**Software Development Life cycle [SDLC]**

1. Water fall model
2. Agile model

***Waterfall model:***

Requirements

Analysis ----> Dotnet/ Java /Python /Oracle /SQL Server/Mysql/weblogic/Jbass etc.

Design ----> Modules,DFD ---2Months

Coding -----> 20 Dev ----6 Months

Testing

Developer

1. Requirements stage will take all modules at once .
2. They decide the technology components in the analysis stage.
3. They break the project into different modules
4. Coding team (Developers) will receive the Dataflow diagrams in the designed sheets from the design team.
5. Once the coding is completed .They will test the application .if it is responding as per the requirements.
6. Deliver the project to the clients

Drawback:

1.There is a dependency to one stage to another stage.This cause lot of Time delay

2.When there is more time the project budget increases,infrastructure increases.

3.waterfall model does’nt ecncouarages the clients to give new requirements

4.Not a good model for complex and object oriented projects.

Ex:

Analysis team will wait for the requirements .

Coding team will start the coding once the design phase is complete.

Testing team will start the testing only once the coding phase is complete.

Design team will wait for the analysis team for their decision.

Note:

Model should be robust, flexbile, should accept the new requirements.

***Agile Model:***

This development method emphasis on iterative,incremental,and evolutionary development.

1. Requirements stage will take first module .
2. They decide the technology components for first module in the analysis stage. At this time requirements stage will be busy with second module.
3. At the design stage ,they will design the DFD for the first module. At this time Anyalisis stage will be busy with the second module.
4. Coding team (Developers) will receive the Dataflow diagrams in the designed sheets from the design team. They will start coding for the first module. At this time design stage will be busy with the second module.
5. Once the coding is completed .They will test the first application .if it is responding as per the requirements. At this time coding stage will be busy with the second module.
6. Deliver the first application to the clients. At this time Testing stage will be busy with the second module.

Advantages:

Parallel Execution happends in this method

Waterfall model---------------------------Agile model

2 Years 1Year

**Delay:**

In spite of the model, still delay?

Gap between development team and operation team (Non-Devolopers).

There should be strong communication between development team and operation team.

Here comes the role of DevOps

Operations Team Responsibilities:

1. Installation of server hardware and OS
2. Configuration of servers, networks, storage, etc…
3. Monitoring of servers
4. Respond to outstages
5. IT security
6. Change control
7. Backup and disaster recovery planning

**DevOps With AWS**

**Introduction To DevOps :**

***DevOps Is a methodology that promotes collaboration between Development and Operations team.This allows deploying code to production faster and in an automated way.it helps to enables rapid deployment of products.***

***That methodology can be implemented by using the following tools.***

1. Linux (pre-request OS)
2. Git hub - it is a source code management tool/version control/reomote control system.
3. Maven - build tool
4. Jenkins - CI/CD tool [Everything Automated]
5. Cloud platform - AWS, GCP, MICROSOFT AZURE
6. Terraform
7. Ansible - Configuration management tool.
8. Docker & Docker Swarm -Containerisation tool & Container orchestration tool
9. Kubernetes (k8s)
10. DevSECops
11. Nagios

We need **AWS account**, **Git hub account**, **Docker hub account**.

DEV: - Developers &QA (testing) team will be working here.

OPS: - whenever our application goes into live.

**DEV side:-**

Developers develop the code and push the code into GIT repository where all our codes will be placed .

Devops engineer clones the code from the GIT repository into his local machine he compiled the code with the help of Maven tool(which compiles any number of codes in a fraction of seconds) we will generates a WAR file (bundle of compiled codes).

Jenkins will be installed in a Devops engineer machine ,inside Jenkins we will create a project we require all the commands i.e

We predefinly write each and every thing like

* To clone the code from GIT repository.
* How to compile the code
* To build the code.
* How to deploy the codes into a server .

Deployment of WAR file continuously into a QA team is called the “Continous Integration (CI)” .

Deployment of WAR file continuously into a Production environment/live environment is called the “Continous Deployement/Continous Delivery (CD)” .

QA team will run the code if it is fine then they will release it into live environment.

WAR file will be deployed into a tomcat servers/production servers (live environment).

**Operation side:-**

Clouds companies maitaines huge servers .For a particular day flipcart/Amazon etc will borrow a particular servers virtually from the cloud and serves the application.

To deploy the servers automatically from the cloud we required Terroform.

In terraform we write one [.TF] file.This file will interacts with the cloud and it generates a servers with in a fraction of seconds.

Inside that server we required some configurations.To install that configurations we use Ansible .

With the help of Cloud, Terraform, Ansible we can create n no. of servers once our work/traffic is usual we will cut of all the services from the cloud we can pay only on that particular day.

Infrastructure,powe bills,realestate,manpower cost will be saved.

**1. LINUX :**

LINUX installation using VMware.

AWS free tier account installation with linux installation

We can connect with AWS linux terminal in 3 ways:

1. Directly we can connect in aws cloud.
2. We can connect with putty.
3. We can connect using git bash terminal.

**Note**: To connect aws linux machine with git bash terminal .we need to download git bash application.And goto the .pem file path,then just just give right click and connect to git bash here option.then give cmd

ssh -i "1.pem" ubuntu@ec2-15-206-82-92.ap-south-1.compute.amazonaws.com

pwd --- present working directory.

mkdir ---- creates a directory.

mkdir -p file ./file/file1 ./file/file2 ---- creating subfolders

mkdir -p file/{file1,file2} ----- creating subfolders

rm -rf - to remove directory/file.

touch - creates a file.

vim - it enables us to enter text to a particular file. i &esc&:wq + enter

vi - it’s a editor

cat - it shows the contents of a file.

cat -n - it shows the contents with numbers.

clear - it clears the command prompt.

cd - change directories one step forward

cd .. - change directories one step backward

ls - listing files

ls -l -it display files with permission ,filenames, timestamp etc

ls -a -it displays hidden files.

grep -it finds particular string in a file.

grep -i -b 2 maven devops.txt -we can get 2 strings before

grep -i -b 2 -a 2 maven devops.txt -we can get 2 strings before&after.

grep -v maven devops.txt -except maven it displays all the strings.

grep -n -v maven devops.txt -except maven it displays all the strings with numbers.

du -gives the size of a particular command.

du -sh -gives the size of a particular command properly.

head -it shows first 10 lines of a doc/file

tail -it shows the last 10 lines of doc/file

man -it will list all commands

man grep -it list all the grep related commands.

sudo apt-get update -it updates all the ubuntu packages &libraries.

sudo apt-get install openjdk-8-jdk -to install java in ubuntu

java -version -to check whether the java is available or not.

touch button.java -to create one java file.

javac button.java -to compile the java file

java button -to execute the code

cp -r source destination -it copies a particular file/folder and pastes it an another file/folder.

cp -r source1 source2 source3 destination -it copies a multiple files/folders and pastes it an single file/folder.

mv source destination -it moves a particular file/folder into another file/folder.

mv source1 source2 destination -it moves a multiple file/folder into single file/folder.

chmod 777 file/directory - giving permission to owner/team/others of file/directory.

chmod 777 -r file/directory -file permissions changes to directoy and inside a directories also.

vimdiff filename1 filename2 - it shows the difference content between two files.

ping [www.google.com](http://www.google.com) -it shows that url is working or not & ctrl+c

wget downloadlink -to download anything without using gui.

jobs -list you all commands running in background.

sleep 60 -command line wont work upto 60sec.

sleep 100 & -if you give & then it will running background

jobs -it will shows what commands are running in background

fg 1 -to makes that background running commands into foreground we have to use this command.

history -it will shows all the commands which are used in previous days.

echo -it is like print statement.

echo $variablename -for ex we gave a=10 then to print value give echo $a

zip filename.zip source1.txt source2.txt -it bundles all the text files into one single zip file.

unzip filename.zip -it extracts all the zipped text files.

**About vim editor :**

i -it enables us to enter data inside vim editor

esc - it disable vim editor to write .

:wq -it saves the file and quits from the vim editor

:q! -without saving quits from the vim editor.

**Basics shortcuts in VIM editor :**

shift+g -this enables u to move u to end of the text file.

shift+gg - this enables u to move u to first line of your text file.

ctrl+f -it takes you one step forward from your text file.

ctrl+b -it takes you one step backward from your text file.

yy -used to copy the line .so place the cursor to a particular line and press yy

p -used to paste the line press p

shift+$ -cursor moves to the end of the line.

0 -cursor will come back to beginning of the line press zero.

dd -line will be deleted.

u -to get the changes back (undo)

x -it will deletes a single character/letter in a particular line

dw -to delete one word from the line

w -it will move you one word forward

ctrl+r -to redo the changes

**2. GIT**

Version control systems are tools that help a software team manage changes to source code over time.

For almost all software projects, the source code is like the crown jewels - a precious asset whose value must be protected.

VCS are sometimes known as SCM (Source Code Management) tool.

Most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source tool originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel.

Two types Version Controlloing.

1) Centrailized Version controlling (Outdated)

2) Distributed Version controlling

Git is Distributed Version controlling.

The below tools acts as a source code management tool/version control/reomote control system.

Free tools paid tools

Git hub TFS

SVN PERFORCE

CVS CLEARCASE

**What is GitBash and how to use :**

GitBash 🡺 it’s Terminal 🡺 which is used to run git commands

git init 🡺 To convert the folder into working directory (when we use git bash terminal)

1. if we give the above command it will convert the folder into working directory
2. it will create one hidden folder (.git)
3. we can see master word in console

**HOW TO INSTALL GIT IN OUR UBUNTU MACHINE (AWS)**

sudo apt-get update -intially we need to update the packages&libraries.

sudo apt-get install git -to install git in ubuntu

which git -it shows the path where the git is installed.(any software)

git - -version -it tells us git version

NOTE:-

sudo apt-get remove git -to uninstall git in ubuntu

**GIT END-TO-END ARCHITECTURE**

1.There is one Git server(GS),inside git server there will be remote reposirtory(RR),in that repository let us assume there are 10k files i.e 1 project is here say PINKY.

2.We downloaded/cloned all the 10k files from gitserver to our machine in local repository(LR).But it is not human understandable language .it is in git format and it’s file name is .git

3.The same project/files will be stored in source area human understandable format as well stage area/virtual area/temporary area in our local machine human understandable format.

4.We started doing modifications inside source area.

5.Added the modifications to stage area.

6.Commit the changes to LR.

7.Push the code into RR.

**How to configure username and email for git**

$ git config --global user.name "sunilkumark11"

$ git config --global user.email [sunilkumark11@gmail.com](mailto:sunilkumark11@gmail.com)

**TO check the configurations**

$ git config --global --list

**Concept :**

Local Repositoy

Staging Area

Working Directory

One.java

Two.txt

Three.py

Four.net

* git status - To track untracked files(Red color) & To track staged files (Green color).
* git add filename -it will move single file from source area/working directory to stage area.
* git add . -it will move all file from source area/working directory to stage area.
* git commit filename -m “it’s a new text file” - Moving Single File into Local Repository.
* git commit -m “it’s a new text file” - Moving All File into Local Repository.
* git log - It will show all the commands.
* git log --oneline - It will also list all commands .
* git restore --staged filename -File will move backward i.e stage to source are/working directory

Note:

By default every file will be considered as a untracked file.

Staging area & local repository are the virtual storage areas.

**Understanding the need for branching :**

This feature is provided in git, so that developers can create code related to different functionalities on separate branches.

This helps the development team in creating the code in an uncluttered way.

Later this code can be merged with master branch.

Default branch of git is "Master".

When developers wants to move some particular functionality to live into production not the complete functionality.we map the Jenkins tool to the particular functionality code and Then move it into the production.This will increase the modularity of the project.

a b z Master branch

F1 F2 F3

T1 c T2 d Child branch

**Whenever you are in Master branch :**

If you are in master branch, the commits belongings to child branch will not be displayed. It only displayed it’s own commits when we use git log i.e a,b,z

Here the latest commit id “z” is the head commit id.

**Whenever you are in Child branch :**

As per the git concept ,whenever the new brach created the commit history will be copied into the new branch. So if we use git log it will shows a,b,c,d

git checkout branchname -To switch from one branch to another branch.

git branch -To list all branches. (current branch shows in green color)

**Merging Child branch with master branch (Normal)**

a b z Master branch(default)

F1 F2 F3

T1 c T2 d Child branch

Whenever you merge the child branch with master branch .The commits which are belonging to the child branch will be added to the master branch linearly.when we use git log then it will show a,b,c,d,z.

After merging it will creates a new commit id .That is the head commit id. So totally 6.

Note: While Trying to merge childbranch with masterbranch you should be in master branch.

**For suppose our requirement is we have 5 java files & 5 class files in working directory and these class files are generated during compilation. Compiled files / Machine generated files/un necessary files should be ignored and should send only source code to local Repository ?**

Ans:

For this we need to create a file called .gitignore place that in working directory (open git bash terminal make it as working directory). In that file it should be mention \*.class If you have multiple compiled files which have different extension then write that in next line .So that it will ignore all the compiled /machine generated files/un necessary files.

When we switch into masterbranch - it will shows only the files which belongs to only master branch.

When we switch into childbranch - it will shows only the files which belongs to only child branch.

**Git Rebase :**

This is called as fastforward merge.

The commits from the child branch are added to the top of the master branch.

This is helpful when we want code from a branch to be reflected as the latest working version on master.

**Merging Child branch with master branch (To be added as a latest commit in master)**

a b z Master branch(default)

F1 F2 F3

T1 c T2 d Child branch

Step 1: git checkout childbranch

Step 2: git rebase masterbranch

Step 3: git checkout masterbranch

Step 4 :git merge childbranch

Step 5: git log --oneline

d,c,z,b,a 🡺 Here “d” is the latest commit called the head commit.

Where you perform rebase command there will be no new commit will be created.

**Re arrange the commit order :**

a b c d e F1 F2 F3 F4 F5

**Our Requirement is to get the order like a e c d b**

Out of 5 commits The position of the first commit cannot be changed its fixed. We have scope for the 4 commits.

git log --oneline

git rebase -i HEAD~4 (Then copy and paste it in your required position)

git log --oneline

i= interactive terminal

**Merge the commit's using "squash"**

**Our Requirement is to remove un necessary commit (removing unnecessary commit is not the removing file)**

Out of 5 commits The position of the first commit cannot be changed its fixed. We have scope for the 4 commits.

git log --oneline

git rebase -i HEAD~4 (Then replace word squash in place of pick )

git log –oneline (it will shaow only your required commits)

i= interactive terminal

**How to selectively pickup commits from child branch and merge in master branch**

Step 1: git checkout childbranch

Step 2: git log --oneline - Take a note of the commit id

Step 3: git checkout master

Step 4: git cherry-pick (commit id’s)

Step 5: git log --oneline

Note: Now the commit id from the childbranch is picked up in the master branch as latest commit id.

**Git stash (To hide the untracked files & staged files):**

Explanation:

This feature is used for leaving unfinished work, in such a way that Git cannot access it and continue work on some other files.

Further commands of git should be able to process only the new files.

This can be done using git stash.

git stash -To Hide the stagged files.

git stash -u -To hide staged and untracked files.

git stash list -To see the list of stashes

git stash pop -To get back the Hide files

git stash pop stash@{stash\_number} -To bring the older stash out

**Git Ammend (Using this We can made changes to the commited file.Then we can push and commit the file .it won't create a new commit) :**

Explanation:

When developer changes some modifications to the committed file.git again starts treating the file as a untracked file.Again we need to push, commit the file.Then it will generate a new commit.

As it is not a great progress(minor changes in file) he don't wants to create a new commit he wants to amend the changes to existing commit only. So that we are using the below command.

$ git commit --amend -m "b"

**How to go back to previous version of code**

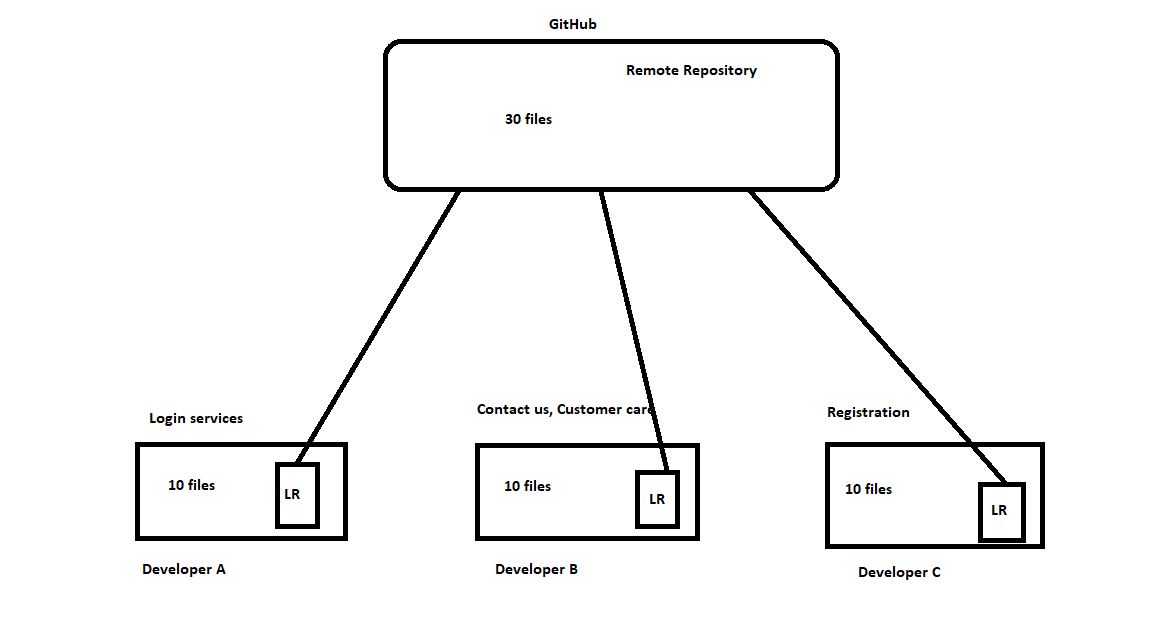
Explanation:

As per the client requirements developers commit the code let say "a".Again client wants new requirements of the same project.Developers modified the code and commit the code let say "b".

But Customers are not satisfy with the new requirement.So clients wants the previous requirement again.in that case we are using the following command.

git reset --hard (prev\_commit\_id)

**GITHUB (Remote Repository) :**

****

Remote repository is the cetralised location where the entire code related to the project will be available.The name of the Remote repository is “GitHub”.

git remote add origin https://github.com/sunildevops77/rep22.git (it will link with LR to RR)

get clone <https://github.com/pavan793/devops-training.git> (it will link with LR to RR)

git push -u origin master (To upload the files from LR to RR)

git push (To upload the files from LR to RR)

**Difference between git and github ?**

git is a toolname.

github is nothing but Remote Repository.it is the centralised location where the entire code related to the project will be available.it is used to only circulates the code.

**For ex:**

We have code in git hub repository ,we need to clone/download that code into our Ubuntu using linix command . for that we use this command.The below link will be available in your git repository page.

* get clone <https://github.com/pavan793/devops-training.git>

To push the code from Ubuntu to github we need to do the following steps.

First we need enter into our directory i.e devops-training folder in Ubuntu i.e

* cd devops-training
* touch first.java
* git add first.java
* git commit first.java –m “first module”
* git push (we need to use our git hub account credentials )

Ex:1

How to clone the code from RR into our local folder.

How to create the files & how to push that file into RR.

How to see file logs.

First we need enter into our directory (local machine)i.e pavan folder in Ubuntu

* git clone <https://github.com/pavan793/devops-training.git> -to clone the code

To push the code from Ubuntu to github we need to do the following steps.

First we need enter into our directory i.e devops-training folder in Ubuntu i.e

* cd devops-training -/pavan/devops-training$
* touch first.txt -first.txt file is created in source area
* git status -whatever you have done modifications (red color)
* git add first.txt -it will move from source area to stage area
* git status -whatever you have done modifications (green color)
* git commit first.txt –m “it’s a new text file” -giving some message to understand in future.
* git push -

git log first.txt -we will get full information about that text file.(created commit id)

git log –oneline -it will also list all commands in a single line

git show (commitid) -to track the file & everything.

----------------------------------------------------------------------------------------------------------------------------------------

Ex: 2

writing some code/text in that file and pushing it into RR

To check the changes in different stages like source,stage areas

vim first.txt -doing some modifications in source area.

cat first.txt -it will shows the content in that file

git status -whatever you have done modifications (red color)

git diff first.txt -to need to check the changes in source area .

git add first.txt -it will move from source area to stage area

git status -whatever you have done modifications (green color)

git diff - - staged first.txt -to need to check the changes in stage area .

git commit first.txt -m “it’s a modified text file” -giving some message to understand in future.

git show - to need to check the changes in local repository .

git push

---------------------------------------------------------------------------------------------------------------------------------

Ex:3

Suppose we modified by writing text/code in that file and saved it ,after sometime we realise that we want previous changes ,we don’t want new changes made to that file then we have to use below.

vim first.txt -adding some data in source area.

cat first.txt -it will display the modified content.

git checkout first.txt -we will get without modified data(previous file),but works only in stage area.

----------------------------------------------------------------------------------------------------------------------------------

Ex:4

Creating & Pushing more than one text files to RR .

How to move backward to previous stage i.e stage to source area/working directory

If we create more than one file in source area then we need to add those files one by one .

To add all files at a time we are using this concept.

touch text.1 -adding some text files in source area.

touch text.2

Touch text.3

git status -used to get info (red color)

git add . –used to add all modified files at a time into a stage area

git status -used to get info (green color)

git reset -to go back from stage area to source area backward

git add .

git commit –m “commit all files” -these are in local repository

git push -it will push the into remote repository

Ex:5

creating & pushing a directory / folder into remote repository.

How to resolve always asking credentials issue while pushing the code

Note: folder/directory should not be empty it must contain atleast one file.

mkdir cloud11 -creating directory in source area

cd cloud11

touch message.txt

cd .. –toget back into our RR folder

git status -used to get info (red color)

git add . –used to add all modified files at a time into a stage area

git status -used to get info (green color)

git commit –m “a folder with file” -these are in local repository

git config - -global credential.helper store -if we give this it won’t ask credentials while pushing.

git push -it will push the into remote repository

Ex:6

creating & pushing more than one folders/directories into remote repository

Note: folder/directory should not be empty it must contain atleast one file.

mkdir cloud11 - creating directory in source area

cd cloud11

touch touch message.txt

cd .. –toget back into our rr folder

mkdir cloud12 - creating directory in source area

cd cloud11

touch touch message.txt

cd .. –toget back into our rr folder

mkdir cloud13 - creating directory in source area

cd cloud11

touch touch message.txt

cd .. –toget back into our rr folder

git status -used to get info (red color)

git add . –used to add all modified files at a time into a stage area

git status -used to get info (green color)

git commit –m “a folder with file” -these are in local repository

git push -it will push the into remote repository

Ex:7

How to skip staging area and push into remote repository [applicable only for existing files]

mkdir project

cd project

git clone <https://github.com/pavan793/devops-training.git>

cd devops-training -all my code will be here remote repository code(we are in source area).

ls -a -u will get all files listed in that folder

vim filename1 -write some text/do modifications in that existing file

vim filename2 -write some text/do modifications in that existing file

vim filename3 -write some text/do modifications in that existing file

vim filename3 -write some text/do modifications in that existing file

git status -used to get info (red color)

git restore filename -to get back to intial positons(no modifications file will shows)

git commit -am “skipping the stage area” -to skip the stage area

git status

git push

------------------------------------------------------------------------------------------------------------------------------------------

Note: stage area is not mandatory for all files

1.if you are creating a new file in source area ,stage process is mandatory.

2.For Already existing files,we are doing modifications on the existing file,stage process is not mandatory.

Ex:8

Trying to skip staging area and push into remote repository [not possible for new files]

cd devops-training -all my code will be here remote repository code(we are in source area).

ls -a -u will get all files listed in that folder

touch one -creating new file

ls -a -u will get all files listed in that folder

git status -used to get info (red color)

git commit -am “trying to skip the stage area” -it won’t work

rm –rf filename.txt -to remove the file from source area

git add first.txt -it will move from source area to stage area

git status

git commit –m “going through stage process” -these are in local repository

git push

----------------------------------------------------------------------------------------------------------------------------------------

Ex:9

How to remove a file which is present in Remote repository

Note : Whenever we are removing a file in RR, no need of stagging process

/pavan/devops-training$ ls -listing the files in source area(assume we have one.txt file)

/pavan/devops-training$ git rm one.txt -to remove the file

/pavan/devops-training$ git commit -m “deleting one.txt file”

git push

Ex:10

we tried to remove file from RR at source area and after sometime I realized that I want that file back so we need to give the below commands

/pavan/devops-training$ ls -listing the files in source area(assume we have one.txt file)

/pavan/devops-training$ git rm one.txt -to remove the file

by using the below command we will get that file back

git restore --staged file2.txt -we will restore that file by giving these two commands

git restore file2.txt

Ex:11

we tried to create a file & remove file from local repository at source area

/pavan/devops-training$ ls -listing the files in source area(assume we have one.txt file)

/pavan/devops-training$ rm –rf one.txt -to remove the file

We will not restore the file again here if we remove that file in source area

Ex:12

Ex: we tried to create a file & remove file from local repository at stage area

/pavan/devops-training$ ls -listing the files in source area(assume we have one.txt file)

/pavan/devops-training$ rm –rf one.txt -file will be deleted (partially)

git restore one.txt -we will get file again

/pavan/devops-training$ rm –rf one.txt -file will be deleted ( but status will shows ,so do completely delete by giving the below command)

git rm one.txt -now file will be completely deleted (no status showed)

-------------------------------------------------------------------------------------------------------------------------------

Ex:13

How to remove a folder/directory in source area which is present in RR .

How to get back that folder again

rm –rf dir1 -this command is enough to remove the directory/fodler in source area

we will not restore the folder again here if we remove that file in source area

Ex:14

How to remove a folder/directory which is present in stage area .

How to get back deleted folder

Note:while creating a folder please create file inside that folder then only process will enable like (add/commit/push).if you only created folder it will be in source area nothing process will be enable like (add/commit/push).

rm –rf foldername -this command is enough to remove the directory/fodler in stage area

git restore dir -we will get back that folder again

/pavan/devops-training$ rm –rf dir -file will be deleted ( but status will shows ,so do completely delete by giving the below command)

git rm -r dir -now file will be completely deleted (no status showed)

----------------------------------------------------------------------------------------------------------------------------------------

Ex:15

How to rename a file which is present in Remote repository

Note : Whenever we are renaming a file in RR, no need of stagging process

/pavan/devops-training$ ls -a -listing the files in source area(assume we have one.txt file)

/pavan/devops-training$ git mv one.txt two.txt -to rename the file

git restore --staged dir1 -it get back to its intial name

/pavan/devops-training$ git commit -m “renaming one.txt file to two.txt”

git push

Ex:16

To display log files or To Track the files

/pavan/devops-training$ git log -it displays n no.of files what we have done.

/pavan/devops-training$ git log --author=ubuntu -it displays all logs what done by Ubuntu author

/pavan/devops-training$ git log --author=ubuntu --grep=stringvalue

/pavan/devops-training$ git log --author=ubuntu --grep=stringvalue

/pavan/devops-training$ git show (commitid) -it displays particular info related to that commit id

/pavan/devops-training$ git log --grep=stringvalue -it displays only particular log.

---------------------------------------------------------------------------------------------------------------------------------------

Ex:17

TO CREATE A BRANCH

pavan/devops-training$ git branch dev\_123

pavan/devops-training$ git push origin dev\_123

------------------------------------------------------------------------------------------------------------------------------------------

Ex:18

CREATING NEW BRANCH IN REMOTE REPOSITORY IN GITHUB & ALSO CREATING FILE IN THAT NEW BRANCH.

SWITCHING FROM ONE BRANCH TO OTHER BRANCH

mkdir cake

cd cake

git clone <https://github.com/pavan793/devops-training.git> -you cloned empty reposirty(assume)

cd devops-training

touch login.java

vim login.java -write some java code

cat login.java

ls -a -list all files .if there is .git file then it is a source area

git status

git add . – it will move from source area to stage area

git commit one.txt –m “login module” -giving some message to understand in future.

git push -it is pushed into remote repository

ls -a -it list files

git branch -it shows the current active branch in green color

git branch dev111 -used to create a new branch

git push origin dev111 -used to push a newly created branch into remote repository

git branch -it shows the current active branch in green color

git checkout dev111 -to switch from one branch to another branch &to activates branch

git branch -it shows the current active branch in green color

git checkout master -to switch from one branch to another branch &to activates branch

git branch -it shows the current active branch in green color

git checkout dev111 -to switch from one branch to another branch &to activates branch

git branch -it shows the current active branch in green color

ls

touch dev111.java -creating a new file,as it consists already one file which is copied from master branch

git add.

git commit –m “dev111 new code”

git push

git push - -set-upstream origin dev111 -sometims it asks while pushing just copy &paste

git branch

ls

git checkout master

git branch

ls

We have the following files in both master & dev111

master branch login.java

dev111 branch login.java dev111.java

dev111.java is the new extra code

Now I need to merge dev111 to master branch

So that we have to stay in master branch first then give command as follows

git merge dev 111 - to merge one branch into another branch

We got like this after merging

master branch login.java dev111.java

dev111 branch login.java dev111.java

Now both master branch and dev111 branch are equal.

-----------------------------------------------------------------------------------------------------------------------------------------

Ex:19

Now by choosing one file in both master branch & new branch changing the values and merging from one branch to another branch.

For ex:

In master branch we have login.java we have a=10

In dev111 branch we have login.java we have a=10

Now I will switch into master branch and I will change the value in login.java from a=200

We will add to stage,commit & pushed the code into remote repository

vim login.java

git add.

git commit login.java -m “a=200”

git push

Now I will switch into dev111 branch and I will change the value in login.java from a=300

We will add to stage & pushed the code into remote repository

vim login.java

git add.

git commit login.java -m “a=300”

git push

Conflict: if one file having two different parameters/values conflict will arrives during merging.

mkdir foldername

cd foldername

git clone <https://github.com/pavan793/devops-training.git>

cd devops-training

git branch -whenever we are using new directory it shows only master branch ,so to activate previous branches we need to activate .

git checkout dev 111

git branch -it will lists branches

now we are in dev111 branch

git checkout master

git branch

ls

vim login.java -we changed a=200

cat login.java -it displayes code

git add.

git status

git commit -m “a=200”

git push

git checkout dev111

git branch

ls

vim login.java -we changed a=300

cat login.java -it displayes code

git add.

git status

git commit -m “a=300”

git push

Conflict : if one file having 2 different parameters or values.conflict will arise during merging.

MERGING TWO BRANCHES

Whenever we are merging we should inside master branch

git checkout master

git branch

git merge dev111 - we will get conflict message here

git status -it will shows both are modified message

vim login.java - here we need to resolve issue to avoid conflict

[We need to resolve this conflict

We need to enter into login.java file using vim editor

Ask confirmation from developers which value I have to choose

Delete all the values which are not needed

We will add to stage & pushed the code into remote repository]

git status -it shows red color

git add.

git commit

git push

Ex:20

How to delete a branch [when we have clumsy/confusion we can delete branches]

mkdir cherry

cd cherry

git clone <https://github.com/pavan793/devops-training.git>

cd devops-training

git branch

git checkout dev 111

git branch

git checkout master

git branch -d dev111 -----this branch deleted in local repository

git push -d origin dev111 ----we deleted that branch in remote repository

git branch -r -to list all the branches in the remote repository.[origin indicates remote repository]

------------------------------------------------------------------------------------------------------------------------ ---

Ex:21 **METHOD - I**

**Two users trying to push the same code into remote repository at a same time**

User1 & user 2 are cloned/downloaded the code from remote repository at a time

User 1 completed his modifications (creating file/ folder/changing values) and pushed into remote repository but user 2 still working on that code

Now user2 also done his modifications(creating file/ folder/changing values),he is trying to push the code into Remote repository,git will not allow him because user1 modified that folder so we will get error.

git pull -whatever the modications done by user1 will fetch here(alt+f2)

vim login.java

git status

ls

git add .

git commit -m “updated”

git push

-------------------------------------------------------------------------------------------------------------------------------------

Ex:22 **METHOD - II**

**Two users trying to push the same code into remote repository at a same time**

User1 & user 2 are cloned/downloaded the code from remote repository at a time

User 1 completed his modifications (creating file/ folder/changing values) and pushed into remote repository but user 2 still working on that code

Now user2 also done his modifications(creating file/ folder/changing values),he is trying to push the code into Remote repository,git will not allow him because user1 modified that folder so we will get error.

git fetch -current modification are in port area temporary storage area.

ls -we will not seen because it will be in only port area/temporary directory

git merge -whatever the modications done by user1 will fetch here(source area)

ls -it will fetch the all the files

git add .

git commit –m “updated”

git push

Note: git pull = git fetch +merge

**3. MAVEN TOOL :**

**What is Maven?**

It is product of apache.

Maven is also a build tool. Where it can compile the programs and create a artifact.

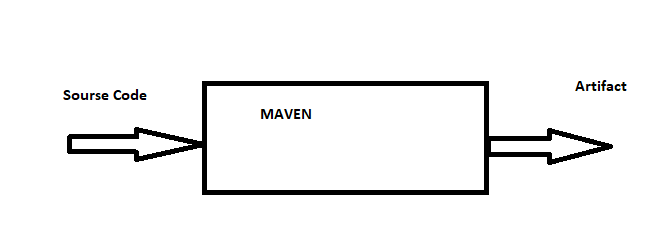
JDK is a dependency for maven. Because MAVEN is developed by using java.so before maven installation we need to install JDK. Maven tool supports upto 8th version of jdk.

With the help of maven tool we can save time by compiling 1000’s of files at a time.

It will zip this all machine understandable class files to a jar/war file.

Jar ---java application related ----------contains only class files.

War ---web application related ----------contains multiple jar files.



Once the development activity is done.Developers Build /package the source code and generate the artifacts (.war/.jar/.msi /.exe files….etc).

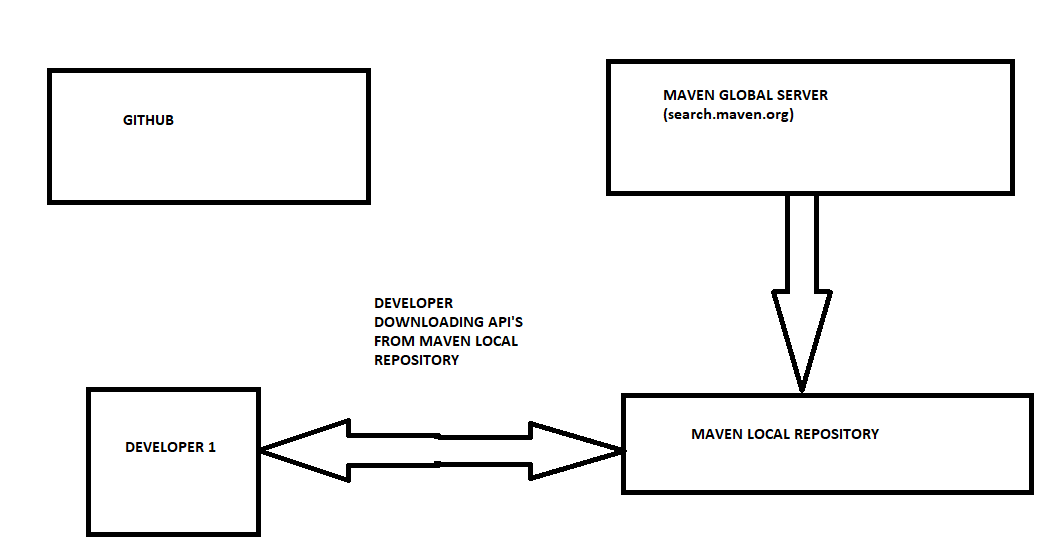
**What is Artifact ?**

Artifact is nothing but the final file which we deployed it into the server.So that the application will be running.

**What is vulnerability ?**

During the development time,developer want to download the API’s code from the 3rd parties.These are open source .if the source code contains API’s from the 3rd parties. Then their may be harmful/destroy the source code finally risk is involved if this type of API codes if we used.

**How the API’s are downloaded from the maven global server**



**What is maven global server?**

All the open source communities(3rd parties) like Jenkins, tomcat, mysql,google map etc upload their updated API's into maven global server/Maven central Repository (search.maven.org ).

**What is Maven local repository?**

Developers connect to the maven global server, download those API that are necessary for the project development.

API related code gets uploaded into github in the form of Read only format.Doesn't have any permission to make some changes or edit some content or damage the other files.

This will help in protecting the code from any threats/viruses that might be present in the API's.The vulnerability issue solved.

**MAVEN installation:**

sudo apt-get update -------it will update our ubuntu packages.

sudo apt-get install maven

mvn --version

which mvn

after cloning the code we need to do like this.

tree [or] tree . --- to see the code in tree form

mvn install ---it builds or compiles the every code

**MAVEN un-installation:**

sudo apt-get remove openjdk\*

**Project creation:- (METHOD -1)**

Java projects which are created by maven,ideally follows below project folder structure.

mkdir devops

cd mkdir

git clone url

mvn install -----it will compile&building the application & target folder is generated.

We will get jar/war file after compilation we need to follow 6 phases.

1.intialize pahse -----target folder structure creation -🡪 **target**

2.compile phase ----it compiles only main code -- > classes

3Test compile phase -----it compiles only testing code. ---->test-classes

4.reports phase ------ report generation ------>surefire-reports

5.packaging phase ----jar/war file creation --> flipkart-1.0-SNAPSHOT.jar

6.install phase

cd target

ls

Ex: Classes flipkart-1.0-SNAPSHOT.jar maven-archiver maven-status surefire-reports test-classes

**NOTE:** mvn clean --------to clean the project

**Project creation:- (METHOD -2)**

**Note : we can use our windows cmd & gitbash terminal or directly we can use in AWS/VMWare/Putty.**

**Step 1:**

mvn archetype:generate - To create maven project

Lets select the default number: 1835 -Enter default number

Choose the version - Enter the default one 8

Define the value for groupID: com.paypal

Give the name for the Artifactid

Ex: webapppaypal ( With this name artifcatid (folder)will be created )

We will get Build success message.

In which location project is created, In that location, we find src folder and pom.xml file

src contains two folders, main and test

Code created by the developer will be in main folder.

Code created by the developer for unit testing will be in test folder.

**Step 2:**

open gitbash

change directory to webapppaypal

git init

git status

git add .

git commit -m "first commit"

**Step 3: (Optional)**

**Let Developer needs 2 more API’s for his requirement**

Developer will search for the required API here 🡺 https://search.maven.org/ -- maven global server

Ex:

1) search for mysql ( select the required API ) - select version

copy the dependency tag and paste in pom.xml

2) search for google map

copy the dependency tag and paste in pom.xml

Now, we need to download the API from Maven Global server to Maven local repository.

Open command prompt

> cd webapppaypal

> mvn compile ====> Here Target folder will be created.

Maven local repository is created with the name .m2

Open git bash

Go to working directory

$ git status

$ git add .

$ git commit -m "second commit"

Suppose The development team identified that they downloaded the wrong API’s .

in that case They rollback into first commit. Again the POM.xml will be rollback and that target will be disappear. Again we can start from the scratch .That’s why we created a first commit.

Lets wantedly generate one more java file in main > java > groupid > app.java &Sample.java

mvn compile 🡺 we can see Target file generated in that we can see class files.

Lets wantedly generate one more java file in Test> java > groupid > AppTest.java &Sample.java

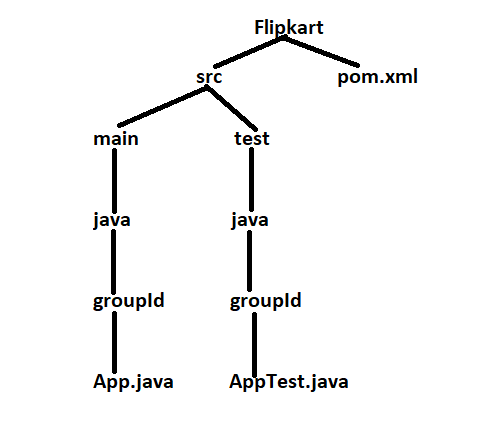
mvn test 🡺 In folder target , test-classes will be created , which contains .class files

**Step 4:**

mvn package - To create Artifact. (In folder target .jar file has been generated)

This artifact should be deployed into to the server.For this Jenkins tools we used.

Project structure



flipkart -is called “project name” /”ArtifactID”

src - source folder which contains the application source code.

Main -contains application’s main functional code.

Test -contains application’s unit testing code

Pom.xml -Maven’s build the file using which we can configure build steps.

IMPORTANCE OF POM.XML:- (Project Object Model)

It is an xml file, which will store all the external API information.

It is a build configuration file. It contains all build related information.

Pom.xml always follow lower camel case. Ex: group Id.

Maven follows convention over configuration feature (convention we cannot change configuration we will change).

Maven understands only xml. Whenever u execute mvn install ----maven 1st interacts with pom.xml.

As a devops engineer we will change 5% code of this pom.xml

* Devops engineer can change from one packaging to another packaging, War/Jar files are created based on pom.xml
* Devops engineer can change from one version to another version.

Ex:

vim pom.xml

Ans:-

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>com.flipkart</groupId> 🡪 Domain name

<artifactId>flipkart</artifactId> 🡪 Project name

<packaging>jar</packaging> 🡪here we can define jar/war file to create

<version>1.0-SNAPSHOT</version> 🡪 Devopps engineer work

<name>flipkart</name>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

Group id, artifact id, packing, version 🡪are maven coordinates [or] GAPV parameters

artifact id-version-packaging 🡪pakaging naming convension

1.0-SNAPSHOT 🡪 this project is under development process

1.0-RELEASE 🡪ready for testing. After confirmation from developers we need to change.

1.1-RELEASE 🡪new releases

All versions are changed by devops engineer based on requirement.

PROJECT 1: JAR

HOW TO COMPILE THE PROJECT

mkdir project1

cd project1

git clone <https://github.com/mahesh577-dell/flipkart123>

cd flipkart123

mvn install ---here target folder will generated.

**NOTE:** mvn clean --- to clean the project

PROJECT 2: HOW TO COMPILE & RUN THE PROJECT LIVE WAR

1.HOW TO INSTALL TOMCAT SERVER IN A FOLDER one time is enough to intall.

mkdir folder1

cd folder1

wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.0.10/bin/apache-tomcat-10.0.10.tar.gz

ls --we will found one zip folder

tar -zxvf apache-tomcat-10.0.10.tar.gz --🡪it will extract zip folder

ls 🡪it will list folders/files

cd apache-tomcat-10.0.6 -> to into apache-tomcat-10.0.6 folder

ls 🡪it will list folders/files

cd webapps 🡪to into webapps folder

copy the path by using (pwd) command------ **/home/ubuntu/folder1/apache-tomcat-10.0.6/webapps**

cd ~ ----to come into home directory

2.HOW TO COMPILE THE PROJECT

mkdir project2

cd project2

git clone https://github.com/mahesh577-dell/iflipkart

cd iflipkart

mvn install

cd target

copy the path by using (pwd) command------ /home/ubuntu/project2/iflipkart/target

cd ~ ----To come into home directory

3.COPYING THE WAR FILE INTO TOMCAT SERVER (Deploying the war file into server)

scp /home/ubuntu/project2/iflipkart/target/mahesh.war /home/ubuntu/folder1/apache-tomcat-10.0.6/webapps

cd ~

4.TO RUN THE Tomcat Server

cd /home/ubuntu/folder1/apache-tomcat-10.0.6/bin

./startup.sh ------🡪to start the server [Browser will shows o/p]

./shutdown.sh ---🡪to stop the server [Browser will not shows o/p]

5.TO SEE THE APPLICATION PRACTICALLY WORKING OR NOT

Syntax:

https://Public IPs:tomcatportnumber/warfilename

https:// 3.15.210.82:8080/mahesh/

-----------------------------------------------------------------------------------------------------------------------------------------

PROJECT 3: WAR

mkdir project3

cd project3

git clone <https://github.com/mahesh577-dell/real>

cd real

mvn install

cd target

copy the path by using (pwd) command------ /home/ubuntu/project3/real/target/maheshguru.war

cd ~ ----To come into home directory

scp /home/ubuntu/project3/real/target/maheshguru.war //home/ubuntu/folder1/apache-tomcat-10.0.6/webapps

cd /home/ubuntu/folder1/apache-tomcat-10.0.6/bin

./startup.sh

https:// 3.15.210.82:8080/maheshguru

---------------------------------------------------------------------------------------------

HOW TO CHANGE TOMCAT PORT:

**By default tomcat port number is:8080**

**To change tomcat port goto**

cd home/ubuntu/folder1/apache-tomcat-10.0.6/conf

vim server.xml

**Open server.xml file and change the port number from 8080 to 8085**

cd ~ ----to goto home directory

cd home/ubuntu/folder1/apache-tomcat-10.0.6/bin

first we need to shutdown the server and again we need to start the server

./shutdown.sh

./startup.sh

------------------------------------------------------------------------------------------------------------------------------------

Conclusion:

1. we deploy the WAR file in “ webapps “ folder.
2. To run the server we need to enter “ bin” folder.
3. To change port number we need to goto “ conf “ 🡪 “ server.xml “.

-----------------------------------------------------------------------------------------------------------------------------------------

* MAVEN DEPENDENCY
  + Dependency on your own code ----it is an advantage we can find error easy.
  + Dependency on third party code …..core team will take predefined codes junit.jar
* BUILD TYPES (4)
  + Full build/complete build/scratch build
  + Incremental build
  + Daily build
  + Nightly build

Full build :-

First time when we create target folder using mvn install command.Then it is called as full build/complete build/scratch build.

Incremental build:

Second time we have 2 choices:

1.u can remove target folder and u can build it from scratch

2.without removing target folder,we can use mvn install,maven takes only the new files, and maven compiles only the new files because target folder is already present and saves lot of time.

Daily build:

If you are triggering the build during day time,it is called daily build.

QA team will skip these phases if the dead line is coming so fast.

Skipped phases

Test compile phase , Report generation phase.

Nightly build:

If you are triggering the build during night time,it is called nightly build.

Specially we execute only

Test compile phase

Reports generation phase

mvn install -Dmaven.test.skip=true 🡪to skip test and report phases.it ia a part of daily build.

mvn compiler:testCompile 🡪to compile only test phase. it ia a part of nightly build.

mvn surefire:test 🡪to compile only report phase. it ia a part of nightly build.