## Jenkins script

```
pipeline {
    agent any

    triggers {
        pollSCM('* * * * *')
    }
    // Got permission denied while trying to connect to the Docker daemon socket at unix.
    // sudo usermod -a -G docker jenkins
    // restart jenkins server ->  sudo service jenkins restart
    stages {

        stage('Maven Compile') {
            steps {
                echo '---------------- This is a compile phase ----------'
                sh 'mvn clean compile'
            }
        }

         stage('Maven Test') {
            steps {
                echo '---------------- This is a Test phase ----------'
                sh 'mvn clean test'
            }
        }

        stage('Maven Build') {
             steps {
                echo '---------------- This is a build phase ----------'
                sh 'mvn clean package -DskipTests'
            }
        }

        stage('Docker Build') {
            steps {
                echo '---------------- This is a build docker image phase ---------'
                sh '''
                    docker image build -t ecom-webservice .
```

```
            '''
        }
    }


    stage('Docker Deploy') {
        steps {
            echo '----------------- This is a docker deploment phase ----------'
            sh '''
             (if  [ $(docker ps -a | grep ecom-webservice | cut -d " " -f1) ]; then \
                 echo $(docker rm -f ecom-webservice); \
                 echo "---------------- successfully removed ecom-webservice ----------------" \
               else \
                echo OK; \
             fi;);
        docker container run --restart always --name ecom-webservice -p 8081:8081 -d ecom-webservice
            '''
        }
    }
  }
}
```

**Docker file**

```
FROM openjdk:8-jdk

COPY ./target/ecom-webservice-0.0.1-SNAPSHOT.jar ecom-webservice-0.0.1-SNAPSHOT.jar

CMD ["java" ,"-jar","ecom-webservice-0.0.1-SNAPSHOT.jar"]

RUN echo "jenkins ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers


#-> Create a image for springboot webservice

# -->  Dockerfile

# - -> docker image build -t <imagetagname> .


#-> run container by using custom image

# -> docker container run  --name <ecom-web-service> -p 8081:8081 -d <ecom-webservice>
```

#-> verify log

# -> docker container log <container-name>