git

GitHub

- Amol Gote
- Vikas Mendhe
- Charles Wagner

Git and GitHub Workshop

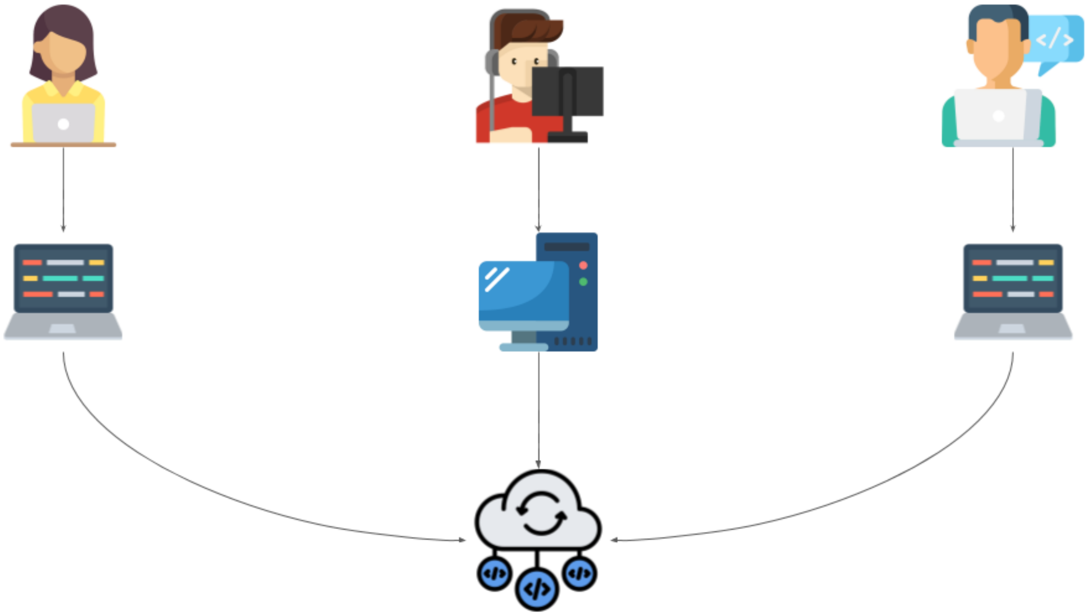
Agenda

1. Version Control
2. Git V/S GitHub
3. Create GitHub Repo
4. Set Up Environment
5. Branches, Pull Request
6. Forks – Open-Source Contributions
7. Git Tools
8. Best Practices
9. Miscellaneous Items
10. Hand On Project

1. Version Control Systems (VCS)

- Introduction
- Why version control is required.
- Different types of version control systems

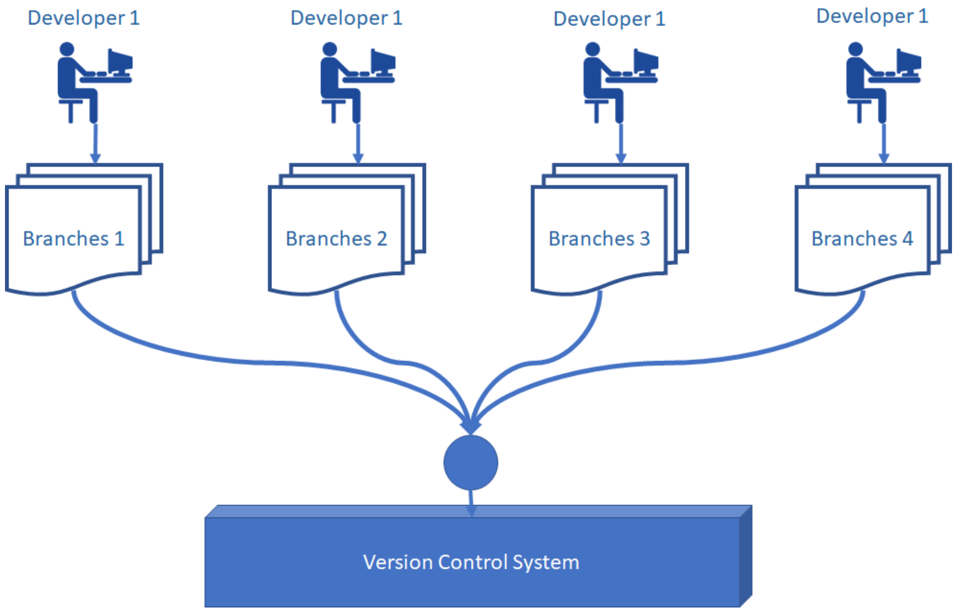
Introduction to Version Control System



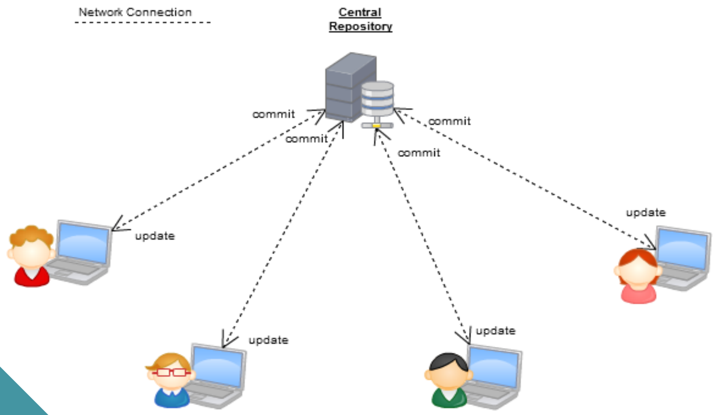
- VCS are tools used to track changes in software development and other collaborative projects. They allow multiple versions of files to be managed and accessed over time.
- They facilitate collaboration among project team members by enabling concurrent work and contributions on different project parts without overwriting each other's changes.

Why Version control is required

- History and Accountability
- Collaboration
- Undo Mistakes
- Compare Changes
- Backup and Restore
- Advance topics
  - Branching and Merging
  - Workflow Management
  - Facilitates Continuous Integration and Delivery (CI/CD)

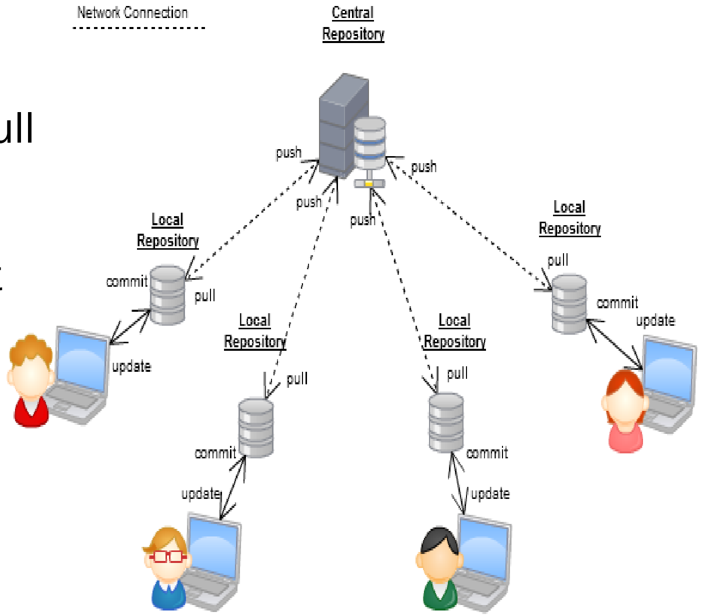


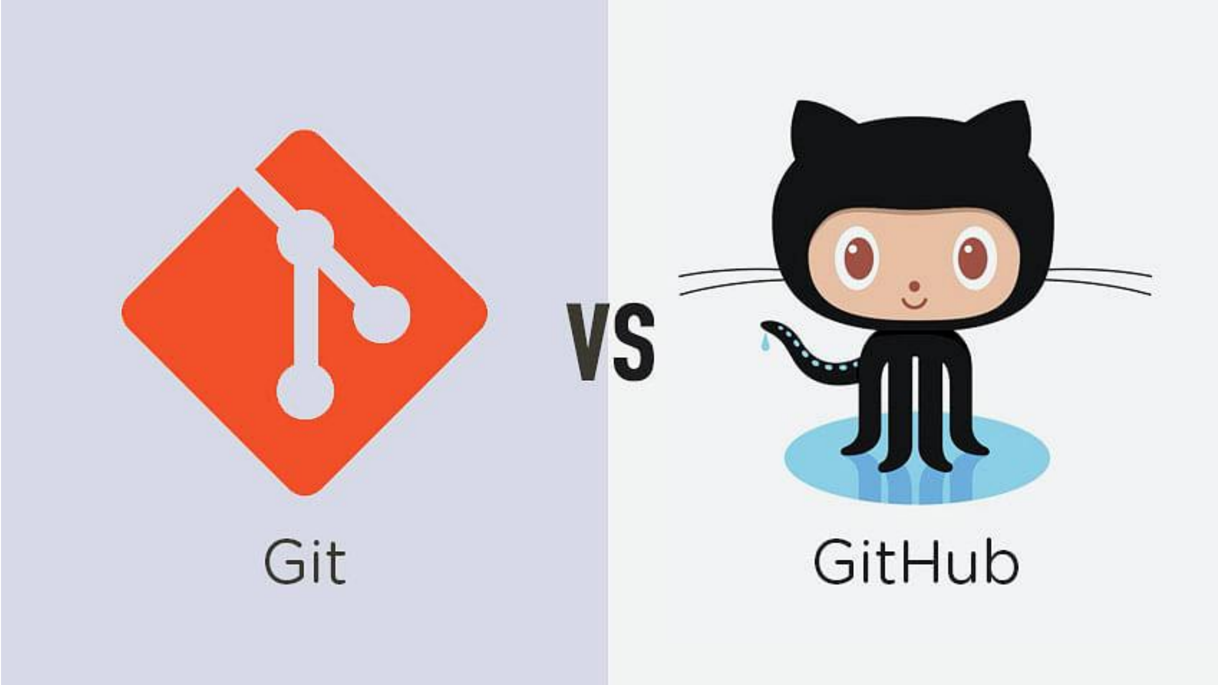
Different types of Version Control Systems



- Centralized VCS (CVCS): All team members work on a single central repository. Popular examples include Subversion (SVN), TFVC (Team Foundation Version Control), and Concurrent Versions System CVS. While it simplifies the collaboration model, it introduces a single point of failure.

- Distributed VCS (DVCS): Every contributor has a full copy of the repository, including its history. Examples include Git and Mercurial. DVCS allows for more flexible workflows and resilience against repository corruption or loss.





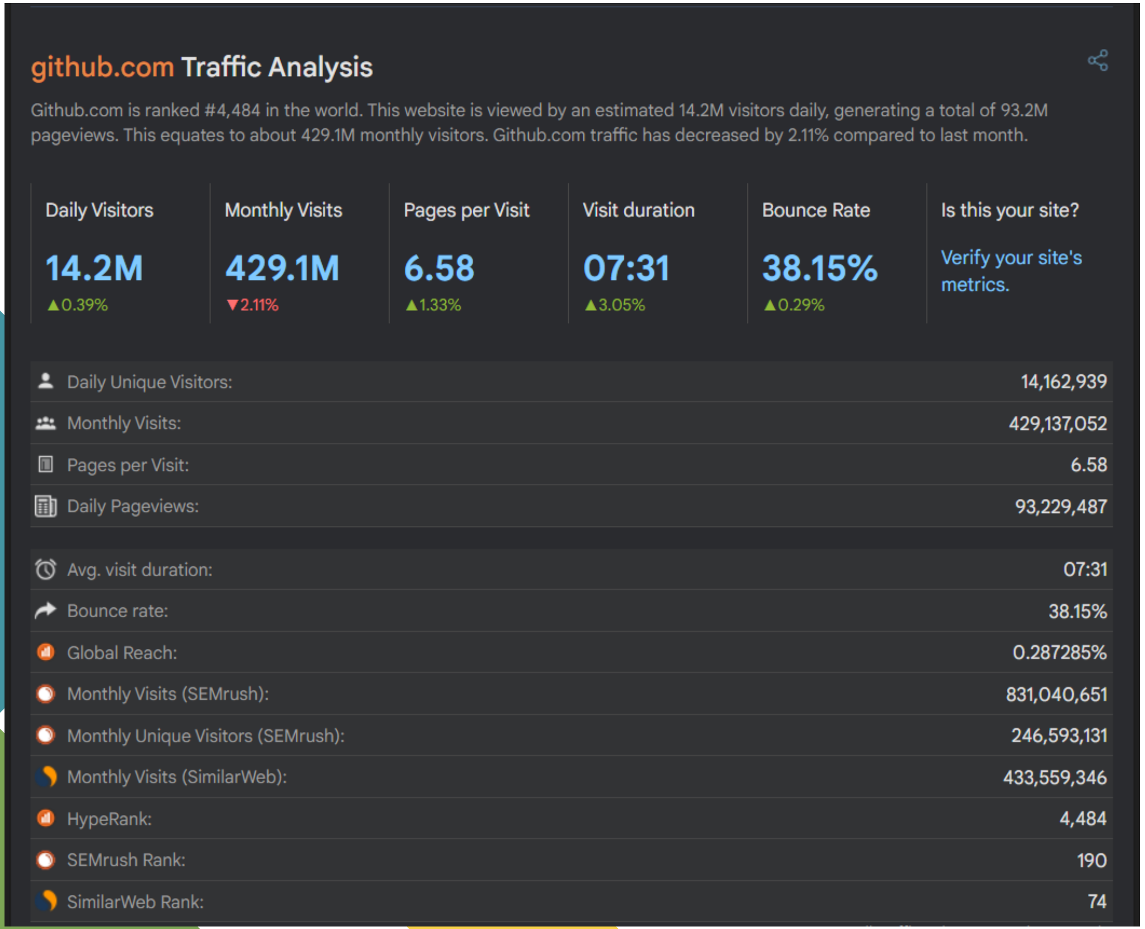
2. Git and GitHub

- Differences
- Advantages of Git and GitHub

Git, GitHub - Differences

- Git is the tool that individuals use to track different versions of their code, manage branches, and handle merging.
- GitHub is the platform that hosts Git repositories online. It adds a layer of collaboration, review, and additional features to Git's functionality.
- *GitHub is a platform for anyone from anywhere to build anything, anything that you think the world needs*
- Complementary Relationship: While Git can be used without GitHub, GitHub requires Git for version control. GitHub enhances Git's capabilities with a collaborative and user-friendly interface.

GitHub - Stats



github.com Traffic Analysis

Github.com is ranked #4,484 in the world. This website is viewed by an estimated 14.2M visitors daily, generating a total of 93.2M pageviews. This equates to about 429.1M monthly visitors. Github.com traffic has decreased by 2.11% compared to last month.

Daily Visitors	Monthly Visits	Pages per Visit	Visit duration	Bounce Rate	Is this your site?
14.2M	429.1M	6.58	07:31	38.15%	Verify your site's metrics.
▲0.39%	▼2.11%	▲1.33%	▲3.05%	▲0.29%	

Daily Unique Visitors:	14,162,939
Monthly Visits:	429,137,052
Pages per Visit:	6.58
Daily Pageviews:	93,229,487
Avg. visit duration:	07:31
Bounce rate:	38.15%
Global Reach:	0.287285%
Monthly Visits (SEMrush):	831,040,651
Monthly Unique Visitors (SEMrush):	246,593,131
Monthly Visits (SimilarWeb):	433,559,346
HypeRank:	4,484
SEMrush Rank:	190
SimilarWeb Rank:	74

- 100 million developers utilizing the platform as of 2023
- Platform hosted over 372 million repositories, including 28 million that are public.
- 413 million contributions open-source contributions made in 2022
- Programming languages used on GitHub, JavaScript, Python, and Java top the list.
- Over 90 percent of Fortune 100 companies use GitHub.

Git and GitHub Advantages

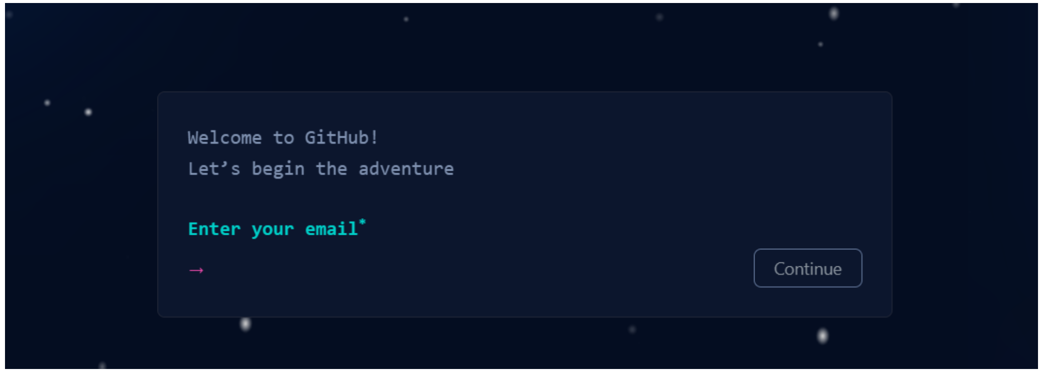
- Distributed Architecture
- Performance
- Branching and Merging
- Data Integrity:
- Workflow Flexibility
- Community and Collaboration
- Integration with Tools and Services
- Pull Requests and Code Review
- Open-Source Initiatives

3. Create a GitHub Repo

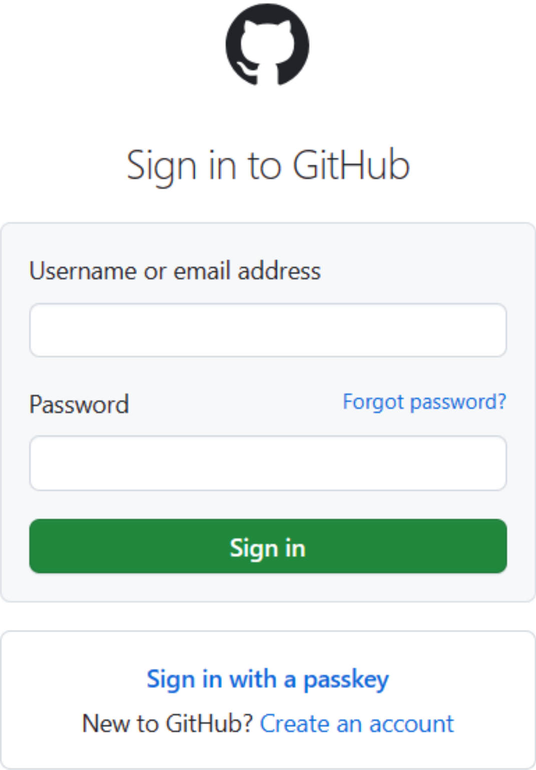
- Sign Up
- New Repository

Create a GitHub Repo - Sign Up

- Ensure you have a GitHub account. If not, sign up at github.com.
- Log in to your GitHub account.



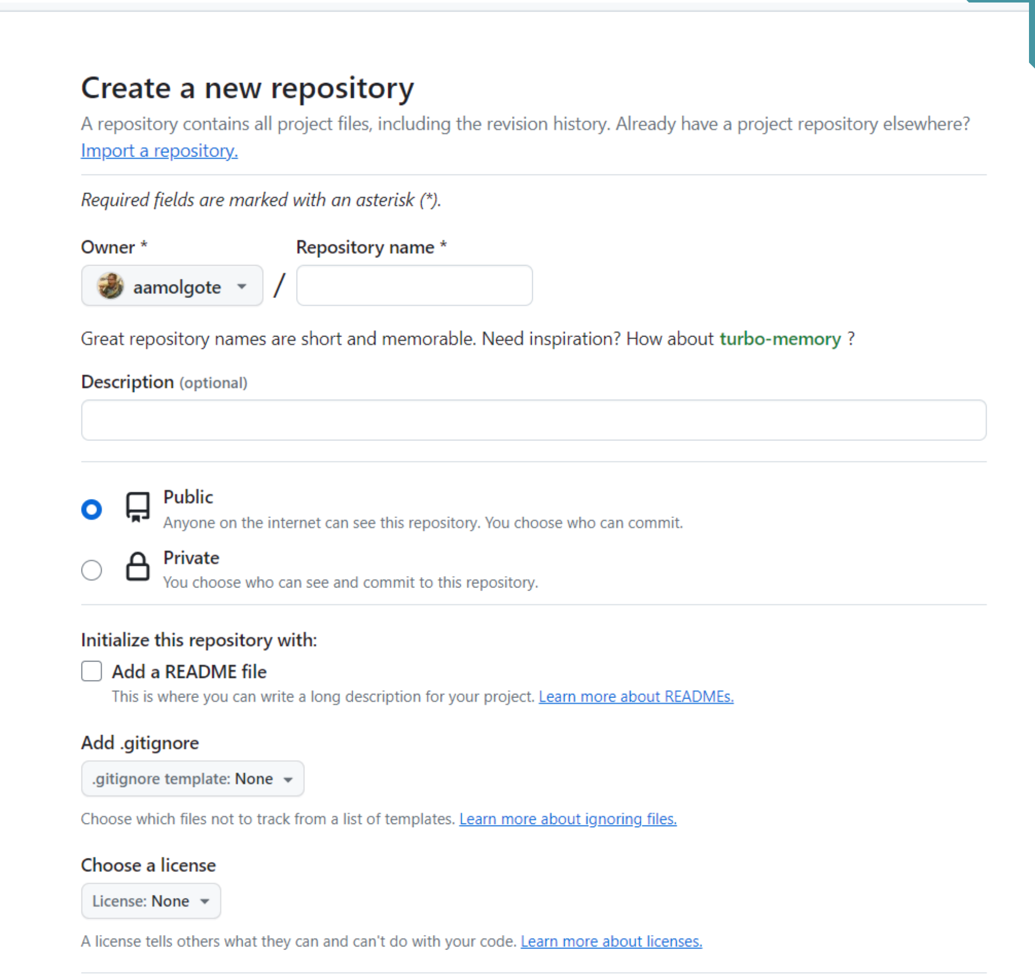
```
Welcome to GitHub!
Let's begin the adventure

Enter your email*
→
```



Sign in to GitHub

Create a GitHub Repo – New Repository

- On the GitHub homepage, look for the "+" icon in the top right corner. Click it and select "New repository" from the dropdown menu.
- Alternatively, you can navigate directly to [Create a new repository.]()
- Repository Name
- Visibility (Public Free)
- Initialize this repository with Readme
- Add .gitignore



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * aamolgote / Repository name *

Great repository names are short and memorable. Need inspiration? How about **turbo-memory** ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

4. Set Up Environment

- Local Dev Set Up
- Clone Repository
- Git Commit

Set Up Environment – Local Dev Set Up

- Git Bash – Download
- Clone Repository
- Commit Change

Set Up Environment – Git Commit

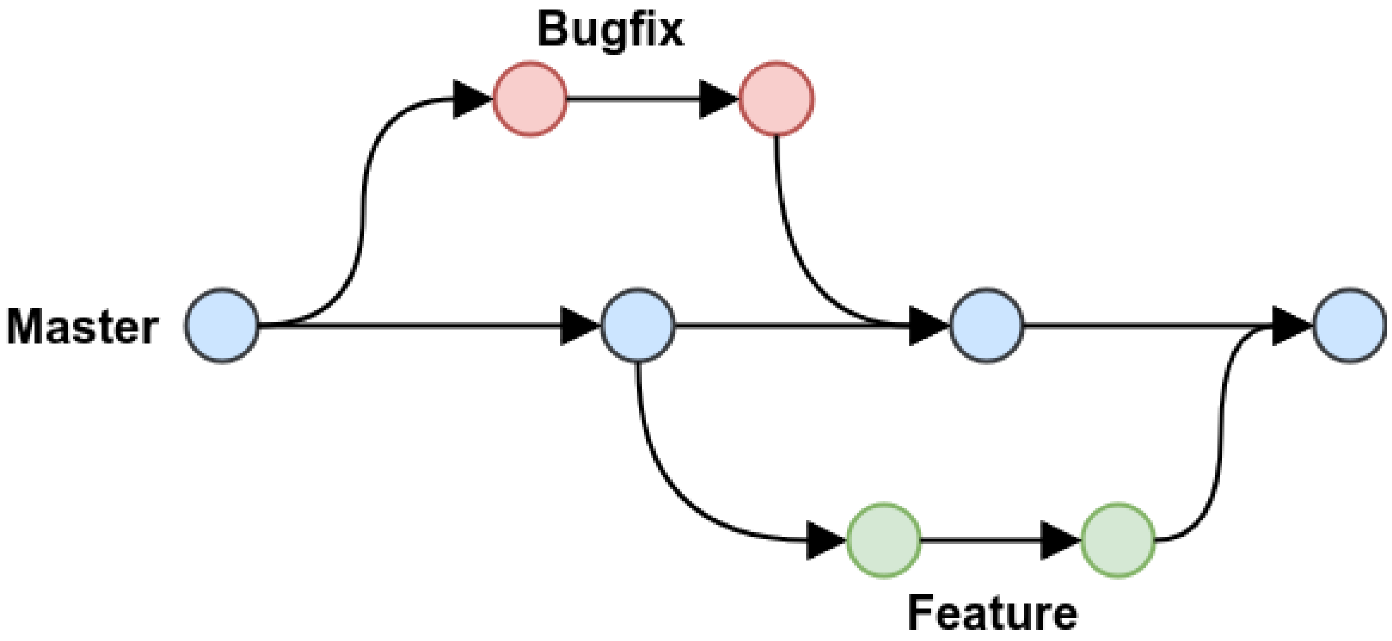
- Key Aspects of a Git Commit
  - Snapshot of Changes
  - Immutable Record
  - Unique Identifier
  - Commit Message
  - Authorship

- Importance of Commits
  - History and Accountability
  - Collaboration
  - Rollback

5. Branches and Pull Requests

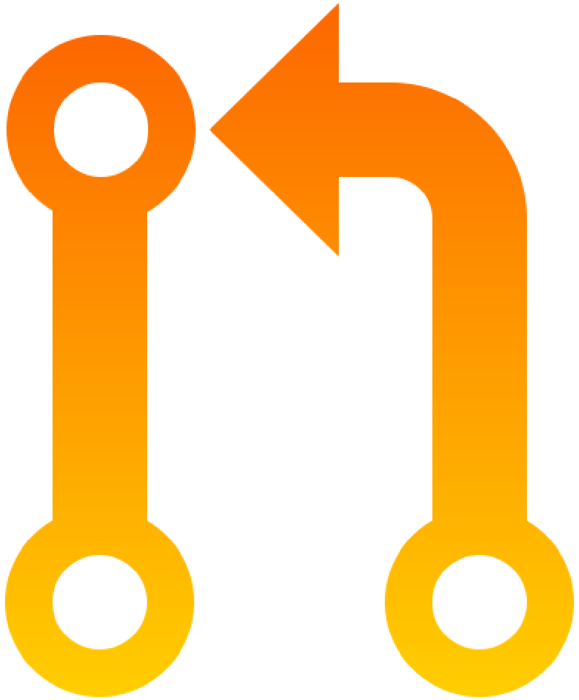
- Branches
- What is a Pull Request
- Create a Pull Request
- Importance of Pull Requests

Branches



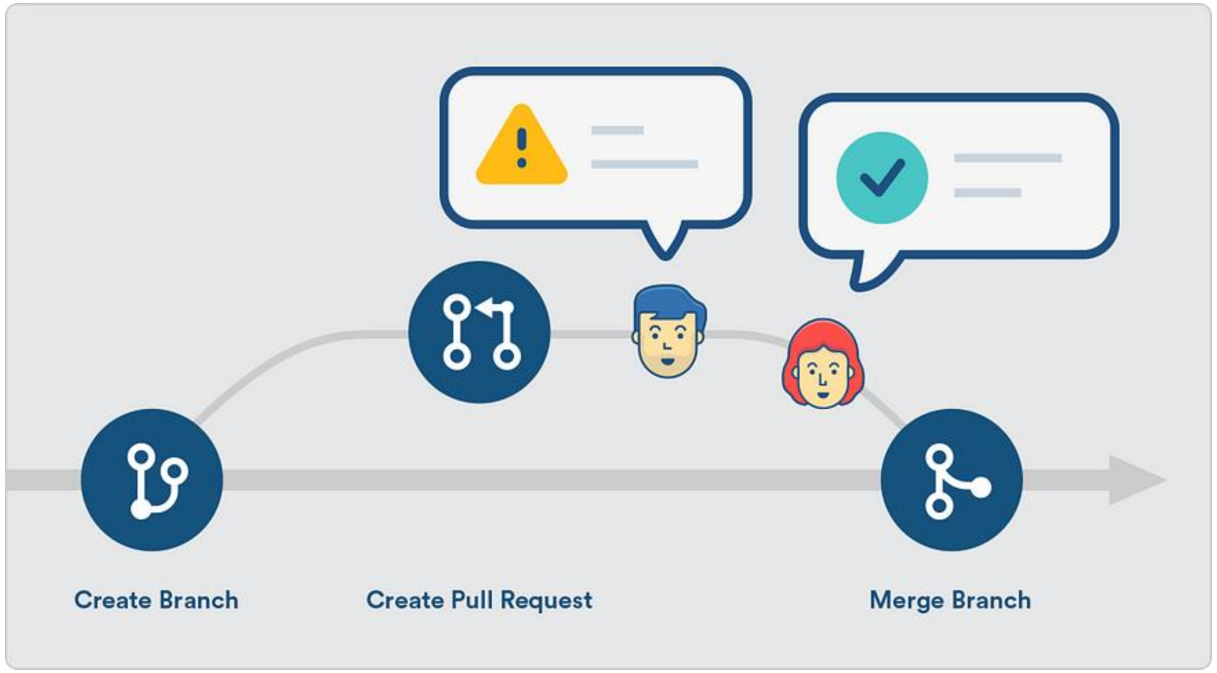
- Isolation of Changes
  - Safe Experimentation
  - Parallel Development
- Simplified Collaboration
  - Focused Contributions
  - Pull Requests for Code Review
- Efficient Merge and Rollback
  - Easier Merge Conflicts Resolution
  - Rollback Capabilities

PRs – What is Pull Request



- **Definition:** A pull request is a feature in version control systems, notably in GitHub, that allows developers to notify team members about changes they've pushed to a branch in a repository. It is essentially a request to review and pull in your contribution to the project's main or base branch.
- **Mechanism:** After pushing changes to a branch in a repository, a developer submits a pull request to the original repository. This action initiates a review process where project collaborators can discuss, review, and eventually accept or reject the proposed changes.
- **Purpose:** The primary aim of a pull request is to facilitate code review and discussion around the code before integrating it into the main project, ensuring quality and functionality.

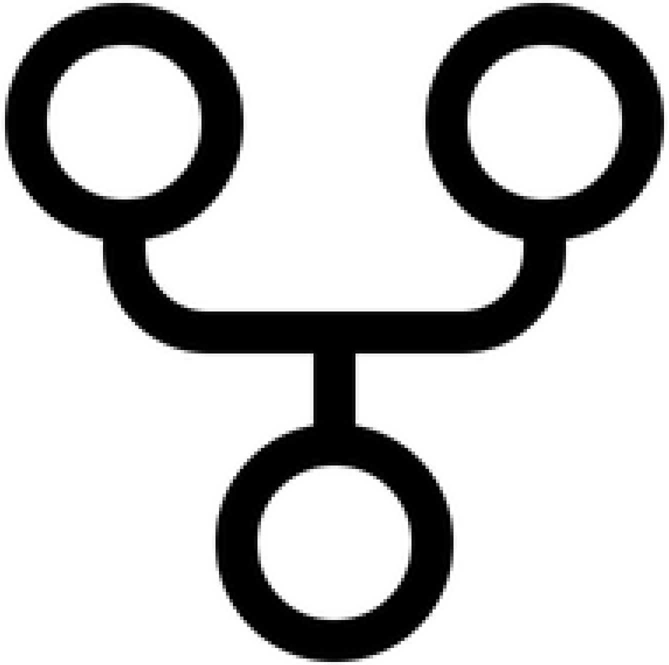
PRs – How to Create Pull Request



- Ensure a Separate Branch for Changes
- Commit Changes to the Branch
- Push the Branch to the Remote Repository
- Initiate the Pull Request on GitHub
- Describe Your Changes
- Submit the Pull Request

PRs – Importance of Pull Requests

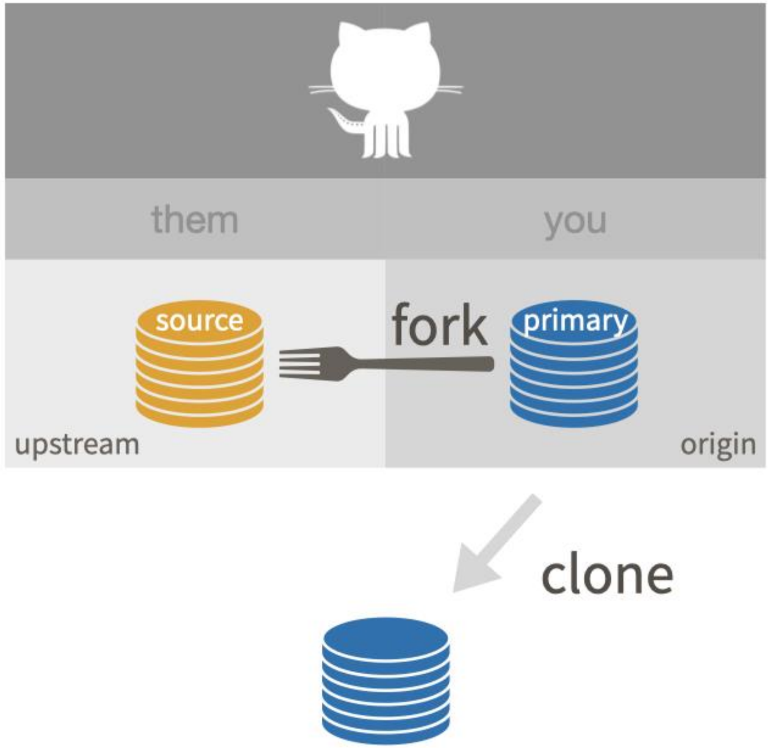
- Facilitates Code Review
- Encourages Collaboration
- Tracks Changes
- Integrates Continuous Integration (CI) Practices
- Improves Project Transparency



6. Forks

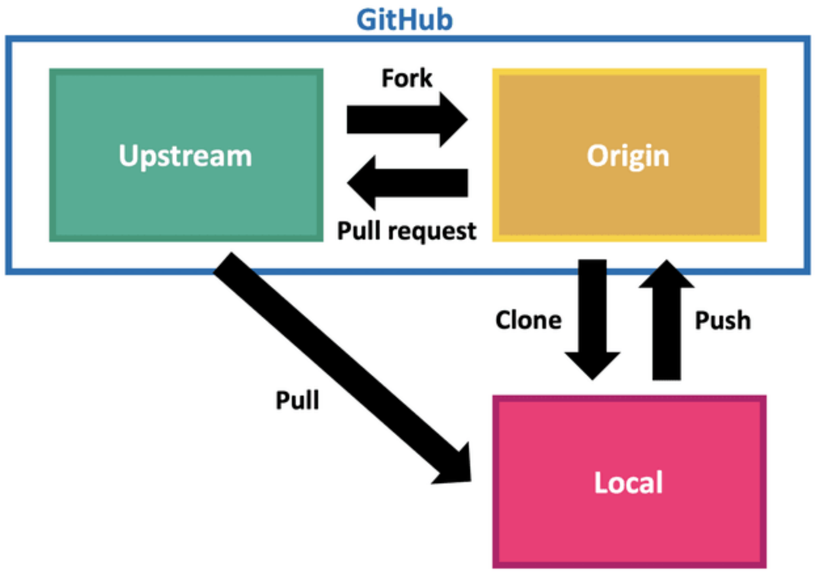
- What are forks
- Working with Forks
- Open-Source Contributions

Forks – What are Forks?



- Identify GitHub Repository to Fork
- Fork the Repository
- Clone Your Fork

Forks – Working with Forks



- Syncing with the Upstream Repository
- Making Changes
- Sending Pull Requests
- Forks and Branches

Forks – Open-Source Contributions

- Essential Tool for Collaboration
- Enables Wide Participation
- Independence in Contribution
- Facilitates Code Review and Integration
- Promotes Project Growth
- Community Engagement
- Contribution Recognition
- Learning and Skill Development

7. Git Tools

- Command Line Git-Bash
- Desktop App
- Git Kraken
- Source Tree
- IDE Integration

8. Best Practices

- Commit Early, Commit Often
- Write Meaningful Commit Messages
- Use Branching Extensively
- Embrace Pull Requests for Collaboration
- Keep the History Clean and Understandable
- Regularly Sync with the Remote Repository
- Use .gitignore
- Backup and Redundancy

9. Miscellaneous

- .git Folder – Purpose, Content, Importance
- .gitignore – Purpose, How it Works, Best Practices
- README.md – Overview, Content, Significance
- GitHub Issues

Quiz - 1

What is a commit in Git?

A. A command to switch between branches

B. A snapshot of your repository at a particular point in time

C. A request to merge one branch into another

D. A way to clone a repository

Answer: B

Quiz - 2

What is the primary difference between Git and GitHub?

- A. Git is a version control system, while GitHub is a cloud-based hosting service for Git repositories.
- B. Git is used for local development, while GitHub is used for online storage only.
- C. There is no difference; they are just different names for the same tool.
- D. GitHub is a version control system, while Git is a code collaboration tool.

Answer: A

Quiz - 3

How can you contribute to an open-source project hosted on GitHub that you do not have write access to?

- A. Directly commit your changes to the main branch
- B. Send an email to the project maintainer with your proposed changes
- C. Fork the repository, make your changes, and submit a pull request
- D. It's not possible to contribute without write access

Answer: C

Quiz - 4

When collaborating on a project using Git and GitHub, what is the purpose of a branch?

- A. To serve as a backup for the main project
- B. To mark the release points of the software
- C. To isolate development work without affecting the main project
- D. To track the issues in the software

Answer: C

Quiz - 5

When you clone a repository in Git, what do you receive?

- A. Only the latest version of the files in the repository
- B. The entire repository history and all branches
- C. A copy of the repository at a specific branch
- D. A compressed archive of the repository's current state

Answer: B

10. Hands On Project

- Create a GitHub Account
- Create Repository
- Clone Repository and create Branch
- Commit Changes and Raise PR
- Raise Pull Request
- Merge Pull Request
- Fork Repository, Raise PR

Ask – Star the project

My Open Source Project

- Request For Star
- Link - https://github.com/iCreateWorks/esigning

Thank you

- LinkedIn
  - https://www.linkedin.com/in/aamolgote
  - https://www.linkedin.com/in/vikas-mendhe-69260012
- ADP List (Mentoring)
  - https://adplist.org/mentors/amol-gote
  - aamolgote@gmail.com