Git Url:

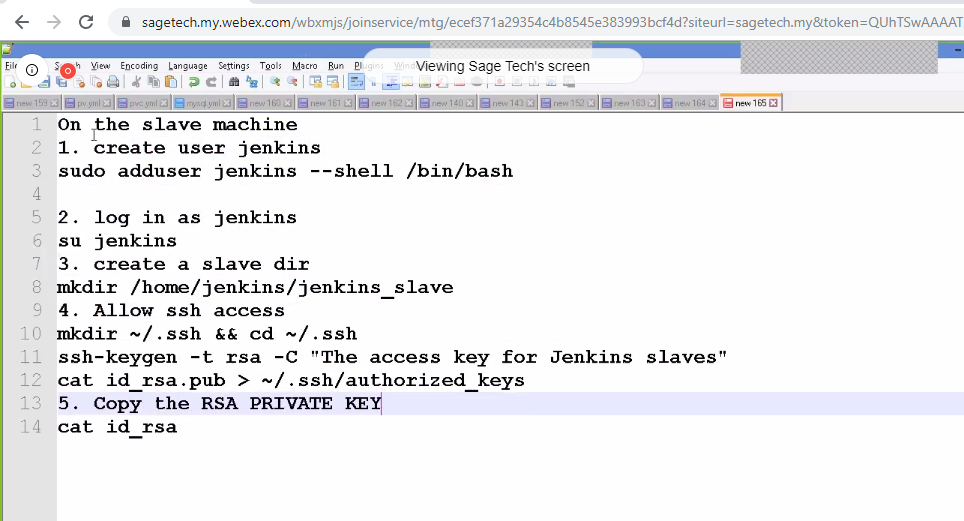
<https://github.com/AdityaSP/itc-aug26-jenkins>

**Docs.ansible.com/**

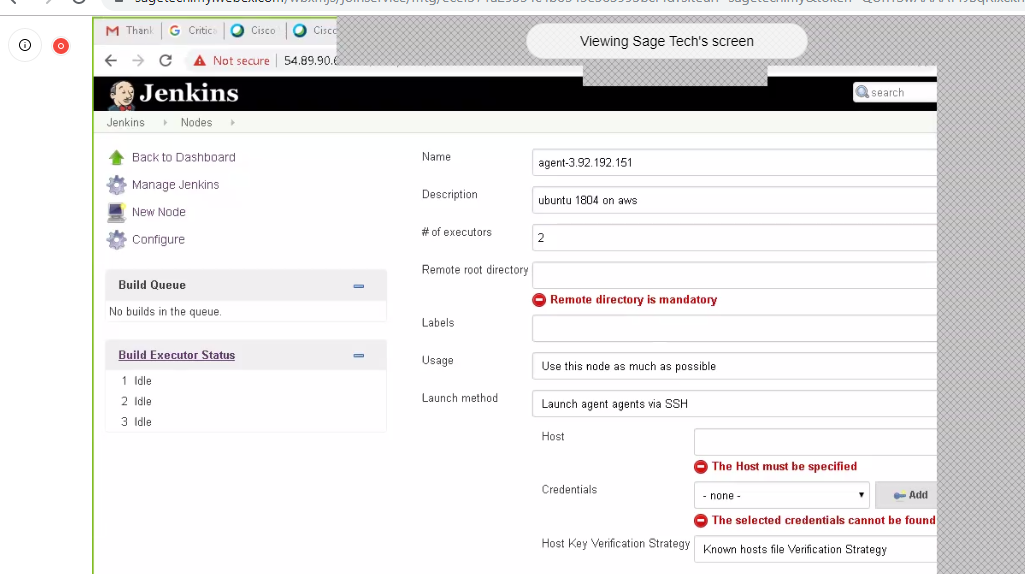
**Git location for ansible docs:**

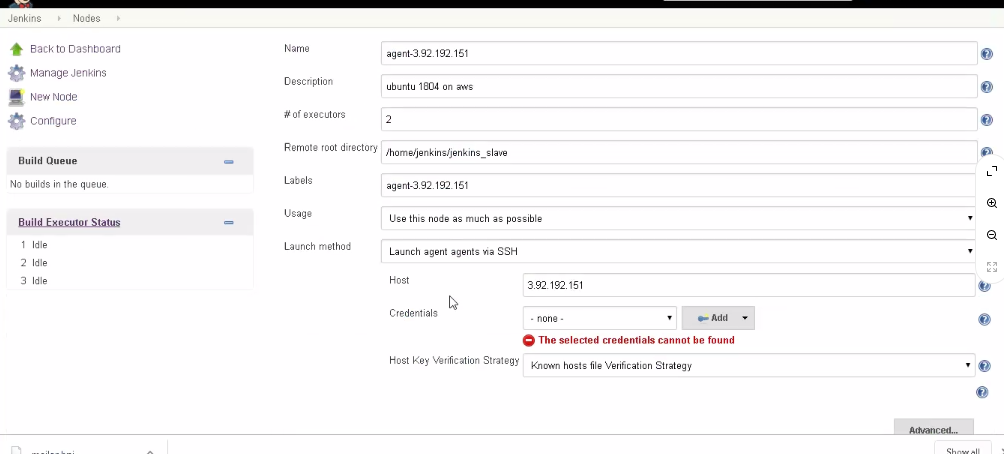
<https://github.com/AdityaSP/itc-aug26-jenkins>



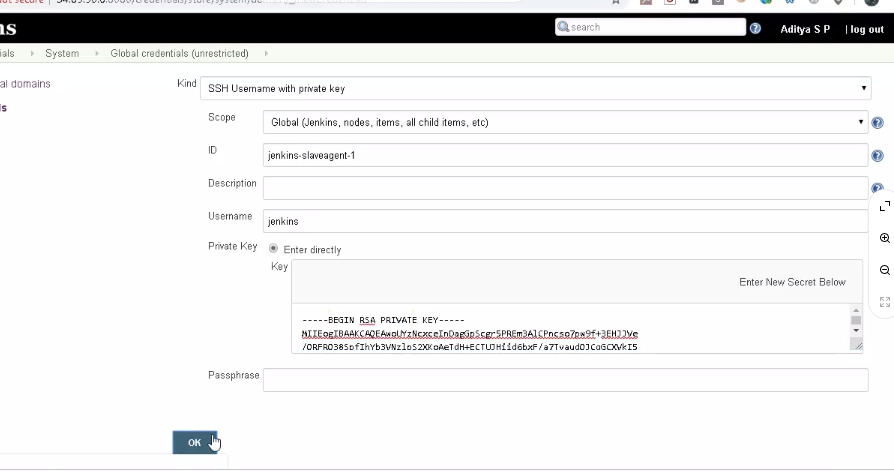


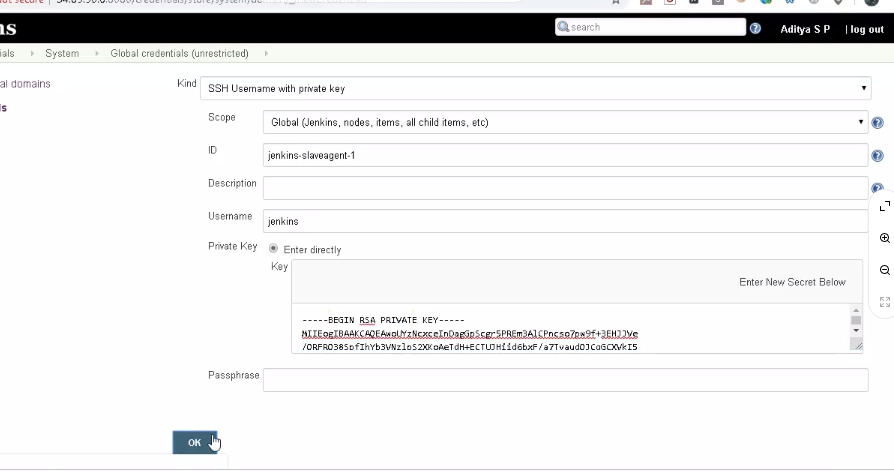
Manage Jenkins🡪Manage Nodes🡪Create a new node





Create credentials: or add private key:

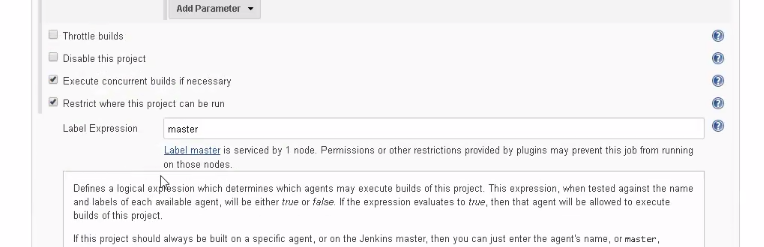




On freestyle project:

Run only master itself (restrict)

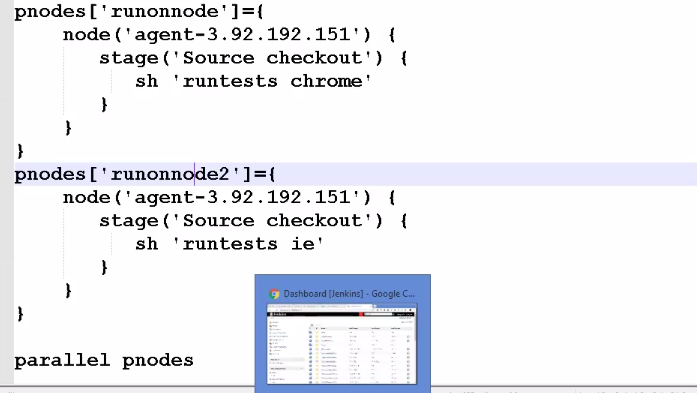
Run concurrent build if only required.



Run on parallel in single job:



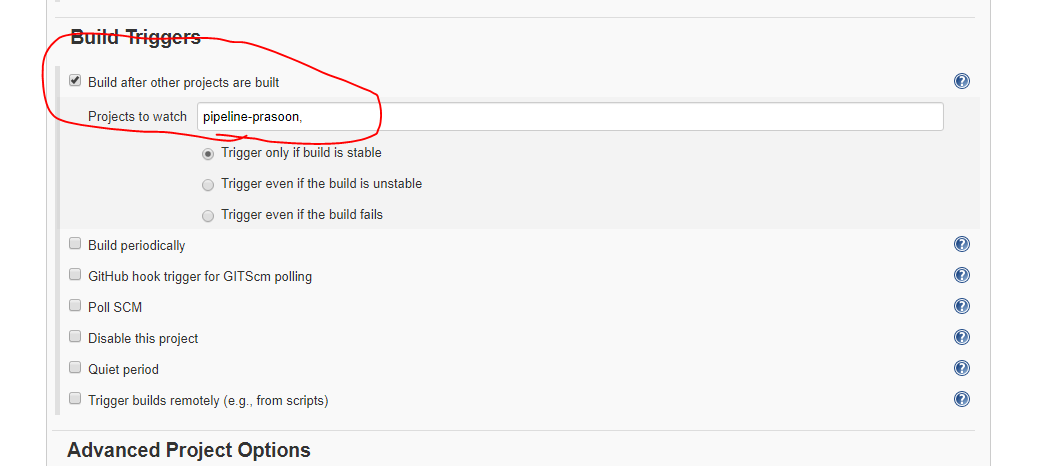
If you want to run different env.:



Dependent build:

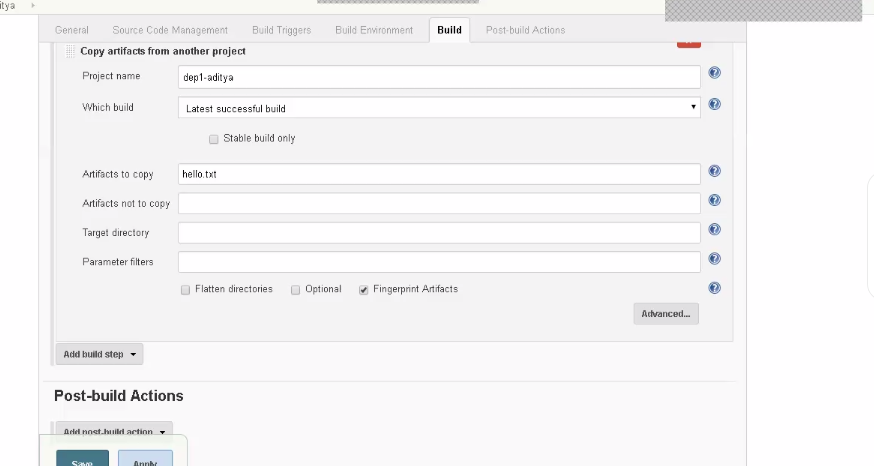
Build Trigger After

Build will trigger if pipeline-prasoon is SUCCESS



Copy artifact plugin>

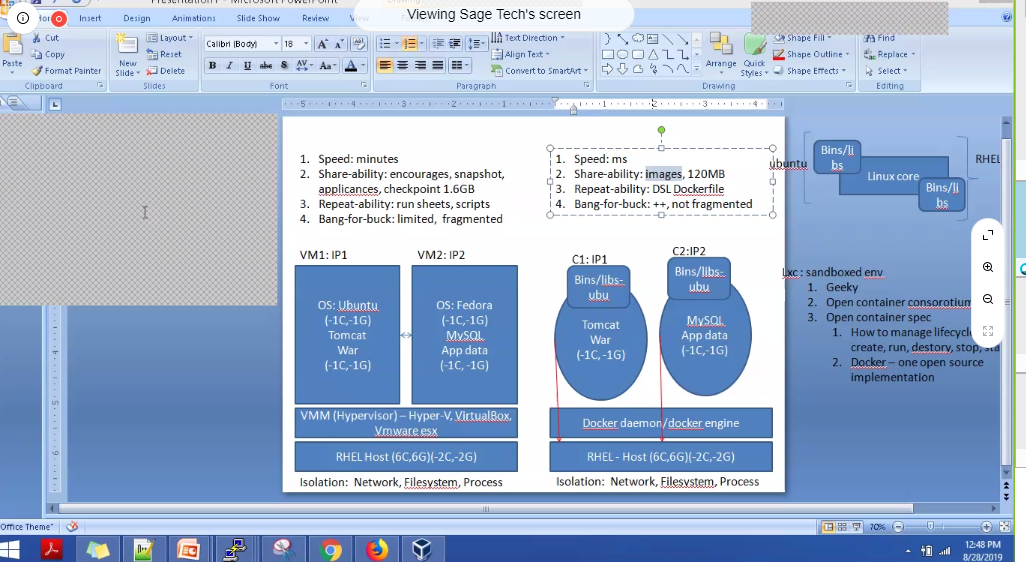
Copy artifact from any successful build and move on any tarhet location, git, nexus, folder



[maven-devops-demo1-1.0.1-SNAPSHOT.jar](http://54.89.90.6:8080/job/Prasoon_job_1/lastSuccessfulBuild/artifact/target/maven-devops-demo1-1.0.1-SNAPSHOT.jar)

docker installed

Docker:



Docker cmd.

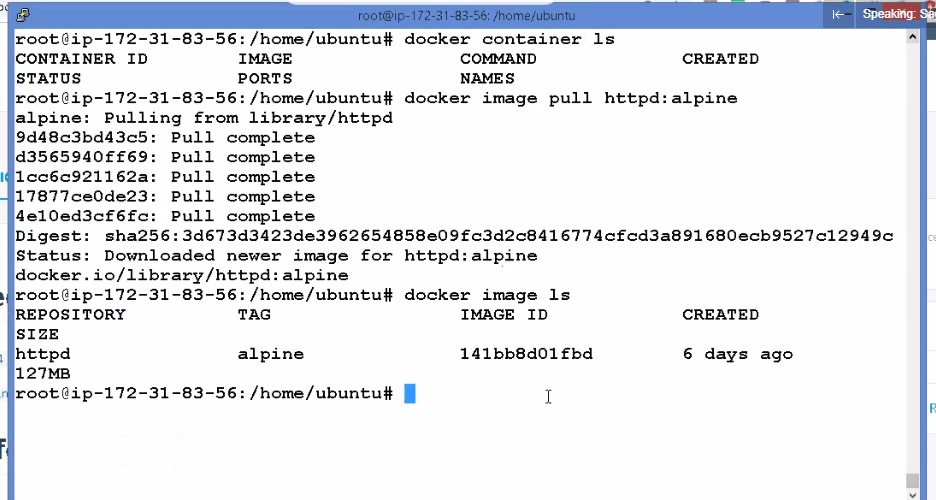
Hub.docker.com is the place you can find images of any s/w

Light weight VM=container

Httpd=apache in redhat

List your docker continer:



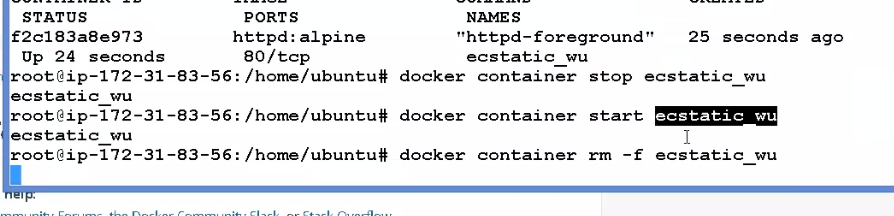


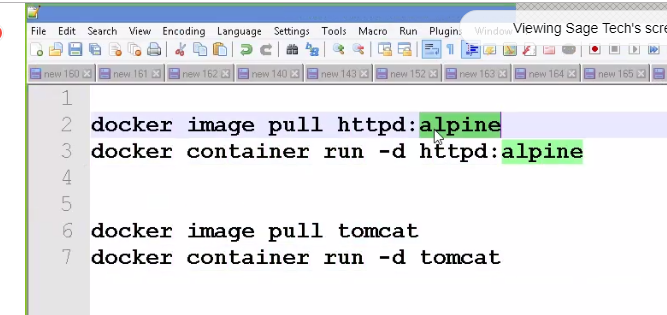
Container = process

Images are binary

Running the container:



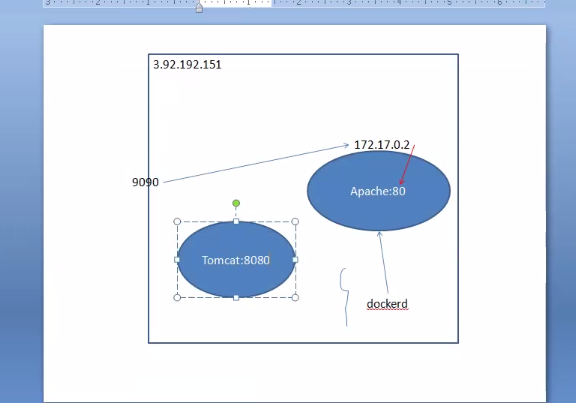


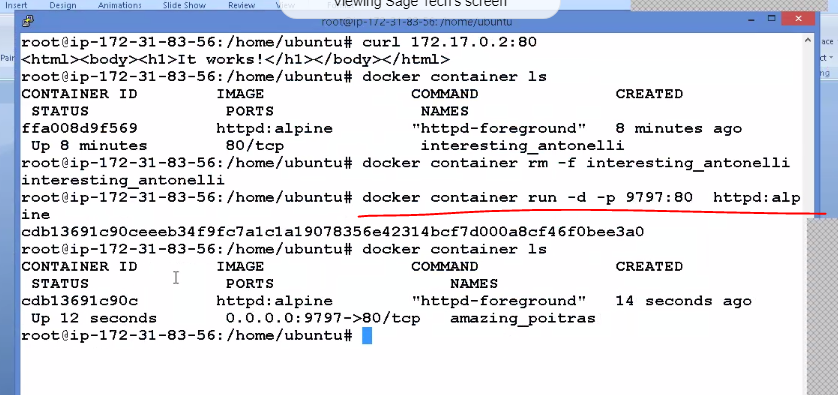


Pacakage your whole product/software and provide the img in docker

Docker continer networking:

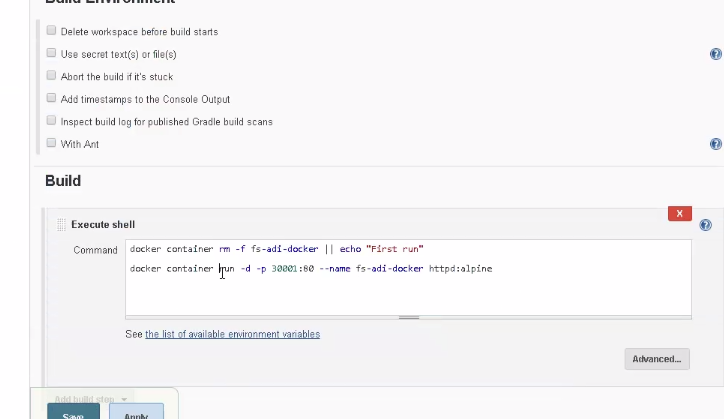
3.92.192.151



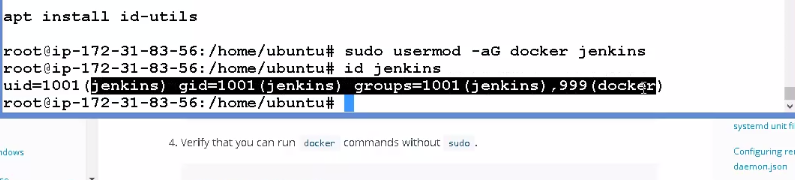


Jenkins itself if we can get as images continer

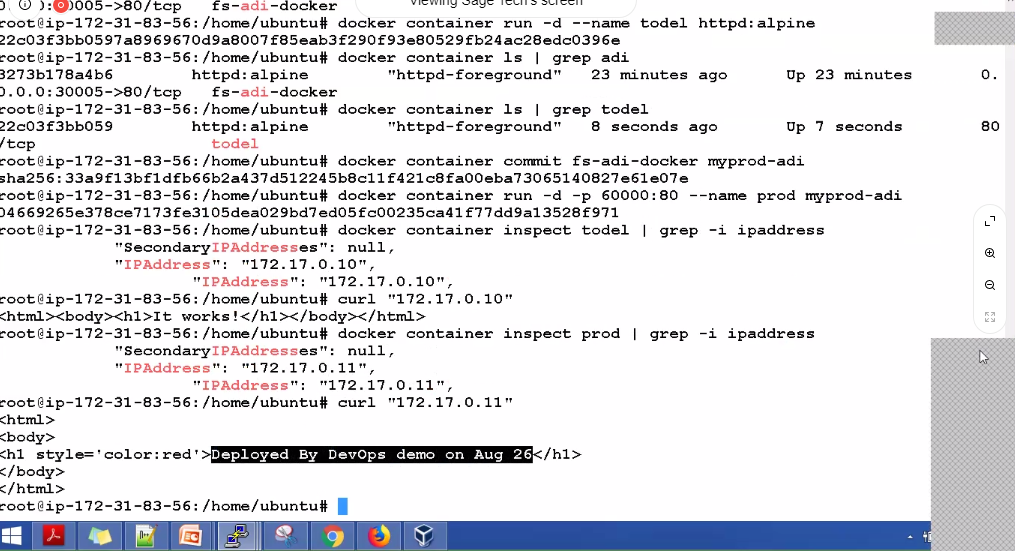
Docker install on Jenkins node



Adding Jenkins user in docker group:

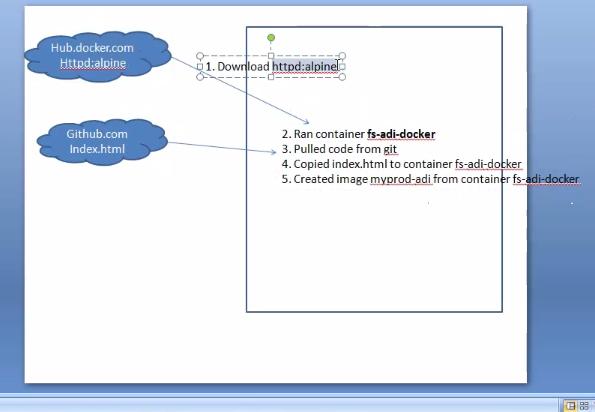


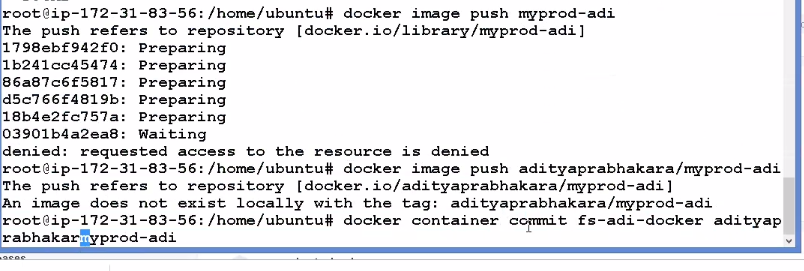
Creating contener images:

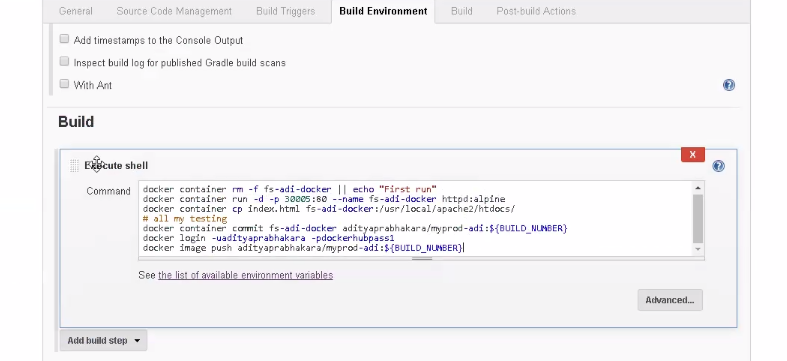


Docker container commit fs-adi-docker myprod-adi

Docker container ispect prod | grep -I ipaddress







docker container rm -f fs-adi-docker || echo "First run"

docker container run -d -p 30005:80 --name fs-adi-docker httpd:alpine

docker container cp index.html fs-adi-docker:/usr/local/apache2/htdocs/

# all my testing

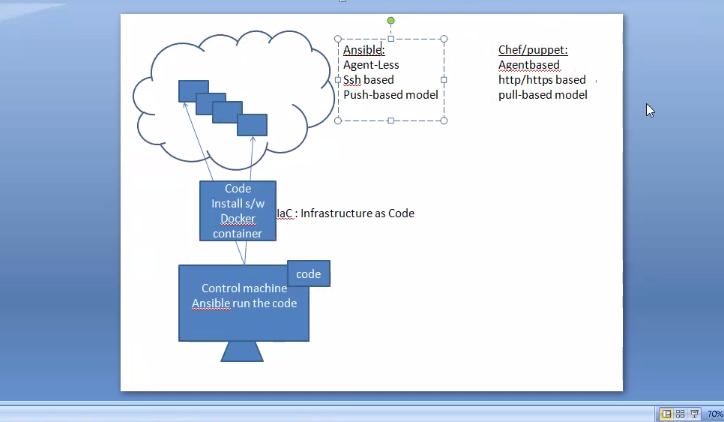
docker container commit fs-adi-docker adityaprabhakara/myprod-adi:${BUILD\_NUMBER}

docker login -uadityaprabhakara -pdockerhubpass1

docker image push adityaprabhakara/myprod-adi:${BUILD\_NUMBER}

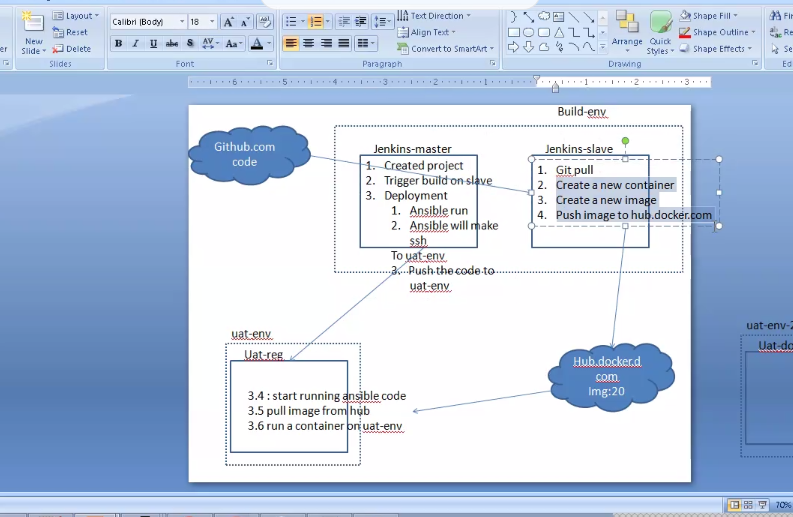
docker push prasoon123/prasoongumasta:tagname

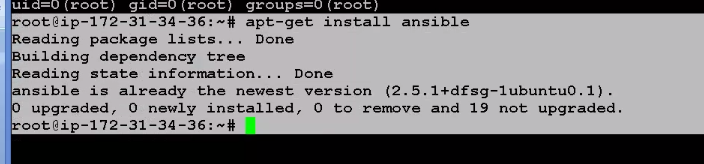
Ansible: also called as agent less system



Installed Ansible on Master machine

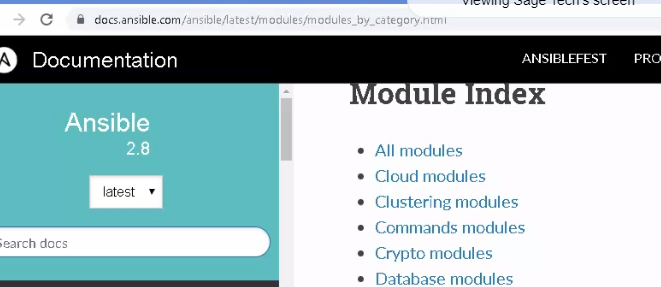
Installed python on agent/slaves machines

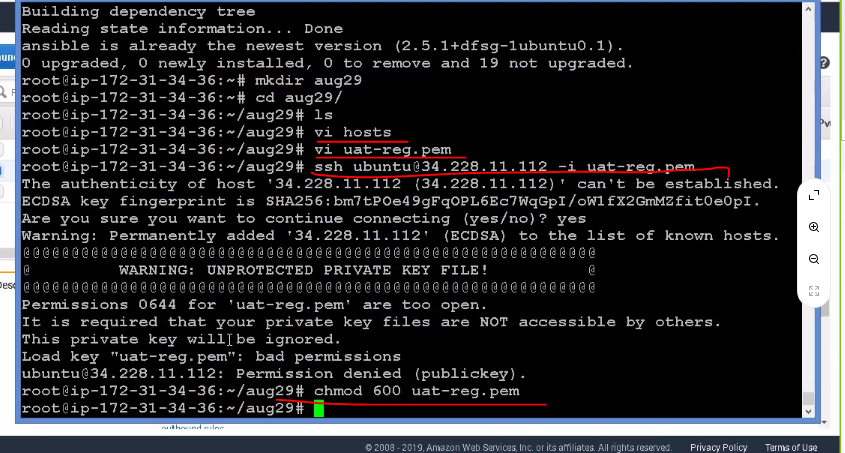




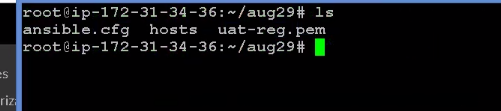
Github link for reference:

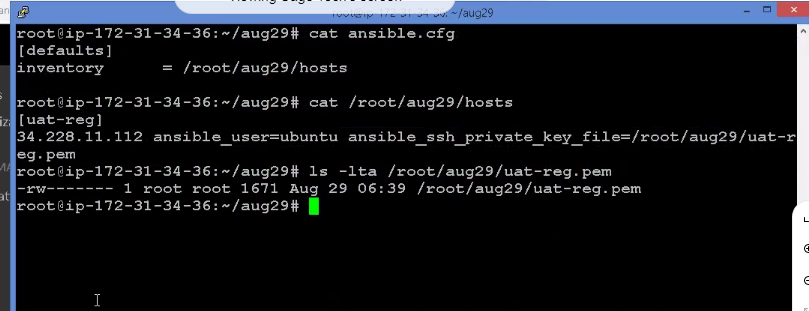
https://github.com/AdityaSP/itc-aug26-jenkins







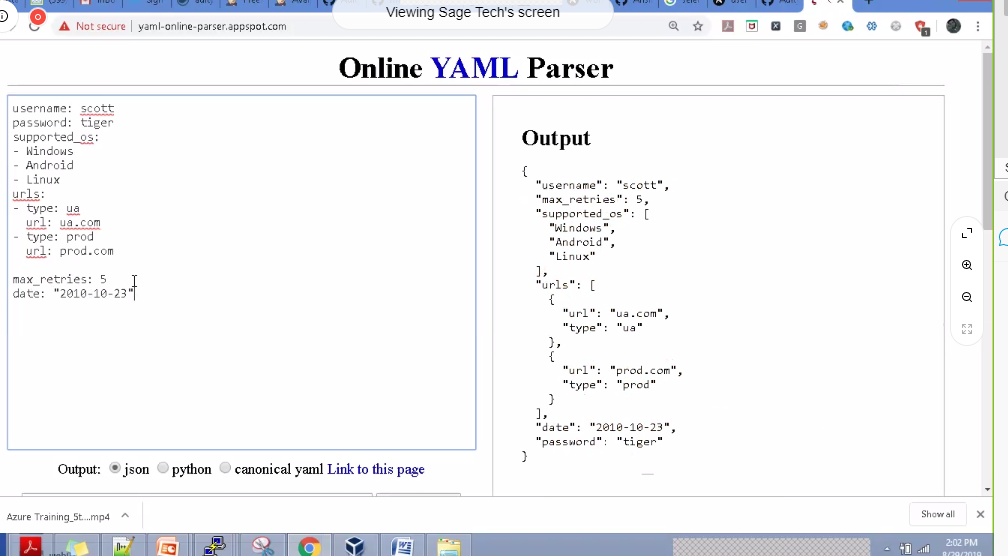


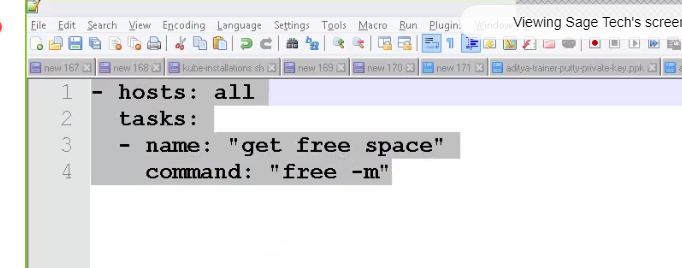


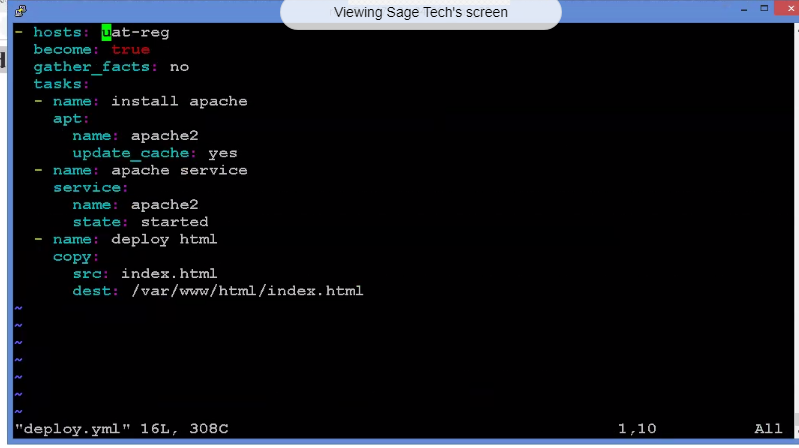
Inventory file/host file: list of all machines which we need to control

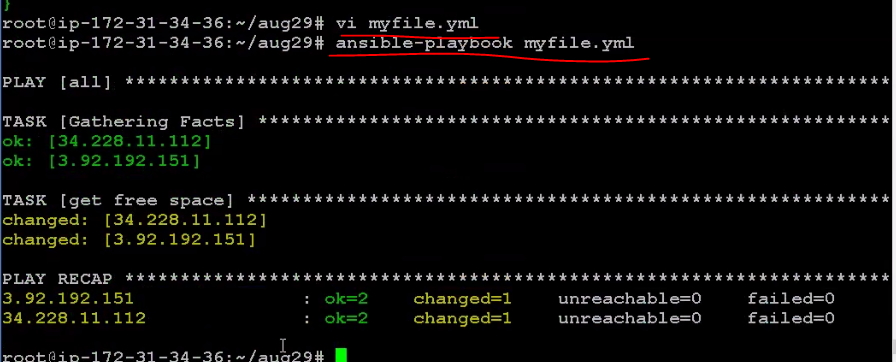
Conf file: the path of inventory file(where its actually located)

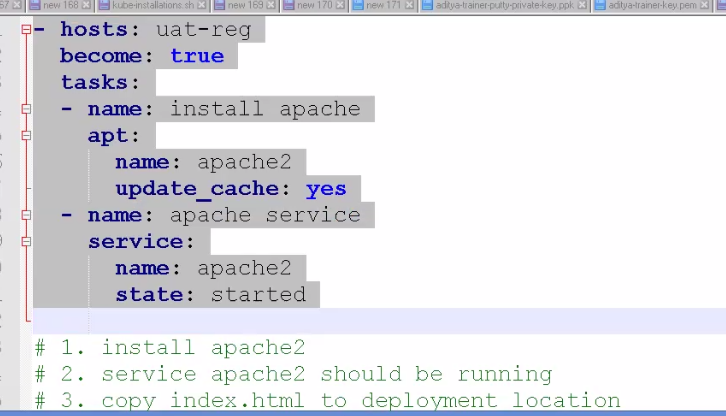
<https://yaml-online-parser.appspot.com/>



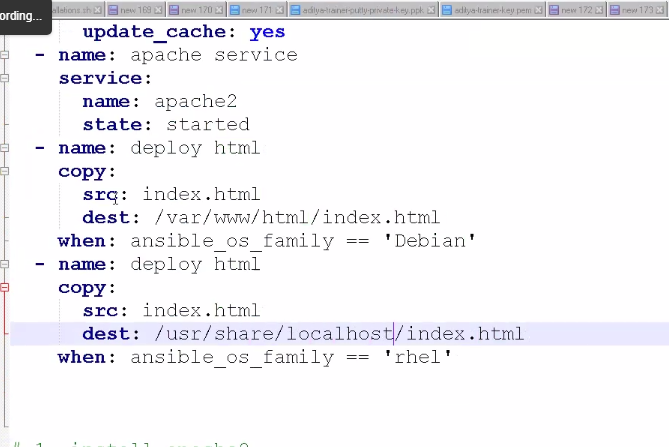


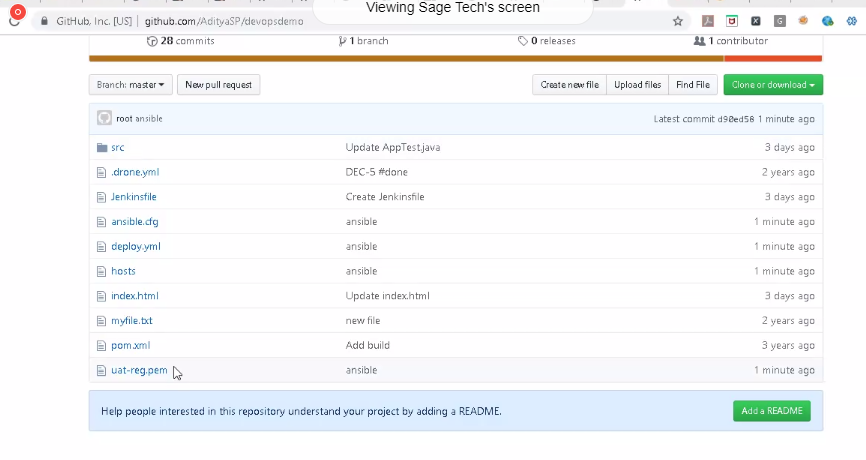






Conditional run on the basis of OS

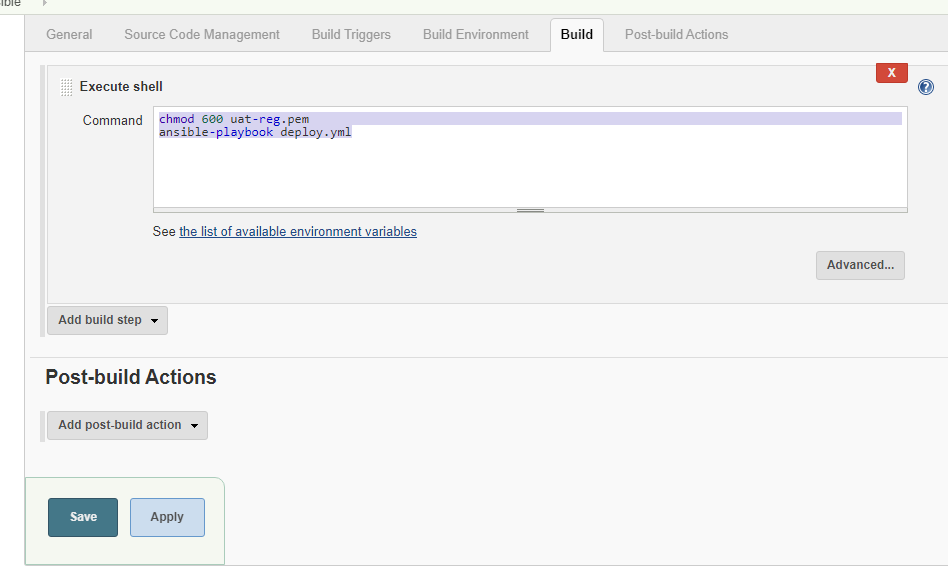




<https://github.com/AdityaSP/devopsdemo/>

chmod 600 uat-reg.pem

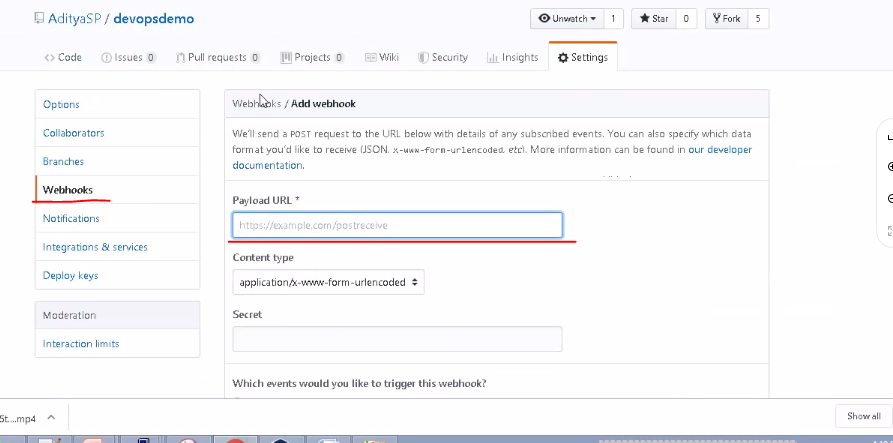
ansible-playbook deploy.yml





**Note:**

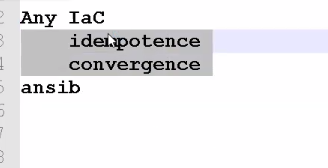
**If we want to directly run the jenkin build for each commit in github then we need to use webhook and set the Jenkins job url.**



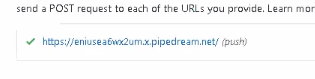
**And Upload URL :your Jenkins job**

**Idempodent**

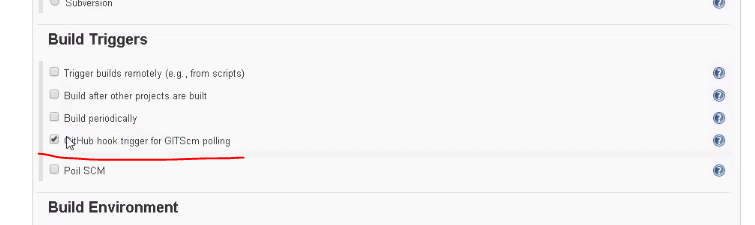
**And convergence**

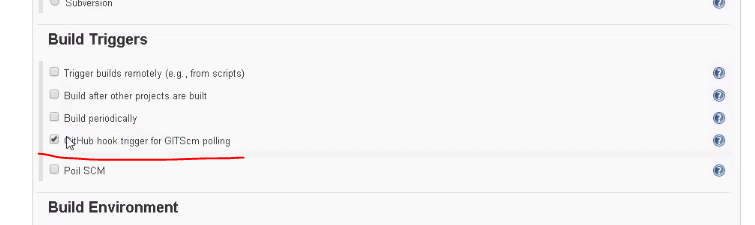


**Add Web HOOK:**



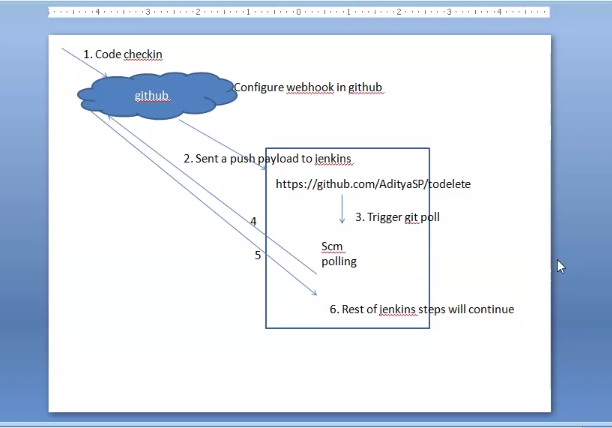
**Every time commit will happen in github, build will trigger**

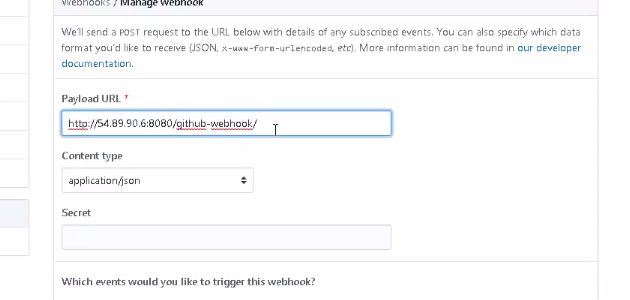




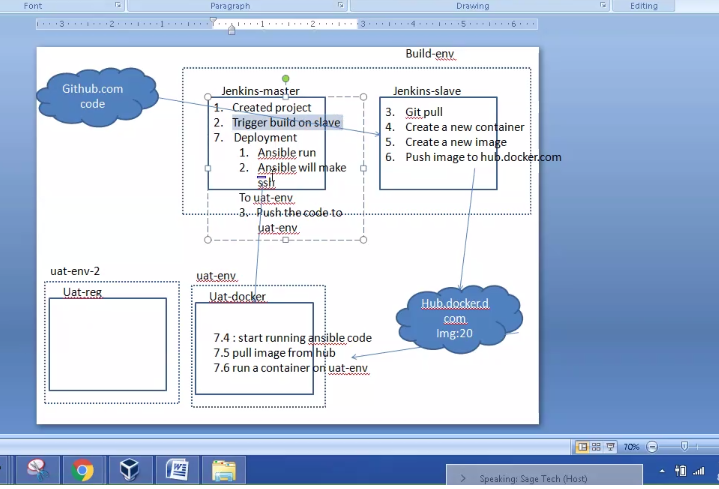
**SCM polling:**

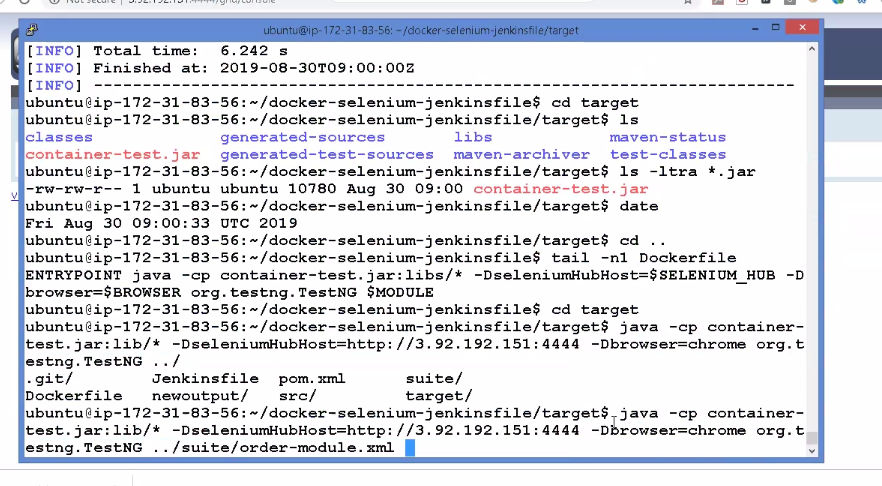
**This will trigger after 15 min if any change in git**





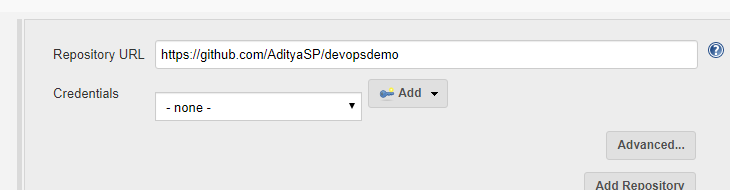
**In addition: Github server plugin (need to enter github url and credentials) will automatically create a webhook in github**





**Jenkins job:**

**Prasoon\_docker:**



docker container rm -f fs-prasoon-docker || echo "First run11"

docker container run -d -p 50001:80 --name fs-prasoon-docker httpd:alpine

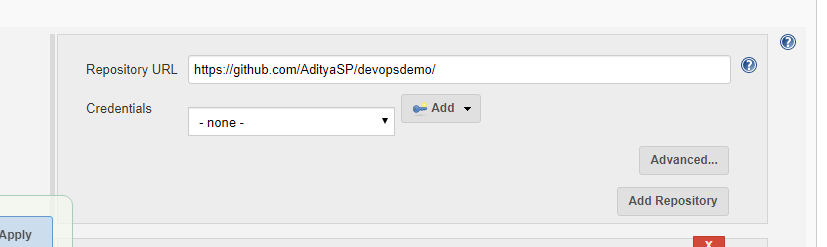
docker container cp index.html fs-prasoon-docker:/usr/local/apache2/htdocs/

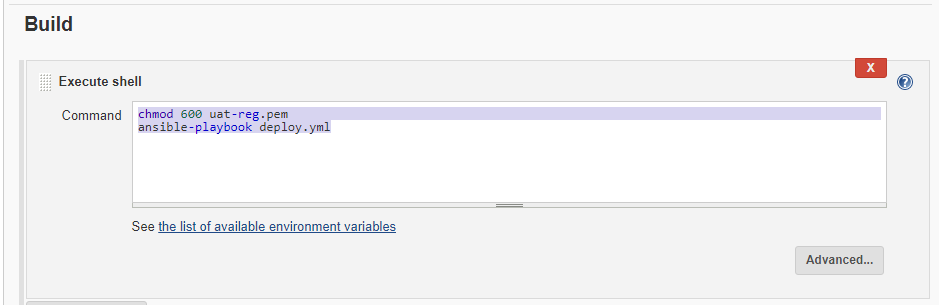
docker container commit fs-prasoon-docker prasoon123/myprod-prasoon:${BUILD\_NUMBER}

docker login -uprasoon123 -pJUly@2019

docker image push prasoon123/myprod-prasoon:${BUILD\_NUMBER}

**Prasoon\_ansible:**

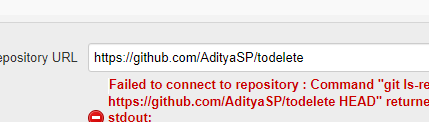


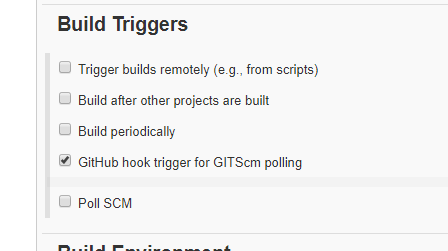


chmod 600 uat-reg.pem

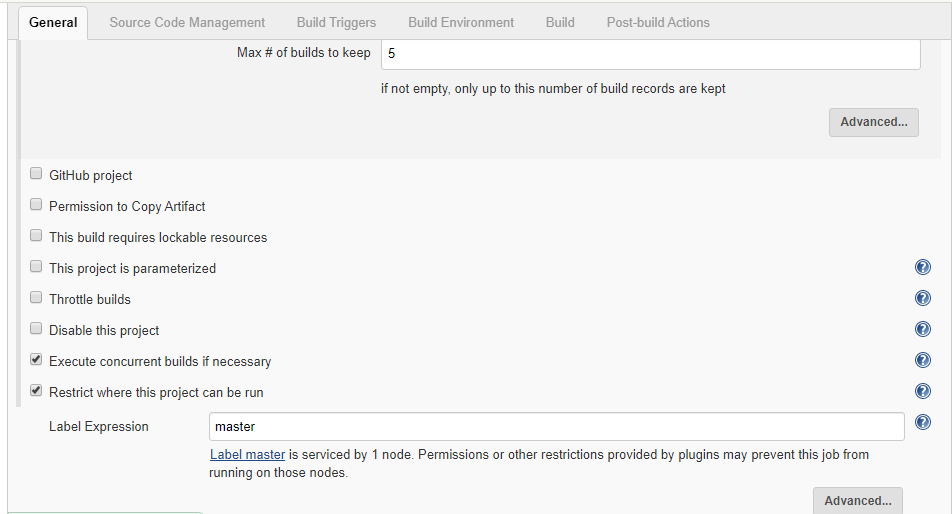
ansible-playbook deploy.yml

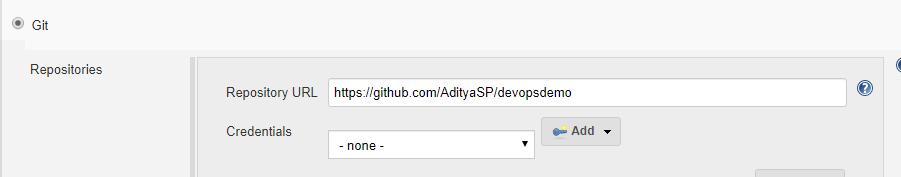
**Prasoon\_webhook**

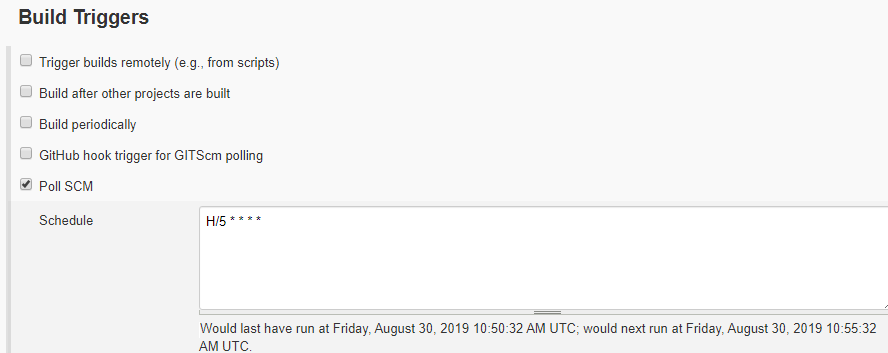


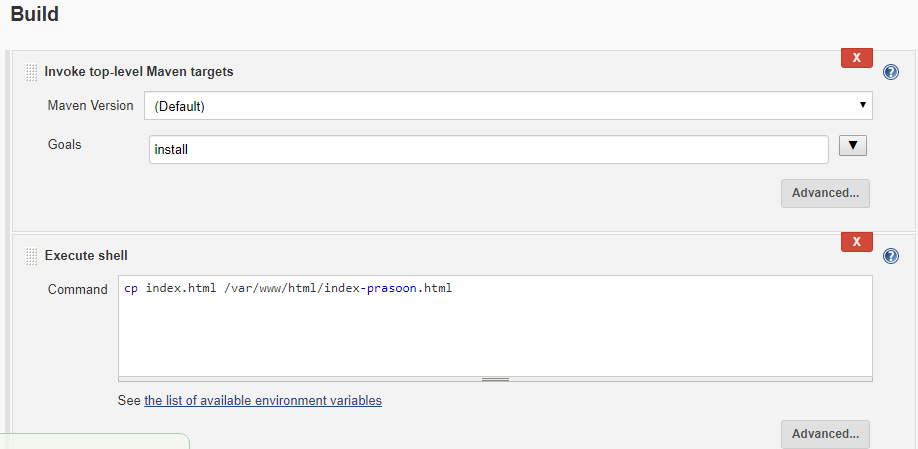


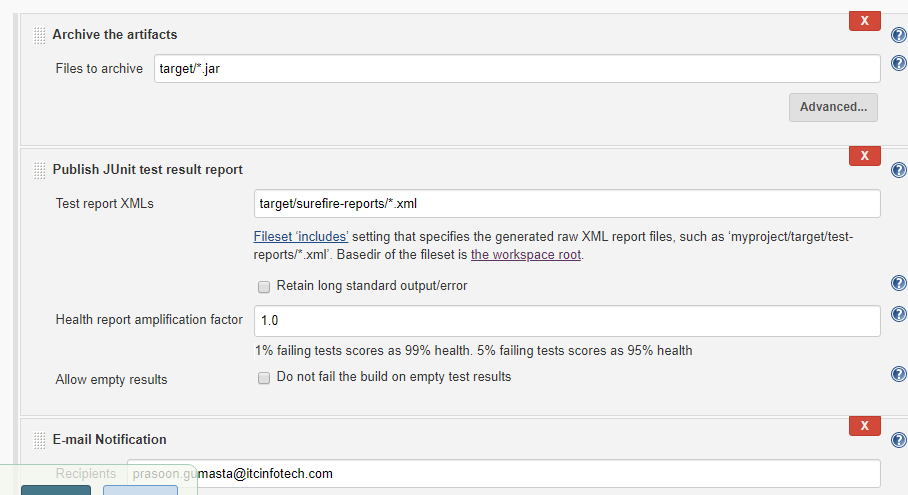
**Prasoon\_job1:**

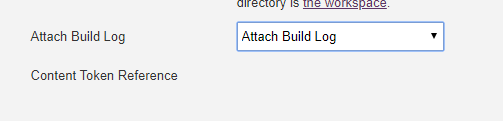




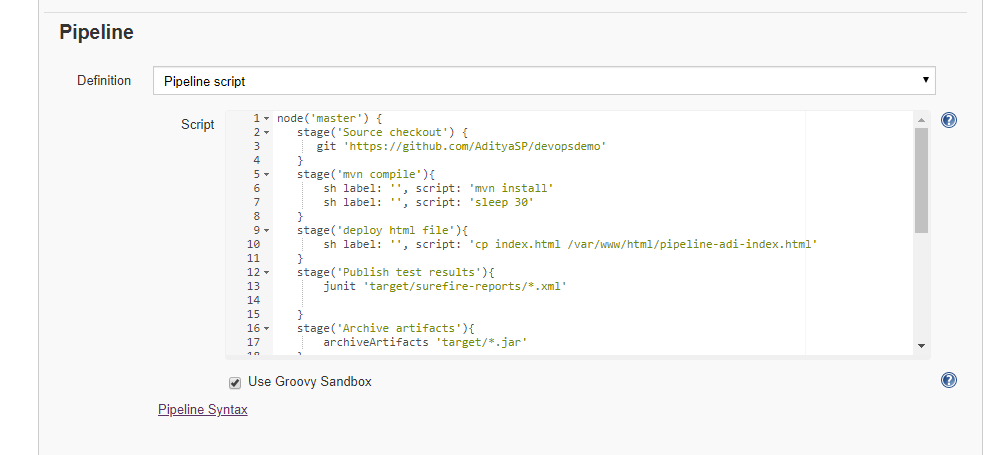








**Pipeline\_prasoon:**



node('master') {

stage('Source checkout') {

git 'https://github.com/AdityaSP/devopsdemo'

}

stage('mvn compile'){

sh label: '', script: 'mvn install'

sh label: '', script: 'sleep 30'

}

stage('deploy html file'){

sh label: '', script: 'cp index.html /var/www/html/pipeline-adi-index.html'

}

stage('Publish test results'){

junit 'target/surefire-reports/\*.xml'

}

stage('Archive artifacts'){

archiveArtifacts 'target/\*.jar'

}

}

node('agent-3.92.192.151') {

stage('Source checkout') {

git 'https://github.com/AdityaSP/MyAppDemo'

}

stage('mvn compile'){

sh label: '', script: 'mvn install'

sh label: '', script: 'sleep 30'

}

stage('Publish test results'){

junit 'target/surefire-reports/\*.xml'

}

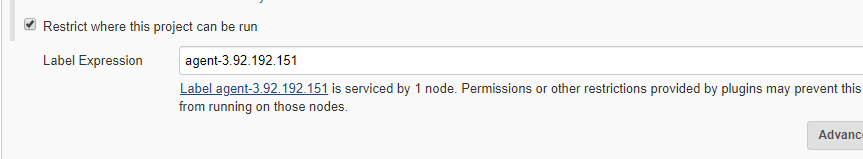
stage('Archive artifacts'){

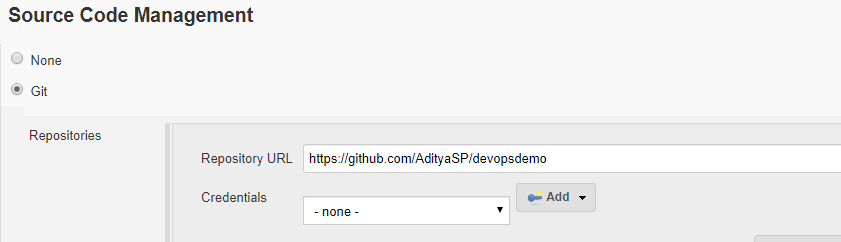
archiveArtifacts 'target/\*.war'

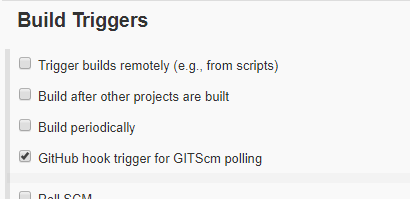
}

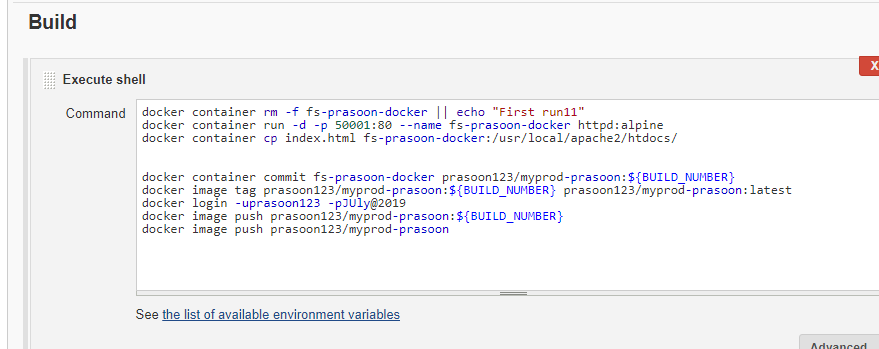
}

**Full-ci-cd\_prasoon:**









docker container rm -f fs-prasoon-docker || echo "First run11"

docker container run -d -p 50001:80 --name fs-prasoon-docker httpd:alpine

docker container cp index.html fs-prasoon-docker:/usr/local/apache2/htdocs/

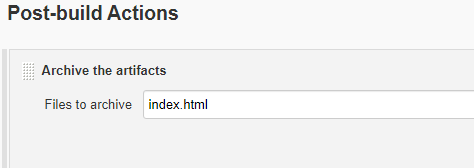
docker container commit fs-prasoon-docker prasoon123/myprod-prasoon:${BUILD\_NUMBER}

docker image tag prasoon123/myprod-prasoon:${BUILD\_NUMBER} prasoon123/myprod-prasoon:latest

docker login -uprasoon123 -pJUly@2019

docker image push prasoon123/myprod-prasoon:${BUILD\_NUMBER}

docker image push prasoon123/myprod-prasoon



**pp-prasoon-pipeline:**





def pnodes= [:]

pnodes['runOnMaster']=

{

node('master') {

stage('Source checkout') {

git 'https://github.com/AdityaSP/devopsdemo'

}

stage('copy artifacts'){

copyArtifacts filter: 'target/maven-devops-demo1-1.0.1-SNAPSHOT.jar', fingerprintArtifacts: true, projectName: 'Prasoon\_job\_1', selector: lastSuccessful()

}

stage('Archive artifacts'){

archiveArtifacts 'target/\*.jar'

}

}

}

pnodes['runOnAgent']=

{

node('agent-3.92.192.151') {

stage('Source checkout') {

git 'https://github.com/AdityaSP/MyAppDemo'

}

stage('mvn compile'){

sh label: '', script: 'mvn install'

sh label: '', script: 'sleep 30'

}

stage('Publish test results'){

junit 'target/surefire-reports/\*.xml'

}

stage('Archive artifacts'){

archiveArtifacts 'target/\*.war'

}

}

}

parallel pnodes

node('master'){

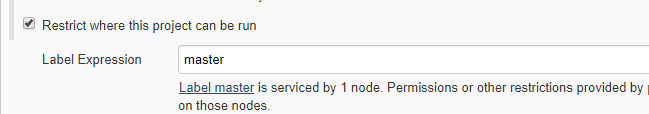
stage('send notification'){

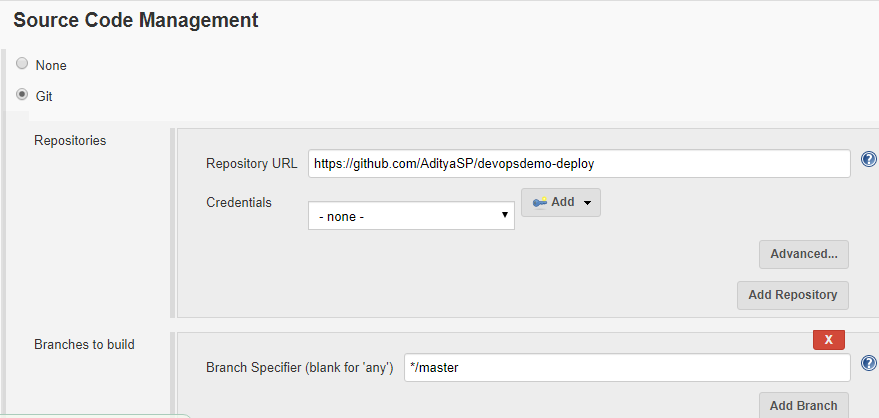
emailext body: '$PROJECT\_NAME - Build # $BUILD\_NUMBER - $BUILD\_STATUS',

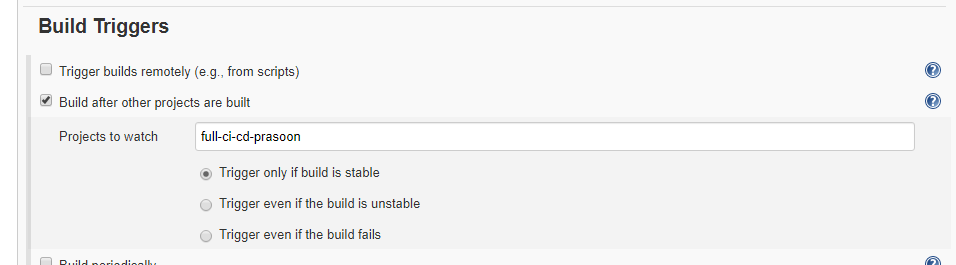
subject: '$PROJECT\_NAME - Build # $BUILD\_NUMBER - $BUILD\_STATUS', to: 'prasoon.gumasta@itcinfotech.com' }

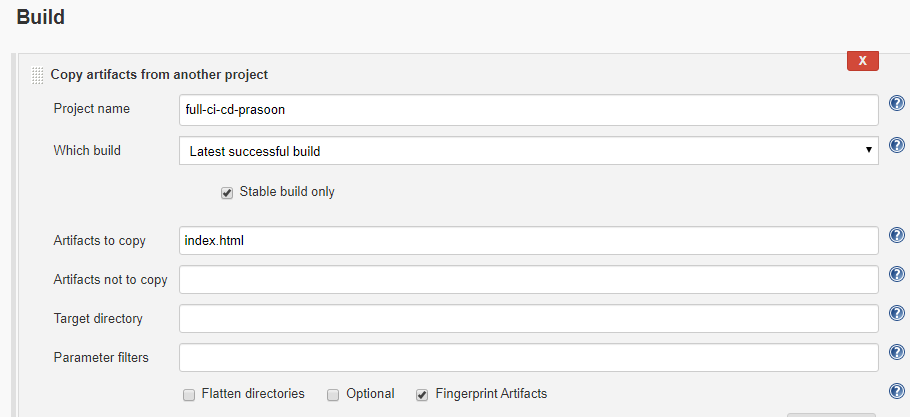
}

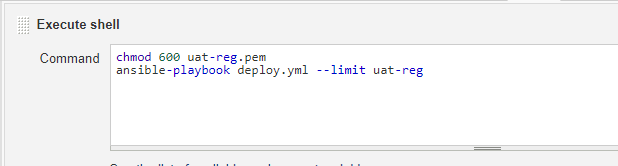
**full-ci-cd-pipeline-prasoon-deploy**











**full-ci-cd-pipeline-prasoon-docker-deploy:**

