|  |  |  |
| --- | --- | --- |
| SCHOOL OF INFORMATION AND TECHNOLOGY | | |
| NAME: Paul Raynee D. Andres | DATE PERFORMED: 11/20/2024 |  |
| Section: IDC1 | DATE SUBMITTED: 11/20/2024 |

# **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 a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.

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

* As you edit files, Git sees them as modified, because you've changed them since your last commit. As you work, you selectively stage these modified files and then commit all those staged changes, and the cycle repeats.

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

* Local Repository: A local Git repository is stored on your own computer. It includes your project files, history of commits, and branches. You can perform all Git operations like commit, branch, and merge locally without needing an internet connection.
* Remote Repository: A remote repository is hosted on a server (like GitHub, GitLab, or Bitbucket) and is typically used to share your work with others. It stores a copy of your project's history and files, allowing collaboration. You push changes from your local repo to the remote repo and pull changes from the remote repo to your local one.

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

* git init: Initializes a new Git repository.
* git clone [repository-url]: Clones an existing repository from a remote server to your local machine.
* git status: Shows the current state of your repository, including changes that are staged, unstaged, or untracked.
* git add [file]: Stages changes for commit.
* git commit -m "[message]": Commits staged changes with a message.
* git push: Pushes your commits to the remote repository.
* git pull: Fetches changes from the remote repository and merges them into your local branch.
* git branch: Lists all branches in your repository.
* git checkout [branch-name]: Switches to a different branch.
* git merge [branch-name]: Merges changes from one branch into the current branch.

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

* You can inspect a Git repository by using the git status command. This command allows you to see which changes have been staged, which haven't, and which files aren't being tracked by Git.

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

* Git branches provide a structured way to manage code changes, enabling a smooth integration of new features, bug fixes, and experimental ideas without disrupting the main codebase. Traditionally, we used the git checkout command to move between branches.

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

* Both GitHub and GitLab operate as a cloud service, while Git operates locally on the host device. Git also doesn't come with a built-in GUI, but you can download a third-party GUI client. Let us further explore the differences between these services.

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

* Create a GitLab repository for the current project.
* Copy the GitLab URL for the new repository to your clipboard.
* Issue the git init command in the base folder of your development project.
* Add all of your project's files to the Git index and then perform a commit.

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

* To collaborate on GitLab or GitHub, first, clone the repository, create a branch for your work, make changes, and push the branch. Then, create a pull request (PR) to propose the changes. Team members can review, discuss, and approve or suggest edits before merging.

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

* To resolve merge conflicts in Git, first, pull the latest changes from the main branch. Git will flag conflicts. Manually edit conflicting files, keeping the changes you need, then mark them as resolved. Commit the merge and push.

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

* A pull request (PR) is a request to merge changes from one branch into another (typically from a feature branch to the main branch). It allows team members to review, discuss, and approve changes before they are merged into the main codebase.

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

* Best practices for writing commit messages include: keeping them concise, using the imperative mood (e.g., "Fix bug" instead of "Fixed bug"), providing context for why changes were made, and following a consistent format (e.g., title and body).