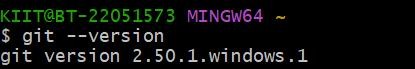
Week - 8

Aditya Raj

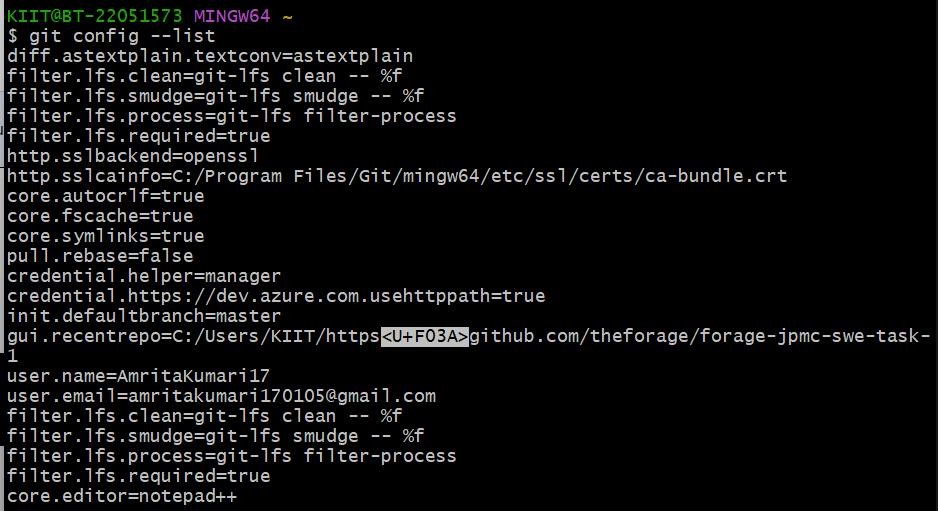
SuperSet Id :6363544

**1.Git-HOL**

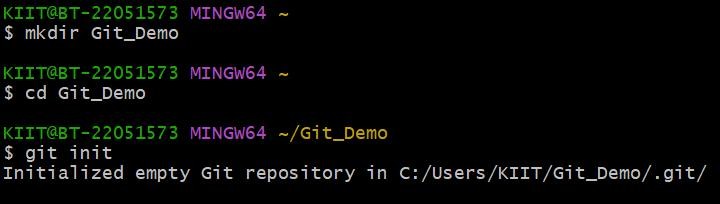
Installing Git



Configuring the username and user-email



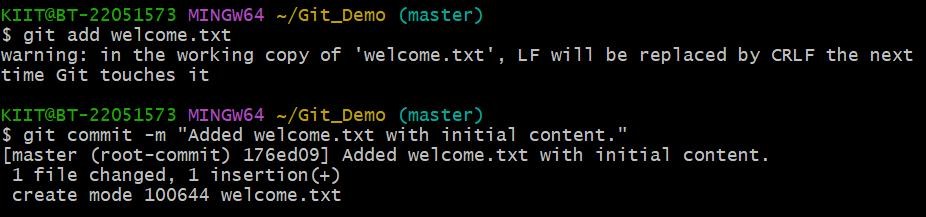
Making new folder and initializing git



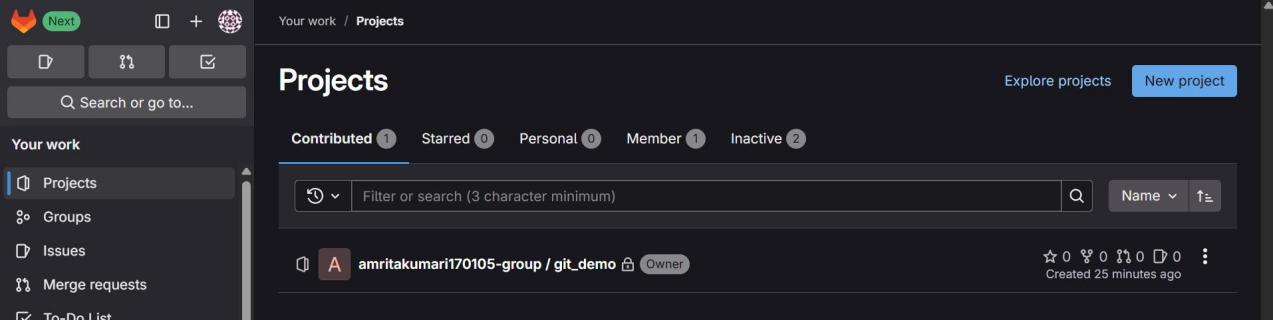
Creating and checking the file



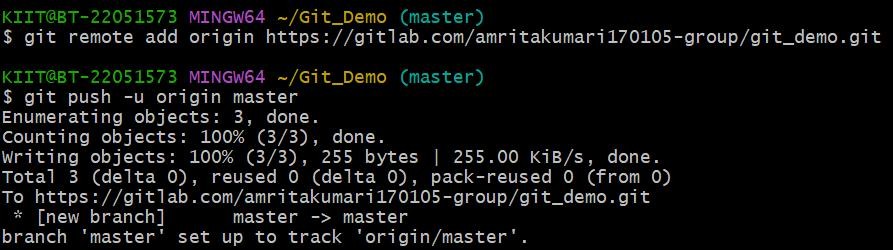
Adding file to git and commit



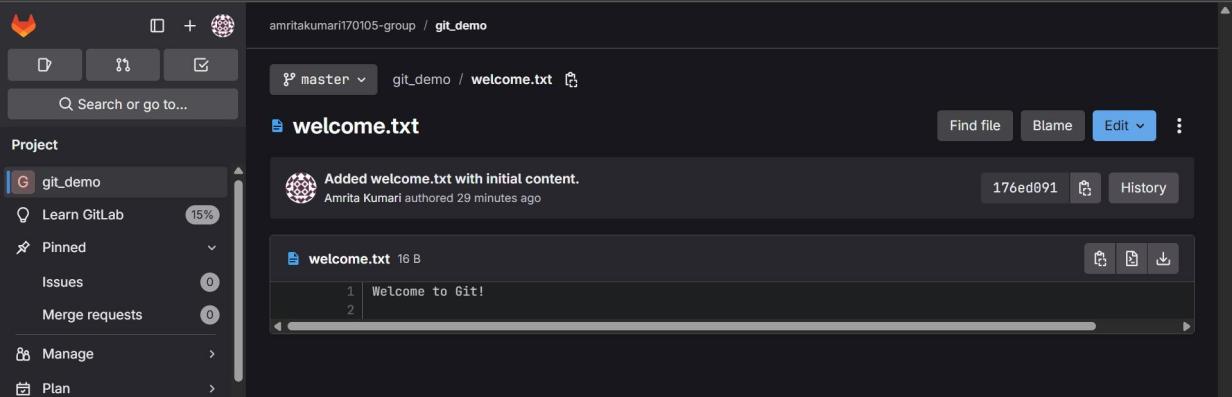
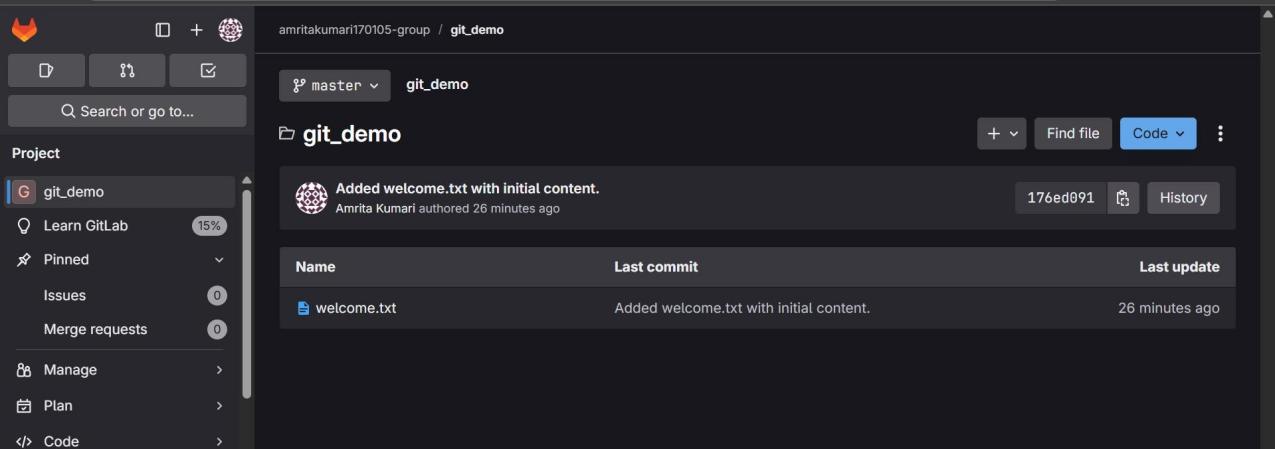
Creating GitLab respository



Link - https://gitlab.com/amritakumari170105-group/git\_demo.git Add remote and push to gitlab



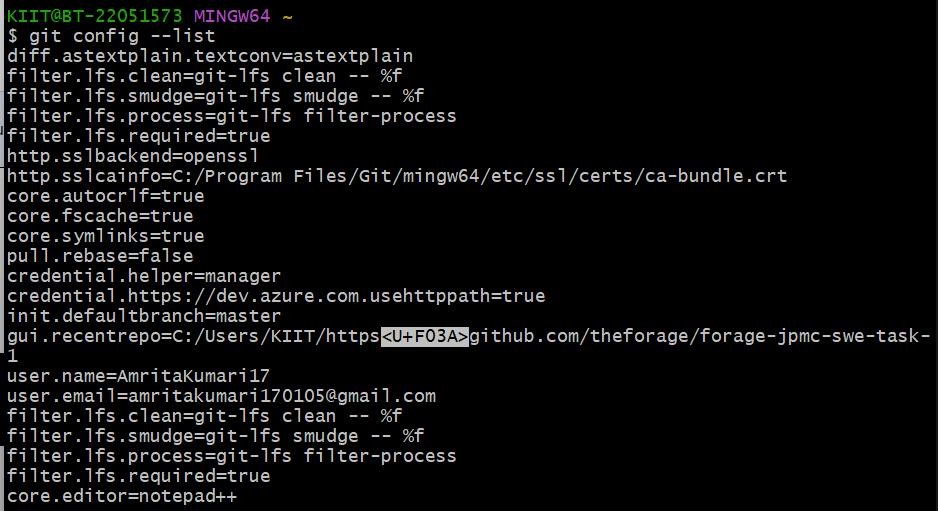
Confirming on GitLab:



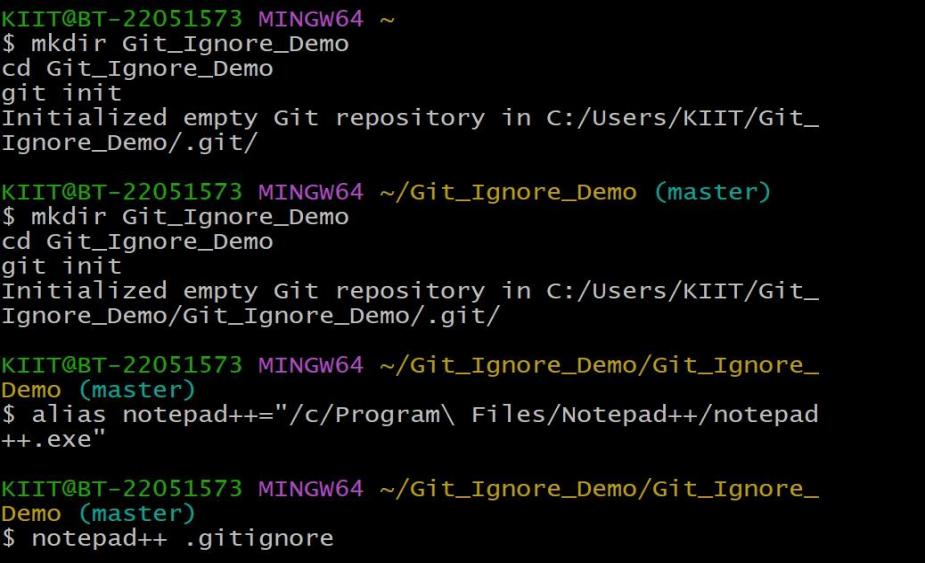
**2.Git-HOL**

Installing Git

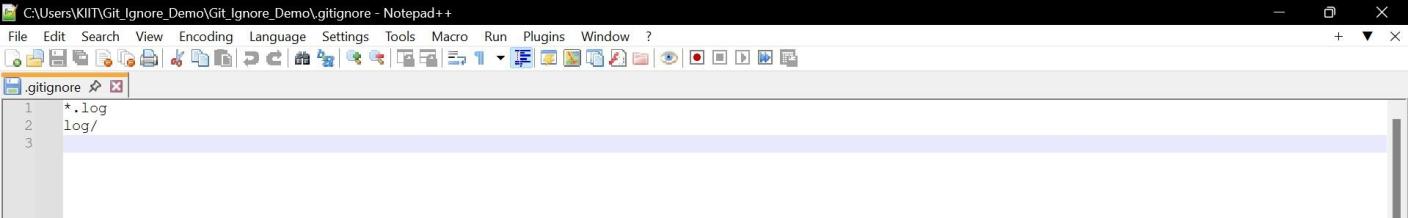
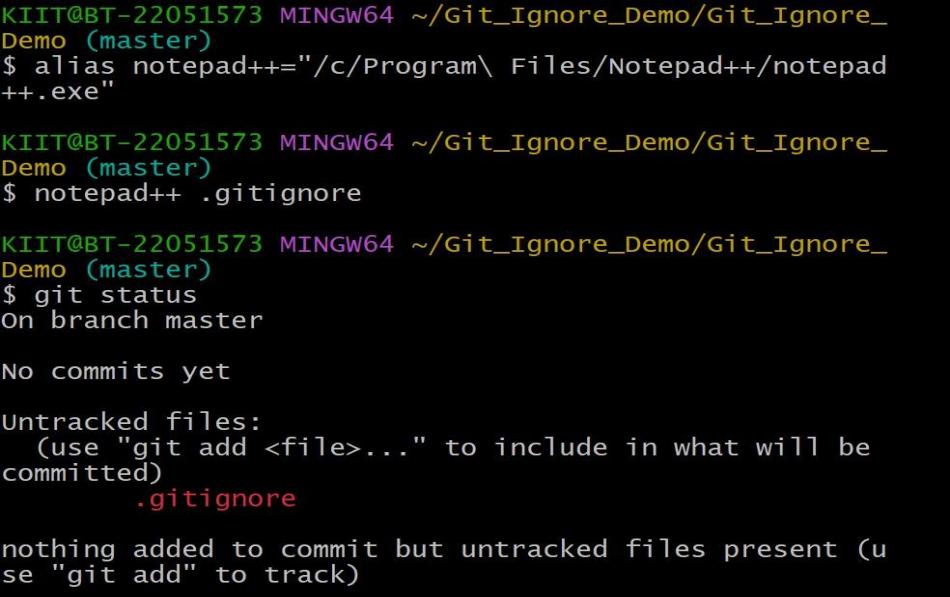
Configuring the username and user-email



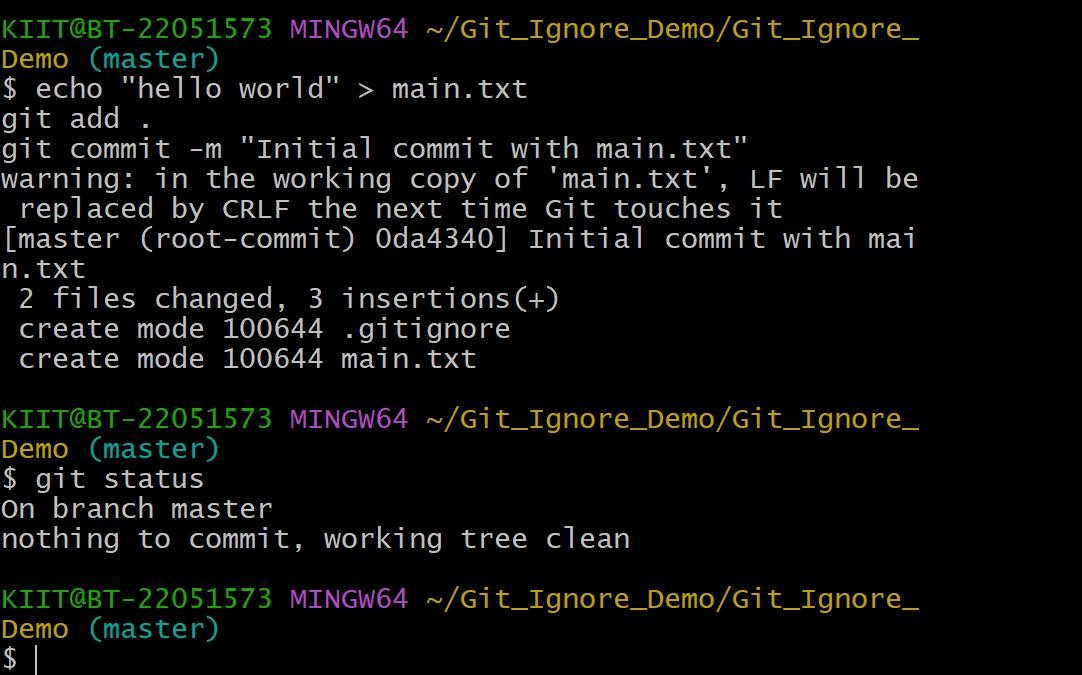
Making new folder and initializing git



Creating and checking the file

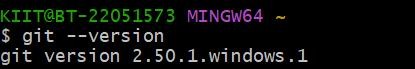


Adding file to git and commit

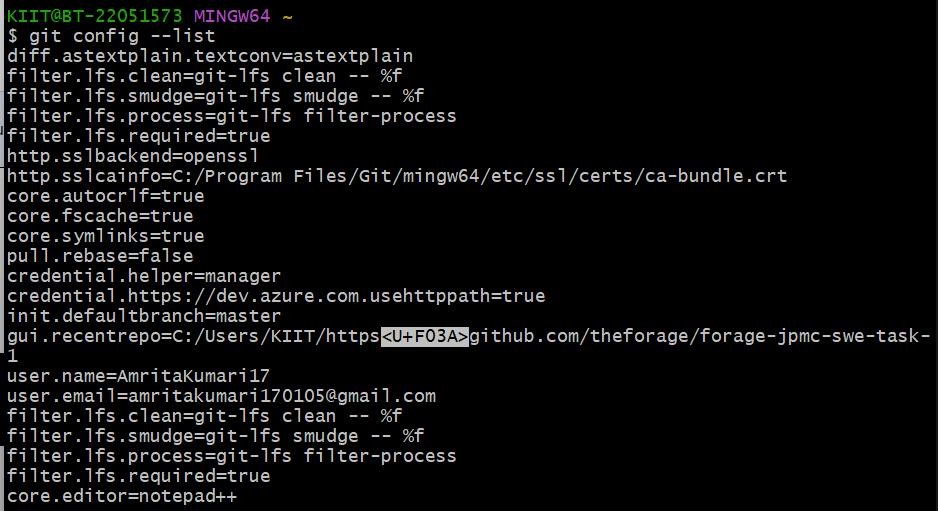


1. Git-HOL

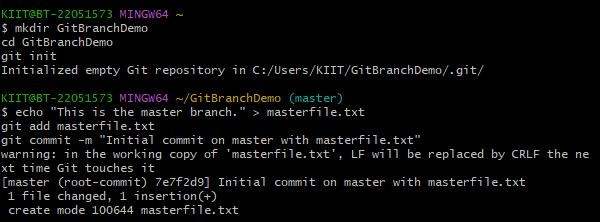
Installing Git



Configuring the username and user-email



Initializing git and creating file



# Branching:

* 1. Create a new branch “GitNewBranch”.



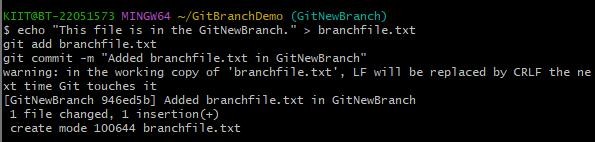
* 1. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.



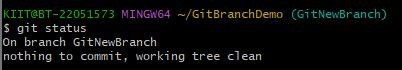
* 1. Switch to the newly created branch. Add some files to it with some contents.



* 1. Commit the changes to the branch.



* 1. Check the status with “git status” command.

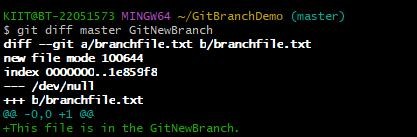


# Merging:

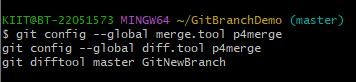
1. Switch to the master



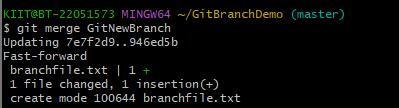
1. List out all the differences between trunk and branch. These provide the differences in command line interface.



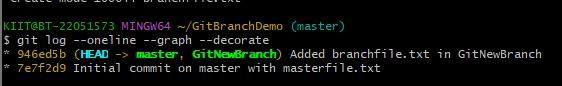
1. List out all the visual differences between master and branch using P4Merge tool.



1. Merge the source branch to the trunk.



1. Observe the logging after merging using “git log –oneline –graph –decorate”

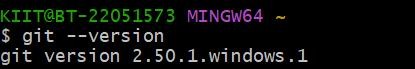


1. Delete the branch after merging with the trunk and observe the git status.

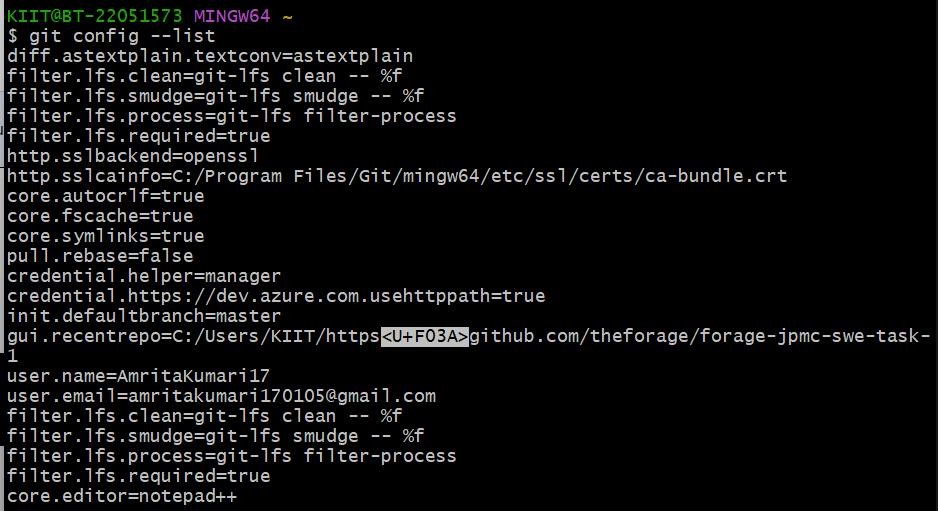


1. Git-HOL

Installing Git



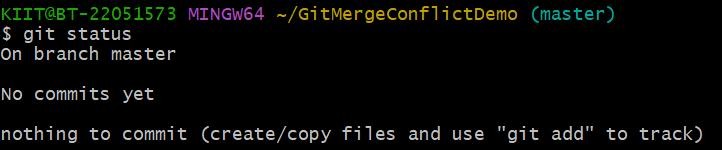
Configuring the username and user-email



Making new folder and initializing git



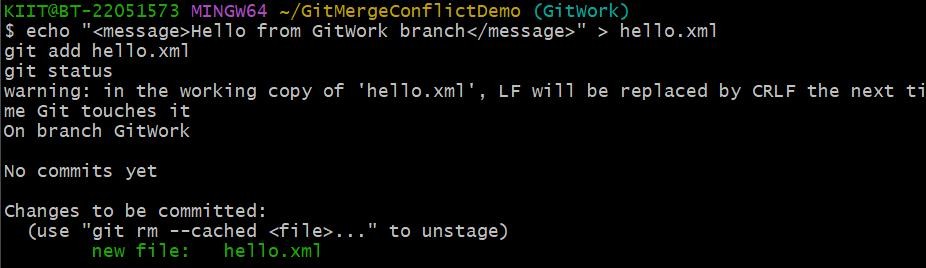
* 1. Verify if master is in clean state.



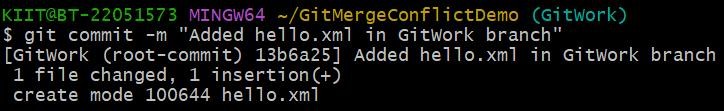
* 1. Create a branch “GitWork”. Add a file “hello.xml”.



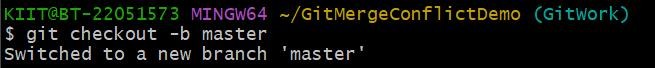
* 1. Update the content of “hello.xml” and observe the status



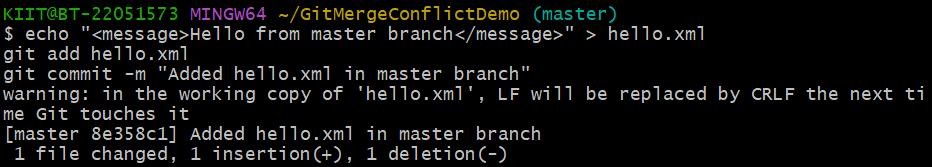
* 1. Commit the changes to reflect in the branch



* 1. Switch to master.



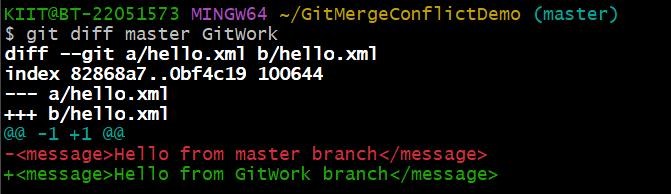
* 1. Add a file “hello.xml” to the master and add some different content than previous and 7. Commit the changes to the master



1. Observe the log by executing “git log –oneline –graph –decorate –all”



1. Check the differences with Git diff tool



1. For better visualization, use P4Merge tool to list out all the differences between master and branch



1. Merge the bran to the master



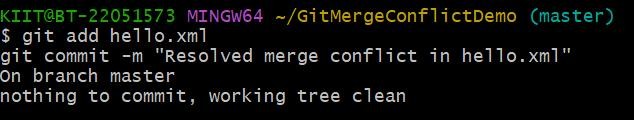
1. Observe the git mark up.



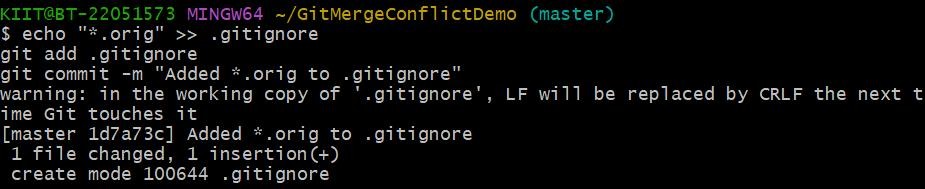
1. Use 3-way merge tool to resolve the conflict



1. Commit the changes to the master, once done with conflict



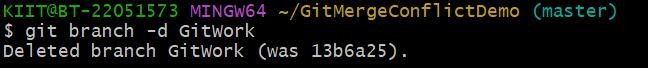
1. Observe the git status and add backup file to the .gitignore file and 16. Commit the changes to the .gitignore



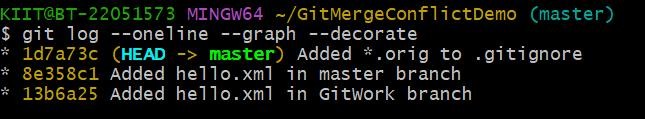
1. List out all the available branches



1. Delete the branch, which merge to master.

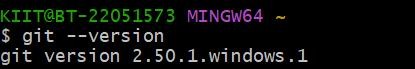


1. Observe the log by executing “git log –oneline –graph –decorate”

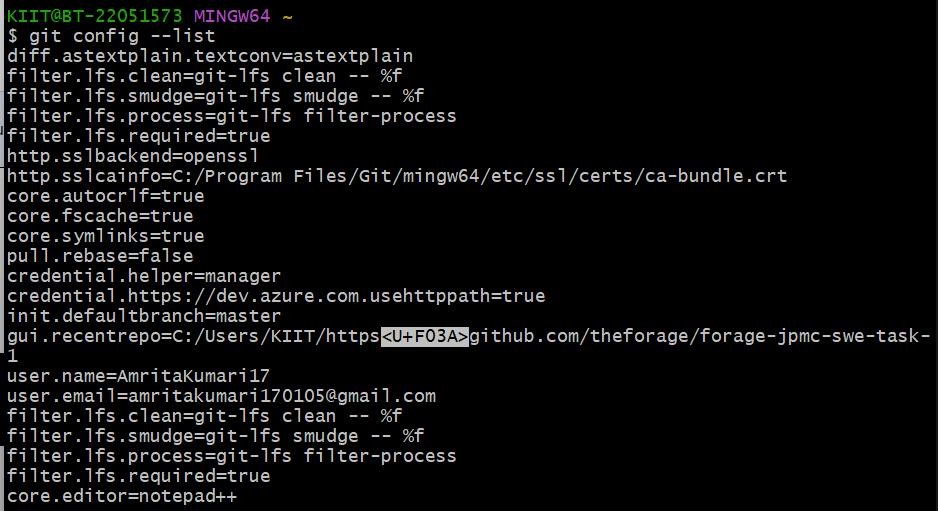


1. Git-HOL

Installing Git



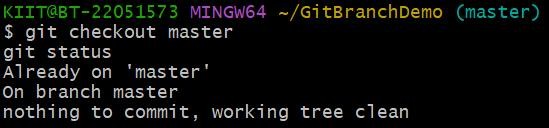
Configuring the username and user-email



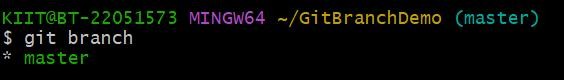
Navigating to the local repo



* 1. Verify if master is in clean state.



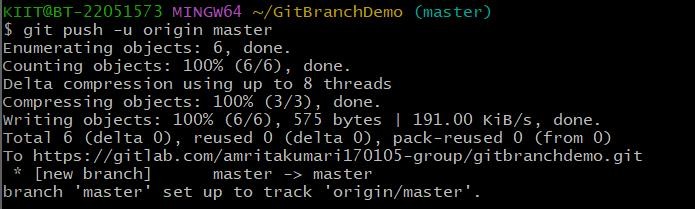
* 1. List out all the available branches.



* 1. Pull the remote git repository to the master



* 1. Push the changes, which are pending from “Git-T03-HOL\_002” to the remote repository.



* 1. Observe if the changes are reflected in the remote repository.

