GIT AND GITHUB
#CREATING BRANCHES

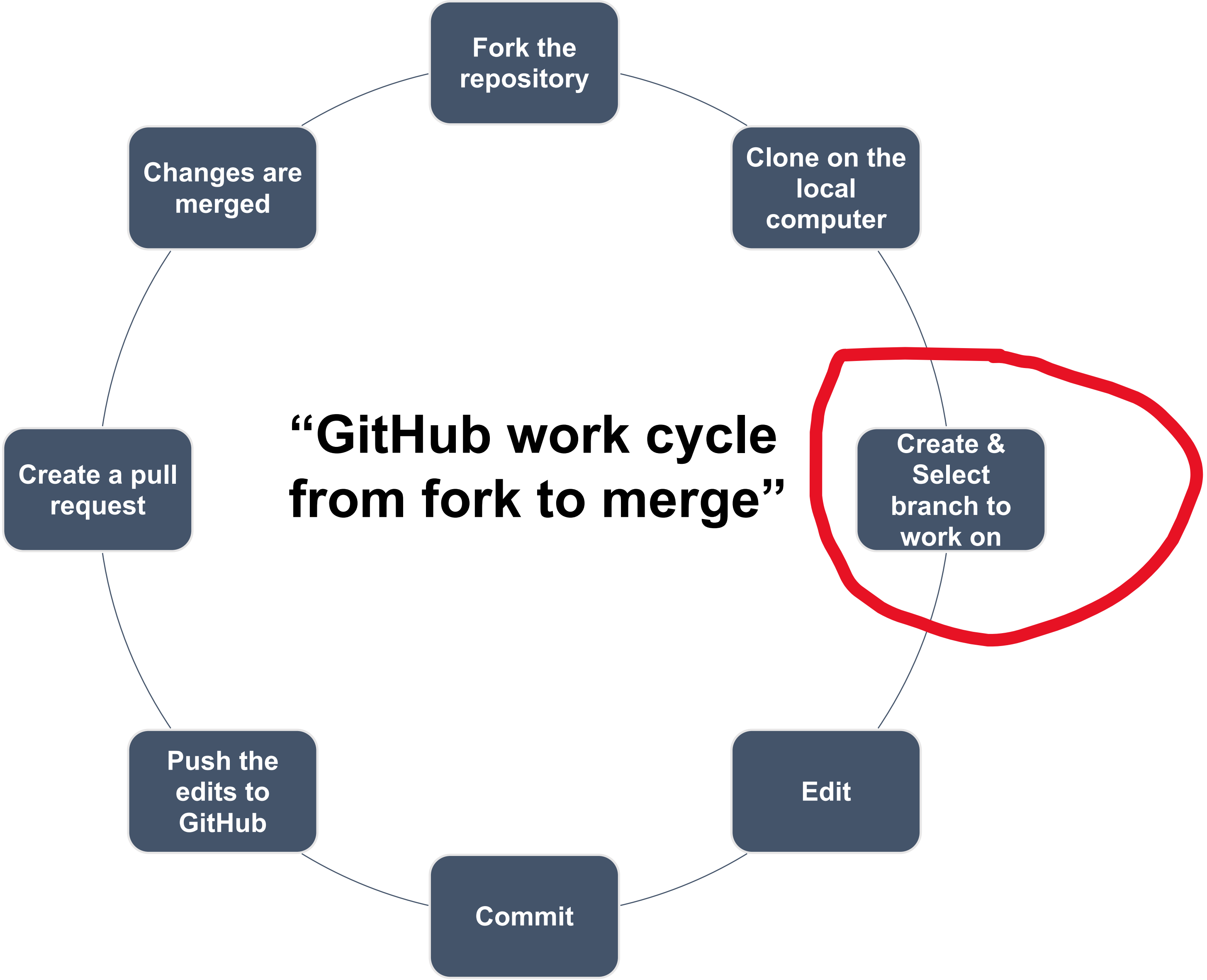
By:

Babirye Sandra Ruth

Introduction to Git and GitHub

- **Git:** A distributed version control system.
- **GitHub:** A web-based platform for hosting Git repositories.
- **Purpose:** collaboration, project and team management, task assignment, etc

Collaborative Development with GitHub



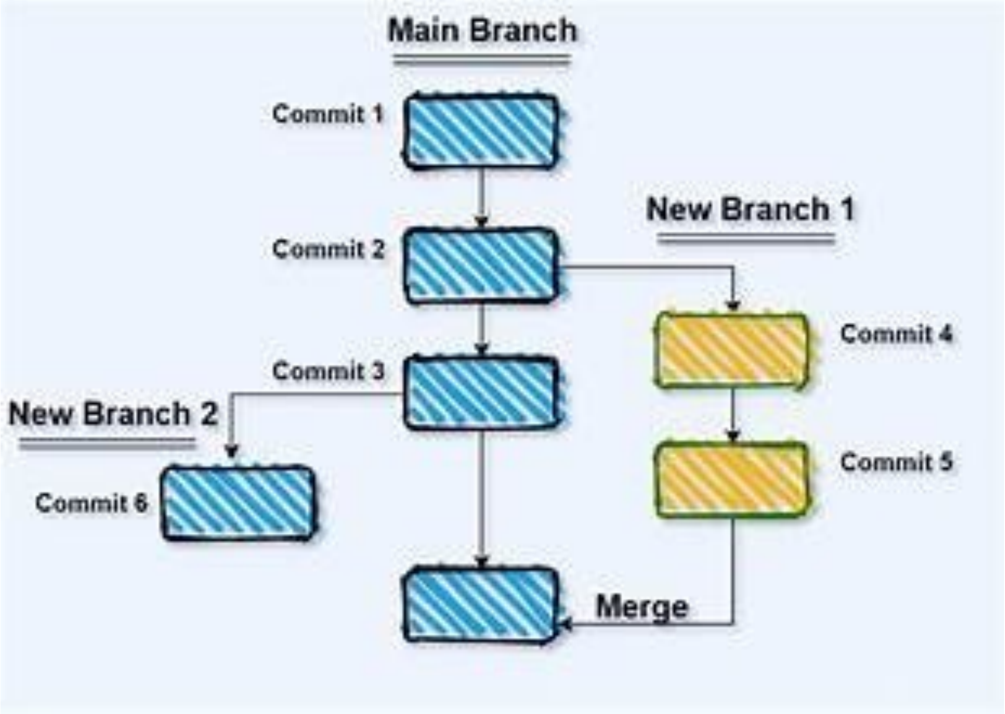
"GitHub work cycle from fork to merge"

Overview of branches on GitHub

- **Definition:** A branch is a separate line of development.
- **Purpose:**

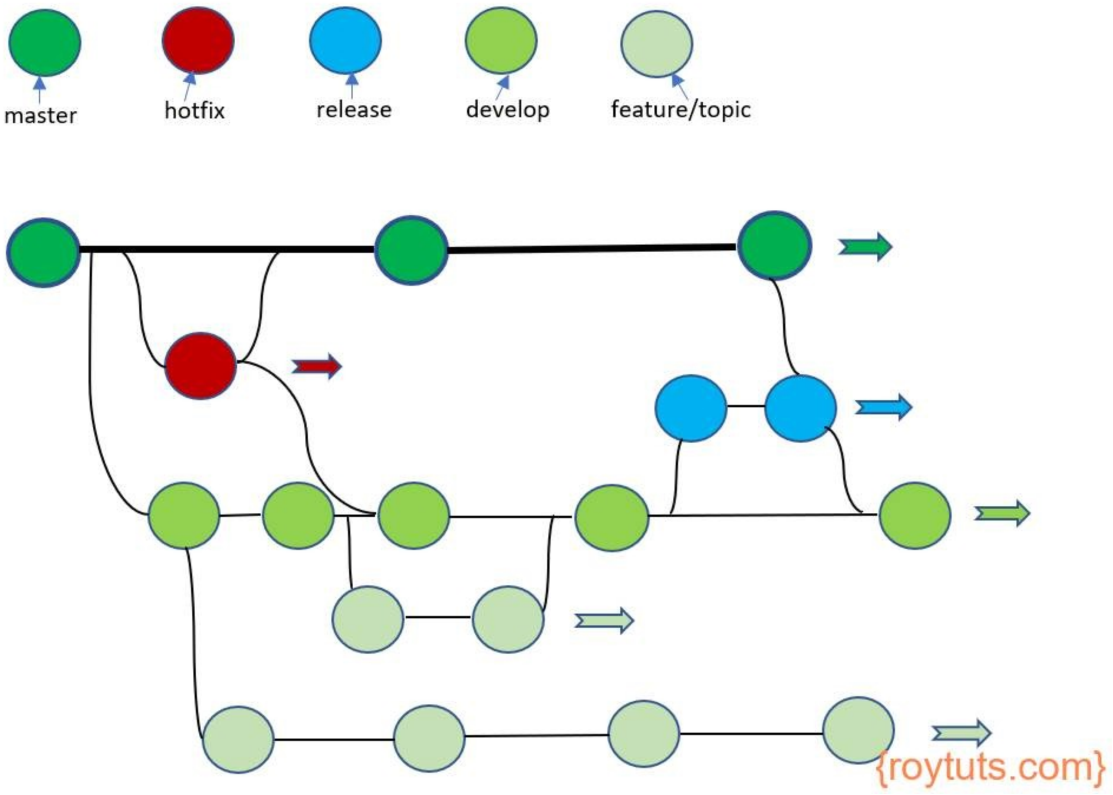
1. Enhanced collaboration
2. Simplified code reviews
3. Safe experimentation
4. Feature enhancement
5. Organized workflow

- **Analogy:** Think of branches like "save points" in a game.



Examples of essential branches

- Main/Master: Stable and production-ready.
- Development (develop): Latest development changes and integration of new features.
- Feature (feature/<name>): Individual new features or experiments.
- Hotfix (hotfix/<name>): Urgent fixes to be deployed immediately.
- Release (release/<version>): Preparing for a new release version.



Basic Git Commands for Branching

1. git branch / git branch –r - List branches.

2. git branch <branch-name> - Create a new branch.

3. git checkout <branch-name> - Switch to a specific branch.

4. git checkout -b <branch-name> - Create and switch to a new branch.

5. git merge <branch-name> - Merge a branch into the current branch.

6. git branch -d <branch-name> - Delete a branch.

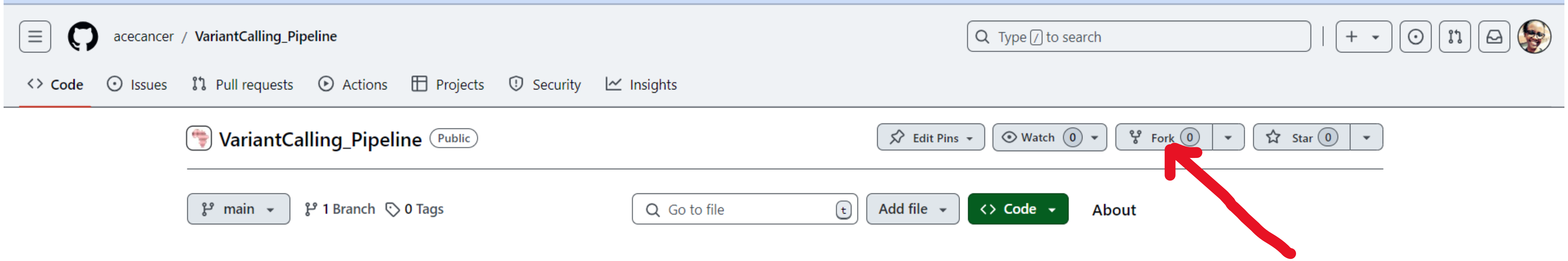
NB: Always use the git cheat sheet for more commands

Creating branches in the terminal demo

Keys steps:

1. Identify the git hub repository you want to contribute to
2. Fork the git hub repository
3. Clone the repository
4. Navigate to the cloned repository
5. Create a branch
6. Switch to the selected branch

Step by step guide

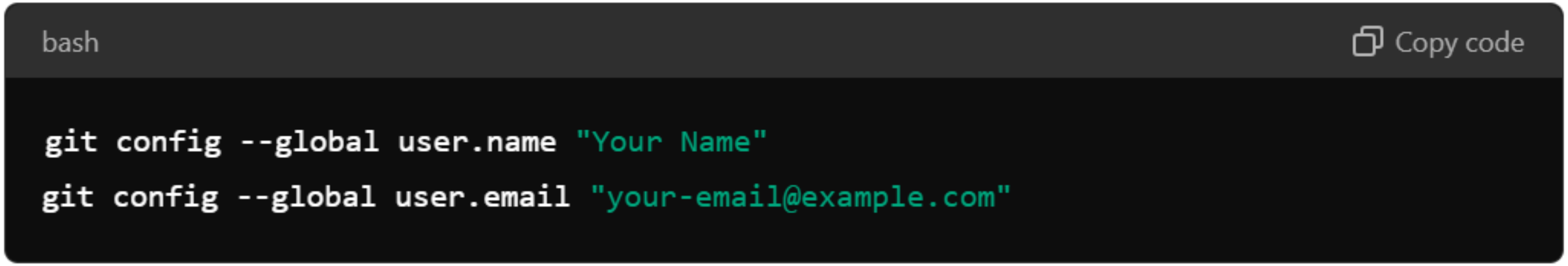


Step1: Fork a Repository on GitHub

- Navigate to the repository you want to work on.
- Click the "Fork" button in the top-right corner to create a personal copy of the repository in your GitHub account.

Step by step guide

Step2: Configure Git (Set Up Identity): Before you start using Git, you need to configure your username and email address. These details are associated with your commits.

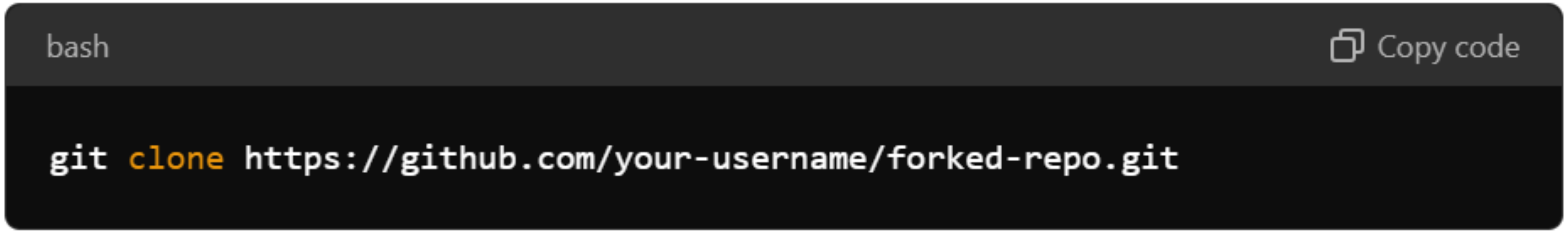


```bash
git config --global user.name "Your Name"
git config --global user.email "your-email@example.com"
```

Step by step guide

Step3: Clone the Forked Repository to Your Local Machine

- Open your terminal or Git Bash.
- Run the command
- Replace the ‘your-username’ with yours and the ‘forked-repo.git’ with the repository name

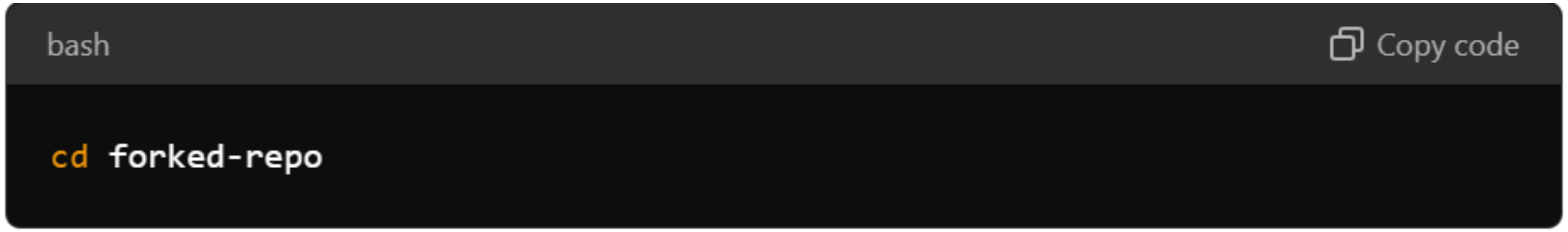


```bash
git clone https://github.com/your-username/forked-repo.git
```

Step by step guide

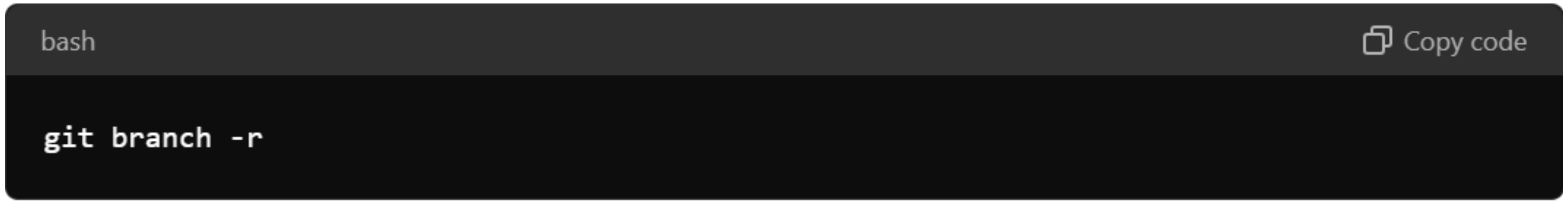
Step4: Navigate to the Cloned Repository:

- Change the directory to your cloned repository



```bash
cd forked-repo
```

Step5: List all branches NB: The currently active branch with be represented with "*"



```bash
git branch -r
```

Step by step guide

Step6: Create a new branch

Command: git branch <branch name>

Step 7: Alternatively if you want to create a branch from another branch as the base

For example, to create a feature-login branch based on develop without first checking out develop:

command: git checkout -b feature-login develop

Or

commands:

```
git checkout develop
git pull – pull the latest changes
git checkout –b  future-login
```

Other basic key commands

- To check the status for the repository – git status
- To add the changes to the staging area- git add <file name> or git add . (all files)
- To commit changes to the repository – git commit –m “commit message”
- To push changes to GitHub – git push origin <branch name>
- To pull changes from GitHub – git pull origin <branch name>
- To delete a local branch – git branch –d <branch name>
- To view remote repositories – git remote –v

Creating branches on GitHub demo

- **Live Demonstration**: (https://github.com/acecancer/VariantCalling_Pipeline)
  - **Create a branch,** make changes*, commit*, push* to GitHub.
  - Open a Pull Request, review, and merge.*
- **Encourage Participation**: Attendees can follow along on their devices.
- **NB:** (This is being done after forking the git hub repository)

Q&A and Discussion

- **Open Floor**: Questions, further explanations, and interactive discussion.

Qn1: How do I retrieve changes made on original repository in my repository

- Response

1. Add upstream repository

command: git remote add upstream https://github.com/original-owner/original-repo.git

2. Fetch the latest changes from the upstream repository

command: git fetch upstream

3. Switch to the branch you want to update

command: git checkout <branch name>

4. Merge the upstream changes into your local branch

command: git merge upstream/<branch name>

5. Push these changes to your forked repository : git push origin <branch name>

Qn2: Retrieving a deleted branch

Response: Git hub usually keeps a reference (commit SHA) to the deleted branches that can be retrieved by running command: git reflog

If the branch you deleted is found then you can restore it using command:

git checkout –b <branch name> commit-hash