

DevOps Final Prop ->

Unit 1 Important Questions ->

Q-1 Define Devops & name various devops tools.

=> (i) Devops is a set of practices that combines software development & IT operations. It aims to shorten the systems development life cycle & provide continuous delivery with high software quality. The goals of devops are achieved by creating a set of practices & patterns around 4 main pillars.

① Communication -> To create better software & do better releases, the team needs to constantly communicate with each other from the planning to release phase to solve impediments & improve the quality of the release process.

② Collaboration ->

② Collaboration -> Collaboration is key for a true devops approach. Better collaboration between the dev & ops team help them achieve a streamlined process.

③ Automation -> Automation is the heart of every successful devops transformation process & the single biggest enabler for devops. The success of devops lies in stable environment, consistent build & test process & happy release.

④ Monitoring -> Monitoring in devops teams are required to provide critical information that ensure service uptime & optimal performance.

(ii) Different tools for devops →

① Version Control Tools →

- (i) GitHub
- (ii) Bit bucket

② Container Management Tools →

- (i) Docker
- (ii) Kubernetes

③ Deployment & Service Monitoring Tools →

- (i) Splunk
- (ii) Data dog

④ CI / Deployment Automation Tools →

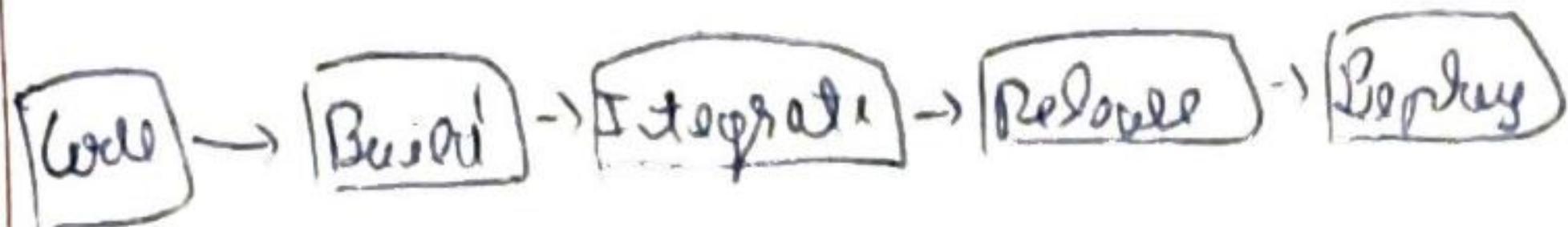
- (i) Jenkins
- (ii) Bamboo

Q 2 what is continuous Integration & continuous Deployment
and continuous delivery?

⇒ Continuous Integration →

- Developers practicing continuous integration merge their changes back to the main branch as often as possible.

- The developer's changes are validated by creating a build & running automated tests against the build.



PAGE:

DATE: / /

- Write → Build → Integrate → Release → Deploy
- Write → Build → Integrate → Release → Deploy
with pull request
continuous delivery
cont. deployment
- By doing so, you avoid integration challenges that can happen when waiting for release day to merge changes into the release branch.

- Continuous integration puts a great emphasis on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.

- Continuous Delivery →

- Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing &/or production environment after the build stage.

- This means that on top of automated testing, you have an automated release process & you can deploy your application any time by clicking a button.

- Continuous Deployment →

- Continuous deployment goes one step further than continuous delivery.

- With this practice, every change that passes all stages of your production pipeline is released to your customers.

- Only a failed test will prevent a new change to be deployed to production.

Q3 Explain various devops tools.

→ (1) Version Control Tools →

- Github: It is considered as one of the largest development platforms in the world. Its main features are collaborative coding, automation (CI/CD), project management.

(2) Container management Tools →

- Docker: It is a light weight tool that aims to simplify & accelerate workflow. A docker image includes everything needed to run an application.

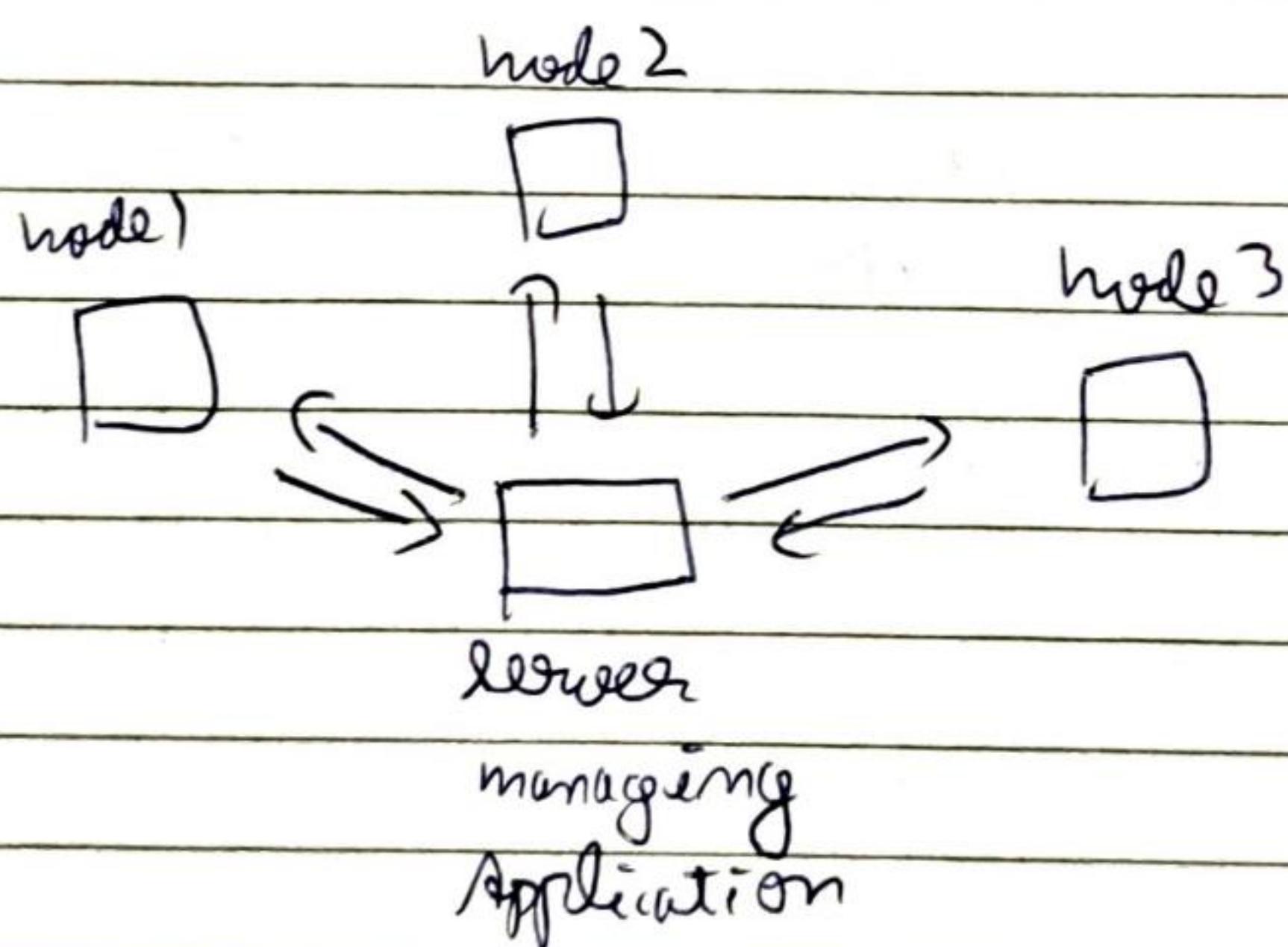
(3) CI/CD Automation Tools →

- Jenkins: Written in Java, Jenkins is an open source platform for continuous integration & continuous delivery that is used to automate release management.

Q4

Q 4 What is distributed computing?

- ⇒ A distributed computer system consists of multiple software components that are on multiple computers, but act as a single system.
- The computers that are in a distributed system can be physically close together & connected by a local ~~social~~ network or connected by a wide area network.
 - The goal of distributed computing is to make a network act as a single computer.
 - The system can easily be expanded by adding more machines as needed.

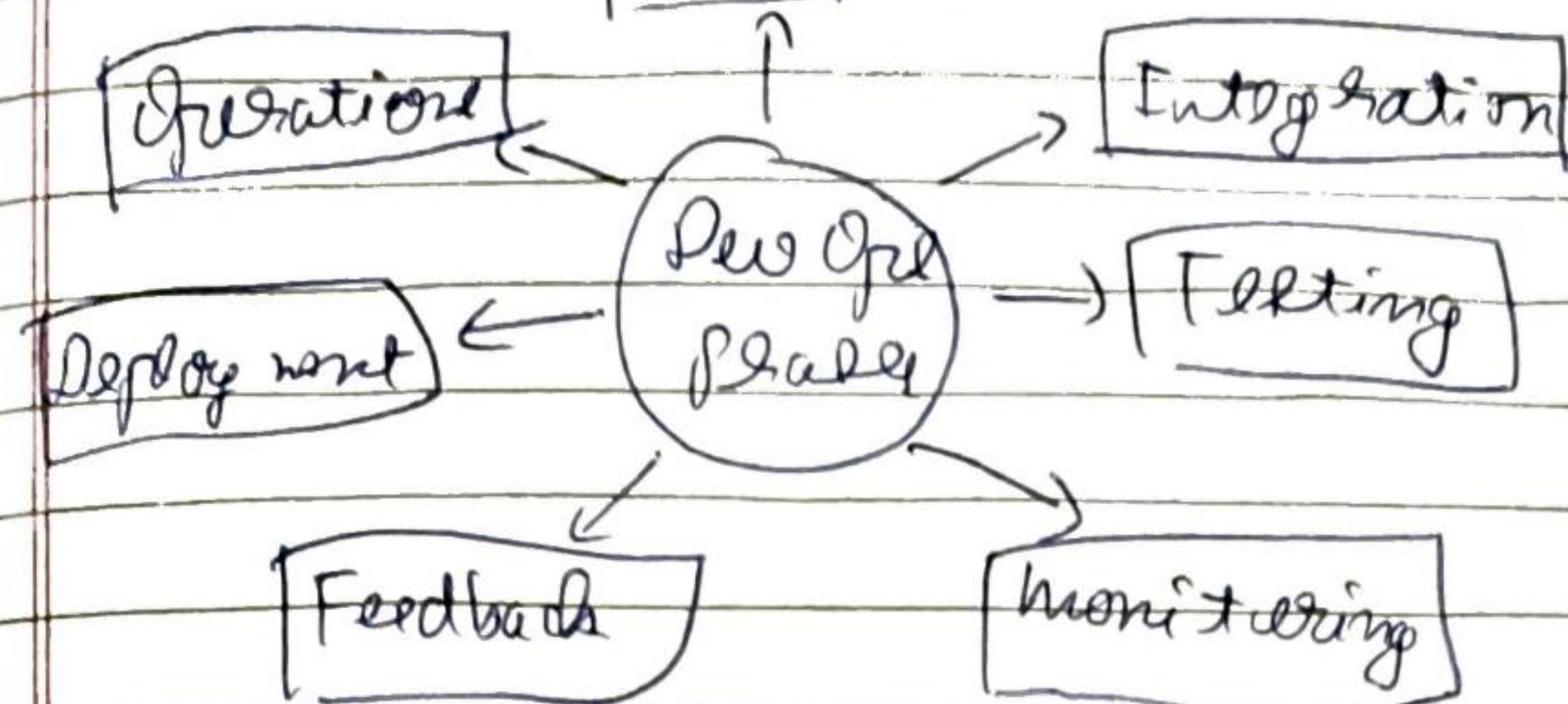


Q5

Explain the phases of devops

Development

=>



① Continuous Development -)

This phase involves the planning & coding of the software. The vision of the project is decided during the planning phase.

② Continuous Integration -)

It is a software development practice in which the developer's require to commit changes to the source code more frequently.

③ Continuous Testing -)

This is the place where the developed software is continuously tested for bugs.

④ Continuous Monitoring -)

monitoring is a place that involves all the operational factors of the entire process, where the important information about the use of software is recorded.

⑤ Continuous Feedback →

The application development is constantly improved by analyzing the results from the operations of the software.

⑥ Continuous Deployment →

In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.

⑦ Continuous Operations →

All devops operations are based on the continuity with complete automation of the release process & allows the organization to actually accelerate the overall time to market.

Q6 What is Unit Test & Integration Test?

⇒ Unit Testing →

① In unit testing, each module of the software is tested separately.

② In unit testing tester knows the internal design of the software.

③ Unit testing is performed first of all testing process.

Integration Testing →

① In integration testing, all modules of the software are tested combined.

② In integration testing tester doesn't know the internal design of the software.

③ Integration testing is performed after unit testing & before system testing.

- | | |
|---|--|
| ④ Unit testing is performed by the developer. | ⑦ Integration testing is performed by the tester. |
| ⑤ Detection of bugs in unit testing is easy. | ⑧ Detection of bugs in integration testing is difficult. |
| ⑥ Unit testing is less costly. | ⑨ Integration testing is more costly. |

Q 7 Explain implementation of devops with example.

Implementation of Devops

- ① Introduce Devops Initiative →
The company's CTO organizes a devops initiative as a part of the IT department program.
- ② Develop Devops Strategy →
To best practice start that enhance team collaboration & facilitate new way of infrastructure.
- ③ Use container Containerization →
Containerization implemented with a tool like Docker solves the problem with reliability of software.
- ④ Integrate Automation with CI CD tools →
By integrating automation tools such as Jenkins, can address configuration management concern & effective deployment.

⑤ Increase the amount of Test automation →
To achieve better delivery cycle, test automation
should be increased

⑥ Application performance monitoring →
Application performance monitoring helps in
detecting, prioritizing & isolating application
defects.

• Examples of defects in real life →

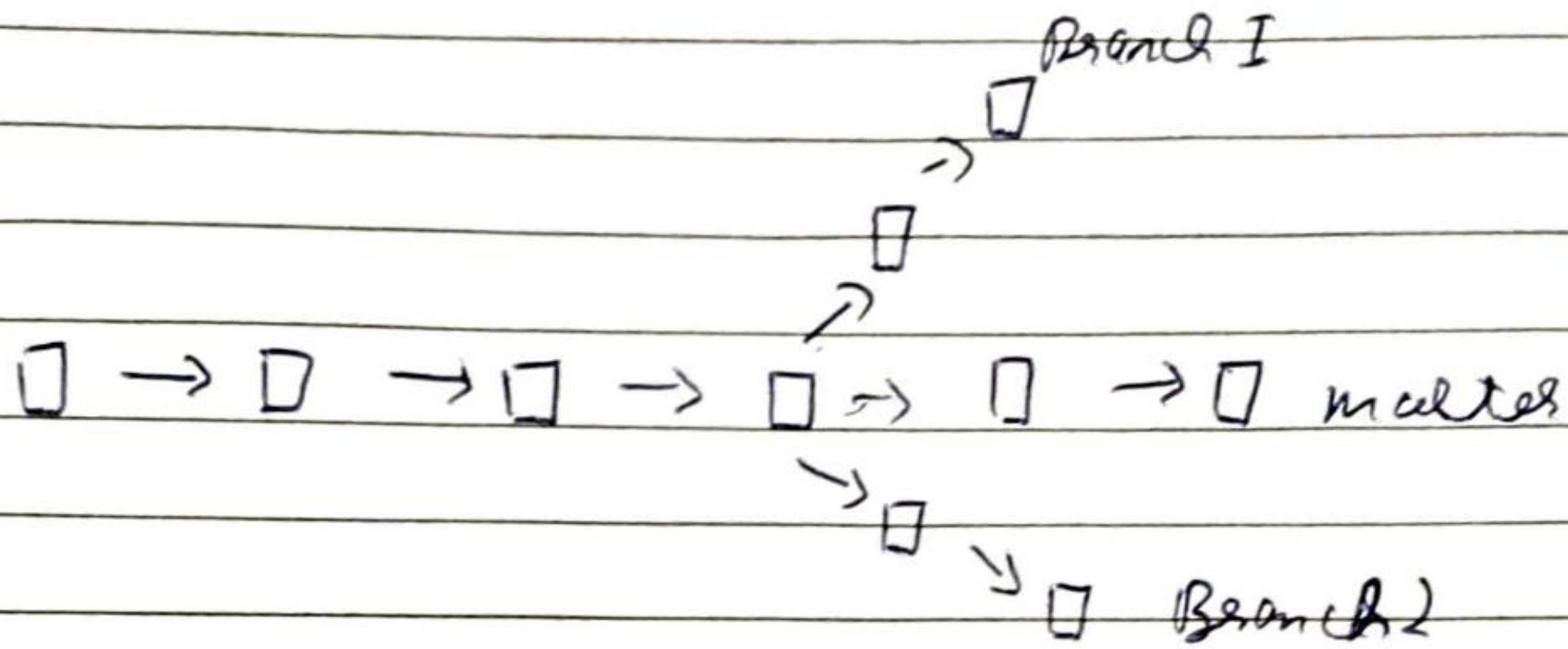
- ① Online Financial Trading
- ② Car manufacturing Industries
- ③ Airlines Industry
- ④ Bug Reduction
- ⑤ Automation Testing -

Unit 2 Important Questions -

Q I

What is branching? Why do we need branching?
How can we create branches from one & how can we switch one? How to delete branches

⇒ (i) Branching → A branch is a version of repository that diverges from the main working project. A git project can have more than one branch. These branches are a pointer to a snapshot of your code.



(ii) need of Branching -

- ① Git branching allows developers to diverge from the production version of code to fix a bug or add a feature.
- ② Developers create branches to work with a copy of the code without modifying the existing version.

(iii) Create branch -

- ① To create a new branch & switch automatically to that branch →

git checkout -b <branch-name>

(i) Create branch without switching to new branch -

git branch <branch-name>

(ii) Pull local branch to remote -

git push -u origin <branch-name>

(iv) Delete a branch -

(i) Delete branch locally -

git branch -d <branch>

(ii) Delete a remote branch -

git push <remote> --delete <branch>

Q2 What is git & its four git commands?

- (1) Git is an open source distributed version control system.
- (2) It is designed to handle minor to major projects with high speed & efficiency.
- (3) It is developed to co-ordinate the work among the developers.

② The version control allows us to track & work together with our team members at the same workplace.

• The four git commands ->

① git config ->

This command is a convenience function that is used to set git configuration values on a global or local project level.

② git init ->

This command creates a new git repository. Executing this command creates a git subdirectory in the current working directory.

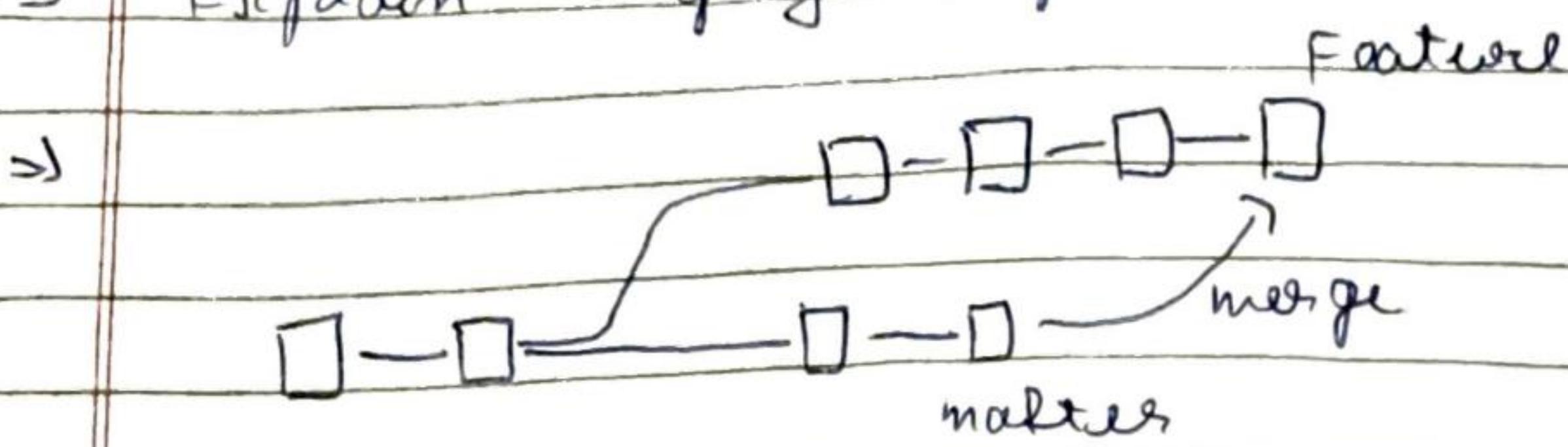
③ git commit ->

This command captures a snapshot of the project currently staged changes.

④ git status ->

This command displays the state of the working directory & the staging area.

Q 3 Explain merging in git :-



- ① In git, the merging is a procedure to connect the branch history.
- ② It joins two or more development history together.
- ③ The git merge command take the data created by git branch & integrate them into a single branch.
- ④ Git merge is used to combine two branches.
- ⑤ Command used is git merge <commit>.

Q 4 Three types of shell .

→ ① Bash shell →

Bash stands for Bourne Again Shell & is the default shell on many Linux distributions. Its features are:-

- (i) Command line editing
- (ii) Job control
- (iii) Unlimited size command history

② Tcsh / csh shell ->

Tcsh is advanced C shell, it can be used as interactive login shell & shell script command processor. Few features are:-

- (i) like syntax
- (ii) command line editor
- (iii) Job control

③ Ksh shell ->

Ksh stands for Korn shell. It is a powerful, high level programming language.

Q5 Diff b/w git revert & git reset

→ git revert ->

- ① Removed the commits from the remote repository.

- ② Used in remote repository.

- ③ Add a new commit to the existing commit history.

- ④ Roll back the changes which we have committed to.

git reset ->

- ① Unstage a file & bring our changes back to the working directory.

- ② Used in local repository

- ③ Alter the existing commit history

- ④ Discards the uncommitted changes.

(S) Do not manipulate your commits or files.

(S) Can be used to manipulate commits or files.

(Q6) Explain steps involved in removing a file from git index without removing from the local file system.

⇒ There are 2 ways of doing this ->

- (i) Using .gitignore file.
- (ii) Using git rm --cached command

(i) ① Go to your git directory.

② In the command prompt, enter the command touch .gitignore.

③ Now open the .gitignore file with your favorite text editor.

④ Add the name of the file you want to remove from the ~~staging~~ git index.

⑤ Now make a new commit & the file will be removed from the git index but not from the local storage.

(ii) ① Go to your git directory.

② Open the command prompt & enter the command → git rm --cached 'filename'

⑦ This command removes the file from the staging area (git index). The file from the working directory will remain intact.

Q7

What is init & why is it used. Explain the scope of init.

=> The git init command creates a new git repository. It can be used to convert an existing project to git repository.

② Executing git init creates a .git subdirectory in the current working directory, which contains all of the necessary git metadata.

③ These metadata can be categorized into object, info & temp files. It also initializes a HEAD pointer for the master branch of the repository.

Usage & Scope ->

① First change your directory to project folder

Usage & Scope ->

The git init command can be used to set up a new repository for new (Blank) project or for an existing project.

Create a repo for blank (new) project

- ① Open command line on the project directory & enter the command
git init
- ② Create a file with a command
touch (file name)
- ③ Add the file to staging area with command
git add .
- ④ Commit the change with the command
git commit -m "commit message"

Create a repo for existing project

- ① Open command line on the project directory & enter command
git init
- ② Add all the files to staging area with command
git add .
- ③ Commit all the changes -
git commit -m "commit message"

Q7

What is a version control system?

- Version control systems are a category of software tools that help in recording changes made to the file by keeping a track of modifications done in the code.
- They are useful for people from since they help them to reduce the development time.
- There are three types of version control system-

① Local Version Control System -

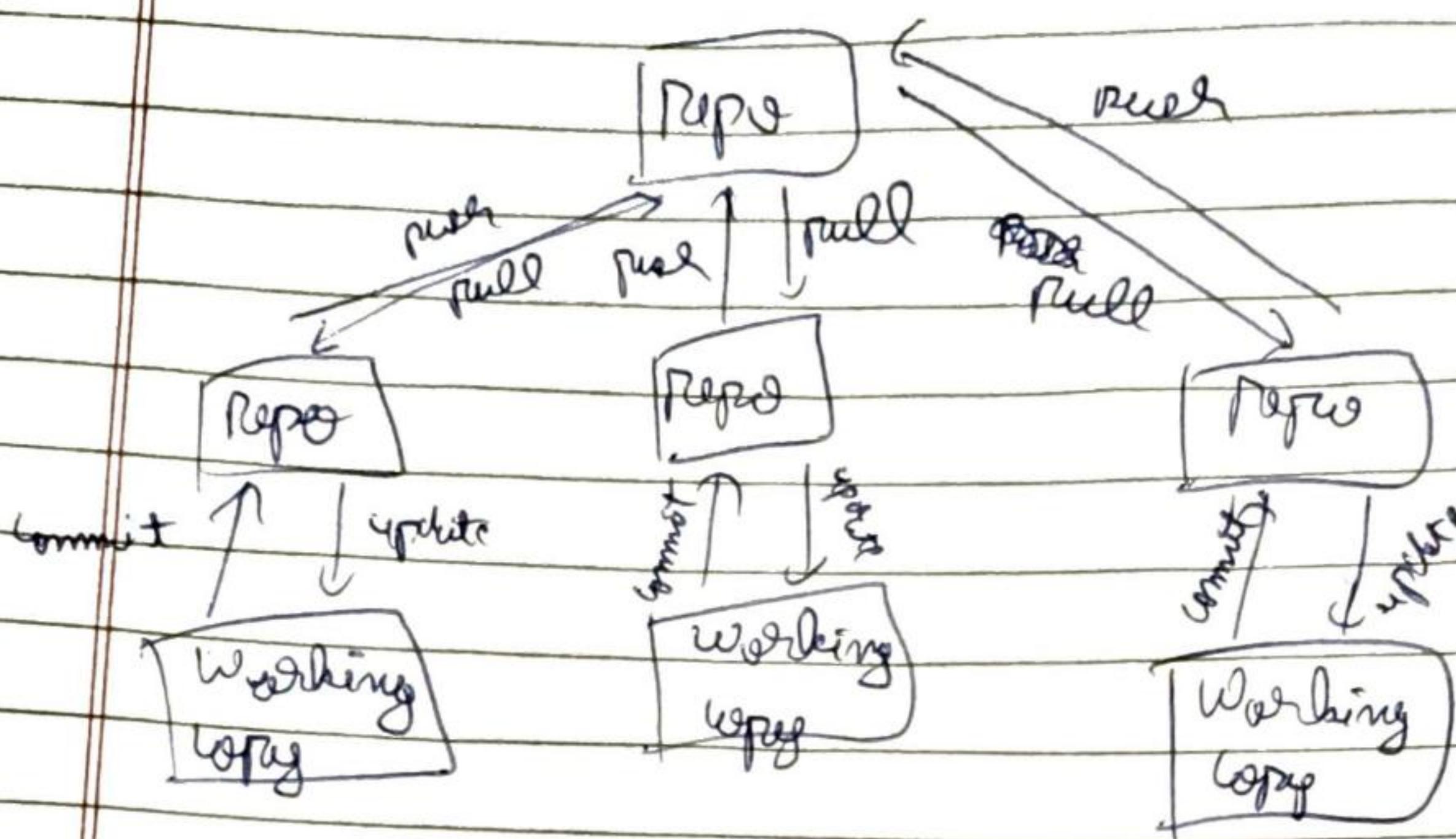
- It is one of the simplest form of local database that keep all the changes to files under version control.

② Centralized Version Control System -

- It contains just one repository globally. Every user need to commit for reflecting one's change in the repository.

③ Distributed Version Control System -

- It contains multiple repositories. Each user has their own repository & working copy. Just committing your changes will not give others access to your changes.



#8 What is a conflict?

- => Conflicts, also known as merge conflicts arise when two people have changed the same lines in a file, or if one developer deleted a file while another developer was ~~was~~ modifying it.
- In these cases, git cannot automatically determine what is correct.
- Conflicts only affect the developer conducting the merge, the rest of the team is unaware of the conflict. It will mark the file as being conflicted & halt the merging process.
- Types of merge conflicts →

i) git fails to start the merge →

A merge will fail to start when there are changes in either the working directory or

cloning area of the current project. Git will start the merge because those pending changes will be overwritten by the commits that are being merged in. The conflict happens with pending local changes.

i) git fail during the merge →

A failure during the merge indicates a conflict between the current local branch & the branch being merged. This indicates a conflict with another developer's code.

- git merge conflicts can be resolved with the help of commands git checkout, git stash, git commit or by changing the line of code due to which merge conflicts may occur.
- 'git reset' command can be used during a merge conflict to reset conflicted files to a known good state.

Q 9 What is a git repository?

- Repository in git contains a collection of files of various different versions of a project.
- These files are imported from the repository in the local server of the user for further updates & notifications in the content of the file.
- Two process of copying the content from an existing git repo ~~and~~ is called cloning. On the cloning of the repo the user gets the complete repository on his

local machine.

- User can also create a new repository or delete an existing repository. To create a new repo, create a (or init) new repo to from git hub. To ~~not~~ delete a repo, just drag the folder containing the repo.

Working with a Repo ->

(i) Adding to a Repo ->

- Add file to the staging area with command "git add .".

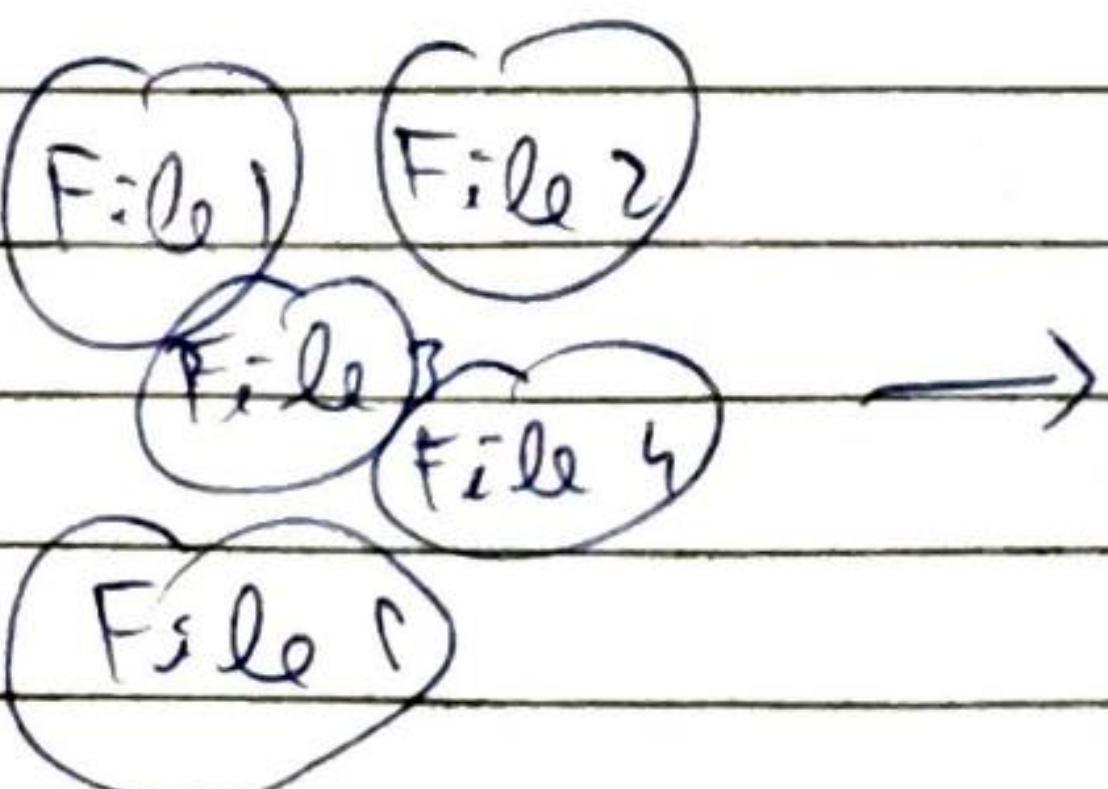
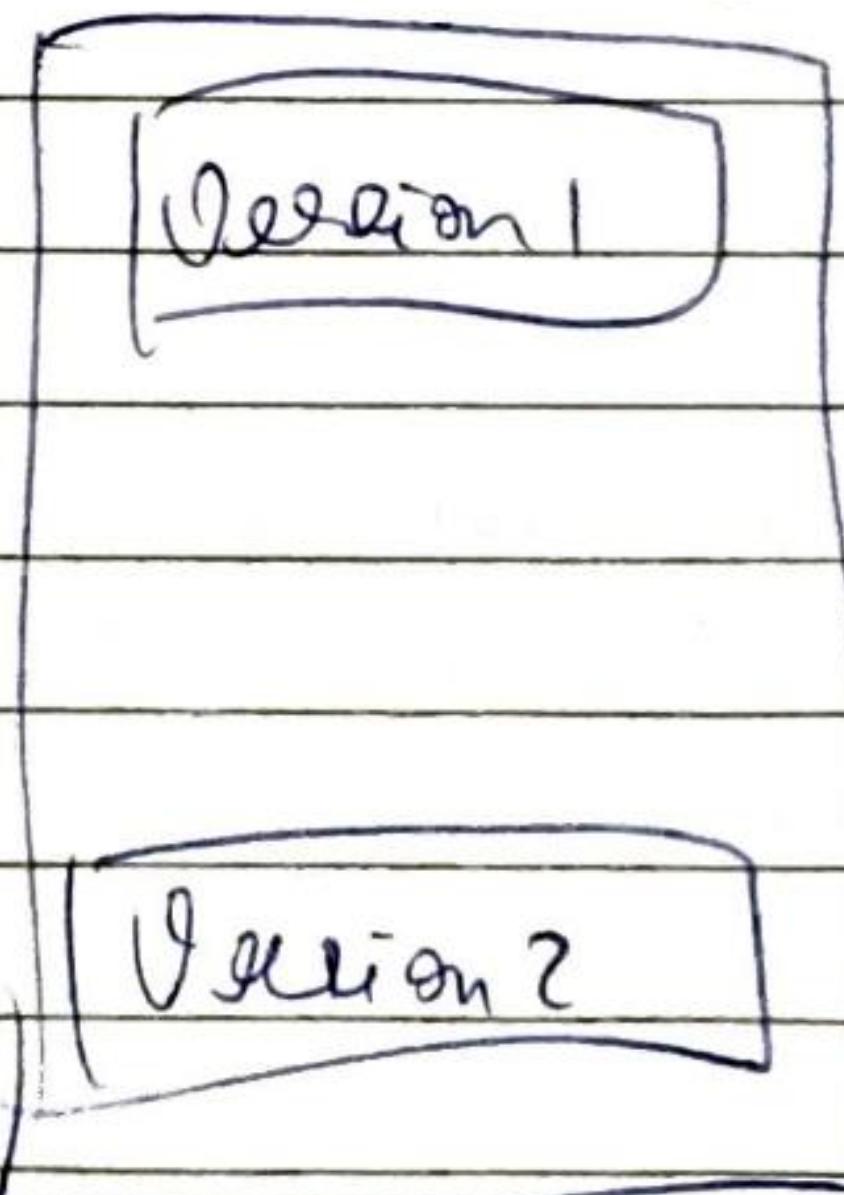
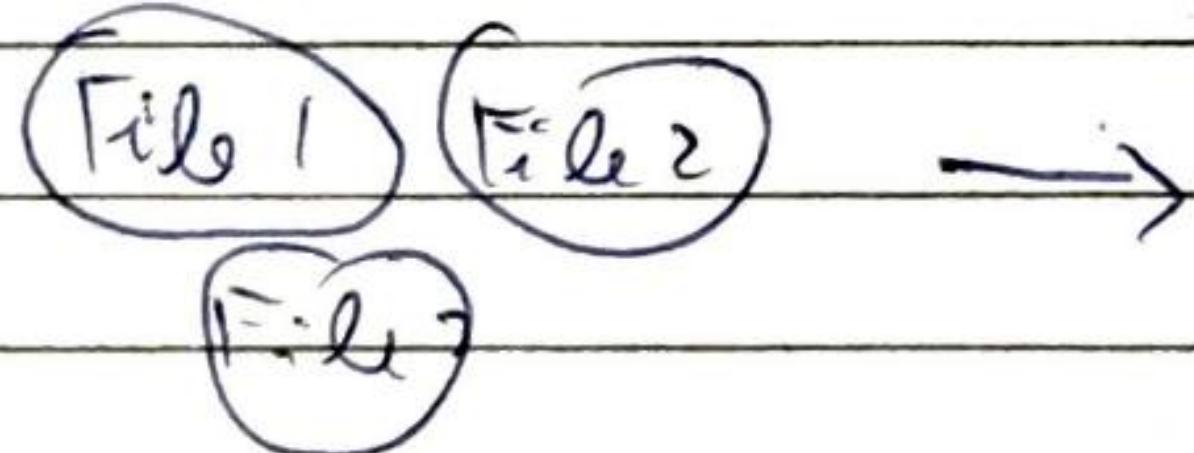
- Commit the change with ->

git commit -m "commit message"

(ii) Synchronizing with Remote Repositories ->

- The synchronization can be done by the command
git push -u origin master

Repository



Q10

What does git clone do?

- It's done primarily used to point to an existing repo & make a clone or copy of that repo.

The git clone command clones the repo through a format URL.

- Usually the original repository is located on a remote server, often from a git repo like GitHub. The remote repo URL is referred to as origin.

git clone examples

- Steps to clone a repo →

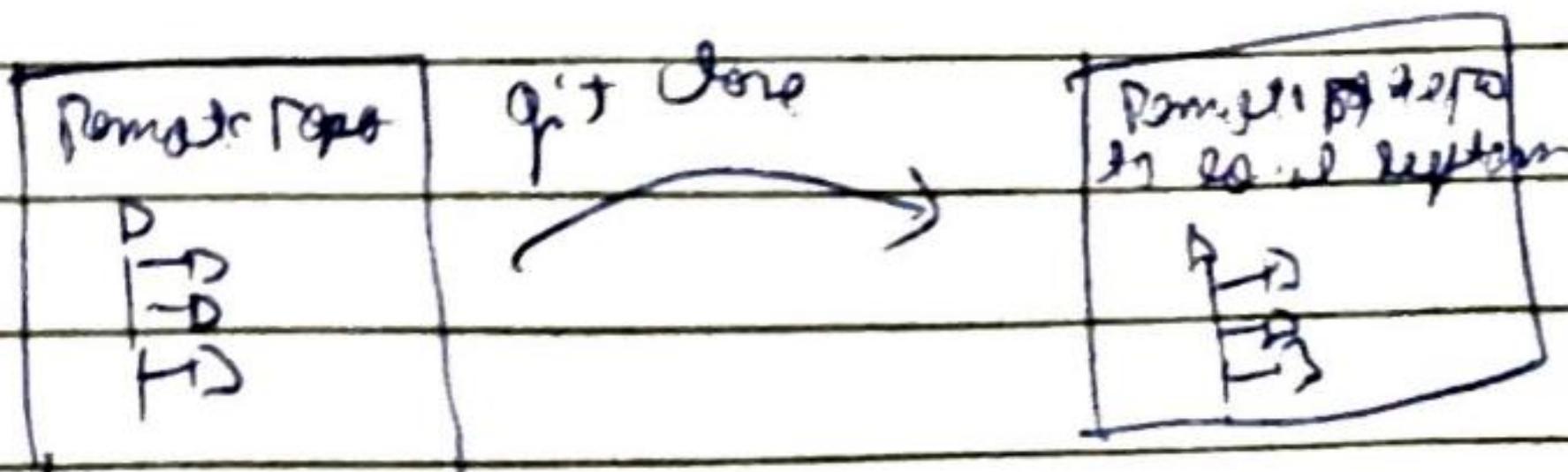
- ① Open GitHub & navigate to main page of the repo
- ② Under the repo name, click on Clone or Download
- ③ Select clone with HTTPS & copy the Repo URL
- ④ From cmd & run the command to your desired location & write the command →

git clone <repo URL>

- ⑤ The cloned repo will be created locally.

- Git also allows making a copy of only a particular branch off from a repo.

git clone -b <branch name> <Repo URL>

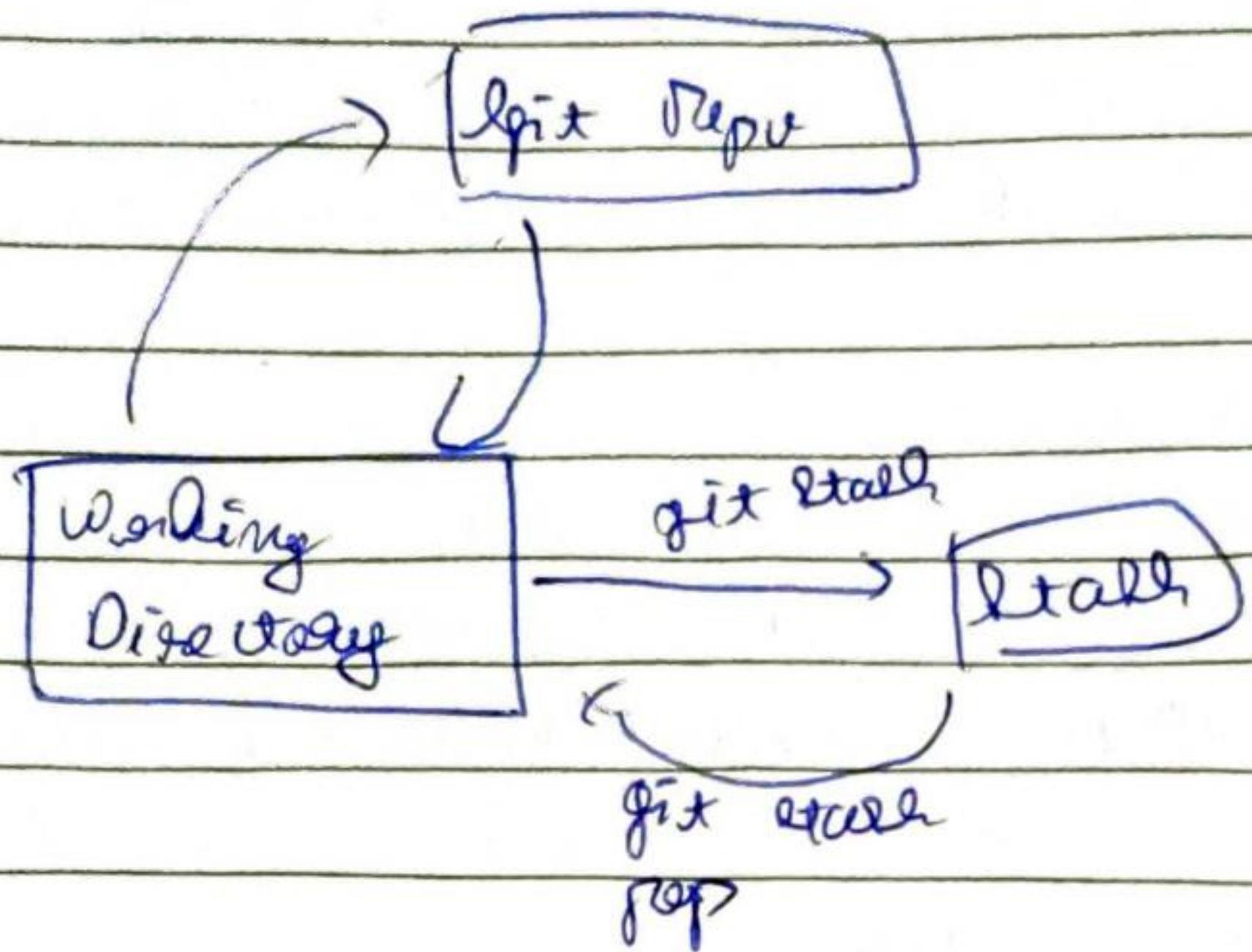


Q11

Explain git stash?

→ Sometimes there is need to switch

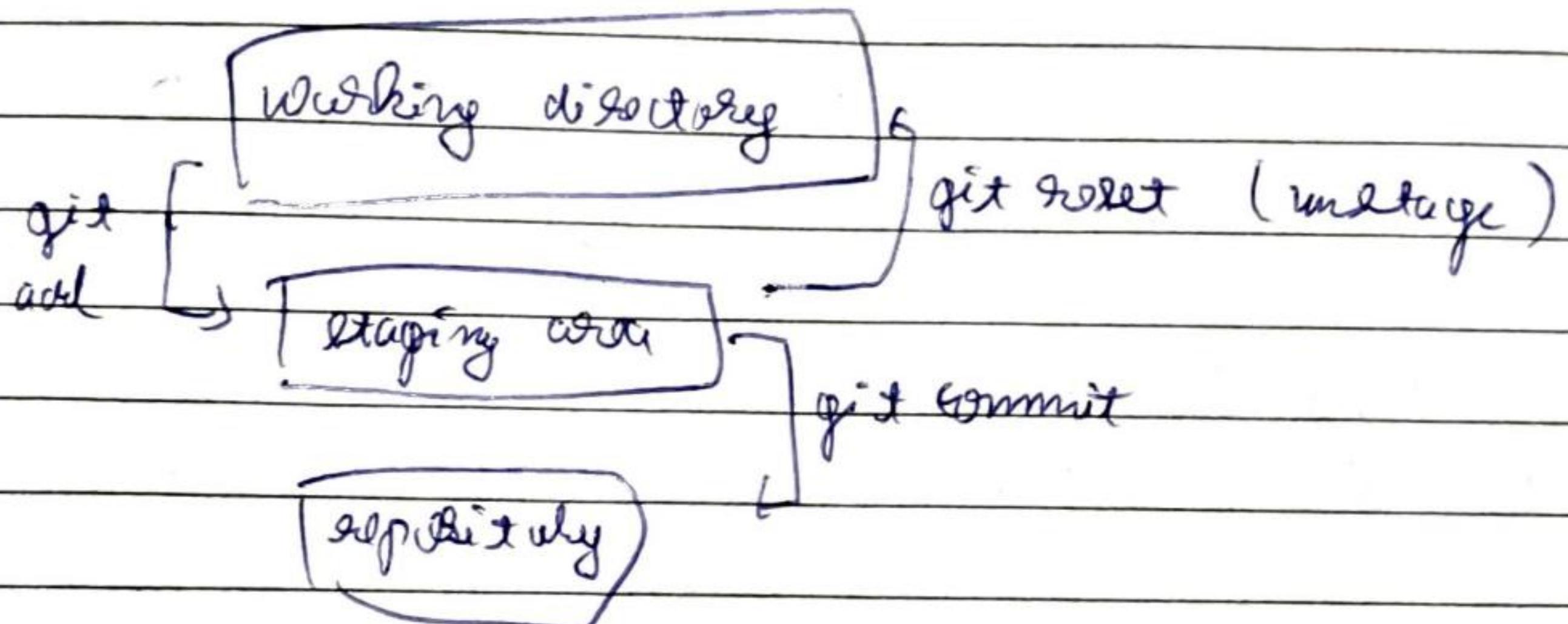
→



- Sometimes there is need to switch the branch, but the work on current branch is not complete
- The git stash command enables to switch branch without committing the current branch.
- Different stash commands -
 - git stash
To save the work in stash
 - git stash save "save message"
To save the work in stash with message
 - git stash list
To check all stored stashes

- git stash apply
Apply to the ~~last saved stash~~
- git stash apply <stash id>
Apply to ~~the saved stash~~ stash with id
- git stash drop
Drop the stash without applying it.
- git stash clear
~~Drop all the stashes in the list at once.~~

θ



Unit 3 Important Questions

Q1 What is docker? Explain its advantages?

- ⇒ Docker is an open source centralized platform designed to create, deploy & run applications.
- Docker uses containers or its hosts OS to run application rather than boating a whole virtual operating system.
- Container ensure that our application works in any environment like development, test or production.

Q2 What are the

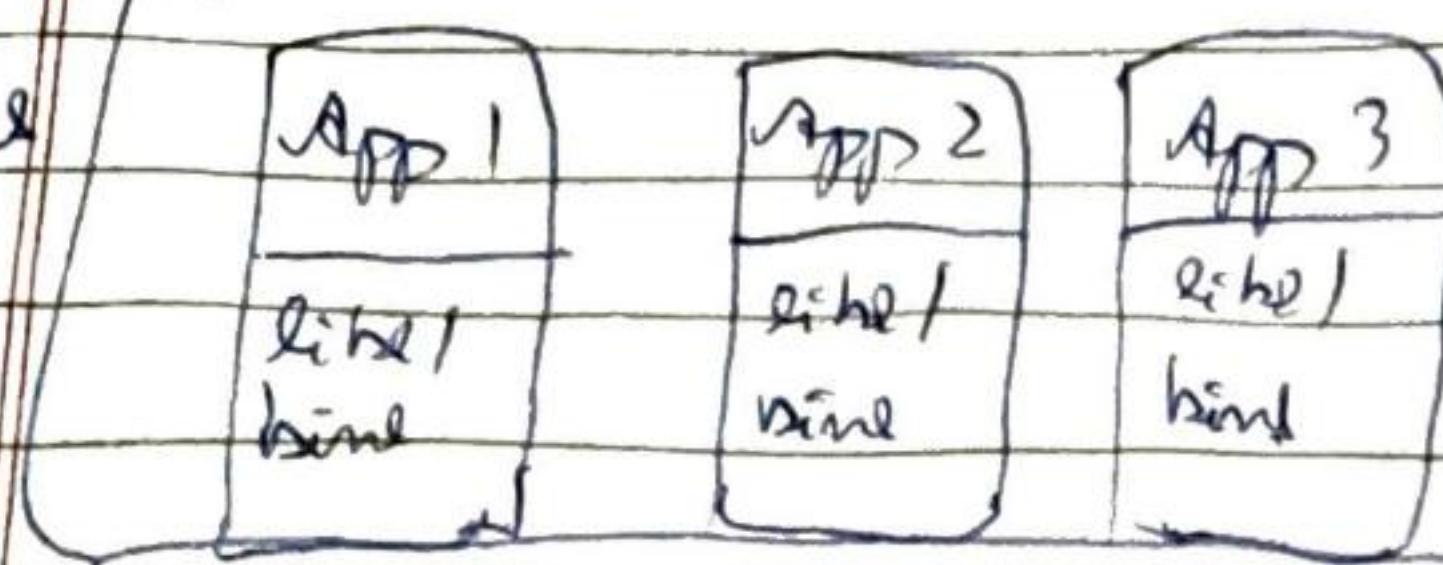
- Advantages of Docker →
 - It runs the container in seconds instead of minutes.
 - It uses less memory.
 - It provides lightweight virtualization
 - It does not require full OS to run applications
 - It uses application dependencies to reduce the risk
- Docker allows you to use a command to share your container with others
- It provides continuous deployment & testing environment.

Q

Define Containerization & its advantages?

- ⇒ Containerization is the technique of bringing virtualization to the OS level. While virtualization brings abstraction to the hardware, containerization brings abstraction to the OS.
- Containerization is however also a type of virtualization. However it is more efficient because there is no guest OS here & it utilizes host's OS, share common libraries & resources as & when needed.
- * Application specific binaries & libraries of containers run on the host kernel, which makes provisioning & migration very fast. Even booting up a container takes only a fraction of a second.
- * Because all OS containers share, host OS & takes only the application related binaries & libraries, they are lighter-weight than VMs.
- * Advantages -
- Containers on the same OS kernel are lighter & smaller.
- Better resource utilization compared to VMs.
- Boot-up process is light & takes a few seconds.

Container



Container Engine

Host OS

Guest Kernel

Containerization

Q Diff. b/w containerization & virtualization?

⇒

Virtualization →Containerization →

- | | |
|---|--|
| ① To provide complete isolation from the host OS & the other VMs. | ① It provides lightweight isolation from the host & other containers but doesn't provide as strong a security boundary as a VM. |
| ② It runs a complete OS including the kernel, thus requiring more system resources such as CPU, memory & storage. | ② It runs the user mode portion of an OS, & can be tailored to contain just the needed services for your app using fewer system resources. |
| ③ Runs just about any OS inside the VM. | ③ Runs on the same OS version as the host. |
| ④ Deploy one individual VM by using hypervisor software. | ④ Deploy individual containers by using Docker. |

⑤ Use a Virtual Hard Disk (vhd) Local Storage from a single VM

⑥ Use Shared disks from local storage (for a single node)

⑦ It uses virtual network adapter

① To use an isolated view of a virtual network adapter

② Virtualization can be achieved by using software like VM Ware

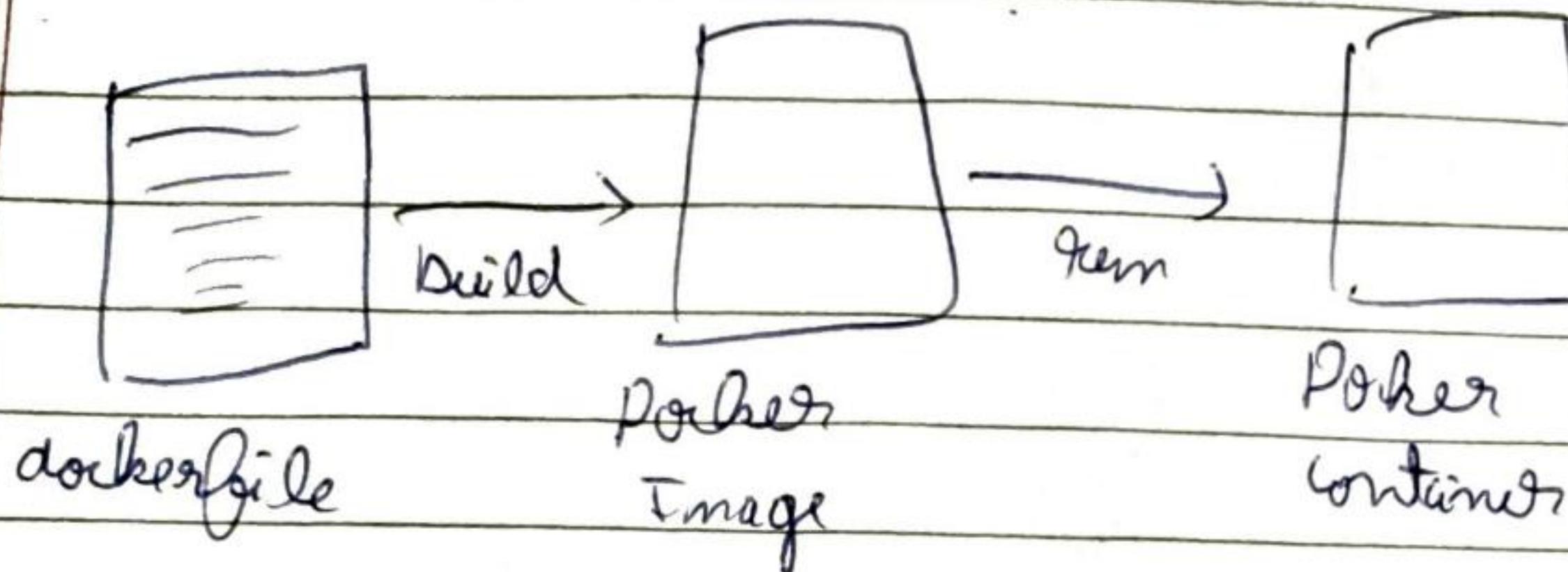
② Containerization can be achieved by using Docker

Q What are docker images?

- ⇒ A docker image is a file used to execute code in a docker container. Docker images act as a set of instructions to build a Docker container like a template.
- Docker images also act as the starting point when using Docker.
 - An image is comparable to a snapshot in VM environments.
 - A docker image contains application code, libraries, tools, dependencies & other files needed to make an application run.
 - Docker images have multiple layers, each one originates from the previous layer but is different from it.
 - The layers speed up docker build while increasing reusability & decreasing disk use.

use of docker image

- Docker image has everything needed to run a containerized application, including code, config files, environment variables, libraries & runtime.
- When the image is deployed to a Docker environment, it can be executed as a Docker container. The docker run command creates a container from a specific image.



Q How to create containers? Stop? Remove?

→ ① Install docker on the machine.

② Create your project. Create a folder on your computer which would contain 2 files

- (a) main.py - python file that contains code to be executed
- (b) dockerfile - file that contains the necessary instructions to create environment

③ Edit python file -

print ("Docker is magic")

⑤ Edit the docker file -

(a) From python interpreter

copy main.py

cmd ["python", "main.py"]

⑥ Create docker image -

(a) docker build -t python-test

⑦ Run docker image - One image is created, so is ready to be launched

(a) docker run python-test

II → Stop container -

① Stop a specific container -

docker stop [container name]

OR

docker kill [container name]

② Stop all running containers -

docker kill \$(docker ps -q)

III → Remove container -

To remove container, we need to stop it otherwise it will give error without stopping command.

~~rm command is used & rm command is used for removing one or more containers~~

- Deletes a specific container (only if stopped)

docker rm [container name]

- Forcefully deleting a container

docker rm -f [container name]

④ Docker container deployment (full application deployment)?

=> ① Install Docker on your machine

Docker makes it easy to run applications anywhere, but there needs to be Docker running. From the official website, download & install Docker.

② Set up a Registry at DockerHub

In order to easily distribute the application, you will need somewhere to share them. DockerHub is like the GitHub of Docker containers. Push your changes to your repository

③ Set up your Dockerized machine

The next step is to get Docker environment ready to run the app. Using command line & some tools, environment can be set

⑦ Create a Docker Image ->

Docker containers are created from Docker Images. To get your app distributed to different machines, you will need to create a docker image to send them. This is done using a Dockerfile.

Ex code for dockerfile ->

```
from python: latest  
copy main.py  
CMD ["python", "./main.py"]
```

⑧ Docker deployment of application ->

Pull your application to your DockerHub repo. Next pull your application from your repo to your other machine.

The 'docker run' command can be used to do all of automatically bind pull & run new application.

Q Docker Architecture -

⇒ Docker follows client-server architecture, which includes the three main components that are Docker Client, Docker Host/Server & Docker Registry.

- Docker Client -

Q What is docker client & docker server?

⇒ Two docker architecture uses client-server model.

- Docker Client -

It enables user to interact with docker. The docker client can reside on the same host as the daemon or connect to a daemon on a remote host. A docker client can communicate with more than one daemon. The docker client provides a command line interface (CLI) that allows you to issue build, run & stop application commands to a docker daemon. The main purpose of the docker client is to provide a name to direct the pull of image from a registry & to save it run on a docker host.

- Docker Server -

It provides complete environment to execute & run application. It consists of the docker daemon, images, containers, network & storage. The daemon is responsible for all container related actions & receives commands via the CLI or the REST API.

It can also communicate with other daemons to manage its service. The docker daemon pulls & builds container images as requested by the client. Once it pulls a requested image, it builds a working model for the container by utilizing a

Set of instructions known as a build file.

To start docker server -

Logon ->

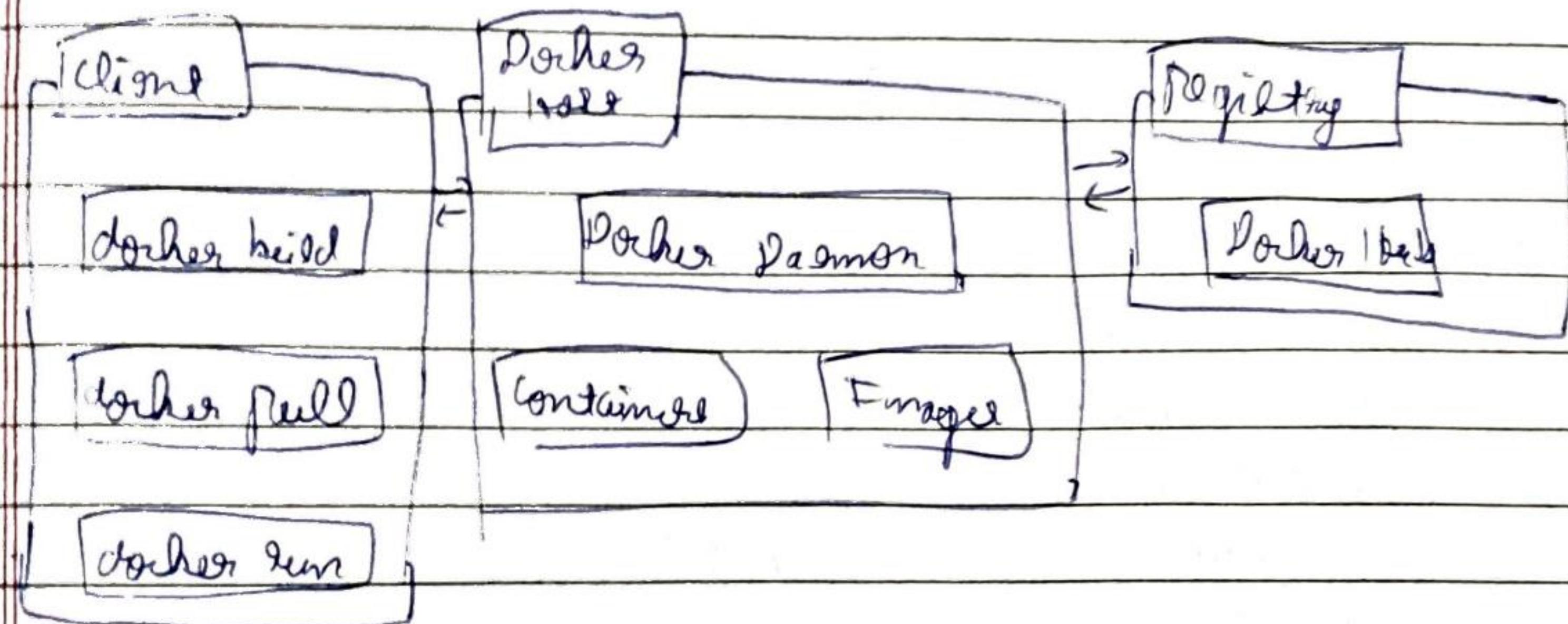
System & Start Docker

Docker Registry ->

Docker registry manages & stores the docker image. There are two type of registries in docker -

Public Registry - It is also called as Docker Hub

Private Registry - It is used to share images within the enterprise



Q

How do containers communicate with each other?

- => Although containers have a level of isolation from the environment around them, they often need to communicate with each other & the outside world.
- Two containers can talk to each other in two ways:-
 - (i) Communication through networking ->
Containers are designed to be isolated. But they can send & receive requests to other applications using networking. For example a web server container might expose a port so that it can receive requests on port on specified port. Or an application container might make a connection to a database container.
 - (ii) Sharing files on a disk ->
Some applications communicate by reading & writing files. These kinds of applications can communicate by writing their files into a volume, which can also be shared with other containers. For ex. a data processing application might write a file to a shared volume which contains customer data, which is then read by another application.
 - Multi-container based applications talk to each other using networking. This basically means that an application running in one container will create a network connection to a port on another container.

① build a custom image

→ ① build a hub container →

docker create -- name nginx-hub -p 80:80 nginx:alpine

② start container →

docker start nginx-hub

③ extract an image from a container →

docker commit nginx-hub

④ custom image is now created. Use this command →

docker image -a

Q

Pulling image to Dockerhub?

=> ① Login into dockerhub from terminal ->

docker login -- username = yourusername -- email = yourmail

Enter your password when prompted

② Check the image ID you want to pull -

docker images

③ Tag your image you want to pull -

docker tag <image ID> yourusername/repositoryname:tag

④ Pull your image to repo you created

docker pull yourusername/repositoryname

Q

What is persistent storage?

=> . Persistent storage refers to storage volumes usually associated with stateful applications such as databases that remain available beyond the life of individual containers.

. OpenStack Swift is a fully distributed storage solution that can be used for storing any form of static data or binary objects, such as media files, massive databases etc.

Q

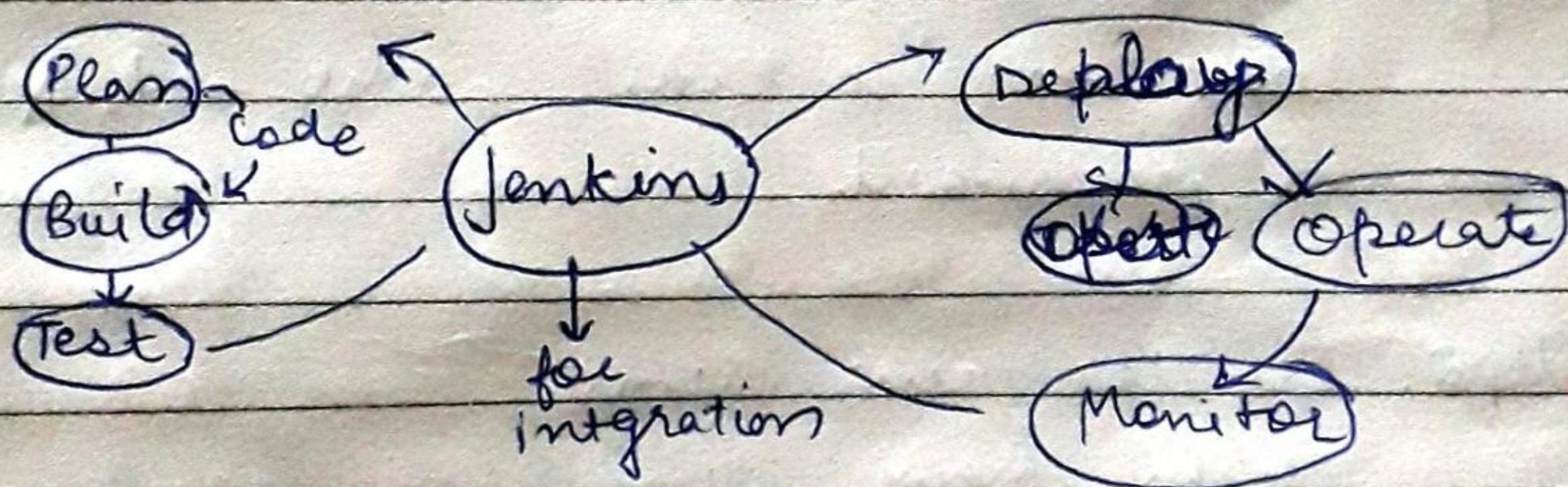
Persistent storage with docker?

->

The volume storage is also known as the persistent data storage layer of the docker. Volume is very useful to prevent data migration - if we are deleting the docker container but the volume is still present, then the same volume can be used for next container.

Q14. What is Jenkins? How to configure it? How to use it to build projects? How to link git jenkins to git repository?

Ans14: Jenkins is an open source automation tool used for continuous integration, written in Java with plugins. It is used to build and test your software projects continuously making it easy for developers to integrate changes to the project & making it easier for user to obtain a fresh built.



Steps to configure Jenkins:-

Step 1:- Navigate to Jenkins web interface and login as Admin.

Step 2:- Now go to 'manage jenkins' & click 'configure system'.

Step 3: Configure the root directory for workspace & build record.

Step 4: Set jenkins master executor to zero, setting up separate instances to be workers & jenkins will only be an Orchestrator.

Step 5: Set the environment variables.

Step 6: Now, configure installed plugins like GitHub or Slack etc.

Use of jenkins in Building Projects:-

Step 1: Login to jenkins:

Login to jenkins dashboard & add user & password.

Step 2: Create new item:

Click new item on your dashboard.

Step 3: Enter the item name, you want to create & select freestyle project > OK.

Step 4: Enter the details of project you want to test.

Step 5: Enter the repository URL, under the Source code management.

Step 6: Tweak the settings under the build section to build code.

Step 7: Click apply and save the project.

Step 8: Now build the source code.

Step 9: Check the status.

Step 10: Click on build no. and then click on console output to see the status of build you run.

Linking Jenkins to Git Repository:-

Step 1: Create a jenkins job:-

Go to Jenkins and select 'create new job' > click OK.

Step 2:- Configure Github hook trigger

go to build triggers & choose Github hook triggers.

Step 3:- Use a Github repository.

Go to pipeline section > pick Git > Type URL of Github repo > Type scriptPath(jenkinsfile)

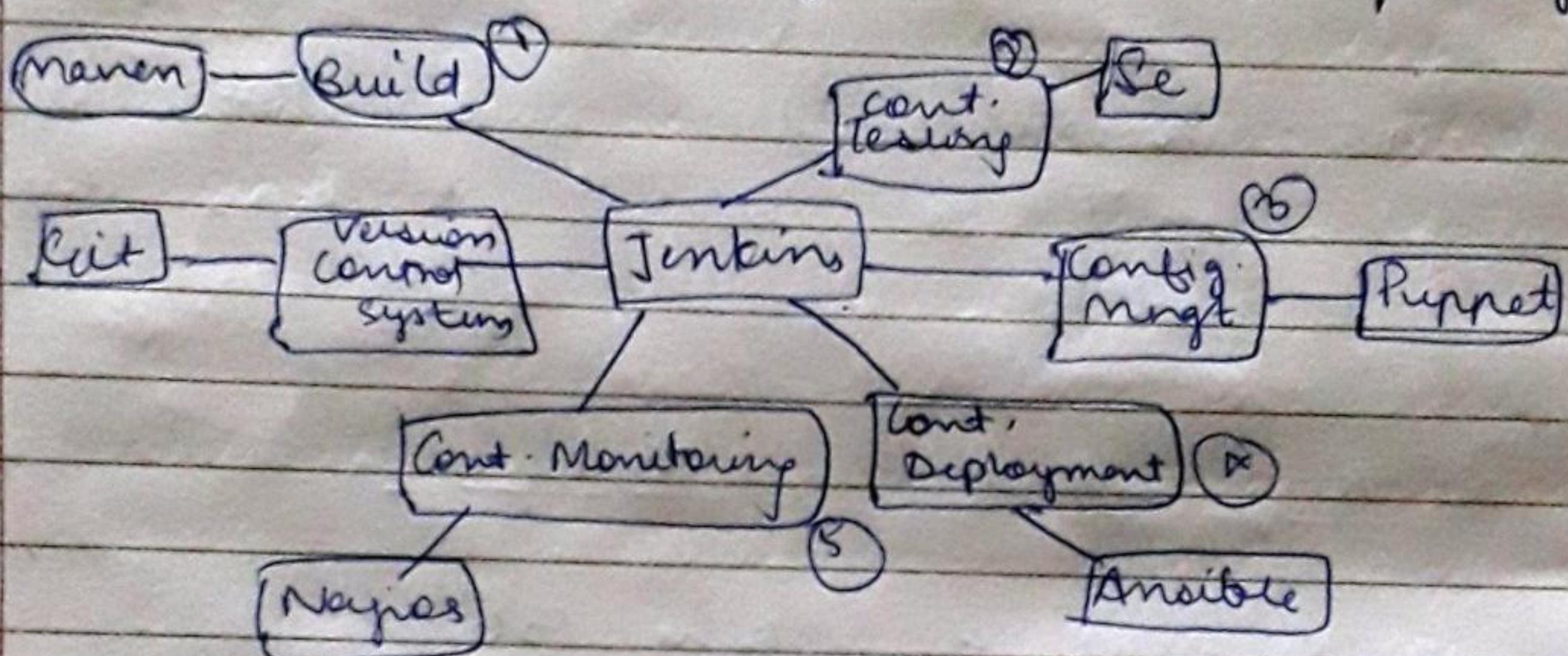
Step 4:- Add a webhook in Github.

Step 5:- go back to Jenkins & build a Jenkins job manually.

Step 6: Go to github repository, & change something, commit & push to check if integration works.

Step 7: Github repo is successfully integrated to jenkins.

Ques: How jenkins is integrated with devops stages?



Jenkins achieves CI with help of plugins.

Plugins allow integration of various devops stages.
For ex. Maven, git, Amazon EC2, htmlpublisher

Steps :- (1) When developers make changes in code, that changes are detected by jenkins

(2) These changes are sent to build scriptie Apache Maven.

(3) If builded code have no errors then it is sent to test environment, selenium.

(4) Then config. mgmt. is done with help of puppet tool.

- (5) If code passes all the test, it is ready for deployment (Ansible)
- (6) After deployment code is continuously monitored using the Nagios tool.

Dockerfile

Base Image → FROM dockerhub
Or use scratch

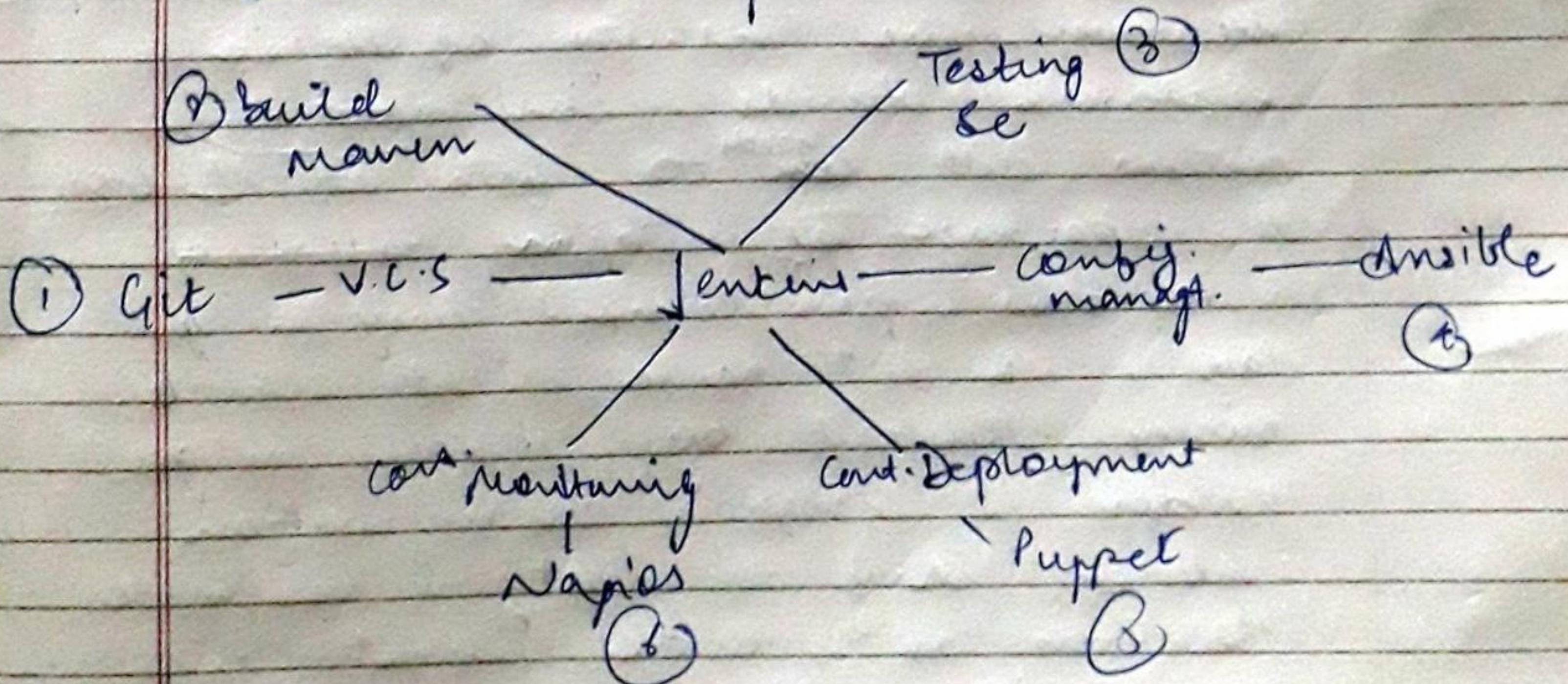
COPY 'main.py' to new image.

cmd - check 'dockefile path'
'cd <dockefile>'

build:- Image will be created with
docker build <dockefile>

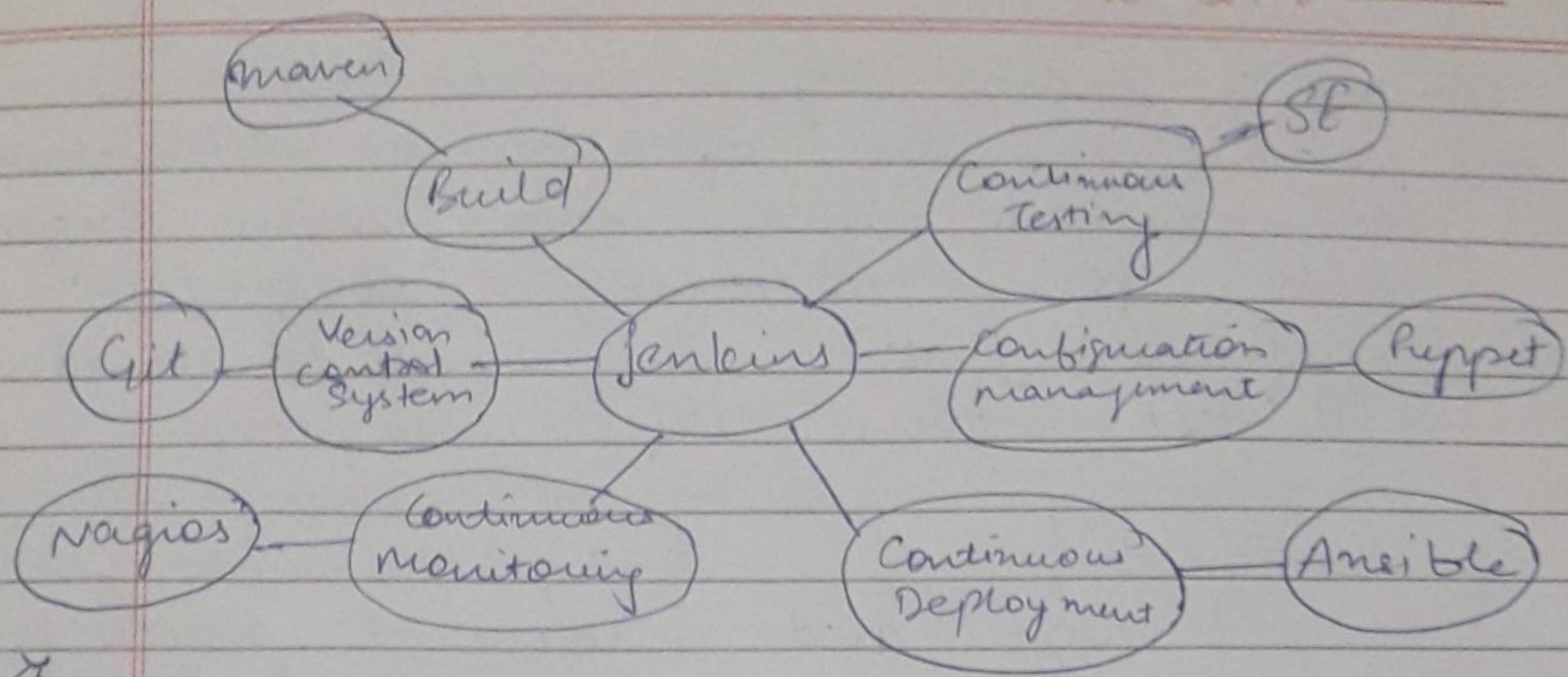
Docker RUN

docker run <image name>
or we can ^{docker} push <dockefile>



Q. what is Jenkins? why we use it?
Ans Jenkins is an open source automation tool written in java with plugins. built for continuous integration purposes. Jenkins is used to build & test your software projects continuously making it easy for developers to integrate changes to the project and making it easier for user to obtain a fresh build. It also allows you to continuously deliver your software by integrating with the large no. of testing & deployment technologies.
With Jenkins, organisations can automate the software development process through automation. Jenkins integrates software development lifecycle process of all types including build, test, document, package, stage, deployment, static analysis & much more.
Jenkins achieve C.I. with help of plugins. Plugins allow integration of various developer stages. If you want to integrate a particular tool, you need to install the plugin for that tool.

Example:- Git plugin, Maven, Amazon EC2, HTML Publisher



Q. How Jenkins is integrated with DevOps stages?

Q. Advantages of Jenkins:-

Ans - Open source

- Easy to Use
- Thousand Plugins
- Free of cost
- Build in Java
- It is secured
- Portable
- Platform Independent (bcz of Java)

Q. Why Jenkins is in high demand?

- Adoption
- Due to its plugins

Q. CI is a development practice in which developers are required to commit changes to the source code in a shared repository several

2 Day

times a day or more frequently. Every commit made in a repository is now built. This allows the team to detect the problem early. Apart from this dependency of the continuous integrator tool, there are several other options further like deploying the built application on test server providing the concerned team with a build & test result.

C.I. with Jenkins:

Let us imagine a scenario where the complete source code of an application was built & then deployed on test server for testing.

It sounds like a perfect way to develop software but few points have many flaws.

- Developers have to wait until the complete software is developed for test results.
- There is a high possibility that the test results might show multiple bugs. It is difficult for developers to locate those bugs because they have to check the entire source code of the application.
- It slows software delivery process.

- Continuous feedback was pertaining to things like coding or architectural issues, build failure, test status & file release upload was missing due to which the quality of software came down.
- The whole process was manual which increases risk of frequent failures.

C. How these problems are overcome with jenkins?

(1) developer commits the code to source code repository. Meanwhile jenkins server checks repository at regular intervals for changes.

(2) After a commit occurs the jenkins server detects the changes that have occurred in ~~soft~~ source code repository. Jenkins will pull those changes & start ~~mainly~~ preparing a new build.

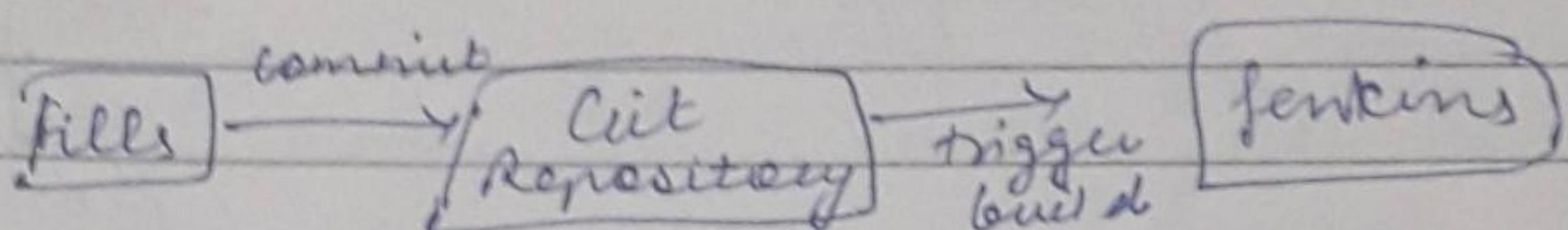
(3) When build fails, concerned team will be notified.

(4) After testing, Jenkins generates a feedback & they notify the developer about build & test results.

(5) It will continue to check source code repository for changes made

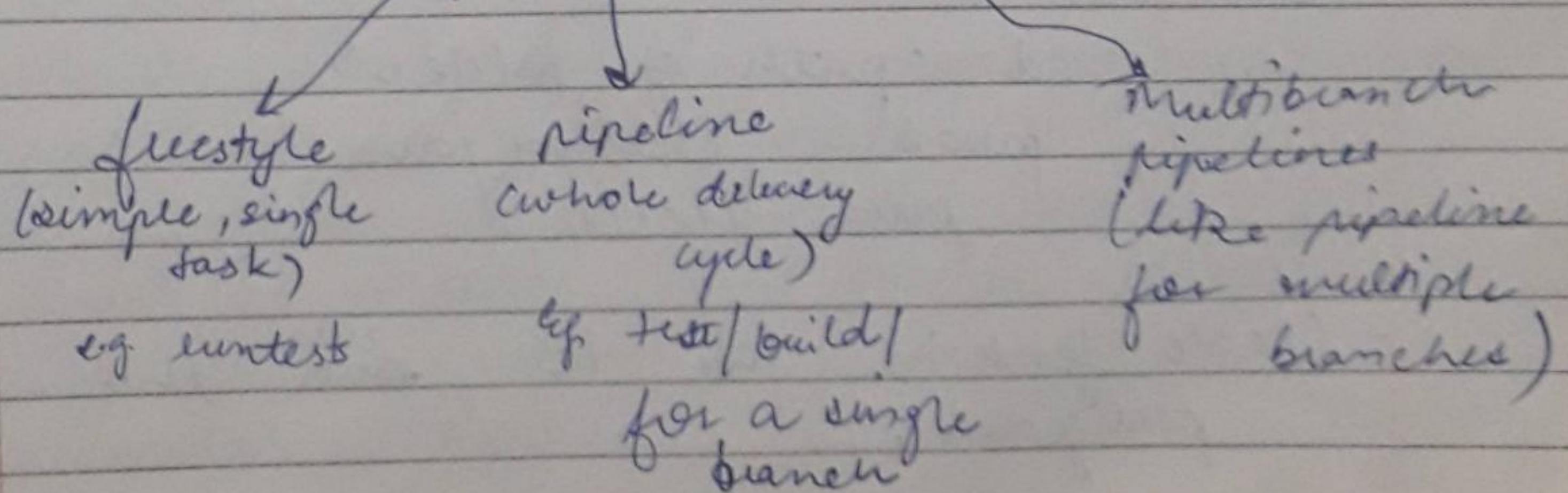
in source code & whole process keeps on repeating.

Using Jenkins to set off Docker containerisation

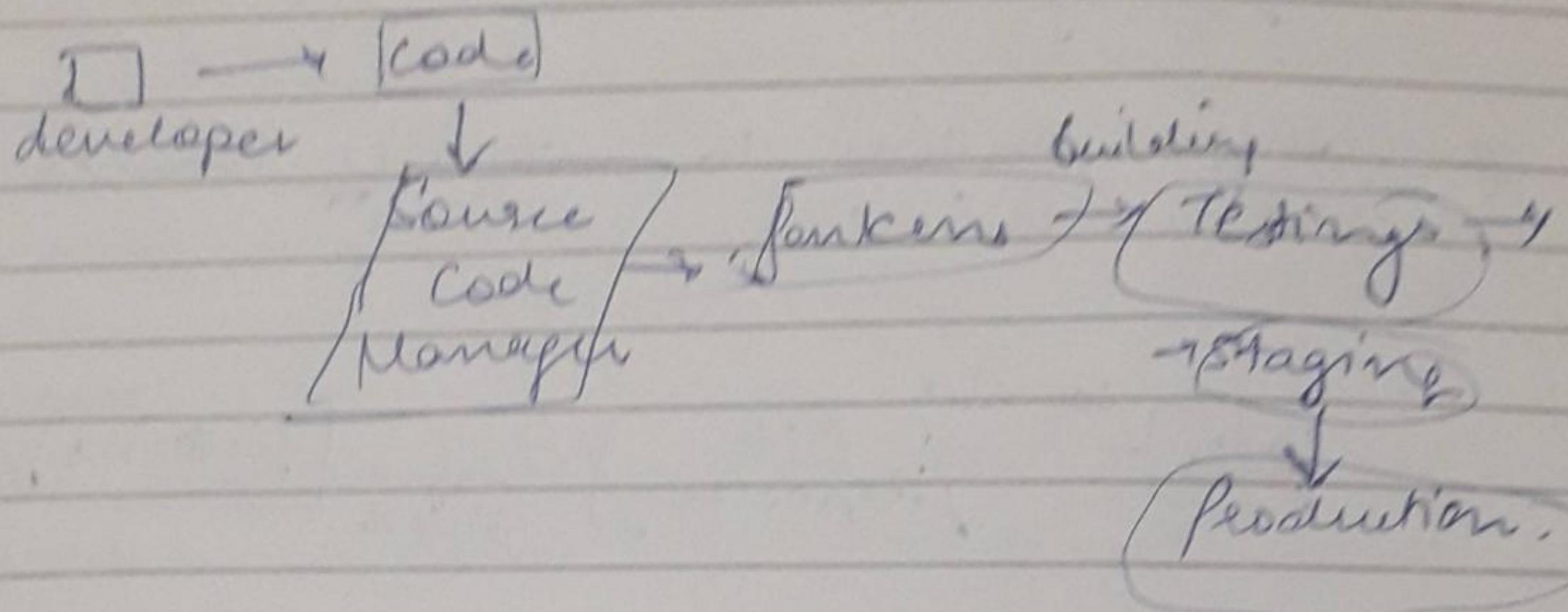


- Run Jenkins in docker container.
- Find ~~official~~^{initial} image of Jenkins in docker
- docker run jenkins/jenkins
- docker run -p 8080:8080 -p 50000:50000 -d -v jenkins_home:/var/jenkins_home jenkins/jenkins:1.6
- docker logs container_id
- localhost:8080

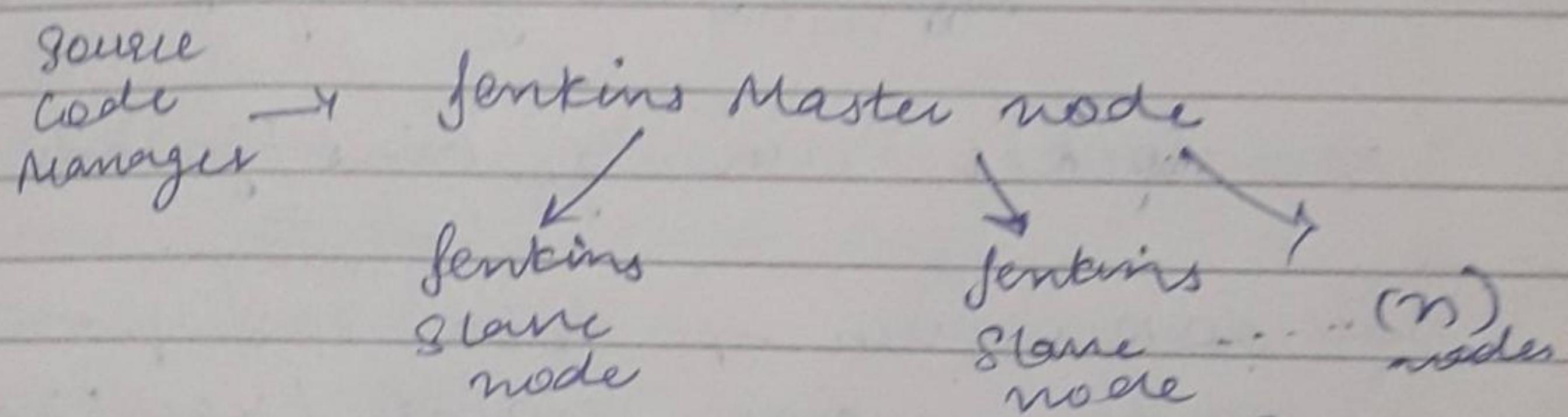
Three types of Jenkins project



Jenkins Pipeline



Master slave architecture of Jenkins 2.



Jenkins integration with Git

- Create a Java program, you can use any editor. (Eclipse).
- Run open the terminal & right command :-

```
cd <path of folder>
javac hello.java
java hello
```
- Create Jenkins job to run the program.

(2m)

Q1. What is Jenkins? what is the difference between Maven & ant and Jenkins?
(Hint: - Maven and ant are build technologies whereas Jenkins is C.I. tool.)

Q2. Which S.C.M. tools does jenkins support?
(Git, Subversion, RCS, Clearcase)

Q3. What is Cont. Integration (CI) in Jenkins?

Q4. What is the req for using Jenkins?
(Git, you need a source code repository which is accessible for eg git repository & working build script like maven.)

Q5. What are the advantages of Jenkins?

Q6. How to make sure that your project build does not break in jenkins?
(Hint: Continuous Integration & synchronization with repository)

Q7. How can you move or copy Jenkins from one server to another?

Ans. Step 1: copy the related job directory & side a job from one installation of Jenkins to another.

Step 2: Make a copy of already existing job by making clone of a job directory by a different name.

Step 3: Renaming & creating job by same a directory.

Q9. Which graph command is used to start Jenkins manually?

Ans9. - Restart :- force or a restart without waiting for build to complete

- Safe restart :- Allows all running builds to complete.

Q10. What are the most useful plugins in Jenkins?

Ans10. Maven2 project, HTML Publisher, Amazon EC2, Copy artifact, Jira, Green Balls

Q11. How to create a backup and copy files in Jenkins?

Ans11. Just copy the directory that saves all the config setting, built artifact & logs of Jenkins in its home directory.

- We can also clone & rename.

Q12. How can you clone a git repository via Jenkins?

Ans12. Execute Git configuration or enter email username of Jenkins history.

Q13. How can you setup Jenkins job?

Ans13. - Select new item from menu.

After that enter a name for the job & select freestyle job.

- Then click ok to create new job in jenkins
- The next page enables you to config your job.

Q14 what are the 2 components jenkins is mainly integrated with?

Ans14 Maven & git
(Build tool) (version control)

like Apache

Q15 what happens when a jenkins build fails?
Ans15 concerned team will be notified

not

Q16 What is req for using jenkins?

Ans16 you need a 'source' code repository which is accessible, for example git repository & a 'working' build script, for example, maven.

Q17 why is jenkins used with selenium?

Ans17 Se. allows you to test jenkins whenever software changes or send to a new environment

Q18 C.I. tools?

Ans18 Jenkins, Bamboo, Teamcity, Gitlab, Buddy, Puppet, codefresh, eduship