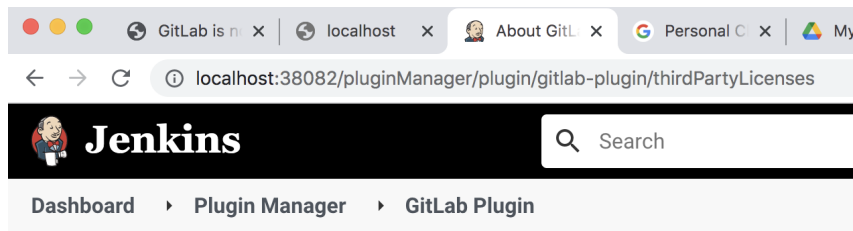


01 - Configuración de los Plugins

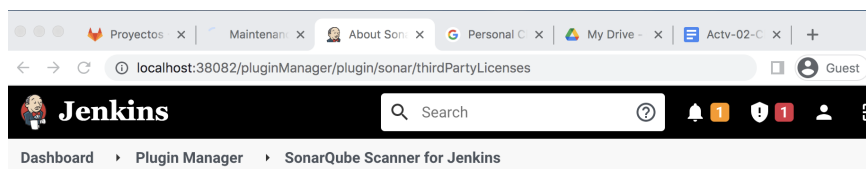
a. Plugins Jenkins y GitLab instalados:



About GitLab Plugin 1.5.29

3rd Party Dependencies

Name	Maven ID
GitLab Plugin	org.jenkins-ci.plugins:gitlab-plugin:1.5.29
Instance Identity	org.jenkins-ci.modules:instance-identity:2.2

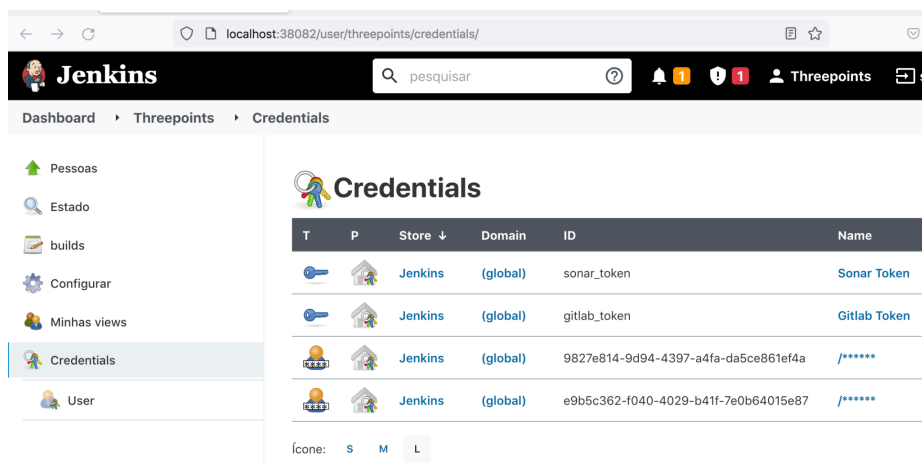


About SonarQube Scanner for Jenkins 2.14

3rd Party Dependencies

Name	Maven ID	License
SonarQube Scanner for Jenkins	org.jenkins-ci.plugins:sonar:2.14	GNU Lesser General Public License (LGPL), v.3
Project Scopes	org.jenkins-ci.plugins:project-scopes:2.14.1	GNU Lesser General Public License (LGPL), v.3

b. Para los plugins las credenciales son guardadas:



c. GitLab y SonarQube configuración:

SonarQube servers

☐ **Environment variables** Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Gitlab

☒ **Enable authentication for '/project' end-point**

GitLab connections

Connection name

A name for the connection

Gitlab host URL

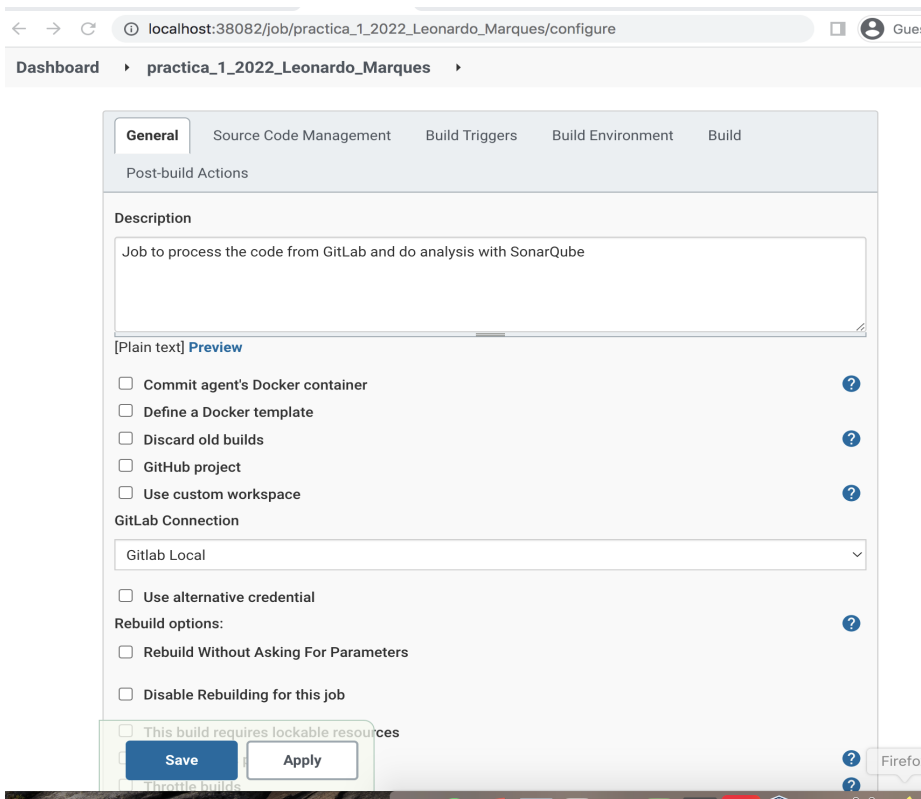
The complete URL to the Gitlab server (e.g. http://gitlab.mydomain.com)

Credentials

API Token for accessing Gitlab

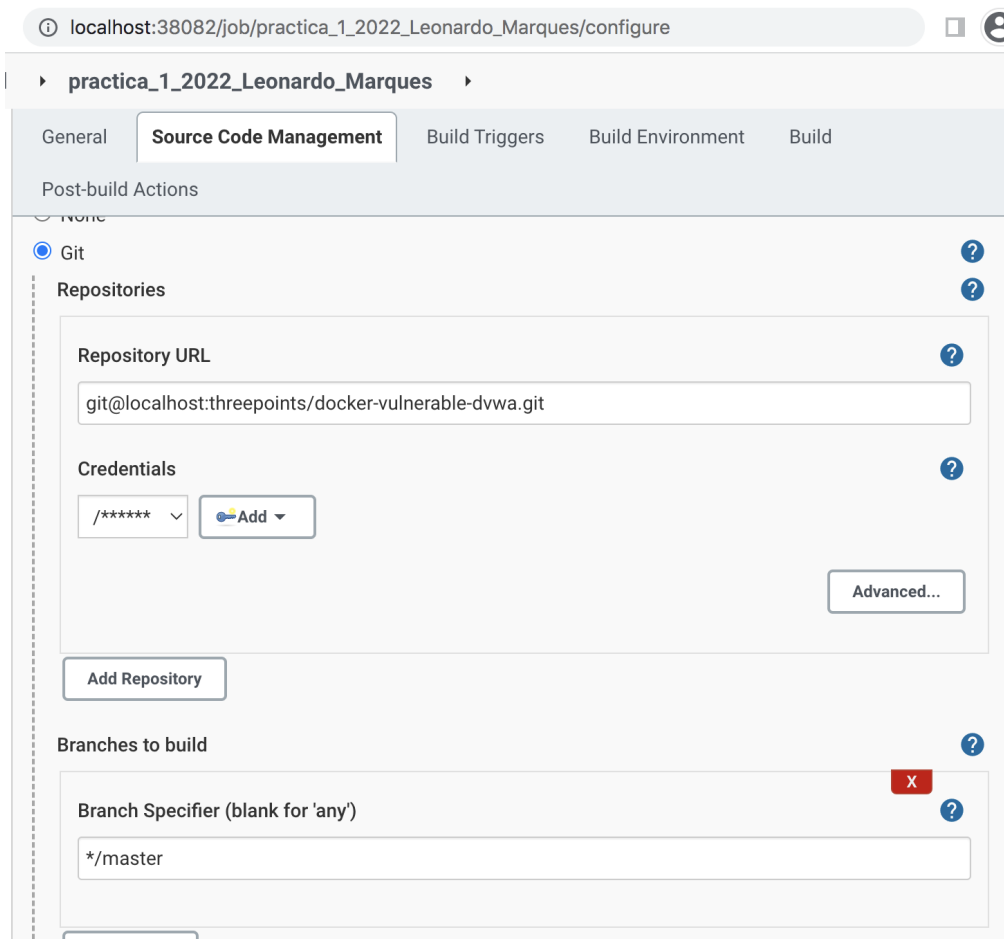
2- Creación de un Pipeline básico de CI:

a. Configuración pestaña general:



The screenshot shows the Jenkins configuration page for a job named 'practica_1_2022_Leonardo_Marques'. The 'General' tab is selected. The 'Description' field contains the text: 'Job to process the code from GitLab and do analysis with SonarQube'. Below the description, there are several checkboxes for configuration options: 'Commit agent's Docker container', 'Define a Docker template', 'Discard old builds', 'GitHub project', 'Use custom workspace', 'Use alternative credential', 'Rebuild Without Asking For Parameters', and 'Disable Rebuilding for this job'. The 'GitLab Connection' dropdown is set to 'Gitlab Local'. At the bottom, there are 'Save' and 'Apply' buttons.


b. Configuración pestaña general:





The screenshot shows the Jenkins configuration page for the same job, but with the 'Source Code Management' tab selected. The 'Git' radio button is selected under 'Repositories'. The 'Repository URL' field contains the text: 'git@localhost:threepoints/docker-vulnerable-dvwa.git'. The 'Credentials' dropdown is set to '/*****'. There is an 'Add' button next to the credentials dropdown. Below the repository section, there is an 'Add Repository' button. In the 'Branches to build' section, the 'Branch Specifier (blank for 'any')' field contains the text: '*/master'. There is a red 'X' icon next to the branch specifier field.

c. Configuración del Sonar:


Build


 **Execute SonarQube Scanner**

Task to run 

JDK 

JDK to be used for this SonarQube analysis


Path to project properties 

Analysis properties 

```
# unique project identifier (required)
sonar.projectKey=test_project

# project metadata (used to be required, optional since SonarQube 6.1)
sonar.projectName=My project
sonar.projectVersion=1.0

# path to source directories (required)
sonar.sources=dvwa
```

Additional arguments 

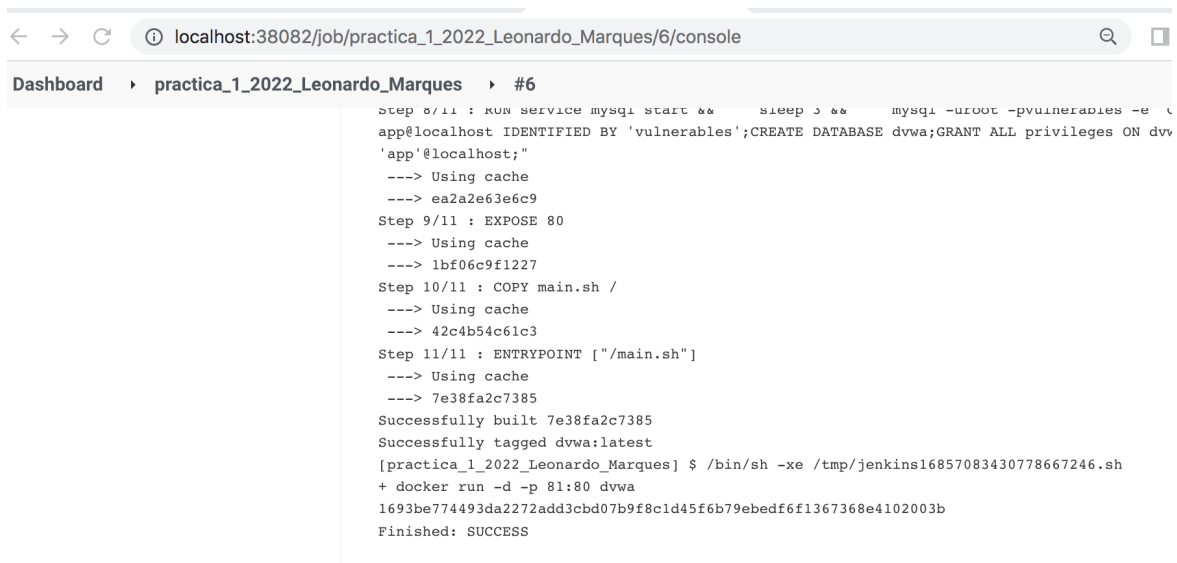
d. Construcción y despliegue de la aplicación después del quality check:

 **Execute shell**

Command

```
docker build -t dvwa .
```

See [the list of available environment variables](#)

e. Job ejecutado con success:


```

Step 8/11 : RUN service mysql start && sleep 3 && mysql -uroot -pvulnerables -e '
app@localhost IDENTIFIED BY 'vulnerables';CREATE DATABASE dvwa;GRANT ALL privileges ON dvwa
' app'@localhost;"
----> Using cache
----> ea2a2e63e6c9
Step 9/11 : EXPOSE 80
----> Using cache
----> 1bf06c9f1227
Step 10/11 : COPY main.sh /
----> Using cache
----> 42c4b54c61c3
Step 11/11 : ENTRYPOINT ["/main.sh"]
----> Using cache
----> 7e38fa2c7385
Successfully built 7e38fa2c7385
Successfully tagged dvwa:latest
[practica_1_2022_Leonardo_Marques] $ /bin/sh -xe /tmp/jenkins16857083430778667246.sh
+ docker run -d -p 81:80 dvwa
1693be774493da2272add3cbd07b9f8c1d45f6b79ebdf6f1367368e4102003b
Finished: SUCCESS

```

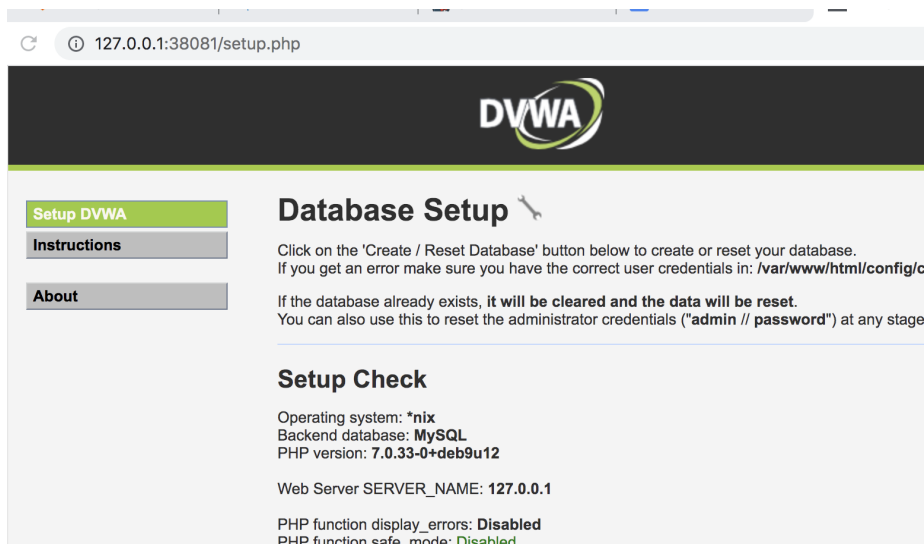
4. Opcional: Circuito con CD**a. Configuración del docker run después del build.**


Execute shell

Command

```
docker run -d -p 81:80 dvwa
```

See [the list of available environment variables](#)

f. Aplicación en ejecución local:


127.0.0.1:38081/setup.php

DVWA

Setup DVWA

Instructions

About

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/config/c`

If the database already exists, **it will be cleared and the data will be reset.**
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **7.0.33-0+deb9u12**

Web Server SERVER_NAME: **127.0.0.1**

PHP function display_errors: **Disabled**
PHP function safe_mode: **Disabled**