https://coderuniversetechnology.info/

https://www.facebook.com/profile.php?id=61569717476118

https://www.instagram.com/coder_universe_technology/

Contact us on: 7709701108

Devops with Linux AWS AI 07.00 AM

==========

Section 1

==========

what is Devops ?

Why is Devops ?

Devops Lifecycle ?

SDLC ?

Waterfall Methodology ?

Agile Methodology ?

Section 2

==========

Devops Tools

Roles of DevOps Engineer

Maven and Github

Jenkins and SonarQube

Docker & Kubernetes

ELK (Elastic search, Log Stash and Kibana) => Prometheus and Grafana

Terraform- Infrastructure as Code

Ansible

Section 3

==========

Java Project Deployment

React Project Deployment etc

Section 4

==========

Resume Preparation

Interview Questions

Mock Interview

Section 5

==========

AWS

SDLC - Software Development Life Cycle

-> Requirement Gathering

-> Requirement Analysis

-> Design/Planning

-> Coding

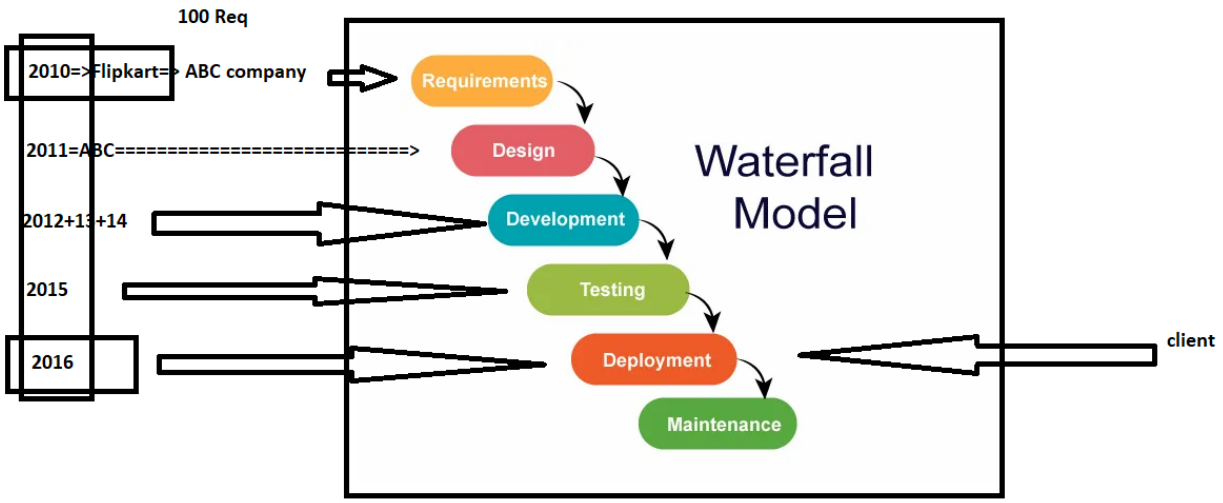
-> Testing

-> Deploy

-> Deliver

-> Support/Maintenance

WaterFall Model



-> Earlier people used to follow waterfall model to devlope the project

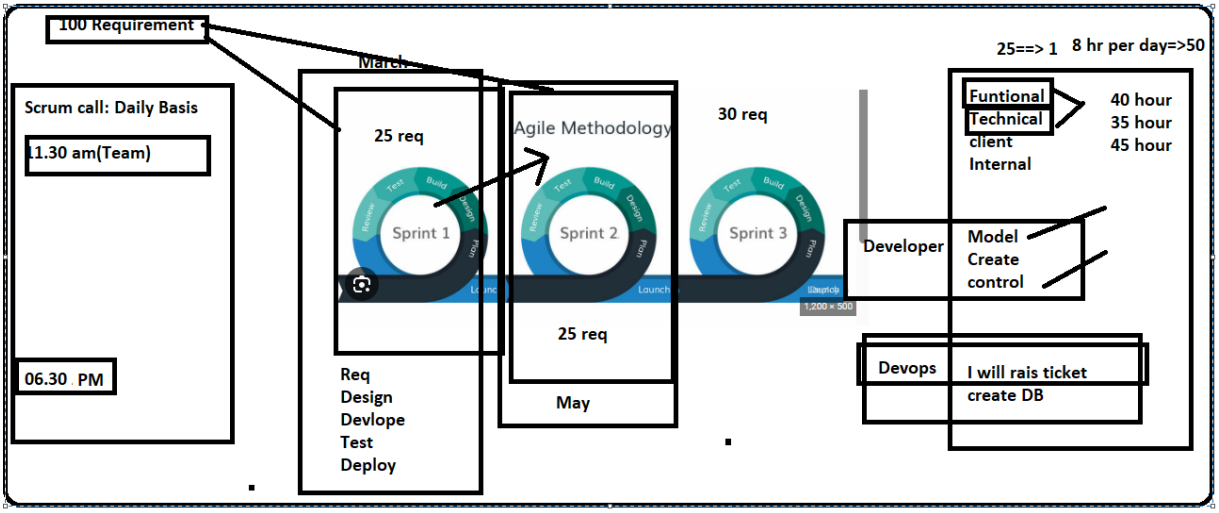
-> Waterfall is an linear methodolgy to develope and deliver the project.

-> Requirement are fixed.

-> Budget are fixed

-> client will see the project at end and their involvement is very less

Agile Methodology



Agile is an Iterative approach to develop and deliver the project

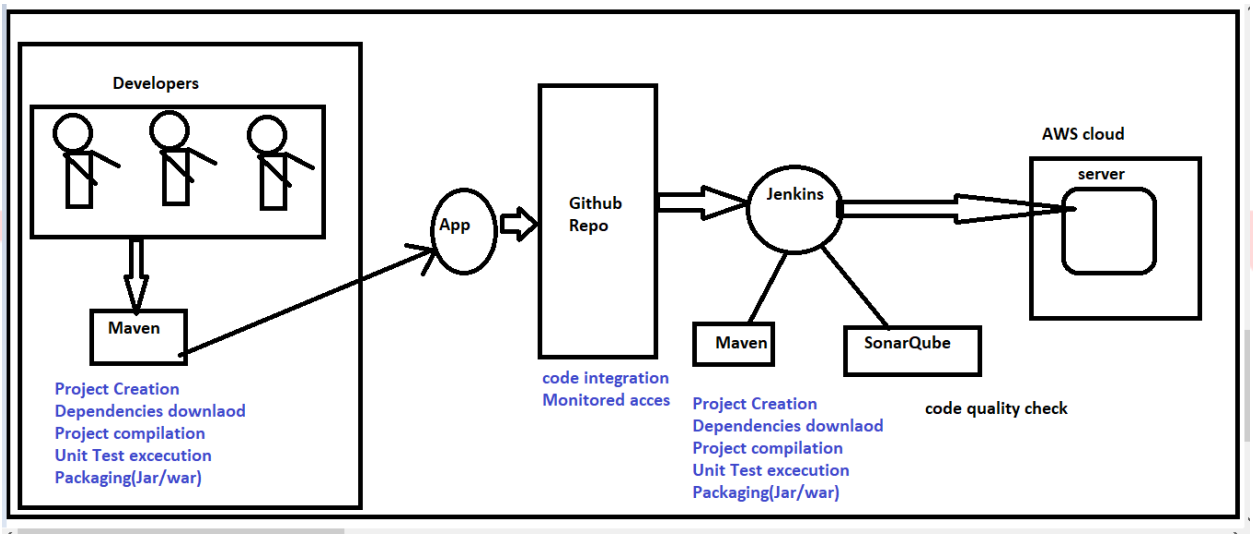
Development and testing can be done in paralled.

Client involvement will be very high.

For every release we will take client feedback.

Requirements are not fixed.

Budget is not fixed.



Maven

-> Maven is an free and open sources provided by Apache company.

-> Maven Software is developed using java software.

-> Maven is used to perform build the code for java project.

-> It is called as Java Build Tool.

-> We can create initial Project structure

-> we can include the dependencies

What is the use maven ?

-> We can create initial project folder structure using maven.

-> we can download project dependencies

(spring boot,hibernate,kafka,email,log4j,junit,mysql,security)

-> pom.xml file be available in project structure.

-> pom ->Project Object Model

-> pom.xml act as input file for maven software.

-> we can compile our project source code using maven.

compilation: converting java source code into byte code.

-> we can package our Java project as Jar or War

-> JAR -- Java Archive (standalone application)

War -- Web archive (Web application required war file)

-> Java is object oriented programming language.

-> Oracle is the company provide this Java s/w

-> Java class will .java class extension

-> Java -> jdk(java development kit)

jre (java runtime environment)

->we need to compile the java files using the command

javac Demo.java ====> Demo.class

-> we need to execute .class file to run the java program.

-> standalone java project will be packaged as JAR file.

->Web application will be packaged as WAR file.

Maven Installation

-> Download and install Java software

-> when we install Java we require two things

1. JDK
2. JRE

-> JDK contains set of tools to develop the java program

-> JRE contains environment to run the Java program.

-> url to install Java s/w : https://www.oracle.com/in/java/technologies/downloads/

-> set the java home variables

-> To download Maven we need to visit apache website

url : https://maven.apache.org/download.cgi

file name : apache-maven-3.9.9-bin.zip

-> Set MAVEN_HOME in system environment variables

MAVEN_HOME : C:\apache-maven-3.9.9-bin\apache-maven-3.9.9

-> set the path for Maven

path : C:\apache-maven-3.9.9-bin\apache-maven-3.9.9\bin

-> to check maven installed or not open command prompt

mvn -version

Maven Terminology

archetype -> It represent the type of project we want to create.

maven-archetype-quickstart :It represent standalone app.

maven-archetype-webapp :It represent web app.

groupId -> Company name or project name

artifactId -> It represent project or module name

packaging -> It represent how we want to package our java app(jar/war)

create standalone project

mvn archetype : generate -DgroupId=com.app.vk

-DartifactId=01-Maven-app

-DarchetypeArtifacId =maven-archtype-quickstart

-DintercativeMode=false

command to create stand alone application :

---

mvn archetype:generate -DgroupId=com.app.vk -DartifactId=01-Maven-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

How maven will download dependencies

-> There are three types of repositories

1. **Central Repository : This repo manage by Apache org.**
2. **Remote : Companies mainting their own repo.**
3. **Local : In local we have .m2**

Maven Goals

-> To perform some action on Maven project.

***clean : It is to clean or clear older jar/war inside target folder**

***compile: To comile the source code**

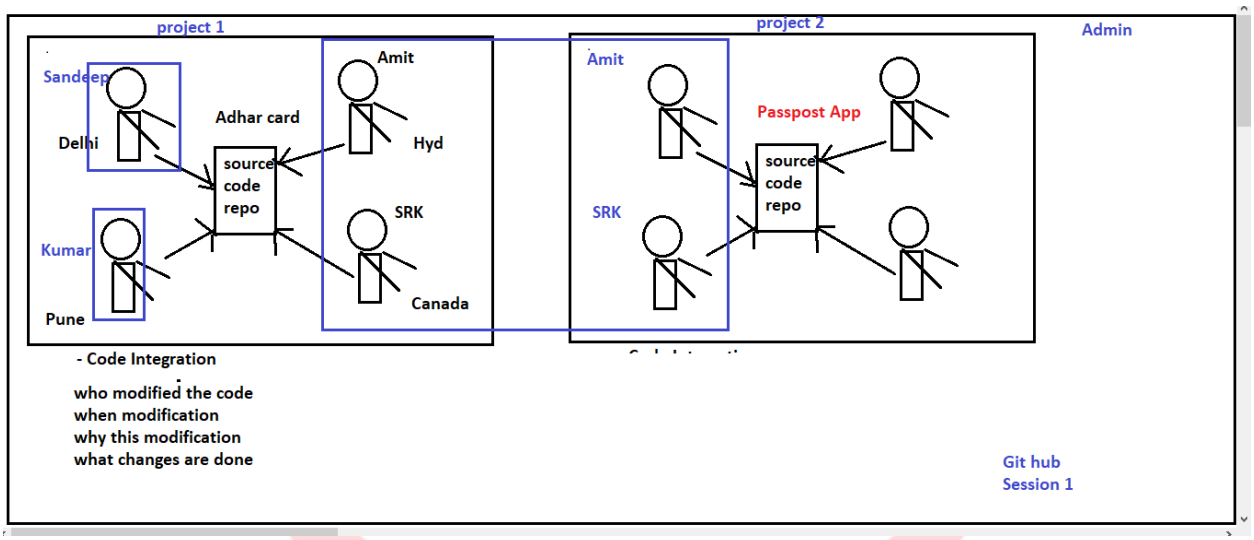
.java===>.class

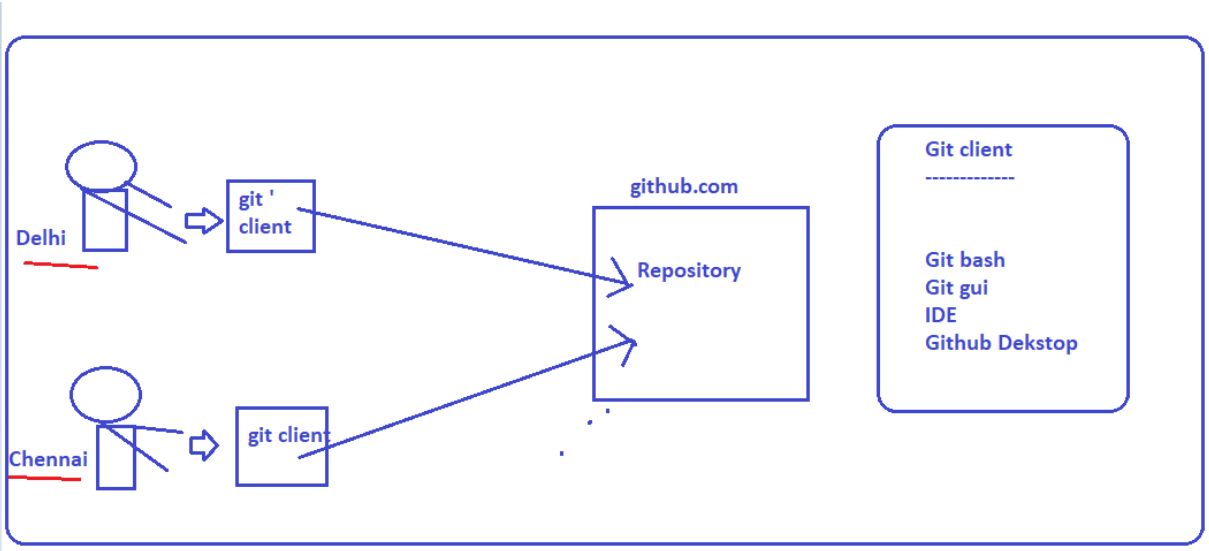
***test : To execute test code of application.**

***package: It is to generate jar/war files**

*install : It is used to install our project as a dependency in maven local repo.

Source Code Repository Tools





->Multiple developer are working on a project may be from same location or may be from different location.

-> One should not keep the code locally

-> All developer code should be store at one place and code integration should happen

-> code monitoring purpose

Advantage

who modified the code

when modification

why this modification

what changes are done

Repository Tools

SVN (outdated)

Git hub

Gitlab

Bitbuckets

Environment Setup to work with Github

1. we have to create Github account -> www.github.com
   -> www.github.com -> Signup -> verify the account -> Sign in
2. Download and install git client software
3. Once installation completed right click on mouse to verify git options are coming or not.

-> Create a repository

-> Repository is used to store project or source code.

Every project have one repository per environment (Dev/QA/Prod)

-> when we create an repository an unique URL is created with

Repository name (Repo url)

https://github.com/codertech123/devops-app-01.git

-> All developers will connect to repository using repo url

-> We can create 2 types of repository

1.public repository

2.private repository

-> public repo : Anyone on the internet can see this repository.

You choose who can commit.

-> private repo :You choose who can see and commit to this repository.

Git Bash

-> Git Bash we can use a git client s/w to perform git operation

-> Dowload and install git client (https://git-scm.com/downloads)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

backfill Download missing objects in a partial clone

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

Working With Git Bash

To configure our email and Name in gitbash with commands

$ git config --global user.name "codertech123"

$ git config --global user.email "coderuniversetechnology@gmail.com"

-> git init : To initialize our folder as git working tree.

-> git clone : To clone git repository from github.com

git clone https://github.com/codertech123/devops-app-01.git

-> git status : It will display stage,un-stage and un track file

stage files: The files which are added for commit

un-stage files:The file which are modified but not added for commit

un-track files:Newly created files

-> touch Demo.java

-> git add Demo.java

-> git commit -m 'my-first-commit'

-> git push

Note : If first time we are doing 'git push' it will ask

to enter github account credential.

We can see the credential in "Credential manager".

=> go to Credential Manager -> Windows Credential

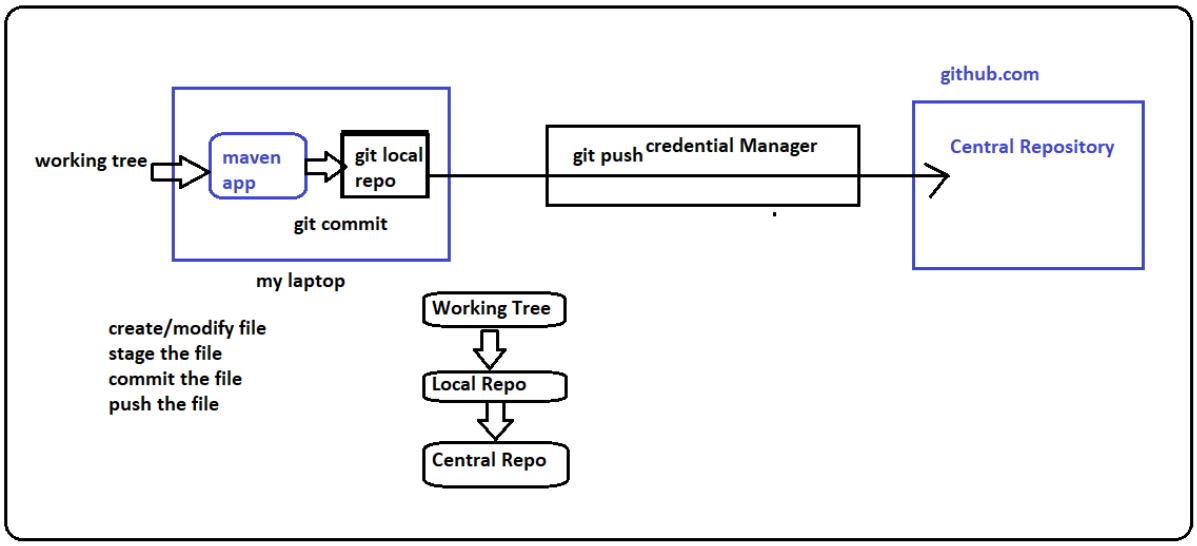
(We can modify or delete the credential according to our need)

-> git log : We can check the history using git log comand.

commit id we are able to see which is of 40 character.

Steps to Commit Maven Project to Github Repository

==========



1. Create an Maven Project
2. Create Github Repository (Demo-Maven-Project-24-April)

-> After creating git repository it will display a set of command

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/codertech123/Demo-Maven-Project-24-April.git

git push -u origin main

3. open git bash inside the project folder and perform above commands.

git restore : To revert the changes.

Git stash

-> To keep the changes in safe mode before working on other changes

Usecase to understand stash command

Monday : Task is assigned to you 'JIRA101' :Change pom.xml version

Tuesday :Morning 11 am You started task 'JIRA101'

By 2pm your lead ask to you 'JIRA102':Create some code

flipkart.java

In this scenario we will use git stash command.

$ git stash list => id

$ git stash apply <id>

Git Branches

-> Branches are used to maintain separate code bases for projects.

-> Development team will integrated the code in 'develope' branch

-> bu-fixes team will integrate the code in "QA" brnach.

-> R & D team will integrate the code in 'r&d' branch

Branches

In Git repository we have multiple branch as below

>>main

>>feature

>>develop

>>qa

>>uat

>>release

>>research

In github we can create branches for different purpose

$ bgit clone -b <branch name> <url>

https://github.com/codertech123/Demo-Maven-Project-24-April/pull/1

>> One project will have one github repo.

>> In one repo we can create multiple branch

>> Branches are used to maintain multiple code base

>> Using of branch team members can do work paralle

>> git repo will be created by admin.

>> any team member will get necessary access(read/write) from admin

repo : main/devlope/qa/features/research/release

Branch Merging

The process of merging one branch into another branch is called

branch merging.

we will create a pull request (PR)

steps:

-> Create an feature branch from main branch

-> clone feature branch

-> create a new file in feature branch then commit and push to feature branch

-> go to central repo then create an pull request and merge the merge.

What is .gitignore ?

>> The .gitignore file is used to configure the files or folders which we dont in our commits.

>> The files and folders which are not required to commit to central repository those things we can configure in .gitignore file.

Example:

In maven project 'target' will be available which is not required to commit to central repository and we can configure in .gitignore file.

----------.gitignore----------

```
HELP.md
### Maven ###
target/
pom.xml.tag
pom.xml.releaseBackup
pom.xml.versionsBackup
pom.xml.next
release.properties
dependency-reduced-pom.xml
buildNumber.properties
.mvn/timing.properties
.mvn/wrapper/maven-wrapper.jar

### Windows ###
# Windows thumbnail cache files
Thumbs.db
ehthumbs.db
ehthumbs_vista.db
```

STS

.apt_generated

.classpath

.factorypath

.project

.settings

.springBeans

.sts4-cache

git merge vs git rebase

Both the command are used to merge changes from one branch to another branch

What is git pull command ?

-> pull command is used to take the latest changes from repository to local

What is source code repository

Need of sources code repo ?

whats are the source code repo available

code integration

monitored the acesss

what is github

what is git

what is version control

what is repository

public vs private repo

cloining of repo

stage vs un-stage vs untracked files

Adding files to staging area

unstaging the file from staging area

discard the changes

working tree

local repository

git add

git restore

git commit

git push

git log

git branch

git checkout

git merge

git rebase

git pull

git rm

Assignment

==========

1. Create an Maven web project >

2. Add "spring-core" dependency in pom.xml file

3. package maven project as jar/war file usign maven goal.

4. push the maven project in github.com (public repo)

5. push the project into github repo using gitbash

6.target folder should not be commited using git ignore file.

7.make the changes in pom.xml and push to github using git bash.

8. create 'feature' in git repo using 'main' branch

9. clone the feature branch using git bash

10. create pull request and merge feature branch to main branch.

++++++++++++++++++++++

Application Environment

++++++++++++++++++++++

Environment : It means an platform which is used to run application.

In realtime we will use multiple environemtns to run the application.

Dev Environment

SIT Environment

UAT Environment

pilot Environment

Prod Enivironment

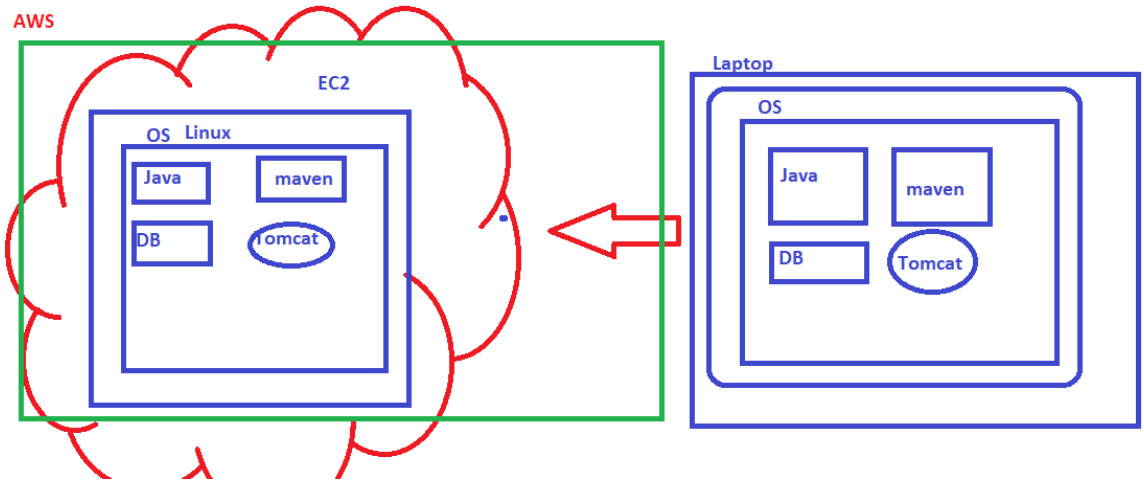
Dev Environment : Developer will use Dev env for integration testing.

SIT environment : System Integration Testing, QA/Testing team , will test the functionality.

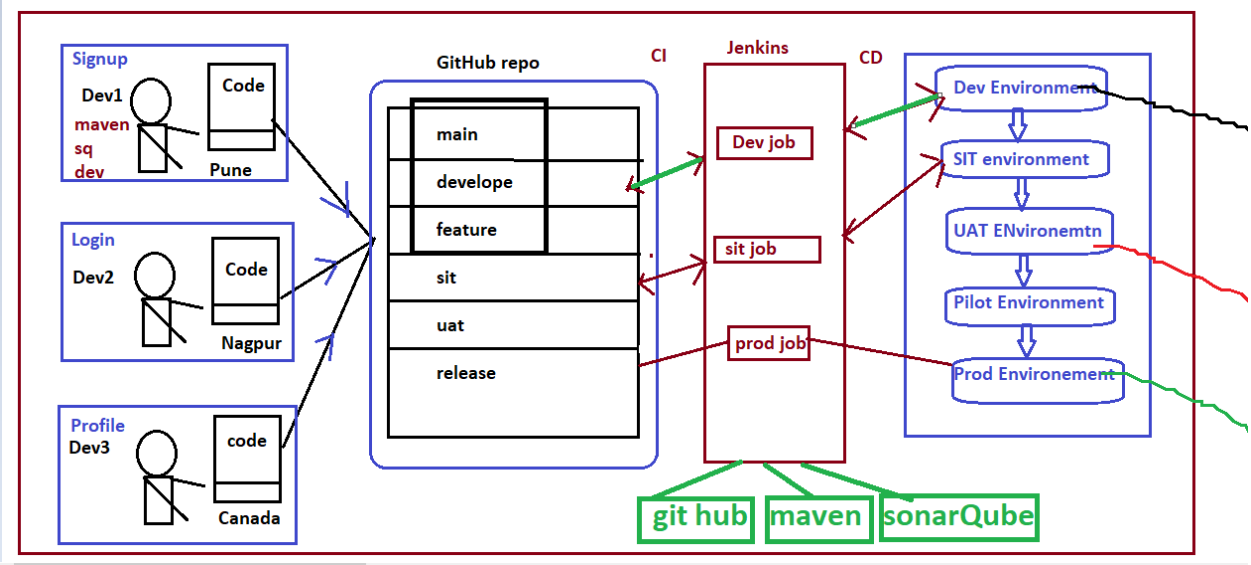
UAT Environemt : User Acceptance Teating, client will do this testing

Manual Deployment

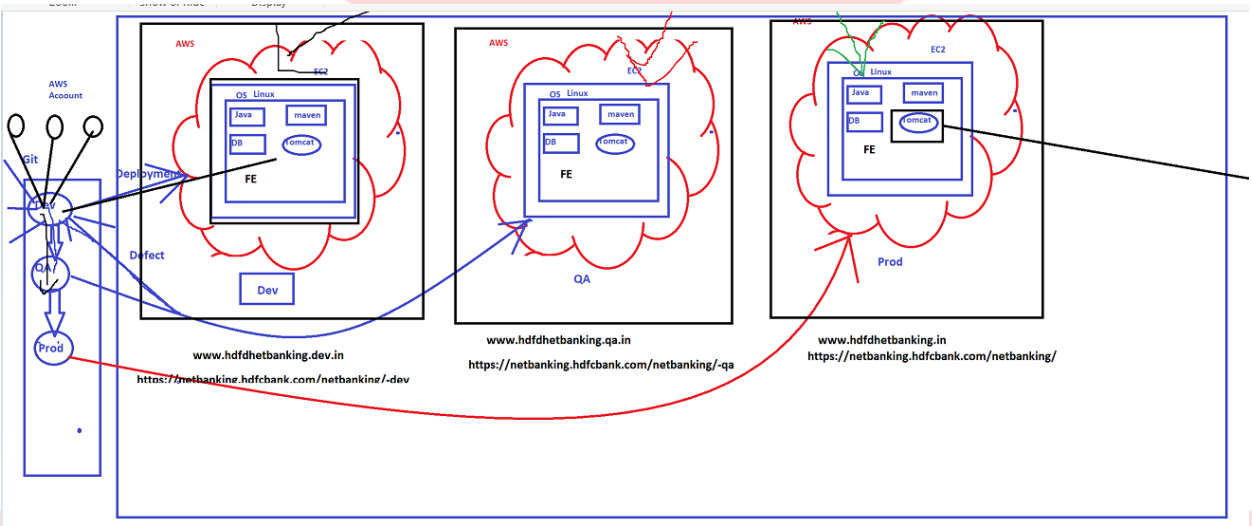
Below diagram helps to understand local laptop vs AWS EC2



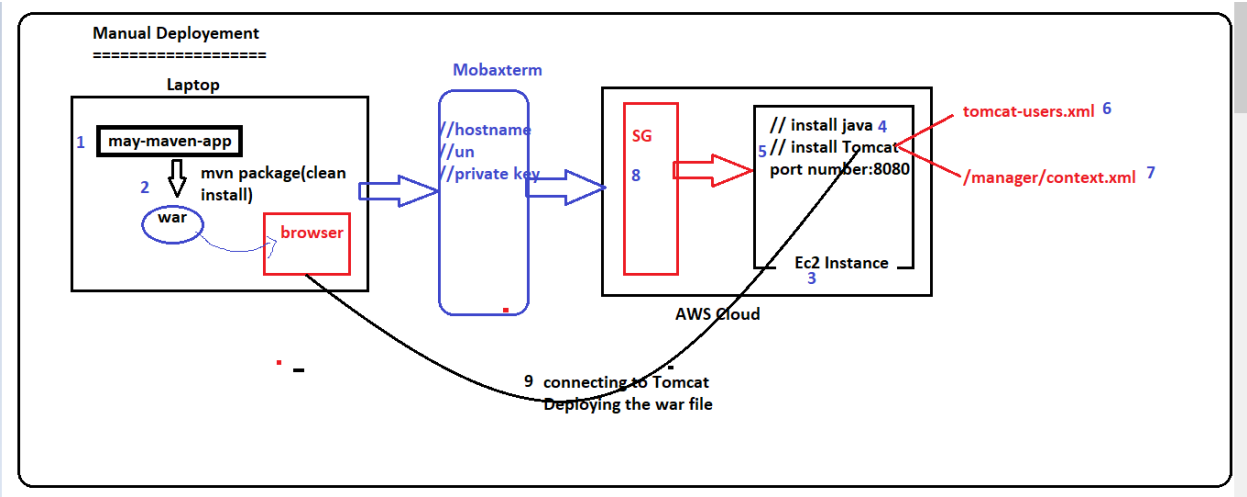
Below Diagram explain Environments like Dev/SIT/UAT/Prod



Below Diagram will help to understand each environment aws configurations



Below Diagram for Manual Deployements and steps



>> To run web application we need server.

>> For Java web application we can use below server

1.Apache Tomcat

2.JBoss

3.Weblogic

4.Websphere

5.GlassFish

>> For .net app we need to use IIS server provided by MS

++++++++++++

Apache Tomcat

++++++++++++

>> Apache Tomcat is a web server

>> Apache Tomcat is used to run Java web application

>> Apache Tomcat is free and open source.

>> Apache Tomcat runs on 8080 by default.

Working with Apache Tomcat in Linux

++++++++++++++++++++++++++++++++++

>> url : https://tomcat.apache.org/download-90.cgi

>> Download zip file and extract it

>>Apache Tomcat Directory

>> bin: It contain files to start and stop the server

windows : startup.bat & shutdown.bat

Linux : startup.sh & shutdown.sh

>> conf : It contains configuration files

server.xml

tomcat-user.xml

>> lib : It contains libraries(jars)

>> logs : It contain server logs

>> temp: It contains temporary data

>> webapps: we will keep application war files in this folder

This webapp folder is called as deployment folder of tomcat

AWS EC2 instance need to create:

>> First we need to create an AWS account

>> Create an Ec2 Intance(Machine)

>> Connect to Ec2 Instance using mobaxterm

>> Install java softwware using below command

sudo yum install java-1.8.0

>> How to verify java installed or not ?

java -version

>> We can download apache tomact from below official website

URL : https://tomcat.apache.org/download-90.cgi

>> command to install apache tomcat in linux machine

$ wget <tomcat-url>

$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.104/bin/apache-tomcat-9.0.104.tar.gz

>> we have to extract tar file which downloaded using below command

tar -xvf apache-tomcat-9.0.104.tar.gz

** ls -ltr : To check the directory

** pwd : Present working directory

** vi <file name>: To edit the file

url : http://3.85.91.250:8080

>> By default the Host Manager is only accessible from browser running on the same machine as Tomcat.

>> If we want to access the Host manager we need to modify "context.xml"

>> file location : tomcat/webapps/manager/META-INF/context.xml

>> We need to edit the context.xml file as below

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
       allow=".*" />
```

>> Add tomcat users in tomcat/conf/tomcat-users.xml

++

```
 <role rolename="manager-gui" />
 <user username="tomcat" password="tomcat" roles="manager-gui"/>
 <role rolename="admin-gui" />
 <user username="admin" password="admin" roles="admin-gui,manager-gui"/>
```


JENKINS

what is Jenkins ?

>> Jenkins is an open source automation tool for CI and CD.

>> Jenkins tool is devloped using Java.

>> Jenkins is part of Hudson

>> Initially It is called as Hudson then later it is rename to Jenkins.

What is CI & CD ?

>> CI and CD are two frequenlty used terminology in latest devlopment practices.

>> CI stands for Continous Integration.

>> CD Continous Delivery or Continous Deployment.

Developers keep on add the new code/update the code to central repository.

S to make this automate end to end process we use Jenkins to save time and performances.

>> Jenkins is a self contained,open sources automation server which can be used to automate all task related to building,testing and delivering software.

Build and Deployment

++++++++++++++++++++++++

1.Take the latest code form central repository

2.Compile the code.

3.Execute Code review

4.Package the war file

5.Deploy the war in server

CODER
UNIVERSE
TECHNOLOGY

CODER
UNIVERSE
TECHNOLOGY