Introduction to DEVOPS and GIT Foundations

— starts at 9:06pm

AGENDA

1. Devops Mindset

2. 12 Factor Application

3. Foundation to Git

    1. Understanding git

    2. Installation

    3. Basic commands

→ Devops Mindset

    → cultural aspect

       → collaboration b/w the developers and operations team.

| dev team | | testing team | | ops team |

→ Product delivery time is increased.

→ Sharing responsibilities
→ joint planning session
→ frequent communication.

Key Principles.

1. Automation.
2. CI CD
3. Monitoring and logging.
4. Communication.

high quality
highly available
highly scalable and
easily maintainable

→ 12 Factor Application.
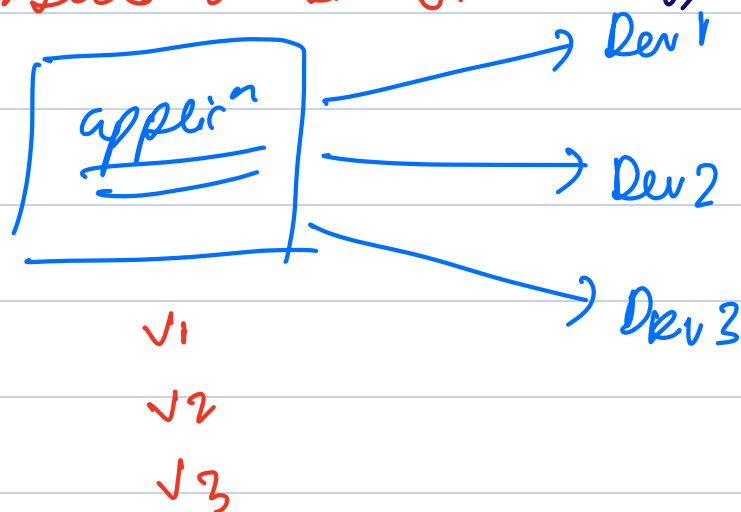
→ list of 12 Best Practices

① Codebase

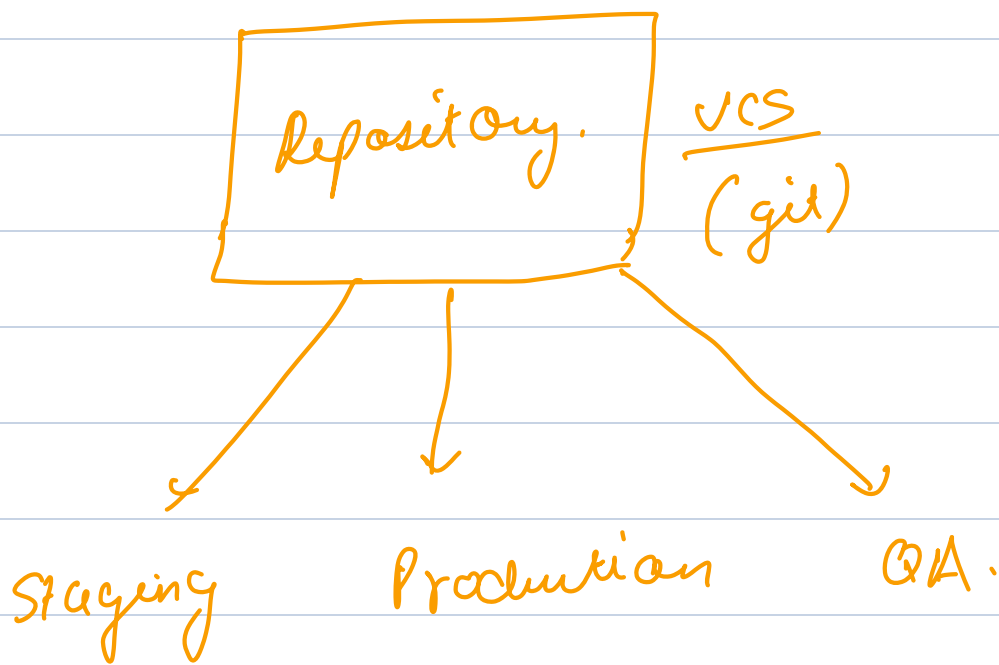→ single codebase per application stored in
a Version Control System

→ VCS

→ tracking changes
→ collaboration b/w developers.
→ ensure versions. (history)

appln

Dev 1

Dev 2

Dev 3

V1
V2
V3

Repository. $\underline{VCS}$ (git)
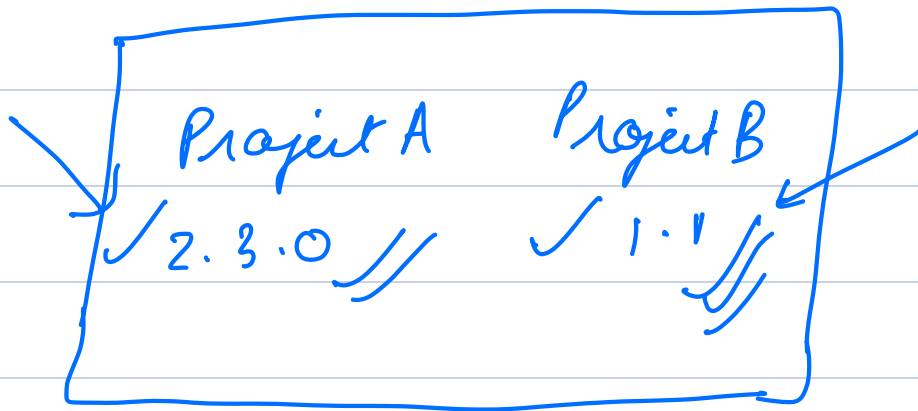
Staging   Production   QA.

② Dependencies

An application should explicitly declare and isolate its dependencies

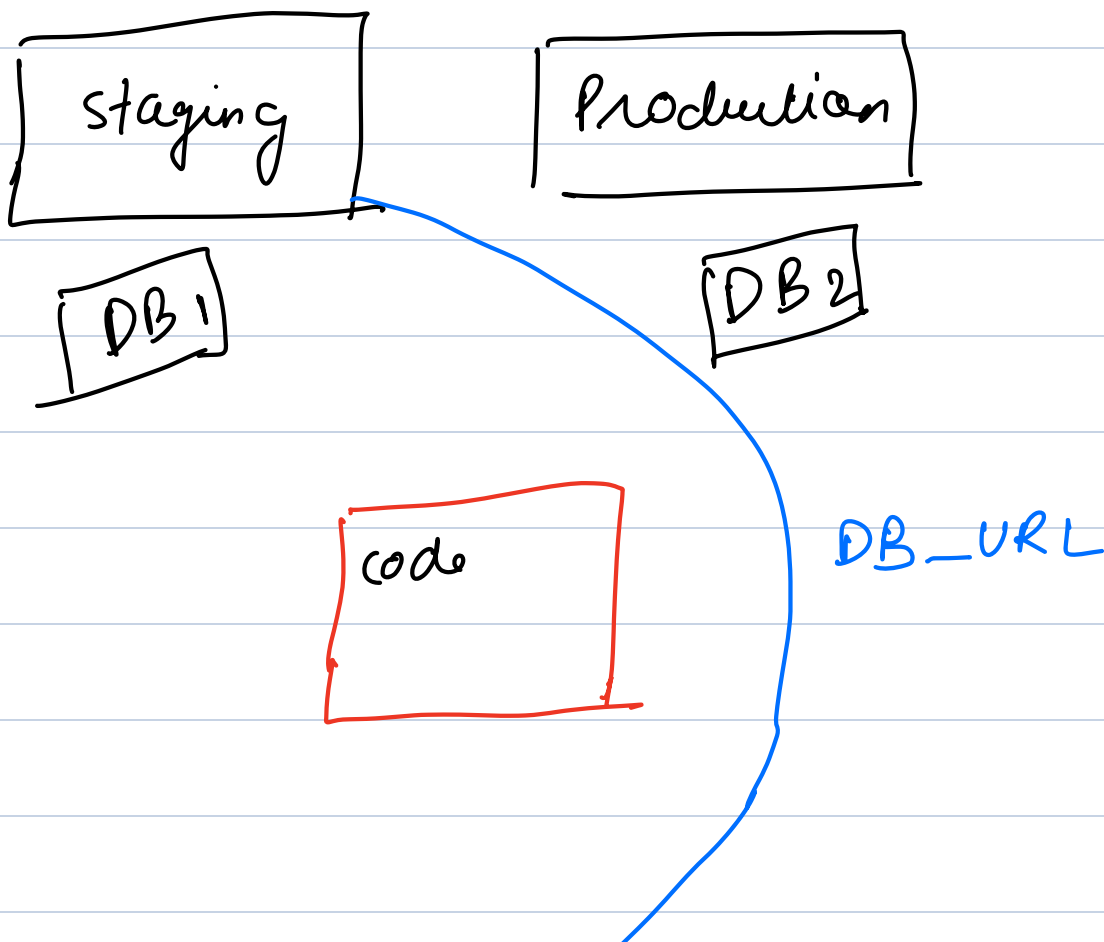dependency → third party packages or libraries.

requirements.txt

Flask == 2.3.0

→ pip install requirements. txt

Project A   Project B
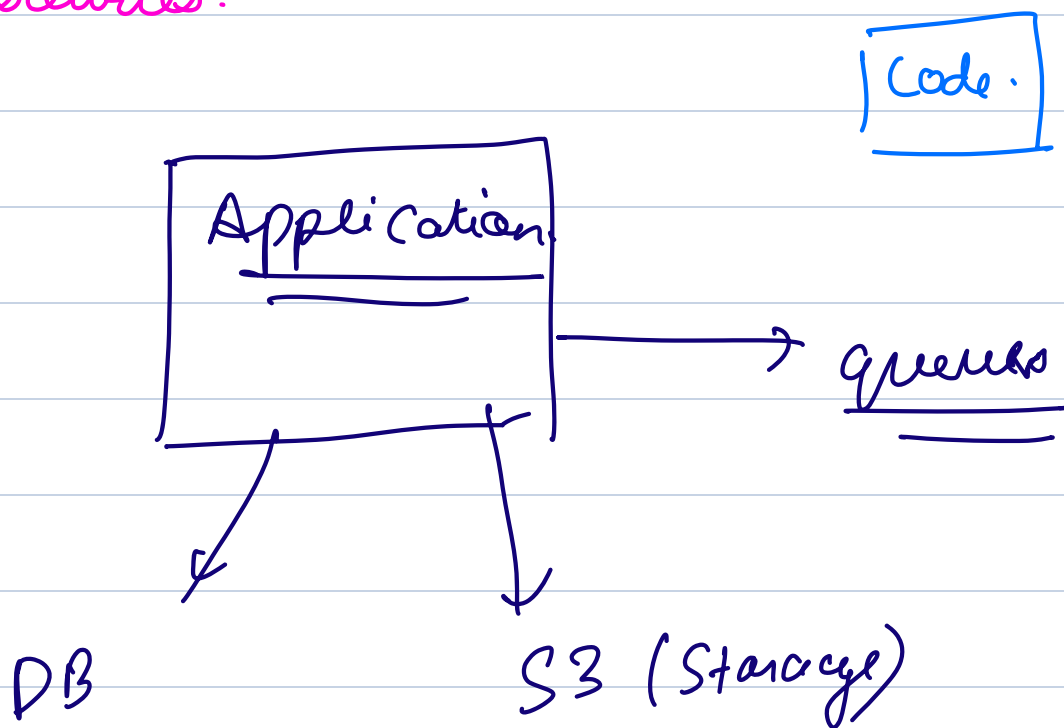√ 2.3.0 //   √ 1.1 //

③ Configuration

Store configuration in the environment.

staging   Production

DB1   DB2

code   DB_URL

```
export  DB_URL = "xyz"
export  DB_Password = "pqr"
```

④ Backing Services

Treat Backing services as attached resources.

Application

Code.

→ queues

DB

S3 (Storage)

⑤ Build Release Run.
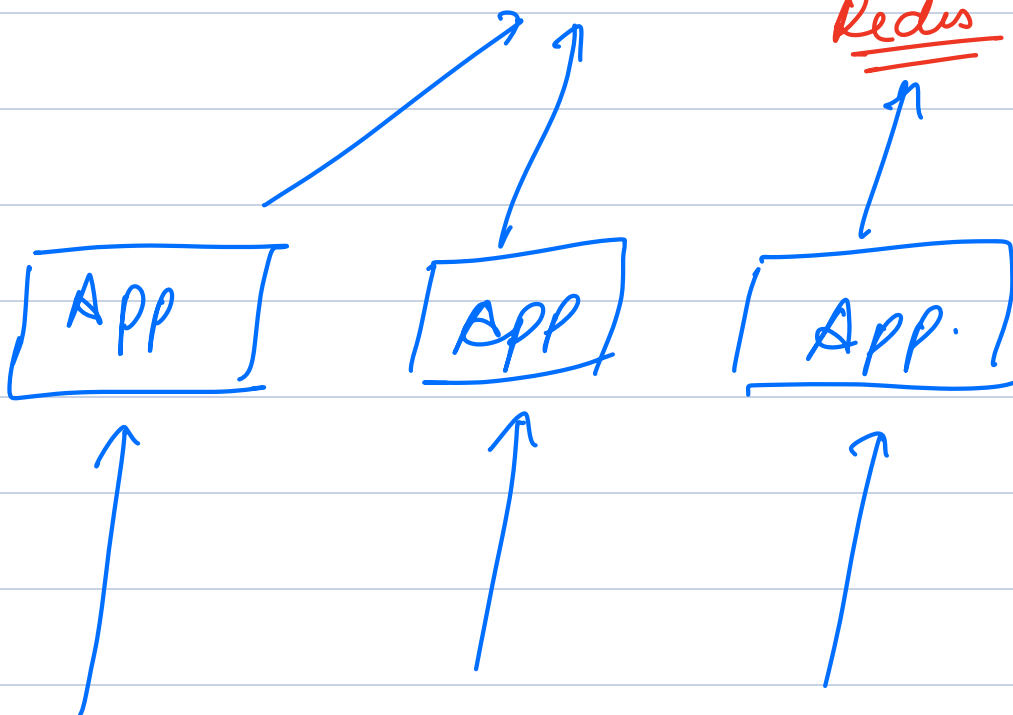
strictly seperate build, release and run stages.

Build → creating exentables.
Release → Build + configuration.
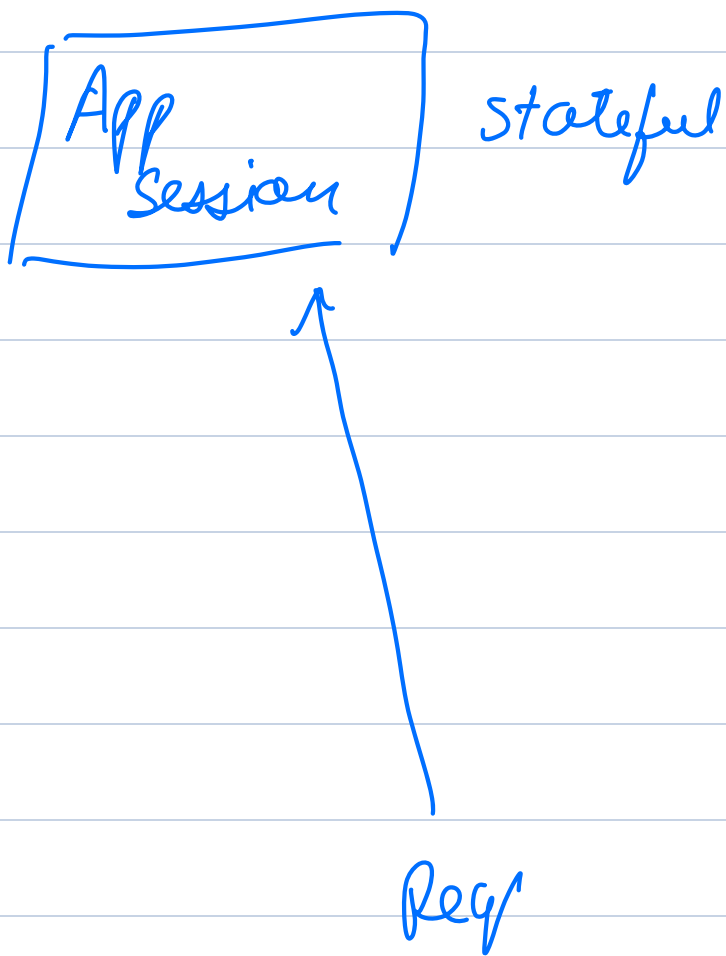
Run → Starting the application using
release.

(6) Processes.

Execute the app^d as one or more stateless
process.

Persistent data → DB   cache
                                Redis

App   App   App.

App
Session

Stateful

↑

Req

⑦ Port Binding.

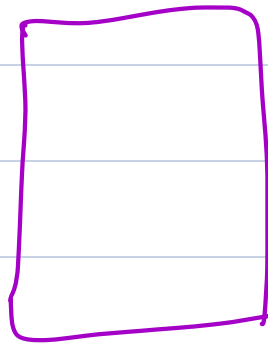→ Export services via port binding.

⑧ Concurrency.

Scale out via the process model.

→ E - commerce web application.

① Web Process

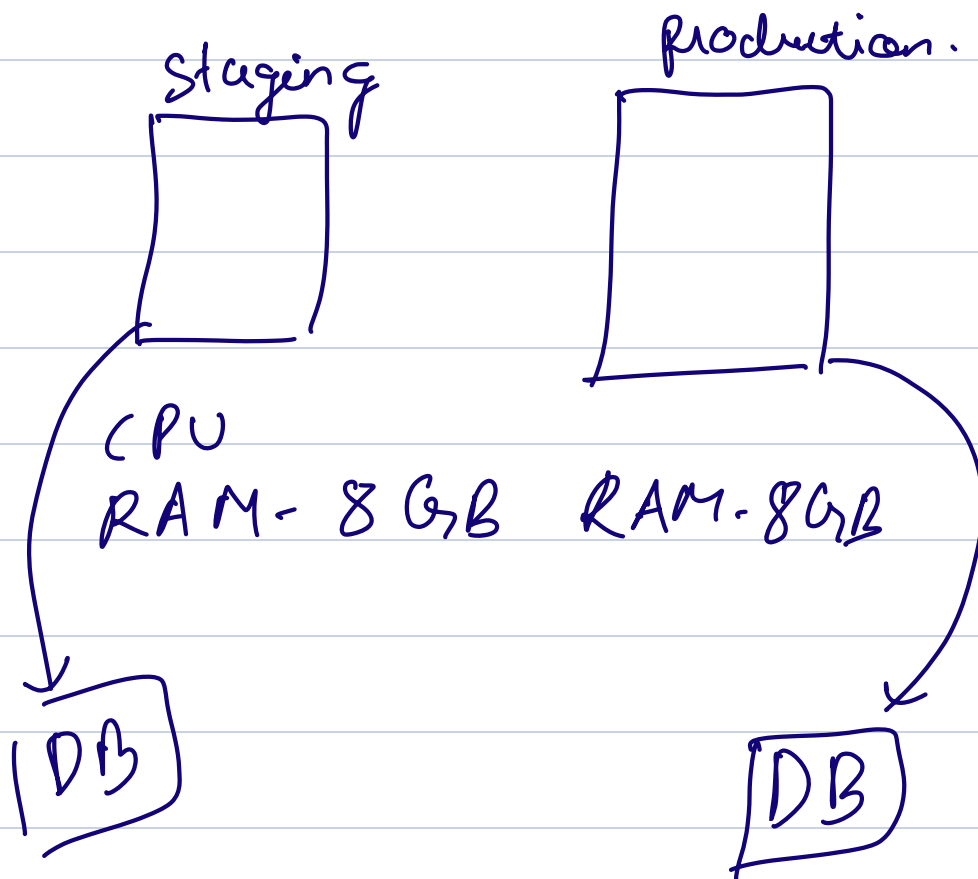② Worker Process

③ Schedular Process.

→ notification

queue.

> 1000

⑨ Risposability

Maximise robustness with fast startup
and graceful shutdown.

(10) Dev/Prod Parity

Keep development, production and staging environment as similar as possible.

staging

production.

CPU
RAM- 8 GB   RAM- 8 GB

DB

DB

(11) Logs.

→ Treat logs as event streams.

(12) Admin Processes.

Run admin / management task as one-off processes.
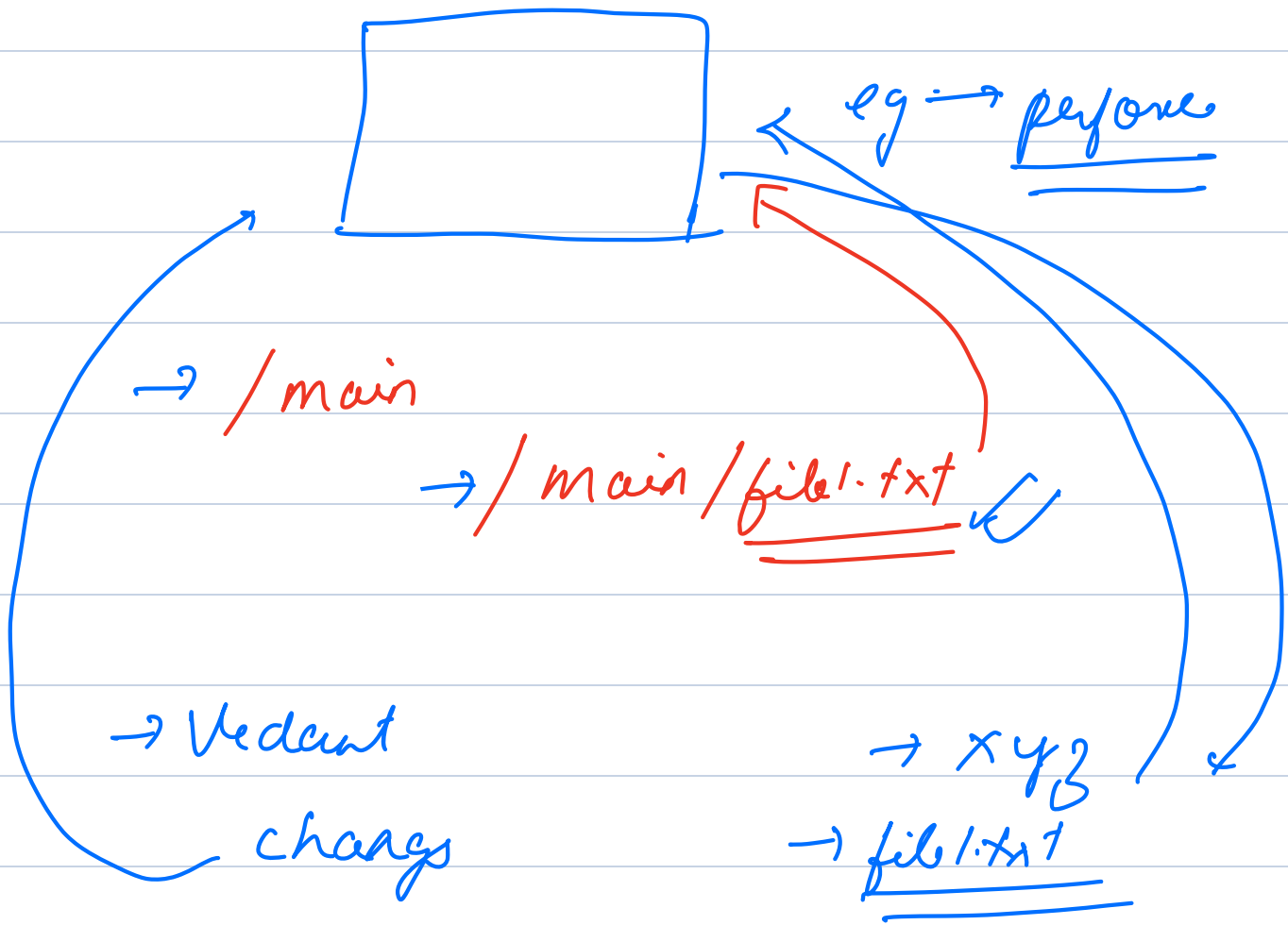
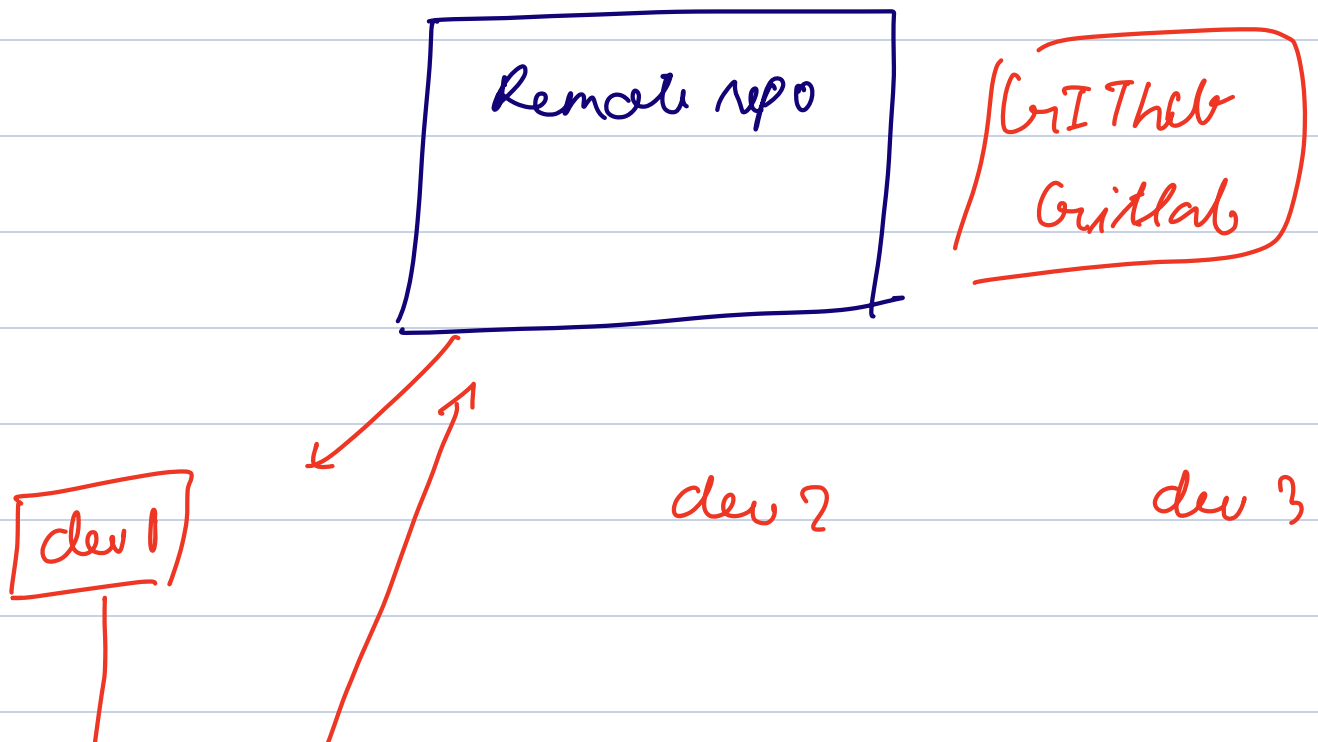(DB backup)
↓
→ Separate from main application.
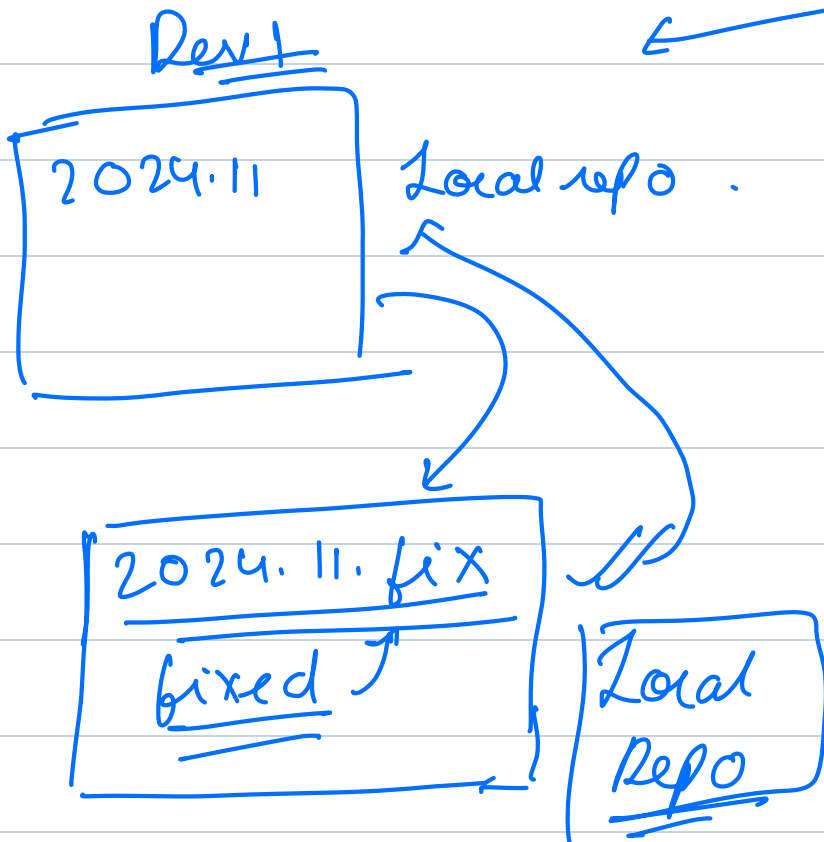
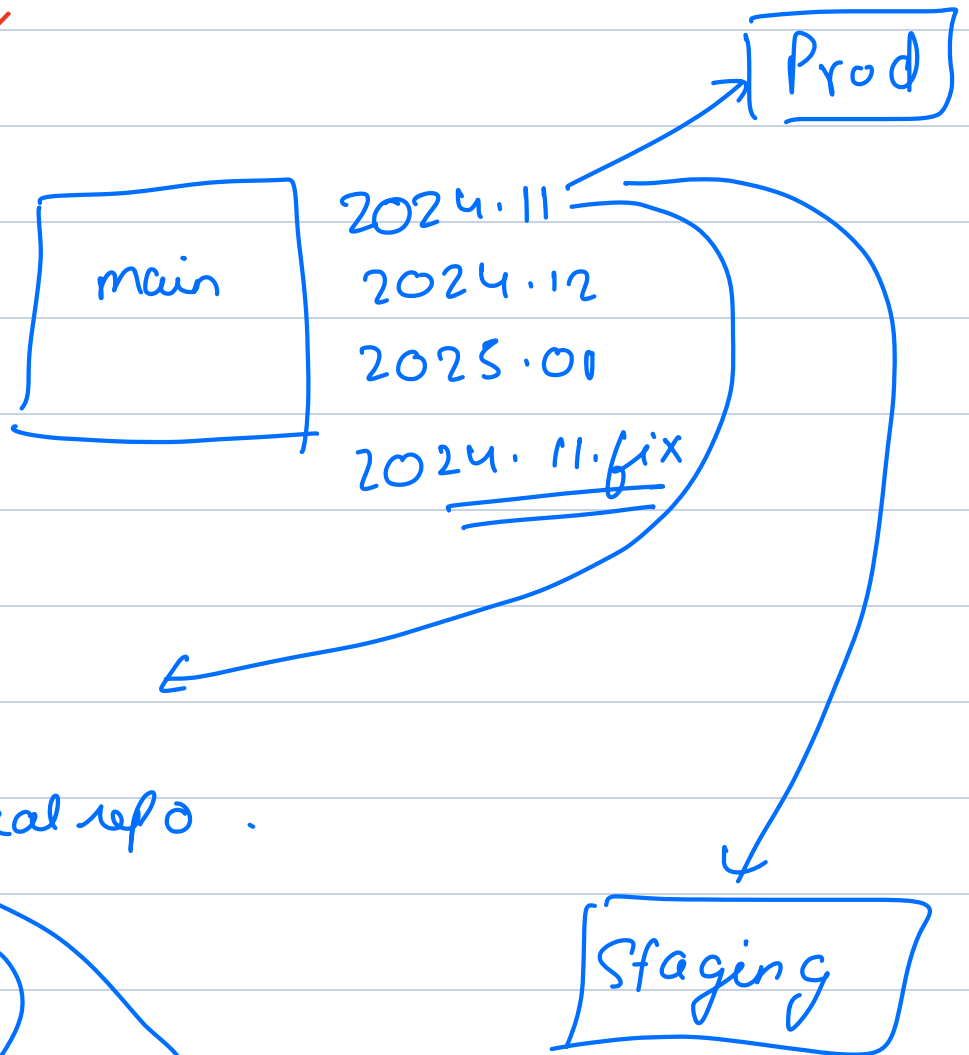| 10 : 35 pm |

→ GIT

→ Early days → Pendrives or hand drives.

→ Google Drives, Dropbox.

→ Remote Repositories.

eg → perform

→ /main

→ /main/file1.txt

→ Vedant
changes

→ xyz
→ file1.txt

→ GIT → VC

Remote repo   GIThub
                     Gitlab

dev 1   dev 2   dev 3

V_1

Local Repo
$E2_2$

V_2

V_3

file1.txt

main
2024.11
2024.12
2025.01
2024.11.fix

Prod

Dev1

2024.11

Local repo.

Staging

2024.11.fix
fixed

Local
Repo

$\boxed{\text{main}}$ $\longrightarrow$ 2024.11 $\longrightarrow$ 2024.11 fix1

$\searrow$ 2024.11 fix2
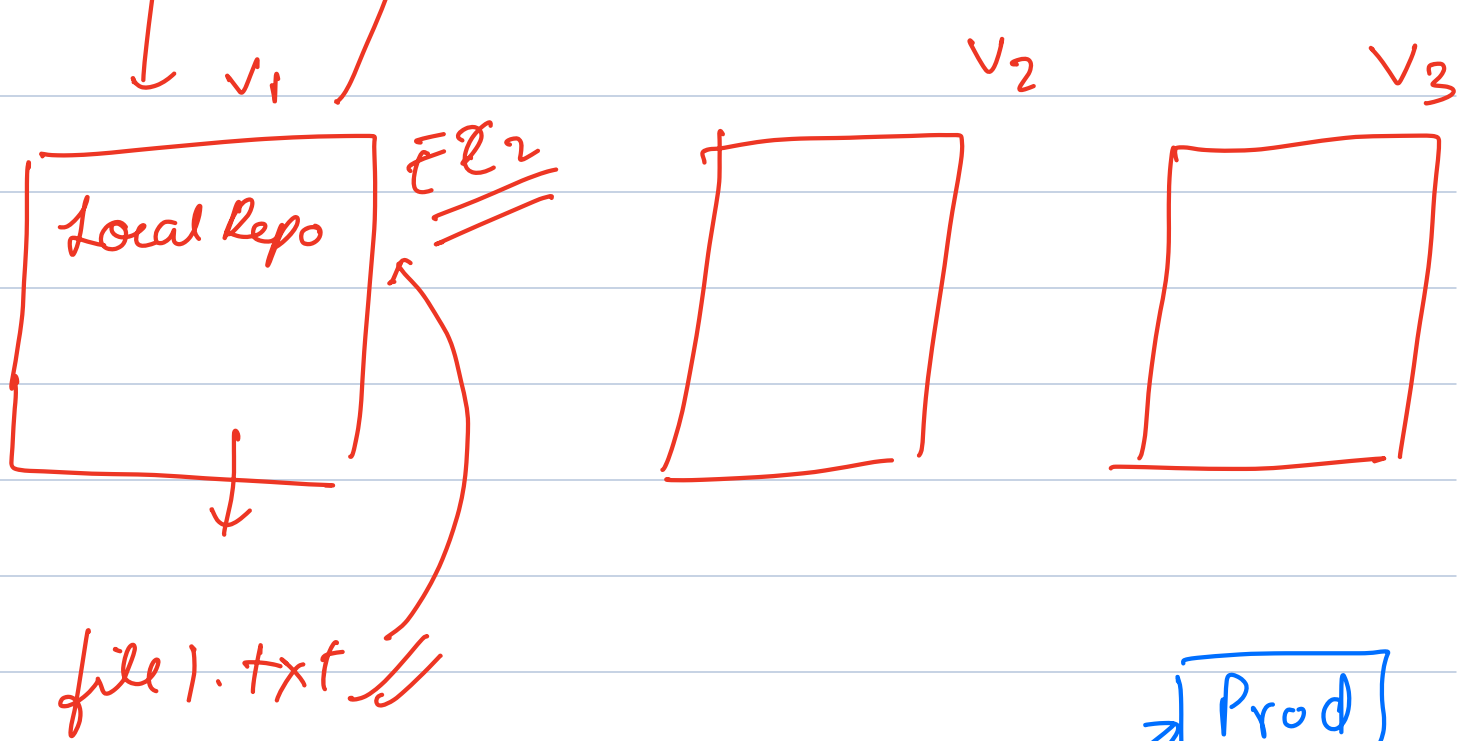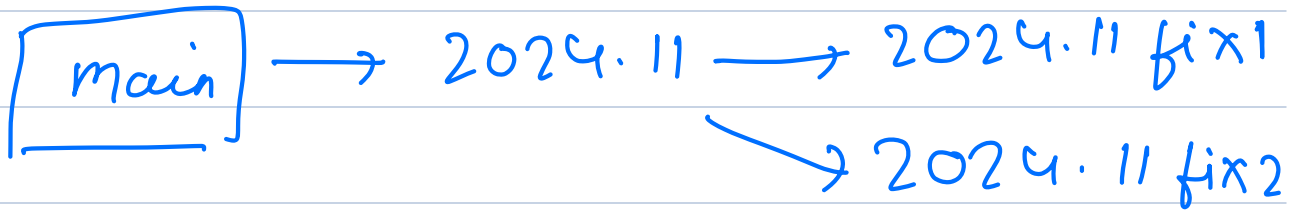
$\longrightarrow$ Demo.

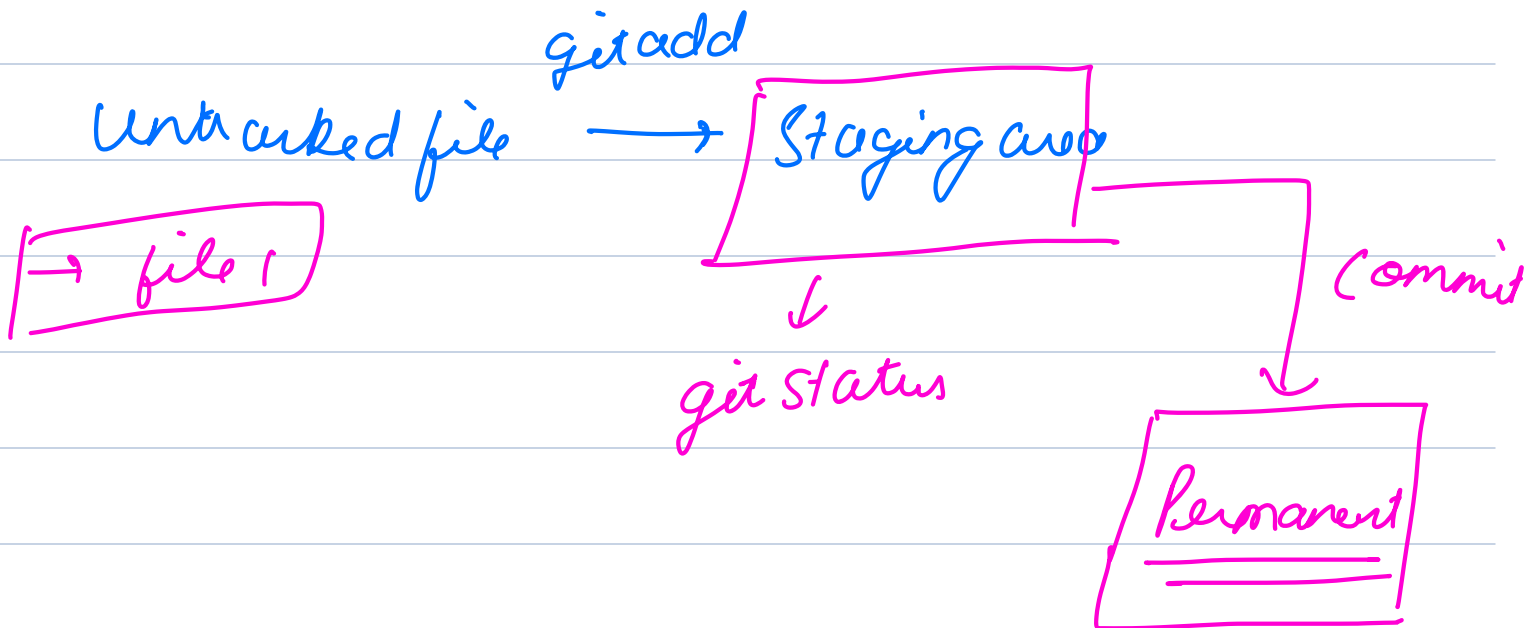$\rightarrow$ sudo apt install git

$\rightarrow$ git init $\longrightarrow$ for initialising

git add

Untracked file $\longrightarrow$ $\boxed{\text{Staging area}}$

$\boxed{\rightarrow \text{file 1}}$

$\downarrow$

git status

commit

$\downarrow$

$\boxed{\text{Permanent}}$

$\rightarrow$ git diff

→ git restore \<filename\>
↓

restore if content is not committed

→ git branch new-branch