Git and Source code management (SCM)

=====================================

Landmark-SS - Paypal:

=========== =======

Develops, Tests, Builds, deploys & monitors applications

Applications are the output from Softwares or written codes

paypal:

Releases:

Releasing an application is taking/deploying it to the client

In realtime we have multiple Releases/versions.

release1: = version1: = 1000lines of code

echo 'Welcome to Paypal' = Welcome to Paypal

release2: = version2: = 2000lines of code

echo 'Welcome to Paypal' = Welcome to Paypal

echo "Your convenient payment gateway"

Release3: = version3: = 1000 lines of code

echo "Paypal the best payment gateway"

The process of releasing a new version is called rollout

Rollout --> version1 --> version2 --> version3 :

===============================================

The client is not happy with the latest/newest version[version3]

of the application. What can be done??? RollBack

1. RollBack = Returning to a previous version

RollBack --> version3 --> version2

2. Remove the bugs and address the client's concern for

full satisfaction

Versioning in Software Development:

===================================

1. Local

2. Centralised

3. Distributed VCS:

Version control systems = VCS:

Local = version1, 1000Lines

version2,[5000LINES ]

version3 [2000LINES ]

cons: Single point of failure

lacks collaboration

in-efficient use of resources including time

can't RollBack

Centralised:

cons: can't RollBack

The central server can fail

Distributed VCS:

===============

GIT: DVCs

- git is a Distributed Version Control system

- Git manages data as snapshots. 0 / 1/ 10 / 20 / 30 / 40 / 50 / 60 /

- It takes a picture of what all your files look like

at that moment and stores a reference to that snapshot.

oneDay Old picture

oneWeek Old picture

oneMonth Old picture

oneYear Old picture

20 years Old picture

version1, 1000Lines

version2,[5000LINES ]

version3 [2000LINES ]

version30

PICTURES = 1DAY /1month/ 1YEAR /10/ 20 / 30 / 40

Benefits: 1. Nearly every operation is local

2. git has data intergrity

3. Git generally only adds data

4. Rollout and RollBack

5. good collaboration

other DVCs:

SVN --> Sub Version

CVS --> Concurrent Version system

TFS --> Team Foundation Server

Git --> 95% plus of companies are using GIT

RENT A PROPERTY = 50 UNITS / APARTMENTS

5 UNITS = 70

Git is a Distributed Version Control systems. DVCS

In our environments (Landmark) we use Git for Versioning. :

-

SCM = Source Code management

=======================-====

SCM service plaforms :

GitHub - MS = 80%

GitLab

BitBucket (Stash) - A

awsCodeCommit

IQ: As a DevOps Engineer, what are your roles & responsilities

in Versioning with Git?

Answer:

1. We ensure that Developers environments are configured and secured

We create an enabling environment for Developers to code/Develop

DevelopmentServers are created in AWS/GCP.

share login details with Developers/Team [ userName/password ]

Ensure that git package/Software is installed

2. Project Onboarding:

- create a github account for your organisation or self if applicable

- Create organisations where applicable in a SCM ['GitHub']

url = https://github.com/landmark34aa

- Create teams in SCM/github and assign members[Developers, etc.]

team URLs :

https://github.com/orgs/landmark34aa/teams/managersteam

https://github.com/orgs/landmark34aa/teams/team35

3. Grant permissions/roles to team members and collaborators:

read -- you can view codes

write -- you can view codes and make changes

admin/maintainer - ownership

4. - Create repositories in SCM [GitHub]

repo url = https://github.com/landmark34aa/paypal

- login into github

website = https://github.com/

- Ensure that Git and relevant IDEs are installed by Developers

Ticket: Onboard a new project for dominion

website = https://github.com/

Git Installations

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

Git Installations differs depending on your operating system:

Pre Requisite Software Download/ Registration :

Install git in Windows :

Install gitbash @ https://git-scm.com/downloads [git.exe]

$ git -v

git version 2.38.1.windows.1

Install in RedHAT Linux servers:

sudo yum install git -y

Install in ubuntu Linux servers:

sudo apt install git -y

=============================

Ticket001:

Install git in your Redhat9 server

Install git in your work station

windows:

solution:

sudo yum install git -y

echo "# paypal" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/landmark34aa/paypal.git

git push -u origin main

Assume the role of a Developer

We are developing for our paypal FinTech client.

1. create a directory for Development

mkdir paypal && cd paypal

2. Initialise this directory to keep track and records of

any change or changes in the files/codes

git init

Git Commands:

git status

git init

create a Local repository [.git]

initialises the git repository

git add

git init

workingArea stagingArea LocalRepo RemoteRepo :

LocalSystem LocalSystem LocalSystem UB

git add \*

git commit

git push

git add fileName

git add .

git add \*

git config --global user.email "legah2000@gmail.com"

git config --global user.name "Legah2000"

git init = to initialised a git empty repository

create the default branch[master, main]

git status = to track file's location

workingArea/stagingArea/LocalRepo/RemoteRepo :

git log = list all commits/versions

git add = move files to staging area from working area

git commit -m = move files to the local repository FROM STAGING AREA

good commit message for tracking changes created by authors are recommended

git show commitID = Shows changes in a commit

git config --global user.email "legah2000@gmail.com"

git config --global user.name "Legah2000"

git remote add paypal https://github.com/landmark34aa/paypal

The Alias for:

paypal --> https://github.com/landmark34aa/paypal =

$ git push paypal master

Username for 'https://github.com':

legah2045

Password/PAT':

ghp\_NI66n7lKAE853AoCISSxY2WxemaqEd0pPr1d

git remote add origin https://github.com/landmark34aa/paypal-fe.git

git remote add pp https://github.com/landmark34aa/paypal-fe.git

git branch -M main

git push -u origin main

Legah

chidi / dominion / mary

workingArea stagingArea LocalRepo RemoteRepo :

LocalServer LocalServer LocalServer SCM/github

RED Green - -

Incomplete completed versioned versioned/Distributed

vi/vim git add git commit git push

Developer2: Mary

project repo: https://github.com/landmark34aa/paypal-fe

git clone https://github.com/landmark34aa/paypal-fe

git config --global user.email "mylandmarktech@gmail.com"

git config --global user.name "Chidi"

copy the remote repository:

https://github.com/landmark34aa/paypal-fe

git clone https://github.com/landmark34aa/paypal-fe

git pull/fetch

git pull aliasName branchName

git pull pp master

git pull origin master

git fetch aliasName branchName

git fetch pl master

IQ: What is the difference b/w git pull, git push, git fetch & git clone?

Git pull brings changes in the code to the local repository and workingArea

Git fetch brings changes to the code in the local repository ONLY

which can be merge after review [git diff]

Git push brings changes in the code to the remote/shared repository from LocalRepo

Git clone brings the entire code to the Local Repo/environment

working area staging localRepo RemoteRepo[GitHub] :

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

Untracked files git add \* git commit git push

red green

-----> <-------git pull/fetch------

Git Branch :

In git a branch is a line of Development

development branch

stage branch

master branch

dev

stage

master

git branch -help

==================

git branch branchName = creates a new branch

git branch development = created the development branch

since we created the development branch from the master branch,

The master branch is the upstream of the Development branch and

The Development branch is the downstream of the master branch

git switch branchName or git checkout branchName

git switch development or git checkout development

git diff development

git merge development

git push aliasName --all = push changes to all branches

git push pl --all

Merge conflict occurs when additinal commits are done in new branch without merging.

For best practices avoid merge conflict by merging before any additinal commits in a new branch

Merging changes in the master from other branches.

1. Create a pull request - PR for other team members to review

2. after reviewing then merge the changes

How to get code from the RemoteRepo in GitHub/SCM to the LocalRepo :

git fetch --> It brings changes in the code to the LocalRepo

git pull --> It brings changes in the code to the localRepo and Working area

git clone --> It brings the entire code to the working area

From LocalRepo to RemoteRepo = git push

From RemoteRepo to LocalRepo = git fetch or git pull OR git clone

Pull Request- PR:

PR is use to review changes in one branch before merging in another branch.

This is recommended for any merge in the master/release branch

development branch

staging branch

master branch

IQ: How many branches are you supporting in your projects?

Branches are used to create lines of development in git.

A branch is a line of development

we support a minimun of 3 branches dev, stage and master branch

How can we merge CODE changes in git??

1. git merge

2. git rebase

IQ: How can we merge a particular commit among many commits in git??

By using the;

3. git cherry-pick commitID

How to merge changes in git:

git merge

fc1[version1] ---> fc2[version2]--->fc3[version3]

= mc1 [master version1]

git rebase

fc1[version1] ---> fc2[version2]--->fc3[version3]

= mc1 [version1] -->mc2 [version2] --->mc3 [version3]

git cherry-pick = merging specific commits

$ git log

commit da5a3cbbb6c984794b0d4e43e2ab23139ec4230e (HEAD -> feature)

Author: Chidi <mylandmarktech@gmail.com>

Date: Thu May 23 19:29:14 2024 -0400

banking.py created

commit 04565922bf3960b54edf4ebe029f8795df604b25

Author: Chidi <mylandmarktech@gmail.com>

Date: Thu May 23 19:27:07 2024 -0400

deployment feature modified

commit c9f3960be994feef70eb167e3882e34f41c7c10c

Author: Chidi <mylandmarktech@gmail.com>

Date: Thu May 23 19:25:44 2024 -0400

online feature created

git cherry-pick 04565

git clone https://github.com/landmark34aa/paypal-fe

Resloving issues: debugging:

working area staging localRepo RemoteRepo :

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

Untracked files git add git commit git push

red green

git clean -n

git clean -f git reset git revert commitID

git clean -n = Indicates files to deleted in the working area

git clean -f = Delete files in the working area

git reset = Moves files from staging back to working area

git revert Cid = Delete the latest committed files

in the LocalRepo for a specific commit

git tags:

=======

git tag paypal-1.0.2

1. = higher version

0. = lower version

2 = patch

paypal-1.0.2.zip

difference b/w tags and branches :

=== Tags ==Branch

immutable mutable

unmodifiable modifiable

After Production development-WIP

Master Recommended any branch

git tag TagName git branch BN

git push an --tags git push an --all

git push AliasName tag TN git push an BM

git tag -d TN git branch -d BN

git branch strategy :

dev branch ---> develoment Environment

stage branch ---> Testing/QA/UAT Environment

master branch ---> PRODUCTION

feature

bug\_fixed

What is the difference between Git & Github:

Git is an open source Distributed Version Control System(DVCS) which records

changes made to files (codes) laying emphasis on speed, and data integrity in a

distributed non-linear workflows.

GitHub is a source code management[SCM] tool maintained by microsoft

Explain your branching strategy in LandmakTechnology ??

Explain your branching strategy in Dominion systems ??

Explain your branching strategy in your Environment ??

1. Number of branches = WE maintain 3 branches

dev branch ---> develoment Environment

stage branch ---> Testing/QA/UAT Environment

master branch ---> PRODUCTION

feature

bug\_fixed

2. Branch security =

All commits in the master branch must go through PR

3. Switching branches = working tree should be clean

DevOps Engineer [dev branch / master--bugfixed]

If neccesary backup incomplete tasks before switching

git stash command

4. git rebase over git merge

5. review changes [git diff] before merging

6. Use PR before committing to the master branch

7. avoid merge conflicts

git stash = to take backup

git stash list

git stash apply = resume wip for the most recent backup

git stash apply stash@{0} = resume wip for the 1st backup

git stash apply stash@{5} = resume wip for 6th backup

git stash drop stash@{0} = will delete backup

git stash pop stash@{1} = It apply & drop backup 1.

git stash pop

$ ssh-keygen

$ ls /home/ec2-user/.ssh/

authorized\_keys

id\_rsa = ssh-private-key

id\_rsa.pub = ssh-public-key

known\_hosts

known\_hosts.old

https url = https://github.com/landmark34aa/paypal-fe

ssh url = git@github.com:landmark34aa/paypal-fe.git

git remote -v

git remote add plssh git@github.com:landmark34aa/paypal-fe.git

git remote add pl https://github.com/landmark34aa/paypal-fe

Github-webhook --- Jenkins :

.gitinore file

Developers use IDE to write code

IDEs include = vscode, esclipes, atom, etc.

.classpath modules

1. workingArea = untracked

2. stagingArea = git is tracking changes

3. localRepo = versioned

4. SCM/github/remoterepo = versioned

.gitinore file are files developers shouldn't be tracking changes in

because it is not part of the Development project.

1. developing FinTech java app for BOA

app.java /

README.md

metadata/commentary

.classpath

modules

Git Best Practices:

Use branching strategy and pull requests

Commit once you finish your task.

Avoid merge Conflicts.

Don’t Commit Half-Done code - backup with git stash

Test your code before committing.

Write Good Commit Messages before committing

Try to use git commands rather than GUI tools.

use ssh url and PAT over https url

use rebase over git merge

use branch protection strategy

use best branching strategy