SVN (Subversion)

1→ Sudo apt-get update

2→ Sudo apt-get install subversion apache2 libapache2-svn apache2-utils -y

3→ Sudo mkdir -p /svn/

4→ Sudo Vi /etc/apache2/mods-enabled/dav_svn.conf

```
<Location /svn>
    DAV svn
    SVN parent path  /svn
    Auth type  Basic
    Auth Name  "Subversion Repository"
    Auth user file  /etc/subversion/passwd
    Require valid user
</Location>
```

5. Sudo htpasswd -c /etc/subversion/passwd user1
   Sudo htpasswd -m /etc/subversion/passwd user2

7. sudo ~~mkdir~~ svnadmin create /svn/testrepo

8. Sudo chown -R www-data:www-data /svn

9. Sudo /etc/init.d/apache2 restart

10. Check Browser http - ../localhost/svn/testrepo.

Switch to root

* mkdir -p /svnproject/{Branches,tags,trunk}
* cd /svnproject
* svn import "url" -m " Branches t t"
* user add user1, user add user2

passwd user1, passwd user2

mkdir -p /home/user1, mkdir -p /home/user2

chown -R user1 : user1 /home/user1

chown -R user2 : user2 /home/user2

su user1

user1# svn checkout "url"

cd testrepo/trunk

vi index.html

{add code}

:wq

svn status → show "?"

not under version control

svn add index.html

svn commit -m "add file"

"goto user 2"

New window

su user2

2# svn checkout "url"

cd testrepo/trunk

vi index.html

{modified}

svn commit -m "add line"

user1 need to update other wise conflict raises

svn update

svn add file

svn commit -m "file modified

→ This time user2 not update and modified file & commited then conflict raise he accepts their side of conflicts

Tags | Show tags svn ls -v ^/tags ⇒ ^/Branch

user1# svn copy -r4 trunk/ tags/tagname
testrepo

svn commit -m "tag created at R4"

svn checkout "copy url of tag"

Branch testrepo#

svn copy -r3 trunk/ Branches/name

svn commit -m "Branch created"

cd branches/name

vi index.html

{modified}

svn commit -m "Branch add"

merge

trunk

#svn merge .. branches/

#svn commit -m name/

Conflict raise : click m

:12

:Y

Svn list → list all directory entries

Svn delete 'url'
remove a file from repository

Svn move filename1 filename2'
{Rename file (or) directory}

Svn commit -m "Rename file"

svn lock filename ("The file will be locked, others cannot modified")

svn unlock filename ("The file will be unlocked")

Revert:

trunk # vi saifile.
{code..}

svn add saifile

svn status → A.

svn revert saifile → ?

svn info

pg①

GIT

sudo su — ✓

apt-get update

git version:→ 2.17.1

apt-get install git-core -y (install git-core) (2.2.y)

mkdir /repos → create directory

cd /repos

git init → Initialize git

git config -- global user.name Saichand

git config -- global user.email Saichand321@gmail.com

git clone "url" (→ git hub repository url)

cd newgit/

vi index.html

git status file in red clour it means file in working area

git add index.html → git status show green clour it means file in Stage/index area

git commit -m "message ↳ 2nd time direclly we use (git commit -am)

git push origin master (ask credentials git-hub details)

modified vi index.html

git commit -am "message"

git push origin master

generate key

Come to root

@root # cd .ssh/

ssh-keygen → Enter → Enter → Enter

cat id_rsa.pub

{ copy key }

goto git hub project → setting → deploy key → Edit → Add deploy key

Come to git hub project repository

root # cd repos/newgit

git remote set-url origin git@github.com : SaiCharanmamilla/newgit.git
(→ github al; newgit.git ↳ repo)

modify file

git commit -am "message"

git push origin master (This type dont ask credentials of git hub because u provide SSH key)

Branch

At a time Created a Branch & switched to that Branch "Git checkout -b branch name"

git branch → shows all branches

git branch branch1 → create a branch
(↳ Branch names)

git checkout branch → switch to branch1 from master

vi index {modified}

git add index

git commit -m "message"

git push origin branch1

Merge

→ git checkout master

git merge branch1

git push origin master

Delete Branch

git branch -d branch1
↳ After merge

Before merge
(∵ no development in master)

git branch -D branch1

Delete remote branch

git push origin --delete branch
↳ But the Branch commits will be in master

Note
" if u create Branch at HEAD after u will merge → No merge commit id generated"

" if u create Branch at particular Cm id (not Head) after u will merge → merge commit id is generated. (also Branch commits in master)

Working with Tags

→ git tag tagname commit id

→ git push origin tagname

Delete Tag

git tag -d tagname

git push origin --delete Tagname

Working with particular commit id

git branch branchname commit id

git rebase (rewrite the project commit history)

merge creates commit id

But rebase don't create commit id it adds at the top of the master

→ git checkout branch1

file modified and push

→ git checkout master

→ git rebase branch1

git stash → The purpose of the stash is now we are working with two file now u have to work one file other file u kept to stash. whenever u want that stash file u can get it.

modify file Vi index

git add index

git stash ⟶ shelf (Temporary memory area)

(or)

git stash save "message"

git stash apply

git commit -m "message"

git push origin master

git stash list ↳ show stash list

git stash apply

git stash pop apply + drop

git stash clear

git commit revert

git reset --hard HEAD^ → file delete commit id delete

git reset --soft HEAD^

	Commit id	code changes
HARD	X	X
SOFT	X	✓

movefile to directory

mkdir dir → directory name

git mv index dir/index
↳ filename

git commit -am "message"

git push origin master

Rename file

git mv index chand

git commit -am "message"

git push origin master

remove file :→

git rm index (file remove in git core)

git commit -am "message"

git push origin master

(Now file deleted in git hub also)

unstage

git reset HEAD index

git checkout index

Unstage file:→

git reset filename

Revert

git revert commit id filename-

logs

git log --oneline

git dif cmid1 .. cmid2

git log filename

git log --help

git log →

ANT (ANOTHER NEAT TOOL)

(Build Tool)

→ Ant will overwrite old file and generates new file its not maintain versions (changes)

NEED ANT :→ Tomcat, JAVA, ANT

Step1: Download:

1. JAK8 → Oracle → Accept licence → linux-tar.gz → download → paste cplink
2. Tomcat7 → core → tar.gz → down → cplink
3. ANT → 1.10.3 tar.gz → download → pause → cplink

Step2:
Editor:

cd /opt/

opt #1. Wget paste link of Java

mv JAVAfile JAva.tar.gz (edit name)

2. Wget paste link of Tomcat
3. Wget paste link of ANT

Step3: Extract:

tar -xzf Javatile.
tar -xzf Ant file
tar -xzf Tomcat

Step:4: Set the paths.

go to root # cd

vi .profile

export JAVA_HOME = "/opt/Java path"
Export PATH = $JAVA_HOME/bin : $PATH

export CATALINA_HOME = "/opt/Tomcat path(...)"
export PATH = $ CATALINA_HOME/bin : $ PATH

export M2_HOME = "/opt/Maven path"
export PATH = $ M2_HOME/bin : $ PATH

Step 5: update path

Source .profile

Which java
Which mvn
Which ant
echo $CATALINA_HOME .

Ant
| .xml ||

⇒ Startup *.sh

apt-get install git-core -y
git clone "copy url of Ant project" (Sathyadevops/ANTproject)
cd ant-project/

ant (Build the code) ⇒ cd dist ⇒ cp AntExample.war Tomcatpath
modify:→
cd web_content/WebInf/JSP
cp localhost:8080/ant-examp

vi user.form
{Edit file}
cd
cd ant_proj/
ant (Build the project)

cd dist/
cp ant_example.war Tomcat path

Browser
ip:8080/ant-example.war

#

MAVEN

Maven will maintain old versions of project

NEED Maven :→ Maven, Tomcat, Java

Step 1. Download :→

1. Jdk8 → oracle → accept licence → Linux-tar.gz → download → pause → cp
2. Tomcat → core → tar.gz → download → cp link
3. Maven → Binary tar.gz → download → pause → copy link

vi Editor

cd /opt

1. wget paste the link of Java

   # Edit Java upto tar.gz

#2. wget paste the link of Tomcat

#3 wget past the link of maven.

Step 3 Extract:

tar -xzf Javafile

tar -xzf mavenfile

tar -xzf Tomcat

Step 4: Set the paths

goTo root # cd

vi .profile

export JAVA_HOME = "/opt/Java path"

export PATH = $JAVA_HOME/bin : $PATH

" Same as catalina

" Same as maven.

Step 5 update path

Source .profile

Startup.sh

Snapshots :→ Snapshot is a special version that indicates a current development copy

Snapshot (VS) version

in case of version, if maven once downloaded the mentioned version say data-service:1.0 it will never try to download a new available in repository. To download the updated code, data-service version is be upgraded to 1.1

Snapshot :→ maven will automatically fetch the latest Snapshot, every time app-dl team build their project.

apt-get install git-core -y

git clone "url of maven project"

cd mvn_project/

mvn clean
mvn compile
mvn Test
mvn package
mvn install

cd .m2/repository/SATHIYA/DEVOPS/3.0

ls Devops-3.0.war-file

cp Devops-3.0.war Tomcat path/webapps

public cp :8080/Devops-3.0 war

cd.

Modify :→

Vi mvnproject/pom.xml

Edit

cd src/main/webapps

Vi index.jsp

mvn install

cd

cd .m2/repository/SATHIYA/DEVOPS/4.0

cp Devops-4.0.war Tomcat path/webapps

Browser

Public cp : 8080/Devops-4.0 war

#

Docker

Install Docker (get docker for ubuntu in CE edition)

root # docker --version ⇒ (18.03.1 - CE)

Docker images → Show the images

docker images -q → list of images id's

docker rmi imageId → To remove particular images

docker rmi $(docker images -q) → To delete all images

docker run ubuntu → (To create a container and Exited)

docker pull ubuntu → (pulling the image from docker hub)

docker run -it ubuntu → (create a container and its running Now u r inside the container)

docker run -itd ubuntu → (create a Container, in detached mode its running background)

docker run -itd --name myubu ubuntu

↳ container name

docker run -itd -p 9090:8080 Tomcat (portgiving to the host)

given name & port

host / container

docker run -itd -p 9090:8080 --name MyTom Tomcat

docker ps (shows running container)

docker ps -a (Show running & stopped containers)

docker ps -aq (Shows Container id's Only)

docker rm container id → (To remove Contained)

docker Stop Containid (To stop the running Containers)

docker Start Conid (To start the stopped Containers)

pg 20

docker rm $ (docker ps-aq) → (To remove all containers)

docker Exit (Exit the container, when ever u r inside the Container)

docker ctl+pq (Exit the contains, but it is running background)

docker attach containid (Enter the Container)

Create your own image:

docker run -it ubuntu → (ubuntu container created ur inside the container)

{ apt-get update
apt-get install git-core →y
mkdir /repos
cd /repos git init
ctr+pqr (Exit container & run it background)

→ docker Commit 7e5ab1 / ctid gitimage / name image (Create image from container)

→ docker login --username = saichand manthena. → Enter

→ pass wd given

Create tag

docker tag imageid Saichandmanthena/gitimage : latest

{ tag Source_image dest_inge : tag latest]

docker push Saichand manthena/gitimage : latest [To push the image in docker hub account)

docker pull Saichand manthena/gitimage [To pull the image from ur account)

Docker Network.

1. Create Docker network

docker network create chand -d bridge
↳ Name of the NW given

Assign a NW to container

docker run -itd -p 80:80 --name webserver --net chand httpd

⇒ docker network disconnect chand Contid → To disconnect a Container from NW

⇒ docker network connect chand ctid → Connect a Container to a NW

⇒ docker network inspect chand → displayed detailed info of NW

⇒ docker network ls → displayed list of NW's

⇒ docker network rm chand → Remove chand (Bridge) Network

⇒ docker network prune → Remove all unused networks.

Docker Volumes:

host # mkdir files → cd files → vi index.html → Create a file

vi sample.war → Create a file.

pwd ⇒ /root/files → cd

docker run -it -v /root/files:/myvol ubuntu

cont# ls → cd myvols → rm index.html → echo 'Hello' > demo.txt

docker run -it -v /root/files:/newvol --name cont2 ubuntu

cont2# ls newvol ⇒ demo.txt. sample.war.

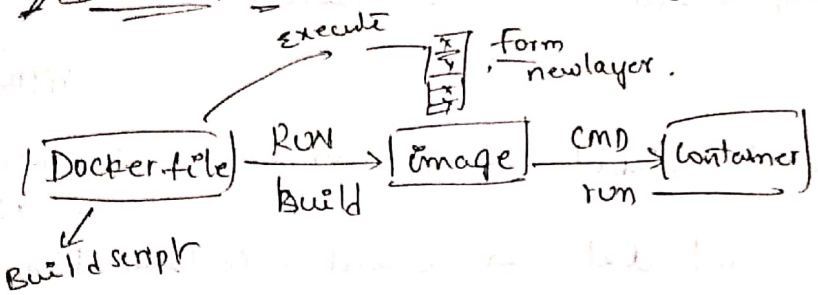
docker run -it -v/root/5:/5 centos

docker run -it -v/root/sai:/chand ubuntu

Host machine will maintain the Volumes in this Volumes have files

→ Container is also add the files to volumes, these files Stored to all Containers

Docker file:→ # (create an own image)



"Container linking"

→ Pull Jenkins image.

→ docker run -itd --name myjenkins Jenkins

→ docker run -itd. --name myubuntu --link myjenk ubuntu

→ docker attach myubuntu

→ Run "env" command, you will notice new variables for linking with the Source.

"Node.js" → is a java script framework that is used for developing Server -Side applications

ENV Key Value

Docker

1. Port Mapping.

(1) Given multiple ports to Single Container

docker run -it -p 1111:8080 -p 2222:80 ubuntu

1111 : tomcat
2222 : apache
ubuntu (have Tomcat & apache)

(2)

Jenkins

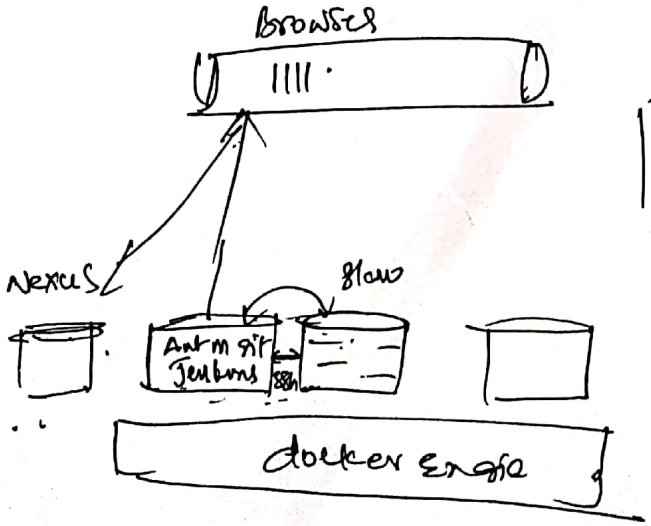
Install Suggested plugins
Folders
Time Stamper
Pipeline
git
PAM Authentication

OWASP Markup Formatter
Workspace cleanup
Github branch Source
Subversion
LDAP

Build Timeout
Ant
Pipeline: Github groovy, Libraries
SSH Slaves
Email Extension

Credential Binding
Gradle.
Pipeline: Stage View
Matrix authorization Strategy
Mailer

Compile
↓ down
Package
↓ post bar
deploy



Install Java on Ubuntu

```
apt-get install openjdk-8-jdk
                              -y
```

1. "Devops" :→ Devops is Combination of "Development" & "Operation"

→ The Collaboration & Communication of both Software developers & other IT professionals.

→ It focusing the delivering s/f product faster & covering the failure rate of release

—°—

Fundamental Difference b/w Devops & Agile :→

Devops :→	Agile
1. Agility in both development & ops	1. Agility in only development
2. Involves process such as CI, CD	2. Involves practices such as
3. Timeliness & quality have equal priority	3. Timeliness have main priority
4. Smaller releases cycle with immediate feedback	4. Smaller releases
Feedback is from self	from customers

3. What is the need for Devops?? :→

(1) Increase deployment frequency ✓ (Continuous s/f delivery)

(2) lower failure rate of new releases ✓

(3) Shortened lead time b/w fixes ✓

Devops tools:

Git/SVN/ ⇒ SCM tools

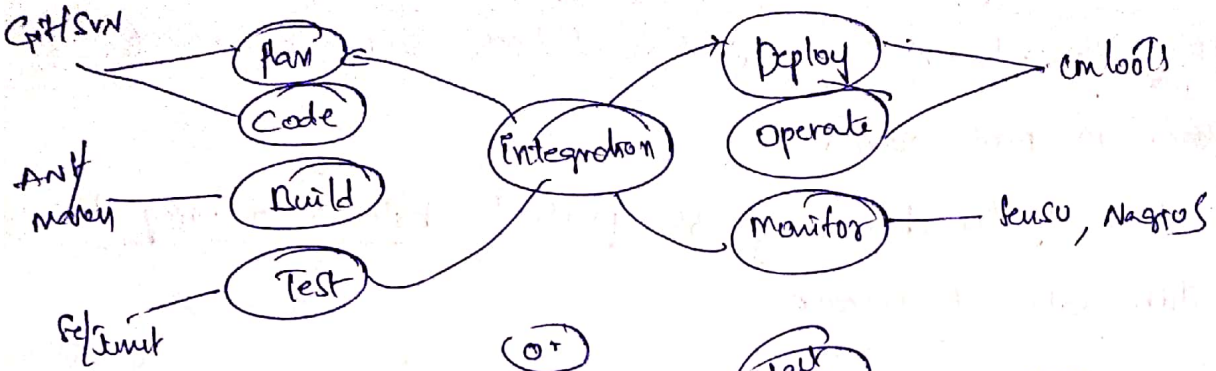
Ant/maven → Build tools

Jenkins → CI/CD tool

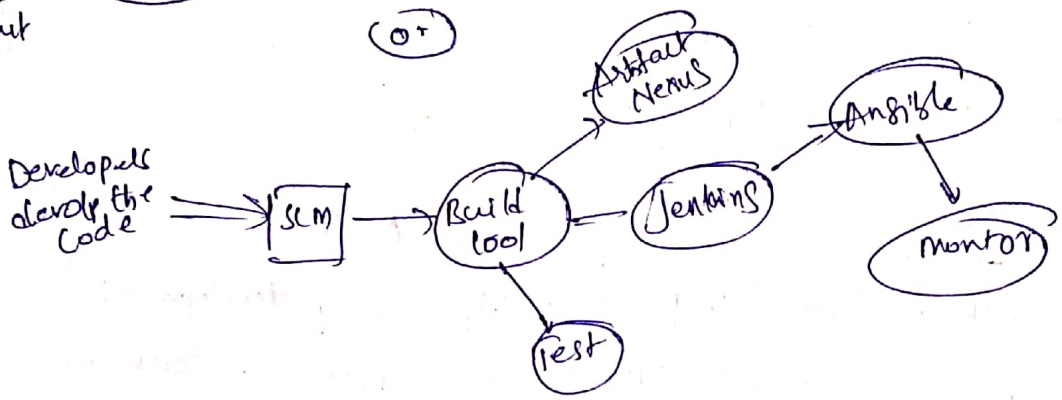
Chef → CMS deployment tool

Docker → Containerisation tool.

⑤ How does all these tools work together.



(or)



⑥ What are the Advantages of Devops!→

Technical benefits:→

→ Continous s/w delivery
→ less Complex problems to fix
→ Faster resolution of problems.

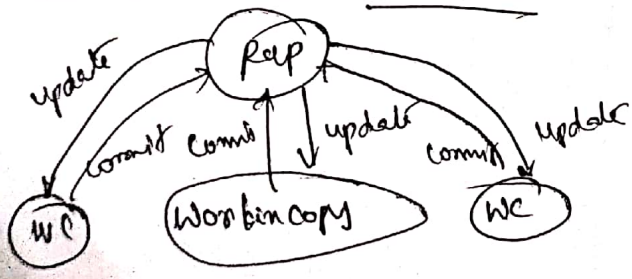
Business benfits:→

⑦ What is the most important thing Devops helps us to acheive?

Clearer Communication & better working relationships b/w Dev teams & Ops teams ∪ will get good product

(SVN)

Centralised version Control System



Svn Checkout url
svn add filename
svn commit -m "message"
Svn update
svn copy -m trunk/ tag/tagname
svn copy -m trunk / branch /branchname
svn merge -- /branch/ branchname /

Agile Software development process:

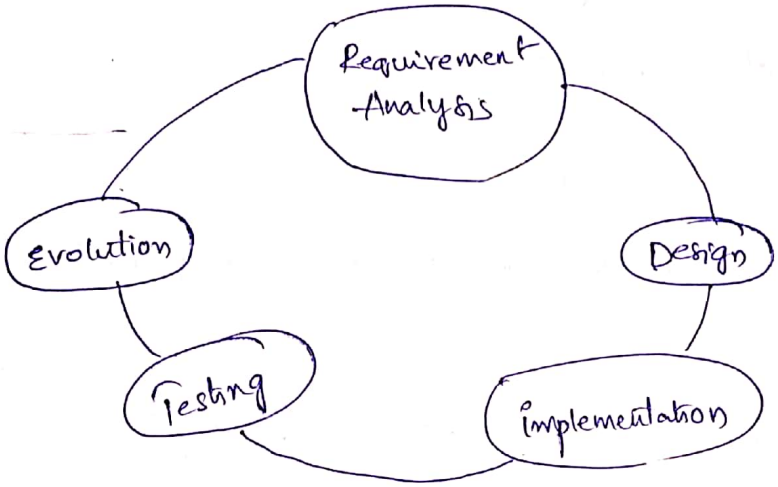
Agile → Quick & easy.

Agile is a collection of Software development methodologies in which Software is developed through Collobration among self-organised teams.

Software development process (or) (SDLC)

Software development life cycle!→

→ It is the process of planning, developing, Testing & deploying Software, Team follow a sequence of phases, and Each phase uses the outcome of the previous phase



Continous integration,→

Continous integration is a Software development practice where developers frequently integrate their work with the projects integration Branch and Create a Build.

Devops :→

Devops stands for "Development", Operations, and the people who manage these operations are called Devops engineers

Development Operations:

- → Build & Release management
- → Deployment management
- → Version Control system administration
- → Software Configuration management
- → All sorts of Automation
- → Implementing Continous Integration
- → Implementing Continous Testing
- → Implementing Continous delivery
- → Implementing Continous deployment
- → Cloud management & Virtualization.

Version Control System (or) Revision Control System:-

This is the most basic & most important requirement to implement Continous integration

- → It is a tool used to "manage ur code history"
- → A version control tool offers many features such as labelling, branching

VCS

"Two of the famous VCS" are ----

- CVCS (Centralised version control system) — SVN (Subversion)
- DVCS (Distributed version Control System) — (GIT)

→ "VCS is a tool used to record changes made to a file (or) set of files over time

Advantage:

You can recall specific version of file (or) a set of files

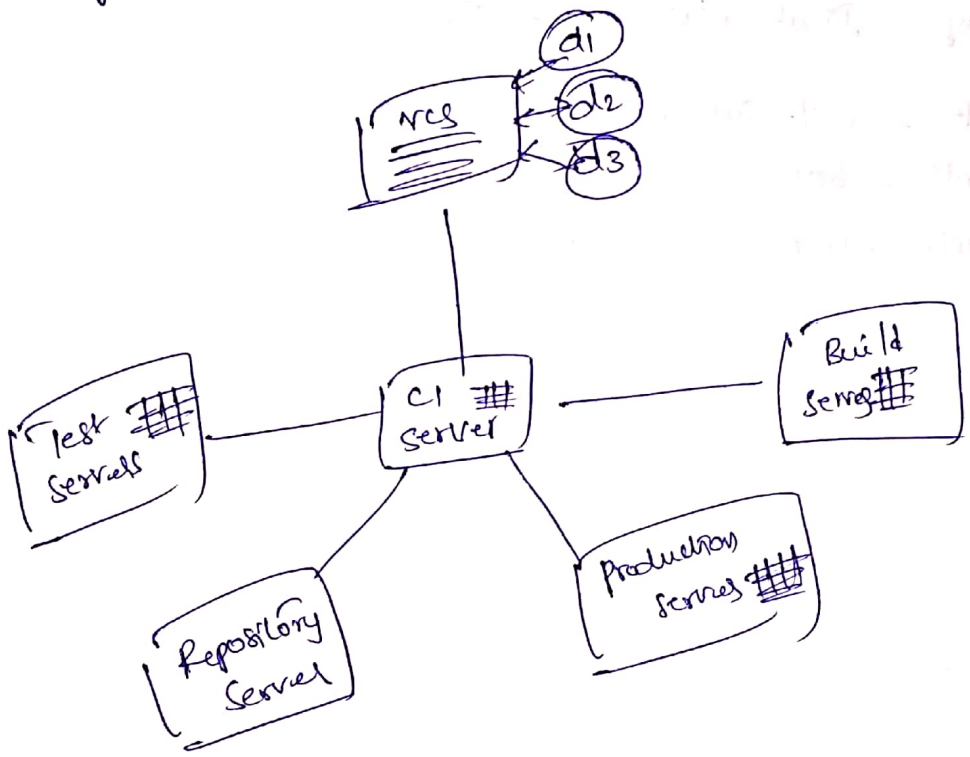
Repository Tools:→

A Repository tool is a Version Control System for "binary files"
(.rar, .war, .exe →)

→ It record what, when, and who created a build package

Continous Integration tool:→

Continous Integration tool is the centre of Continous integ
-ration system and is connected to VCS tool, Build tool, repository,
testing and production environments, quality analysis tools.



Maven:→

→ Maven is a build tool used mostly to complie Java code.

→ It uses JAVA libraries

→ The code to be built is described using an 'XML' file
(Extensible markup language)

Jenkins

→ Jenkins is a Opensource CI/CD tool

→ The Jenkins tool is written in "Java"

→ Community-based support

→ lot of plugins
→ Jenkins has a Cloud support

Running Jenkins inside a Container :→

→ Glass Fish, JBoss, Jetty, Jonas, liberty profile, Tomcat, weblogic

Jenkins Job Status :

Green — Build Success
Red — Failure
Gray — Disabled / Never executed

Sunny represent — 100% Success rate
Cloudy represents — 60% Success rate
Rainy represents — 40% Success rate