos y actualizamos utilizando los siguientes comandos:

- ☐ **Actualizamos librerías:** *apt update*
- ☐ **Instalamos wget para después instalar nuestro runner:** *apt install wget*
- ☐ **Instalamos nuestro servidor web Apache:** *apt install apache2*
- ☐ **Instalamos git para utilizarlo más adelante:** *apt install git*
- ☐ **Inicializamos nuestro servidor:** *service apache2 start*

Finalmente deberíamos llegar a la siguiente imagen, en donde nos va a devolver la ip de nuestro servidor Apache.

```
docker exec -it 29e3c0f41102 bash
```

- ☐ Actualizamos librerías: *apt update*
- ☐ Instalamos wget para después instalar nuestro runner: *apt install wget*
- ☐ Instalamos nuestro servidor web Apache: *apt install apache2*
- ☐ Instalamos git para utilizarlo más adelante: *apt install git*
- ☐ Inicializamos nuestro servidor: *service apache2 start*

```

root@29e3c0f41102:/#
Setting up libxdmcp6:amd64 (1:1.1.3-0ubuntu5) ...
Setting up libxcb1:amd64 (1.14-3ubuntu3) ...
Setting up libedit2:amd64 (3.1-20210910-1build1) ...
Setting up git (1:2.34.1-1ubuntu1.8) ...
Setting up libx11-6:amd64 (2:1.7.5-1) ...
Setting up libxmu6:amd64 (2:1.1.3-3) ...
Setting up openssl-client (1:8.9p1-3ubuntu0.1) ...
update-alternatives: using /usr/bin/ssh to provide /usr/bin/rsh (rsh) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rsh.1.gz because associated file /usr/share/man/man1/ssh.1.gz (of link group rsh) doesn't exist
update-alternatives: using /usr/bin/slogin to provide /usr/bin/rlogin (rlogin) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rlogin.1.gz because associated file /usr/share/man/man1/slogin.1.gz (of link group rlogin) doesn't exist
update-alternatives: using /usr/bin/scp to provide /usr/bin/rcp (rcp) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/rcp.1.gz because associated file /usr/share/man/man1/scp.1.gz (of link group rcp) doesn't exist
Setting up libxext6:amd64 (2:1.3.4-1build1) ...
Setting up xauth (1:1.1-1build2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
root@29e3c0f41102:/# service apache2 start
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
*
root@29e3c0f41102:/#

```

Tutorial realizado por Matías Catrina



7 - Nos tocaría ahora reemplazar dicha web, por nuestro proyecto que tenemos en GitLab. Ejecutamos lo siguiente:

- `cd /var/www/html` (nos posicionamos dentro del directorio del servidor)
- `ls` (verificamos nuestro que está nuestro archivo index.html)
- `rm *` : (borramos todo lo que exista dentro del directorio actual)
- `git clone <el que utilizamos en el paso 3> /var/www/html/` (clonamos nuestro repositorio)

Nos debería quedar algo similar a esto:

```

root@29e3c0f41102: /var/www/html
root@29e3c0f41102: /var/www/html# cd /var/www/html/
root@29e3c0f41102: /var/www/html# ls
index.html
root@29e3c0f41102: /var/www/html# rm *
root@29e3c0f41102: /var/www/html# ls
root@29e3c0f41102: /var/www/html# git clone http://172.17.0.2/root/my-web.git /var/www/html/
Cloning into '/var/www/html'...
Username for 'http://172.17.0.2': root
Password for 'http://root@172.17.0.2':
remote: Enumerating objects: 183, done.
remote: Counting objects: 100% (183/183), done.
remote: Compressing objects: 100% (171/171), done.
remote: Total 183 (delta 7), reused 183 (delta 7), pack-reused 0
Receiving objects: 100% (183/183), 1011.83 KiB | 56.21 MiB/s, done.
Resolving deltas: 100% (7/7), done.
root@29e3c0f41102: /var/www/html# ls
LICENSE.txt  about.html  css          ice-cream-shop-website-template.jpg  index.html  lib  product.html  service.html
README.txt  contact.html  gallery.html  img                                   js          mail  scss
root@29e3c0f41102: /var/www/html#

```

8 - Instalamos nuestro runner en el servidor para que se ejecuten las tareas de nuestro pipeline automáticamente. Ejecutamos los siguientes comandos:

- `wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64`
- `chmod 777 /usr/local/bin/gitlab-runner`
- `gitlab-runner register`

Nos quedaría de la siguiente manera:

```
root@29e3c0f41102: /var/www/html# wget -O /usr/local/bin/gitlab-runner https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
--2023-03-14 18:18:04-- https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64
Resolving gitlab-runner-downloads.s3.amazonaws.com (gitlab-runner-downloads.s3.amazonaws.com)... 52.216.49.145, 52.217.33.28, 52.217.114.121, ...
Connecting to gitlab-runner-downloads.s3.amazonaws.com (gitlab-runner-downloads.s3.amazonaws.com)[52.216.49.145]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 57411232 (55M) [application/octet-stream]
Saving to: '/usr/local/bin/gitlab-runner'

/usr/local/bin/gitlab-runner      100%[=====] 54.75M  8.70MB/s   in 14s

2023-03-14 18:18:19 (3.92 MB/s) - '/usr/local/bin/gitlab-runner' saved [57411232/57411232]

root@29e3c0f41102: /var/www/html# chmod 777 /usr/local/bin/gitlab-runner
root@29e3c0f41102: /var/www/html# gitlab-runner register
Runtime platform                                arch=amd64 os=linux pid=3468 revision=d540b510 version=15.9.1
Running in system-mode.

Created missing unique system ID                  system_id=r_aqLF9MuoDOAZ
Enter the GitLab instance URL (for example, https://gitlab.com/):
http://172.17.0.2/
Enter the registration token:
GR1348941Nzcdjco2pdGyZXmYc3s5
Enter a description for the runner:
[29e3c0f41102]:
Enter tags for the runner (comma-separated):

Enter optional maintenance note for the runner:

WARNING: Support for registration tokens and runner parameters in the 'register' command has been deprecated in GitLab Runner 15.6 and will be replaced
with support for authentication tokens. For more information, see https://gitlab.com/gitlab-org/gitlab/-/issues/380872
Registering runner... succeeded                    runner=GR1348941Nzcdjco2
Enter an executor: virtualbox, docker+machine, docker-ssh+machine, instance, kubernetes, docker, shell, parallels, ssh, custom, docker-ssh:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
root@29e3c0f41102: /var/www/html#
```

Cuando registramos el runner lo configuramos de la siguiente manera:

```
Enter the GitLab instance URL (for example, https://gitlab.com/):
http://172.17.0.2/

Enter the registration token:
GR1348941Nzcdjco2pdGyZXmYc3s5
```

Ambos datos los obtenemos desde:

proyecto > settings > CI/CD > Runners

Project runners

These runners are assigned to this project.

Set up a project runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:
`http://172.17.0.2/`

And this registration token:
`GR1348941Nzcdjco2pdGyZXmYc3s5`

Reset registration token

Show runner installation instructions

En las opciones de descripción, tags omitimos en caso de que quieran configurar esas opciones lo pueden hacer. Y en la ultima configuración escribimos *shell*. Porque en este caso vamos a querer ejecutar comandos shell en este runner.

```
Registering runner... succeeded runner=GR1348941Nzcdjco2
Enter an executor: virtualbox, docker+machine, docker-ssh+machine, instance, kubernetes, docker, shell, parallels, ssh, custom, docker-ssh:
shell
```

En GitLab nos encontramos con lo siguiente:

Project runners

These runners are assigned to this project.

Set up a project runner for a project

1. Install GitLab Runner and ensure it's running.
2. Register the runner with this URL:
`http://172.17.0.2/`

And this registration token:
`GR1348941Nzcdjco2pd6yZXmYc3s5`

[Reset registration token](#)

[Show runner installation instructions](#)

Assigned project runners

 #1 (skJGzz5q6)    [Remove runner](#)

29e3c0f41102

Para ejecutar activar nuestro runner, ejecutamos el comando:

```
gitlab-runner run
```

Nos quedaría de esta manera:

Assigned project runners

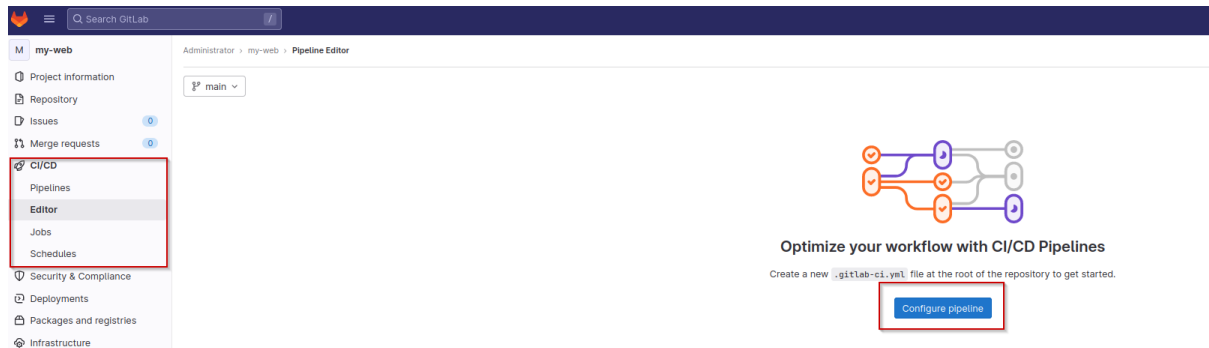
 #1 (skJGzz5q6)    [Remove runner](#)

29e3c0f41102

9 - Creamos nuestro primer pipeline:

proyecto > CI/CD > Editor y le damos click al botón

Configure pipeline



Borramos lo que esté dentro del archivo y pegamos lo siguiente (previamente modificando los parámetros username,password,url del repositorio):

stages:

- init
- despliegue

hello:

stage: init

script:

- echo "Nuestro primer pipeline"

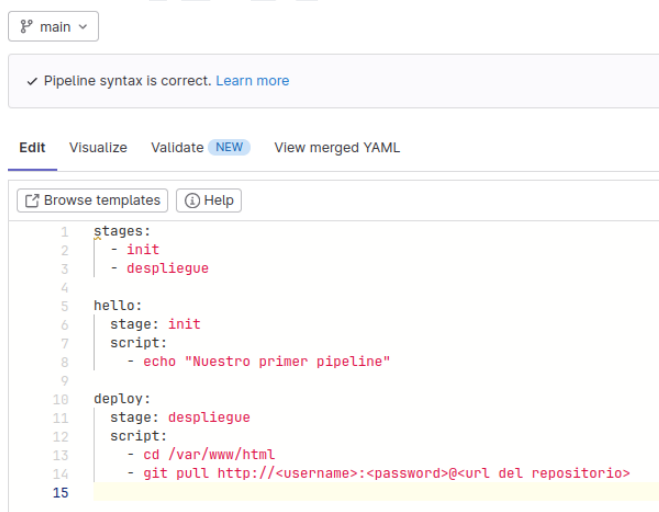
deploy:

stage: despliegue


script:

- cd /var/www/html
- git pull http://<username>:<password>@<url del repositorio>

Nos quedaria algo asi:



Verificamos que nos dé el check de que la sintaxis de nuestro yaml está bien escrita

 main ▾

✓ Pipeline syntax is correct. [Learn more](#)

Procedemos a commitear los cambios

Commit message

Update .gitlab-ci.yml file

Branch

main

Commit changes

Reset

Y podemos verificar que en pipelines ya figura en correcto funcionamiento nuestro commit:

M my-web

Project information

Repository

Issues 0

Merge requests 0

CI/CD

Pipelines

Editor

Jobs

Schedules




Security & Compliance

Deployments

Administrator > my-web > Pipelines

All 1 Finished Branches Tags

Filter pipelines

Status	Pipeline	Triggerer	Stages
<div>passed</div> <div>00:00:01</div> <div>1 minute ago</div>	<div>Update .gitlab-ci.yml file</div> <div>#1  main → e11f1847 </div> <div>latest</div>		<div>✓</div> <div>✓</div>

Finalmente ya podemos realizar las modificaciones, que cuando hagamos el push se realizará de manera automática.