**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

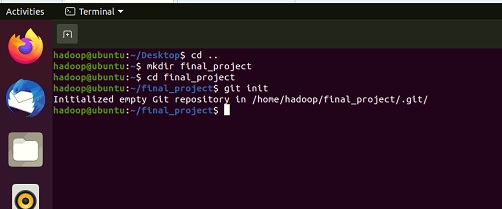
**Background of the problem statement:**

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.

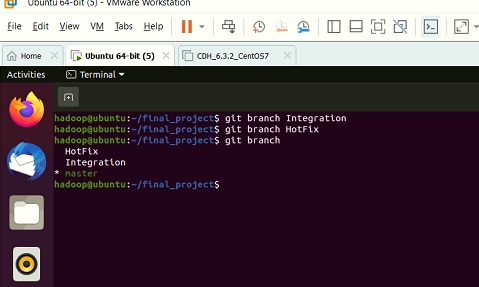
So, as a DevOps Architect, you have been asked to build a branching model to demonstrate the Git Feature Branch Workflow for the company’s engineering team. In the branching model, you are required to create a Production branch which will act as the main (master) branch, an Integration branch which will 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:**

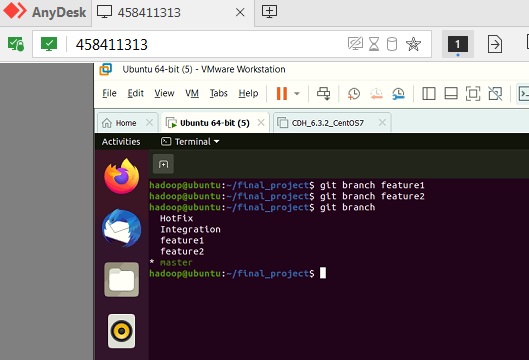
* Login to Github and create repository for project – final\_project
* Open terminal and create a local directory named – final\_project
* Initialized git in this directory



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

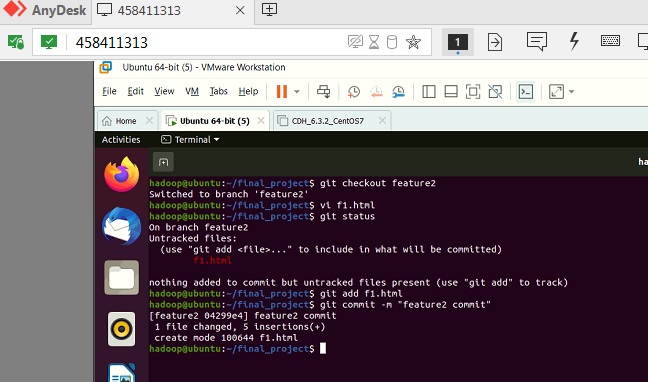


1. Subsequently, create Feature 1 and 2 branches that integrate to the Integration branch as shown in the above figure

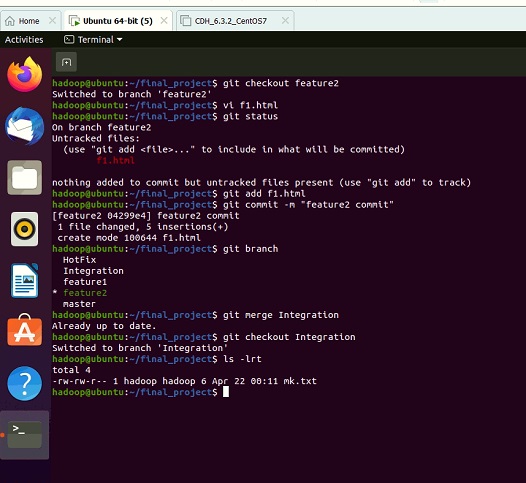


1. Commit some changes in the Feature 2 branch and merge it into the Integration branch. Delete this branch once merging is complete

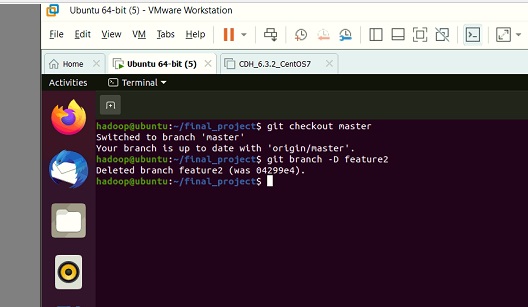
* Checkout to feature2
* Create file in this branch



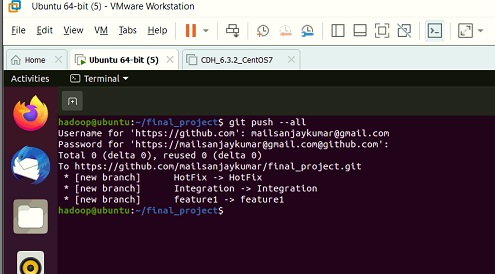
* Merge this f1.html to Integration branch

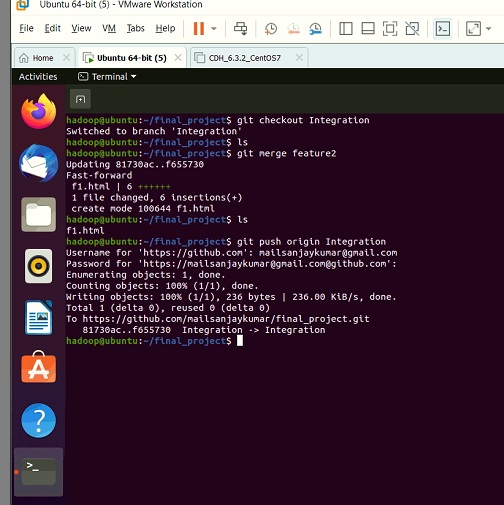


* Delete the branch feature 2

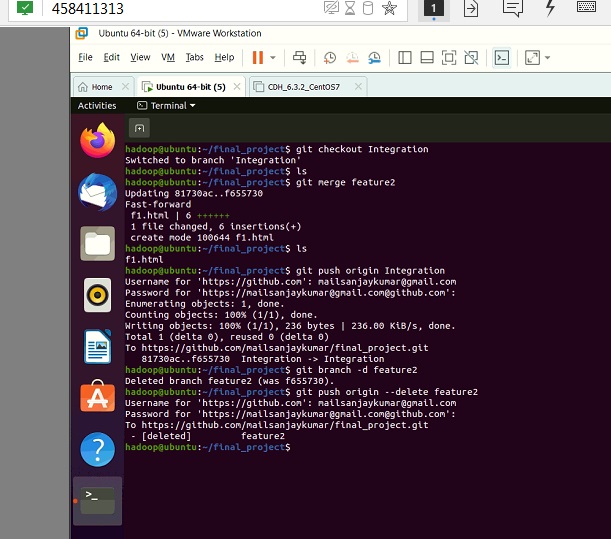


* Pushing all changes to remote repository



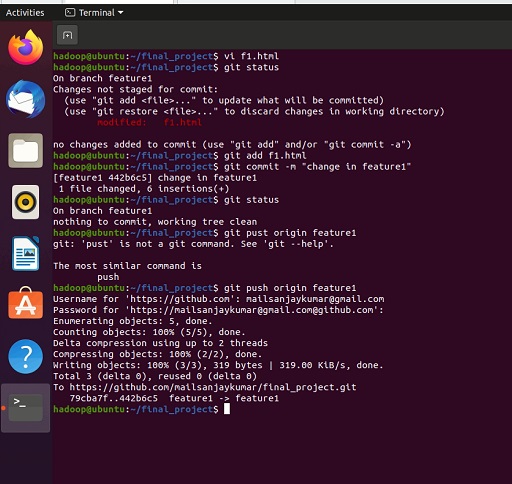


* Deleting branch feature from local and remote repository

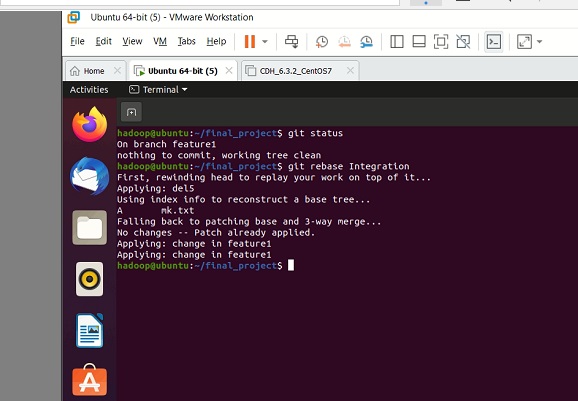


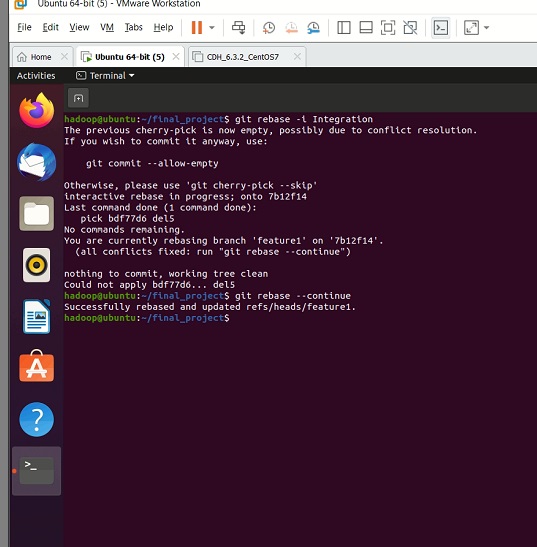
1. Commit some changes in the Feature 1 branch and rebase it to the Integration branch

* Change the file f1.html in feature1
* Make changes and push to remote



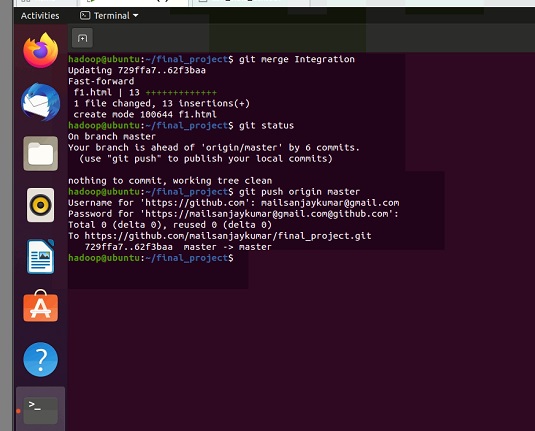
* Rebase feature1 branch to Integration branch

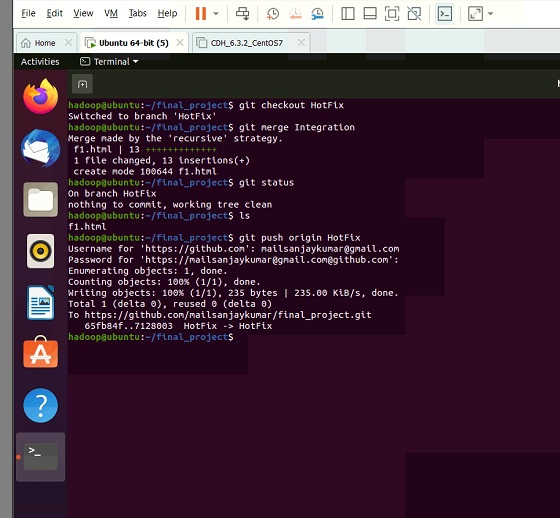




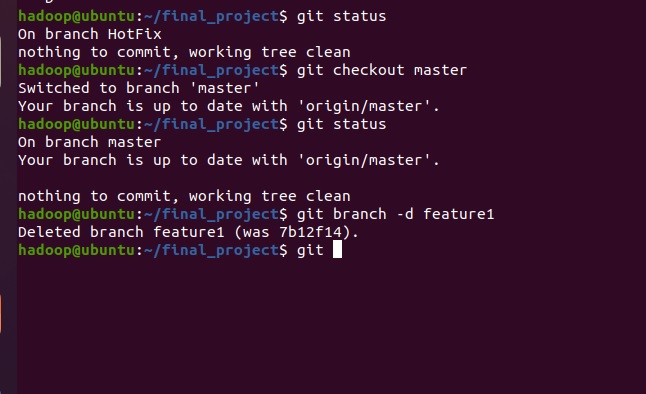
1. Merge the Integration branch into Hotfix and Production branch to update these branches

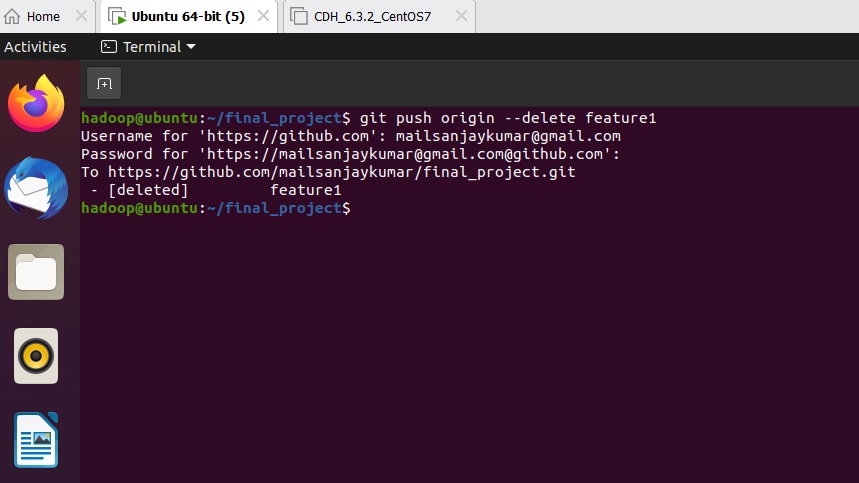
* Mergeing the integration branch to hotfix and production branch(master branch)



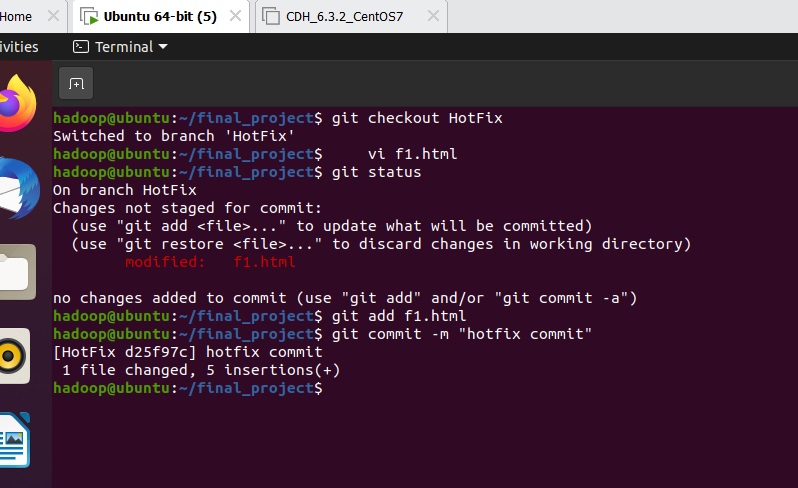


1. Commit some changes in Feature 1 branch, and then merge it into Integration, Hotfix, and Production branch. Delete this branch once merging is complete
   * Deleting feature1 branch from local and remote repository

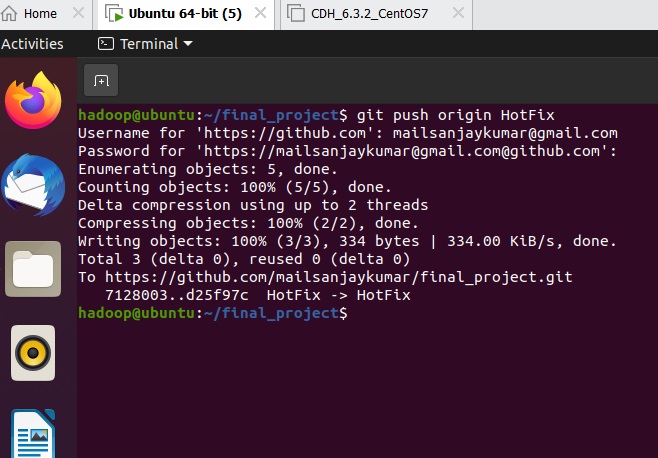




1. Commit some changes in the Hotfix branch and merge it into the Production as well as the Integration branch
   * Checkout to hotfix branch and do changes in f1.html file
   * Add some change in file, Adding and committing



* + pushing to remote repository



* + Merging to integration and production (master) branch

