

SESSION: 107

Jenkins

- Jenkins is a tool that helps in CI/CD deployment of code

Continuous Integration
Continuous Deployment

Continuous Delivery

Simple Jenkins

Cloud-based Jenkins

Enterprise Jenkins

Cloud-based Jenkins

Cloud

Cloud

Cloud

Cloud

SESSION: III

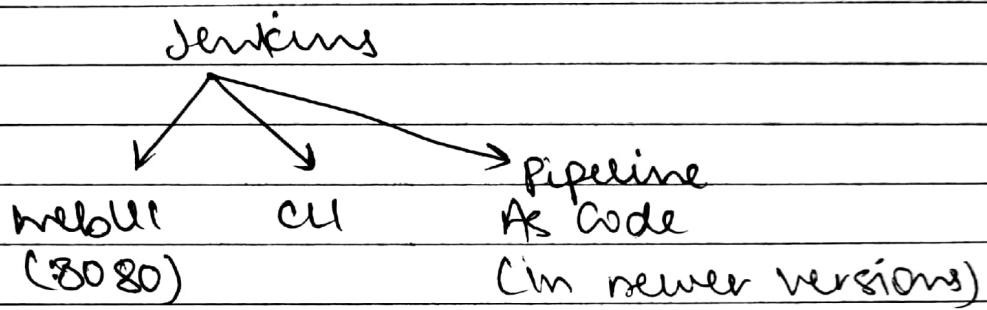
Jenkins
made a
Guru

Install new PHE VM.

↳ 2Gb RAM

↳ 10-20Gb storage

- We need plugins to integrate jenkins with github and create a job.
- If we execute a job in jenkins, that concept is known as build.



Systemctl & restart jenkins

netstat -nlp | grep java

↳ 0:::8080

↳ jenkins works on port 8080

Web: ip : 8080

Login page of jenkins,

~~Github~~

Create a new repo

↳ jenkins-class-Repo

- Jenkins job is provisioned in slave
- # rpm -ivh — rpm

Go to browser.

Up: 8080

↳ set ~~user~~ passwd. & install plugins.

Ansible Playbook

Ansible Playbook

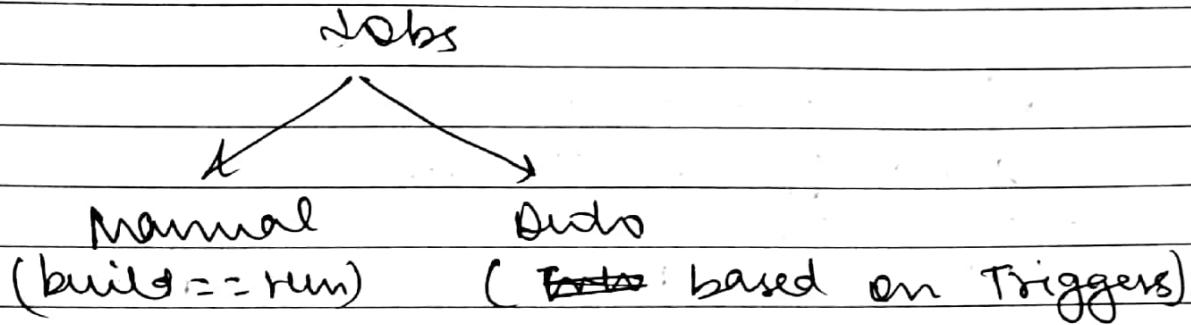
Ansible Playbook

Ansible Playbook

Ansible

Ansible

Ansible



Systemic status jenkins

— step firewall

~~GitHub~~

Create a repos

↳ Jenkins-training-repol

git bash

\$ git clone <git user>

\$ cd Jenkins-training-repol

\$ vi index.php

<?php

print "first code";

?>

\$ git add .

\$ git commit -m "One".

\$ git push .

curl

yum install httpd -y

enable httpd --now

Sun's initial

```
$ vim index.html  
first pg of home !!!  
$ git add .  
$ git commit -m "comment add"  
$ git push.
```

Build ↗ all the files

```
sudo cp * /var/www/html/ -f
```

- we need to create a sudo privilege for jenkins user.

```
# @ vim /etc/sudoers  
jenkins ALL=(ALL) NOPASSWD: ALL
```

Build Triggers

↳ Build periodically

```
# 50 23 * * *
```

(min hr date month day)

```
* * * * *
```

(runs automatically ~~every min~~)

- Triggers are very imp in automation.

C. in Jenkins.

→ Pull SCM

→ GitHub

→ API

VPN/Tunneling

~~will use~~

ngrok download

cd downloads/

unzip ngrok - .zip

./ngrok http 80

Jenkins server.

new job (jobz)

↳ Build triggers

↳ SCM

↳ Build periodically

(After certain time on job
will be run by itself)

↳ SCM

(Build the job only if sending
changes)

↳ * * * * *

(goes to github every min.)

- The diff. is they (Pull SCM) only download the code only if it changes otherwise they won't. If we want to take backup we can use build periodically,

- ↳ GitHub Web trigger for GitSCM polling
 (Only go to GitHub & execute job button)
- (Launch a new job only when ~~new~~ changes are made in GitHub)
- (Here GitHub comes to Jenkins).

↳ Build cmd

↳ Bash myjob.sh

- In GitHub Web trigger we also need to enable webhook in GitHub also.

GitHub (repo that you want to config)

↳ settings

↳ webhooks

↳ <Payload URL → /github-webhook

(http://192.168.100.11:8080)

↑ Jenkins ip

↳ This is port ip.: won't work
we need to provide

-Ingress http 8080 public ip

- replace the public ip to payload URL

followed by /github-webhook

↳ Trigger build remotely
(we can call the trigger through
a button on DPI or mobile
app).

↳ Shows how to build a project
using the build API.

↳ Shows how to build a project
using the build API.

↳ Shows how to build a project
using the build API.

↳ Shows how to build a project
using the build API.

SESSION: 115

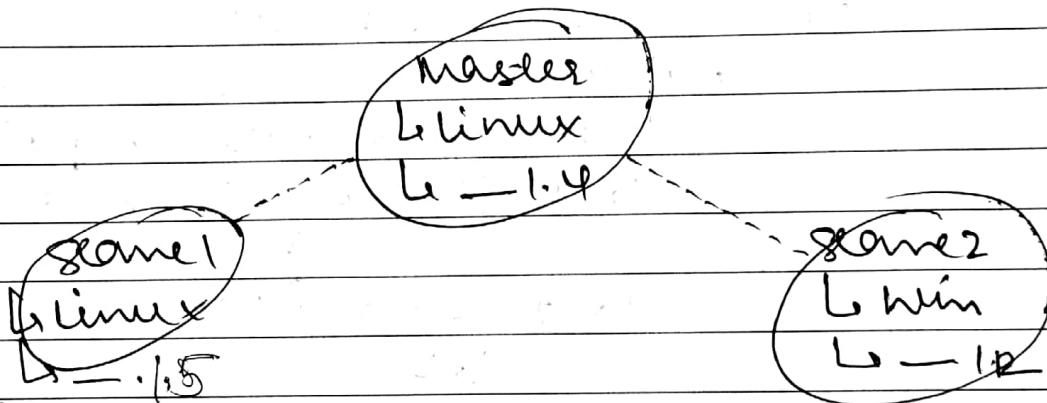
Jenkins

Jenkins Cluster

↳ Distributed Jenkins Environment

- In cluster master node launches the OS as per the requirements of the client.
- Hadoop : storage cluster
K8S : management cluster
Jenkins : job cluster
- Based on the job we are distributing to the worker nodes. ∴ known as distributed ~~job~~ cluster.

Jenkins Cluster



Web UI

↳ Dashboard > Nodes > master

↳ Configure

of executors

↳ 2

Create a new job
↳ job10: name.

↳ Build → Execute shell

sleep 30

- Max of 2 diff jobs we can run in parallel. Depends on the no of executors.

In this exp. no of executors were 2.

- Similarly we can run same job in parallel.

↳ concurrent jobs.

Manage Jenkins

↳ Manage Jenkins Master

↳ New Node slave node

↳ Node name: slave1 py3.8

↳ Permanent Agent



↳ # of executors

↳ 2

↳ Remote Root dir

↳ /root/myws

~~node slave~~
Systemctl start sshd ↳ we need to create
Should be enabled for this dir manually
connectivity

- we first need to download a SSH build agent plugin. Please see cluster.
- To configure a slave/worker node we need an Agent program and slave - master should be connected via network (SSH protocol)

Install plugin: ssh build Agent
↳ Restart jenkins

Launch as node

↳ Launch Method

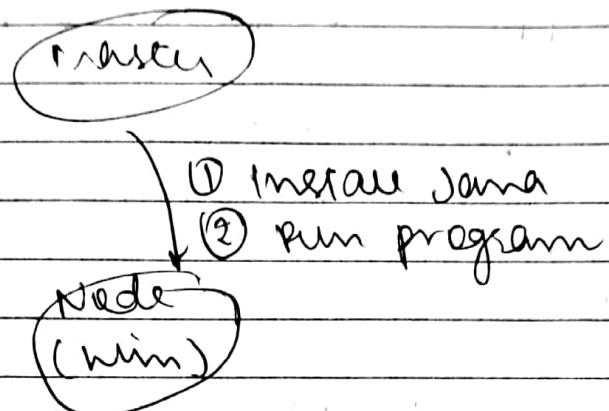
↳ Launch agent via SSH

↳ Set host & credentials

↳ Host key verification strategy

↳ Non verifying verification strategy

strategy



- In windows, by default ssh is not available. & we need to install it.
Jnlp is a program for master to agent communication.

~~wina~~ # java -version
↳ we already have java hotspot

Manage Nodes & Cluster

↳ New Node

↳ win mode

↳ Launch method: Launch agent by connecting to master

↳ Global security

↳ Security realm: None anything

↳ Authorization: Anyone can do it

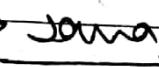
• copy cmd jnlpaws https:// .jnlp

* → paste and

- This will first install 'jupyter' program
and then connect it

SESSION: 122

Jenkins

- maven  pre-req: JDK
 - ↳ automation tool
 - ↳ automate build process
 - ↳ can help you to download all the useful libraries, modules for your code at run-time.
 - ↳ maven central (centralized repository)

• pom.xml

↳ config file for maven.

• Maven - uses

- life cycle or Build pipeline } uses
- ↳ download packages with dependencies
 - ↳ compiles all code
 - ↳ unit testing
 - ↳ report

maven download URL

(Apache-maven-3.6.3-bin.tar.gz)

tar -xvf apache-___.tar.gz

cd apache-___.bin

~~git clone~~ maven /maven

git clone <git url of maven>

wmalinukwesl13/simple-java-maven

* <pwd> /ws5/simple-java-maven-app
 • always run maven cmd from the
 die given pom.xml file.

mvn compile
 L maven: cmd not found

mvn apache-maven-3.6.3 /maven3
 L maven die from treat to /
 bcs. here all the users can
 go.

{ # export
 # MAVEN_HOME=/maven3/
 L env variable
 # export PATH=/maven3/bin/:\$PATH

vim /root/.bashrc
 export MAVEN_HOME= —
 export PATH= —
 L To make it permanent

cd /ws5/simple-java-maven-app
 # mvn compile
 # ~~mvn package~~
 # mvn package
 # java -jar my-app.jar
 L To run the package.
 # mvn test
 L test the compiled code

Manage Jenkins

↳ Plugins

↳ Available

↳ search: Maven Integration

Plugin

(Install)

systemctl restart jenkins -now

Manage Jenkins

↳ Global Tool Configuration

↳ Maven: Add Maven

↳ install automatically
or (v3.6.3)

Maven Home

(tell where it is already
installed)

Create new proj.

↳ myjavaproj:Name

↳ Build

↳ invoke top-level Maven targets

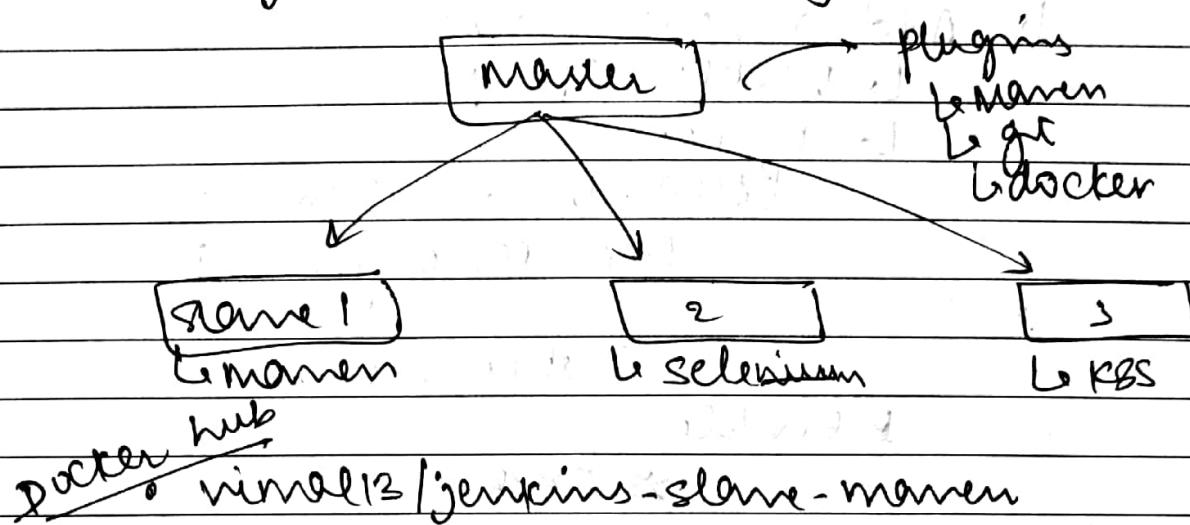
↳ clean: test

>> Build Now

SESSION: 123

Jenkins

* Dynamic Provisioning of Worker nodes



Manage Jenkins

↳ Plugins

↳ docker plugin

(Install)

<Restart Jenkins>

vim /usr/lib/systemd/docker.service

ExecStart= :// -H tcp://<ip>:2376

~~local~~

systemctl daemon-reload

a restart docker

Manage Jenkins

↳ Nodes

~~Configure Nodes:~~

↳ Configure Clouds

- For jenkins docker engine is a cloud

 - ↳ Add a new cloud

 - ↳ Docker (plugin)

 - ↳ Docker

 - HTTP://192.168.1.3:2376

 - ↳ docker engine up

 - ⇒ Test connection

 - Enable

 - ⇒ ~~Save~~

 - # systemall swap firewalled

 - # ————— disable —————

 - ↳ Docker Default Template

 - ↳ Add Docker template

 - ↳ Image: nirmalb/jenkins-slave name

 - ↳ Name: myslavenamen

 - ↳ Labels: myslavenamen

 - ⇒ same

 - # New job

 - ↳ docker: name

 - ↳ Freestyle

 - ↳ Build

 - cmd: docker

 - sleep 30

 - ↳ Restrict where project run

 - ↳ myslavenamen

 - > same

#Build now

- Repeat last practical of maven

↳ git:

↳ ~~mvn~~ ~~mvnw~~ invoke top level maven
↳ I.goal

↳ ~~maven~~ test

↳ II.goal

↳ ~~maven~~ package

- In dynamic slave one we is deleted as soon as the job is done.

↳ post build

↳ Archive-the artifacts

↳ target/*.jar

- Create a post-build (archive the artifacts) and the artifacts are stored in master.

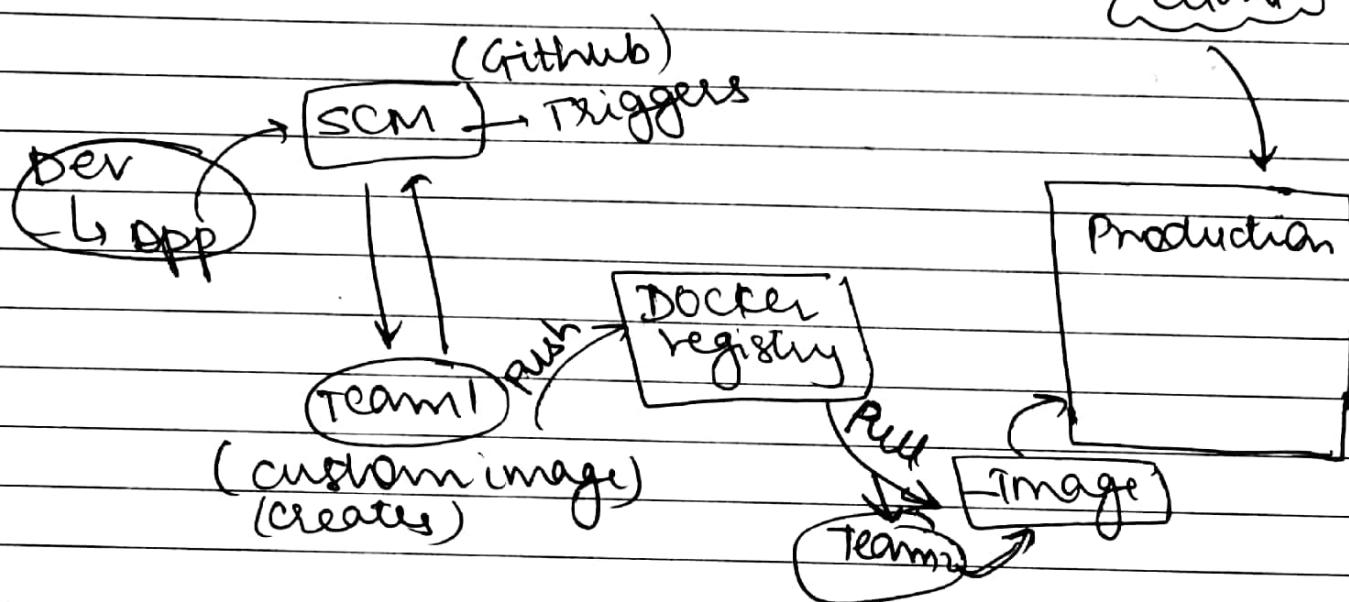
SESSION: 125

Jenkins

Jenkins

↳ Docker Images

Dynamically



- As soon as new code is written Team1 has to create the custom image and name to give to production.

Team2 → K8S / Openshift

Team1 → ~~Jenkins~~ Jenkins

mkdir dj

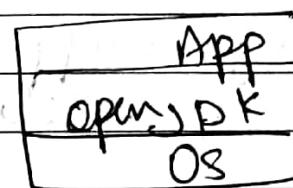
git clone <simple maven app>

cd simple-java-maven-app

→ mvn package

cd target/

↳ myapp-1.0-SNAPSHOT.jar



cd # ls

↳ simple-java-maven-app

vim Dockerfile
FROM openjdk:8

COPY target/*.jar myapp.jar
ENTRYPOINT ["java", "-jar", "myapp.jar"]

docker build -t japp:v1 .

docker ~~commit~~ tag japp:v1

↳ docker push nimal13/japp:v1

docker run nimal13/japp:v1

↳ output shows !!

Plugin to Jenkins

↳ CloudBees Docker Build & Publish

↳ docker-build-step

new job

↳ javaapp-custom-app
↳ freestyle

↳ GitHub Project

↳ Project URL: <git link mavenjava>

↳ SCM

↳ Repo URL: <maven java app>

> Build now

↳ Configure

↳ Build

↳ Invoke top level maven

↳ goals: package

> Build now

① ↳ Docker build & publish

↳ repo name: javapp

↳ registry: dockerhub

↳ credentials

- To create a docker file. we need to have the Dockerfile at the same place where we have our code.

i.e. git hub.

> Build now

usermod ~~username~~ -aG docker jenkins

↳ usually docker is always run with root cmd but we have to use jenkins user.

∴ we need to make jenkins user a group member of docker.

systemctl restart jenkins

After booting up the system it will automatically start Jenkins.

Now we can see Jenkins is running on port 8080.

Open browser and type <http://localhost:8080>.

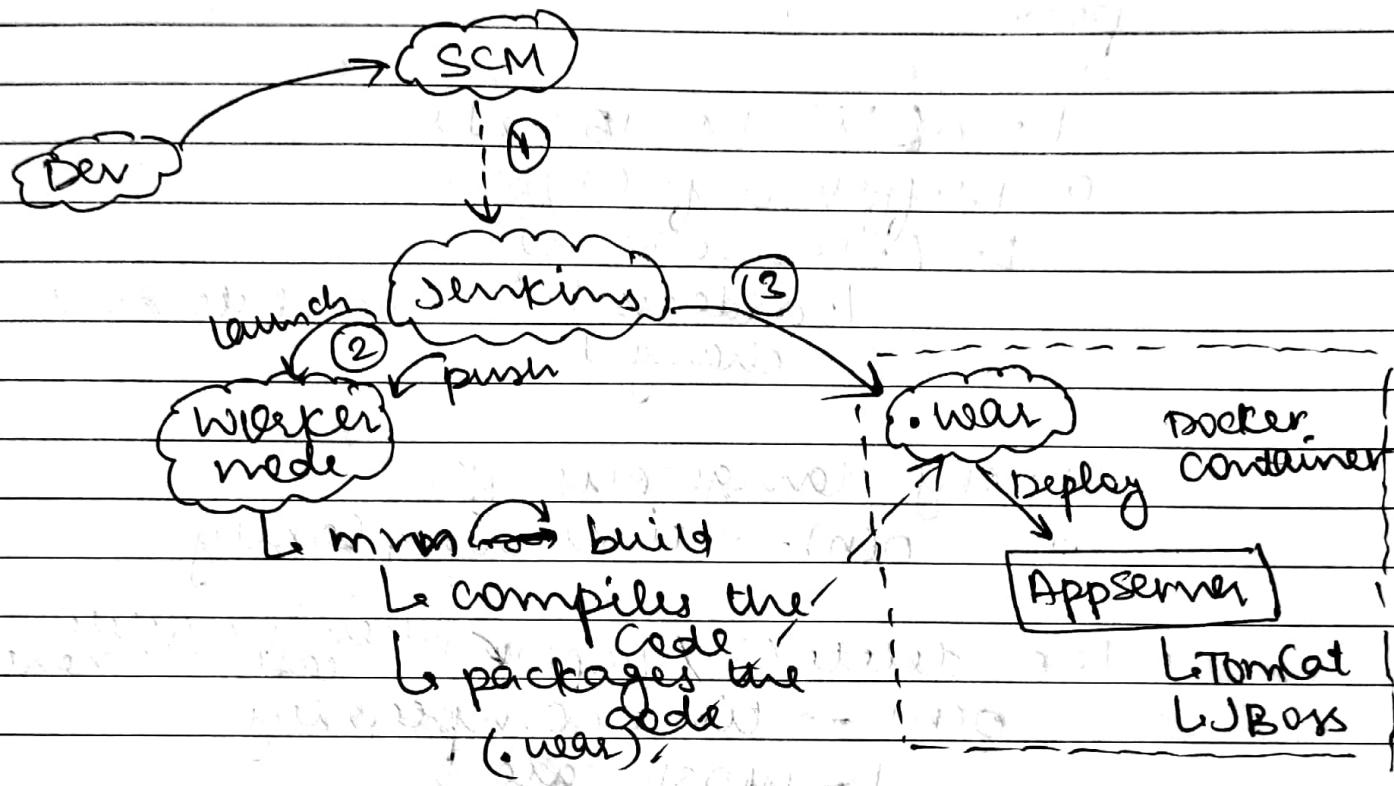
Now Jenkins dashboard will appear.

Now click on "Manage Jenkins" link.

Now click on "Configure System" link.

Now click on "Edit" link.

Session : 129 Jenkins Pipeline



Processes

Jenkins
Master

1 job

2

3

git

2

min

3

deploy

- Achieving this setup with single jobs is very hard (impossible). Jobs are assigned for each node.

- ∴ we create different jobs for each steps.

↳ Job1 - git

- # ~~these~~ jobs ↳ Job2 - miniflux
- ~~these~~ ↳ Job3 - deployment

Pipeline

↳ Job1 → Job2 → Job3
 ↳ n1 ↳ n2 ↳ n3

- ∴ pipeline is the end to end process.

w.r.t job2, job1 is upstream job
 & job3 is downstream job

Add Job

↳ view name: myview

↳ list view

↳ defining job filters

↳ select all the jobs that you want to view,

New job

↳ job

↳ freestyle

↳ build

↳ sleep to

↳ same

ajb2

↳ freestyle

↳ Build

↳ sleep 5

ajb3

↳ freestyle

↳ Build

↳ sleep 4

~~↳ Freestyle~~~~↳ Restricted~~

- All the jobs are independent of each other.

aj1

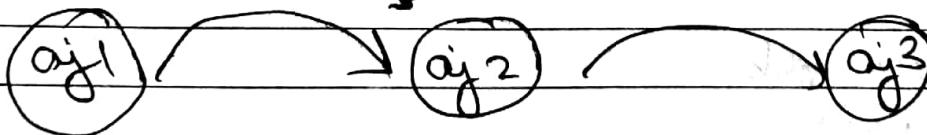
↳ configure

↳ Build Triggers

↳ Build after other projects are build

↳ project to watch: aj1

trigger ↳ Trigger only if build is stable



aj3

↳ Build trigger

↳ Build after a, pref

↳ aj2 ↳ build is stable

- Now as soon as aj1 is built successfully it will launch aj2 and then aj3.

~~Config~~

Manage Jenkins

↳ Install

↳ delivery pipeline plugin

(Install)

~~DU~~

↳ View name: dprent1

↳ @ Delivery Pipeline View

> OK

↳ Pipelines

↳ Components (add)

↳ aj1 : initial job

(automatically pulls all
the downstream jobs)

> OK

~~Config~~ Manage Jenkins

↳ Plugins

↳ Build pipeline plugin

View

↳ Job View

↳ build Pipeline view

↳ @ Initial Job

↳ aj1

> OK

- Jenkins Pipeline (CPAC)
- Pipeline as a code
- Language: Groovy

• Pipeline as a code

• Configuration as code

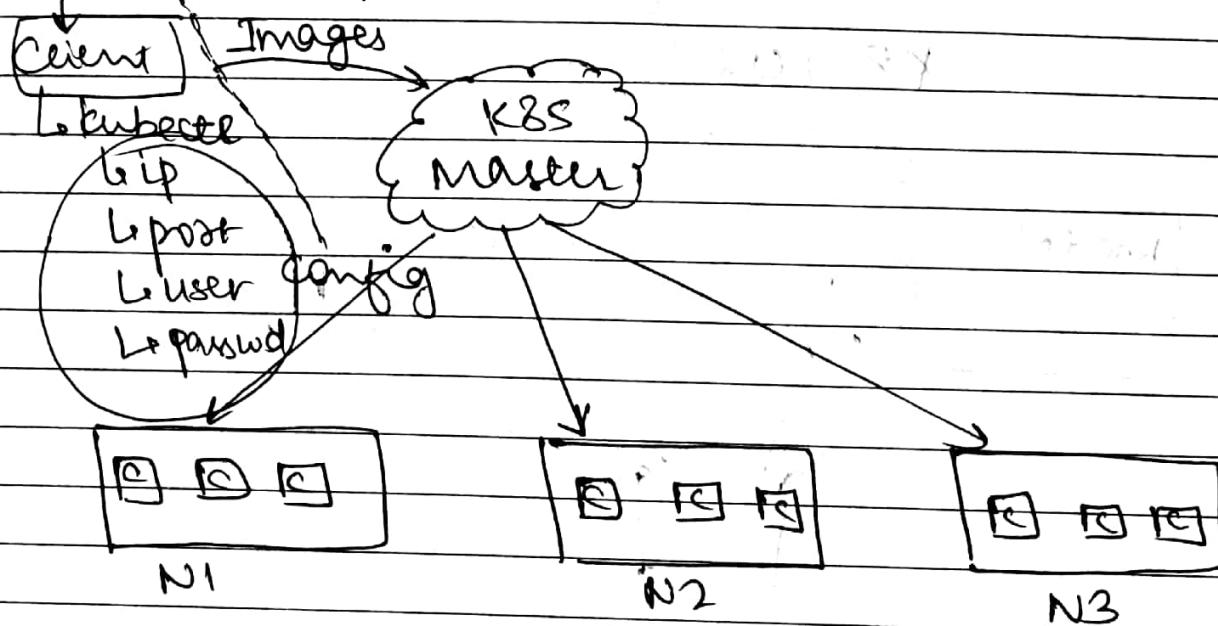
• Configuration management

SESSION: 131

Jenkins

Jenkins with Kubernetes

~~Jenkins~~ (plugins for k8s)



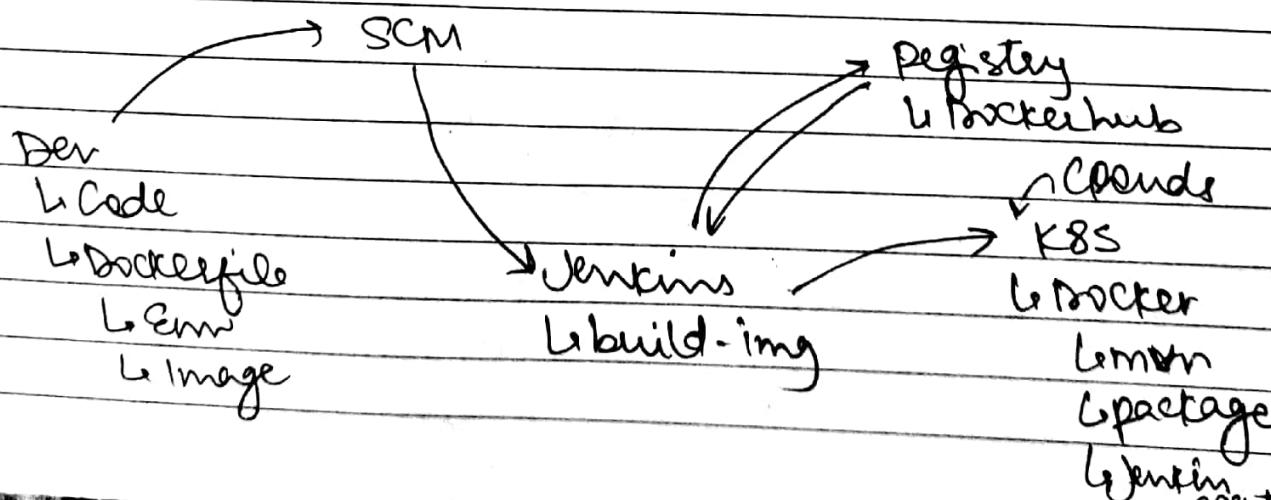
Manage Jenkins

↳ Plugins

↳ Install : Kubernetes plugin

~~minikube~~

kubectl clusterinfo



Manage Jenkins

↳ Nodes

↳ Configure clouds

↳ Add a new cloud: kubernetes

↳ k8s cloud details

↳ ~~host~~: 192.168.99.100

minikube (VM)

↳ login: docker

↳ passwd: user

\$ sudo su \$ - root

\$ cd /etc/kubernetes/

↳ admin.conf

very critical file contains
all the data!

\$ cp admin.conf /home/docker

~~scp from host~~ \$ chmod 644 admin.conf

↳ admin.conf

- Bec only root user has power to read this file but we login with docker user and set perm to 644.

- copy file to /home/docker & chmod 644 Admin.conf

netpad admin.conf

Server: https://~~192.168.99.100~~:8443

In win change this -

... ↴ K8S cloud details

↳ Add (Credentials)

↳ Kind : Secret file

↳ File: choose file

↳ admin.conf

↳ Credentials : admin.conf

> Test Connection

Jenkins URL

↳ Settings > Network

↳ Adapter 2

↳ Host-Only

↳ Adapter #2

(Should be same
as of minikube)

... ↴ Jenkins URL

↳ http://192.168.1.5:8080

Jenkins-ip

↳ Pod Labels

↳ jenkins-worker

↳ jenkins-worker

Also add a label to FDS.

↳ kube-pedagent

new job: jk1

↳ restrict who this project can be run

↳ disablepedagent

↳ Execute shell

↳ date

> Same setting as Jenkinsfile

> Build

• Error: API is not available

≠ ≠

Manage Jenkins

↳ Global security

↳ Agents

↳ ⚡ Fixed: 50,000

↑ Any port no

↳ more commonly used port.

> Same.

... ↳ configure cloud

↳ Jenkins tunnel

↳ 192.168.1.5:50000

↳ jenkins:ip

> Same.

Job

manual

Run (Job As A Code)

Le code (DSL)

Le Groovy ~~and~~ Syntax

- DSL : Domain Specific Language

Manage Jenkins

Le plugin

Le Job DSL : instead

ITI Domains

Le run

Le Jenkins

→ Seed Job

Job

→ Child Job

Job

↳ ~~lived job : name~~

↳ freestyle

↳ Build

↳ Process job DSL (work on filesystem)

↳ DSL Script

or ~~git~~ use SCM for
source code (github)

↳ job('name') { }

description('my first job')

+ steps { }

shell('date')

+ { }

SCMs

github()

+ -> steps { }

+ step { } { command('package')}

shell('date')

{ }

+ step { }

{ }

+ step { } { command('build')}

+ step { } { command('test')}

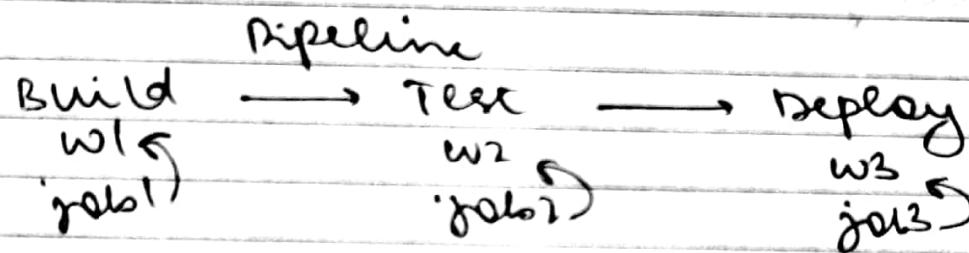
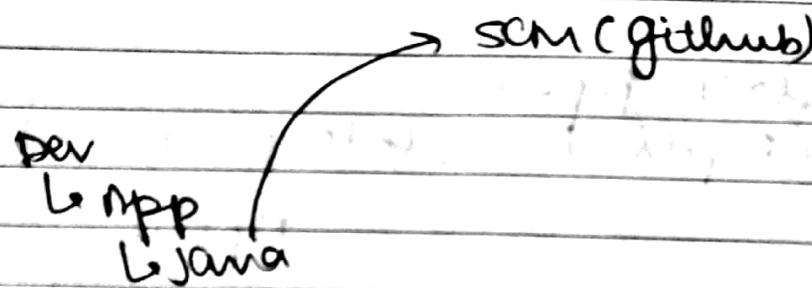
+ step { } { command('analyze')}

+ step { } { command('deploy')}

+ step { } { command('archive')}

+ step { } { command('clean')}

+ step { } { command('post')}



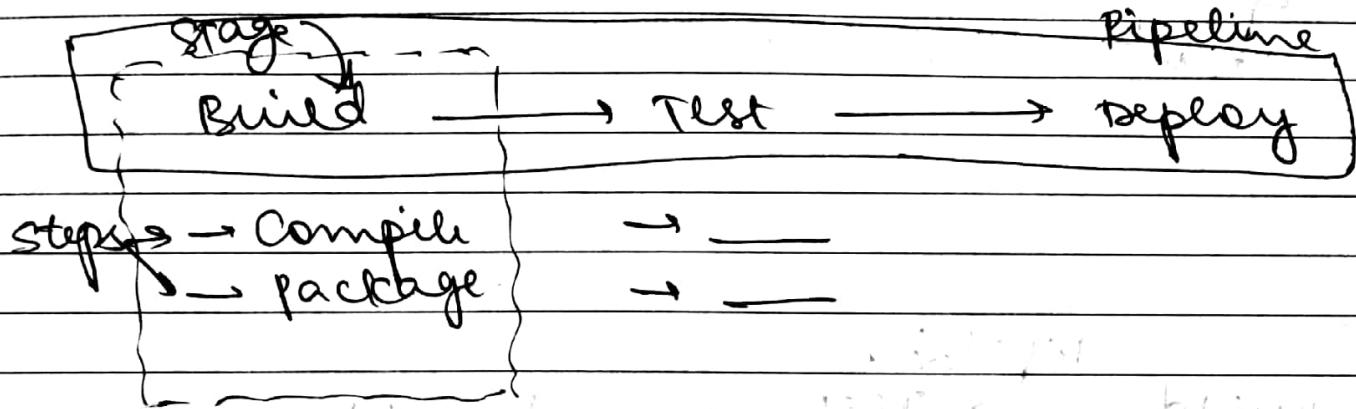
- Job as code \rightarrow DSL
 - ↳ we can create 1 job with the script.
- Pipeline as code \rightarrow DSL
 - ↳ creates the whole pipeline with the script.
- DSL is a plugin created by netflix
- Pipeline is a native plugin created by Jenkins.

Manage Jenkins

↳ Plugin

↳ Install Pipeline

(Pipeline as code with Jenkins)



new job: apipel

↳ (Pipeline)

↳ Pipeline

↳ script:

```

pipeline {
    stages {
        stage('Build') {
            steps {
                echo "my first stage"
            }
        }
        stage('Test') {
            steps {
                echo "test"
            }
        }
        stage('Deploy') {
            steps {
            }
        }
    }
}
  
```

```

stage('Build') {
    steps {
        echo "my first stage"
    }
}
  
```

```

stage('Test') {
    steps {
        echo "test"
    }
}
  
```

```

stage('Deploy') {
    steps {
    }
}
  
```

```

stage('Deploy') {
    steps {
    }
}
  
```

even "deafly"

3

17.11.2017 1900-05

b 2019-05

1st of

1st of 1st of 1st of

xx

2019-05-01 2019

2019-05-01

1

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

2019-05-01 2019

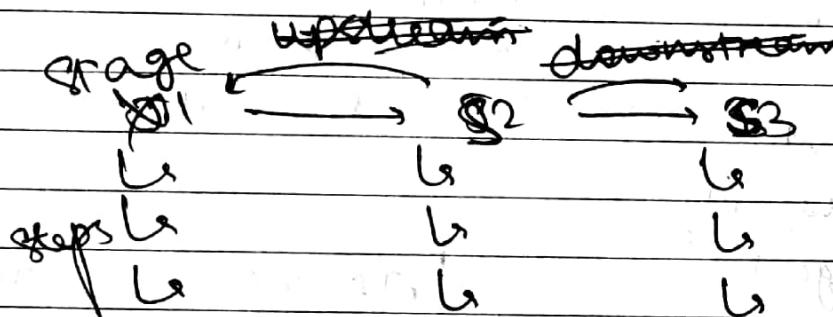
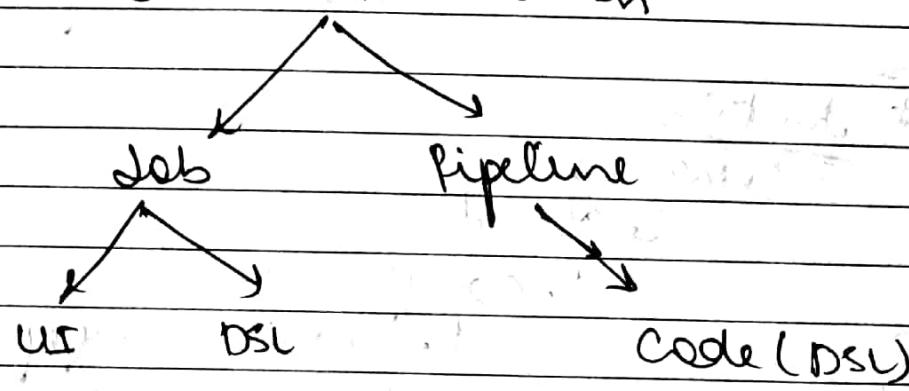
2019-05-01 2019

2019-05-01 2019

SESSION: 138 Jenkins

Pipeline As A Code

Jenkins Automation



Pipeline

Stage 1 → S2 → S3 → S4
 ↗ SCM ↗ Build ↗ Test ↗ Deploy
 ↗ git

Master

Worker

Cloud

leg1 : git (S1)

leg2 : maven (S2)

leg3 : Docker (S3, S4)

~~worker~~

sun docker

netstat -trep | grep 2376

↳ docker should expose 2376 port

 ~~Jenkins~~

Jenkins

↳ Manage

↳ Cloud

↳ config Cloud

agents {

↳ docker setup

node {

label 'myslaveware' ↳ for git

{}

create docker templates

↑ global agent

all stages will
run in same
agent.

↳ manan

↳ tomcat

new : apdockerd

(pipeline)

↳ Pipeline

↳ Script

If we run date
cmd as a step it
failed bcz it is
shell step . so

steps {

sh 'date'

pipeline {

agent ~~any~~ {....}

stages {

stage ('git scm') {

steps {

~~date~~ 'echo "date"'

{}

stage ('build') {

steps {

~~date~~ 'echo "build"'

{}

stage ('test')
steps { echo "test"

pipeline ?
agent none local agent
stage ('gitsem') ↗ for git stage
agent ?
label 'mislabeled'

steps {
 echo 'git

stage 2 $\{$ \equiv at domain level
agent 3 \equiv $\{$ willing
stage steps $\{$ \equiv $\}$

Stage 3 \equiv stage 1

```
# stage('git scm') {
    agent {
        label 'my'
    }
    steps {
    }
```

~~steps~~ *< paste the code generated by snippet >*

① Use pipeline snippet

```
steps {
    echo 'git ...'
    git 'https ...'
    sh 'mvn package'
    sh 'java -jar target/*.jar'
```

use snippet → 6 — for archive —

• If we want to use our tools
pipeline {

agent none

- tools {

maven 'mymaven3'

}

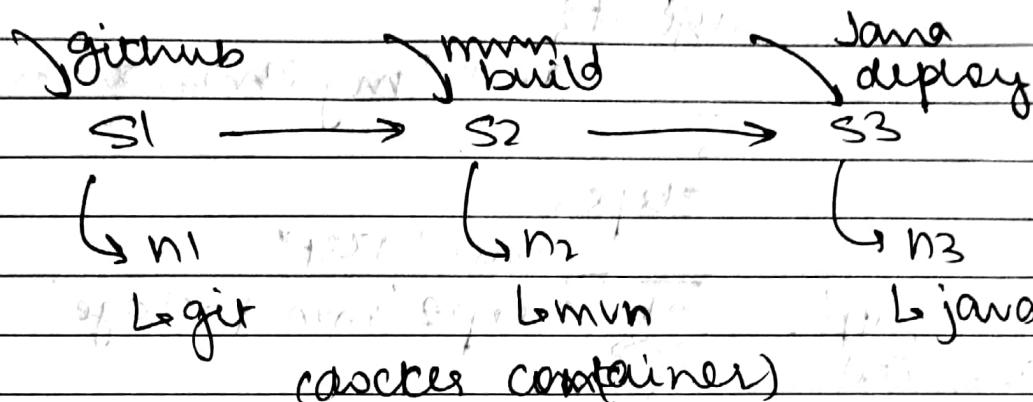
stages {

≡

}

}

session: 143 Jenkins



- When n1, downloads the code in its storage. Its job is completed. Now it is terminated. With it also its code. Now how will get the code?
- We say it uses centralized st. but code is in ip1 but we need the code in ip2... how to do that?

pipeline: ~~git api~~ pipeline

pipeline {

 agent none

 stages {

 stage('Build') {

 agent {

 label 'my slave name'

 }

 steps {

 echo "Build"

~~git 'use'~~

 stash includes: "~~path~~",

 name: 'myapp'

 }

 }

 }

stage('Test') {

steps

agents {

label 'myslave машину'

}

steps {

echo "Test"

sh sleep2 'mvn package'

unstash 'myapp'

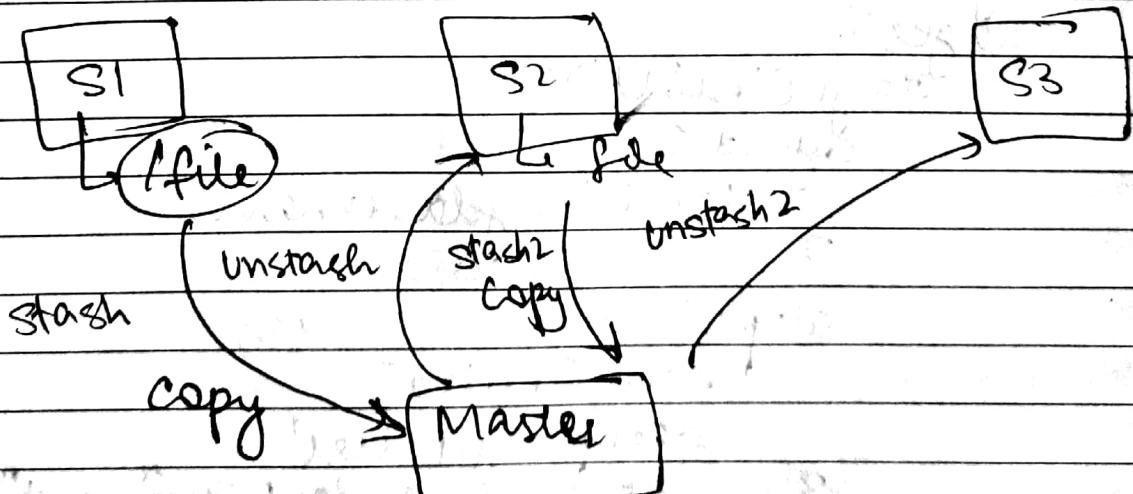
stash includes:

target/*.jar

name: 'myjar' stage('Deploy') { sh 'java -jar target/*.jar'

unstash 'myjar'

- Stage2 will fail bec git code is in n1 and we have not transferred it in n2 and are running mvn package in it.



- we can also store parameters in master.

pipeline }

parameters }

string (name: 'x', defaultvalue: 'w')

defining the pipeline in yml file (yaml file)

(question 52)

I am trying with general question
question 52. what problem

↳ what do you think

↳

↳ general question

↳ question 52. what may be

↳ answer to question 52

↳

↳ general question what may be

↳ question 52. what may be

↳ answer to question 52

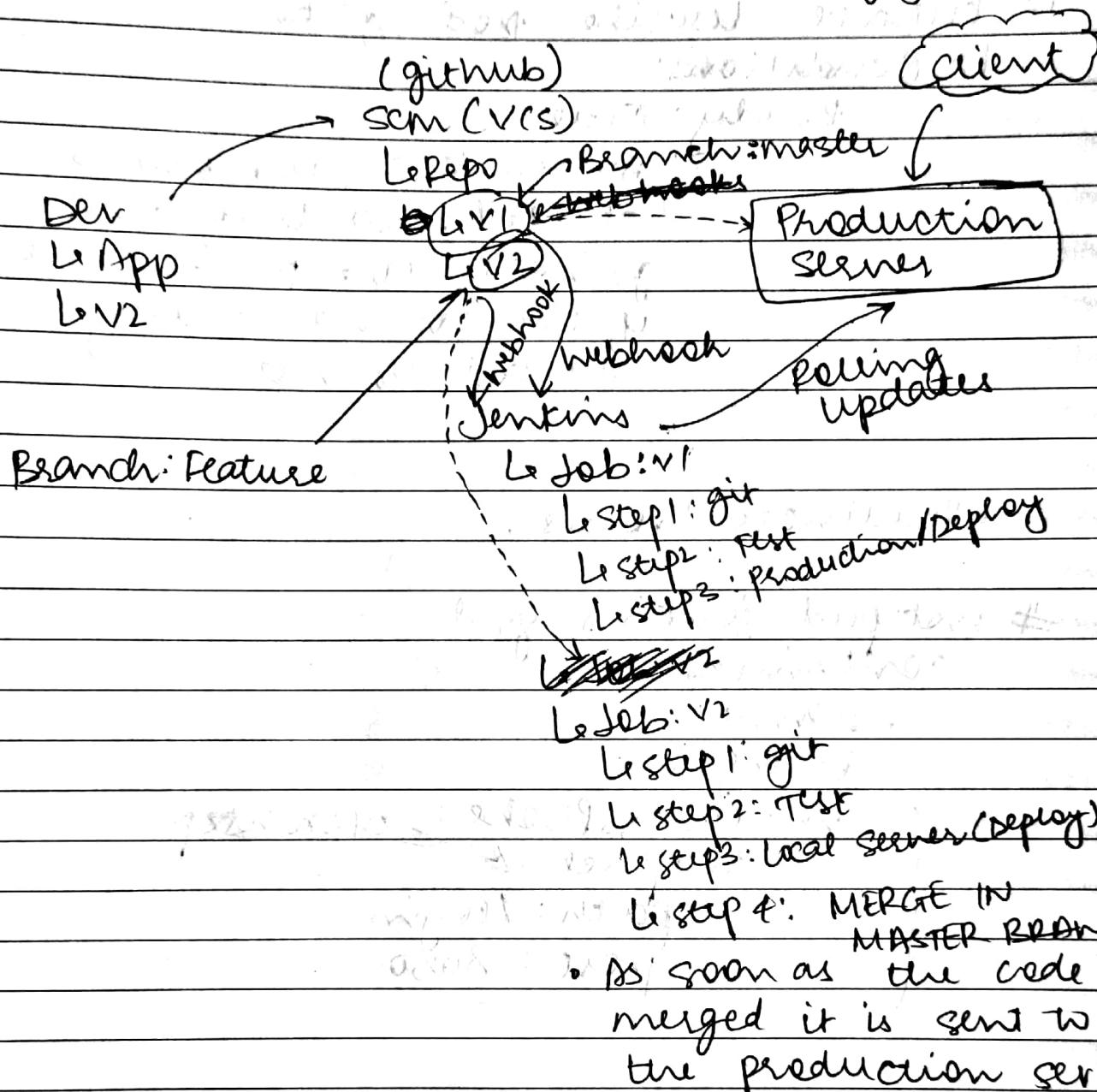
↳ general question what may be

↳ question 52. what may be

Session: 145

Jenkins

Multi-Branche Build / configuration



Jenkins IP
192.168.1.3: 8080

Manage Jenkins

↳ Nodes

↳ Config. Cons.

↳ Docker

Plugin

↳ Pipeline : Install

↳ Pipeline: Multibranch : Install

↳ Docker Pipeline : Install

New Item

↳ name: Multibranch project

↳ ~~Branches~~ Multi branch pipeline

Github Repo: Simple ~~Java~~ App

↳ Create new branch: fix-123.

↳ Branch Source

↳ Project repos: <4H>

↳ Build Configuration

↳ Mode: Jenkinsfile

Jenkinsfile (Github)

pp

Pipeline {

 agent none

 stages {

 stage ('Build') {

 agent ?

 label 'mylambdafunction'

 }

 steps {

 echo "my master br"

 }

 }

 }

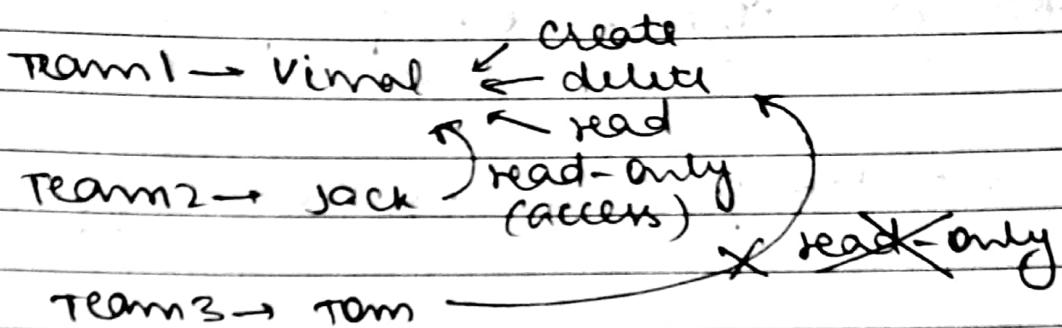
Similar in fix-123.

Session: 148

Jenkins

Jenkins Security → Role Based Access Control

- ↳ IAM (RBAC)



Manage Jenkins

- ↳ security: Global Security

- ↳ Security realm

- ↳ Jenkins' own user database

- (Now might ask login/pw)

- ↳ Allow users to sign up

Create User

- ↳ jack: user

- ↳ password

- ↳ name

- ↳ email

~~Jenkins~~

\$ ps -aux | grep jenkins

cd var/lib/jenkins

ls

cd users /

↳ all the user files are here
+ user.xml

- RBAC : Role Based Access Control
 - every user has different powers/roles.

Authorization

↳ role-based strategy

plugin : role-based authorization strategy (Inetall)

Global Security

↳ auth

↳ role-based strat.

- New option will come

↳ Manage & Assign Roles

↳ Manage Roles

↳ Assign Roles

↳ Anonymous : Admin

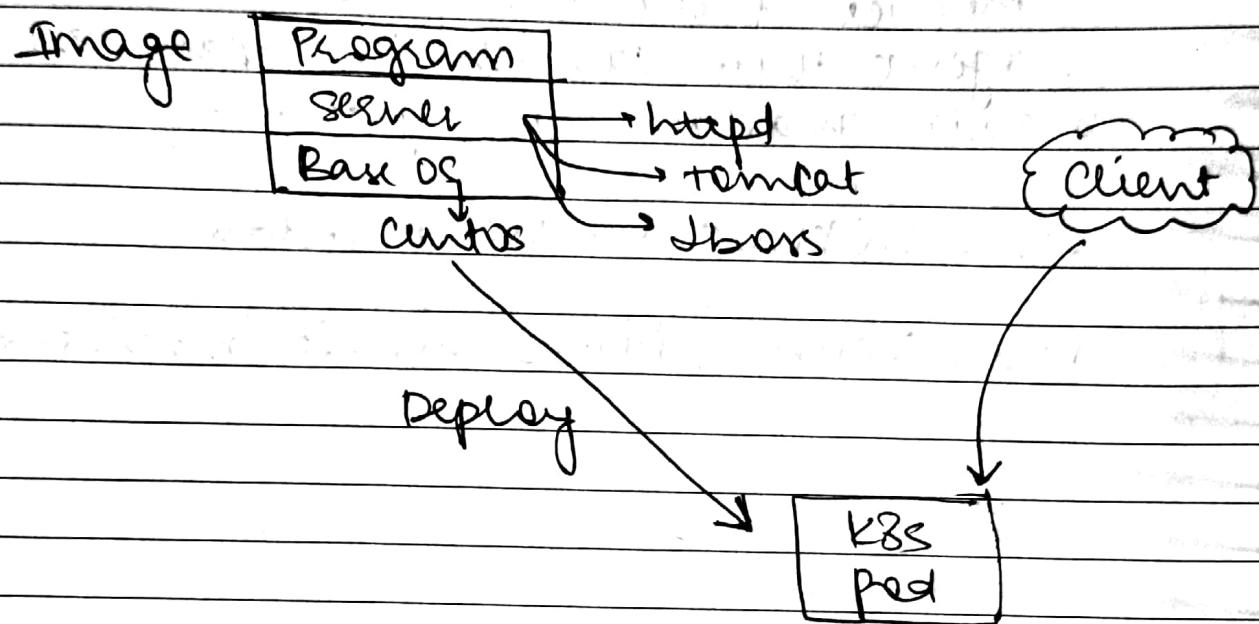
- This is why we can do all the things with anonymous user/ without logging in.

- To remove anonymous user from admin power? to first we need to assign some other user the power of admin. otherwise we will stuck in an infinite loop.

↳ User/add : jack .

Session 153 Jenkins

Migration of Jenkins with K8s



Manage Jenkins

- ↳ Plugins
- ↳ Kubernetes Continuous Deploy plugin (Install)

New Item

- ↳ a kubepipe (name)

- ↳ Pipeline

- ↳ Pipeline

- ↳ received pipeline script from SCM

- ↳ Cut

- ↳ Copy paste git repos URL.

Create new repo : Jenkins -> kube -> pipe -> deploy
↳ add README file
↳ Add a file : Jenkinsfile

```
# pipeline
  agent any
  stages{
    stage('abc'){
      steps{
        echo "hi"
      }
    }
  }
```

run the pipeline (Build now)

~~curls~~
kubectl create deployment myweb1
-image = nginx:1.13/apache
with --dry-run=client -o yaml > d.yaml
↳ upload this code to github.
↳ deploy.yaml

Jenkinsfile
steps{

~~apply~~ apply -f deploy.yaml
sh "kubectl get pods --kubeconfig
`/admin.conf`"

~~Session 103~~~~Session 103~~~~Session 103~~# download kubectl ~~for~~ in jenkins.

wget curl -LO "https://kubernetes.io/minikube/releases/v0.29.0/minikube_0.29.0_amd64.deb"

cp kubectl /usr/bin/

chmod +x /usr/bin/kubectl

(copy config file)

ls /

↳ admin.conf

kubectl get pods --kubeconfig admin.conf

↳ Now it will run fine.

Jenkins pipeline

↳

Build now

~~using~~

kubectl get pods

↳ Now the pod is run by Jenkins

Pipeline (Jenkinsfile) (If we want to run.)

= steps {

↳ KubernetesDeploy(configs: "deploy.yaml",
kubeconfigId: "mykubeconfigfile")

Manage Jenkins

↳ Manage Credentials

↳ Jenkins

(118) ↳ Global credentials : add

↳ Kind

kubeconfig

Scope: Global

Id: mykube configfile

Desc:

↳ kubeconfig: Enter directly

↳ Copy & Paste entire data
of admin.conf.

- We can also write conditions in Jenkinsfile

stage('abc') {
when {
expression {

true

else

}

steps {

sh " kubectl apply -f deploy.yaml
--kubeconfig /admin.conf"

{

agents

~~pass~~

parameters {

booleanParam(name: "allow_ns",
"isdeployed": true)

"for pod?", defaultvalue:
true)

stages {

stage('abc')

=

expression {

params: isdeploypod params, isdeploypod
=

=

peer {

success {

echo "all good"

=

failure {

echo "failed..."

=

always {

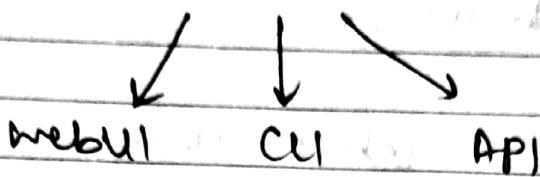
echo "always..."

} — pipeline closed

Session 15

Jenkins

Jenkins



Manage Jenkins

↳ Jenkins cli

↳ lists all the cmd

↳ we need to download jar file (driller)

↳ jama -jar

win dir

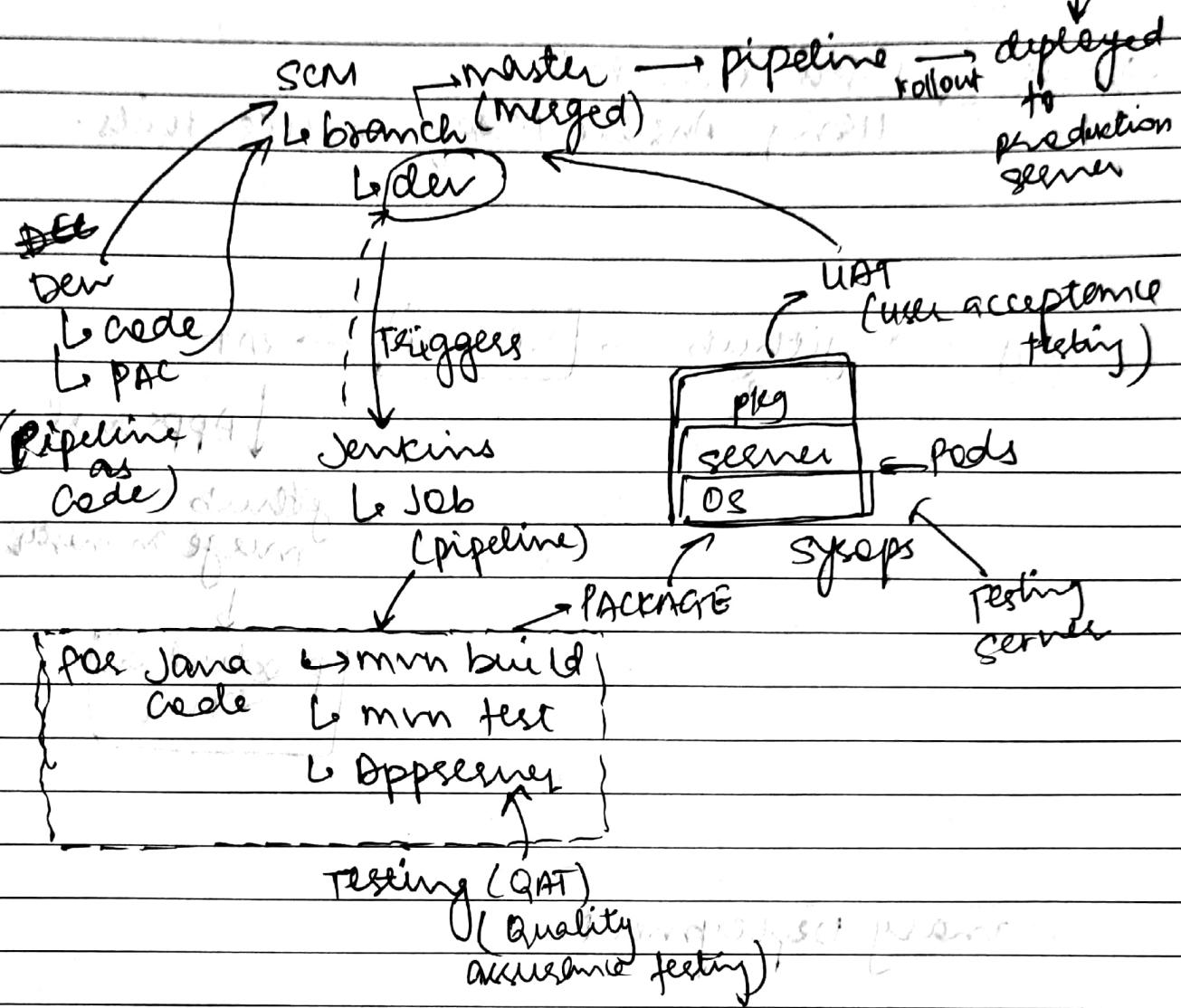
cd & go to dir of driller

paste jama -jar jenkinscli.jar

jama who-am-i

jama list-jobs

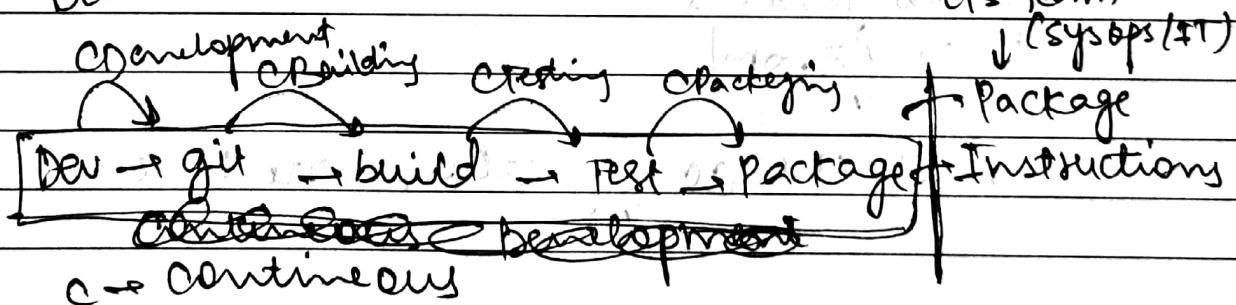
client



- Automation Tools

Config Management: Ansible / Shell scrpt. (IAAC)

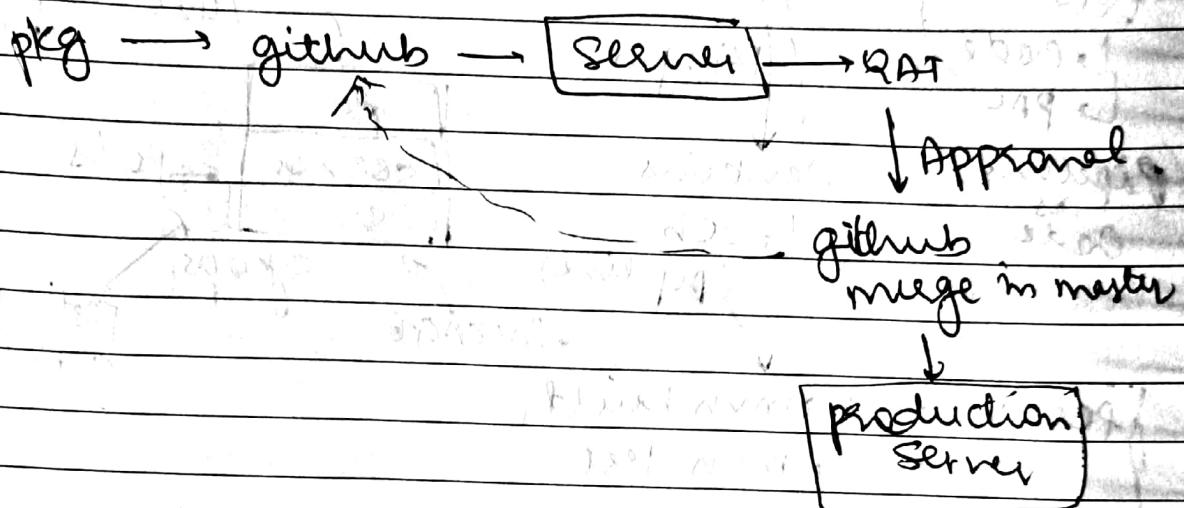
DEV



Build pipeline

OPS

Sysops/IT → Infra is created
 Using Ansible/Puppet/like tools.



• Canary Deployment

~~Jenkins~~

/var/lib/jenkins
 config.xml

All the configurations are here

infra build

QAT app.

new job
↳ pipeline

se:

pipeline {

agent any
stages {

stage (build) {

steps {

echo "build"

}

stage (test) {

steps {

echo "test"

}

stage (uat) {

steps {

echo "uat"

}

input {

message "do u want to app?"

}