DEVOPS

April 2023

Bala Sudhakar Gubbala

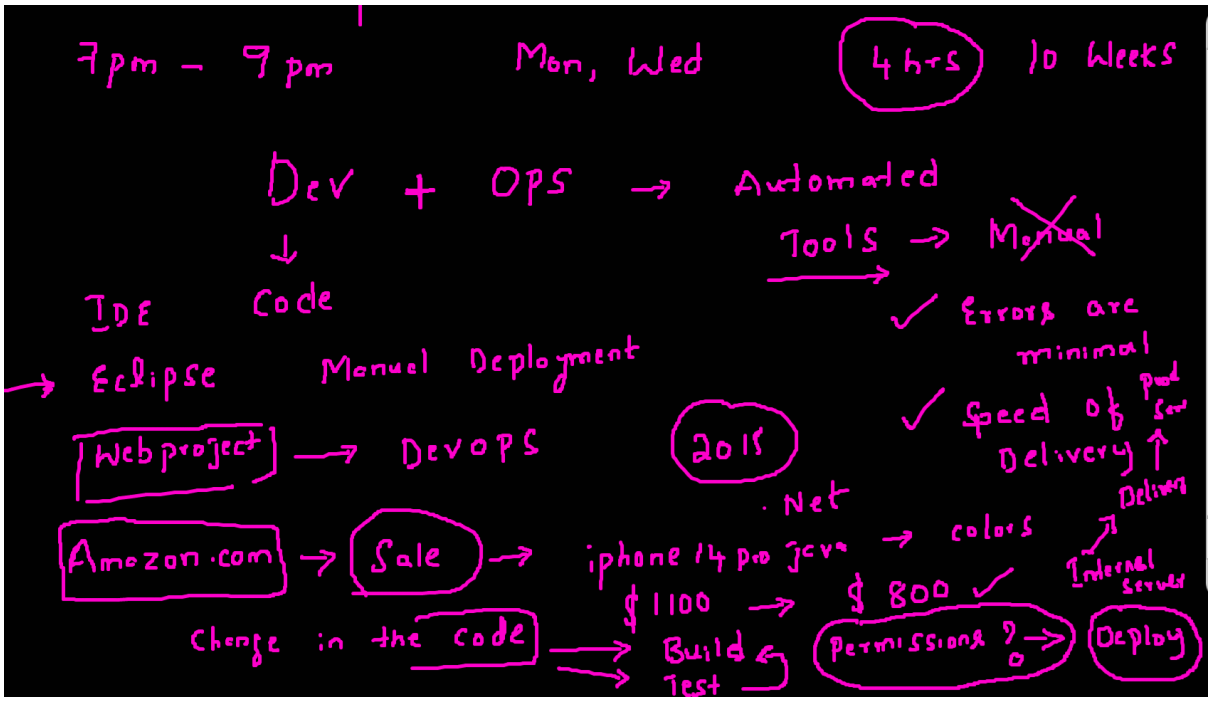
codewithbala@gmail.com

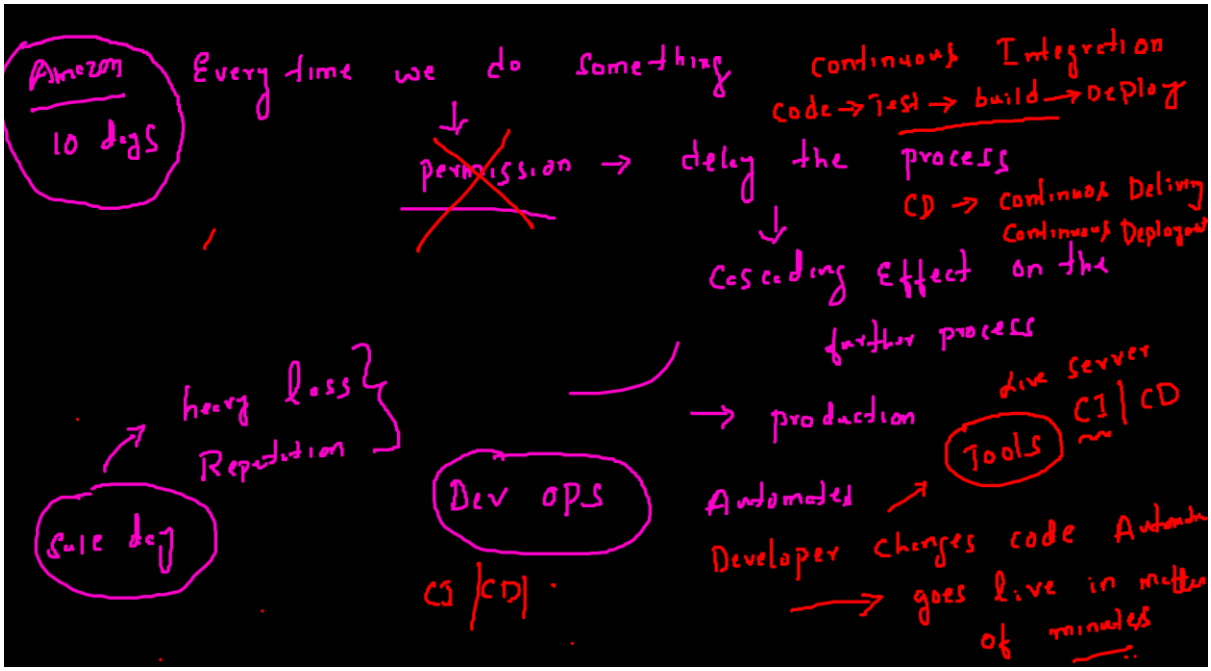
Contents

DevOps

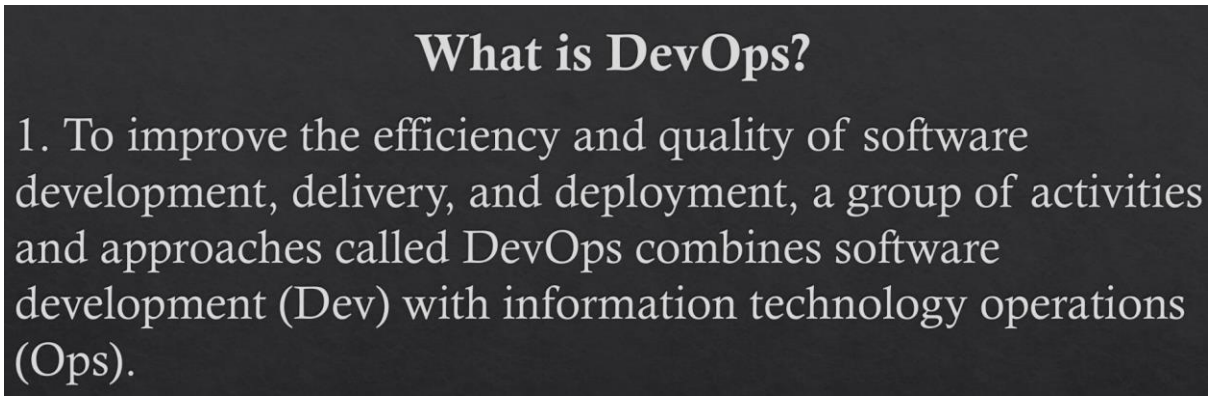
Session 1 (7pm-9pm) 24th April 2023

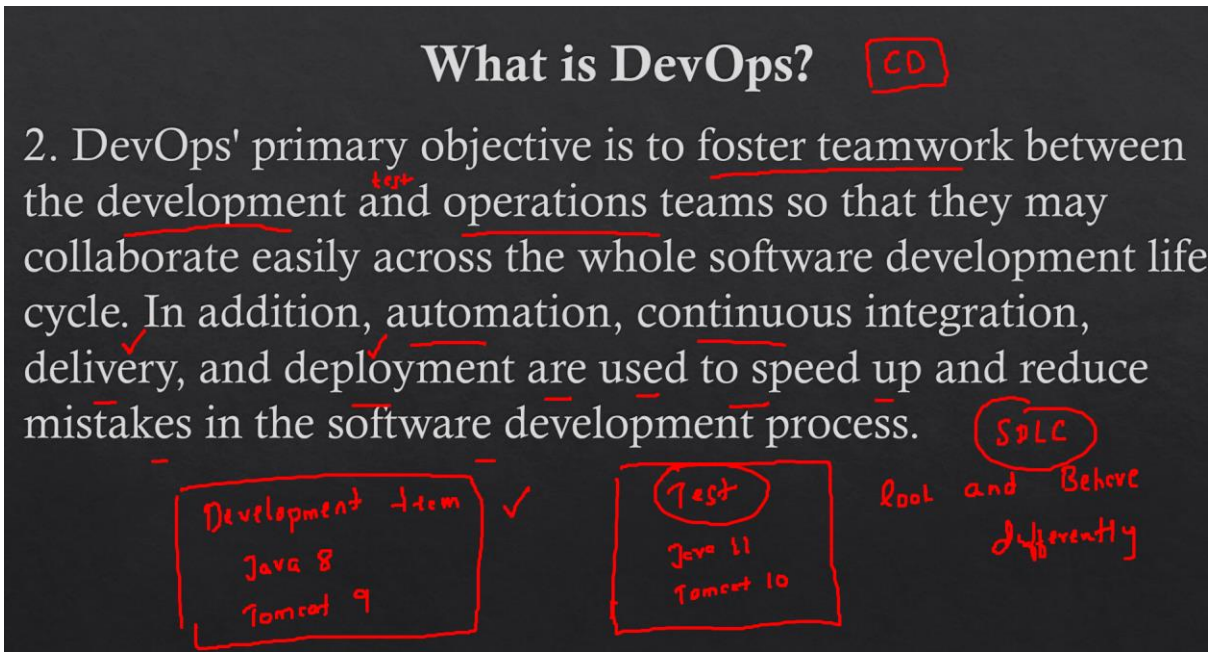
DevOps Intro





What is DevOps



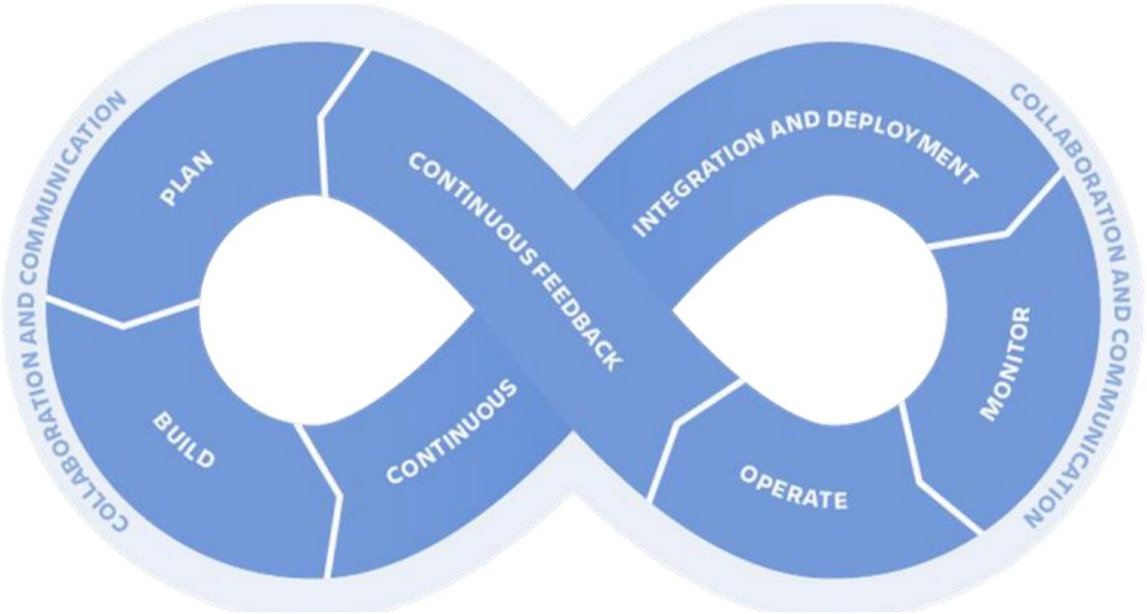


What is DevOps?

3. Monitoring and feedback are also emphasized in DevOps, which enables the development and operations teams to see problems early and proactively handle them. Using DevOps methods, businesses may improve their agility, competitiveness, and overall productivity by achieving quicker release cycles, higher-quality software, and enhanced team cooperation.

DevOps

- DevOps is **the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity**: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.



Linux Commands

lsb_release -a

832 cat /etc/os-release

833 clear

```
834  date
835  ls
836  ls -la
837  pwd
838  cal
839  apt install cal
840  cal
841  ls -l
842  ls -al
843  ls -l -a
844  man ls
845  man pwd
846  mkdir mydir24
847  cd mydir24
848  touch file1
849  ls
850  ls -la
851  cat file1
852  echo "Hello File" -> file1
853  cat file1
854  echo "Hello Again" -> file1
855  cat file1
856  echo "Third attempt" >> file1
857  cat file1
858  echo "fourth attempt" >> file1
859  cat file1
860  touch 1.txt
861  touch 2.doc
862  touch 3.html
863  touch 4.java
864  touch 5.js
```

865 ls

866 cat 1.txt

867 vim 1.txt

868 cat 1.txt

869 vim 1.txt

870 cat vim 1.txt

871 cat 1.txt

872 vim 3.html

873 cat 3.html

874 ls

875 vi 4.java

876 cat 4.java

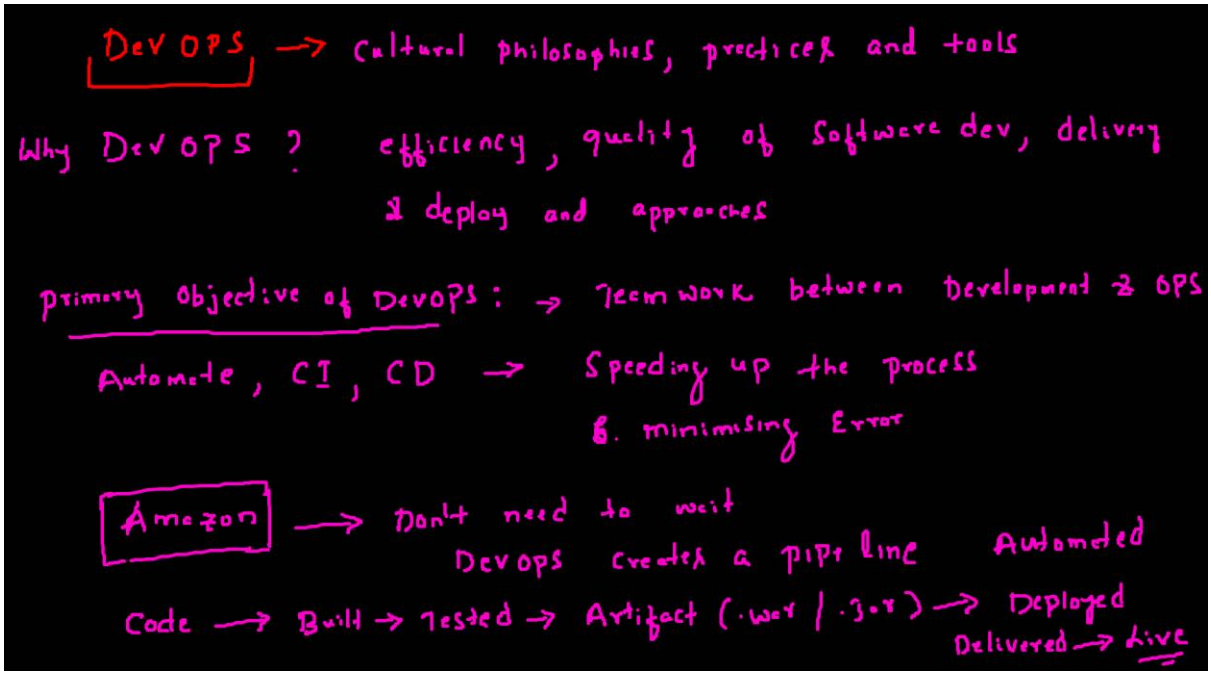
877 vim 4.java

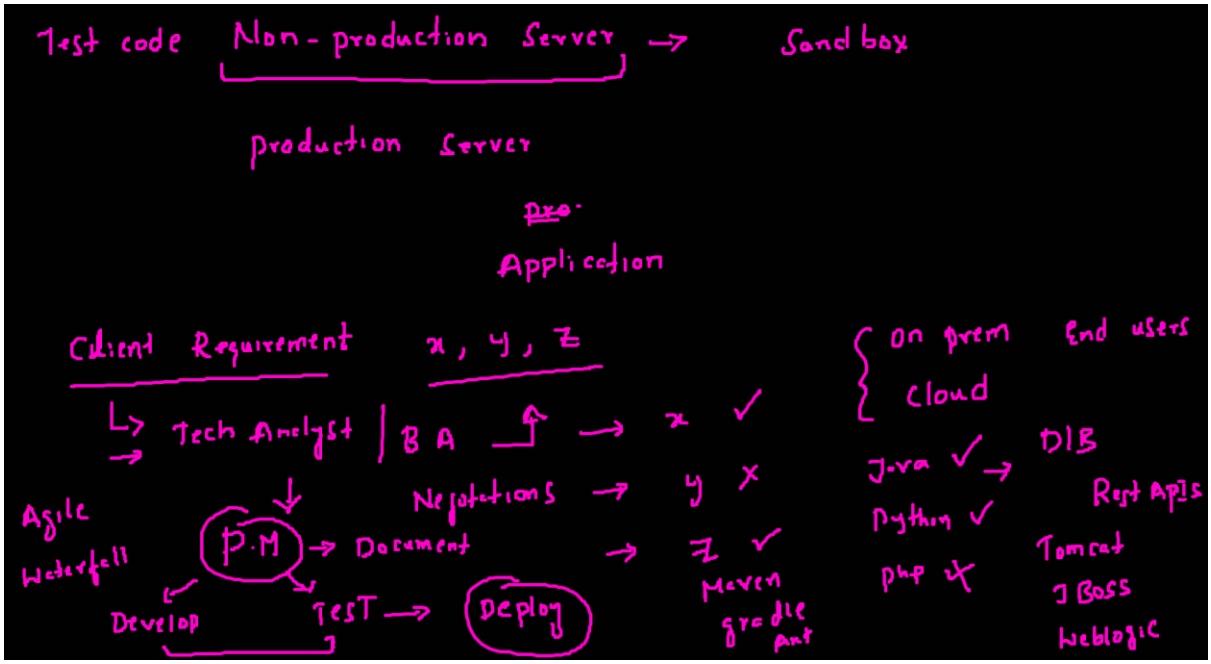
878 cat 4.java

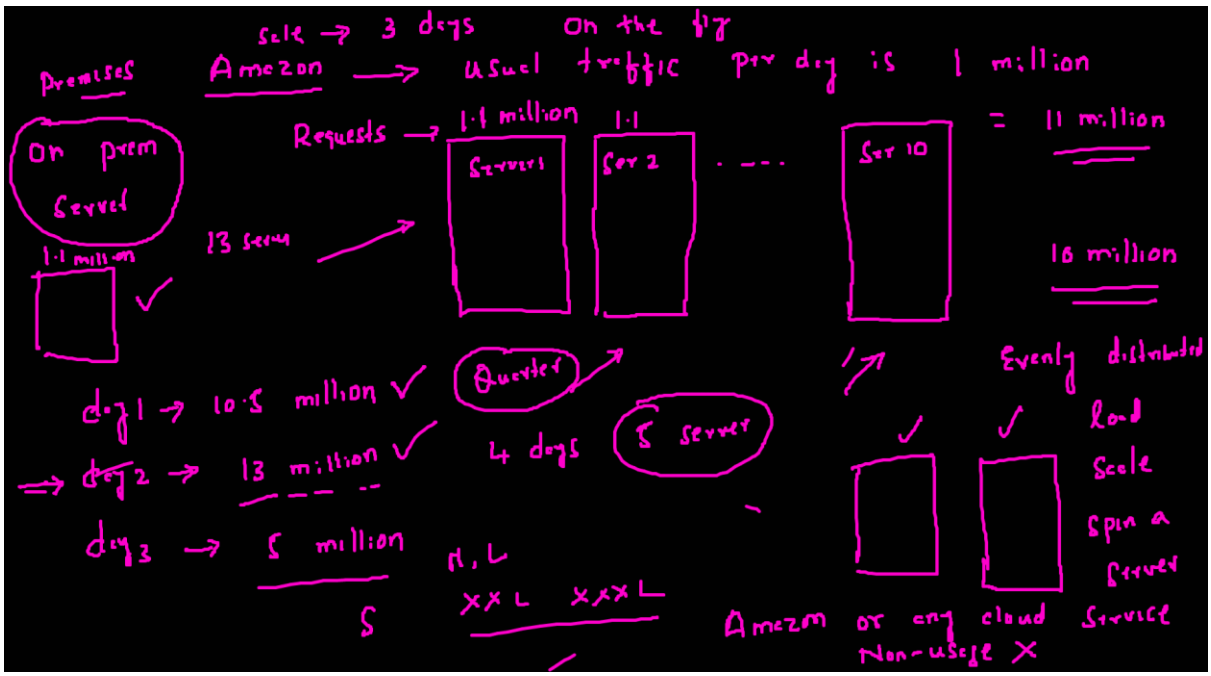
879 history

Session 2 (7pm-9pm) 26th April 2023

Recap




Test code
Non-production Server
Sandbox
Production Server
Pre-
Application
Client Requirement
x, y, z
Tech Analyst
BA
Negotiations
Agile
Waterfall
P.M
Document
Develop
TesT
Deploy
Maven
gradle
Ant
On prem
Cloud
End users
Java
Python
php
DB
Rest APIs
Tomcat
JBoss
Weblogic


Scale → 3 days
On the fly
Premises
Amazon
Usual traffic per day is 1 million
On prem
Served
1.1 million
Requests
1.1 million
1.1
Server1
Ser 2
Ser 10
= 11 million
13 servers
16 million
day1 → 10.5 million
day2 → 13 million
day3 → 5 million
Queued
4 days
5 server
Evenly distributed
load
Scale
Spin a
Server
S
M, L
XXL
XXXL
Amazon or any cloud Service
Non-usage X

End → End , less documentation

Agile → client is involved and a feedback is taken

fine tune

Major tech change → Never too late

→

Waterfall → End → Client → On prem

1 year →

Cloud

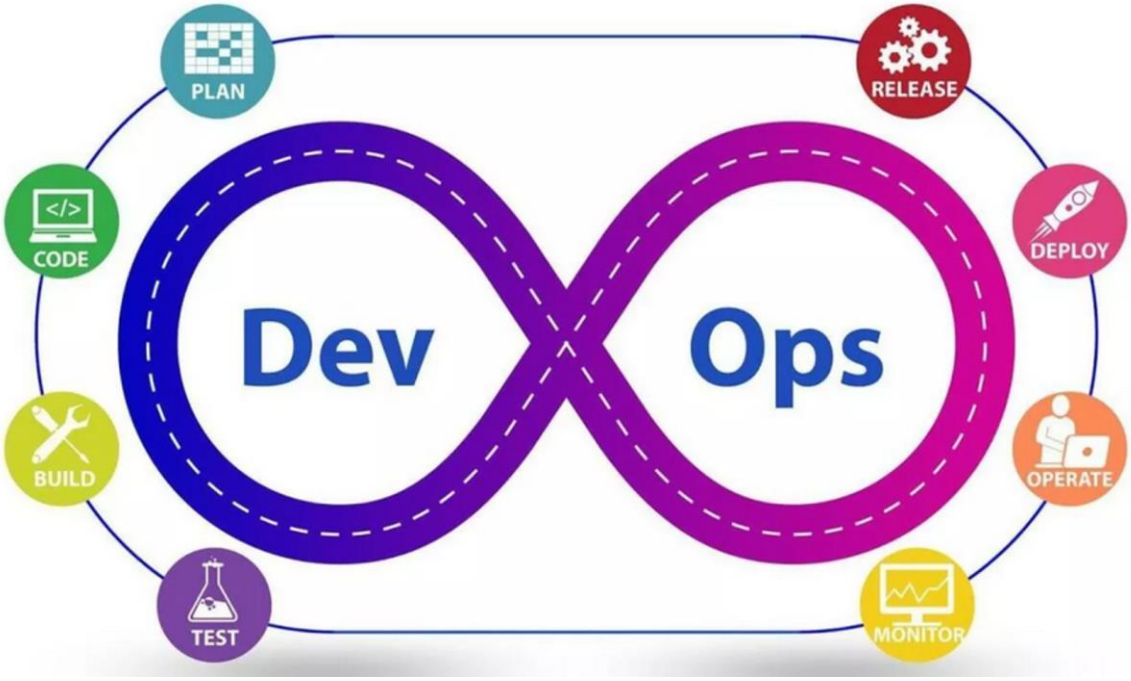
Scale up
Scale down
} minutes

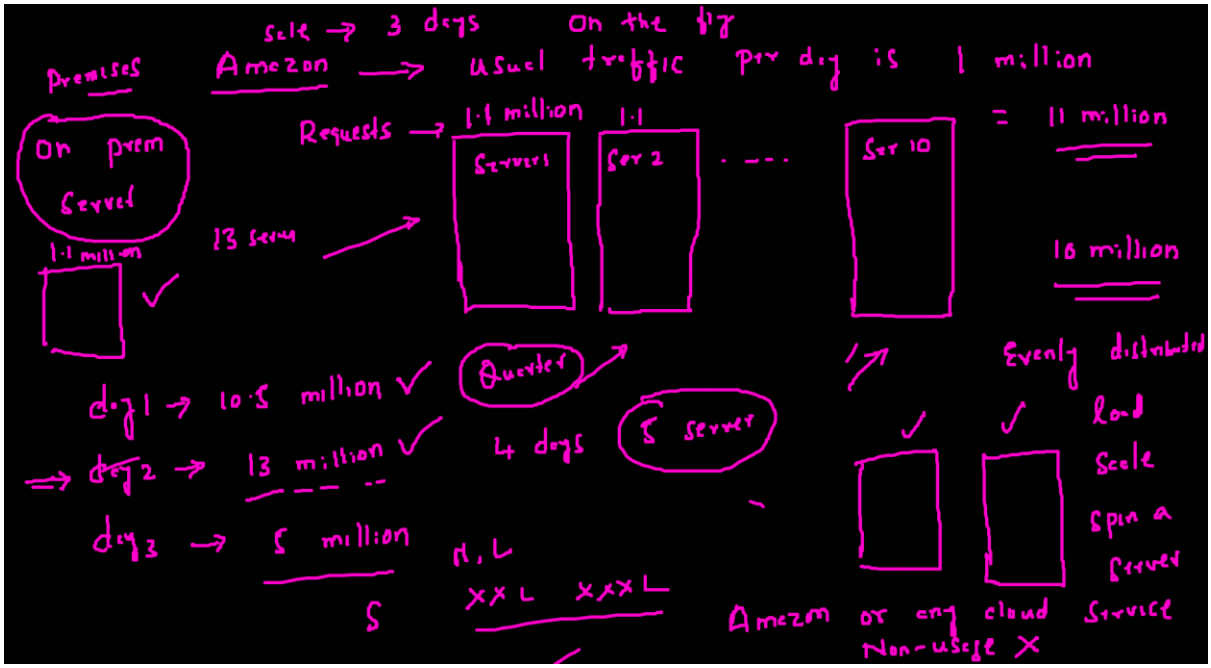
Vertical
Horizontal Scaling

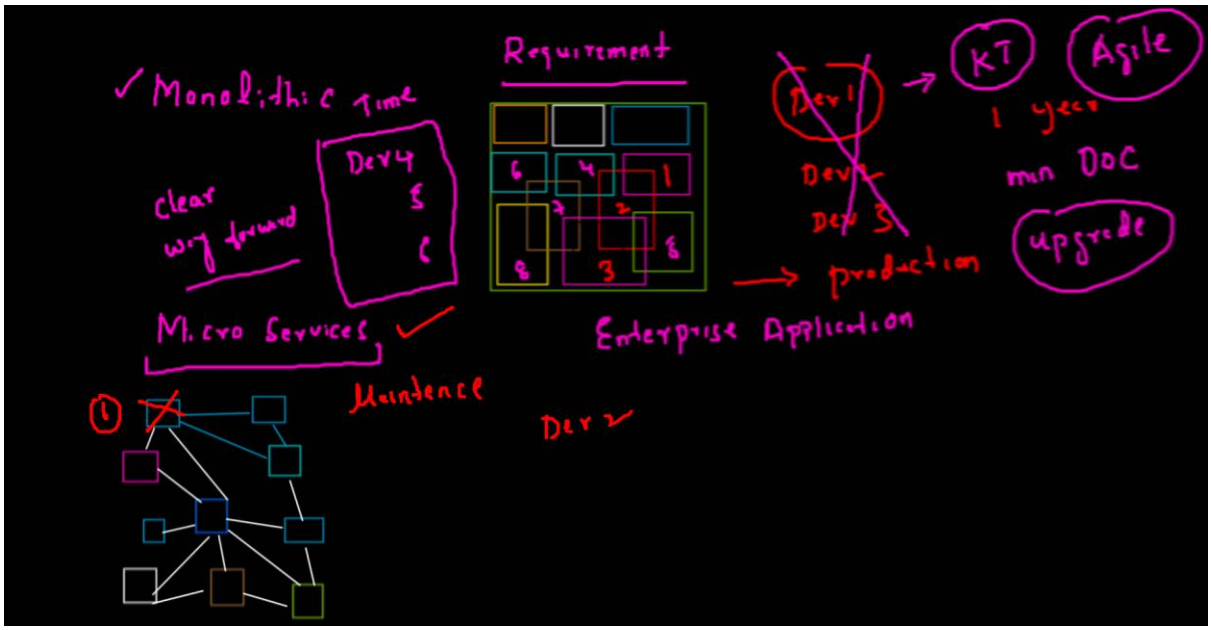
adding more
costs

→ for less expensive

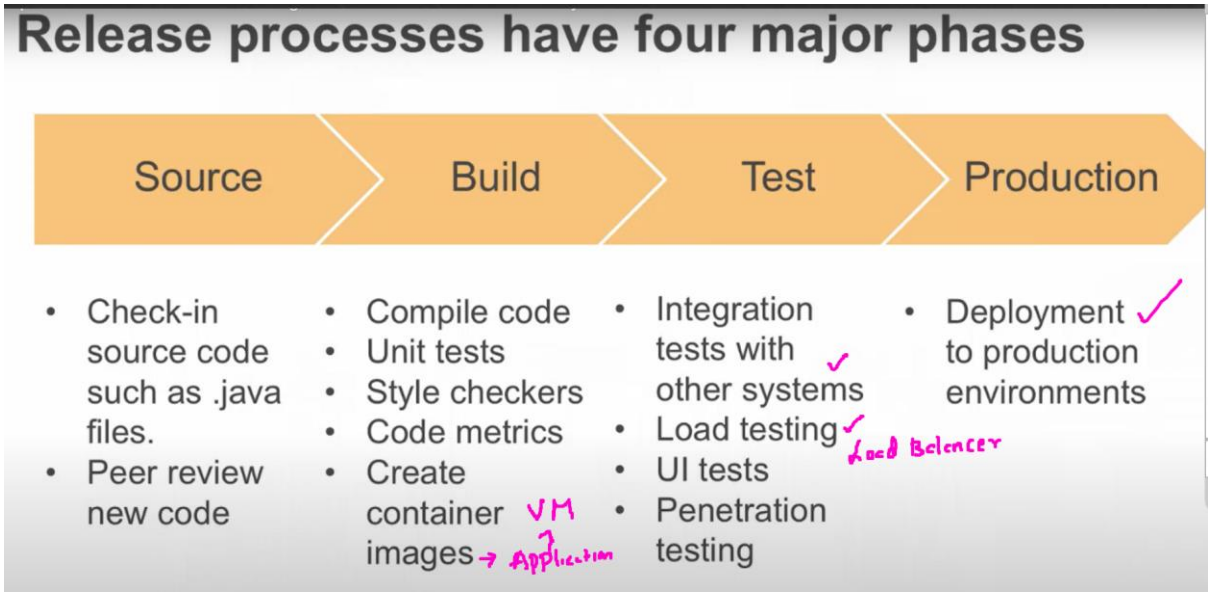
Scale Buy Server
infrastructure
Maintenance personell



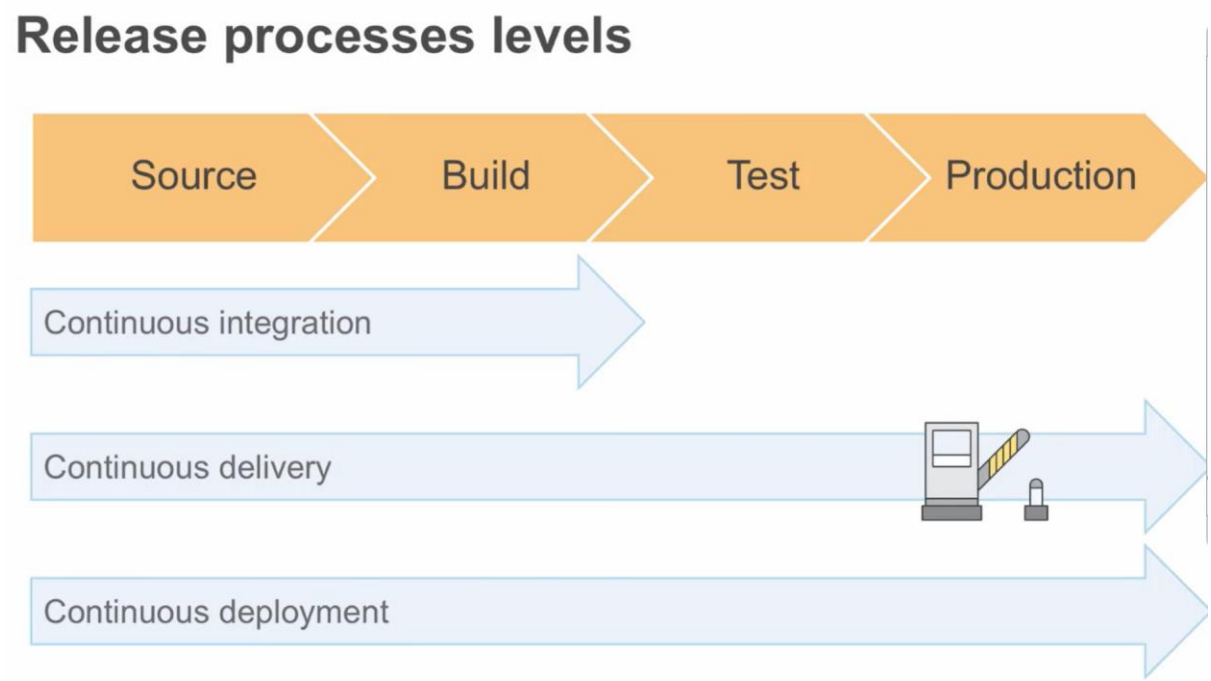

Sale → 3 days
On the fly
Premises
Amazon → usual traffic per day is 1 million
On prem Server
1.1 million
Requests → 1.1 million
1.1
= 11 million
Server1
Ser 2
Ser 10
13 servers
18 million
day1 → 10.5 million ✓
day2 → 13 million ✓
day3 → 5 million
Queued
4 days
5 server
Evenly distributed load
Scale
Spin a Server
S
M, L
XXL XXXL
Amazon or any cloud service
Non-usage X

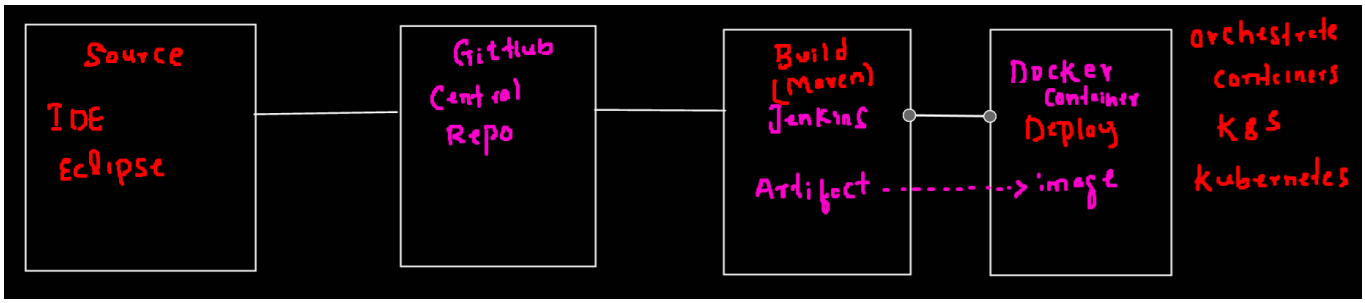

✓ Monolithic Time
Requirement
Dev4
clear way forward
Micro Services ✓
Maintence
Dev 2
Dev1
Dev 3
Dev 5
production
Enterprise Application
KT
Agile
1 year
min DOC
upgrade

Release Processes



CI/CD





Vi editor commands

How to exit the vi editor

Command	Description
q	If the file is not modified, the system returns to the Shell interface. Otherwise, the system prompts you to save the file.
q!	Forcibly exit without saving the modification.
wq	**w** indicates that the file is saved, and **q** indicates that the vi editor exits. The overall function is to save and exit.
wq!	**w** indicates that the file is saved. **q!** indicates forcibly exits the vi editor. The overall function is to save and exit.
zz	Save the file and exit, which is equivalent to wq.

vim

What is vim

- Vim is also a Linux text editor. It is an upgraded version of **vi**, and its full name is **vi improved**. In addition to the functions of vi, it has the features of multi-level undo, visual operation, syntax highlighting, and ease of use.
- multi-level undo
- In the vi command mode, you can only undo the previous command by pressing **u**. In the vim command mode, you can undo the command without restriction.
- Ease of use
- **vi** can run only on Linux, while **vim** can run on multiple operating platforms such as linux, Windows, and Mac.

What is the difference between vi and vim

- Both **vi** and **vim** are editors in Linux. The difference is that vim is advanced and can be regarded as an upgrade version of vi. vi is used for text editing, but vim is more suitable for coding.

Common vi commands

Command	Description
Move the cursor(Command mode)	
j	Move one row down
k	Move one row up
h	Move the cursor one character to the left
l	Move the cursor one character to the right
Ctrl+b	Move up one screen
Ctrl+f	Move down one screen

Common vi commands

Command	Description
↑	Move up
↓	Move down
←	Move left
→	Move right

Nano editor

How to Edit Text

- These are the commonly used shortcuts when editing a text in Nano.
- To **select text**, go to the beginning of the desired text and press **ALT + A**. This will set a mark for selecting. Then, you can move over the text with arrow keys.
- Press **ALT + 6** to copy the selected text to the clipboard.
- To cut the highlighted text, press **CTRL + K**.
- If you want to paste the text, navigate to the intended place and press **CTRL + U**.

Linux commands history

mkdir mydir26

```
1052  cd ..
1053  exit
1054  ls
1055  cd mydir26
1056  nano index.html
1057  cat index.html
1058  nano index.html
1059  LS
1060  ls
1061  nano index1.html
1062  cat index1.html >> index2.html
1063  cat index1.html
1064  cat index2.html
1065  nano index1.html
1066  ls
1067  cat index.html >> index2.html
1068  cat index2.html
1069  sudo bash -c 'cat index2.html >> index1.html'
1070  cat index1.html
1071  mv index2.html index20.html
1072  ls
1073  pwd
1074  cd ..
1075  cd mydir24
1076  pwd
1077  ls
1078  cd ..
1079  mv /root/mydir26/index20.html /root/mydir24
1080  cd mydir24
1081  ls
1082  cd ..
```

1083 cd mydir26

1084 ls

1085 cd..

1086 cd ..

1087 mv /root/mydir24/1.txt 2.doc /root/mydir26

1088 mv /root/mydir24/*.doc /root/mydir26

1089 mv /root/mydir24/*.* /root/mydir26

1090 ls /root/mydir24

1091 ls /root/mydir26

1092 ls -la /root/mydir26

1093 useradd bala123

1094 cat /etc/passwd

1095 ls -la

1096 ls /root/mydir3

1097 mkdir mydirdel

1098 rmdir mydirdel

1099 ls

1100 mkdir mydirdel1

1101 cd mydirdel1

1102 mkdir dirdel2

1103 cd dirdel2

1104 mkdir dirdel3

1105 touch one two three four

1106 ls

1107 history

Session 3 (7pm-9pm) 01st May 2023

Recap

Who is the father of DevOps?

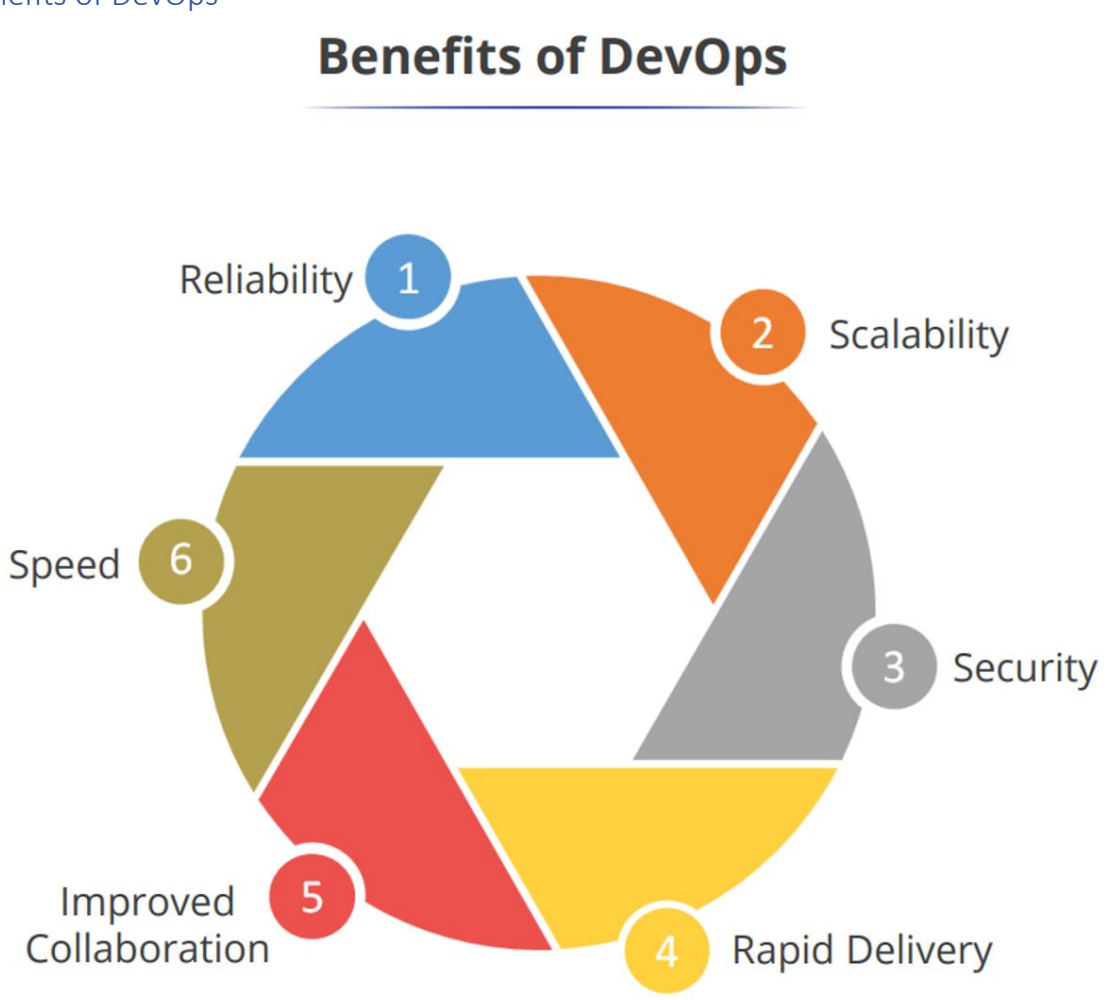
- Patrick Debois
- Also in October of 2009, **Patrick Debois**, often called "The Father of DevOps," held the first DevOpsDays conference in Ghent, Belgium. It was described as, "The conference that brings development and operations together." This is where the term "DevOps" was first used.

DevOps Vs Agile

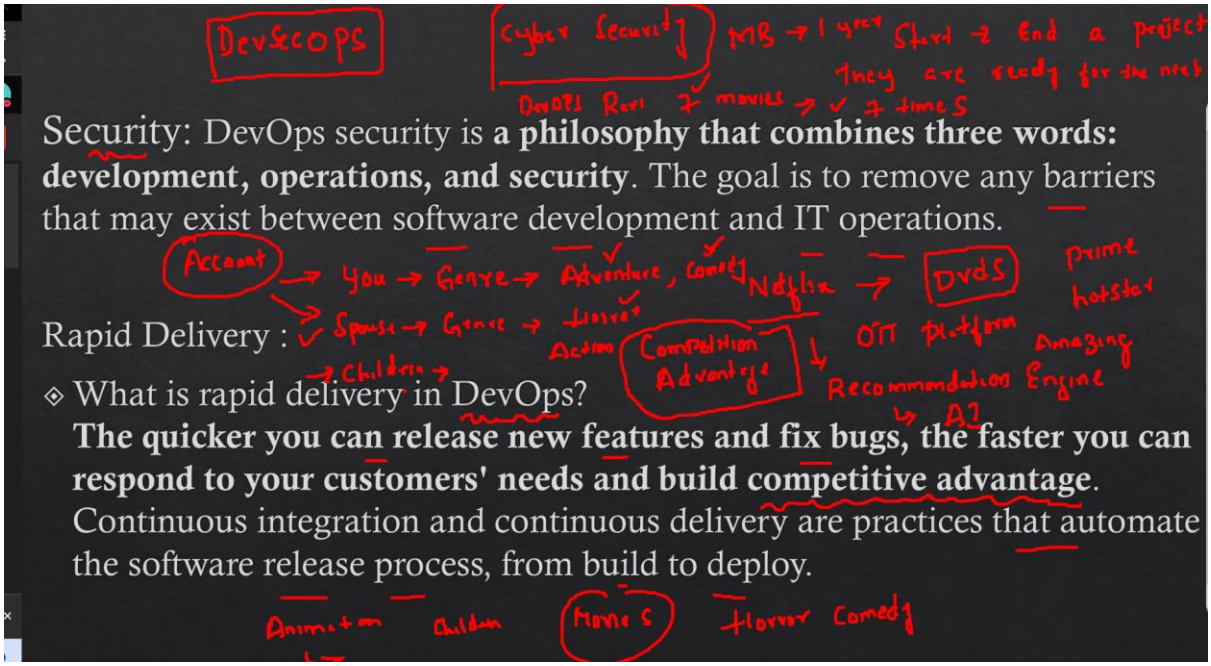
What is DevOps vs agile?

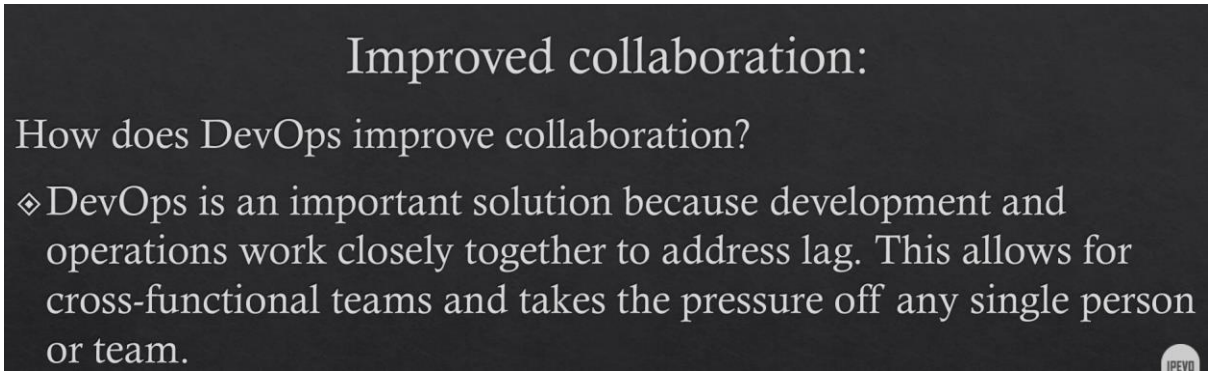
- Difference between Agile and DevOps.
- The key difference between Agile versus DevOps is that **Agile is a philosophy about how to develop and deliver software, while DevOps describes how to continuously deploy code through the use of modern tools and automated processes**.

Benefits of DevOps

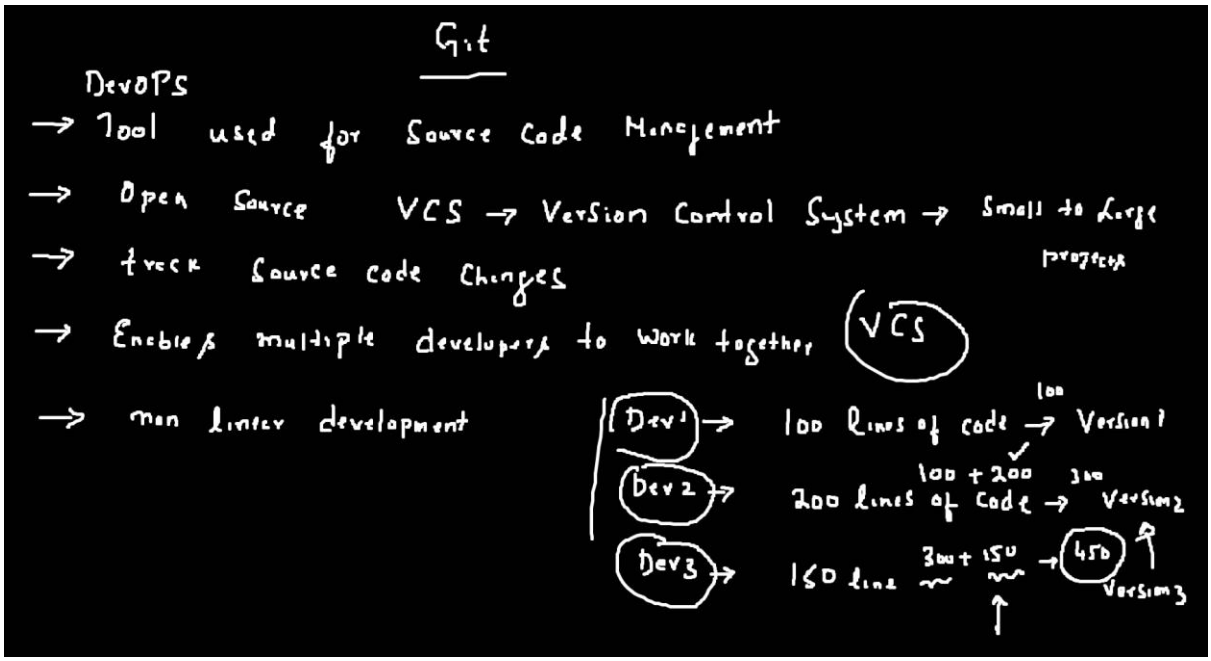


- Reliability: We have a set of tools (automation), if there is human intervention then it could be error-prone. So we can rely heavily on DevOps
- Scalability: Scalability refers to the ability of the business to set up its systems to grow during times of high demand and scale back when demand decreases (Increase and decrease the number of resources). **The optimum environment for achieving the appropriate level of scalability is DevOps, thanks to specific techniques**.

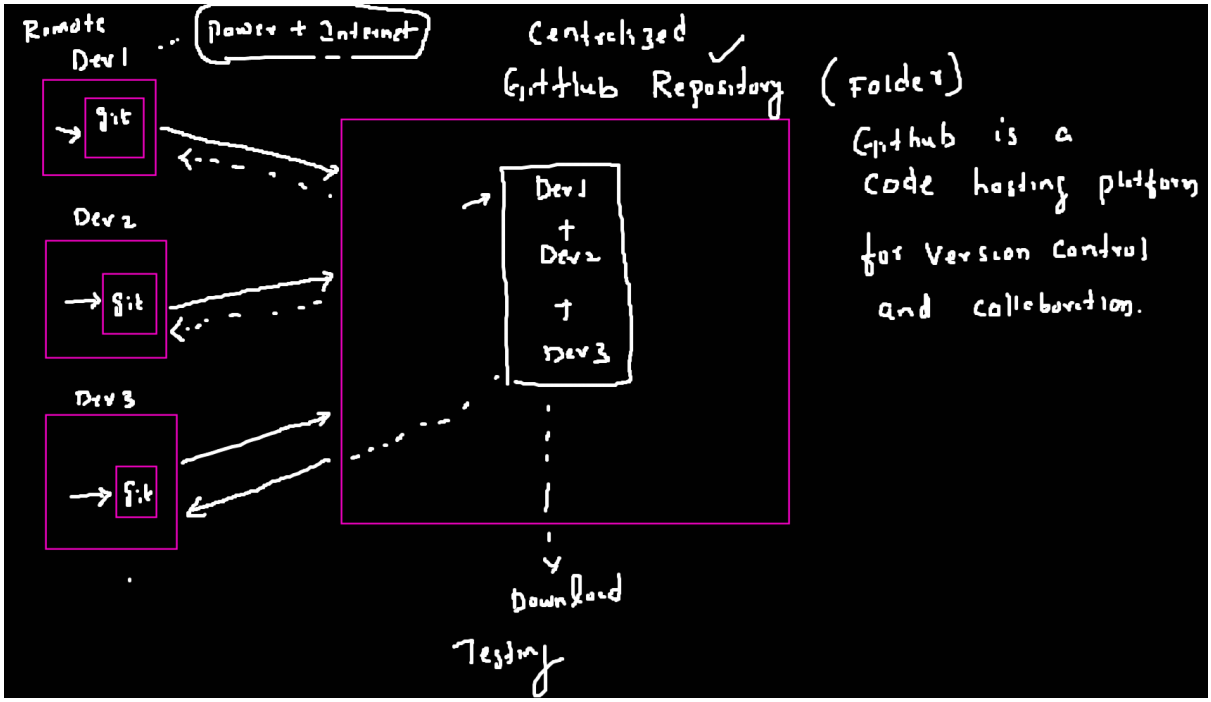

DevSecOps
Cyber Security
Security: DevOps security is a philosophy that combines three words: development, operations, and security. The goal is to remove any barriers that may exist between software development and IT operations.
Rapid Delivery :
What is rapid delivery in DevOps?
The quicker you can release new features and fix bugs, the faster you can respond to your customers' needs and build competitive advantage. Continuous integration and continuous delivery are practices that automate the software release process, from build to deploy.


Improved collaboration:
How does DevOps improve collaboration?
DevOps is an important solution because development and operations work closely together to address lag. This allows for cross-functional teams and takes the pressure off any single person or team.

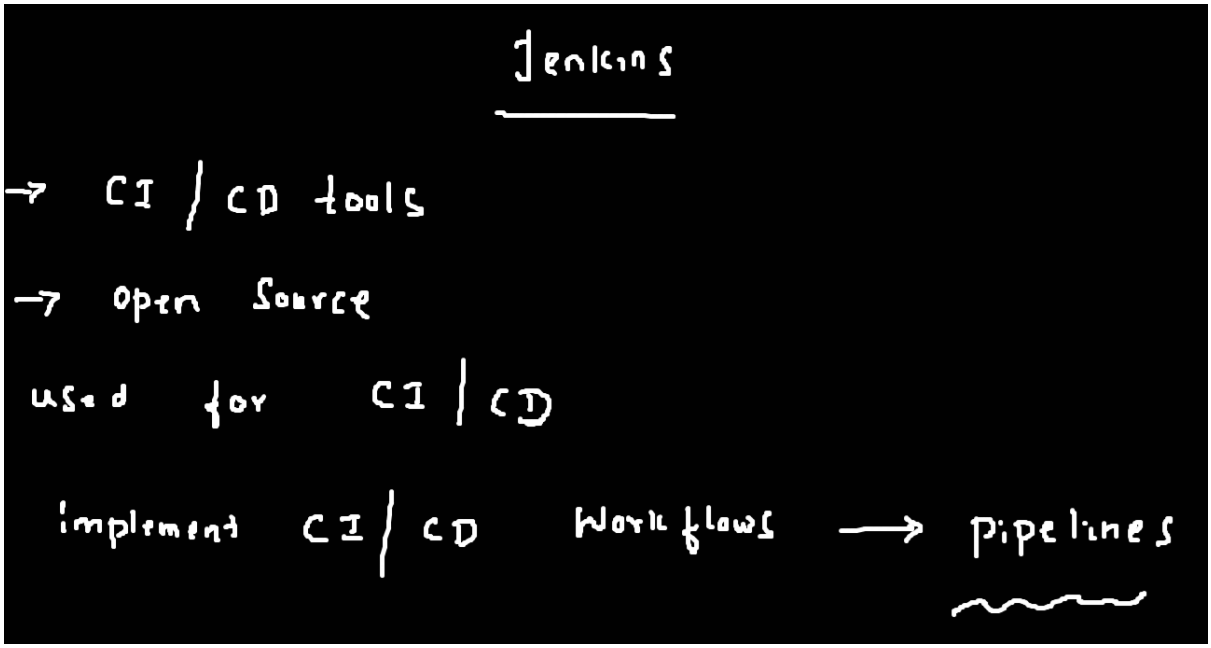
Git


Git
DevOPS
→ Tool used for Source Code Management
→ Open Source VCS → Version Control System → Small to Large projects
→ track Source code Changes
→ Enables multiple developers to work together
VCS
→ non linear development
Dev1 → 100 lines of code → Version1
Dev2 → 200 lines of Code → Version2
Dev3 → 150 lines → Version3

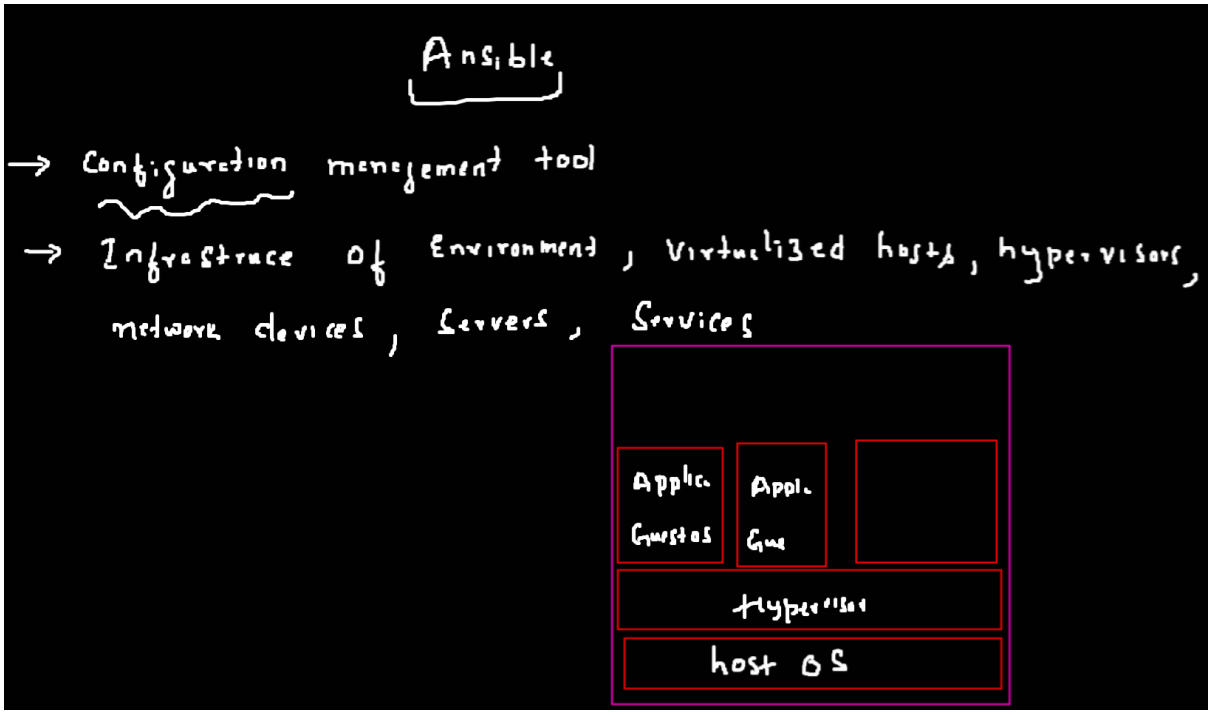
GitHub



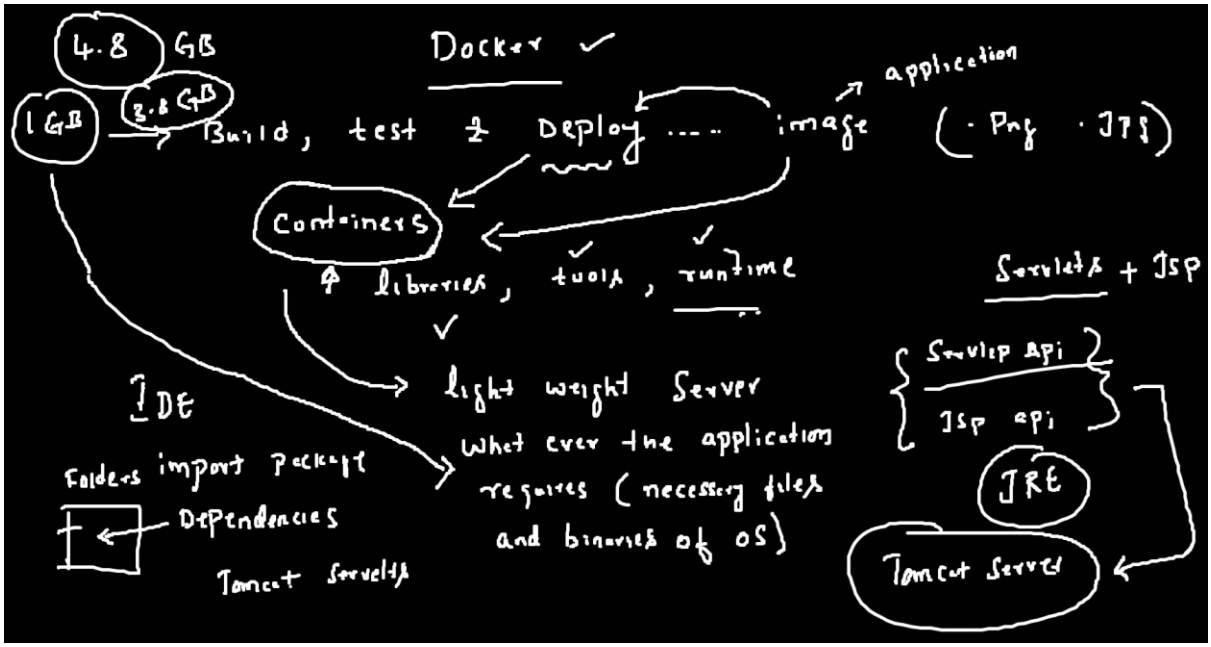
Jenkins



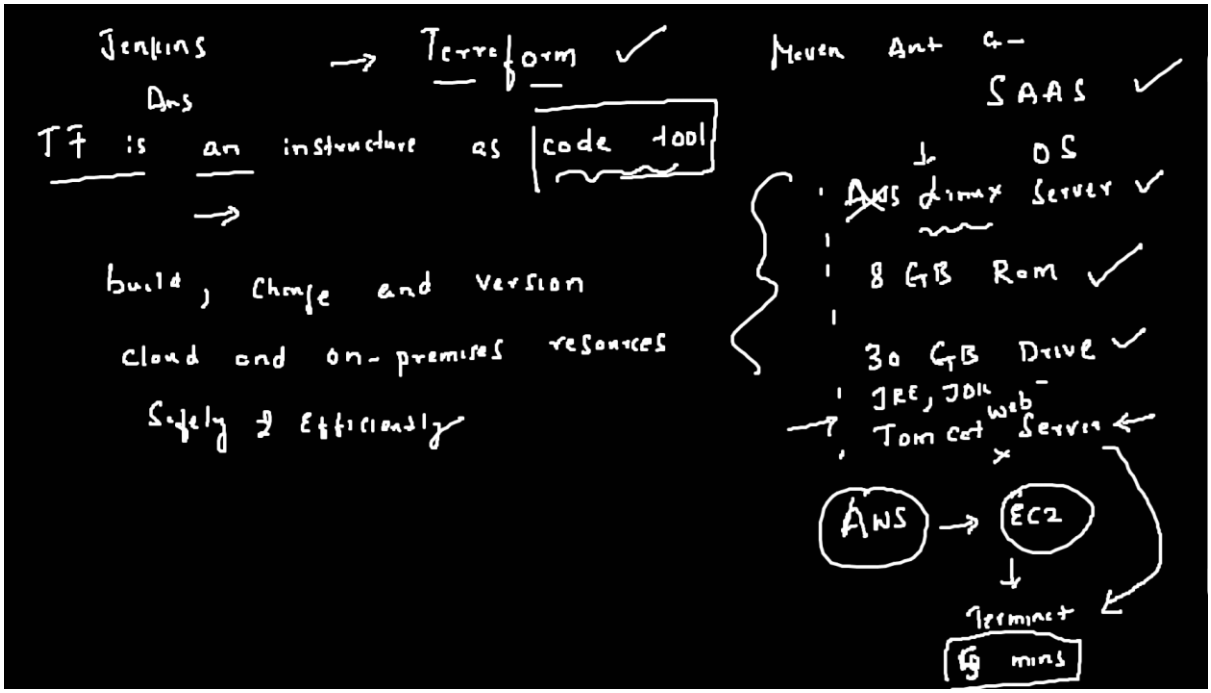
Ansible



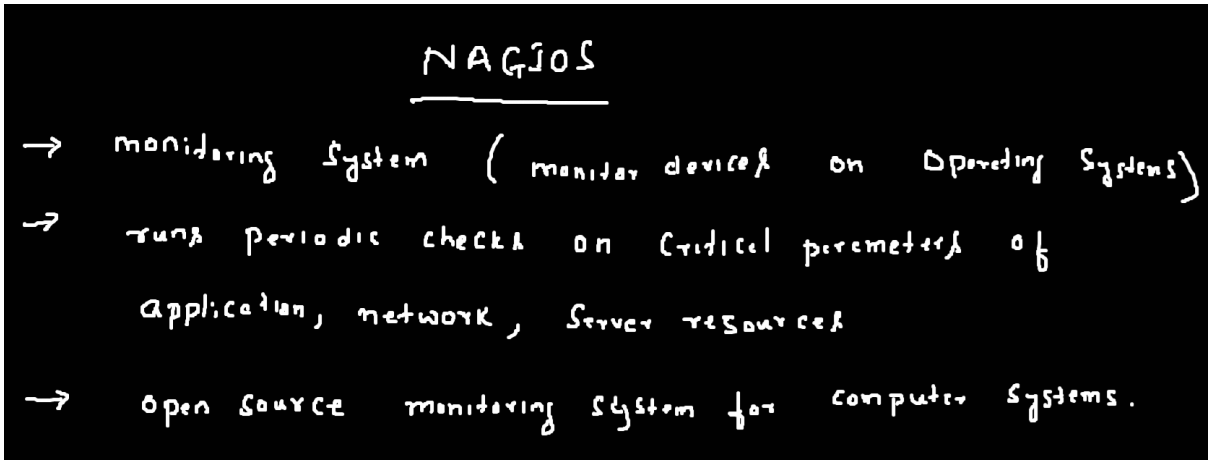
Docker



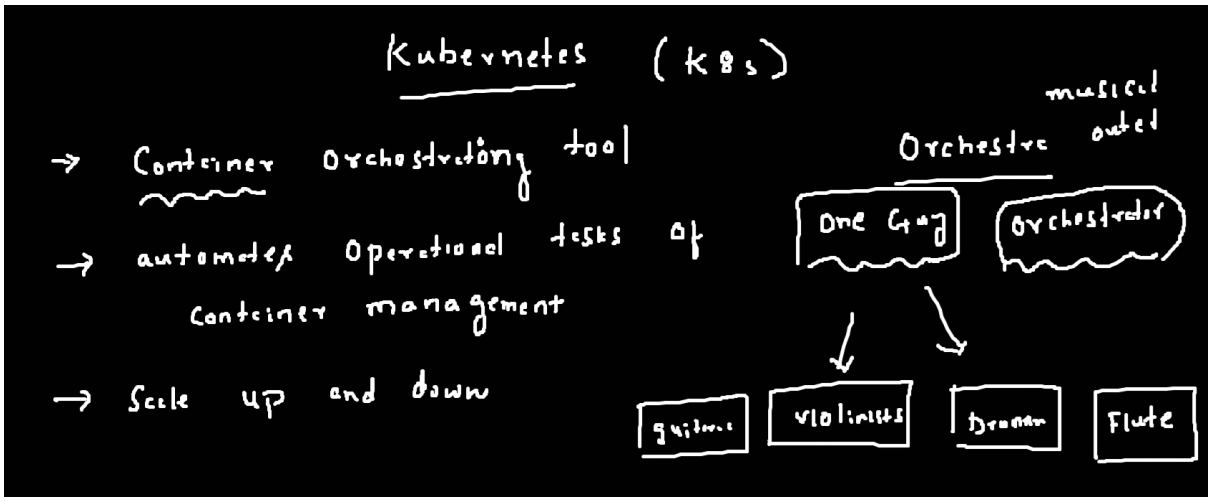
TerraForms



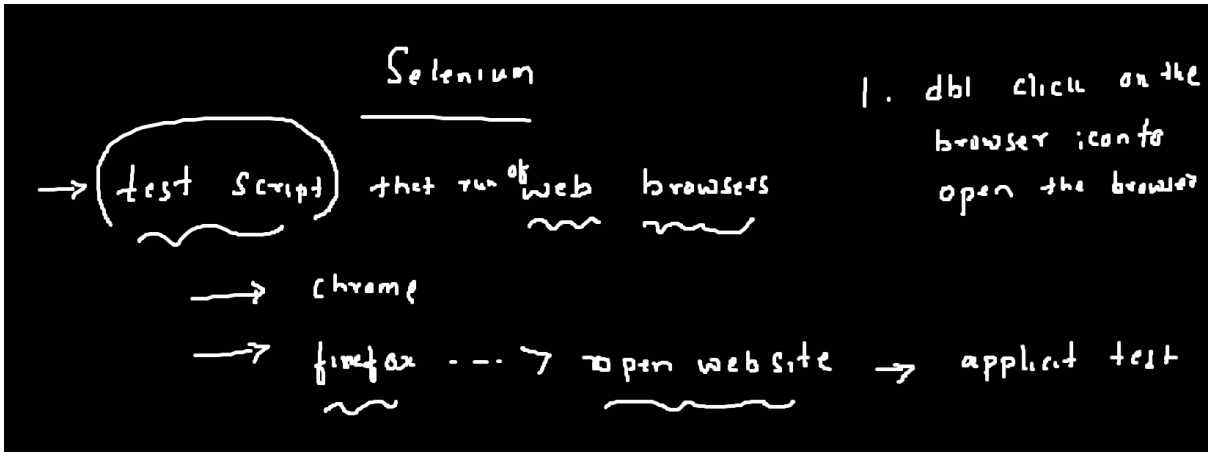
Nagios



Kubernetes



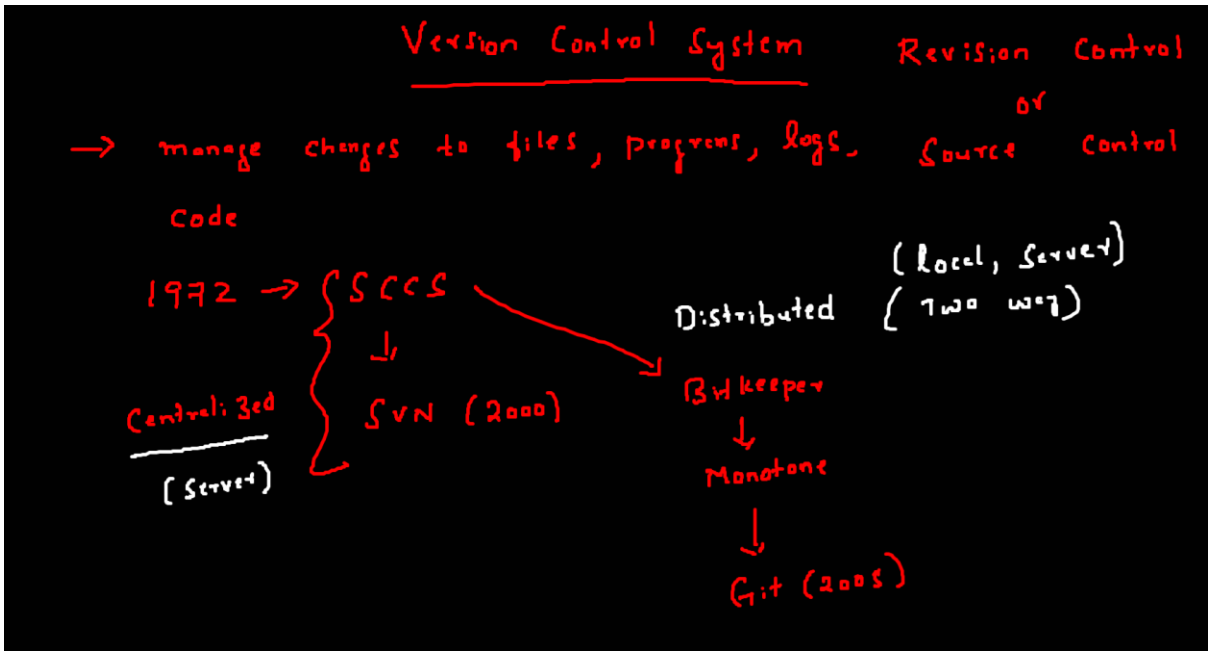
Selenium

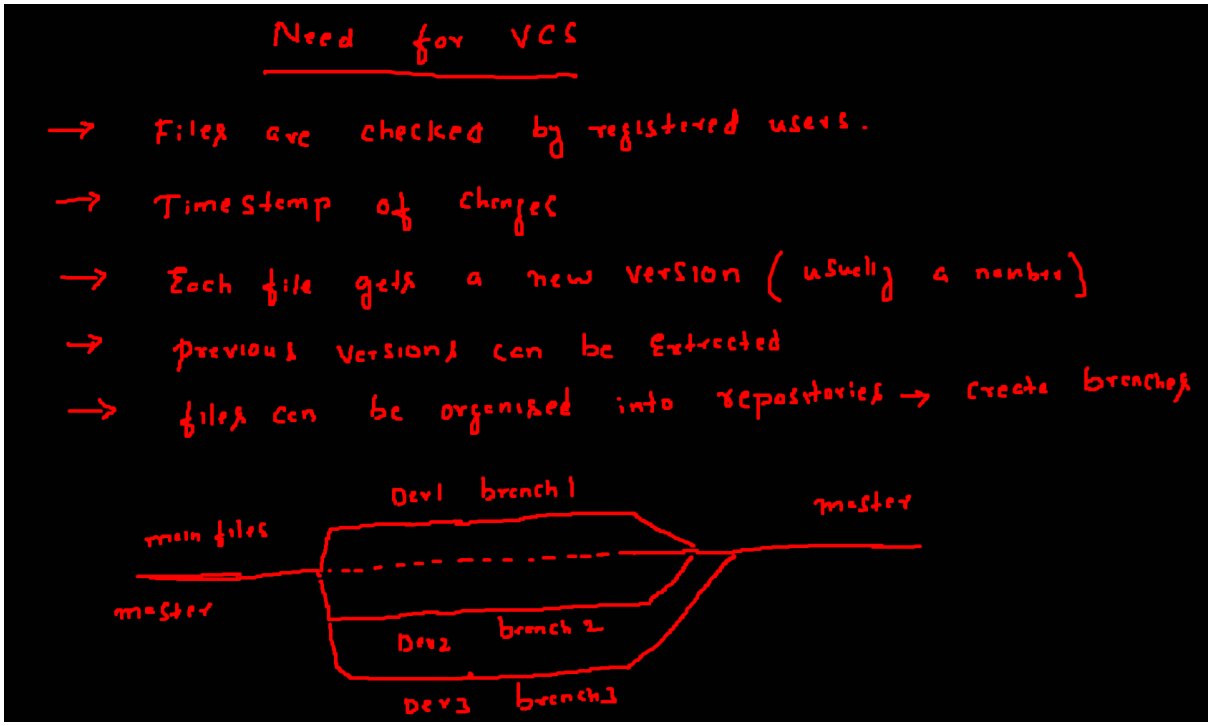


Session 4 (7pm-9pm) 02nd May 2023

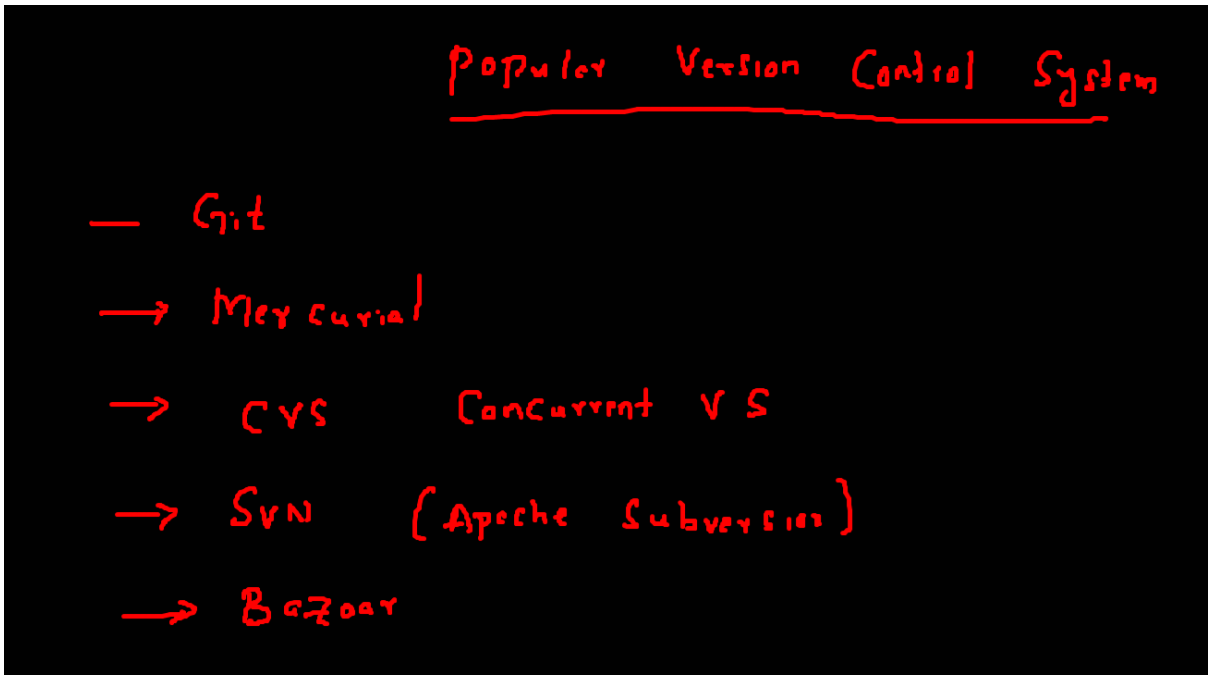
Recap

Version Control System

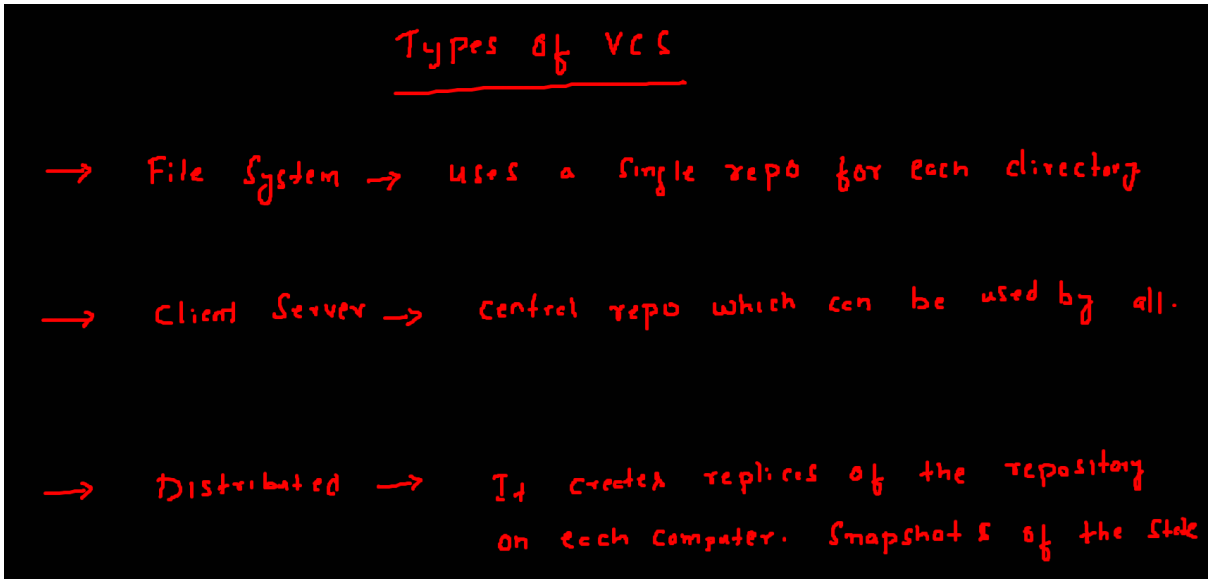



Need for VCS
→ Files are checked by registered users.
→ Timestamp of changes
→ Each file gets a new version (usually a number)
→ previous versions can be extracted
→ files can be organised into repositories → create branches
Devl branch1
master
main files
master
Dev2
branch 2
Dev3
branch3

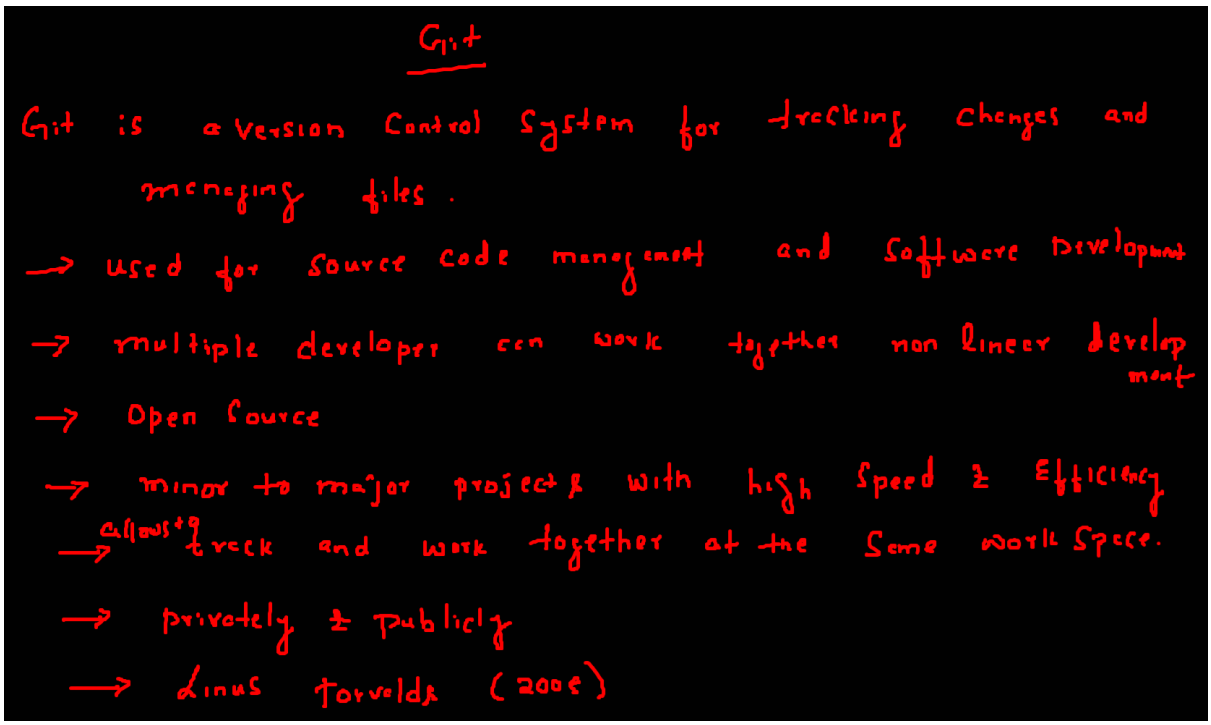
Popular VCS


Popular Version Control System
— Git
→ Mercurial
→ CVS Concurrent VS
→ SVN (Apache Subversion)
→ Bazaar

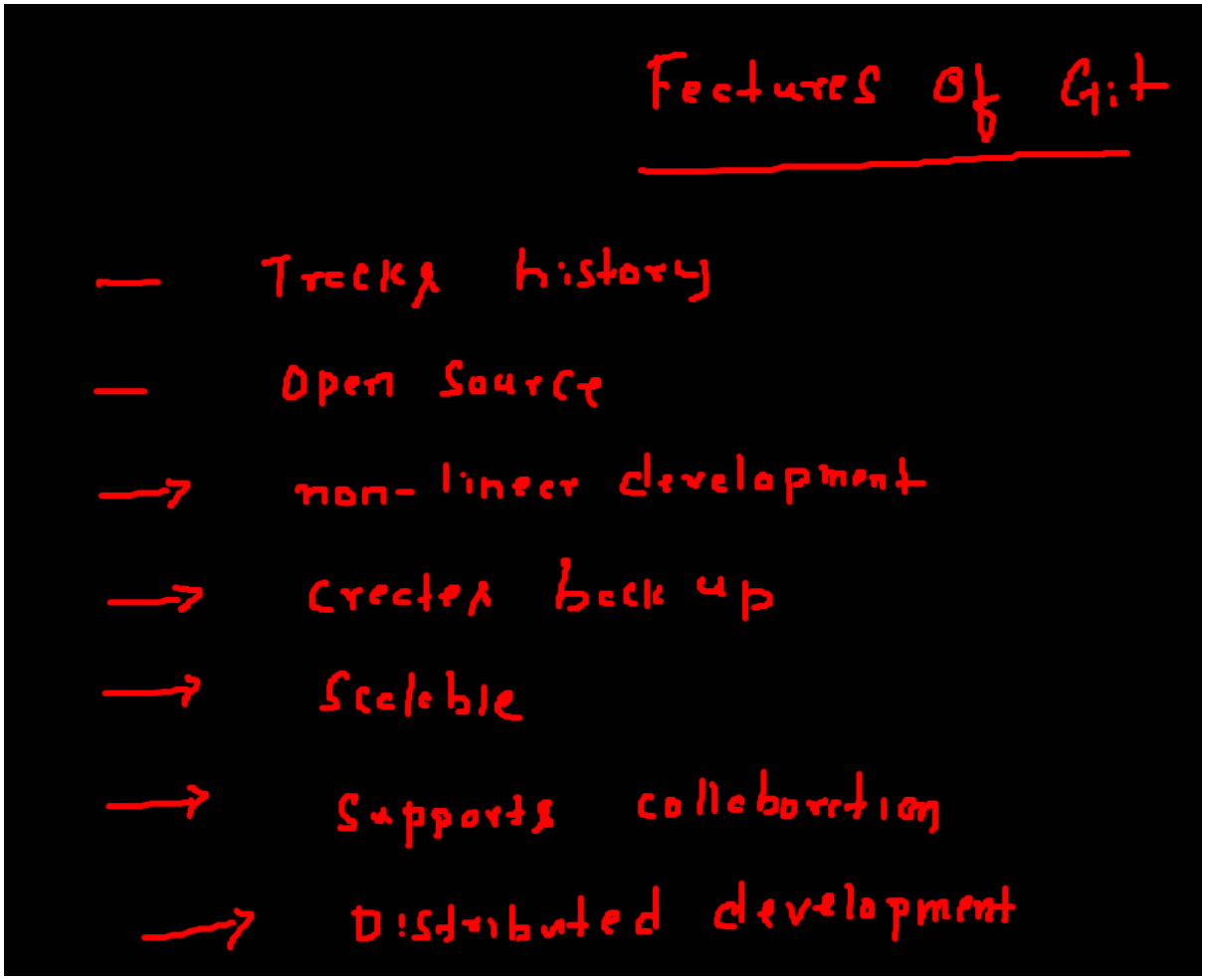
Types of VCS

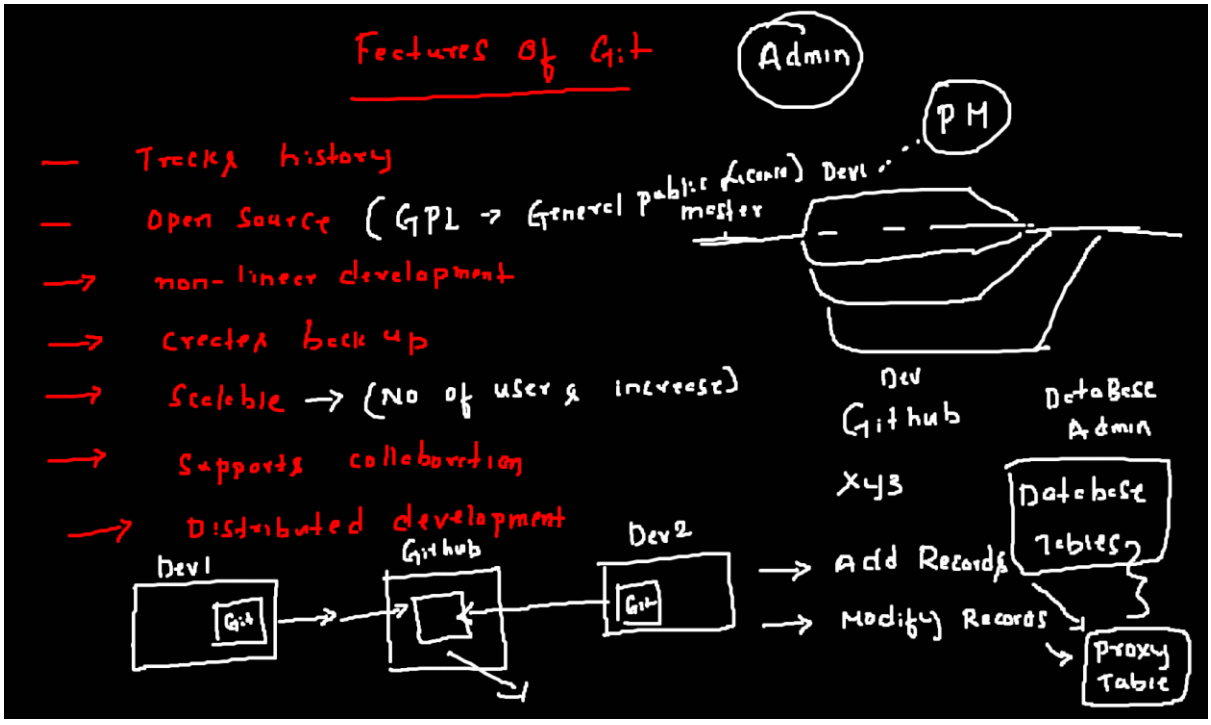


Git



Features of Git



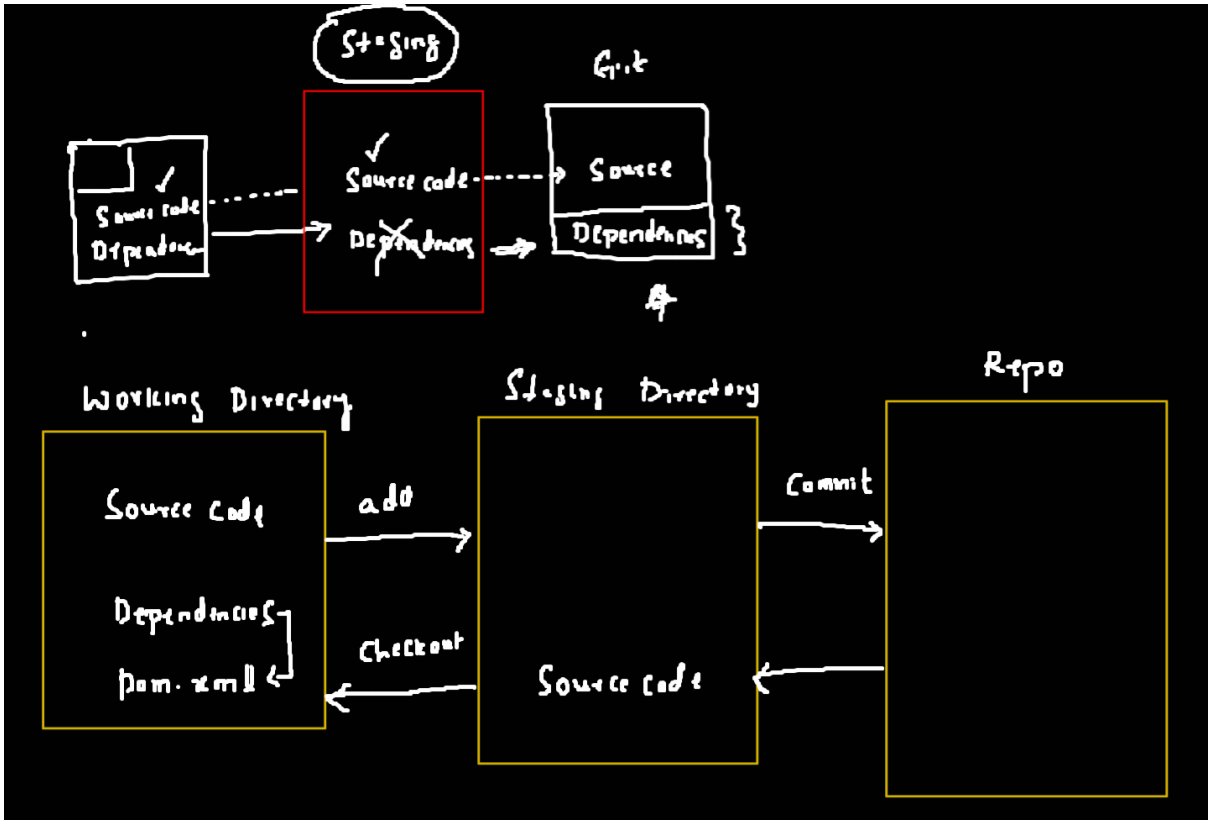


Security : Git uses SHA1 to name and identify objects within its repo. Files and commits are checked and retrieved by its checksum at the time of checkout.

Commit id

Once published, one cannot make changes to its old version

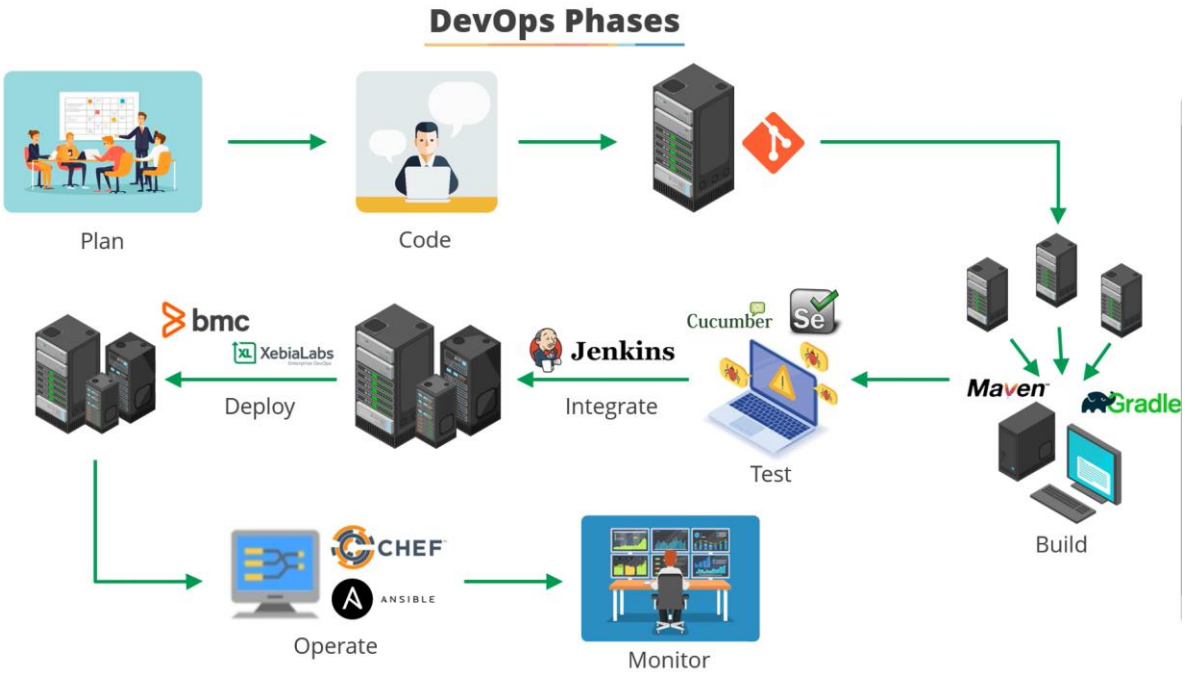
Git working Directory-> Staging Directory-> Local Repo



Comparison of VCS

Comparison of Version-Control Software			
Software	**Repository Model**	**Concurrency Model**	**Platforms Supported**
CVS	Client-server	Merge	Unix-like, Windows, OS X
Git	Distributed	Merge	POSIX, Windows, OS X
SVN	Client-server	Merge or lock	Unix-like, Windows, OS X
Mercurial	Distributed	Merge	Unix-like, Windows, OS X
Monotone	Distributed	Merge	Unix-like, Windows, OS X

DevOps Phases



About READMEs

You can add a README file to a repository to communicate important information about your project. A README, along with a repository license, citation file, contribution guidelines, and a code of conduct, communicates expectations for your project and helps you manage contributions.

For more information about providing guidelines for your project, see "Adding a code of conduct to your project" and "Setting up your project for healthy contributions."

A README is often the first item a visitor will see when visiting your repository. README files typically include information on:

- What the project does
- Why the project is useful
- How users can get started with the project
- Where users can get help with your project
- Who maintains and contributes to the project

If you put your README file in your repository's hidden `.github`, root, or `docs` directory, GitHub will recognize and automatically surface your README to repository visitors.
If a repository contains more than one README file, then the file shown is chosen from locations in the following order: the `.github` directory, then the repository's root directory, and finally the `docs` directory.
If you add a README file to the root of a public repository with the same name as your username, that README will automatically appear on your profile page. You can edit your profile README with GitHub Flavored Markdown to create a personalized section on your profile. For more information, see "Managing your profile README."

You can configure Git to ignore files you don't want to check in to GitHub.

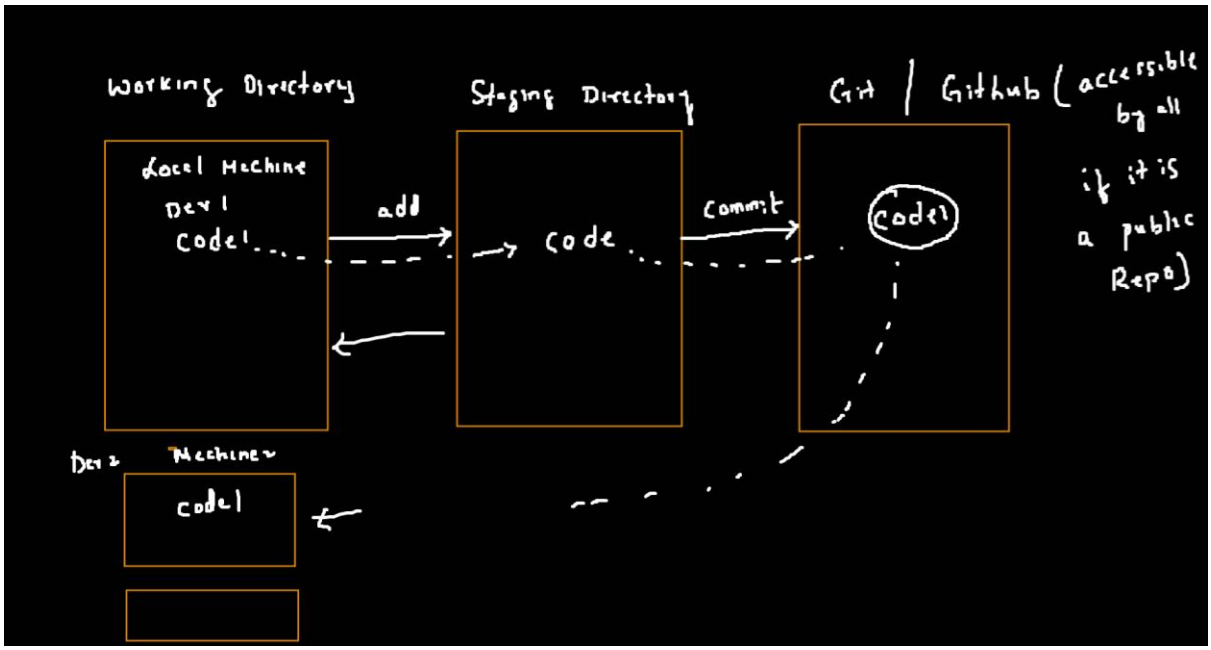
MacWindowsLinux

Configuring ignored files for a single repository

You can create a .gitignore file in your repository's root directory to tell Git which files and directories to ignore when you make a commit. To share the ignore rules with other users who clone the repository, commit the *.gitignore* file in to your repository.

GitHub maintains an official list of recommended *.gitignore* files for many popular operating systems, environments, and languages in the `github/gitignore` public repository. You can also use gitignore.io to create a *.gitignore* file for your operating system, programming language, or IDE. For more information, see "github/gitignore" and the "gitignore.io" site.

Session 5 (7pm-9pm) 03rd May 2023



https://github.com/codewithbala/git0305/blob/main/README.md