|  |  |  |
| --- | --- | --- |
| SCHOOL OF INFORMATION AND TECHNOLOGY | | |
| NAME: Barroga, Shaira B. | DATE PERFORMED: Nov 21, 2024 |  |
| Section: IDC2 | DATE SUBMITTED: |

# SYSADM1 – Git Basics

Answer the following research questions about Git, GitLab desktop and GitHub.

1. **What is Git, and why is it important in software development?**

Git is an open-source distributed version control system (DVCS) that manages source code and tracks changes. It is important in software development because it optimizes operations and makes development workflows efficient. Also, its flexibility allows teams to collaborate efficiently.

1. **How does Git track changes in a project?**

Git tracks changes by using snapshots. It creates subfolders that contain metadata for version control and monitor commit histories. Any changes made are saved in the repository's history, and Git shows which files have been modified.

1. **What is the difference between a local repository and a remote repository in Git?**

|  |  |
| --- | --- |
| **Local Repository** | **Remote Repository** |
| * Located on the user's personal computer and can be accessed or modified offline * Used for version control tasks such as committing, viewing logs, or modification * Most git commands can be used within the repository (e.g. git add, git commit, git status) | * Located on a server that can be accessed or modified online * Used to share data over the server for collaboration with other users online * Only some commands can be used to interact with the repository (e.g. git push, git pull, git clone) |

1. **What are the basic Git commands?**

|  |  |
| --- | --- |
| **Git Command** | **Description** |
| Git init | Creates a new repository |
| Git clone | Creates a copy of an existing repository |
| Git commit | Saves changes to the repository’s history |
| Git status | Displays the status of the working directory (modified, untracked, etc.) |
| Git branch | Lists, deletes, or creates branches in the repository |

1. **How do you check the status of a Git repository?**

To check the status of the Git repository, use the “git status” command. This command displays whether the files in the repository are being tracked, as well as modifications made.

1. **What is the purpose of branches in Git, and how do you create and switch between them?**

Branches in Git allow the user to modify files without affecting other files in the repository. This also helps to secure other branches in the repository when multiple users are modifying different branches; if one branch has errors, the files in other branches are not affected, preventing conflicts and disruptions to work being done in the repository.

To create a new branch in the repository, use the “git branch <branch-name>” command. As for switching between branches, use the “git checkout <branch-name> or git switch <branch-name>” command.

1. **What are GitLab Desktop and GitHub, and how are they different from Git?**

|  |  |  |
| --- | --- | --- |
| **Git** | **GitLab** | **GitHub** |
| * Locally installed command-line tool that tracks changes in the code * Allows users to make modifications offline | * Open-source DevOps platform that offers repository hosting, project management, and built-in CI/CD features * More secure as it has security features such as Static Application Security Testing (SAST: scans code and detects security flaws) and Dynamic Application Security Testing (DAST: analyzes responses to malicious inputs) | * Open-source hosting platform that offers private and public repositories and documentation * Focused on code hosting and collaboration * Enables sharing of files online for collaboration * Has limited CI/CD features * Less secure as it only has basic security features |

1. **How do you connect a local Git repository to a GitLab or GitHub repository?**

You can connect a local Git to GitHub or GitLab with the use of these commands:  
1st step:

git remote add origin <repository-URL>

* + This command links the local Git repository to the remote repository in GitHub or GitLab.

2nd step:

git push -u origin main

* + This pushes the local repository to the remote one in GitHub or GitLab.

1. **What are the steps to collaborate with others using GitLab or GitHub?**

1st step: Clone the repository using the “git clone <repository-URL>” command

2nd step: Create a branch to avoid conflicts in the main branch using the “git checkout -b <branch-name>” command

3rd step: After making modifications use the following commands to push updates to the remote repository

* + git add
  + git commit -m: -m is not required, but it is used to create comments or descriptions such as details about the work you did on the repository
  + git push origin <branch-name>

4th step: To merge the branches created by collaborators, use the following commands

* + git checkout main: enters the main branch
  + git pull origin main: gets the branches from the remote repository and puts them into the local repository
  + git merge <branch-name>: merges the branches into the main branch
  + git push origin main: puts the updated local repository in the remote repository which can again be modified by collaborators

1. **How do you resolve merge conflicts in Git?**

To resolve merge conflicts, you first have to identify what is causing the conflicts with the use of commands such as:

* + Git log –merge: shows which commits are causing the conflict
  + Git diff: shows the differences between the conflicts in the files

After identifying the cause of the conflicts, you have to manually fix it by opening the file and choosing which change to keep, adjusting the code that is causing the conflict, or redoing the conflicted area. Using the following codes could also help resolve the conflict issue

* + Git checkout --<file-name>: reverts the file to before the conflict occurred
  + Git reset --mixed: undo the merge and reset the changes to before the conflict occurred
  + Git merge --abort: returns the repository to before the files were merged

1. **What is a pull request, and why is it used in GitHub?**

A pull request is a proposal to merge changes from one branch into another. It allows team members to review and approve changes before they are merged into the main branch. This ensures that only approved modifications are merged, which maintains code quality. Also, it provides details of the changes, allowing team members to assess whether the modifications are necessary and appropriate, preventing disruptions and disagreements.

1. **What are some best practices for writing commit messages?**

It is best to avoid mixing unrelated changes in one commit to make modifications easier to understand and make troubleshooting easier. Use the -m flag whenever possible to provide a detailed commit message about the changes made, helping collaborators understand the purpose of those changes. Test and verify changes to ensure the commands are working, preventing disruption to the workflow of other collaborators. Use automated testing tools such as CI/CD to ensure that code is functioning properly, enabling quick action and improving efficiency.

References:

* Atlassian. (n.d.). *What is Git | Atlassian Git Tutorial*. https://www.atlassian.com/git/tutorials/what-is-git
* *About Git - GitHub Docs*. (n.d.). GitHub Docs. https://docs.github.com/en/get-started/using-git/about-git
* *Introduction to remote repositories*. (n.d.). Learn Git Ebook (CLI Edition). https://www.git-tower.com/learn/git/ebook/en/command-line/remote-repositories/introduction
* Atlassian. (n.d.). *Basic Git Commands | Atlassian Git Tutorial*. https://www.atlassian.com/git/glossary#terminology
* Holcombe, J. (2022, January 28). *How to inspect a GIT repository on GitHub*. GreenGeeks. https://www.greengeeks.com/tutorials/inspect-a-git-repository/
* Atlassian. (n.d.). *Git Branch | Atlassian Git Tutorial*. https://www.atlassian.com/git/tutorials/using-branches
* Darshil-Kansara. (2024, August 6). *GitHub vs GitLab: Which is Best to Choose in 2024?* Radixweb. https://radixweb.com/blog/github-vs-gitlab
* McKenzie, C. (2024, March 1). How to git push an existing project to GitLab. *TheServerSide*. https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-to-add-and-push-an-existing-project-to-GitLab
* *Inviting collaborators to a personal repository - GitHub Docs*. (n.d.). GitHub Docs. https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-personal-account-on-github/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository
* Afreen, S. (2024, July 15). *How to resolve merge conflicts in Git?* Simplilearn.com. https://www.simplilearn.com/tutorials/git-tutorial/merge-conflicts-in-git
* *About pull requests - GitHub Docs*. (n.d.). GitHub Docs. https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests
* Manzoor, S. (2024, July 19). *Good Commit ✔ VS. Bad Commit ❌: Best Practices for Git*. DEV Community. https://dev.to/sheraz4194/good-commit-vs-bad-commit-best-practices-for-git-1plc