**PROJECT**

**Branching Development Model**

DESCRIPTION

Create a branching model to help your team understand the Git Feature Branch Workflow for faster and efficient integration of work

**Background of the problem statement:**

M-theta Technology Solutions hired you as a DevOps Architect. It is undergoing an infrastructural change to implement DevOps to develop and deliver the products. Since M-theta is an Agile organization, they follow the Scrum methodology to develop the projects incrementally. Hence, the company wants to adopt Git as a Source Code Management (SCM) tool for faster integration of work and smooth transition into DevOps.

So, as a DevOps Architect, you have been asked to build a branching model to demonstrate the Git Feature Branch Workflow for the company’s engineering team. In the branching model, you are required to create a Production branch which will act as the main (master) branch, an Integration branch which will again have two branches inside it namely Feature 1 and Feature 2, and a Hotfix branch which will be used for fixing any issues that could come up from Integration or Production branches.

**Following Steps to perform to achieve above Goal:**

* Login to Github and create repository for project – final\_project
* Open terminal and create a local directory named – final\_project

mkdir final\_project

cd final\_project

* Initialized git in this directory

git init

* Initialied remote respository

git remote add origin https://github.com/mailsanjaykumar/final\_project.git

1. Start with the Production branch (master branch), and then create a HotFix  and Integration branch

* Make branch hotfix and integration

git branch integration

git branch Hotfix

1. Subsequently, create Feature 1 and 2 branches that integrate to the Integration branch as shown in the above figure

* Make branch feature1 and feature2

git branch feature1

git branch feature2

1. Commit some changes in the Feature 2 branch and merge it into the Integration branch. Delete this branch once merging is complete

* Checkout to feature2

git checkout feature2

* Create file in this branch

vi f1.html

* Merge this f1.html to Integration branch

git merge integration

* Delete the branch feature 2

git checkout master

git branch –D feature2

* Pushing all changes to remote repository

git push --all

* Deleting branch feature from local and remote repository

git branch –D feature2

git push origin –delete feature2

1. Commit some changes in the Feature 1 branch and rebase it to the Integration branch

* Change the file f1.html in feature1

Vi f1.html

* Make changes and push to remote

git status

git add f1.html

git commit –m “ change”

git push origin feature1

Enter username and password to push

* Rebase feature1 branch to Integration branch

git rebase –i Integration

git push –f origin feature1

1. Merge the Integration branch into Hotfix and Production branch to update these branches

* Mergeing the integration branch to hotfix and production branch(master branch)

git checkout master

git merge Integration

git push origin master

git checkout HotFix

git merge Integration

git push origin HotFix

1. Commit some changes in Feature 1 branch, and then merge it into Integration, Hotfix, and Production branch. Delete this branch once merging is complete
   * Checkout to feature branch and do changes in f1.html file

Git checkout feature1

vi f1.html

* + Add some change in file

<h1>Feature1 branch new addition</h2>

* + Adding and committing and pushing to remote repository

git add f1.html

git commit –m “feature new inte change”

git push orign feature1

* + Merging to integration,HotFix and production (master) branch

git checkout Integration

git merge feature1

git push origin Integration

git checkout HotFix

git merge feature1

git push origin HotFix

git checkout master

git merge feature1

git push origin master

* + Deleting feature1 branch from local and remote repository

git checkout master

git branch –d feature1

git pull --all

git push origin –delete feature1

1. Commit some changes in the Hotfix branch and merge it into the Production as well as the Integration branch
   * Checkout to hotfix branch and do changes in f1.html file

git checkout HotFix

vi f1.html

* + Add some change in file

<h1>HotFix branch new addition</h2>

* + Adding and committing and pushing to remote repository

git add f1.html

git commit –m “Hotfix new inte change”

git push orign HotFix

* + Merging to integration and production (master) branch

git checkout Integration

git merge HotFix

git push origin Integration

git checkout master

git merge Hotfix

git push origin master