

PRUEBA TÉCNICA PARA MERCADONA



Miguel Carrillo Álvarez

En esta prueba técnica se ha pedido crear una pipeline completa en Jenkins que debe contener una serie de steps mínimos. Con la implementación de una aplicación que se debe crear dentro de un Docker. Se ha escogido para la pipeline el modelo declarativo en lenguaje Groovy y en la aplicación se ha elegido, para aligerar el proceso, una aplicación en nodejs que muestra “hola mundo” en un servidor HTTP. Todo el código está disponible en el repositorio de GitHub:

<https://github.com/m1c4lv/mercadona/tree/main>

Se presentan una serie de comentarios a tener en cuenta sobre la prueba:



- Es conveniente enlazar el Git con Jenkins mediante Webhook para que cada vez que se realicen cambios en el repositorio se lance la pipeline automáticamente. Como es un repositorio público no se necesitan credenciales de acceso.
- Se parametriza la pipeline para que sea reutilizable con otras imágenes de Docker.
- El paso 5 “build” no aplica ya que no se necesita compilar la aplicación.
- Para el paso 7 se utiliza *secret* de Jenkins para guardar la contraseña del repositorio.
- El paso 8 “Despliegue en kubernetes” se comentan las línea de código al no tener un cluster de kubernetes donde desplegar la imagen de Docker.

Prueba de uso, primero guardo las credenciales del repositorio:

Global credentials (unrestricted)

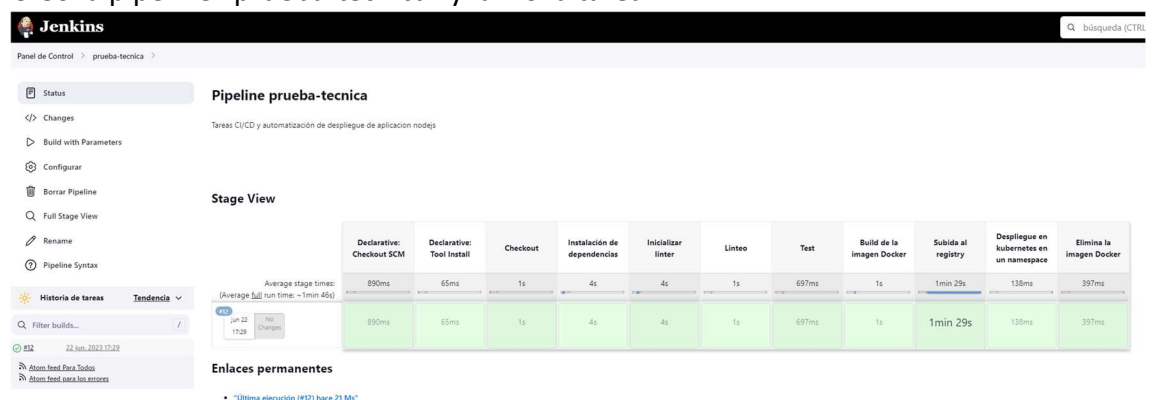
+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 dockerhubcredentials	Dockerhub Credentials	Username with password	Dockerhub Credentials 

Icono: S M L

Creo la pipeline “prueba-tecnica” y lanzo la tarea:



The screenshot shows the Jenkins interface for a pipeline named "prueba-tecnica". The left sidebar contains navigation links: Status, Changes, Build with Parameters, Configurar, Borrar Pipeline, Full Stage View, Rename, and Pipeline Syntax. The main area displays the "Stage View" of the pipeline, which consists of 11 stages: Declarative: Checkout SCM, Declarative: Tool Install, Checkout, Instalación de dependencias, Inicializar lint, Linteo, Test, Build de la imagen Docker, Subida al registry, Despliegue en kubernetes en un namespace, and Elimina la imagen Docker. Each stage has a duration bar and a tooltip showing the average stage time. Below the stage view, there is a "Historia de tareas" (Task History) section showing a list of builds with filters and a "Tendencia" (Trend) graph. The "Enlaces permanentes" (Permanent Links) section at the bottom provides a link to the latest execution.

Stage	Duration
Declarative: Checkout SCM	890ms
Declarative: Tool Install	65ms
Checkout	1s
Instalación de dependencias	4s
Inicializar lint	4s
Linteo	1s
Test	697ms
Build de la imagen Docker	1s
Subida al registry	1min 29s
Despliegue en kubernetes en un namespace	138ms
Elimina la imagen Docker	397ms

Los logs que obtiene son los siguientes:

Started by user m1c4lv

Obtained Jenkinsfile from git <https://github.com/m1c4lv/mercadona>

```

[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/prueba-tecnica
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
  > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/prueba-tecnica/.git # timeout=10
Fetching changes from the remote Git repository
  > git config remote.origin.url https://github.com/mlc4lv/mercadona # timeout=10
Fetching upstream changes from https://github.com/mlc4lv/mercadona
  > git --version # timeout=10
  > git --version # 'git version 2.34.1'
  > git fetch --tags --force --progress -- https://github.com/mlc4lv/mercadona
+refs/heads/*:refs/remotes/origin/* # timeout=10
  > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision e67cc6a00a4b5d05f62c3922da3e609e19ddd084 (refs/remotes/origin/main)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f e67cc6a00a4b5d05f62c3922da3e609e19ddd084 # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ rm -rf mercadona
[Pipeline] sh
+ git clone https://github.com/mlc4lv/mercadona.git
Cloning into 'mercadona'...
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Instalación de dependencias)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv

```

```

[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] sh
+ npm install

added 180 packages, and audited 181 packages in 2s

79 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[Pipeline] sh
+ yes
+ npm install eslint

up to date, audited 181 packages in 1s

79 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Inicializar linter)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] sh
+ ./node_modules/.bin/eslint --init
You can also run this command directly using 'npm init
@eslint/config'.
[?251[36m? [39m [1mHow would you like to use ESLint?[22m
[2m...[22m
  To check syntax only
[36m[39m [36m[4mTo check syntax and find problems[24m[39m
  To check syntax, find problems, and enforce code
style[3A[39G[?25h
[Pipeline] sh
+ rm -rf .eslintrc.js
[Pipeline] sh
+ cp my-eslintrc.js .eslintrc.js
[Pipeline] sh
+ yes
+ npm install eslint-plugin-jest

added 29 packages, and audited 210 packages in 2s

86 packages are looking for funding
  run `npm fund` for details

```

```
found 0 vulnerabilities
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Linteo)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] sh
+ npm run lint
```

```
> hola-mundo@1.0.0 lint
> eslint . --ext .js
```

```
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] sh
+ npm test
```

```
> hola-mundo@1.0.0 test
> echo "No hay pruebas definidas"
```

```
No hay pruebas definidas
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build de la imagen Docker)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] sh
+ docker version
```

Client: Docker Engine - Community
Version: 23.0.1
API version: 1.42
Go version: go1.19.5
Git commit: a5ee5b1
Built: Thu Feb 9 19:47:01 2023
OS/Arch: linux/amd64
Context: default

Server: Docker Engine - Community

Engine:
Version: 23.0.1
API version: 1.42 (minimum version 1.12)
Go version: go1.19.5
Git commit: bc3805a
Built: Thu Feb 9 19:47:01 2023
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.6.16
GitCommit: 31aa4358a36870b21a992d3ad2bef29e1d693bec
runc:
Version: 1.1.4
GitCommit: v1.1.4-0-g5fd4c4d
docker-init:
Version: 0.19.0
GitCommit: de40ad0

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] withEnv

[Pipeline] {

[Pipeline] sh

+ docker build -t m1c4lv/nodejs:v1 .

#1 [internal] load build definition from Dockerfile

#1 transferring dockerfile: 163B done

#1 DONE 0.0s

#2 [internal] load .dockerignore

#2 transferring context: 2B done

#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/node:latest

#3 DONE 0.5s

#4 [1/5] FROM

docker.io/library/node@sha256:2f0b0c15f97441defa812268ee943bbfaaf666ea6cf7cac62ee3f127906b35c6

#4 DONE 0.0s

#5 [internal] load build context

#5 transferring context: 902B done

#5 DONE 0.0s

#6 [4/5] COPY app.js .

#6 CACHED

#7 [2/5] WORKDIR /usr/src/app

#7 CACHED

#8 [3/5] COPY package.json ./

```

#8 CACHED

#9 [5/5] RUN npm install
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image
sha256:e35d1f455b854ada8ca92c6959600dd0b17d051ca132e2f98dc0347eebfcc97
b done
#10 naming to docker.io/mlc4lv/nodejs:v1 done
#10 DONE 0.0s
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Subida al registry)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/prueba-tecnica/mercadona
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u mlc4lv -p ***** https://index.docker.io/v1/
WARNING! Using --password via the CLI is insecure. Use --password-
stdin.
WARNING! Your password will be stored unencrypted in
/var/lib/jenkins/workspace/prueba-tecnica/mercadona@tmp/b1debc05-5714-
4f8c-97b5-0737c64bc1ea/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential
s-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag mlc4lv/nodejs:v1 mlc4lv/nodejs:v1
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push mlc4lv/nodejs:v1
The push refers to repository [docker.io/mlc4lv/nodejs]

```

```
13d33c9cc9ad: Preparing
9675538cb7f8: Preparing
4e69a43ada43: Preparing
4298c1c98cec: Preparing
7c206aeelc45: Preparing
3f24647bd2e9: Preparing
52c74b9dd81c: Preparing
1fdf7be6e30b: Preparing
e67fb4bad8f4: Preparing
964529c819bb: Preparing
2f98f42985b1: Preparing
332b199f36eb: Preparing
1fdf7be6e30b: Waiting
964529c819bb: Waiting
3f24647bd2e9: Waiting
e67fb4bad8f4: Waiting
332b199f36eb: Waiting
2f98f42985b1: Waiting
52c74b9dd81c: Waiting
7c206aeelc45: Mounted from library/node
9675538cb7f8: Pushed
4e69a43ada43: Pushed
4298c1c98cec: Pushed
3f24647bd2e9: Mounted from library/node
e67fb4bad8f4: Mounted from library/node
1fdf7be6e30b: Mounted from library/node
52c74b9dd81c: Mounted from library/node
2f98f42985b1: Mounted from library/node
964529c819bb: Mounted from library/node
332b199f36eb: Mounted from library/node
13d33c9cc9ad: Pushed
v1: digest:
sha256:401241970d025cbc5cb9cd5ef38797b6e14f3eab1279c18292e83c6de3216ef
2 size: 2837
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Despliegue en kubernetes en un namespace)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
Simulando despliegue en kubernetes
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
```



```

[Pipeline] { (Elimina la imagen Docker)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker rmi --force m1c4lv/nodejs:v1
Untagged: m1c4lv/nodejs:v1
Untagged:
m1c4lv/nodejs@sha256:401241970d025cbc5cb9cd5ef38797b6e14f3eab1279c1829
2e83c6de3216ef2
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

En el repositorio de DockerHub se ha guardado la imagen:

