



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

**Ingeniería de Software 3
TRABAJO PRÁCTICO N°7**

Profesor: Ing. Fernando Bono

Alumno: Menel Angelo (1804789)

EJERCICIO 1:

- Utilice la versión de docker de jenkins, pero mi problema fue que el contenedor de docker no tenía instalado docker por lo tanto para realizar el último ejercicio hice lo siguiente:
 - Creó un archivo **JenkinsDockerfile** donde hago una imagen propia de Jenkins en el cual le instalo docker.

```
FROM jenkins/jenkins:lts
USER root
RUN apt update && apt install -y \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
RUN mkdir -p /etc/apt/keyrings
RUN curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
RUN echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
    | tee /etc/apt/sources.list.d/docker.list > /dev/null
RUN apt update && apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
RUN usermod -aG docker jenkins
```

- Luego buildeo la imagen usando el comando **docker image build -t {image_name}**.
- Para crear el contenedor uso el comando **docker run -it -v /var/run/docker.sock:/var/run/docker.sock -v ~/jenkins:/var/jenkins_home -p 8082:8080 -p 50000:50000 {image_name}**

EJERCICIO 3:

- Me instalo el plugin de del manage jenkins> manage plugins. Busco **Docker Pipeline** y lo agrego.
- Instalo maven en manage jenkins>global tool configuration.

EJERCICIO 4:

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Hello World'
8       }
9     }
10  }
11 }
12
```

Hello World

Save

Apply

Stage Logs (Hello)

Print Message -- Hello World (self time 6ms)

Hello World

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History

trend

Filter builds...

#1 Sep 27, 2022, 6:30 PM

Atom feed for all

Atom feed for failures

Stage view

Average stage times:

(Average full run time: ~2s)

165ms

165ms

Sep 27 15:30

No Changes

Permalinks

REST API Jenkins 2.361.1

- El script nos muestra el pipeline con todos los stages agrupados, cada **stage** se le indica un nombre y luego se instancian los **steps**, en este caso solamente es un **echo "Hello World"**.
- La salida nos muestra que se corre el **stage** Hello y nos pinte el "Hello World".

EJERCICIO 5:

Pipeline script

Script ?

```

5      // Install the Maven version configured as "M3" and add it to the path.
6      maven "M3"
7  }
8
9  stages {
10     stage('Build') {
11         steps {
12             // Get some code from a GitHub repository
13             git 'https://github.com/jglick/simple-maven-project-with-tests.git'
14
15             // Run Maven on a Unix agent.
16             sh "mvn -Dmaven.test.failure.ignore=true clean package"
17
18             // To run Maven on a Windows agent, use
19             // bat "mvn -Dmaven.test.failure.ignore=true clean package"
20         }
21
22         post {
23             // If Maven was able to run the tests, even if some of the test
24             // failed, record the test results and archive the jar file.
25             success {
26                 junit '**/target/surefire-reports/TEST-*.xml'
27                 archiveArtifacts 'target/*.jar'
28             }
29         }
30     }
31 }
32 }
33

```

GitHub + Maven

Stage Logs (Build)

Use a tool from a predefined Tool Installation -- M3 (self time 23ms)

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 26ms)

Git -- https://github.com/jglick/simple-maven-project-with-tests.git (self time 1s)

Shell Script -- mvn -Dmaven.test.failure.ignore=true clean package (self time 1min 50s)

Archive JUnit-formatted test results -- **/target/surefire-reports/TEST-*.xml (self time 103ms)

Archive the artifacts -- target/*.jar (self time 49ms)

Rename

Pipeline Syntax

Build History trend

Filter builds...

Average stage times:

(Average full run time: ~2min 1s)

#1

Sep 27 16:20

No Changes

5s

1min 52s

- Se crea el **stage** Build con el comando **git** y la url al repositorio (se hace un clone).
Se hace el comando **mvn** para compilar el proyecto maven.

EJERCICIO 6:

Script ?

```

1 pipeline {
2   agent any
3
4   tools {
5     // Install the Maven version configured as "M3" and add it to the path.
6     maven "M3"
7   }
8
9   stages {
10    stage('Build') {
11      steps {
12        // Get some code from a GitHub repository
13        git branch: "main", url: 'https://github.com/AngeloMenel98/spring-boot.git'
14
15        // Run Maven on a Unix agent.
16        sh "mvn -Dmaven.test.failure.ignore=true clean package spring-boot:repackage"
17
18      }
19
20      post {
21        // If Maven was able to run the tests, even if some of the test
22        // failed, record the test results and archive the jar file.
23        success {
24          junit '**/target/surefire-reports/TEST-*.xml'
25          archiveArtifacts 'target/*.jar'
26        }
27      }
28    }
29  }

```

Stage Logs (Build)

Use a tool from a predefined Tool Installation -- M3 (self time 17ms)

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 23ms)

Git (self time 5s)

Shell Script -- mvn -Dmaven.test.failure.ignore=true clean package spring-boot:repackage (self time 3min 33s)

Archive JUnit-formatted test results -- **/target/surefire-reports/TEST-*.xml (self time 21ms)

Archive the artifacts -- target/*.jar (self time 41ms)

Filter builds...

#1

Sep 27, 2022, 7:26 PM

Atom feed for all Atom feed for failures

Average stage times:

Declarative: Tool Install

Build

57ms

3min 39s

57ms

3min 39s

- En el comando **git** se indica la **branch** a utilizar y la **url** con el repositorio donde esta el proyecto **spring-boot**. Para compilar se agrega **spring-boot:repackage** igual que como haríamos manualmente.

EJERCICIO 7:

- Se agregan credenciales con el username y password de docker-hub en manage jenkins>manage credentials>global.

The screenshot shows the Jenkins web interface for creating a new global credential. The breadcrumb trail at the top reads: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >. The form is titled 'Kind' and has a dropdown menu set to 'Username with password'. Below this, the 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains the text 'angemenel98'. There is an unchecked checkbox labeled 'Treat username as secret'. The 'Password' field is currently empty and masked with dots. At the bottom left of the form is a blue 'Create' button.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

angemenel98

☐ Treat username as secret ?

Password ?

.....

Create

