GIT:-

(Not required for the lab, just if you guys are interested, these commands were automatically run when the container was created)

After initializing the git repo inside a git repository, you may run into some error about dubious ownership of the directory.

To avoid this run the following command:-

git config --global --add safe.directory /home/labDirectory/working_directory

This is because we have created a git repo inside another git repo, which is a security issue for git.

There is a very interesting reason why git has this feature, for those who are interested feel free to

read up here :-
https://github.blog/2022-04-12-git-security-vulnerability-announced/#cve-2022-24765

git config user.email “24b0973@iitb.ac.in”

git config user.name “Shishir”

(Don't use the global tag, it may mess up the permissions of the VLab Git)

—--------

Note:-

There are three folders to think about with git working directory,local repo and origin repo

Working directory -> local repo is commit

Local repo -> origin repo is push

Origin repo -> local repo is fetch

Local repo -> working directory is checkout

Origin repo -> working directory is pull

—--------

IMPORTANT git commands

git log --oneline --name-only -> this allows you to see a view all all the prev commits and which files were modified in each one(you wont see what each modification is though for that remove –name-only)

–oneline just makes it more compact which is good the commit hash will also be there

git log --oneline --<path-to-file> shows you which commits a file was modified

now you can use **git checkout <commit hash>** to access that commit make any changes and then save the new thing as a NEW branch via **git switch -c “name”**

git checkout master will move HEAD back to master :)

To commit every thing all changes use **git add -A** and then git commit. If you need to modify previous commit then use **git commit –amend**

Rename branch: git branch -m old-name new-name

Delete branch: git branch -d new-branch (-D to force)

Create branch: git branch new-branch

Switch branch: git switch new-branch (git checkout new-branch)

Create & switch: git switch -c new-branch (git checkout -b new-branch)

Git checkout master makes HEAD go to master branch

git merge -m "merging" testing this will create a new commit in the current branch you are on with message “merging” and combine testing with it

How to deal with merge conflict?
Once you get merge conflict what you do is that you resolve the conflict by opening all the files and checking everything and then you **git add -a** and then **git commit -m “resolved Merge conflict”**

git show abc123:file4.cpp > file4_oldversion.cpp THIS insanely good command can be used to take the file4.cpp from the branch abc123 and put its contents in file4_oldversion.cpp