Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Introduction to Git

- **Git** is a **Distributed Version Control System (DVCS)** used to manage source code.
- Initially, developers worked in **local folders**, shared code manually, leading to **dependency issues**.
- **Centralized Version Control Systems (CVCS)** (e.g., SVN) introduced a central repository, but dependency on the central server created risks.
- **Distributed Version Control Systems (DVCS)** like Git solved this by allowing each developer to maintain their **own local repository** while syncing with a central remote repository.

Key Advantages of Git

1. **Open Source** – Free to use and allows contributions.
2. **Super Fast** – Efficient code management and version tracking.
3. **Scalable** – Works for small teams to enterprise-level projects.
4. **Easy to Learn & Use** – Only a few commands are needed to start.
5. **Supports Multiple Users** – Enables team collaboration without direct dependencies.
6. **Works Across Platforms** – Available for Windows, macOS, and Linux.
7. **Efficient Code Tracking** – Maintains history, tracks changes, and allows easy rollbacks.

Git Complete Crash Course - Detailed Notes

N M VIJAY
www.linkedin.com/in/nmvijay

Git Installation & Setup

1. **Checking Git Installation**
2. git --version
    - Ensures Git is installed.
3. **Initializing a Repository**
4. git init
    - Converts a directory into a Git repository.
5. **Checking Repository Status**
6. git status
    - Displays the state of the working directory.

Git Workflow

Git has three key areas:

1. **Working Directory** – Where files are modified.
2. **Staging Area** – Where files are added before committing.
3. **Local Repository** – Where committed changes are stored.

Basic Workflow Steps

1. Modify files in the **working directory**.
2. Add files to the **staging area** using:
3. git add filename # Adds a specific file
4. git add . # Adds all changes
5. Commit changes to the **local repository** using:
6. git commit -m "Commit message"

Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Git Branching & Merging

- **Branches** allow developers to work on different features independently.
- **Master/Main Branch** is the default branch.
- **Creating a New Branch**
- git branch branch_name
- **Switching to a Branch**
- git checkout branch_name
- **Creating & Switching to a New Branch**
- git checkout -b branch_name
- **Merging a Branch into Main**
- git checkout main
- git merge branch_name
- **Deleting a Branch**
- git branch -d branch_name

Handling Merge Conflicts

- Occurs when different branches modify the same file.
- Git marks the conflict:
- <<<<<<< HEAD
- Your changes
- =======
- Other changes
- >>>>>>> branch_name
- Manually edit the file, then commit the resolution:
- git add resolved_file
- git commit -m "Resolved merge conflict"

Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Working with Remote Repositories

- **Cloning a Repository**
- git clone repository_url
- **Adding a Remote Repository**
- git remote add origin repository_url
- **Pushing to Remote Repository**
- git push -u origin main
- **Pulling from Remote Repository**
- git pull origin main

Stashing & Resetting

- **Stashing Changes**
- git stash
    - Saves uncommitted changes for later.
- **Applying Stashed Changes**
- git stash apply
- **Resetting Commits**
- git reset --hard commit_hash
    - Moves HEAD and discards commits after the specified commit.

Git Logs & Reverting

- **Viewing Commit History**
- git log
- **Reverting to a Previous Commit**
- git revert commit_hash
    - Creates a new commit undoing the changes of the specified commit.

Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Pull Requests & Forking

- **Pull Requests (PRs)**
  - Used for code review before merging changes into the main branch.
  - Typically used in platforms like GitHub.
- **Forking a Repository**
  - Copies another user's repository into your account.

Common Git Commands Reference

Command	Description
git init	Initialize a new Git repository
git clone URL	Clone a remote repository
git status	Show current branch and file status
git add .	Add all changes to the staging area
git commit -m "message"	Commit changes with a message
git branch branch_name	Create a new branch
git checkout branch_name	Switch to another branch
git merge branch_name	Merge a branch into the current branch
git push origin main	Push commits to remote repository
git pull origin main	Fetch and merge changes from remote
git log	View commit history
git revert commit_hash	Undo changes without deleting history
git reset --hard commit_hash	Reset to a previous commit and discard changes
git stash	Temporarily save changes

Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Command	Description
git stash apply	Restore stashed changes

Interview Questions & Answers

Basic Git Questions

1. **What is Git?**
    - Git is a **Distributed Version Control System (DVCS)** used to track changes in source code during software development.
2. **What is the difference between Git and GitHub?**
    - Git is a version control system, while GitHub is a remote hosting service for Git repositories.
3. **What is a commit in Git?**
    - A commit represents a snapshot of changes in the repository.
4. **What is the difference between git pull and git fetch?**
    - git pull fetches changes from a remote repository and merges them.
    - git fetch only downloads changes without merging.

Branching & Merging Questions

5. **How does Git branching work?**
    - Branching allows multiple developers to work on different features without affecting the main codebase.
6. **What happens if a merge conflict occurs?**
    - Git marks conflicts in the file, and developers must manually resolve them.
7. **What is the difference between git merge and git rebase?**
    - git merge combines branches, keeping their history.
    - git rebase applies commits from one branch on top of another.

Git Complete Crash Course - Detailed Notes

N M VIJAY

www.linkedin.com/in/nmvijay

Advanced Git Questions

8. **What is the difference between git reset and git revert?**
    - git reset removes commits from history.
    - git revert creates a new commit that undoes previous changes.
9. **What is git stash used for?**
    - It temporarily saves changes without committing them.
10. **How do you undo a commit that has already been pushed?**
    - Use git revert commit_hash to create a new commit undoing changes.
    - If force removal is needed, use git reset --hard commit_hash and git push --force.

Conclusion

This guide covers essential Git concepts, commands, and interview preparation tips. Practice by setting up local repositories, creating branches, and pushing to remote repositories. Git is a powerful tool for collaboration, ensuring code integrity and seamless version management.