Task1

Primary purpose of Git and Github

1. Git help to solve the problem faced in synchronizing document when working on same project. All copies reside on local work area. Git helps to manage changes on the projects
2. It helps us to keep track of changes – versioning
3. Collaboration with the documents happens in a web-based tools called Github
4. Github has a user interface where we can collaborate and work with projects, we can record project on Git and virtualize changes on Github,
5. Github provide team with feedback to make peer review while collaborating on projects
6. Provides details because its webhost, it is an open source, hence there is a huge community to help each other to collaborate on project for free.
7. Tracking changes using git to manage the project
8. Git records all changes, step-by- step, therefore problem identification becomes possible
9. It allows multiple developers to work together
10. It supports non-linear development through its thousands of parallel branches

2.

Explain the concept of forking repository on Github. How does it enable collaborative coding

Fork is to create a copy of the project on Github. Forking will create copy of same repository in owns account. This becomes easy to work on the project without affecting the original project.

Once changes are made and a bug is fixed, you can commit it to the original owner if accepted.

3.

In software development version control is the management of changes to documents, files, or any other type of data,

* it is essential for managing and tracking changes to the codebase
* it helps by ensuring code quality, reducing errors,
* and improving collaboration among developers and all team members for ease of project delivery

Without version control, managing and tracking code changes would be a difficult and error-prone task.

4.

Importance of peer review process on Github

1. Peer review provides better knowledge for new and experienced developers. New areas of the code can be experienced and also helps team to reduce their bus factor.
2. Peer review is very important in software development, it is very essential for improving code quality
3. Peer reviews helps developers to collaborate among team members, this can be done by frequent code reviews

5.

Name at least three popular Git Gui tools that developers can use to manage their git repositories. What is the advantage of using Git Gui over CLI for git operations?

* TortoiseGit :

It is an open-source and free software

Windows shell interface for Git.

* Sourcetree:

free Git GUI client

that works on both Windows and Mac.

* GitHub Desktop

Web-based and open source

Advantage of using Git Gui over CLI for git operations:

* Ease of use
* It does not required writing and learning command
* Git GUIs offer a more visually pleasing and interactive experience. You can look at the commit history, you can click on each commit and see what has happened in that commit, see who made it and so on.
* GUIs offer better multitasking and control
* It is user-friendly than a command line
* a visual file system is utilized
* GUI users have windows that enable a user to view, control, manipulate, and toggle through multiple programs and folders

TASK2

1. What is the differences between git and Github:
2. Git helps to keep track of changes and file. It helps to manage all version of the file.
3. With git, you can manage source code, fixing bug
4. GitHub, meanwhile, is where people can share and collaborate on the files they have created.
5. Easier to make changes to the project using git

GitHub

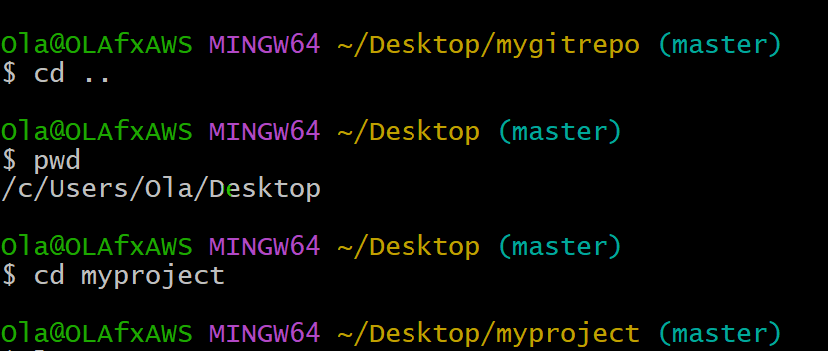
Github helps to

* Create : It is called “a social coding platform”GitHub lets users create requests from one another and internally discuss the iterations
* Store : it is a SCM – source code manager
* Change : It allows developers to edits and make changes that the other collaborators also see and interact.
* Merge : developers can make changes in github, changes merged during collaboration
* collaborate on files or code : Github is used in software development by helping access the GitHub repository.

1. How do you initiate Git repository and significant of git add and git commit commands in version control:

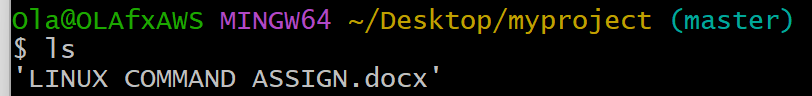
To initiate git repository, we follow the following steps:

1. Create a folder or repository in our local
2. Assuming this folder created is still empty –
3. We make sure we are in the file by using “pwd” command
4. We use cd command to move into our local repository



From above illustration, we are in a directory called myproject (can be found on the Desktop)

We have used git add . command to add document to this folder( by doing “ls” command. Ls linux command tells us if the file is empty. In this case, we have



Linux Command Assign.docx in “myproject” folder.

Git Commit: for git to make changes, we need to track the changes by committing the changes: We use “git commit” command.

“Git commit” take a snapshot of the code

1. Explain the purpose of using branches in git, and how to create branch using git command

Branching is used to protect the working copy by creating a branch. We can work on the branch without affecting the original work. Git use branching to deal with that.

Branch is a working copy of the develop branch. This helps to keep the previous working copy save and secure somewhere, then changes can be made without affecting the original/previous copy.

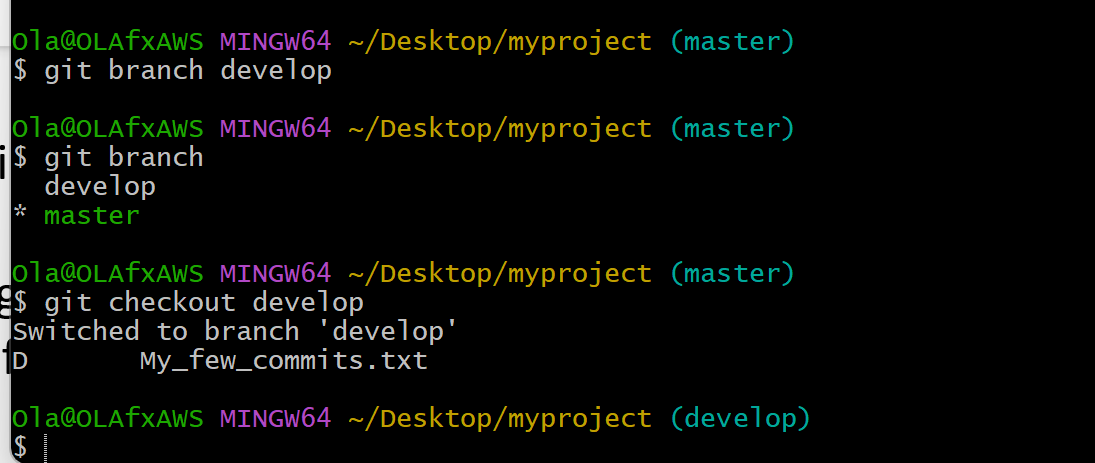
Lets create our first branch command

Git branch develop

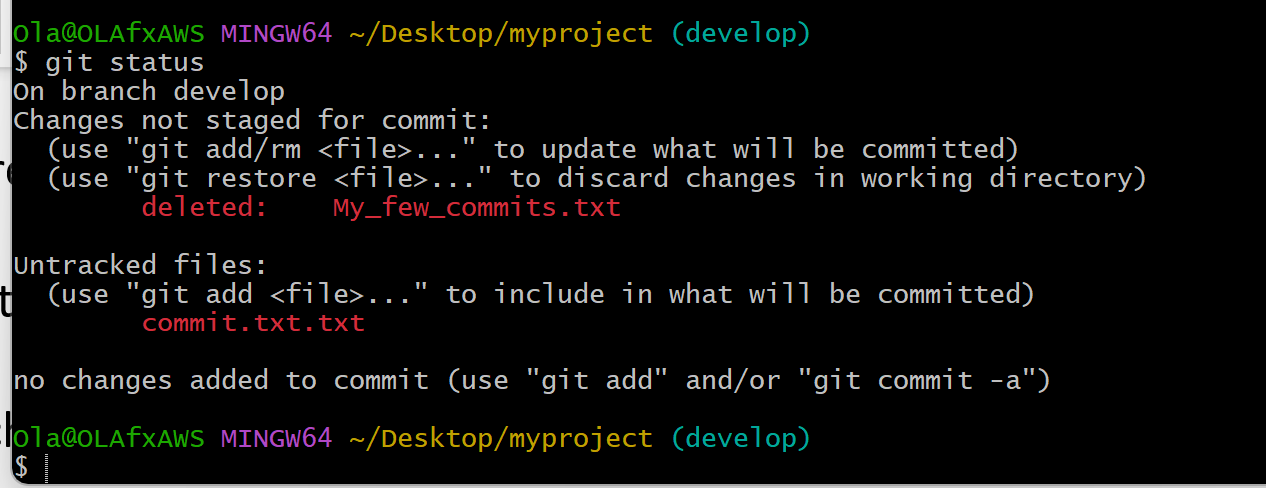
Git branch

----we can then switch to the using the command:

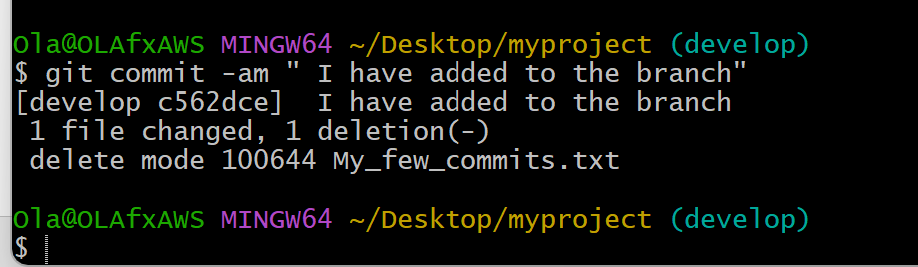
Git checkout develop ------ this command help to move to branch



Lets check our git status



As seen above, we are now in the develop branch. We can do git add to move from working area to stagging area. Alternatively, lets do ------------- “git commit -am” git commit -am will help skip the git add and commit to branch.



We can also create branch using command:

git checkout -b feature/new-feature

create a new branch reference existing branch we are working on. Ass explained above:

git checkout develop ------- that is the working branch

then in summary:

git checkout -b feature/new-feature ---------- this will create a new branch reference off the develop branch(previous), then create a new branch based on develop branch. If on master branch, it will reference off master branch instead.

1. What is the importance of creating descriptive commit messages when recording changes in Git repository?

Whenever we are looking at our snapshot to know where to go back to, detailed and clear message is vital good to see what we have done. Detail message tells us what we did on the previous commit should we need to go back to the commit.

Commit help us/ or point in time where we can save the project and come back to it at a later date. We have done the following: “git commit -m”

-m shows adding detail message to the project.

1. Making some changes
2. Adding to the changes
3. And commit the changes to github(save)

5.

1. Explain the purpose of using branches in git, and how to create branch using git command

Branching is used to protect the working copy by creating a branch. We can work on the branch without affecting the original work. Git use branching to deal with that.

Branch is a working copy of the develop branch. This helps to keep the previous working copy save and secure somewhere, then changes can be made without affecting the original/previous copy.

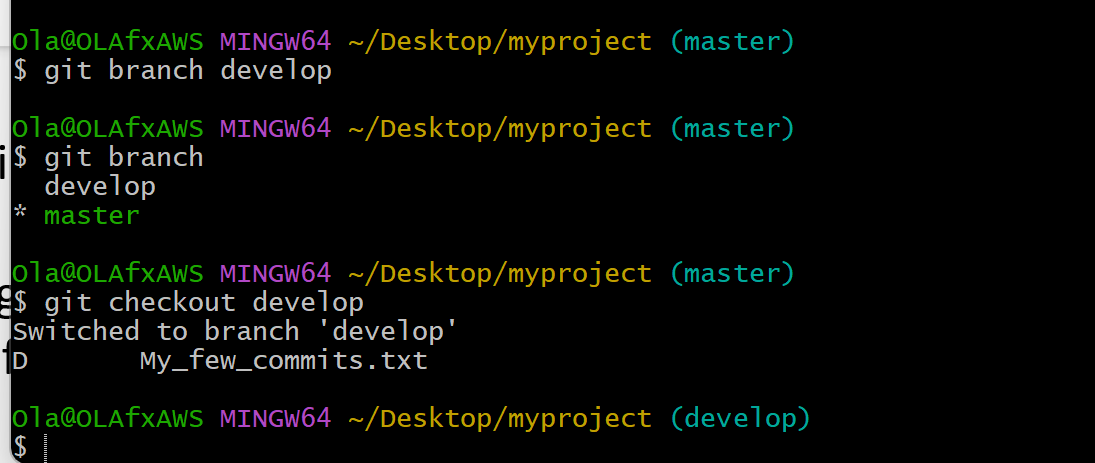
Lets create our first branch command

Git branch develop

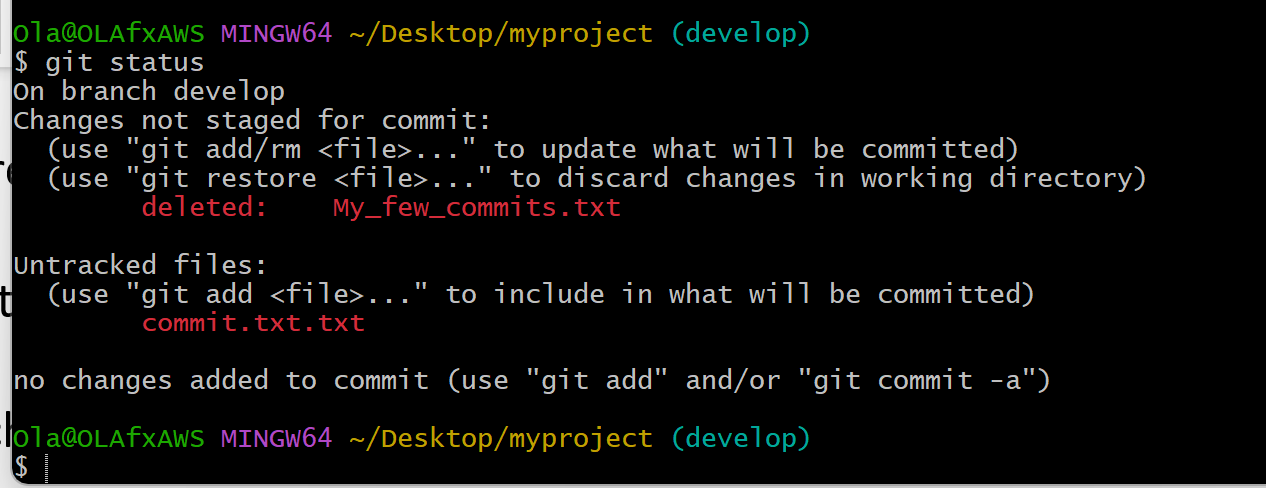
Git branch

----we can then switch to the using the command:

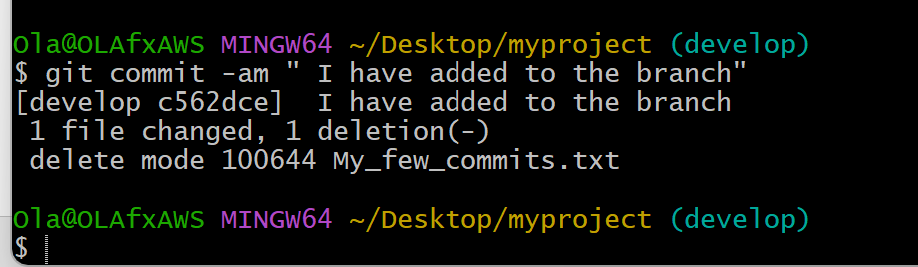
Git checkout develop ------ this command help to move to branch



Lets check our git status



As seen above, we are now in the develop branch. We can do git add to move from working area to stagging area. Alternatively, lets do ------------- “git commit -am” git commit -am will help skip the git add and commit to branch.



We can also create branch using command:

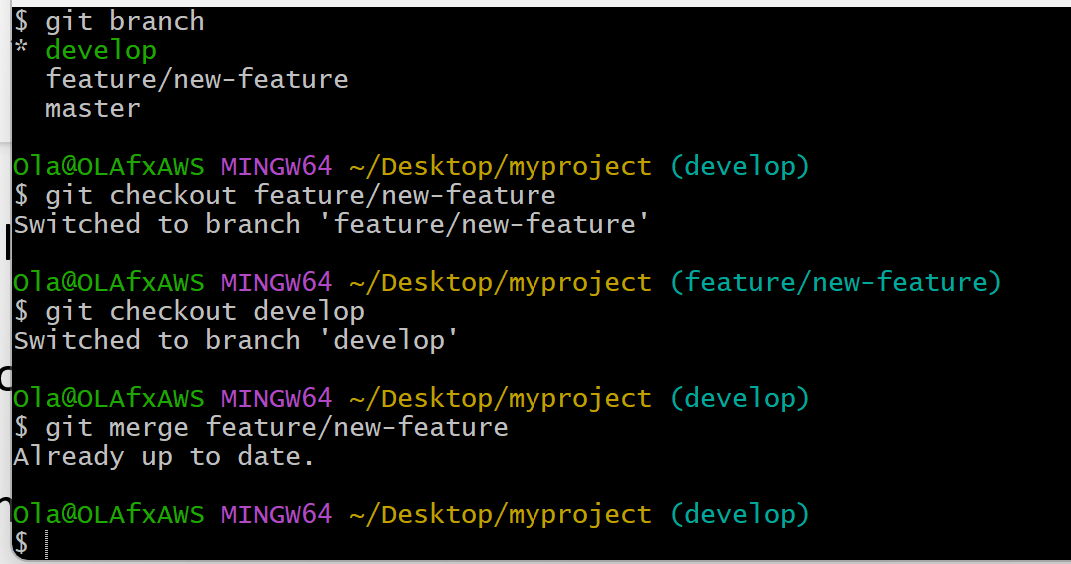
git checkout -b feature/new-feature

create a new branch reference existing branch we are working on. Ass explained above:

git checkout develop ------- that is the working branch

then in summary:

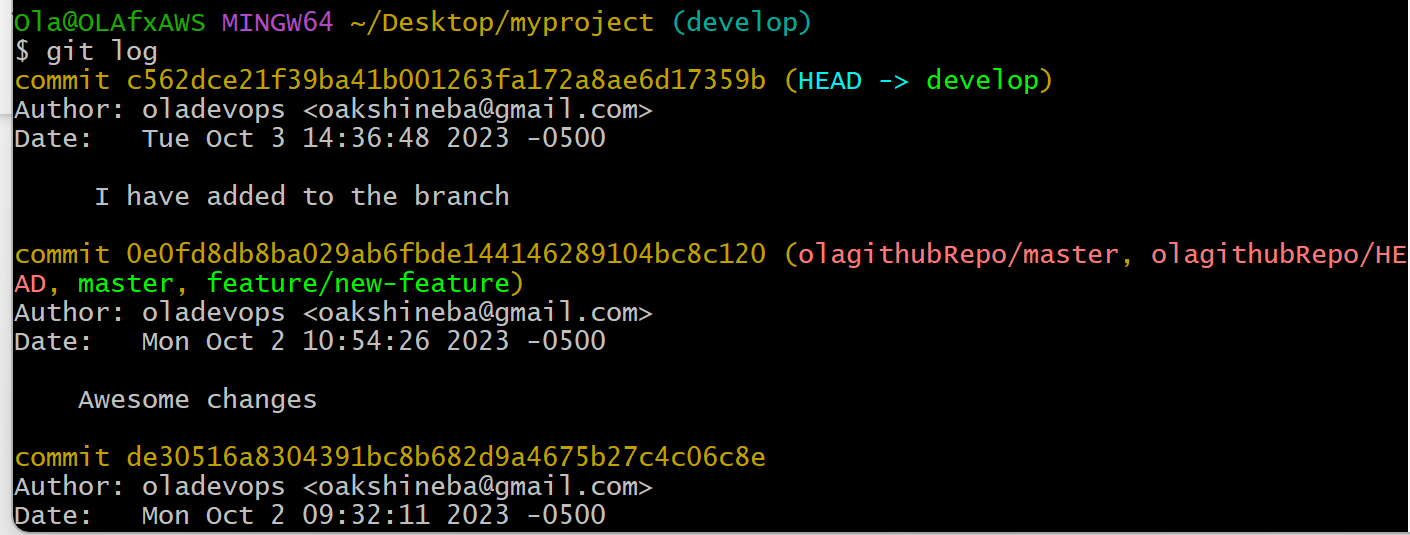
git checkout -b feature/new-feature ---------- this will create a new branch reference off the develop branch(previous), then create a new branch based on develop branch. If on master branch, it will reference off master branch instead.



Git merge

Here we merged the new feature branch into our develop branch as seen above and ran the log below.

Git log



Task 3.

Process of creating a pull request

1. Find out the project to contribute to

2. Submit a pull request

3. Folk a repo – create a copy of another user’s repo

4. Clone the fork on the working area to start development – this is done by downloading a remote repo to our local machine

5.Use the code editor to open up the project

1. Use git branch command to create a new branch for the changes to make
2. Use git checkout to move to the branch
3. After making changes, use git add to stage the changes
4. And git commit to commit the codes to the repository
5. Next is to get the file/code back to Github – use git push command (push command upload local changes to the remote repo
6. We can use git push origin – origin is remote repository on Github follow by the branch name currently working on. Our changes can be seen on Github
7. Ensure that contribution guidelines for the owner is followed strictly

