Lec-36 | What is (CI)/CD Pipeline? What is Jenkins?

Before Continuous Intigration
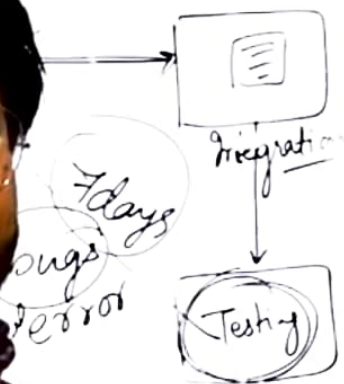
Repository

Developer

github
DVCS

7days

bugs

3days

and error

Integra...

Testi...

After Continuous Integration
(fetch changes) notification

updated

Source Code
Repo

Build | Test | Deploy

CI Server

notify/success/failure

Continuous Integration

Continuous Build + Continuous Test

Before Continuous Integration

After Continuous Integration
Fetch changes   notification

updated

Build Test Deploy
CI Server

Source code
Repo

Developer

notify / success / failure

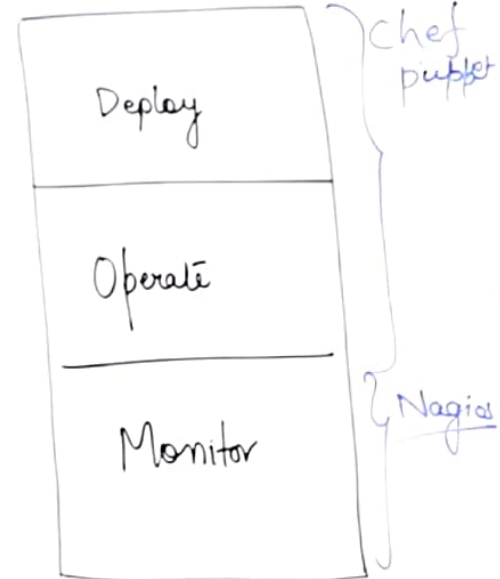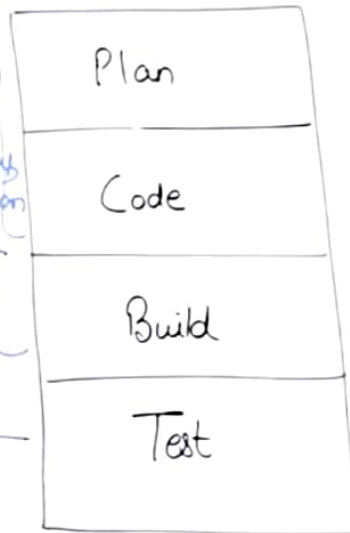Integration

7days
bugs
error
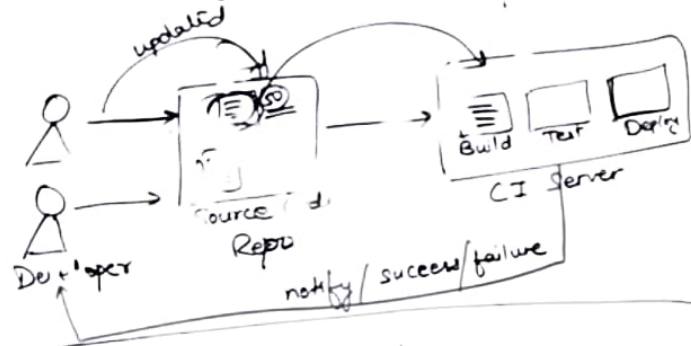
Testing

Continuous Integration
= Continuous Build + Continuous Testing

Lec-36 | What is (CI)/CD Pipeline? What is Jenkins?
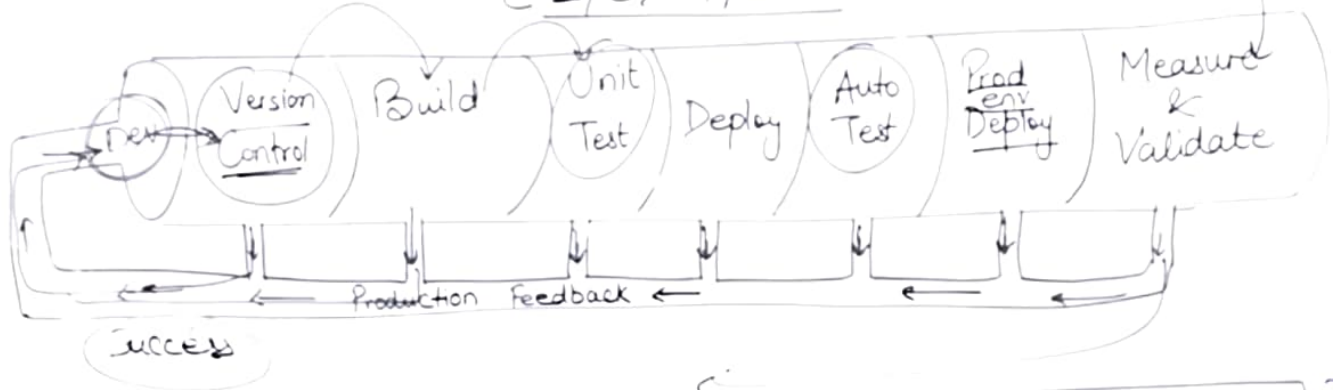
CI/CD Pipeline

QA

Test → Version Control → Build → Unit Test → Deploy → Auto Test → Prod env Deploy → Measure & Validate

Production feedback ←

Success

---

code, Plan, Deploy, Release, Monitor, Operate — github, Subversion

Maven, Gradle →

Selenium, Junit

Plan
Code
Build
Test

Deploy
Operate
Monitor

Chef, puppet

Nagios

Lec-36   What is (CI)/CD Pipeline? What is Jenkins?

CI/CD Pipeline

QA

Version Control — Build — Unit Test — Deploy — Auto Test — Prod env Deploy — Measure & Validate

Production Feedback ←
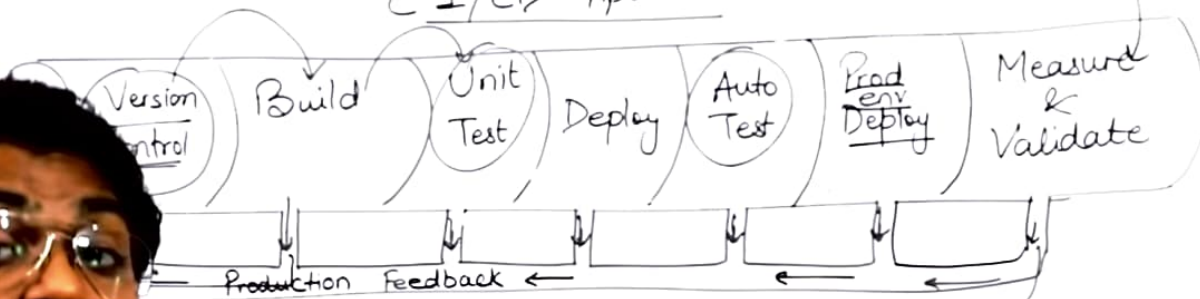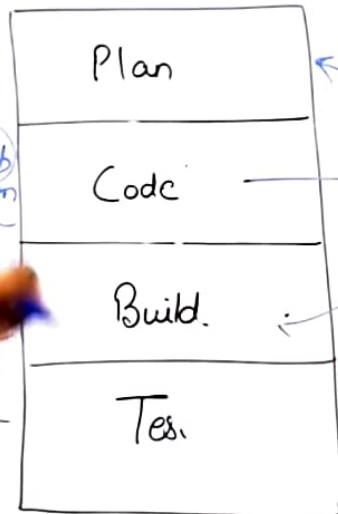
Bamboo
Travis CI
Buildbot

Plan
Code
Build.
Tes.

Integration Tool

Deploy
Operate
Monitor

chef
puppet

Nagios

Github
Subversion

JENKINS

→ Jenkins is an Open-source project Written in Java that runs on Windows, macOS and Other Unix-like Operating Systems It is Community Supported and might [be the] first choice tool for CI

→ Jenkins automate the entire Software deve[lop]-ment life Cycle

→ Jenkins was Originally developed by Sun Microsystem in 2004 under the name hudson

→ The project was later named Jenkins when Oracle bought Microsystems

→ It Can run on any major platform without any Compatibility issues

→ Whenever developers Write Code, we integrate all that Code of all developers at that [point] of time and we build, test an[d] the Client This [   ]eve this [   ] will be [   ]ectified fast [   ] development

→ Jen[kins   ]

JENKINS

→ Jenkins is an Open-source project Written in Java that runs on Windows, macOS and Other Unix-like Operating Systems. It is free, Community Supported and might be your first choice tool for CI.

→ Jenkins automate the entire Software development life Cycle

→ Jenkins was Originally developed by Sun Microsystem in 2004 under the name

hudson ← Jenkins (free) Hudson (Enterprise Edition)

→ The project was later named Jenkins when Oracle bought Microsystems

→ It Can run on any major platform without any Compatibility issues.

→ Whenever developers Write Code, we integrate all that Code of all developers at that point of time and we build, test and deliver/Deploy to the Client This process _____ CI/CD _____ this.

→ Jenkins _____ be

→ Because _____ feet reported _____ opment So the _____ happens

JENKINS

→ Jenkins is an Open-source project Written in Java that runs on Windows, macOS and Other Unix-like Operating Sy_____ ____ free, Community Supported first choice tool for

→ Jenkins automate the ___ -ment life Cycle

→ Jenkins was Originally devel___ Microsystem in 2004 under ____ hudson ___ Jen___

→ The project ___ when Oracle

→ It Can run___ any Comp___

→ Whenever developers Write Code, we integrate all that Code of all developers, at that point of time and we build, test and deliver/Deploy to the Client This process is Called CI/CD
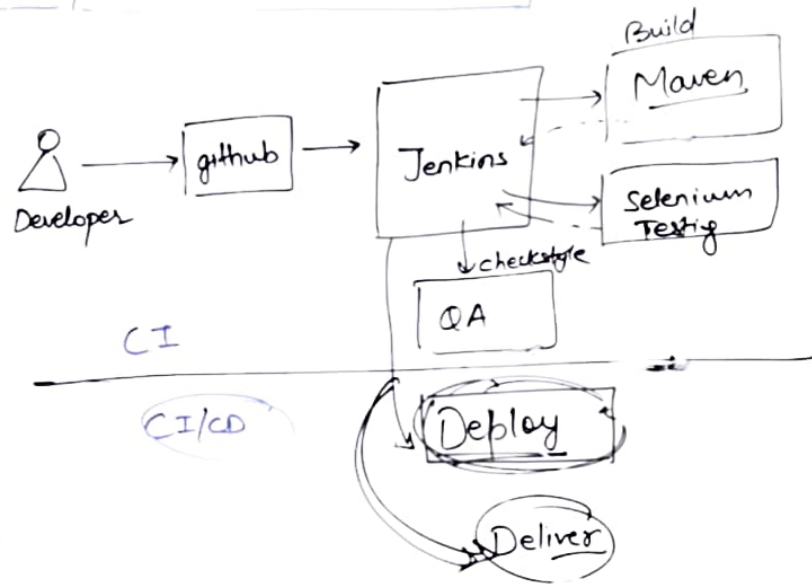
→ Jenkins helps us to achieve this.

→ Because of CI, Nows bugs error will be reported fast and get rectified fast So the entire Software development happens fast

Workflow of Jenkins

→ We can attach ~~~~ Maven, Selenium
and A~ ~ Jenkins

→ Once ~~~~ in github,
Jenkins ~~~~ Send to
Maven

→ Once build~~~~ ins full
that Code~~~~ Selenium
for te~~

→ ~~~~ Jenkins
Send

Developer → github → Jenkins → Maven (Build)
→ Selenium Testing
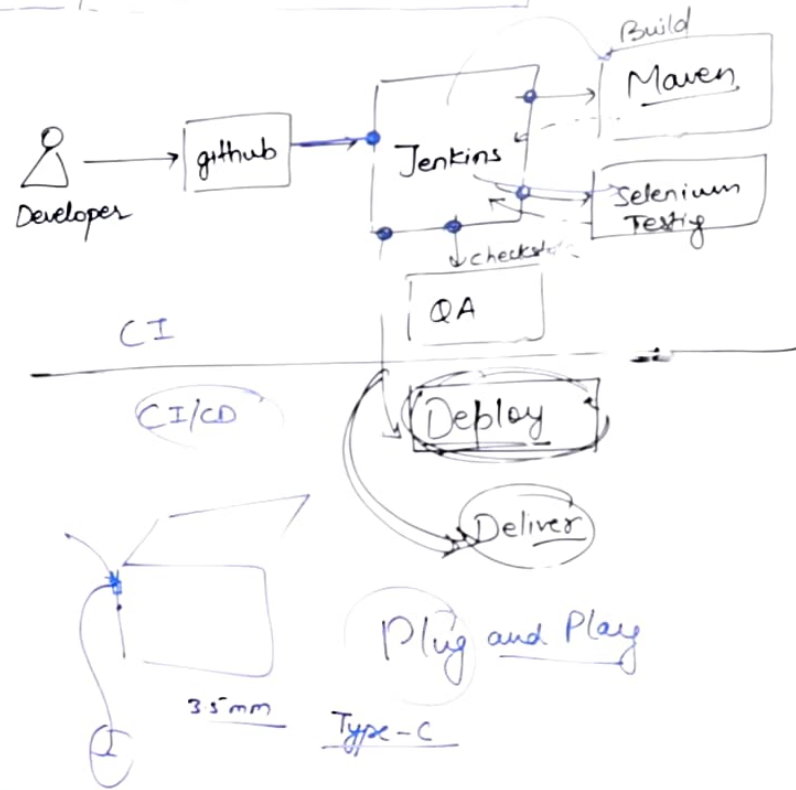↓ checkstyle
QA

CI

CI/CD

Deploy

Deliver

| Lec-36 | What is (CI)/CD Pipeline? What is Jenkins? |

Workflow of Jenkins

→ We can attach git, Maven, Selenium
and Ar[...] [...] to Jenkins

→ Once [...] in github,
Jenkins Send to
Maven

→ Once bu[...] [...]nkins full
that Cod[...] to Selenium
for t[...]

[...]nkins

Developer → github → Jenkins → Maven **Build**

Jenkins → Selenium Testig

checks[...]

QA

CI

CI/CD

Deploy

Deliver

Plug and Play

35mm   Type-C

Workflow of Jenkins

→ We Can attach git, Maven, Selenium and Artifactory (plugins) to Jenkins

→ Once developers puts Code in github, Jenkins pull that Code & Send to Maven for Build

→ Once build is done, jenkins pull that Code and Send to Selenium for testing

→ Once testing is done, then jenkins will pull that Code and Send to (artifactory) as per requirement and so on. Archiving purpose

→ We Can also deploy with Jenkins
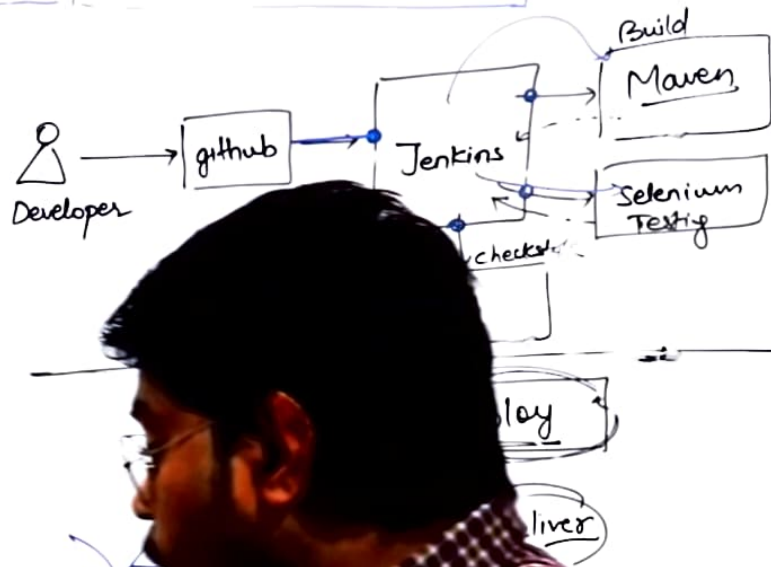
Build
Maven

Selenium
Testing

Developer → github → Jenkins
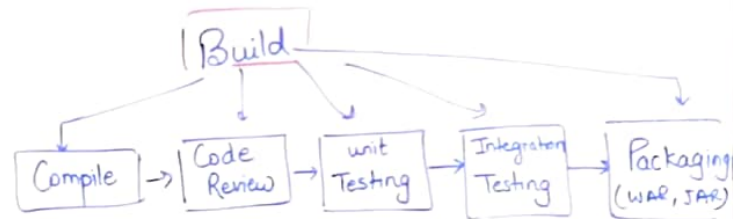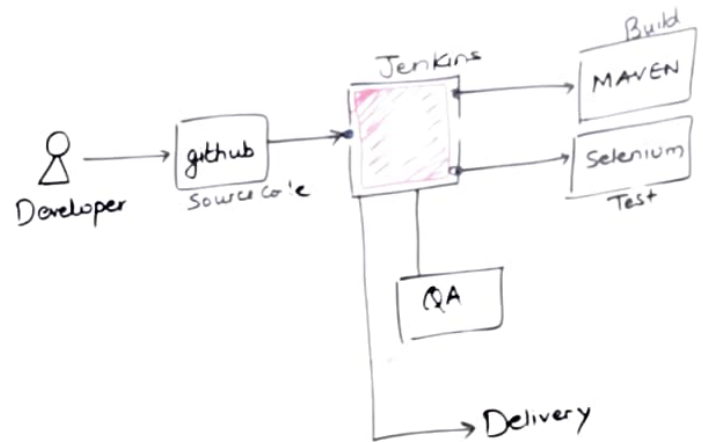
checks

loy

liver

3.5

Advantages of Jenkins

→ It has lots of (plug-ins) available
→ You can write your own plug-in
→ You can use Community Plug-in
→ Jenkins is not just a tool. It is a framework
  ie You can do whatever you want. All
  you need is plug-ins

→ We can attach Slaves (nodes) to Jen
  master It instruct others (slaves) to do
  Job If Slaves are not available,
  Jenkins itself does the Job.

→ Jenkins also behave as Crone Server
  Replacement. ie Can do Scheduled task.

→ It Can Create Labels.

Jenkins

Build
MAVEN

Selenium
Test

→ Delivery

Packaging
(WAR, JAR)

Advantages of Jenkins

→ It has lots of (plug-ins) available
→ You can write your own plug-in
→ You can use Community ~~~~~~~work
→ Jenkins is not just a ~~~~
  ie You can do whatever
  you need is plug-~~~

→ We can attach Slaves
  master It instruct oth
  Job If Slaves
  Jenkins

→ Jenkins
  Replaceme

→ It

Build

Jenkins

MAVEN

Selenium

Test

Developer → github
Source Co'e

QA

→ Delivery

| Build |

Compile → Code Review → unit Testing → Integration Testing → Packaging (WAR, JAR)

Jenkins

→ It _____ available
→ You _____ plug-in
→ You _____ in
→ Jenkin _____ It is a framework
  ιe You _____ you want. All
  you

→ We _____ (nodes) to Jenkins
  _____ (slaves) to do
  _____ lable,

Developer → github (Source code) → Jenkins [1] [2] → MAVEN (Build) / Selenium (Test)

Master → Slave, Slave, Slave

QA

→ Delivery

Build
Compile → Code Review → unit Testing → Integration Testing → Packaging (WAR, JAR)