Collaborative asset management using GitHub

upGrad knowledge hut

LEARNING OBJECTIVES

- Using GitHub for collaborative work
- Portfolio creation in GitHub
- Understand how to create repositories
- Understand how to clone a repository
- Understand forking of a repository
- Understand how to raise a PR (Pull request)
- Understand the concept of conflicts and how to resolve them

Collaborate using GitHub

GitHub for collaborative work

- GitHub is a collaborative platform
- Resources can be shared across multiple teams and locations
- Has powerful version control features
- Multiple stakeholders can access or edit the same piece of code
- GitHub is the home for thousands of Open-Source Software (OSS) projects

Git Credential Manager (GCM)

Git credential manager (GCM) ensures the authentication of the GIT repositories, easy and secure irrespective of where and how the code is stored.

Installation of GCM: [GitHub - Git Credential Manager Installation Link]

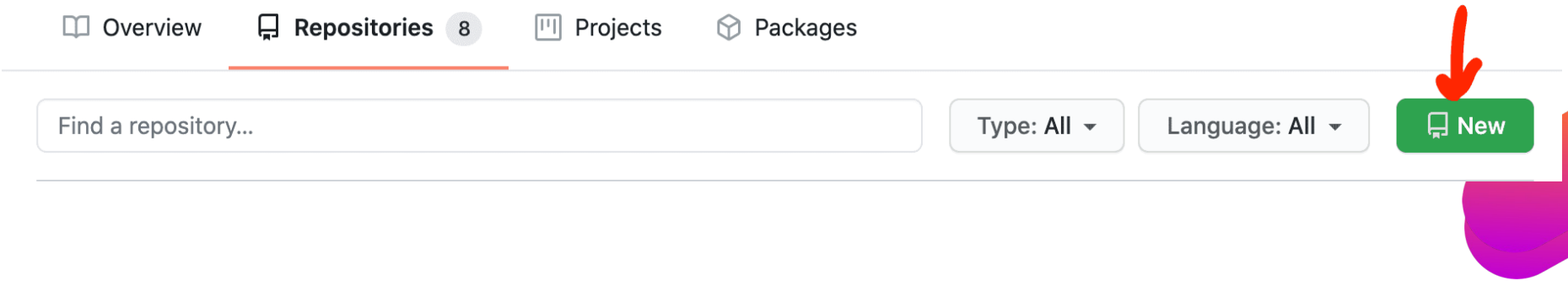
GitHub portfolio

- For developers, creating repositories on GitHub is the best way to create a portfolio to showcase skills.
- Collaborators can be easily added to projects so that they can contribute to the repository.

To add a collaborator, go to the repository in GitHub then go to **Settings → Manage Access, Click on Invite Collaborators,** and type the username or email address to send an invitation.

Create a GitHub repository

- Here are the steps to create a Github repository.
- Open your GitHub profile (github.com/<your_username>)
- Go to 'Repositories' tab
- Click the 'New' button, on the upper-right corner, New Repository Button
- Create a new repository by clicking the 'New' button



Common Terminologies

Frequently used terms by GitHub users

- **Creating a repository** for collaborative work.
- **Master in a repository:** This is the final version that is considered ready to use by the team or outside if the repository is public.
- **Creating a Branch:** Create branches in your project, for trying out new ideas. You can make changes in the branch without affecting the master unless a pull request is accepted.
- **Adding Commits** to keep track of the progress on branch.

Frequently used terms by GitHub users

- **Forking a repository:** It creates a copy for you to work on independently without any changes to the others. If needed send a pull request to owner so that the owner can incorporate the recent amends.
- **Pull requests:** Pull Requests initiates a discussion about your commits or changes made to a code.
- **Issues:** Highlight bugs or issues with codes that need correction. Issues remain open unless they are resolved, which can be filtered out and labeled as a bug.
- **Markdown:** Mark down is a way to style text on the web.

Cloning and Forking

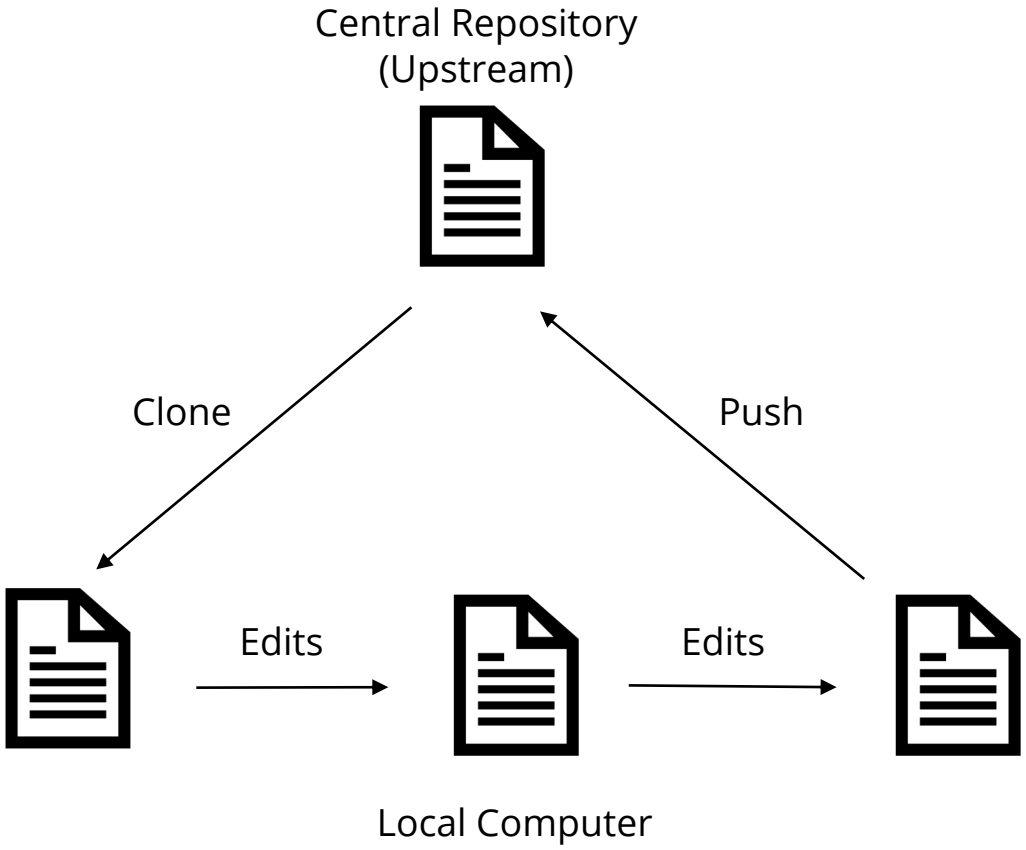
Cloning

- Create an identical local copy of your work based on the online copy that is present in the remote repository.
- When you clone a repository, all the files are downloaded to the local machine, but the remote git repository remains unchanged.
- You clone a repo to work locally and push changes to the online repository.
- Enables multiple developers to work on the same project by contributing to the online repository.

Cloning a repository

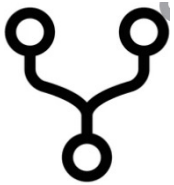
1. The user starts from the upstream repository on GitHub. And they clone the repository into their local machine.
2. After cloning, contributors provide their contribution to the local repository.
3. Iteration, testing, and refinement.
4. Once the changes are done all the modifications can be pushed to the upstream repository.

Cloning


Central Repository
(Upstream)
Clone
Push
Edits
Edits
Local Computer

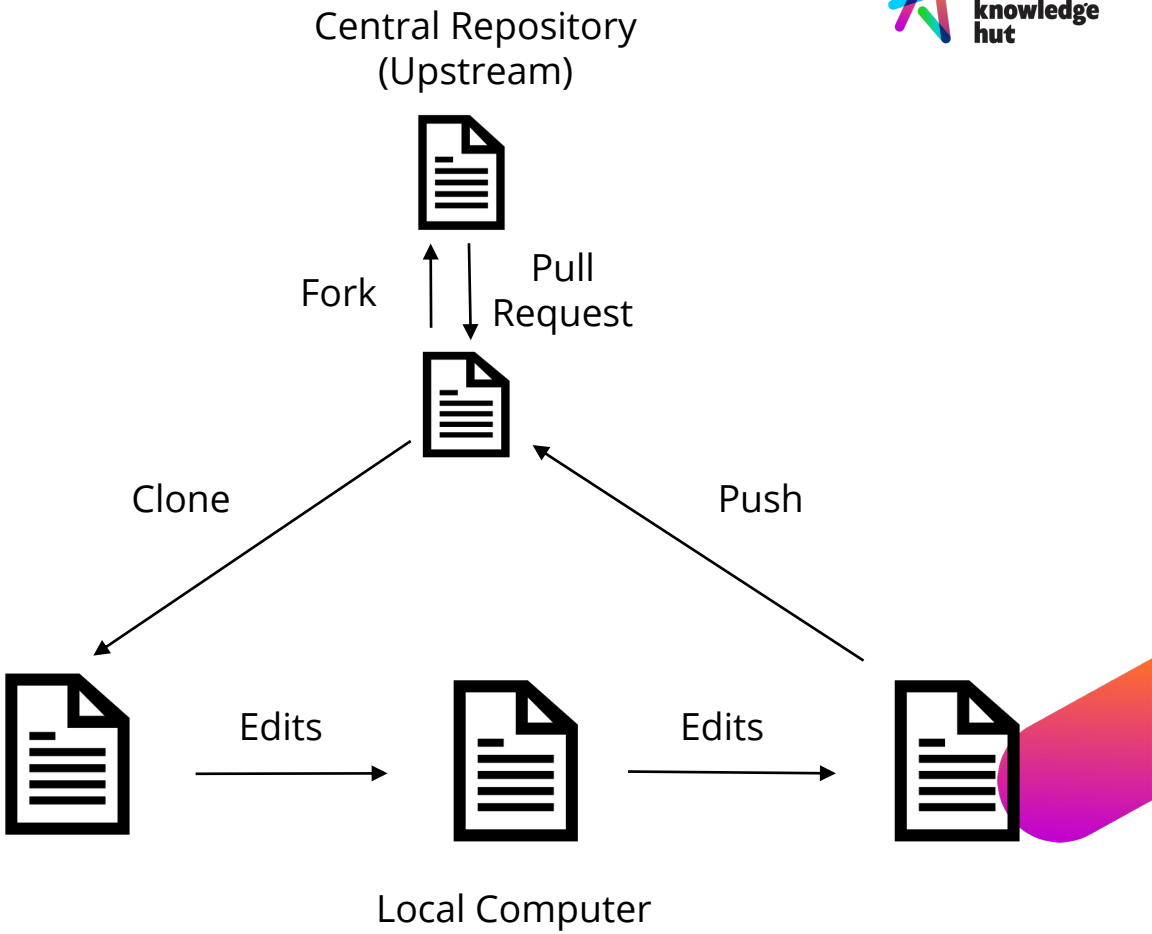
Forking

1. A forked repo is an independent copy of the original repository.
2. You can fork an existing project and build your own independent version with updates and/or modifications.
3. These changes do not affect the original repository or project in any way.
4. If needed, you can raise a PR (Pull Request) to the original repository's author, and the author can choose to merge changes from your forked copy into the original project.
5. This enables the open-source community to contribute to ongoing projects.



Forking

1. Create a fork
2. Clone the fork locally
3. Make modifications
4. Push changes to forked repository
5. Raise a Pull Request (PR) if needed



Pull Request

Pull Request in GitHub

Pull request is given when the changes made in one branch needs to be merged into another. It is a development process which is interactive way of reviewing the code.

IT IS IMPORTANT BECAUSE:

- It helps in improving the quality of code
- Receiving feedback on the code is simplified.
- Can be treated as documentation as it is open and collaborative.

Pull Request in GitHub

To create a pull request within the same repository and merge changes to the main branch follow these steps:

1. On GitHub, navigate to the main page of your repository.
2. In the "Branch" menu, choose the branch that contains your commits.
3. Above the list of files, click Pull request.
4. Use the base branch dropdown menu to select the branch to merge your changes into, then use the compare branch drop-down menu to choose the topic branch you made your changes in.
5. Type a title and description. To create a pull request that is ready for review, click Create Pull Request. After the pull request gets reviewed, it can be merged into the main branch of your repository.
6. In case, you are not sure about submitting the pull request, you could also create a draft pull request. You could use the drop-down and select Create Draft Pull Request, then click Draft Pull Request.

Pull Request in GitHub

To raise a pull request to merge your changes into the upstream repository follow these steps.

1. Navigate to the original repository where you created the fork.
2. The process of creating the pull request is almost the same as earlier. Above the list of files, click on the Pull Request.
3. On the Compare page, click compare across forks.
4. In the "base branch" drop-down menu, select the branch of the upstream repository you'd like to merge changes into.
5. In the "head fork" drop-down menu, select your fork, then use the "compare branch" drop-down menu to select the branch you made your changes in.
6. Type a title and description for your pull request.
7. On user-owned forks, if you do not want to allow anyone with push access to the upstream repository to make changes to your pull request, un-click Allow edits from maintainers.
8. And finally, to create a pull request that is ready for review, click Create Pull Request.

Merge Conflict

Merge conflict in Git

1. Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.
2. Conflicts cannot be avoided; we must know how to resolve them.
3. When we raise pull request to merge our changes, often we might encounter merge conflicts.

Resolve a merge conflict in Git

Easiest way to resolve a merge conflict is:

- Open the conflicted file and make any necessary changes
- After editing the file, use the **git add** command to stage the new merged content
- Create a new commit with **git commit** command
- Git will create a new merge commit to finalize the merge

If the merge conflict is caused by competing line changes, You can resolve it on GitHub using the conflict editor.

For all other types of merge conflicts, you must resolve the conflict locally on the command line.

Resolve a merge conflict in Git

1. To resolve in GitHub, navigate to your repository and then the specific pull request with conflicts. Then click on the “Resolve conflicts” button.

2. Finally, after resolving all the conflicts, you could push the changes back to the remote or raise a pull request.

Summary

GitHub: Initial workflow

Steps to use GitHub:

- Go to the link: https://github.com/ . Sign up to create an account.
- Click on “Start a new project”.
- Enter any repository name and click on “Create Repository”. Give description to your repository.
- Once the central repository is created, you are ready to commit, pull, push and perform all the other operations.
- Create Branches and Perform Operations: commit, pull push, merge etc.

GitHub: Tips

- Create a unique and attractive GitHub profile: GitHub profiles are used as resumes by many developers, so create your GitHub profile readme and showcase your work. Add images, gifs or anything to support your profile.
- Check and manage the notifications: Manage and filter your notifications by repository, organization, or author. You can also manage notifications for different organization to get desired information. Click on the bell icon next to your profile, for getting notifications.
- Choose your display mode: There are various modes like dark, light, dimmed mode as per the new GitHub feature.
- Collaborate: Use the platform to collaborate with developers across the world. You can use the pull requests, project boards and discussions etc. for collaborations.
- Star everything: Whenever you browse any project, remember to star the open-source git hub repos.
- Work on GitHub from different places (Command Line, desktop version, mobile application)

Thank You