## **Step 1: Create a New Repository on GitHub**

Go to [GitHub](https://github.com/) and **log in**.

Click on **+ (New Repository)**.

Enter a **repository name** (e.g., github-tutorial).

Check **"Add a README file"** (optional).

Click **Create Repository**.

**GitHub Repository Created!**

## **Step 2: Clone the Repository to Your Local Machine**

Now, let's **clone** the repository (download it to your computer).

### **Run the following command in your terminal (Git Bash / Command Prompt / VS Code Terminal)**

git clone <https://github.com/YOUR-USERNAME/github-tutorial.git>

🔹 Replace YOUR-USERNAME with your GitHub username.

**Go inside the cloned repository**

cd github-tutorial

**You have successfully cloned the repository!**

## **Step 3: Create a New File and Push to GitHub**

Now, let’s add a new file and push it to GitHub.

1 **Create a new file (e.g., hello.txt)**

echo "Hello, GitHub!" > hello.txt

**2 Check file status**

git status

3 **Stage the file**

git add hello.txt

**4 Commit the changes**

git commit -m "Added hello.txt"

**5 Push the changes to GitHub**

git push origin main

**Check your repository on GitHub – your new file should be there!**

## **Step 4: Pull Latest Changes from GitHub**

If multiple developers are working on the same project, you need to **pull the latest changes** before making new changes.

git pull origin main

**Your local repository is now up to date with the latest changes from GitHub.**

## **Step 5: Create a New Branch**

Branches allow you to work on features **without affecting the main code**.

1 **Create a new branch (e.g., feature-branch)**

git branch feature-branch

**2 Switch to the new branch**

git checkout feature-branch

🔹 You can also use:

git switch feature-branch

**Now you're working on a separate branch!**

## **Step 6: Make Changes and Push a Branch to GitHub**

Now, let's add another file in the feature-branch.

**1 Create a new file**

echo "This is a feature branch file" > feature.txt

**2 Stage, commit, and push to GitHub**

git add feature.txt  
git commit -m "Added feature.txt in feature-branch"  
git push origin feature-branch

**Your branch is now visible on GitHub!**

## **Step 7: Merge the Feature Branch into Main**

After testing, let's merge the feature into the **main branch**.

1 **Switch back to the main branch**

git checkout main

**2 Pull the latest changes (to avoid conflicts)**

git pull origin main

3 **Merge feature-branch into main**

git merge feature-branch

4️⃣ **Push the merged changes to GitHub**

git push origin ma in

**The feature is now merged into the main branch!**

## **Step 8: Delete the Feature Branch (Optional)**

Once merged, the feature branch is no longer needed.

1**Delete the branch locally**

git branch -d feature-branch

**2 Delete the branch from GitHub**

git push origin --delete feature-branch

**The branch is now deleted!**

## **Bonus: Handling Merge Conflicts**

If there are conflicts while merging, Git will notify you. To resolve:

**1 Open the conflicted file in VS Code or any editor.**  
2 **Look for <<<<<< HEAD and ======= sections.**  
3 **Keep the correct version and remove conflict markers.**  
4 **Stage, commit, and push the resolved file.**

git add conflicted-file.txt  
git commit -m "Resolved merge conflict"  
git push origin main

**Merge conflict resolved successfully!**