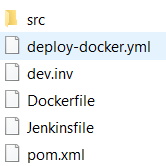
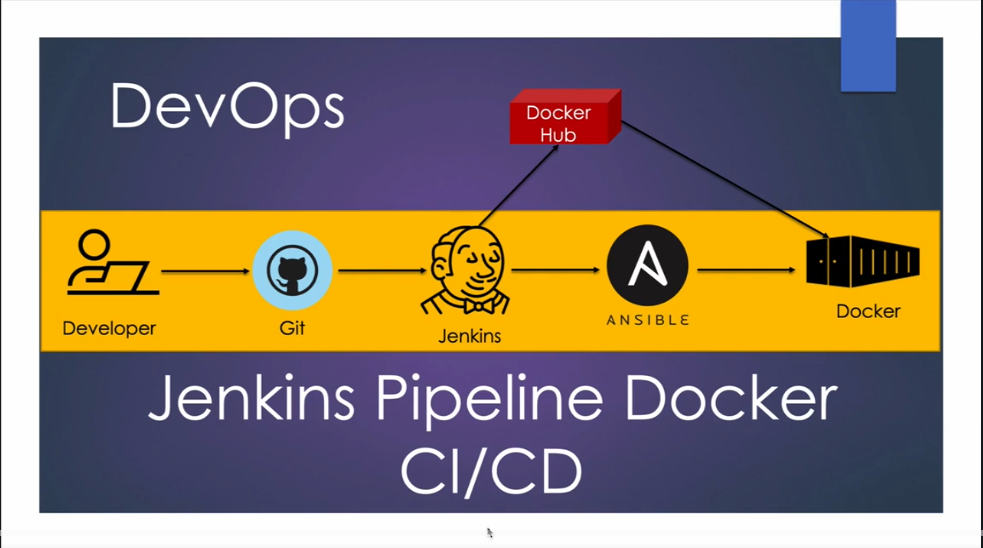
**Jenkins Ansible Docker Intégration**

Structure du projet GIT





Chaine CICI



**-- Jenkins --**

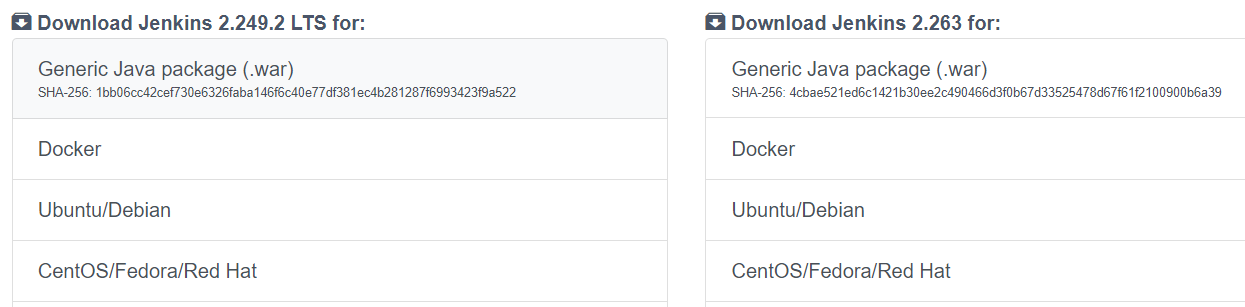
Plan

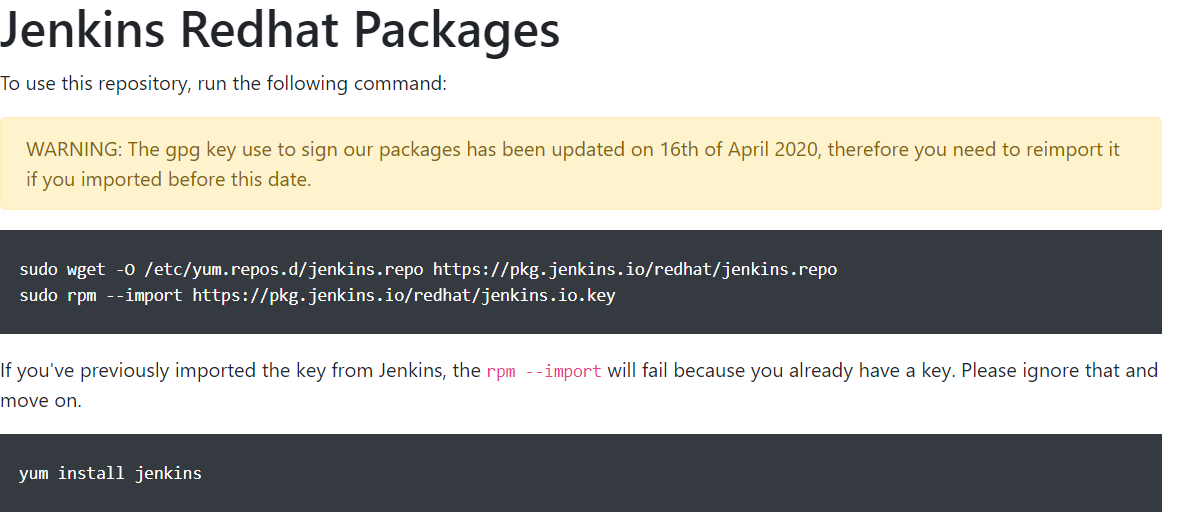
1. Download jznkins
2. Install Java
3. Start Jenkin server
4. Install maven & Git

**Jenkins Pipeline**

1. **Download jenkins**

<https://www.jenkins.io/download/>





[root@ip-172-31-28-93 opt]# sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat/jenkins.repo>

[root@ip-172-31-28-93 opt]# sudo rpm --import <https://pkg.jenkins.io/redhat/jenkins.io.key>

[root@ip-172-31-28-93 opt]# yum install jenkins

1. Install java

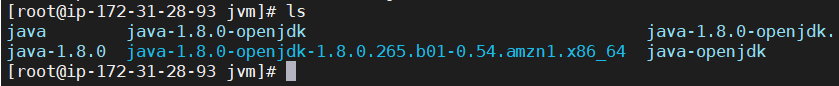
[root@ip-172-31-28-93 ~]# yum remove java-1.7\*

[root@ip-172-31-28-93 ~]# yum install java-1.8.\*

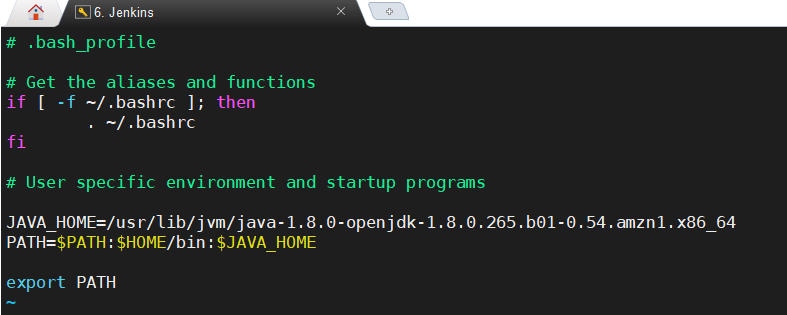
[root@ip-172-31-28-93 ~]# cd ~

[root@ip-172-31-28-93 ~]# vi .bash\_profile

[root@ip-172-31-28-93 ~]# cd /usr/lib/jvm



[root@ip-172-31-28-93 ~]# vi .bash\_profile

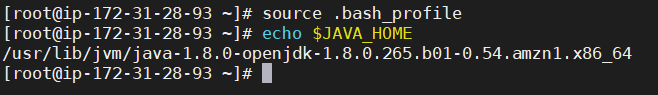


[root@ip-172-31-28-93 ~]# source .bash\_profile

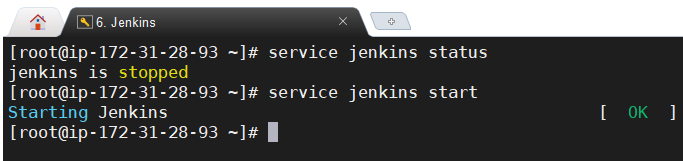
[root@ip-172-31-28-93 ~]# echo $JAVA\_HOME

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.265.b01-0.54.amzn1.x86\_64

[root@ip-172-31-28-93 ~]#

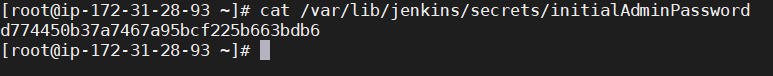


1. **Start Jenkins Server**

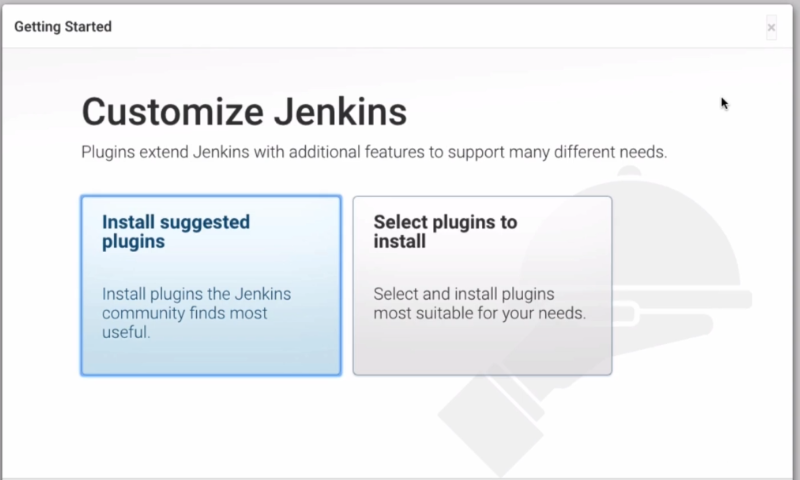


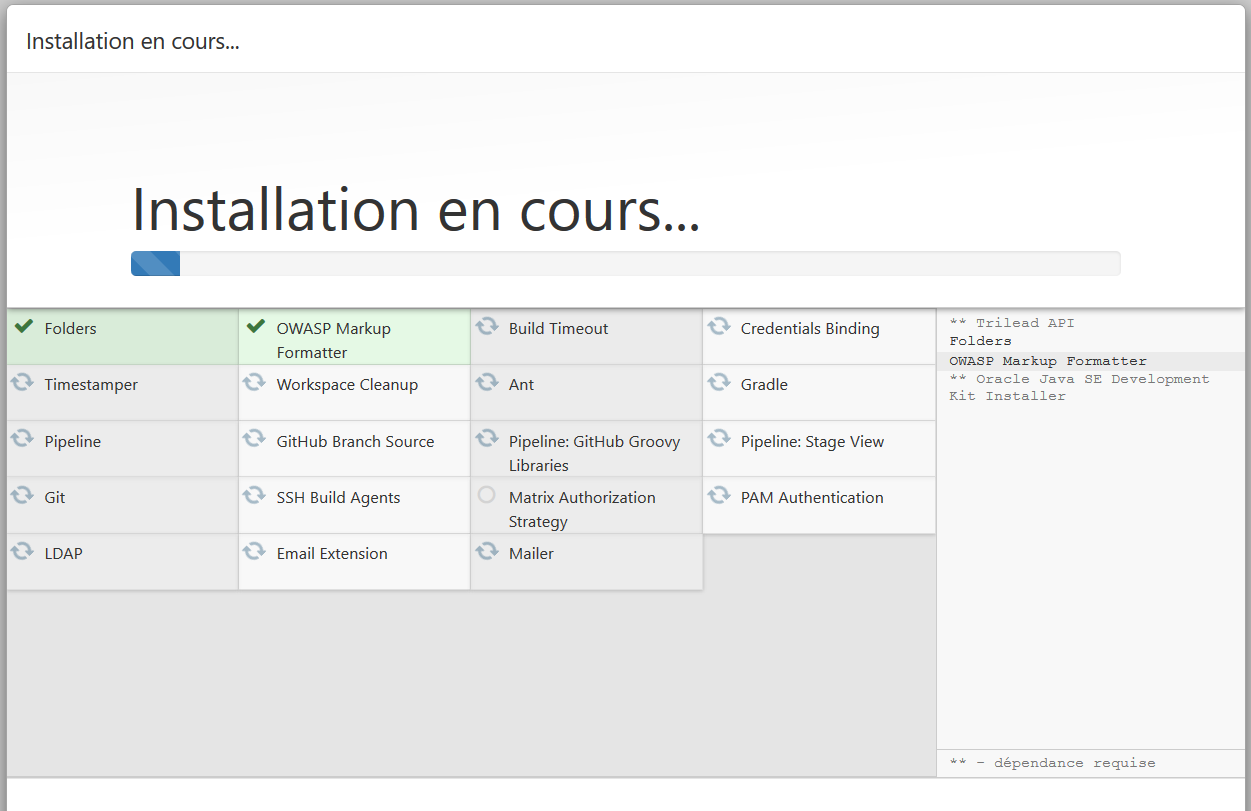
<http://15.237.108.80:8080/>











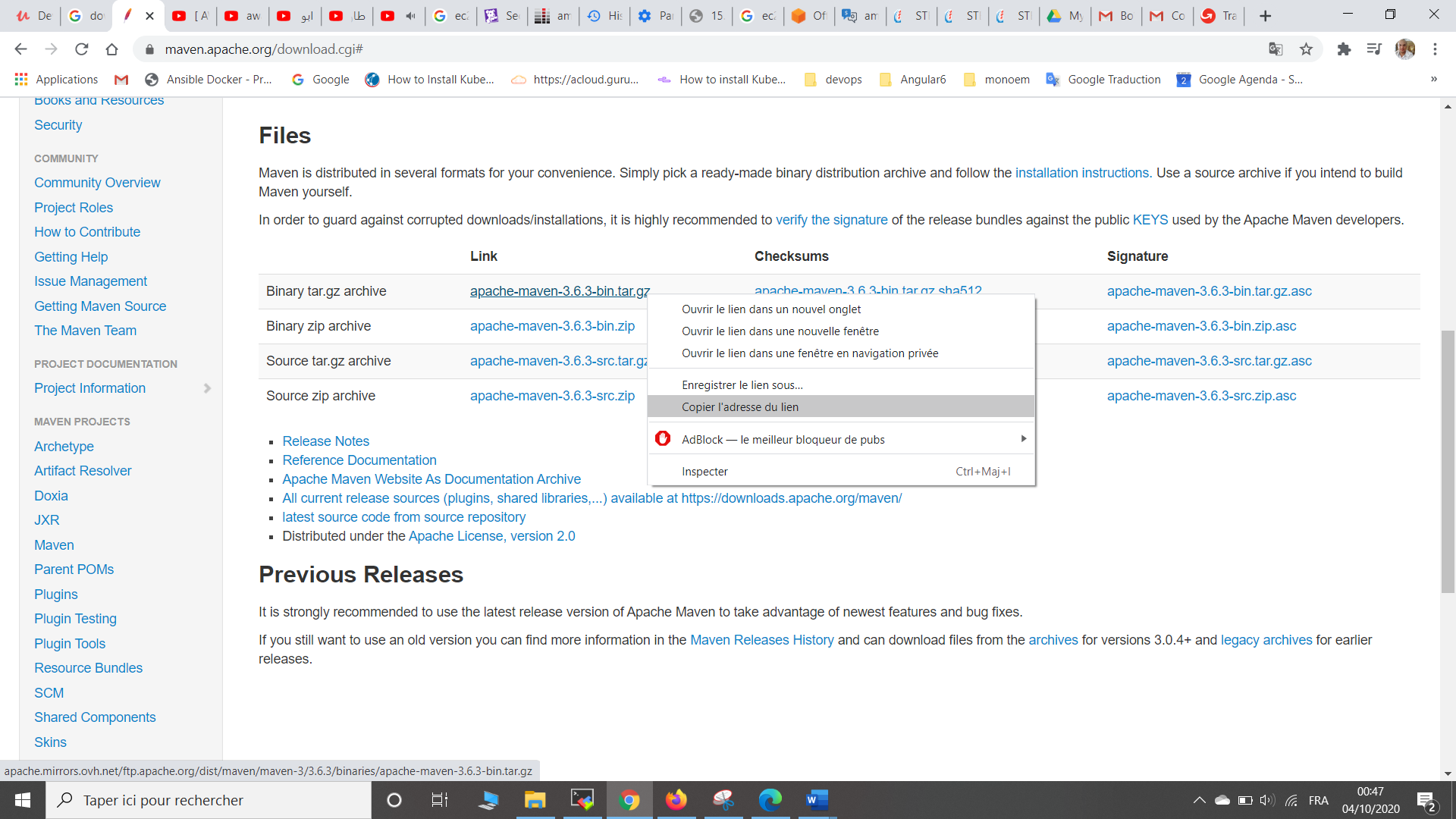


[root@ip-172-31-28-93 ~]# sudo chkconfig jenkins on

1. **Install Maven And Git on Jenkins Box**

Download maven

<https://maven.apache.org/download.cgi>



[root@ip-172-31-28-93 opt]# wget <https://apache.mirrors.benatherton.com/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz>

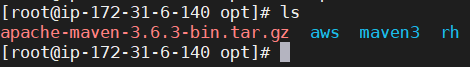
[root@ip-172-31-28-93 opt]# ls

apache-maven-3.6.3-bin.tar.gz aws jenkins

[root@ip-172-31-28-93 opt]#

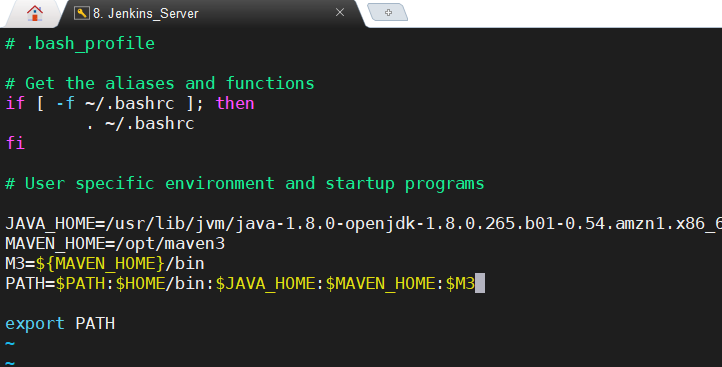
[root@ip-172-31-28-93 opt]# tar -xvzf apache-maven-3.6.3-bin.tar.gz

[root@ip-172-31-6-140 opt]# mv apache-maven-3.6.3 maven3

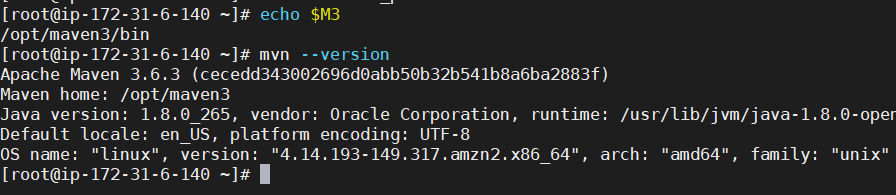


[root@ip-172-31-28-93 opt]# cd ~

[root@ip-172-31-28-93 ~]# vi .bash\_profile



[root@ip-172-31-28-93 ~]# source .bash\_profile



Install GIT

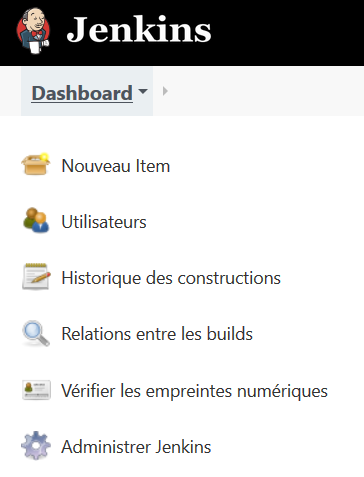
[root@ip-172-31-28-93 ~]# yum install git -y

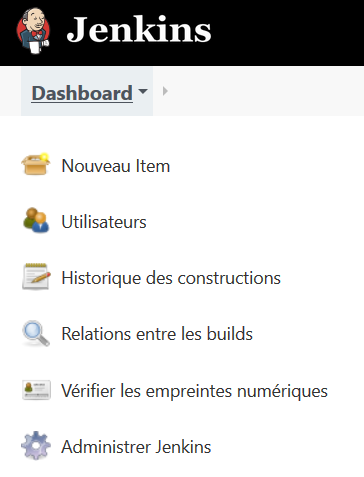
[root@ip-172-31-28-93 ~]# git --version

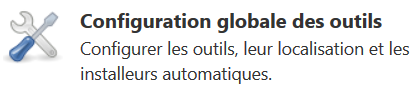
git version 2.18.4

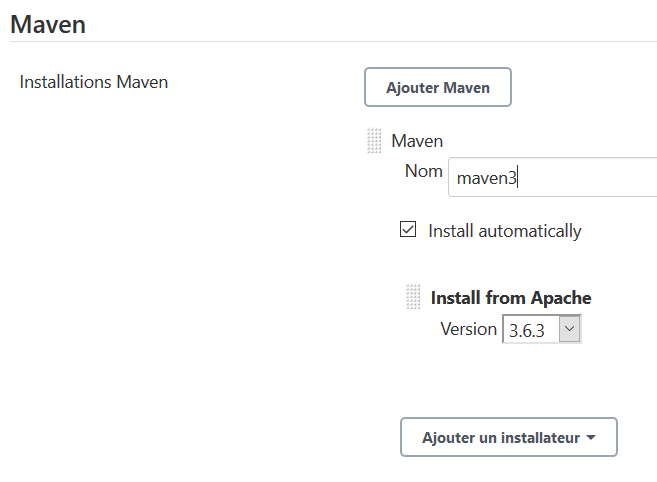
[root@ip-172-31-28-93 ~]#

Install maven from jenkins ui



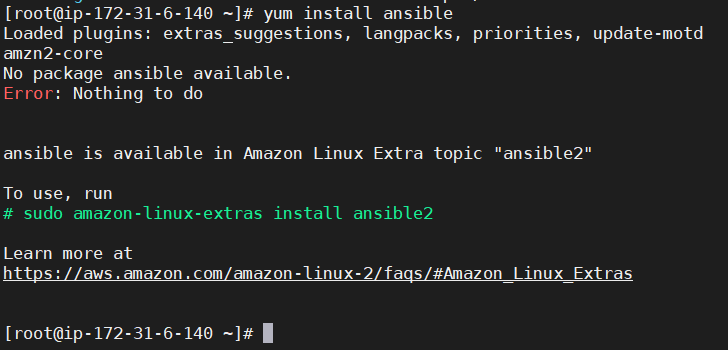




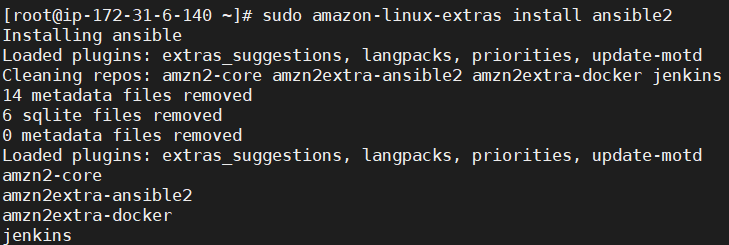


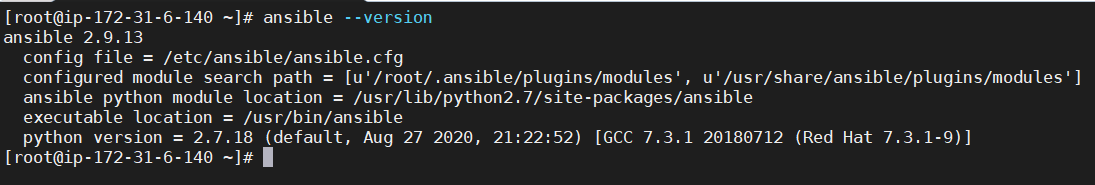
1. **Install and integration Ansible in Jenkins**

[root@ip-172-31-6-140 ~]# yum install ansible

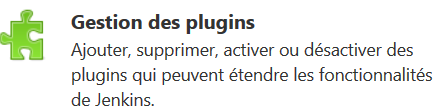


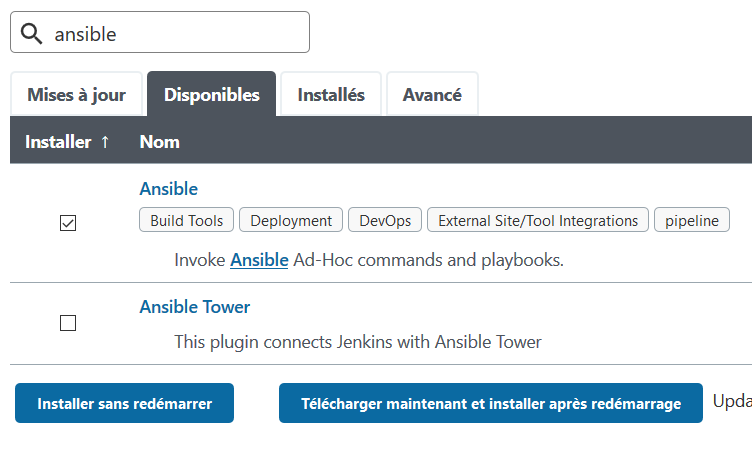
[root@ip-172-31-6-140 ~]# sudo amazon-linux-extras install ansible2 -y

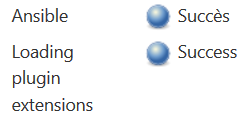


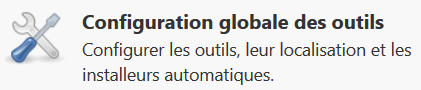


**Integration ansible in Jenkins**





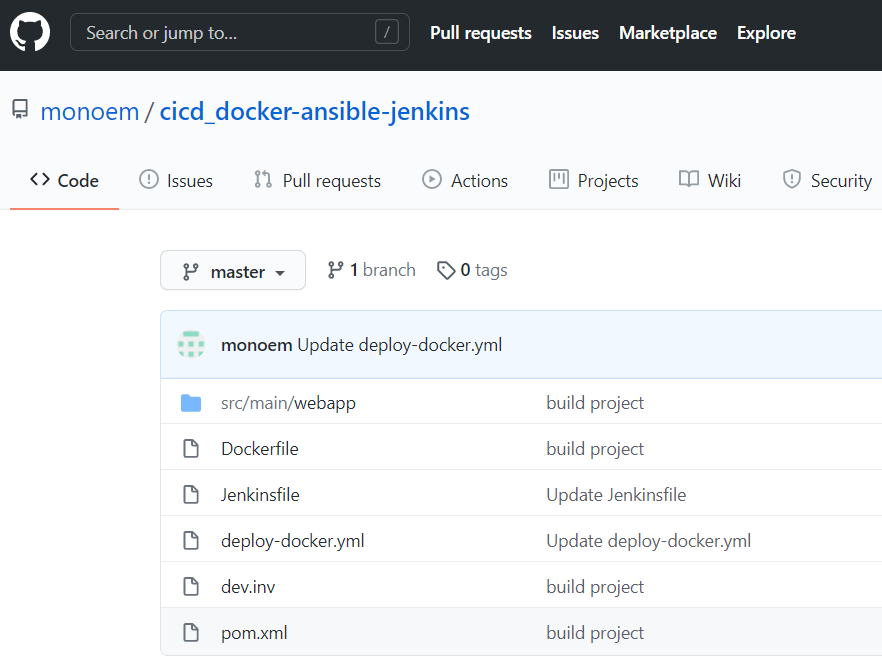


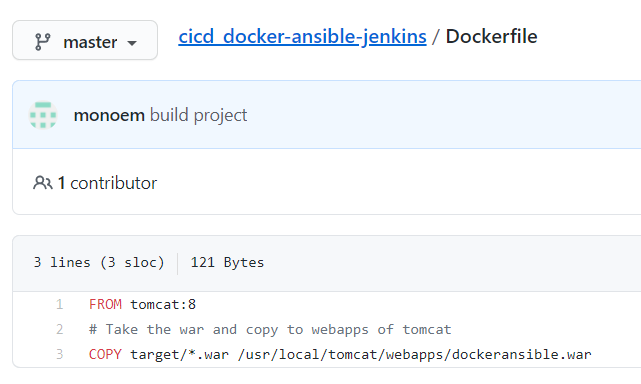






**Take the project** <https://github.com/monoem/cicd_docker-ansible-jenkins>



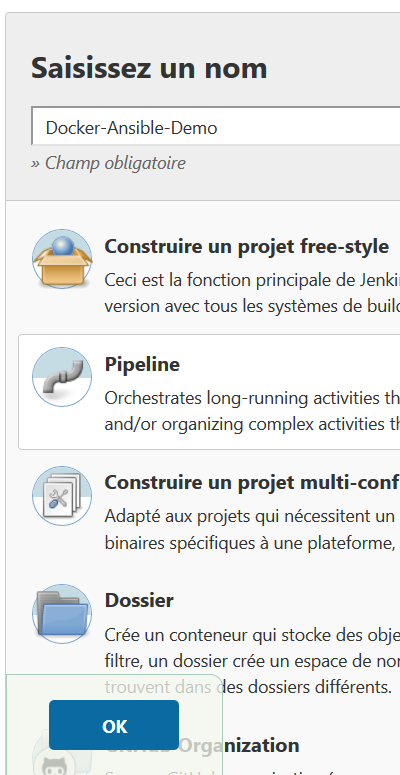


FROM

tomcat:8

# Take the war and copy to webapps of tomcat

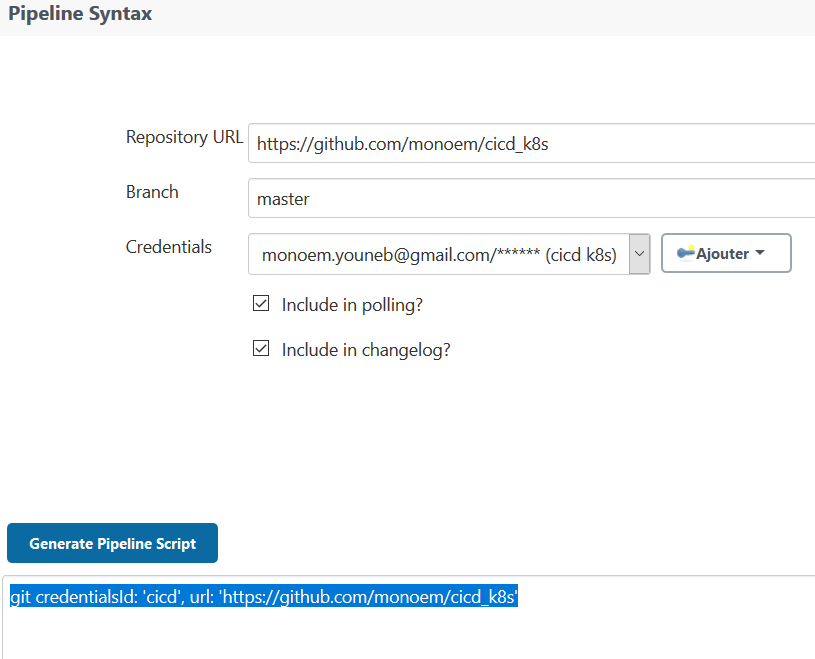
COPY target/\*.war /usr/local/tomcat/webapps/dockeransible.war



**-- Ansible Playbook - -**

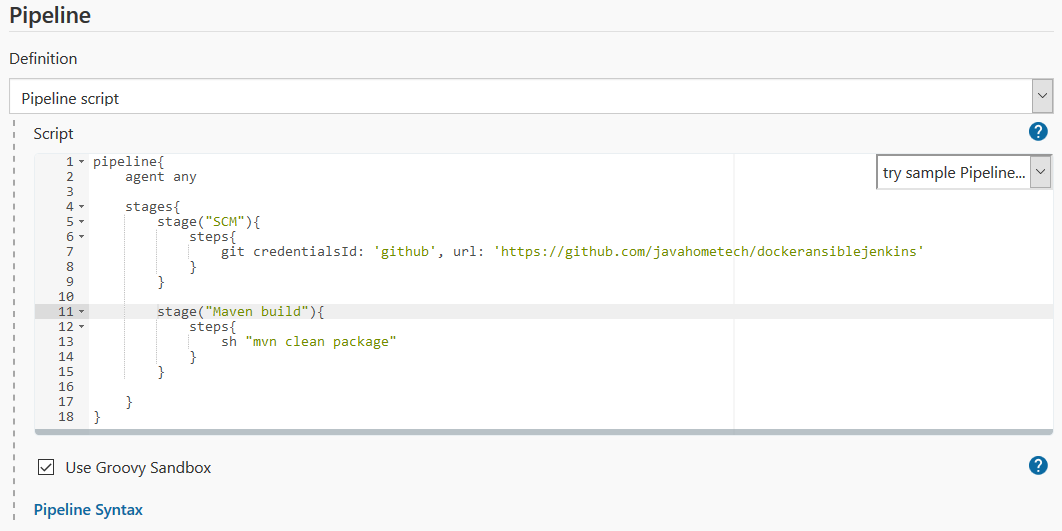
**Ansible Playbook**





git credentialsId: 'github', url: 'https://github.com/javahometech/dockeransiblejenkins'





pipeline{

agent any

stages{

stage("SCM"){

steps{

git credentialsId: 'github', url: 'https://github.com/javahometech/dockeransiblejenkins'

}

}

stage("Maven build"){

steps{

sh "mvn clean package"

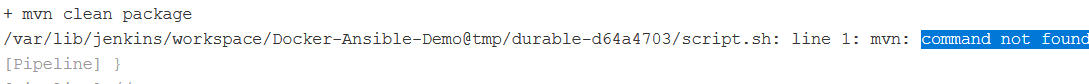
}

}

}

}

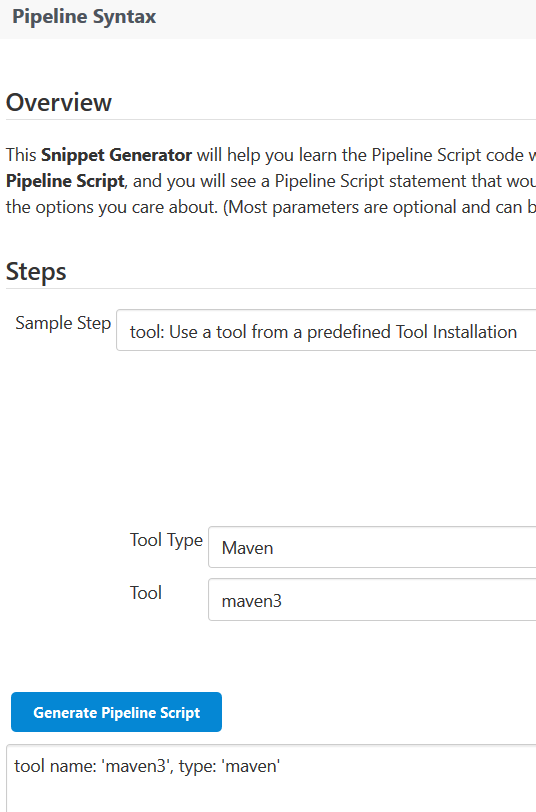






We must declar mvn





tools{

maven 'maven3'

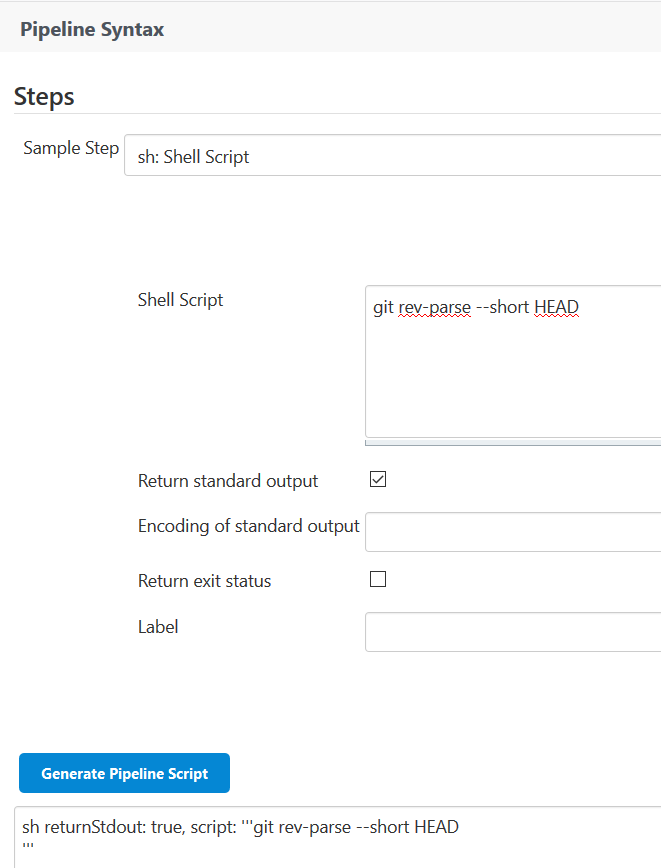
}





**Create Docker image**





pipeline{

agent any

tools{

maven 'maven3'

}

stages{

stage("SCM"){

steps{

git credentialsId: 'cicd', url: 'https://github.com/monoem/cicd\_k8s'

}

}

stage("Maven build"){

steps{

sh "mvn clean package"

}

}

stage("Docker build"){

steps{

sh "Docker build -t monoem/cicd\_k8s:0.0.1"

}

}

}

}

def getVersionDokerImage(){

def commitHash = sh label : '', returnStdout: true, script: 'git rev-parse --short HEAD'

return commitHash

}

Create environment Bloc

environment{

DOCKER\_TAG = getVersionDokerImage()

}

result

pipeline{

agent any

tools{

maven 'maven3'

}

environment{

DOCKER\_TAG = getVersionDokerImage()

}

stages{

stage("SCM"){

steps{

git credentialsId: 'cicd', url: 'https://github.com/monoem/cicd\_k8s'

}

}

stage("Maven build"){

steps{

sh "mvn clean package"

}

}

stage("Docker build"){

steps{

sh "docker build -t monoem/cicd\_k8s:0.0.1"

}

}

}

}

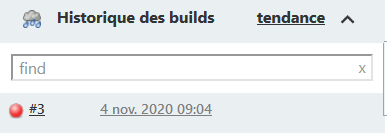
def getVersionDokerImage(){

def commitHash = sh label : '', returnStdout: true, script: 'git rev-parse --short HEAD'

return commitHash

}

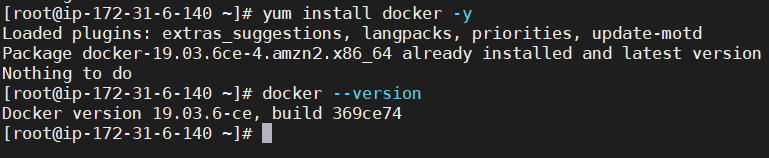






1. **Install Docker in jenkins machine**

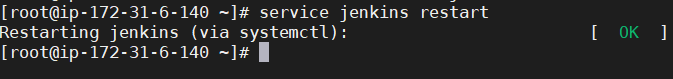
[root@ip-172-31-6-140 ~]# yum install docker -y



Add Jenkins user to docker group

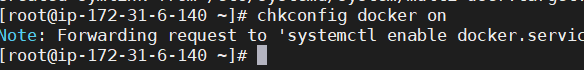


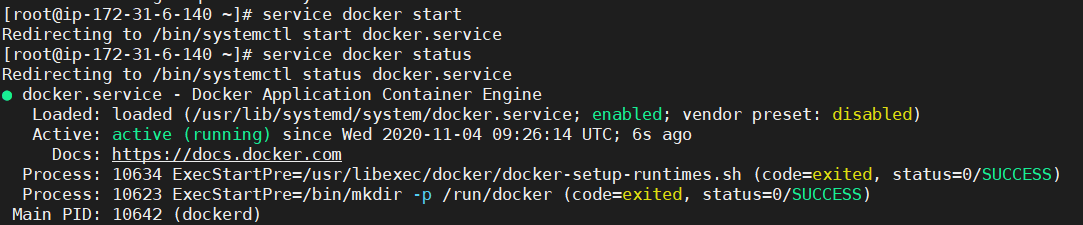
Restart jenkins



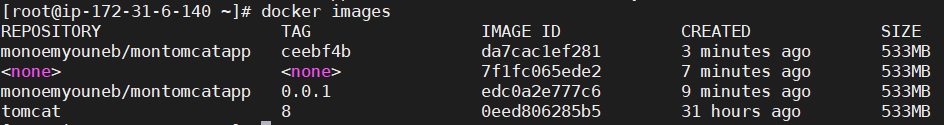
Macke docker up to start after jenkins rebbots

[root@ip-172-31-6-140 ~]# chkconfig docker on



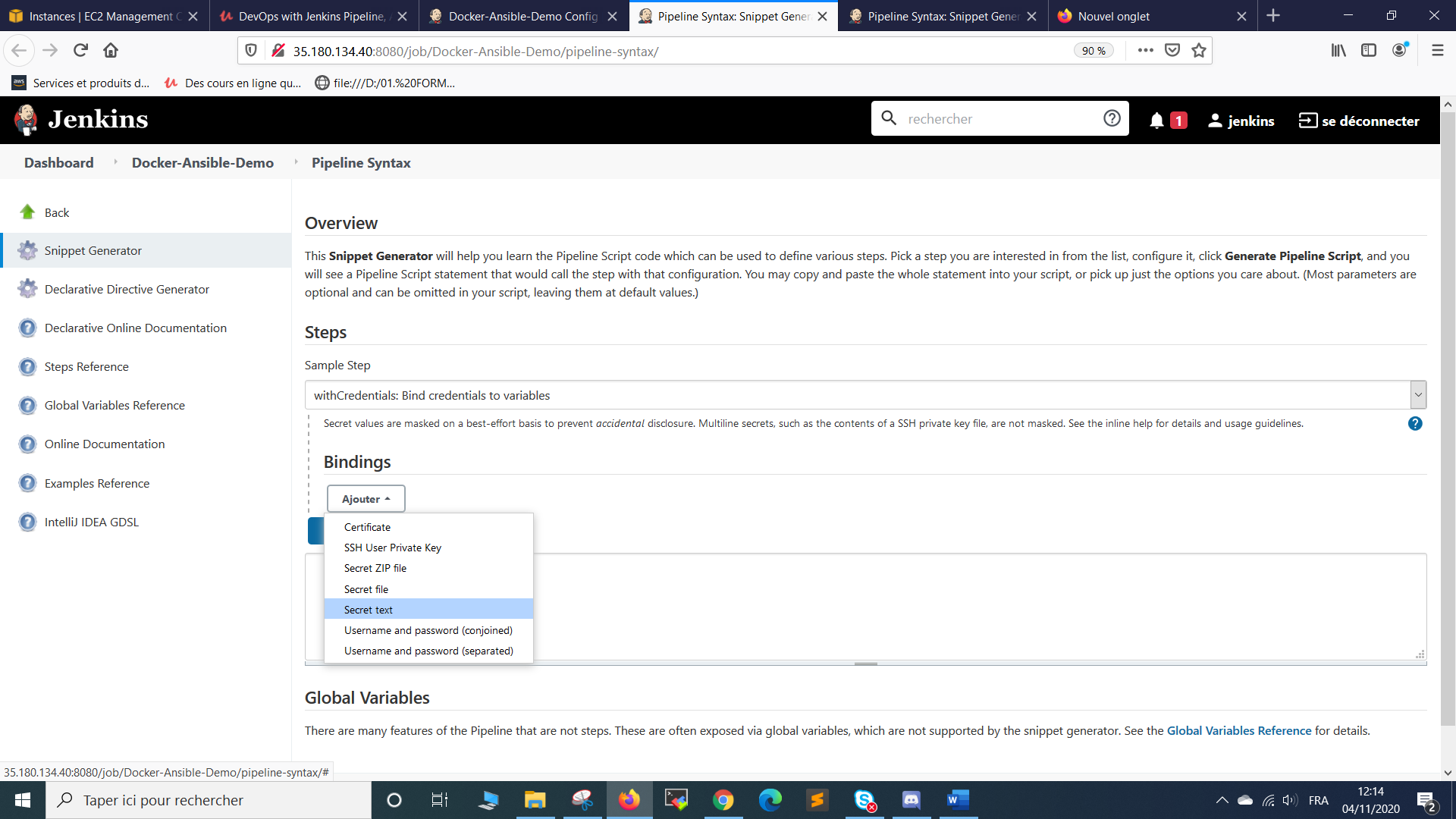


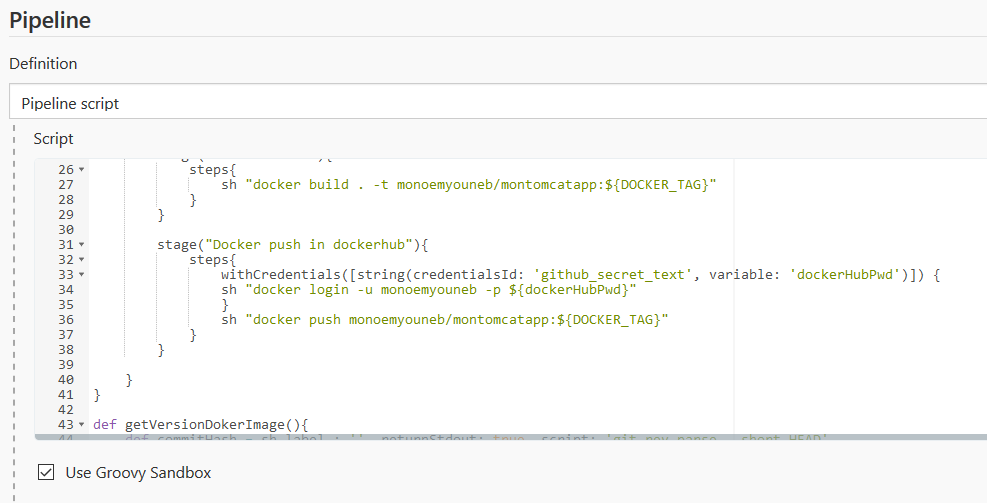
Docker images in VM Jenkins



Push in DockerHub

Stock secret pwd dockerhub in a credentials





pipeline{

agent any

tools{

maven 'maven3'

}

environment{

DOCKER\_TAG = getVersionDokerImage()

}

stages{

stage("SCM"){

steps{

git credentialsId: 'github', url: 'https://github.com/javahometech/dockeransiblejenkins'

}

}

stage("Maven build"){

steps{

sh "mvn clean package"

}

}

stage("Docker build"){

steps{

sh "docker build . -t monoemyouneb/montomcatapp:${DOCKER\_TAG}"

}

}

stage("Docker push in dockerhub"){

steps{

withCredentials([string(credentialsId: ' dockerhub\_secret\_text', variable: 'dockerHubPwd')]) {

sh "docker login -u monoemyouneb -p ${dockerHubPwd}"

}

sh "docker push monoemyouneb/montomcatapp:${DOCKER\_TAG}"

}

}

}

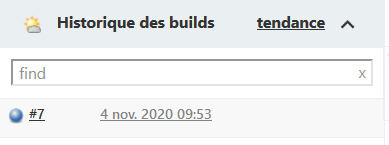
}

def getVersionDokerImage(){

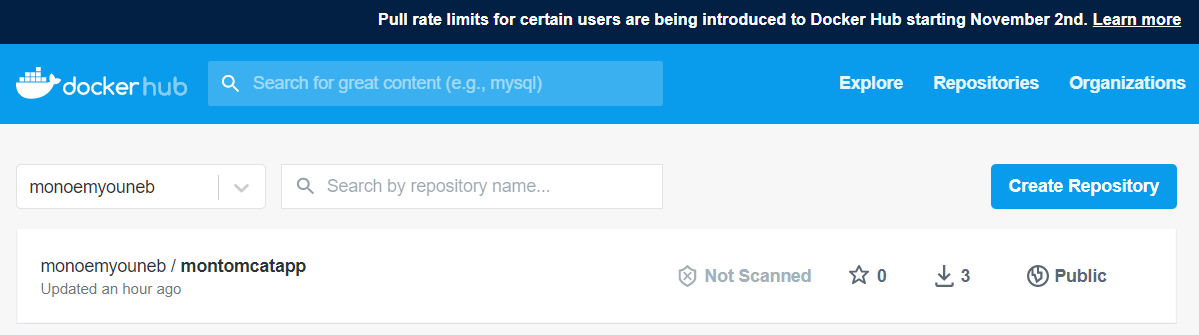
def commitHash = sh label : '', returnStdout: true, script: 'git rev-parse --short HEAD'

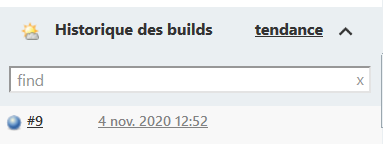
return commitHash

}



<https://hub.docker.com/>



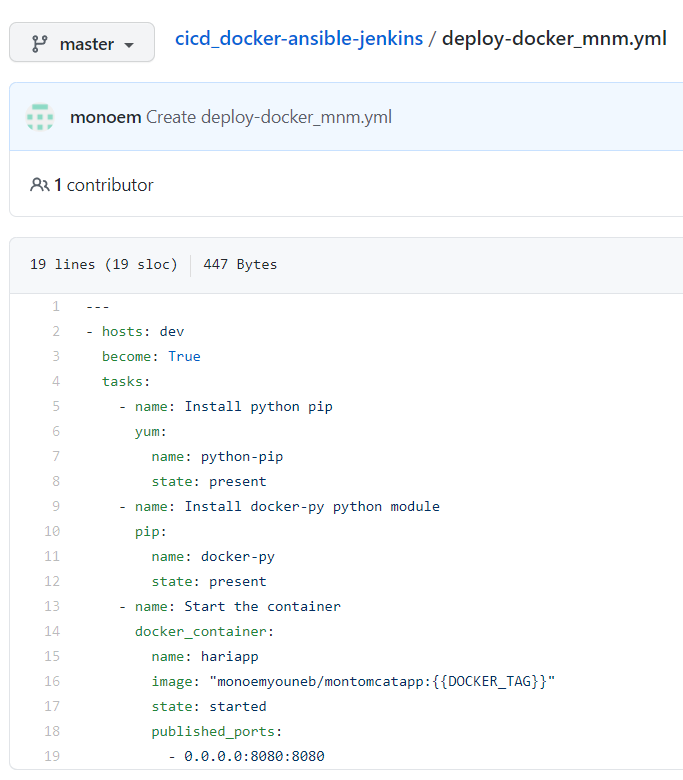


1. **Write a playbook to deploy war file into docker\_server \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***



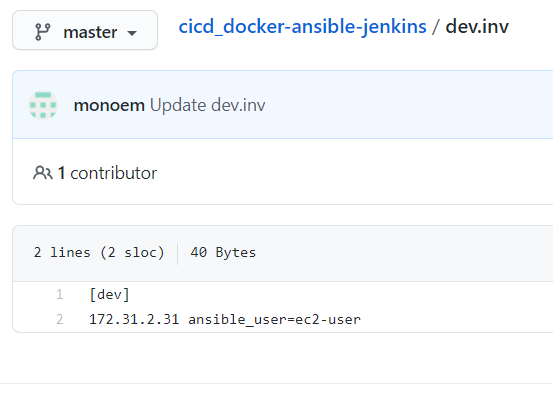
<https://github.com/monoem/cicd_docker-ansible-jenkins>

Create deploy-docker\_mnm.yml



|  |
| --- |
| -- |
|  | - hosts: dev |
|  | become: True |
|  | tasks: |
|  | - name: Install python pip |
|  | yum: |
|  | name: python-pip |
|  | state: present |
|  | - name: Install docker-py python module |
|  | pip: |
|  | name: docker-py |
|  | state: present |
|  | - name: Start the container |
|  | docker\_container: |
|  | name: hariapp |
|  | image: "monoemyouneb/montomcatapp:{{DOCKER\_TAG}}" |
|  | state: started |
|  | published\_ports: |
|  | - 0.0.0.0:8080:8080 |

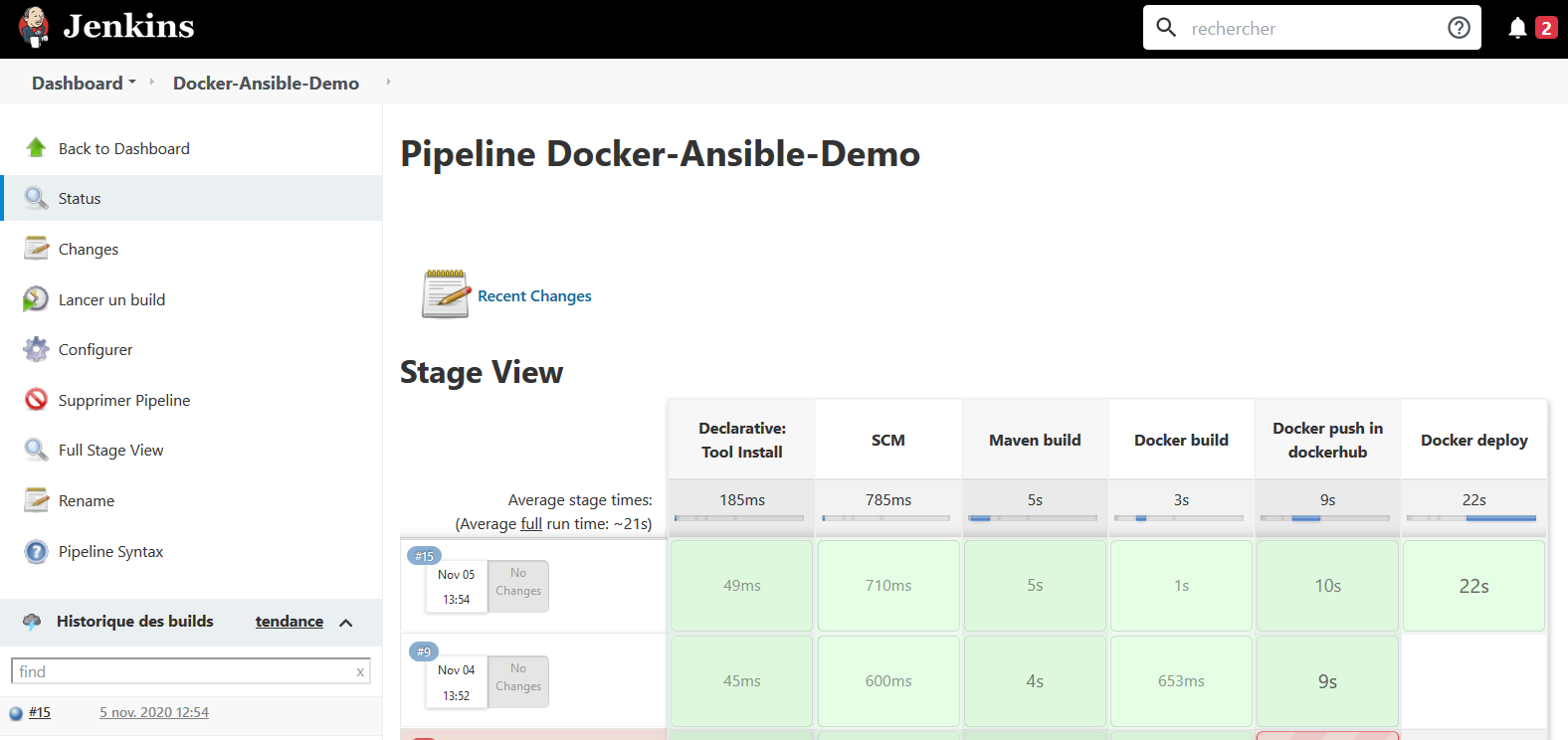
Enventry file



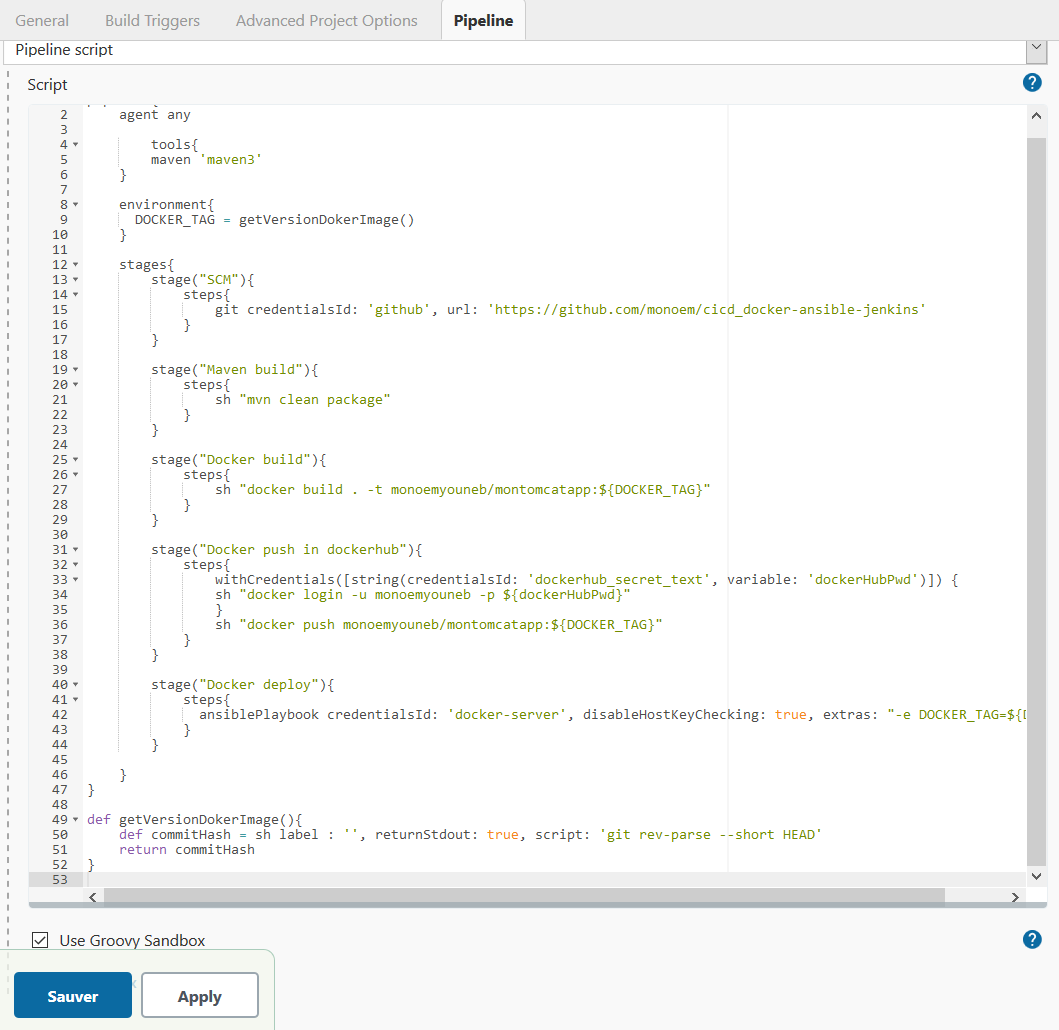
Use ansible plugin

**Job :** [Docker-Ansible-Demo](http://35.180.134.40:8080/job/Docker-Ansible-Demo/)

**Syntaxe**: 







pipeline{

agent any

tools{

maven 'maven3'

}

environment{

DOCKER\_TAG = getVersionDokerImage()

}

stages{

stage("SCM"){

steps{

git credentialsId: 'github', url: 'https://github.com/monoem/cicd\_docker-ansible-jenkins'

}

}

stage("Maven build"){

steps{

sh "mvn clean package"

}

}

stage("Docker build"){

steps{

sh "docker build . -t monoemyouneb/montomcatapp:${DOCKER\_TAG}"

}

}

stage("Docker push in dockerhub"){

steps{

withCredentials([string(credentialsId: 'dockerhub\_secret\_text', variable: 'dockerHubPwd')]) {

sh "docker login -u monoemyouneb -p ${dockerHubPwd}"

}

sh "docker push monoemyouneb/montomcatapp:${DOCKER\_TAG}"

}

}

stage("Docker deploy"){

steps{

ansiblePlaybook credentialsId: 'docker-server', disableHostKeyChecking: true, extras: "-e DOCKER\_TAG=${DOCKER\_TAG}", installation: 'ansible-server', inventory: 'dev.inv', playbook: 'deploy-docker\_mnm.yml'

}

}

}

}

def getVersionDokerImage(){

def commitHash = sh label : '', returnStdout: true, script: 'git rev-parse --short HEAD'

return commitHash

}

<http://15.237.36.161:8080/dockeransible/>

