

~~16/09/2021~~ → Evening

16/09/2021

Jenkins

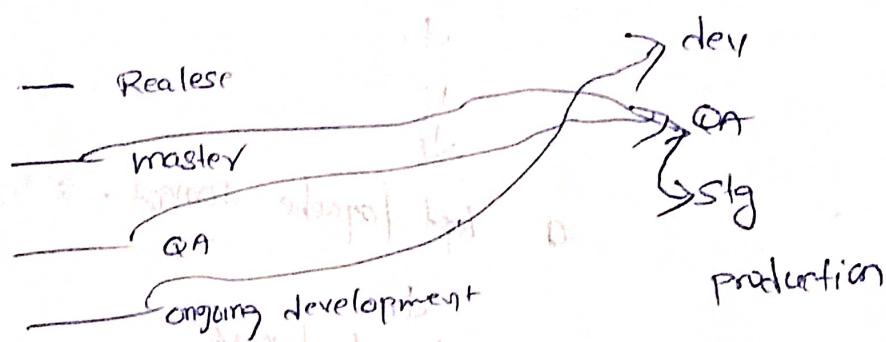
Jenkins

CI / CD

→ providing plugin support

CI - continuous Integration

- Agile methodology



Jenkins - is CI/CD platform where it will integrate all other third party softwares & tools to enable CI/CD process & Automation.

Jenkins Installation

→ Redhat → Rpm based dist

→ Ubuntu → Debian

→ Generic → on behalf tomcat

root + 0 chomop
Jenkins

Down Jenkins

J1

Jenkins.io | down

↓
copy link address of

Generic java package (war)

J1

cd /opt/apache-tomcat-8.5.3/webapps

J1

wget url

J1

ls

↓

cd /opt/apache-tomcat-8.5.3/bin

J1
sh startup.sh

J1

ipaddress: 8080/jenkins

↓

unlock Jenkins

To ensure

/root/.jenkins/secrets/initialAdminPassword

Administrator password

J1

cd root

J1
ls -la

copy link address of
/root/.jenkins/secrets/initialAdminPassword

↓
cat /root/.jenkins/secrets/initialAdminPassword

↓
copy fff94 (→ after this step
copy)

paste in administrator password in jenkins

Jobs :

Empty Job

Maven Jobs

free style jobs

declarative pipeline jobs

scripted pipeline jobs

multi Branch Pipeline jobs

~~Job~~ ~~Job~~ - ~~declarative~~

~~Job~~ - ~~scripted~~

mostly used

declarative

→ write code customize

free style

↓
Install suggested plugins

↓

you can get

Getting started

↓

Then

↓

Playing with Jenkins Create first admin user

User Name

devops

password

devops

confirm password

devops

Full name

~~dev~~ dev

E-mail Address

vijaybooc@gmail.com

↓

Save & Continue

↓

Instance configuration

Jenkins url:

http://1118.223.24.167:8080/jenkins

↓

Save and Finish

↓

Jenkins

↓

click on New Item

↓

Enter an item name

Your job

Freestyle project

↓

OK

↓

Save

↓

click on new item

↓

Item name:

myjob

↓

OK

↓

Save

↓

Jenkins dashboard

↓

click on

manage jenkins

↓

manage users

↓

create user

↓

create user

↓

Username:

Password:

confirm password:

Full name:

E-mail Address:

[redacted]

[redacted]

[redacted]

[redacted]

developer

developer

developer

developer

Vijayboopathi@gmail.com

↓

create user

↓

username: tester

password: tester

confirm password : tester

full name : tester

e-mail Address : vijayboob@gmail.com

↓

click on create user

↓

User id	Name
developer	developer
devops	devops
tester	tester

↓
Logout

↓
login with developer

↓
User name : developer

↓
password : developer

↓

Logout

↓
Login with tester

↓

User name : tester

↓
password : tester

↓

Logout

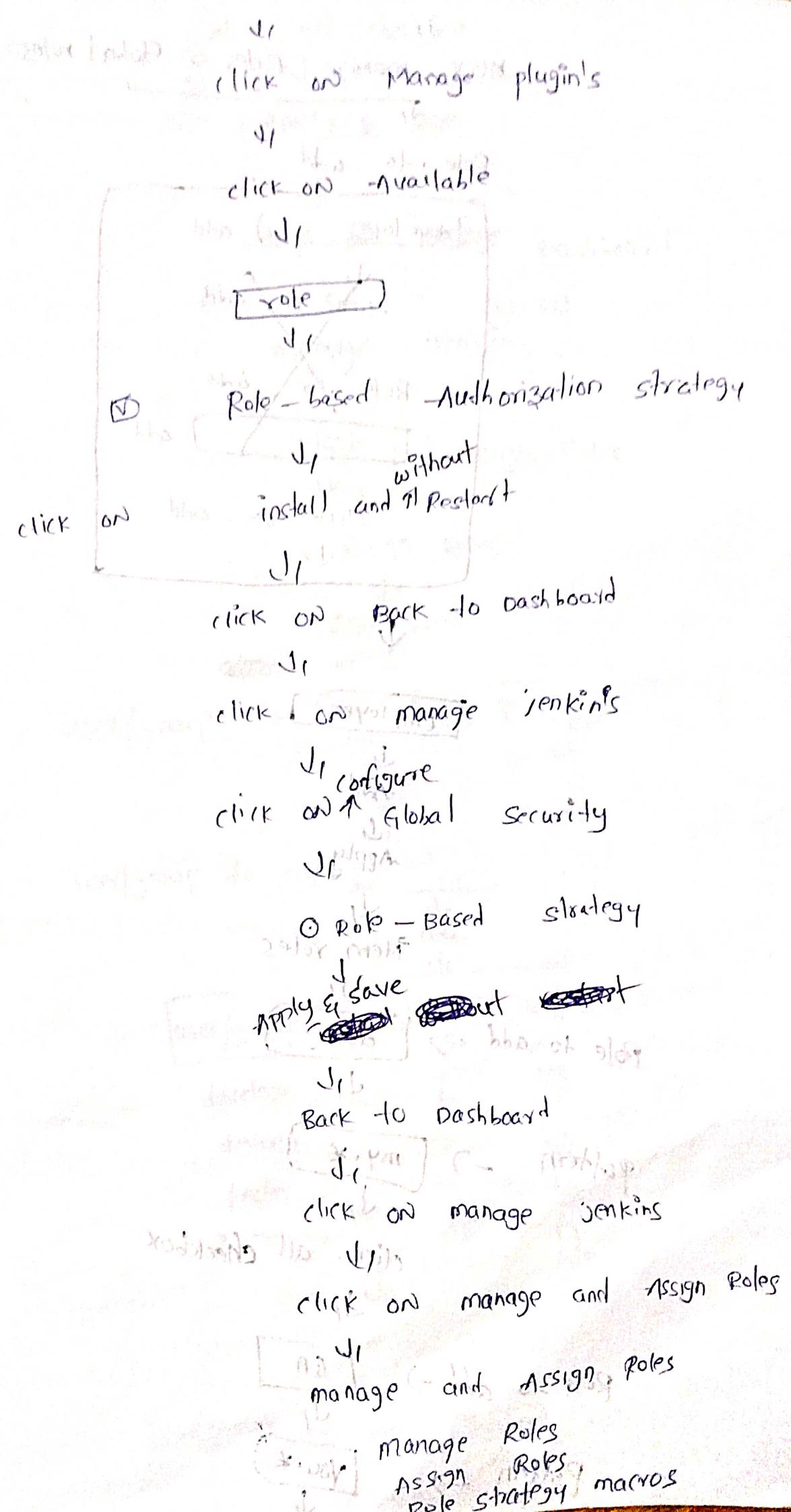
↓
Login with devops

↓
User name : devops

↓
password : devops

↓

click on manage jenkins



click manage Roles → Global roles

Role to add

developer add

click on add

Role add

tester add

click on add

Employee

Add

Apply

Item roles

Role to add → dev

↓

pattern → my.*

click all checkbox

Role to add → QA

↓

Pattern → you.*

↓

click all & check box

Jenkins → click on group → new

APPLY & SAVE

check opn manage dashboard

Jr.

manage

Jenkins,

manage and assign Roles

Jr.

assign Roles

Global roles

global Jr.

user/group → add developer

add

add

add

add

add

user/group → add tester

tester

add

add

user/group	admin	employee
------------	-------	----------

checkboxes: developer

devops

tester

anonymous

↓
item roles

user/group → add developer

add

user/group to add is **tester**

add

jr

board item roles

User/group	dev	qa
developer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
tester	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>

close application

apply & save

cancel

Dashboard

jr

group new

logout

username

login without tester

password: tester

login with username: developer

M

J

password: developer

O

S

N

R

G

B

libra

depository

V

E

A

H

I

group new

10/09/2021

slack

go to slack and click config

-tomstart - tomcat

start -tomcat server

↓ for b3

go to chrome

ip address : 8080 / jenkins

come to terminal

↓
cd /root

↓
ls -la

↓
cd .jenkins

↓

ls

↓
cd users

↓
ls

cat users.xml

↓

cd ..

↓
jobs

cd ~~jobs workspace~~

↓

ls

↓

go to jenkins ui

click on my jobs

↓

Then click on build now

↓

go to terminal

cd ..

cd workspace

ls

↓

cd atnames

↓

cd jobs

↓

ls

↓

cd builds

↓

ls

↓

cd 1

↓

cat log

↓

Go to jenkins ui

↓

click on my job

↓

permalink s

↓

click on console output

↓

check too outputs

open slack app, go to chrome

go to chrome

if

search slack or op

visit https://slack.com

↓

try free

click on

https://slack.com

click on

provide email

continuation

continue

company con team

devops channel

to contact

devops

team working

skip this step

skip this step

click on ok

launching devopschannel

initial channel created

↓

click on channels +

create a channel

initial address

Jenkins

channel name

jenkins

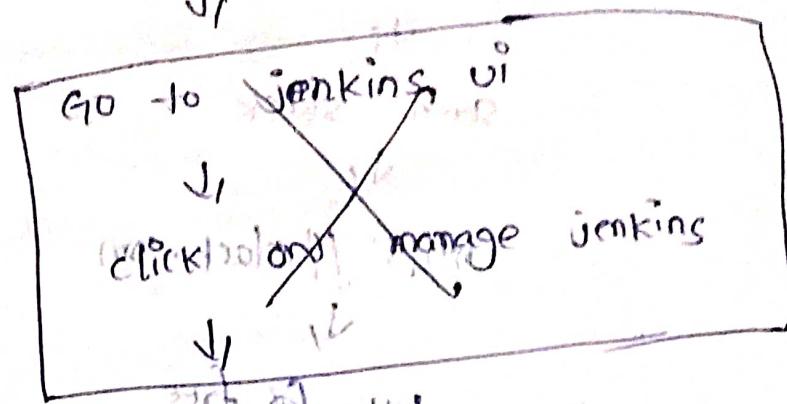
Build status

Description

jenkins

Add people → skip this step

✓ is drop down



click on Devopschannel ✓

From sidebar [] ↓

settings & administration

click on manage apps

Search App Directory

Jenkins CI

Jenkins CI

Jr

Add -> slack

From sidebar [] ↓

choose a channel name: jenkins

slack

↓

Add Jenkins & integration

click on

Manage Slack

↓

Manage Slack

↓

Go to Jenkins UI

↓

Manage Jenkins

↓

Manage Plugins

Available

slack

all slack check box

Install Jenkins

Install without Restart

Jenkins

click on dashboard

Jenkins

go to manage jenkins

copy JIP

click on configure system

to tools

slack

workspace
paste

↓

credential

none

→ Andy

add jenkins

JIP

on slack → copy

copy

team : devops -
subdomain : dfe 9211

credential RP:

OS4K4

Add credentials

kind

secret text

secret

paste

JP

myle

credential

my id

default channel

jenkins

jenkins token

click on test connection

apply & save

database for jenkins

go to slack ui

Ji

click on jenkins

Ji

go to jenkins ui

Ji

click on myjob

Ji

go to slack

Ji

click on jenkins

Ji

check ip address of slack & jenkins

go to jenkins ui

Ji

click & manage jenkins

↓

configure system

↓

add it to Jenkins

↓

Jenkins location

Start 12.10

https://

localhost

port 8080

localhost

Go to stack ui

↓

click on jenkins

↓

ip

↓

check ip address of both

↓

if different IP

change → to AWS instance copy public IP address

Same

No changes

click dashboard

↓

click my job

↓

click on ~~configure~~ config

↓

click on

→ click on post-build actions

J, V

post-build action

order

select slack notifications

J, V

- ✓ Notify Build start
- ✓ Notify success

J, V notify unstable

J, V notify failure

to date of J, V

J, V - Apply & Save

J, V building now

J, V click on Built now

J, V click on Built now

J, V

J, V go to Slack channel

J, V check notification

J, V build status

Free style job:

Start tomcat server

↓

Go to jenkins ui

↓

Select freestyle job (myjob)

↓

Configure

↓

Source code management

↓

Click on git

↓

Repository url : https://github.com/RavitejaAttipudi
/varnsi.git

↓

Add

↓

click on build

Environment
~~targets~~

↓

click on build step

↓

Select execute shell

↓

command:

```
mvn clean install
```

↓

mvn clean install

↓

Apply & Save

↓

click Build now

J,

now go to putty terminal

↓

cd /root

↓

ls -la

↓

cd .jenkins/

↓

cd workspace

↓

ls

↓

cd myjob

↓

ls

↓

tar -zcvf package

↓

cd maveweappdemo/

↓

go to jenkins ui

↓

my job

(freestyle job)

↓

configure

↓

Build Environment

↓

command :

[]

mvn -f maveweappdemo/pom.xml install

↓

apply & save

Build Now

Go to putty terminal

cd jenkins

cd /workspace/myjob/mavewebappdemo

ls

cd

to jenkins ui

myjob

(free style job)

configure

Build triggers

click on Add Build step

Execute shell

select

command

[]

cp -R mavewebappdemo/target

/* /opt/apache-format-8.5.3

/webapps

APPLY & SAVE

↓

Build now

↓

to chrome

GO

| https://shiva896/mavenwebappdemo

IP address: 8080 | mavenwebappdemo

Automatic triggering :-

GO → Jenkins UI

(Job option mode)

↓

select ~~new job~~ your job

(your job)

free style job

↓

click on configuration

↓

click on Source code management

↓

GO → GitHub

↓

copy -api handlers URL

↓

GO → jenkins

↓

paste in

↓

Repository URL

[]

https://github.com/shiva896/shiva.git

click on add

Apply, Save

click on build

click on Build

now go to GitHub

click on settings

click on webhooks

click on Add webhook

payload url

[redacted]

https://ip address : 8080/jenkins/

github-webhook/

content type:

[redacted]

application / js

Add webhook

go to jenkins ui

↓

select

your job

4) $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

Configure

118 1960 1000

click on Build - triggers

J. J. G. H. M. S. 1888

Github hook triggered for Git SCM Polling

and the β at $\beta_1 \approx 3$ is shown.

apply & save

1075

Go do good things

三

-Add a new -file

1978-1980

then commi -

٦

Geologic History

1-6

Select your job

5

check up the bookings

Automatic time zone in Jenkins :-

→ do build automatically

Build triggers



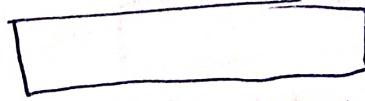
configuration file

Build periodically



Schedule

schedule part



*/2 * * * * →

* * * * * →

minutes h D m D (month) (week)

parameterized job :-

parameter \rightarrow variable which can store the value.

select new item

parameter job



-Free style job



ok



Source code management



url : <https://github.com/sagheerrahman/prince.git>



Add



click on Hubot



This project is parameterized



Add parameter \checkmark

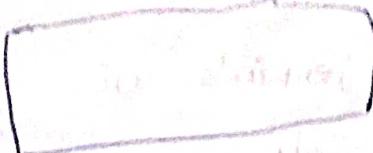


choice parameter

Name

Branch

choices

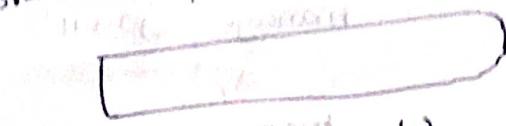


master

bog
sugheer
saghra/ahmed

Branches to build

Branch specifier



* / \$ (branch)

\$

Apply & Save

J1

Build with parameters

click

selected branches dropdown J1

go to dashboard

Install plugin at jenkins

click on

manage plugins jenkins

J1
manage plugins

J1

Available

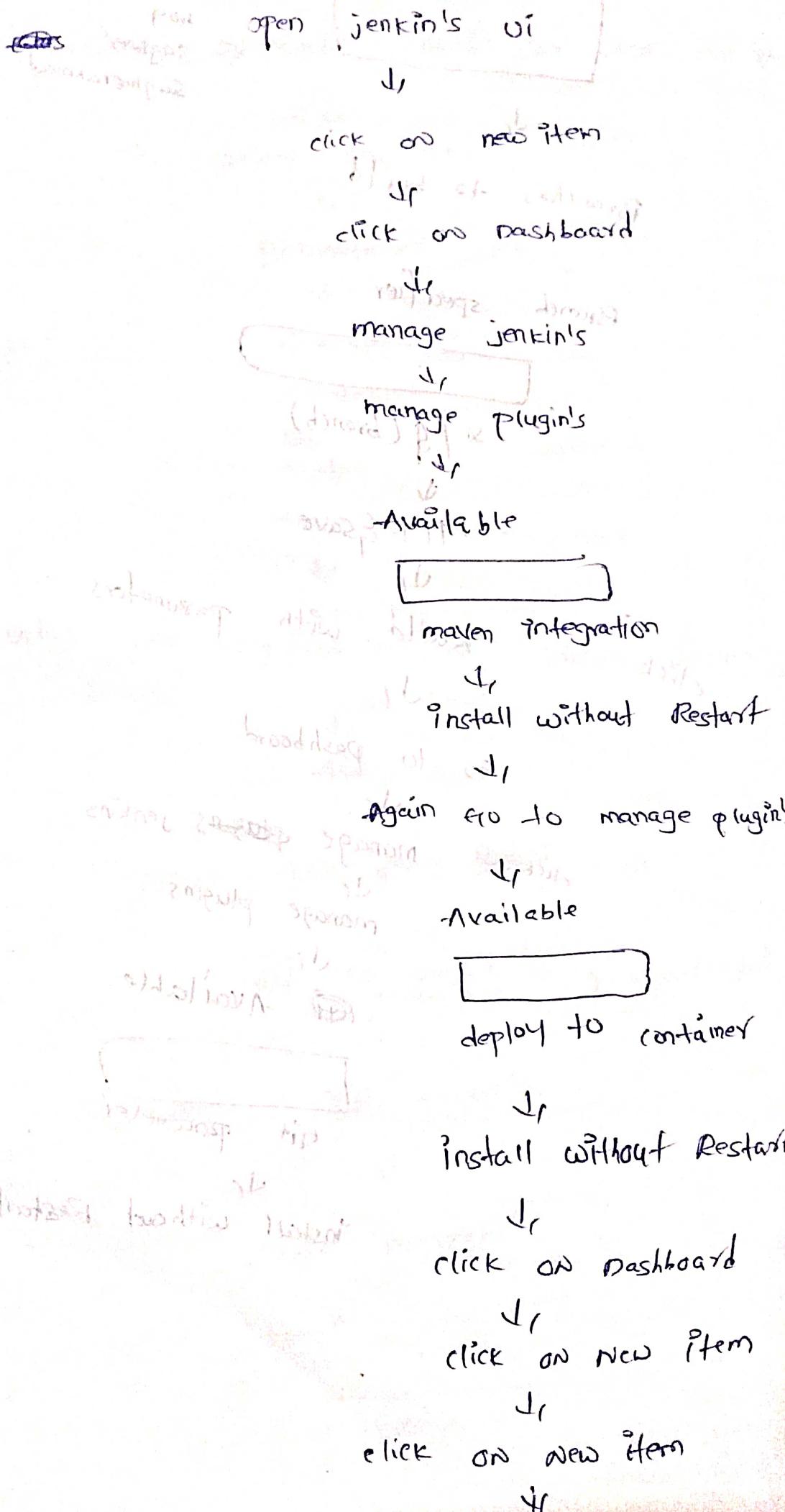


git provider

J1

install without Restart

Maven job :-



Jenkins maven job
git plugin JPL

Select maven project

git repository Jenkins OK

source code management

Build on Git

JPL

repository
URL

<https://github.com/RavitejaAdopudi/java-with-github>

username JPL

click on Add

password JPL

check Build

jenkins need

Apply & Save

Build 2.1.6 JPL

click on Dashboard

JPL

click on manage Jenkins

JPL

click on Global Tool configuration

Java

JDK

Add JDK

click on Add JDK

Remove install automatically

On check box install automatically

name

JAVA_HOME

JAVA_HOME

Go to terminal

cd /root

cd /opt

echo \$JAVA_HOME

/opt/jdk1.8.0_301

copy

Go to jenkins's UI

click on JAVA_HOME

/opt/jdk1.8.0_301

mvn

↓
click on add maven

↓
Name

↓
MAVEN_HOME

↓
MAVEN_HOME

↓
MAVEN_HOME

↓
Go to terminal

↓
echo \$MAVEN_HOME

↓
echo \$MAVEN_HOME

↓
echo \$MAVEN_HOME

↓
/opt/apache-maven-3.5.3

copy

that

↓
↓

↓
Go to Jenkins UI

↓
Go to Jenkins UI

↓
MAVEN_HOME

↓
MAVEN_HOME

↓
/opt/apache-maven-3.5.3

↓
↓

↓
Apply & save

↓
↓

↓
Go to Dashboard

↓
↓

↓
click maven job

↓
↓

click on configure

1

check the build

1

click on pre - steps

~~2000~~ 1999

Root - POM

param. XML

Goals & options

(700e- 9-9
1000 fm est. 51 57)

— 1 —

clean

Apply & save

34

click on Build

四

click or config.

click on post steps

1

post-build Actions

Select → Deploy war / ear to a container

WAF / EAR files

click on question mark
(8)

COPY ~~xx~~ / ~~xx word~~

paste in book

WAR | EAR files

1988-01-11 2011-01-01

xx 14 wav

click on container

1960 11 25

add container

select Tomcat8.x Remote

in credentials

Add

Click on Add

select

User
name

password

Admin

~~password~~

PIB

my fan craft

click on Add

Credentials

--none--

四

① click on none

25/09/2021

Select Admin

↓
copy forced url

https://1118.219.86.151:8080

↓
end of xx

↓

Apply & save

↓

click on build

Backup strategy:-

Go to jenkins

↓

manage plugin's

↓

manage plugin's

Thin

Backup plugin

Install

without Restart

uncategorized

Go to manage jenkins → ThinBackup

↓
settings

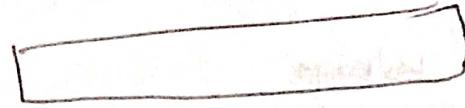
↓

Backup directory

↓
[]

↓
/root/backup

Backup schedule for full backups



* * * * *

selected build - op

build number file

↓

build number file

select

wait until Jenkins

Backup build results

Backup 'user content' folder

Backup next build number file

Backup plugins archive

Backup addition files

Cleanup differential backups

move old backups

↓

Save

↓

click on ~~Restore Backup & Restore~~

↓

Go to terminal

↓

cd proof

↓

ls

↓

Master & slave :-

crash - memory leakage

CPU utilization — load increase

↓
HFI proxy — to solve issues.

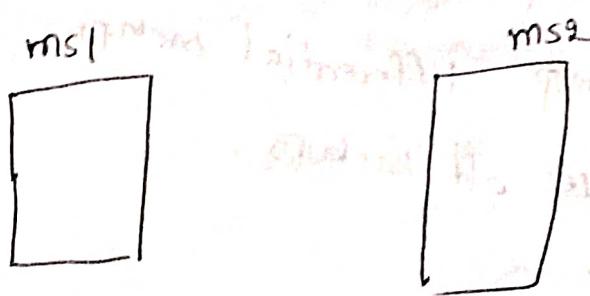
we can maximum 7 nodes.

minimum 3 nodes.

two ways to connect master & slave:

- * SSH → Linux S
- * RDP → windows

password Authentication



ssh-keygen

↓
Enter 3 time

check keys

cd ls -la

cd ssh

↓

ls

ssh-copy-id ms2 public IP address

↓
ssh ms2 public address

ls -la

↓

cd ssh

↓

cd /etc/ssh

↓

ls

↓

vi sshD_config

↓

permit root login : yes

password Authentication : yes

↓
save
↑ source sshD restart

→ connect two missions with password authentication

→ In mission2 (node1) install - java

yum -> install - java -y

↓ Jenkins

go to jenkins ui

↓

manage jenkins

↓

manage nodes & cloud

↓ add a slave

new node

↓

Name : permit Agent

↓

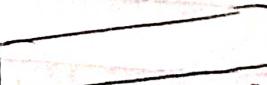
Open port Agent

↓

OK

↓

Remote root directory



host/jenkin's

↓

labels



node1

↓

launch method

↓



launch Agents via ssh

host :
(node) missing public ip address

credential add
J, click on add
J, select jenkins
J, ~~username: root~~
password: 12345
(missing root password)

forget add host key
J, click none
J, Select root
J,

Host key verification strategy

manually tested key (3)
J,

click on save

J, Go to dashboard

J, Select new item

Name: master and slave

↓
select any job
? select freestyle

Go to configure of the job

Hudson

J_j
 Restrict where this project has to run

The name of J_j is given as build

label Expression

[label] node

Build

[Build]

J_j
select Execut shell

command

[shell command]

touch root/vi.txt

Apply & save

Go to node terminal

J_j

ls -la

J_j

cd jenkins

J_j

ls

cd workspace

↓

ls

26/09/2021

scripted Pipeline :-

pipeline job using grove language to sure set of tasks like build test deploy will be performed in sequential order is nothing but a pipeline.

Go to jenkins ui

new team

↓

name : scripted pipeline

pipeline → K

configure

click on pipeline

pipeline syntax :-

select pipeline script

Syntax :-

```
pipeline {
    agent any
    stages {
        stage ('clone the code from github') {
            steps {
                git url:'https://github.com/RavitejaAdepudi/java-war.git'
            }
        }
        stage ('testing the code') {
            steps {
                sh 'mvn clean test'
            }
        }
        stage ('building the code') {
            steps {
                sh 'mvn clean install'
            }
        }
        stage ('deploy the code')
    }
}
```

steps

{

sh 'cp -R root/.jenkins/workspace/scriptedpipeline/H

/* /opt/apache-tomcat-8.5.3/webapps)

}

}

8

3
27/09/2021

Declarative Pipeline :-

Go to jenkins UI



Select new Item



Name : DeclarativePipeline



Select pipeline



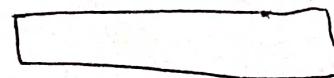
Set OK



Click on pipeline



Pipeline



Select pipeline script from SCM



Repository

[redacted]

https://github.com/purini798/vijay.git

Branch specifier

script path

[redacted]

Jenkins file

→ automatically generated

Go to GitHub

Create new file

filename: Jenkinsfile

data:

pipeline

{ agent any

stages

{ stage ('ai file')

steps

{
sh

~~git add . & git commit -m "initial commit"~~

(touch a1.txt)

stage

(a2 file)

{
steps

{
sh

(touch a2.txt)

-

{
}

step

stage

{
commit

click commit

↓

now go to Jenkins UI

↓

save & apply

↓

click on build

↓

now go to Github

↓

select Jenkinsfile

↓

Edit Jenkinsfile

changes added

add data in file

stage (1 & 3 file)

{

steps

{

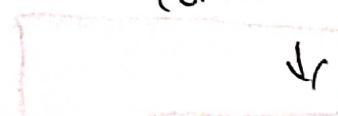
sh ('touch' file)

}

{

commit changes

commit changes



Come to jenkins ui

↓

click on Build

↓

go to git hub

check commit's /