

Notas y comandos usados en el proyecto

Docker comandos

// Descargar docker image

```
docker pull jenkins/jenkins:its
```

// Listar docker images

```
docker images -a
```

// Crea y corre la docker image

```
docker run -d -v $(pwd)/jenkins_github:/var/jenkins_github -p 8080:8080 -p 50000:50000 --name jenkinsGithub jenkins/jenkins:its
```

// Mostrar un listado de los contenedores en ejecución,

```
docker ps [OPTIONS]
```

//Mostrar todos los contenedores presentes en Docker

```
-a
```

//Mostrar todos los contenedores filtrados por status

```
-f status=running
```

// Monitorizar con docker logs

```
docker logs -f jenkinsGithub
```

// Get Jenkins password

```
docker exec jenkinsGithub cat /var/jenkins_home/secrets/initialAdminPassword
```

// Docker Start Container

```
docker start (CONTAINER_NAME)
```

// Docker Stop Container

```
docker stop [OPTIONS] [CONTAINER_NAME]
```

// Docker Stop in a brutal way

```
docker kill [CONTAINER_NAME]
```

// Docker Remove Container

```
docker rm [OPTIONS] [CONTAINER_NAME]
```

// Correr /bin/bash como root user

```
docker exec -it -u root [DOCKER-CONTAINER-NAME] bash
```

// usuario de jenkins

```
su -s /bin/bash [JENKINS-USER]
```

// Generar llave publica y privada

ssh-keygen

NGROK comandos

// Install ngrok

// Windows

choco install ngrok

// Linux

```
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee
/etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-
agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list &&
sudo apt update && sudo apt install ngrok
```

// Conectar cuenta

```
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee
/etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-
agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list &&
sudo apt update && sudo apt install
```

// Fire it up

ngrok http [PORT]

The screenshot shows the Docker Desktop interface. The top section is titled 'Containers' and includes a toggle for 'Only show running containers' and a search bar. Below this is a table of containers. The bottom section is titled 'Webhooks' and includes an 'Add webhook' button and a list of webhooks.

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	jenkinsGithub e9163ef152cd	jenkins/jenkins:latest	Running	50000-50000 8080-8080	32 minutes ago	

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://8ada-200-87-93-94.sa.ngrok... (push)	Edit Delete
---	--	---------------------------------------

```
ngrok
Add Single Sign-On to your ngrok dashboard via your Identity Provider: https://ngrok.com/dash530

Session Status      online
Account             Angel Leonardo Mendieta Castillo (Plan: Free)
Version             3.1.0
Region              South America (sa)
Latency             107ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://8ada-200-87-93-94.sa.ngrok.io -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
   95     0    0.00   0.00   5.00   5.05

HTTP Requests
-----
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
GET /job/JenkinsGithub/wfapi/runs 200 OK
```

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Quiet period ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/LioX-Code/JalaUniversity_DevOps.git

Credentials ?

- none -

+ Add

Advanced...

Add Repository

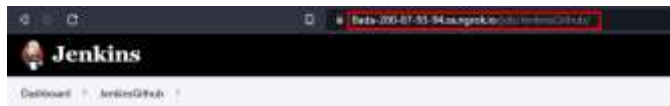
Branches to build ?

Script Path ?

Branch Specifier (blank for 'any') ?

*/Jenkins_Github

Jenkinsfile



- States
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Resume
- Pipeline Syntax
- GitHub Hook Log

Pipeline JenkinsGithub

Stage View

		Declarative: Checkout SCM	Hello
Average stage times (Average last run time: ~5s)		1s	111ms
Nov 22 12:14	1s	1s	111ms
Nov 21 14:11	1s	1s	111ms
Nov 21 14:20	1s	1s	111ms
Nov 21 14:17	1s	1s	111ms
Nov 21 14:11	1s	1s	111ms

Console Output

```
started by github:push by lian-code
obtained Jenkinsfile from git https://github.com/Lian-Coder/JavaUniversity_Develop.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/JenkinsGithub
[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/jenkins_home/workspace/JenkinsGithub/.git # timeout=10
fetching changes from the remote git repository
> git config remote.origin.url https://github.com/Lian-Coder/JavaUniversity_Develop.git # timeout=10
fetching upstream changes from https://github.com/Lian-Coder/JavaUniversity_Develop.git
> git --version # timeout=10
> git --version # 'git version 2.30.2'
> git fetch --tags --force --progress -- https://github.com/Lian-Coder/JavaUniversity_Develop.git --ref=refs/*:refs/remotes/origin/* # timeout=10
checking out revision f4055320d0c0a0e494e7331457310f9f7be7 (refs/remotes/origin/JenkinsGithub)
> git config core.sparsecheckout # timeout=10
> git checkout -f f4055320d0c0a0e494e7331457310f9f7be7 # timeout=10
Commit message: "update Jenkinsfile"
> git rev-list --no-walk 661e96de1b77edc0a0e494e7331457310f9f7be7 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] {
[Pipeline] node
hello world
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```