Table of contents

- Table of contents
  - Installation of Git
  - Git Local Operations
    - Create Local Repository, adding files and commit changes
    - Initialize a Git Repository
    - Working with Git Local Repo
  - Github Account Remote Repository.
  - Reference information
    - Changing a git remote URL
  - Reference:

Installation of Git

- Windows
  - Downloading Git on Windows https://git-scm.com/download/win
- Debian - Ubuntu
  - Installing on Debian-based distribution, such as Ubuntu `sudo apt install git-all`
- Install git on respective OS from https://git-scm.com
- Download Github for Desktop from here

> Execute all below commands in `Git Bash` shell

Git Local Operations

Create Local Repository, adding files and commit changes

Initialize a Git Repository

- Run `git --version` to check git version.
- Start a new repository. Check Your git Settings
- Create a new Project Folder and initialize git inside it.

```
git --version
mkdir git-practical
cd git-practical
git init
ls -altR  # List all the files recursively
git config --list
```

- `.git` directory -> We will not modify anything inside this path.

- The `git init` command creates an empty Git repository - basically a `.git` directory with subdirectories for `objects, refs/heads, refs/tags, and template files`.
- Git configuration can be `global` and `local`.
- If you have multiple git servers, you need to config your git as per git directory bases. Or you can use global config.
- For local git repo config, use:

```
git config --local user.name "TestUser"
git config --local user.email "testuser@example.com"
```

- This will result in [user] section added to .git/config file:

```
cat .git/config

[user]
name = Yourname
email = name@example.com
```

- You can use global config also if you only have one git server.

```
git config --global user.email "testuser@example.com"
git config --global user.name "TestUser"
```

- Then the [user] section will be present at `~/.gitconfig`, with the same content as in the `.git/config` file.
- To get the specific config value that is currently set

```
git config user.name
git config user.email
```

Working with Git Local Repo

- Go to the directory where you want to initialize as git repo
- List all the files in the current directory

```
ls -al
echo 'This is Repo Created for Devops Demo' >> file.txt
git status
```

- To track the file, add this file into `staging area`

```
git add file.txt
git status
```

- To commit a change, there should be a commit message provided.

```
git commit -m "changes made in the particular file"
git log
```

- Make some more changes in the file:

```
echo 'This is content written after 1st Commit' >> file.txt
git status
# Changes not staged for commit => File is changed in working area
git add file.txt
# Changes to be committed: => File is in Staging Area and can be committed

git commit -m "added changes to same file for new commit"
git show <COMMIT_ID>
```

- If you edit a file that is already staged, it will appear in `"Changes to be committed:"` and `"Changes not staged for commit:"`
- Viewing Your Staged and Unstaged Changes
- This command compares your staged changes to your last commit:

```
git diff --staged
```

- make some changes in the file that is already staged, below command will show the differences

```
git diff
```

- Make a commit
- Viewing the Commit History and view commit details in one line

```
git log
git log --oneline
```

- To view the only last two entries

```
git log -p -2
```

- More common options for `git log`

```
git log --stat
git log --pretty=oneline
```

- To change the commit message of an existing commit

```
echo "this is testing file" >> newfile.txt
git add newfile.txt
git commit -m '1st commit'
git log -p -2
git commit --amend -m "an updated commit message"
git log -p -2
```

- Unstaging a Staged File

```
echo "adding some content to unstage changes" >> newfile.txt
git add *
git reset newfile.txt
```

Github Account Remote Repository.

- Sign Up into www.github.com and verify email id.
- Connection to your GitHub Account using SSH
- Generating an SSH Key, Use the github sign-in email address in the below command.

```
ssh-keygen -t rsa -C "<GITHUB_EMAIL_ID>@gmail.com"
```

- Above command will create a Public and Private Key Pair.
- Add the Public Key file content into your Github Account Settings under: `Settings` > `SSH and GPG keys.` > `New SSH key` > `Paste the Public Key Content` > `Save`
- Verify SSH authentication

```
ssh -T git@github.com
OR
```

```
ssh -i <PRIVATE_KEY_PATH> -T git@github.com
# Hi GITHUB_USERNAME! You've successfully authenticated, but GitHub does not
provide shell access.
```

- You should get above similar message if connection to github account is successful using SSH
- Create a New empty repository in Github using browser.
- Current Local Repository already has files and changes in commits
- To Push changes from local Repo to Remote repo for master branch using ssh, add a SSH remote origin URL for Local Repository.

> To change the remote URL click here

- Using git remote add command allows us to associate a remote repository. Normally, you want to paste in the full URL for the remote repository given to you by your Git host (GitHub). By convention, the first or primary remote repository is named origin.

```
git remote add origin git@github.com:<GITHUB_USERNAME>/git-practical.git
```

- The git remote command lists the names of all the remote repositories and the -v parameter (verbose) will display the full URL of the remote repository for each remote name listed.

```
git remote -v
```

- First time push command use below `-u` parameter

```
git push -u origin master
```

- To have multiple commits created, add or modify some files -> commit and push it to GitHub.

```
git add <FILENAME>
git status
git commit -m "Message for the commit"
```

- If there are any changes made in the Remote Repository, to have those changes present in Local Repository, use below command:

```
git pull origin master
```

- Push changes in GitHub

```
git push origin master
```

- To get or display the content of the file as per particular commit.

```
git show e4b71efa7f76c0fc0875e0562d5fb6d7dadbff9c:newfile.txt
```

Reference information

Changing a git remote URL

1. Switching remote URLs from SSH to HTTPS

- List your existing remotes in order to get the name of the remote you want to change.

```
git remote -v
origin  git@hostname:USERNAME/REPOSITORY.git (fetch)
origin  git@hostname:USERNAME/REPOSITORY.git (push)
```

- Change your remote's URL from SSH to HTTPS with the git remote set-url command.

```
git remote set-url origin https://hostname/USERNAME/REPOSITORY.git
```

- Use below command to clone the Repository with SSH URL.

```
git clone git@github.com:<GITHUB_USERNAME>/<REPO_NAME>.git
git remote -v
```

- The next time you `git pull, or git push` to the remote repository, you'll be asked for your GitHub username and password.

2. Switching remote URLs from HTTPS to SSH

- Change your remote's URL from HTTPS to SSH with the git remote set-url command.

```
git remote set-url origin git@github.com:USERNAME/REPOSITORY.git
```

- Send Changes to Remote

```
git push -u remote-name branch-name

git push remote-name branch-name
```

- The git push sends all your local changes (commits) on branch `branch-name` to the remote named `remote-name`.
- The `-u` parameter is needed the first time you push a branch to the remote.
- Receive Changes from Remote

```
git pull remote-name branch-name
```

- The git pull receives all your remote changes (commits) from the remote named remote-name and on branch branch-name.

> Getting help from git

```
git help <verb>
git help config
git add -h
```

Reference:

- Download Github for Desktop from https://desktop.github.com/