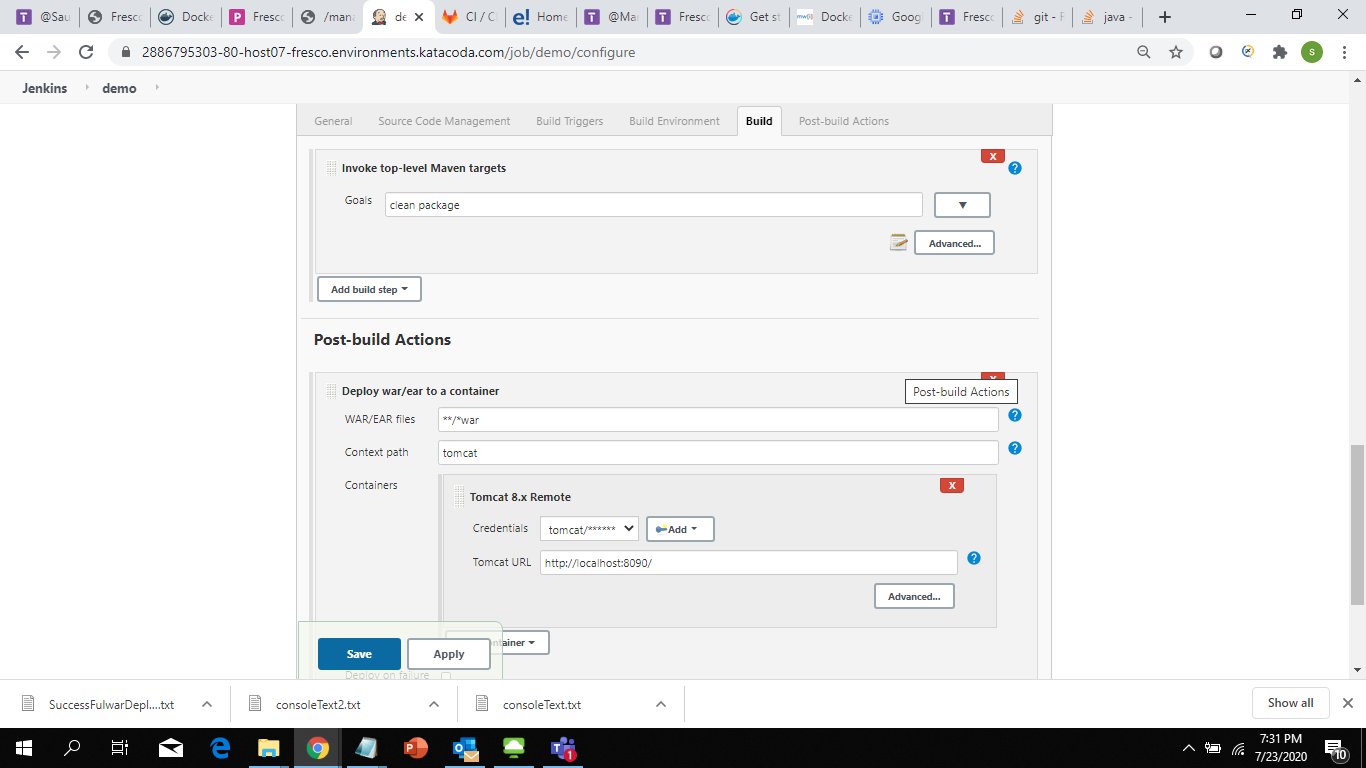
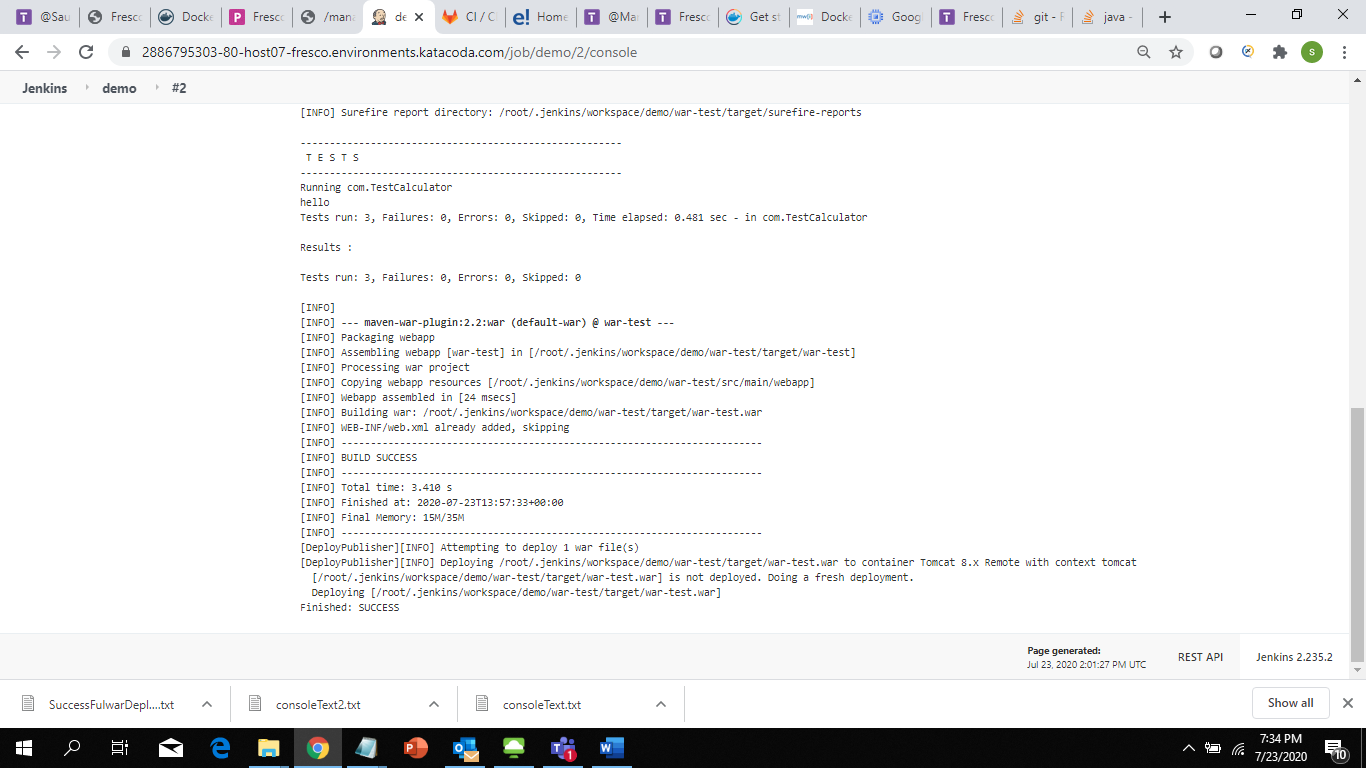
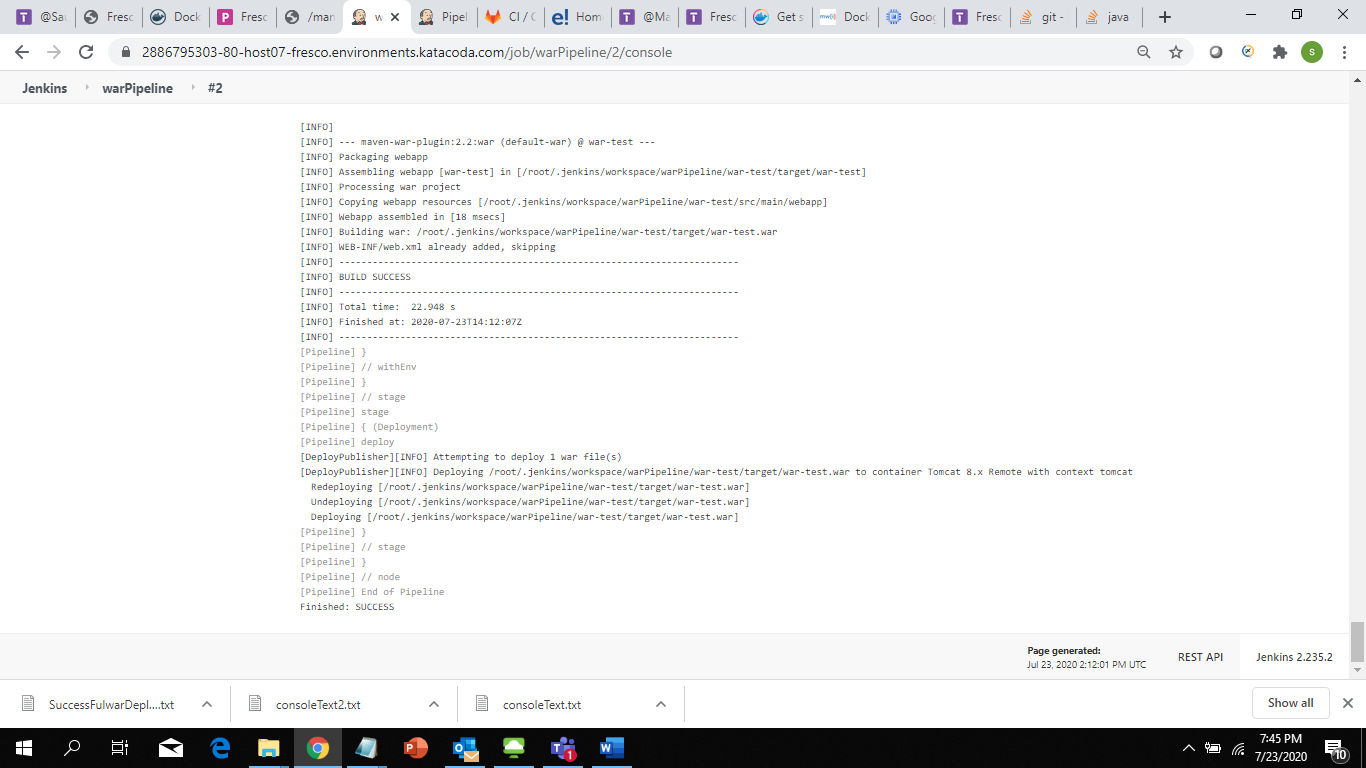
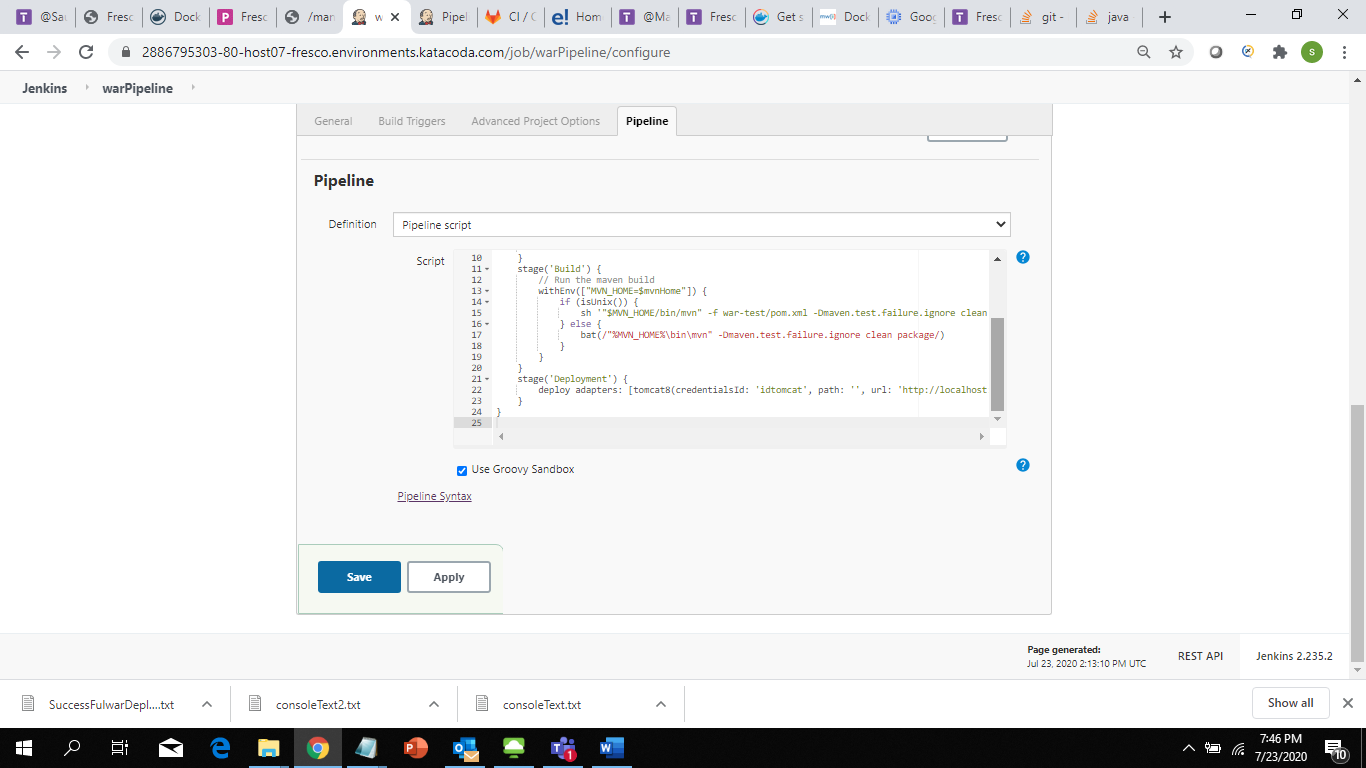
War deployment with external tomcat - Freestyle:

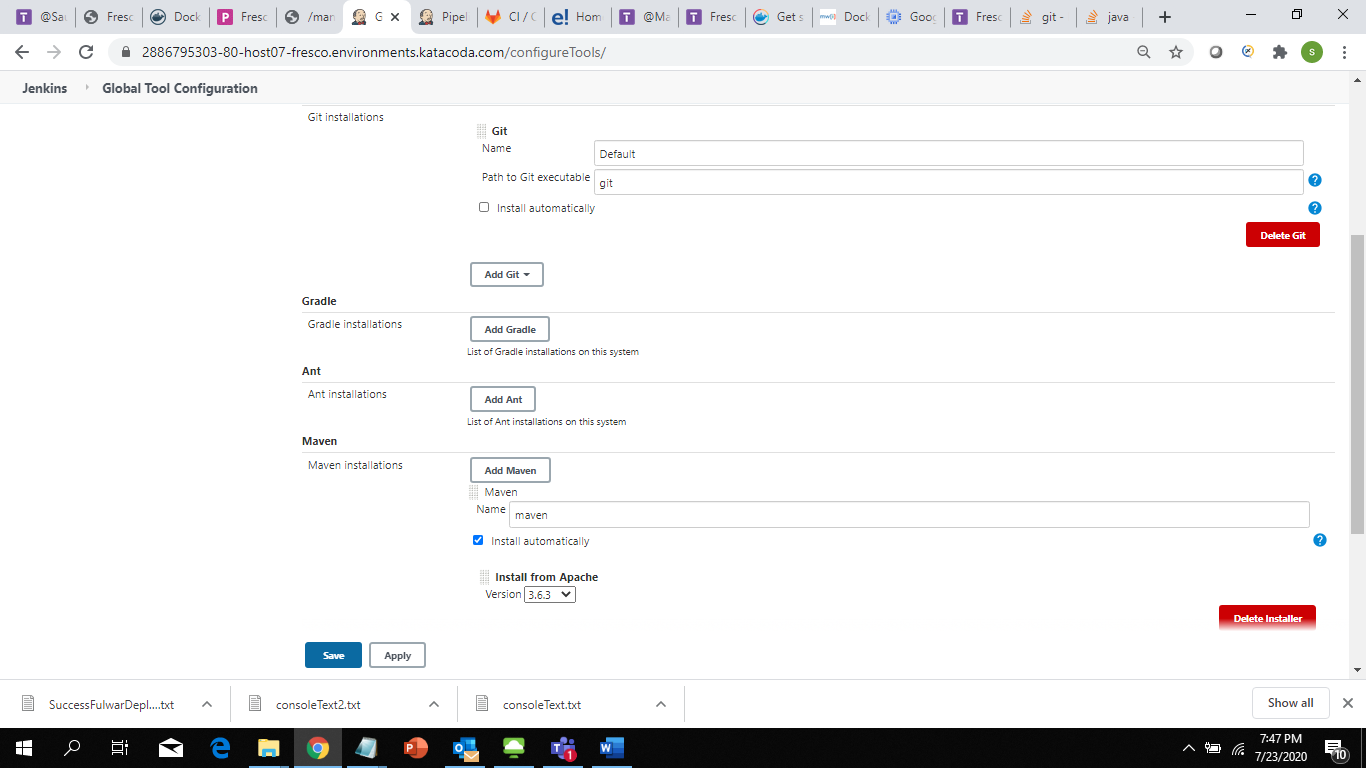




War deployment with pipeline:







**With war pipeline:**

node {

def mvnHome

stage('Preparation') { // for display purposes

// Get some code from a GitHub repository

git 'https://github.com/sivasusil/BootCamp05.git'

// Get the Maven tool.

// \*\* NOTE: This 'M3' Maven tool must be configured

// \*\* in the global configuration.

mvnHome = tool 'maven'

}

stage('Build') {

// Run the maven build

withEnv(["MVN\_HOME=$mvnHome"]) {

if (isUnix()) {

sh '"$MVN\_HOME/bin/mvn" -f war-test/pom.xml -Dmaven.test.failure.ignore clean package'

} else {

bat(/"%MVN\_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package/)

}

}

}

stage('Deployment') {

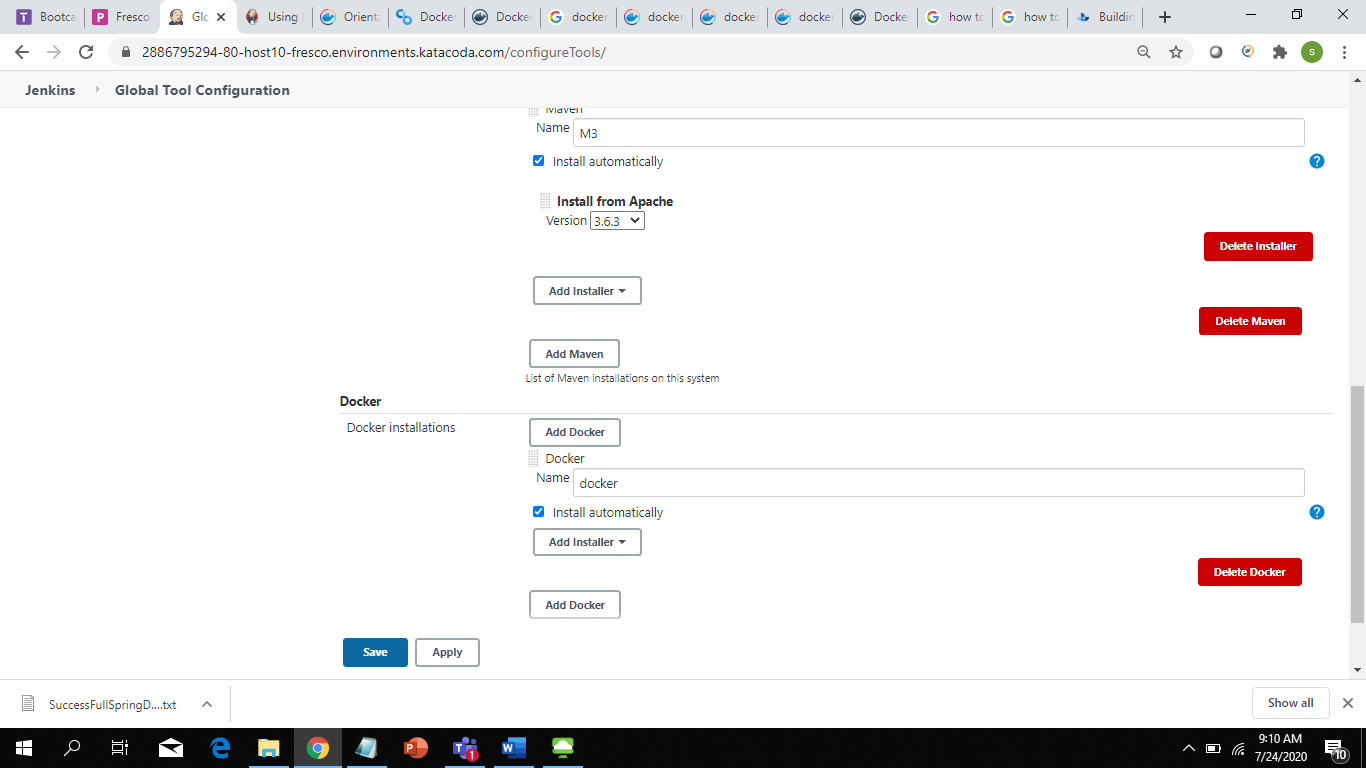
deploy adapters: [tomcat8(credentialsId: 'idtomcat', path: '', url: 'http://localhost:8090/')], contextPath: 'tomcat', war: '\*\*/\*war'

}

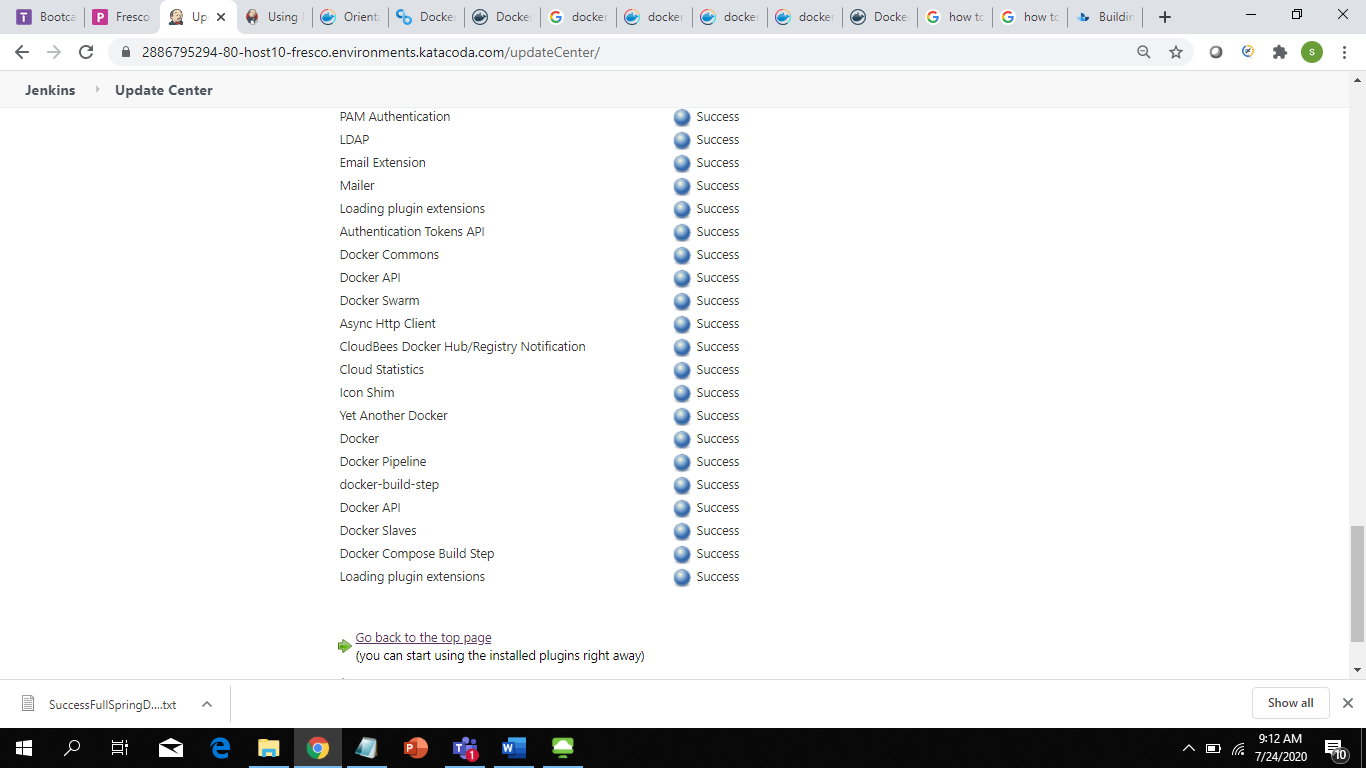
}

**Spring data:**

**Preparation:**



**Plugins:**



Docker for spring boots:

node {

def mvnHome

stage('Preparation') { // for display purposes

// Get some code from a GitHub repository

git 'https://github.com/sivasusil/BootCamp05.git'

// Get the Maven tool.

// \*\* NOTE: This 'M3' Maven tool must be configured

// \*\* in the global configuration.

mvnHome = tool 'M3'

}

stage('Build') {

// Run the maven build

withEnv(["MVN\_HOME=$mvnHome"]) {

if (isUnix()) {

sh '"$MVN\_HOME/bin/mvn" -f cicd-pipeline/pom.xml clean package'

} else {

bat(/"%MVN\_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package/)

}

}

}

stage('Docker') {

def springbootImage = ''

docker.withRegistry('', 'dockerHub') {

springbootImage = docker.build("sivasusil/springbootimage:1.0.0", "./cicd-pipeline")

/\* Push the container to the custom Registry \*/

springbootImage.push()

}

docker.image('sivasusil/springbootimage:1.0.0').withRun('-p 3306:8090') {

/\* do things \*/

}

}

stage('DockerFromServer') {

sh 'docker build ./cicd-pipeline -t sivasusil/springbootapp:1.0.0'

withCredentials([usernamePassword(credentialsId: 'dockerHub', passwordVariable: 'password', usernameVariable: 'username')]) {

sh 'docker login -u="$username" -p="$password"'

sh 'docker push sivasusil/springbootapp:1.0.0'

}

}

stage('Deply to Cloud Foundry'){

withCredentials([usernamePassword(credentialsId: 'PCF', passwordVariable: 'password', usernameVariable: 'username')]) {

sh 'cf login -a "https://api.run.pivotal.io" -u "$username" -p "$password" -o "sivasusil\_org" -s "development"'

}

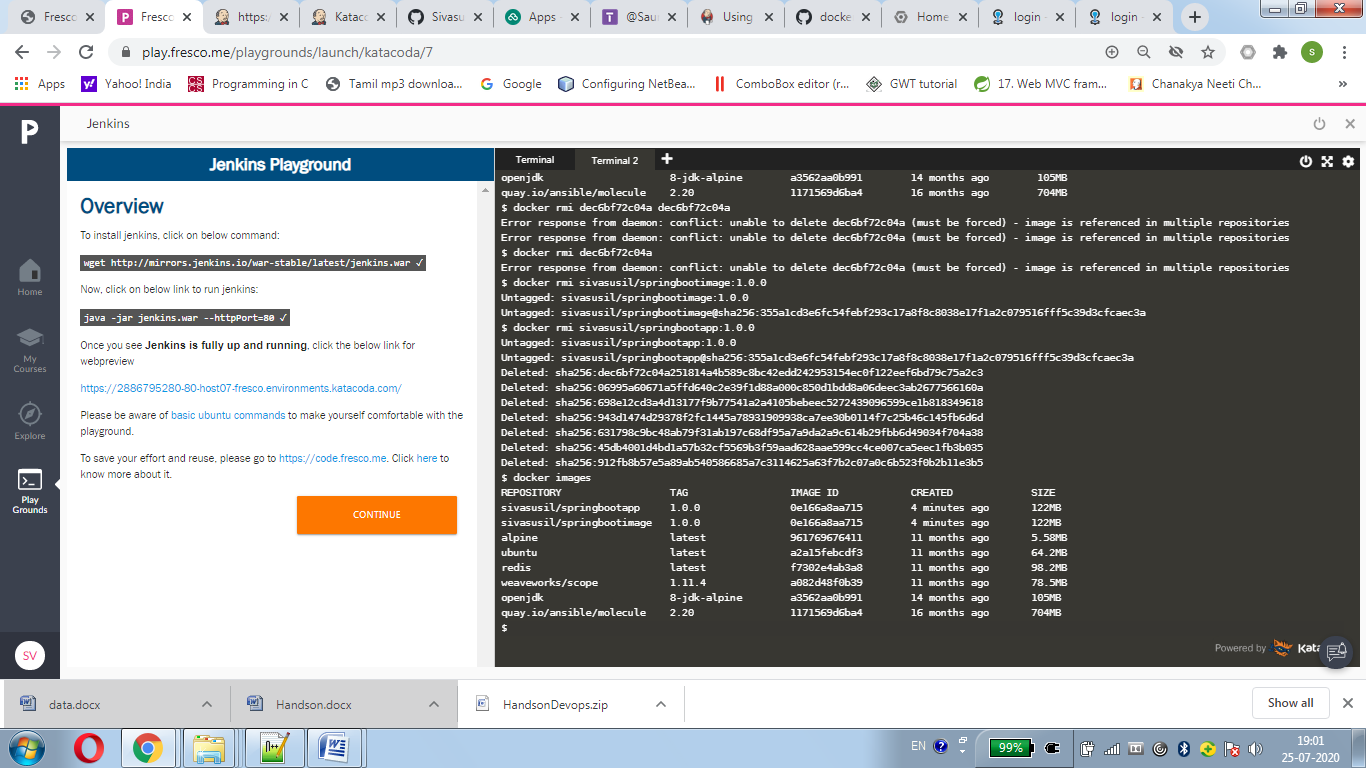
sh 'cf push springboot-1.0.0 --docker-image sivasusil/springbootapp:1.0.0'

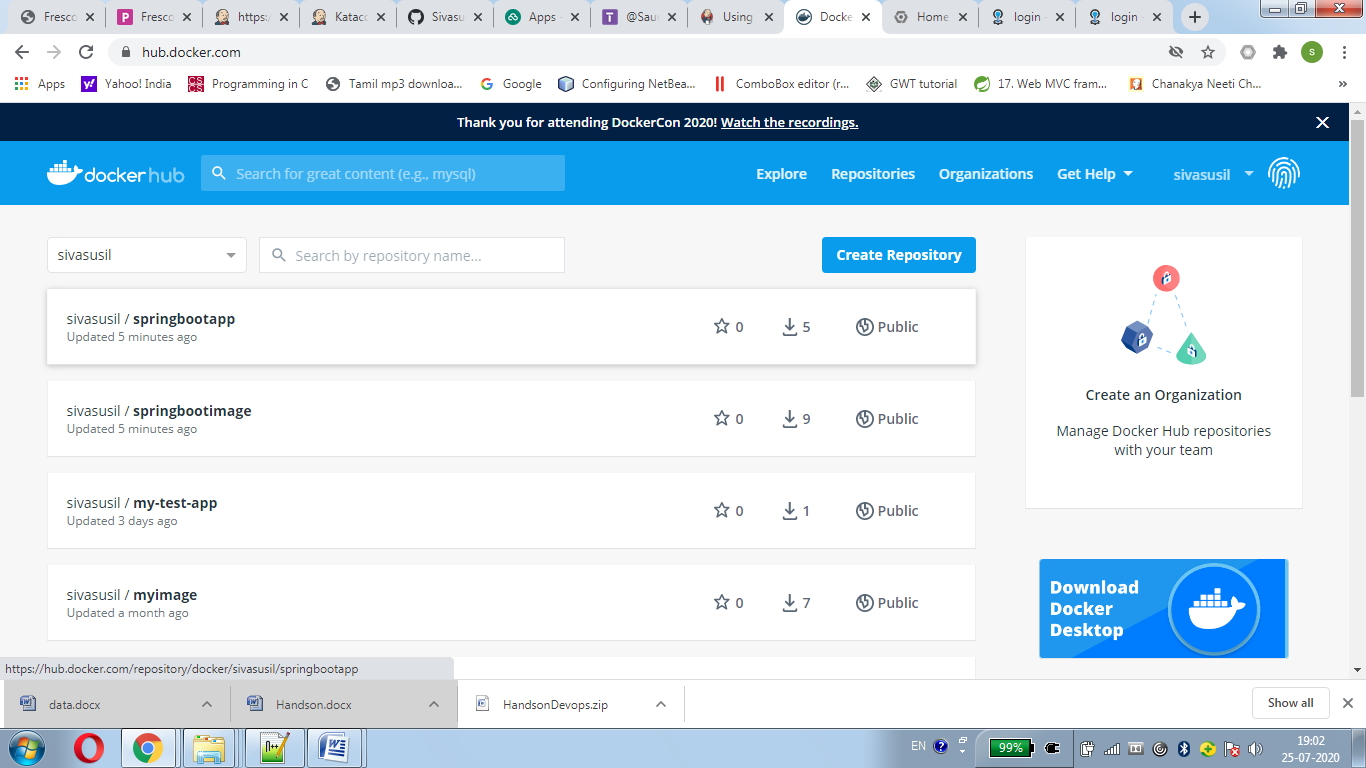
//

}

}

Docker images published:





Cloud foundry:

