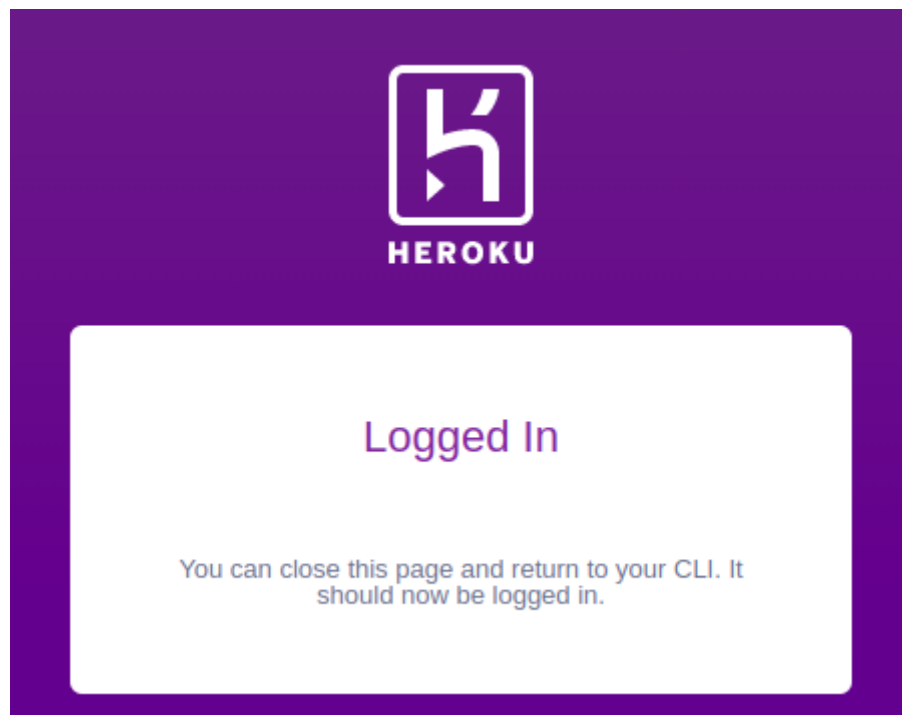


1- Configurando Heroku



2- Creando y Desplegando la aplicación Payroll

Se modificó el dockerfile como se indicaba y se ejecutó el comando heroku create

```
manuel@manuel-Aspire-A315-54K:~/Documents/Repo Ing Soft III/TP 12 - Despliegue/spring$ heroku create
Creating app... done, ● limitless-taiga-46682
```

el dockerfile quedó

```
FROM maven:3.5.2-jdk-8-alpine AS MAVEN_TOOL_CHAIN
COPY pom.xml /tmp/
RUN mvn -B dependency:go-offline -f /tmp/pom.xml -s
/usr/share/maven/ref/settings-docker.xml
COPY src /tmp/src/
WORKDIR /tmp/
RUN mvn -B -s /usr/share/maven/ref/settings-docker.xml package

FROM openjdk:8-jre-alpine

EXPOSE 8080

RUN mkdir /app

COPY --from=MAVEN_TOOL_CHAIN /tmp/target/*.jar /app/spring-boot-application.jar

ENV PORT=8080

CMD ["java", "-Xms32m", "-Xmx128m", "-jar", "-Dserver.port=${PORT}",
"-Djava.security.egd=file:/dev/./urandom", "/app/spring-boot-application.jar"]
```

```
manuel@manuel-Aspire-A315-54K:~/Documents/Repo Ing Soft III/TP 12 - Despliegue/spring$ heroku container:release web --app=desolate-fortress-44132
Releasing images web to desolate-fortress-44132... done
```

← → ↻ ⚠ Not secure | secure-reaches-68810.herokuapp.com

```
{"message":"Spring boot says hello from a Docker container"}
```

3-

```
pipeline {
  agent any
  tools {
    // Install the Maven version configured as "M3" and add it to the path.
    maven "M3"
  }
  stages {
    stage('Build') {
      steps {
        // Get some code from a GitHub repository
        git branch: 'main', url:
          'https://github.com/Manuel-Bobadilla/spring-boot.git'
        // Run Maven on a Unix agent.
        sh "mvn -Dmaven.test.failure.ignore=true clean package"
        // To run Maven on a Windows agent, use
        // bat "mvn -Dmaven.test.failure.ignore=true clean package"
      }
      post {
        // If Maven was able to run the tests, even if some of the test
        // failed, record the test results and archive the jar file.
        success {
          junit '**/target/surefire-reports/TEST-*.xml'
          archiveArtifacts 'target/*.jar'
        }
      }
    }
  }
  stage('Heroku Upload'){
    steps {
      sh 'heroku container:login'
      sh 'heroku create mbsb'
      sh 'heroku container:push web --app=mbsb'
      sh 'heroku container:release web --app=mbsb'
    }
  }
}
```