Remote Git Repository & Github

Divyansh Rajesh Jain

What is a Remote Git Repository?

- Hosted Git Repository

- Allows people to collaborate on one repo

- Types of Remote Repositories

  - Public (Github, Gitlab, etc.)

  - Private (Bitbucket, self-hosted, etc)

Local vs Remote Repository

- Local → Git Repository on your own computer

- Remote → Git Repository stored on the server

- When we are using a Github repository, are we using

  a local repository or remote repository?

Review of Git Internals

- Objects → "Database" of Git Repository

  - Commit

  - Tree

  - Blob

- References

  - Branches → Symbolic link to commit

  - Remotes → References to remote repository

UC DAVIS

Connection to Remotes

- HTTPS → https://github.com/user/repo.git

  - Uses the HTTPS protocol to connect to remote

- SSH → git@github.com:user/repo.git

  - Uses SSH protocol to connect to remote

UC**DAVIS**

Operations on Remotes

- Push → Uploads local changes to the remote

- Fetch → Gets changes from the remote locally

- Pull → Fetches and merges to local repo

- Clone → Bring the repo from remote to local

Git Push Internals

- Changes are committed locally → Creates new objects

- Locally created Objects are pushed to the remote. Transferred to the remote which now stores new objects

Git Pull Internals

- Two Step Operation

  - Fetch → Brings the Objects from remote to the local

  - Merge → Merges newly fetched objects with existing

    objects in repo

- What are common problems with git pull?

Local vs Remote Branches

- Local Branches → Exist only locally

  ○ Symbolic Link to a Commit Object

- Remote Branches → Exist on the Remote Repository

  ○ Symbolic Link to a Commit Object

  ○ But can be in a different state locally vs remote

Git Remotes Internals

- .git/refs/remotes → Stores the remote branches

  ○ /<remote name>/<branch name> → Ref object that

    corresponds to the remote branch

- These remote refs are used in push and pull to

  resolve differences between remote and local repo

Git Remote Repository Commands

- git remote add <name> <url> → Connect local repo

  to remote repository

- git remote -v → View remotes

- git fetch → Get changes from remote but don't

  merge

Git Remote Repository Commands

- git pull [--rebase] → Retrieve changes from remote

  and merge in local repository

- git push → Upstream local repository changes to the

  remote repository

- git clone <url> → Initialize remote repository locally

Introduction to Github

- Popular platform to share Git repositories

- Offers many advanced features

  - Issues

  - Pull Requests

  - Github Actions → CI/CD directly on Github

UC**DAVIS**

Common Github Use Cases

- Forking → Make a copy of repository to your account

    ○ Useful for open-source contributions, and ECS 198F!

- Pull Requests → Propose changes from a different

branch or fork

UCDAVIS

Best Practices for Git(hub)

- Commit after logical conclusion → clear commit

  message

- Use Branches for organization and use one branch to

  "track" production or released software

- Pull changes frequently so to avoid conflicts

Introduction to Open Source

- What is Open Source?

  - Software released with license to let anyone use source

    code

- Benefits

  - Very fast development with community help

  - Transparency through code visibility

UC**DAVIS**

Contributing to Open Source

- Step 1: Forking Repository

- Step 2: Make changes on the forked repository

- Step 3: Create a pull request to propose the change

- Step 4: Maintainers will review your pull request and

  leave comments and/or reject pull request

Thank You!

Next Time: Introduction to Docker and Containerization