

GitHub Workflow Documentation

This documentation outlines the step-by-step process for setting up a GitHub repository, creating branches, and simulating a complete development workflow.

1. Create a Repository on GitHub

Steps:

1. Log in to your GitHub account.
2. Click **New** to create a new repository.
3. Enter a repository name and description, and choose whether it's public or private.
4. Click **Create repository**.
5. Next we have to install **Git** on our local.

```
W: http://archive.ubuntu.com/ubuntu/dists/jammy/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/owncloud.gpg are ignored as the file has an unsupported filetype.
W: https://ppa.launchpadcontent.net/ondrej/php/ubuntu/dists/jammy/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/owncloud.gpg are ignored as the file has an unsupported fi
e.
W: http://security.ubuntu.com/ubuntu/dists/jammy-security/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/owncloud.gpg are ignored as the file has an unsupported filetype.
W: http://archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/owncloud.gpg are ignored as the file has an unsupported filetype.
W: http://archive.ubuntu.com/ubuntu/dists/jammy-backports/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/owncloud.gpg are ignored as the file has an unsupported filetype.
root@node215347-env-1462448:~# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libxmlrpc-epi0
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 37 not upgraded.
Need to get 4,146 kB of archives.
After this operation, 21.0 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.11 [955 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3,165 kB]
Fetched 4,146 kB in 3s (1,362 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 46489 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1:2.34.1-1ubuntu1.11_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.11) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1:2.34.1-1ubuntu1.11_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.11) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.11) ...
Setting up git (1:2.34.1-1ubuntu1.11) ...
root@node215347-env-1462448:~# clear
```

Activate Windows

2. Clone the Repository

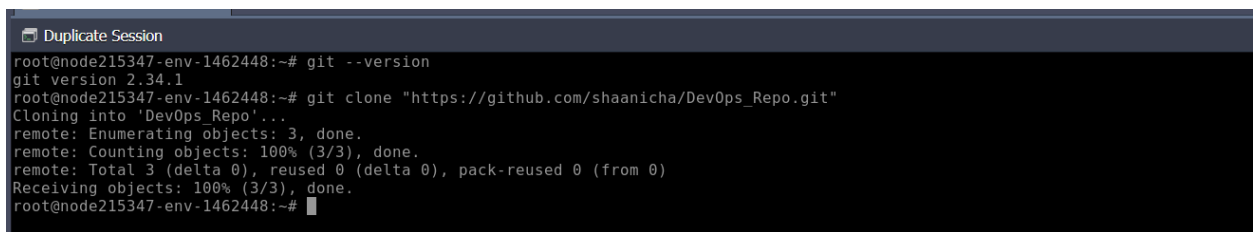
Command:

-> `git clone https://github.com/your-username/repository-name.git`

-> `cd repository-name`

Explanation:

- `git clone`: Clones the repository to your local machine.
- `cd`: Navigates into the cloned repository directory.

A terminal window titled "Duplicate Session" showing a series of commands and their outputs. The commands are: `git --version`, `git clone "https://github.com/shaanicha/DevOps_Repo.git"`, and `cd DevOps_Repo`. The outputs show the git version (2.34.1), the cloning progress (Enumerating objects, Counting objects, Total 3), and the directory change to DevOps_Repo.

```
root@node215347-env-1462448:~# git --version
git version 2.34.1
root@node215347-env-1462448:~# git clone "https://github.com/shaanicha/DevOps_Repo.git"
Cloning into 'DevOps_Repo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
root@node215347-env-1462448:~#
```

3. Create the Main Branch

-> `git checkout -b main`

-> `git push -u origin main`

Probably we used this above commands to create main branch. But I already create main branch as default branch on github manually.

Explanation:

- `git checkout -b main`: Creates a new branch named `main` and switches to it.
- `git push -u origin main`: Pushes the `main` branch to the remote repository and sets it as the upstream branch.

4. Create Development and Feature Branches

Commands:

-> git checkout -b feature/feature1

```
root@node215347-env-1462448:~#  
root@node215347-env-1462448:~# git checkout -b feature/feature1  
Switched to a new branch 'feature/feature1'  
root@node215347-env-1462448:~#  
root@node215347-env-1462448:~# git status  
On branch feature/feature1
```

Commands:

-> git checkout -b develop

-> git push -u origin develop

Explanation:

- **git checkout -b develop**: Creates and switches to the **develop** branch.

```
📄 Duplicate Session  
root@node215347-env-1462448:~# git checkout -b develop  
Switched to a new branch 'develop'  
root@node215347-env-1462448:~# git push -u origin develop  
Username for 'https://github.com': shaanicha  
Password for 'https://shaanicha@github.com':  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0  
remote:  
remote: Create a pull request for 'develop' on GitHub by visiting:  
remote:      https://github.com/shaanicha/DevOps_Repo/pull/new/develop  
remote:  
To https://github.com/shaanicha/DevOps_Repo.git  
* [new branch]      develop -> develop  
Branch 'develop' set up to track remote branch 'develop' from 'origin'.  
root@node215347-env-1462448:~# █
```

5. Work on a Feature Branch

Commands:

-> echo "Line from feature/feature1" > conflict-file.txt

-> git add conflict-file.txt

-> git commit -m "Add conflict file from feature branch"

-> git push origin feature/feature1

```
root@node215347-env-1462448:~#
root@node215347-env-1462448:~# echo "Line from feature/feature1" > conflict-file.txt
root@node215347-env-1462448:~#
root@node215347-env-1462448:~# git add conflict-file.txt
root@node215347-env-1462448:~#
root@node215347-env-1462448:~# git commit -m "Add conflict file from feature branch"
[feature/feature1 a235e40] Add conflict file from feature branch
Committer: root <root@localhost.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 conflict-file.txt
root@node215347-env-1462448:~#
root@node215347-env-1462448:~# git push origin feature/feature1
Username for 'https://github.com':
```

Explanation:

- **echo**: Adds content to a new file named **feature1.txt**.
- **git add**: Stages the file for commit.
- **git commit**: Commits the changes with a descriptive message.
- **git push**: Pushes the changes to the remote feature branch.

5. Work on a Develop Branch:

Commands:

-> echo "Line from develop" > conflict-file.txt

-> git add conflict-file.txt

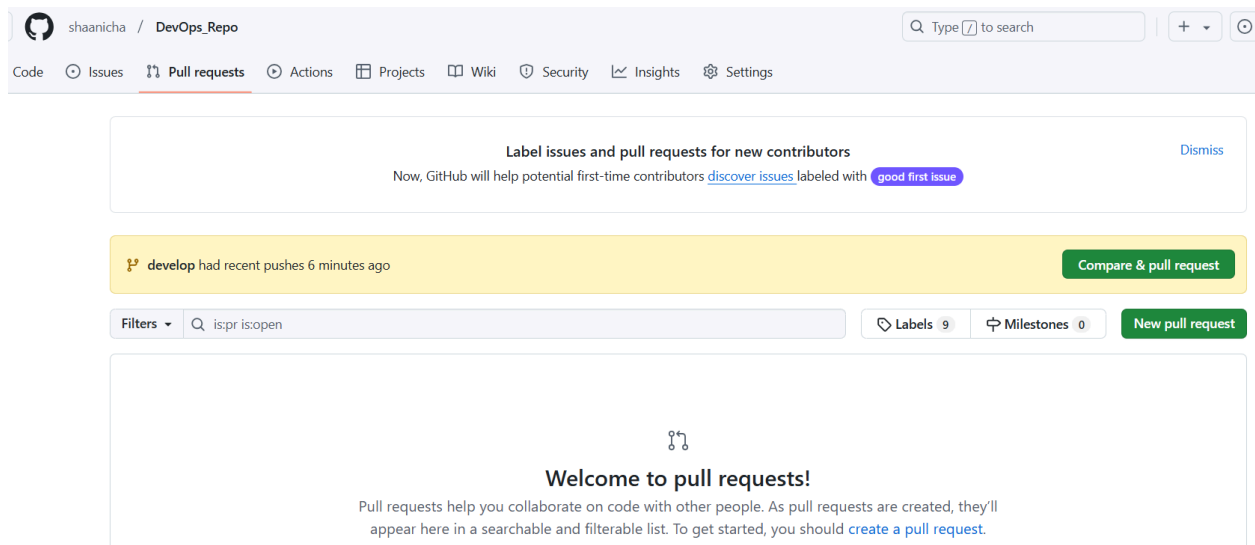
-> git commit -m "Add conflict file from develop"

-> git push origin develop

6. Create a Pull Request (PR)

Steps:

1. Navigate to the repository on GitHub.
2. Go to the **Pull Requests** tab and click **New Pull Request**.
3. Select feature/feature1 as the source branch and develop as the target branch.
4. Add a title and description, then click **Create Pull Request**.



1. **Create a Pull Request from feature/feature1 to develop:**
 - On GitHub, create a pull request.

- This will now show a **merge conflict** because the same file (**conflict-file.txt**) has conflicting changes in both branches.

The screenshot shows the GitHub web interface for comparing changes between two branches. At the top, the navigation bar includes links for Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the 'Comparing changes' section is active, displaying a comparison between 'base: develop' and 'compare: feature/feature1'. A red error message states: 'Can't automatically merge. Don't worry, you can still create the pull request.' A blue bar below the comparison offers to 'Create pull request'. The commit summary shows '1 commit', '1 file changed', and '1 contributor'. A commit from 'root' on Dec 10, 2024, is highlighted with the message 'Add conflict file from feature branch'. The diff view for 'conflict-file.txt' shows a conflict: the base branch has a line starting with '...', while the feature branch has a line starting with '1 + Line from feature/feature1'. The interface also includes a 'Split/Unified' view toggle and a file explorer on the left.

7. Resolve Merge Conflicts

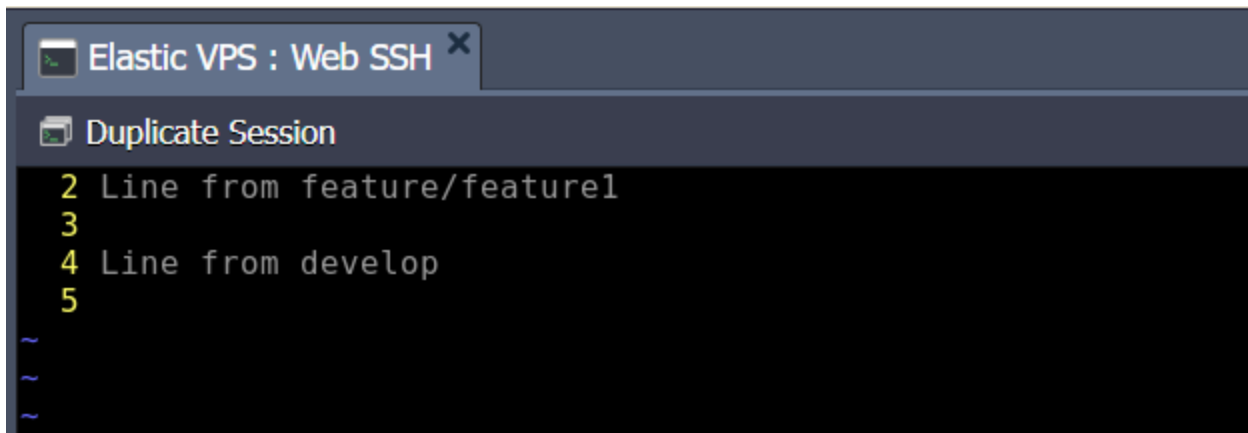
1. Identify the Conflicted File: From the output:

2. Open the Conflicted File: Open **conflict-file.txt** in a text editor

-> **vim conflict-file.txt**

Resolve the Conflict: Decide which content to keep, combine, or modify.

For example:



The screenshot shows a terminal window titled "Elastic VPS : Web SSH" with a "Duplicate Session" button. The terminal content displays a merge conflict resolution for a file named "conflict-file.txt". The conflict is resolved by taking "Line from feature/feature1" (lines 2 and 3) and "Line from develop" (line 4). Line 5 is also present. The terminal shows the following lines:

```
2 Line from feature/feature1
3
4 Line from develop
5
~
~
~
```

3. Save and Exit the File:

4. Mark the Conflict as Resolved: Add the resolved file to the staging area:

-> `git add conflict-file.txt`

5. Commit the Changes: Once the conflict is resolved, commit the changes:

-> `git commit -m "Resolve merge conflict in conflict-file.txt"`

6. Push the Updated Branch: Push the resolved branch back to the remote repository:

-> `git push origin feature/feature1`

```
Elastic VPS : Web SSH x
Duplicate Session
root@node215347-env-1462448:~# git add conflict-file.txt
root@node215347-env-1462448:~# git commit -m "Resolve merge conflict in conflict-file.txt"
[feature/feature1 13eab36] Resolve merge conflict in conflict-file.txt
Committer: root <root@localhost.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

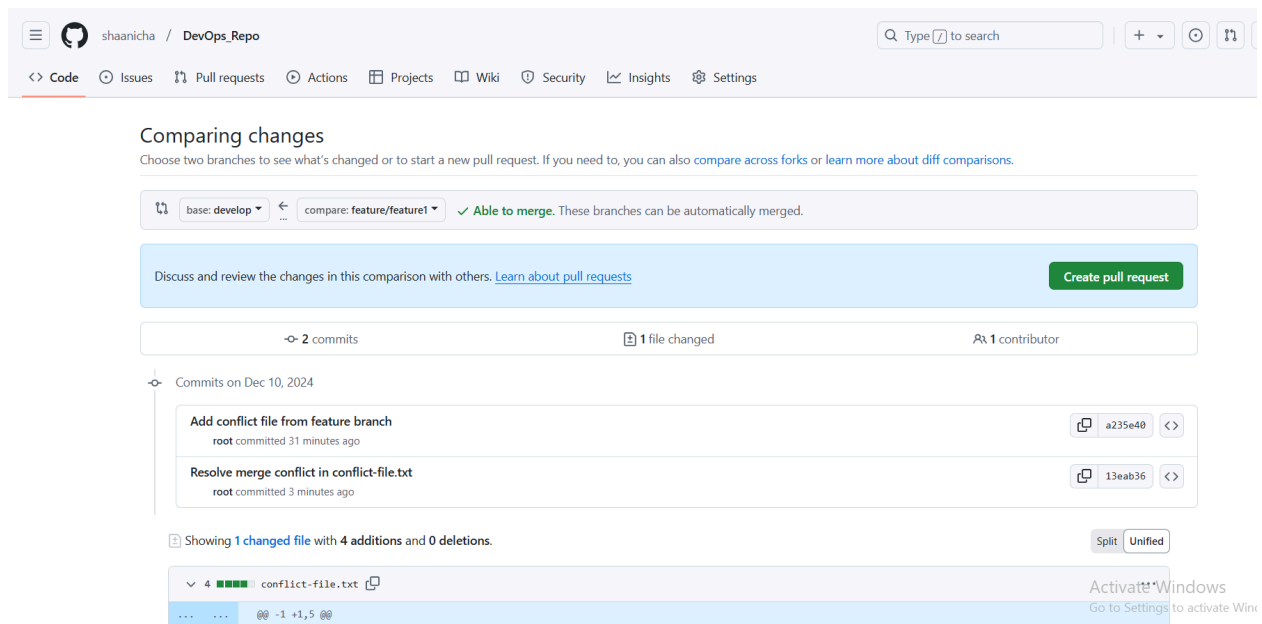
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

root@node215347-env-1462448:~# git push origin feature/feature1
Username for 'https://github.com': shaanicha
Password for 'https://shaanicha@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 365 bytes | 365.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/shaanicha/DevOps_Repo.git
a235e40..13eab36 feature/feature1 -> feature/feature1
root@node215347-env-1462448:~#
```

7. Check the Pull Request on GitHub

- Go back to your GitHub repository and view the pull request.
- It should now indicate that the conflict is resolved and allow you to merge the branches.



7. Merge Develop into main

Commands:

-> `git checkout main`

-> `git merge develop`

-> `git push origin main`

Explanation:

git merge: Integrates changes from develop into main.

git push: Updates the remote main branch with the merged changes.

Created By

Shraddha Chaudhari