## **Git and GitHub: R&D Technical Documentation**

### **1. Setting Up Git**

1. **Install Git**To install Git on your machine, use the following commands based on your operating system:

**Windows**: Download the installer from [Git’s official site](https://git-scm.com/download/win) and follow the setup wizard.

**Mac**:  
bash  
brew install git

**Linux**:  
bash  
sudo apt-get install git

**Configure Git**After installation, configure Git with your name and email. This information will be associated with your commits.  
bash  
git config --global user.name "Your Name"

git config --global user.email "[you@example.com](mailto:you@example.com)"

**Verify Installation**bash  
git --version

### **2. Basic Git Workflow**

**Initialize a Repository**Create a new Git repository by navigating to your project directory and initializing Git.  
bash  
git init

**Adding Files to Staging Area**Add specific files or all files to the staging area.

bash  
git add <file\_name> # Add a specific file

git add . # Add all files

**Committing Changes**Commit the staged changes to the repository.  
bash  
git commit -m "Initial commit message"

**Viewing Commit History**View the history of commits for the project.  
bash  
git log

### **3. Branching in Git**

**Create a New Branch**Branches allow for feature development independent of the main codebase. Use the following command to create a new branch.  
bash  
git branch <branch\_name>

**Switch to a Branch**To switch to a different branch:  
bash  
Copy code  
git checkout <branch\_name>

**Merge a Branch**Merge changes from a feature branch into the main branch.  
bash  
git checkout main

git merge <branch\_name>

**Delete a Branch**Once merged, the branch can be deleted if it’s no longer needed.  
bash  
Copy code  
git branch -d <branch\_name>

### **4. Working with GitHub**

1. **Create a GitHub Repository**
   * Go to [GitHub](https://github.com) and log in.
   * Click on **New** repository, provide a name, and create it.

**Push Local Repository to GitHub**After creating a local repository, you can link it to GitHub by adding a remote and pushing the code.  
bash  
git remote add origin https://github.com/username/repo\_name.git

git push -u origin main

**Cloning a Repository**Clone an existing GitHub repository to your local machine.  
bash  
git clone https://github.com/username/repo\_name.git

**Pulling Changes from GitHub**To keep your local repository updated with remote changes:  
bash  
git pull origin main

**Creating Pull Requests**A pull request is a way to propose changes in one branch to be merged into another. After pushing changes to a branch:

* + Go to your repository on GitHub.
  + Click on **Pull requests** -> **New pull request**.
  + Choose the branches to compare and create the pull request.

### **5. Collaboration Tips**

**Forking a Repository**To contribute to a repository you don’t own, first fork it.

* + Go to the repository page and click **Fork**.
  + Clone your fork to your local machine.

bash  
git clone https://github.com/your\_username/repo\_name.git

**Syncing Forks**Keep your forked repository up to date with the original repository.  
bash  
git remote add upstream https://github.com/original\_owner/repo\_name.git

git fetch upstream

git merge upstream/main

**Resolving Merge Conflicts**If you encounter merge conflicts when merging branches, edit the conflicting files manually, then add and commit the resolved changes.

### **6. Advanced Git Commands**

**Stashing Changes**Save uncommitted changes temporarily.  
bash  
git stash

To reapply stashed changes:  
bash  
git stash pop

**Viewing Differences**See what changes were made compared to the last commit.  
bash  
git diff

**Undoing Commits**Revert changes from the last commit.  
bash  
git revert HEAD