

Ingenieria 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 tenia 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.

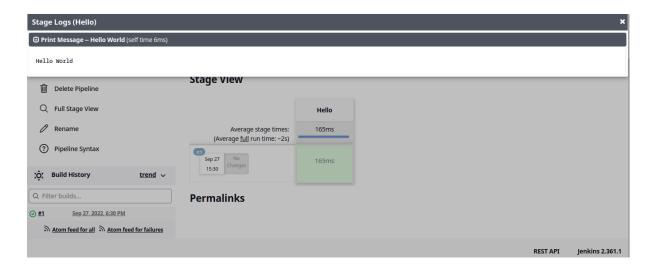
- Luego buildeo la imagen usando el comando docker image build -t {image_name}.
- Para crear el contenedor uso el comando docker run -it -v
 /var/run/docker.sock:/var/run/docker.sock -v
 ~/jenkins:/var/jenkins_home -p 8082:8080 -p 50000:50000 {image_name}

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 { Hello World agent any 3 4 stages { stage('Hello') { 5 + 6 + steps { echo 'Hello World' 8 9 10 } 11 } 12 Save Apply



- 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 printea el "Hello World".

EJERCICIO 5:

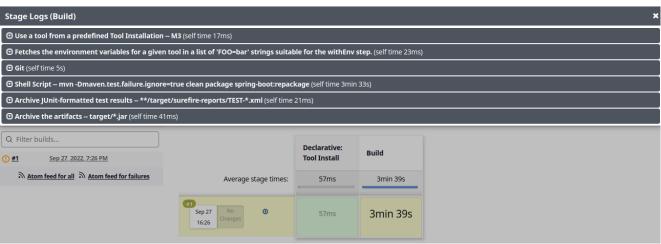
```
Pipeline script
  Script ?
                  // Install the Maven version configured as "M3" and add it to the path.
                                                                                                                             GitHub + Maven
      6
                  maven "M3'
      9 +
              stages {
                  stage('Build') {
     10 -
     11 -
                      steps {
   // Get some code from a GitHub repository
     12
                          git 'https://github.com/jglick/simple-maven-project-with-tests.git'
     14
     15
                          // Run Maven on a Unix agent.
                          sh "mvn -Dmaven.test.failure.ignore=true clean package"
     16
     17
                          // To run Maven on a Windows agent, use
     19
                          // bat "mvn -Dmaven.test.failure.ignore=true clean package"
     20
     21
     22 +
     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 {
                              junit '**/target/surefire-reports/TEST-*.xml'
     26
     27
                              archiveArtifacts 'target/*.jar'
     28
     29
     30
     31
```



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:

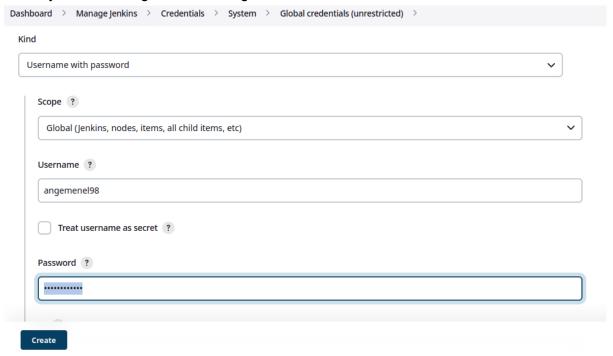




 En el comando git se indica la branch a utilizar y la url con el repositorio donde esta el proyecto <u>spring-boot</u>. Para compilar se agrega spring-boot:repackage igual que como haríamos manualmente.

EJERCICIO 7:

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



```
4 =
                                                   tools {
    5
                                                                         // Install the Maven version configured as "M3" and add it to the path.
     6
     7
     8
    9 +
                                                   stages {
10 -
                                                                         stage('Build Spring-Boot') {
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 {
                                                                                                                         // If Maven was able to run the tests, even if some of the test % \left( 1\right) =\left( 1\right) \left( 
21
22
                                                                                                                          // failed, record the test results and archive the jar file.
23 -
                                                                                                                          success {
                                                                                                                                                 junit '**/target/surefire-reports/TEST-*.xml'
24
25
                                                                                                                                                 archiveArtifacts 'target/*.jar'
26
27
28
29
30 +
                                                                           stage('Build Image') {
31 ▼
                                                                                                  steps{
32 -
                                                                                                                          script{
                                                                                                                                                  def image = docker.build "angemenel98/spring-boot-jenkins"
33
                                                                                                                                                  docker.withRegistry('https://registry.hub.docker.com', 'docker-hub-credentials') {
34 ₹
35
                                                                                                                                                                          image.push("${env.BUILD_NUMBER}")
36
                                                                                                                                                                          image.push("latest")
37
38
```

- Se crearon 2 stages uno para el build de spring y otro para el build de la image y subirla a docker-hub.
- Se corrio el pipeline y podemos ver que se subio correctamente en docker-hub:

