**Password: ghp\_sZbyUCN7tWkAcf5VGwUVoV6WYEZ7kT1no0Dc**

**Username: Gangasagarhl**

**Short:**

1. Install git in laptop
2. Go to a folder where you want to clone the github repository with the local repository.
3. Open terminal from that folder, then type.
4. **git --version** : check whether the git is installed or not, else please install.
5. **git clone HTTP COPIED CODE/SSH CODE :** This is used clone the remote folder(gets automatically downloaded) as github repository, your folder is github repository.
6. **git status:** Go to a folder which has got cloned, then type the command and enter, tells untracked files present
7. **git add “name\_of\_the\_file”:** The file will not be committed, instead tells git to keep in track.
   1. **git add .** : (new or modified content)The dot is a shorthand notation that represents the current directory and all its contents, recursively. It includes:
      1. All modified files
      2. All newly created (untracked) files
      3. All deleted files
      4. Files in subdirectories of the current directory
8. **git commit [-m “message”]:** even it can be normally commited, but adding message will push msg with the file.
9. But every time you commit needs email and username:
   1. **git config --global user.email [“gangasagar.hl1112@gmail.com”](mailto:\“gangasagar.hl1112@gmail.com\”)**
   2. **git config --global user.name “Gangasagarhl”**
10. **git pull:** It is used to get the contents or the files in repository(server) to the local folder/ local repository.
11. The **git log [OPTIONS]** command is used to display a log of commits in a Git repository. It provides a detailed history of the changes made, including commit messages, author information, dates, and commit hashes.
    1. Options:
       1. -n 5: last 5 commits
       2. **--oneline** : You can condense the log output to one line.
       3. **--graph --oneline** : graphical representation
       4. **-- filename.txt:** changes in some specific file
       5. **--author="John Doe" : specific author**
12. s

**Actual**

1. Distributed version control tool - collaborate with others, so that they too can help, access and managing.
2. Branching: Isolate yourself from others, so if you have done some great things, then you will be allowed to merge to main branch. If done some mistake , you can delete the branch too.
3. Github: it is nothing but a git running on a server somewhere. Initially git was only developed, but how would they share the files with somebody so that they can have chitchat regarding the file/project they are developing.
4. Sold to microsoft for 9 billion dollars.
5. git doenot have security buitin.
6. They provide authentication services, private no one has the access to enter into private.
7. Workflow:
   1. In Git, a pull request is a mechanism for initiating a discussion around a feature branch. It allows team members to review the code, provide feedback, and suggest changes before merging the code into the main branch.

To create a pull request, follow these general steps:

* 1. Fork the repository (if necessary) and create a feature branch based on the main branch.
  2. Make your changes and commit them to your feature branch.
  3. Push your feature branch to the remote repository.
  4. Navigate to the repository on the hosting service (e.g., GitHub, Bitbucket) and create a pull request, specifying the source and target branches.
  5. Describe the changes and provide any necessary context for the reviewers.
  6. Team members review the code, leave comments, and propose modifications if needed.
  7. After the review process, the code can be merged into the main branch.
  8. The specific steps and terminology may vary slightly depending on the Git hosting service, but the basic concept remains the same. Let me know if you need more details on any of these steps!

1. Forking: Forking a repository in the context of Git refers to creating a personal copy of someone else's project. This is commonly done on hosting platforms like GitHub, Bitbucket, or GitLab. When you fork a repository, you are essentially making a copy of the entire repository, including all of its files, commit history, and branches, under your own account.

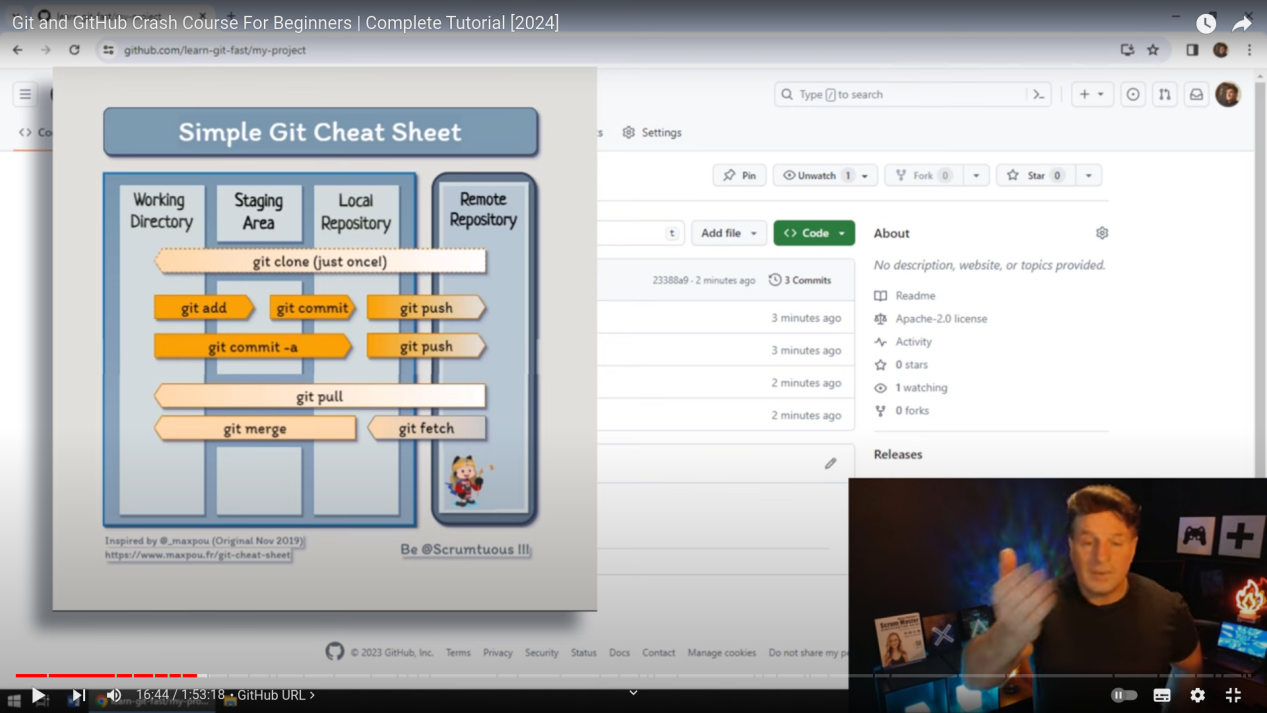
Forking a repository is often used as a way to propose changes to someone else's project. You can make changes to the code in your forked repository and then open a pull request to propose those changes back to the original repository. This is a common workflow in open source development, as it allows contributors to make changes without directly affecting the original project until those changes have been reviewed and accepted.

Forking a repository also allows you to freely experiment and make changes to a project without affecting the original codebase. It provides a level of isolation, where you can create new branches, add commits, and modify the code without affecting the original project.

Once you have forked a repository, you can work with your fork just like any other Git repository, including pushing changes, creating branches, and merging code.

1. Follows
2. Allows free hosting

Git Repository:

1. It is nothing but the project you are working on.
2. maintain the history of commits, branches of commits, branches etc, everything.
3. Cloning: The repository where you work is local, but the directory to be uploaded is at serer, the gap is removed with the help of cloning. **<>code, section HTTPS, SSH** etc. HTTPS, SSH etc
4. When you want to address some issues, you do it locally, then add, commit and then push.
5. 
6. Understand the above diagram.
7. s