

Case Study 1 - Git Workflow - Method - 2

Problem Statement:

You work as a DevOps Architect in Zendrix Softwares. The company has been struggling to manage their product releases. The releases should happen on the 25th of every month. Suggest a Git workflow architecture for this requirement.

Simulate this workflow by creating pseudo code files and branches and upload the same to your GitHub account.

As a part of the solution, share the link to your GitHub repository.

Solution:-

I suggest git workflow for this task git flow work flow. For this Git-flow we used the given below steps:-

```
$ sudo su
# apt update
```

```
ubuntu@Git-CaseStudy-1-Mtd-2:~$ sudo su
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
```

```
# apt upgrade
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
#
# An OpenSSL vulnerability has recently been fixed with USN-6188-1 & 6119-1:
# CVE-2023-2650: possible DoS translating ASN.1 object identifiers.
# Ensure you have updated the package to its latest version.
#
The following NEW packages will be installed:
  linux-aws-5.19-headers-5.19.0-1028 linux-headers-5.19.0-1028-aws
  linux-image-5.19.0-1028-aws linux-modules-5.19.0-1028-aws
```

```
# apt install git
# which git
# git --version
# git config --global user.name "Amit Tiwari"
# git config --global user.email "redhat.amitiwari@gmail.com"
# git config --list
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.9).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# which git
/usr/bin/git
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# git --version
git version 2.34.1
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# git config --global user.name "Amit Tiwari"
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# git config --global user.email "redhat.amitiwari@gmail.com"
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# git config --list
user.name=Amit Tiwari
user.email=redhat.amitiwari@gmail.com
```

Simulate this workflow by creating pseudo code files and branches:-

```
# mkdir git-flow_workflow
# cd git-flow_workflow
# apt install git-flow
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# mkdir git-flow_workflow
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu# cd git-flow_workflow
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# apt install git-flow
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  git-flow
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 38.6 kB of archives.
After this operation, 329 kB of additional disk space will be used.
```

```
# git flow init
# git branch
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/git-flow_workflow/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/ubuntu/git-flow_workflow/.git/hooks]
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
* develop
master
```

```
# git flow feature start feature1
# git branch
# touch feature1.txt
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow feature start feature1
Switched to a new branch 'feature/feature1'

Summary of actions:
- A new branch 'feature/feature1' was created, based on 'develop'
- You are now on branch 'feature/feature1'

Now, start committing on your feature. When done, use:

    git flow feature finish feature1

root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
develop
* feature/feature1
master
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# touch feature1.txt
```

```
# git add feature1.txt && git commit -m "feature1.txt file commit in
feature1 branch." feature1.txt
# git flow feature finish feature1
```

```

root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git add feature1.txt
&& git commit -m "feature1.txt file commit in feature1 branch." feature1.txt
[feature/feature1 2738302] feature1.txt file commit in feature1 branch.
 1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 feature1.txt
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow feature finish feature1
Switched to branch 'develop'
Updating 7e3dcda..2738302
Fast-forward
 feature1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 feature1.txt
Deleted branch feature/feature1 (was 2738302).

Summary of actions:
- The feature branch 'feature/feature1' was merged into 'develop'
- Feature branch 'feature/feature1' has been locally deleted
- You are now on branch 'develop'

```

```

# git flow feature start feature2
# git branch
# touch feature2.txt

```

```

root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow feature start feature2
Switched to a new branch 'feature/feature2'

Summary of actions:
- A new branch 'feature/feature2' was created, based on 'develop'
- You are now on branch 'feature/feature2'

Now, start committing on your feature. When done, use:

    git flow feature finish feature2

root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
  develop
* feature/feature2
  master
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# touch feature2.txt

```



```
# git add feature2.txt && git commit -m "feature2.txt file commit in feature2 branch." feature2.txt
# git branch
# git flow feature finish feature2
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git add feature2.txt
&& git commit -m "feature2.txt file commit in feature2 branch." feature2.txt
[feature/feature2 086a738] feature2.txt file commit in feature2 branch.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature2.txt
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
  develop
* feature/feature2
  master
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow feature finish feature2
Switched to branch 'develop'
Updating 2738302..086a738
Fast-forward
 feature2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature2.txt
Deleted branch feature/feature2 (was 086a738).

Summary of actions:
- The feature branch 'feature/feature2' was merged into 'develop'
- Feature branch 'feature/feature2' has been locally deleted
- You are now on branch 'develop'
```

```
# git checkout master
# ls
# git merge develop
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git checkout master
Switched to branch 'master'
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# ls
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git merge develop
Updating 7e3dcda..086a738
Fast-forward
 feature1.txt | 0
 feature2.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature1.txt
 create mode 100644 feature2.txt
```

```
# git branch
# git flow hotfix start 1.0
# touch h.txt
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
  develop
* master
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow hotfix start
1.0
Switched to a new branch 'hotfix/1.0'

Summary of actions:
- A new branch 'hotfix/1.0' was created, based on 'master'
- You are now on branch 'hotfix/1.0'

Follow-up actions:
- Start committing your hot fixes
- Bump the version number now!
- When done, run:

    git flow hotfix finish '1.0'

root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# touch h.txt
```

```
# git add h.txt && git commit -m "last minute h.txt file changes
commit in hotfix/1.0 branch." h.txt
# git branch
# ls
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git add h.txt && git
commit -m "last minute h.txt file changes commit in hotfix/1.0 branch." h.txt
[hotfix/1.0 24aec85] last minute h.txt file changes commit in hotfix/1.0 branch.
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 h.txt
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git branch
  develop
* hotfix/1.0
  master
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# ls
feature1.txt  feature2.txt  h.txt
```

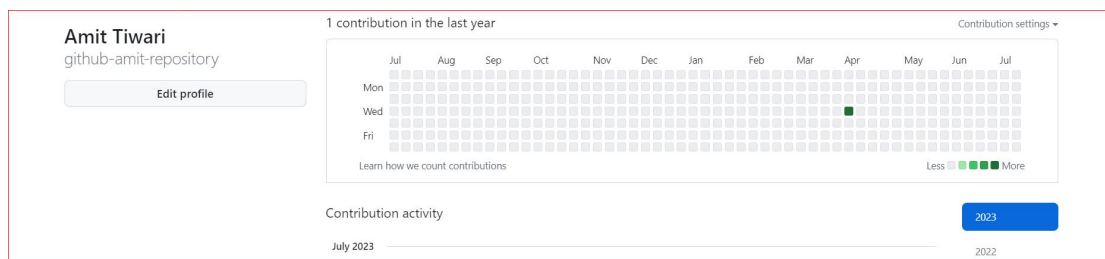
git flow hotfix finish 1.0

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git flow hotfix finish
h 1.0
Switched to branch 'master'
Merge made by the 'ort' strategy.
 h.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 h.txt
Switched to branch 'develop'
Merge made by the 'ort' strategy.
 h.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 h.txt
Deleted branch hotfix/1.0 (was 24aec85).

Summary of actions:
- Hotfix branch 'hotfix/1.0' has been merged into 'master'
- The hotfix was tagged '1.0'
- Hotfix tag '1.0' has been back-merged into 'develop'
- Hotfix branch 'hotfix/1.0' has been locally deleted
- You are now on branch 'develop'
```

* Upload the same to your GitHub account.

* Create github account



* github account login

The screenshot shows the GitHub login interface. At the top is the GitHub logo and the text 'Sign in to GitHub'. Below this is a form with two input fields: 'Username or email address' (containing 'github-amit-repository') and 'Password' (masked with dots). A 'Forgot password?' link is next to the password field. A green 'Sign in' button is at the bottom of the form.

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 *

 github-amit-repository

Repository name *

Case_Study_1-Git_Workflo

✓ Case_Study_1-Git_Workflow-Method-2 is available.

Great repository names are short and memorable. Need inspiration? How about [shiny-invention](#) ?

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:

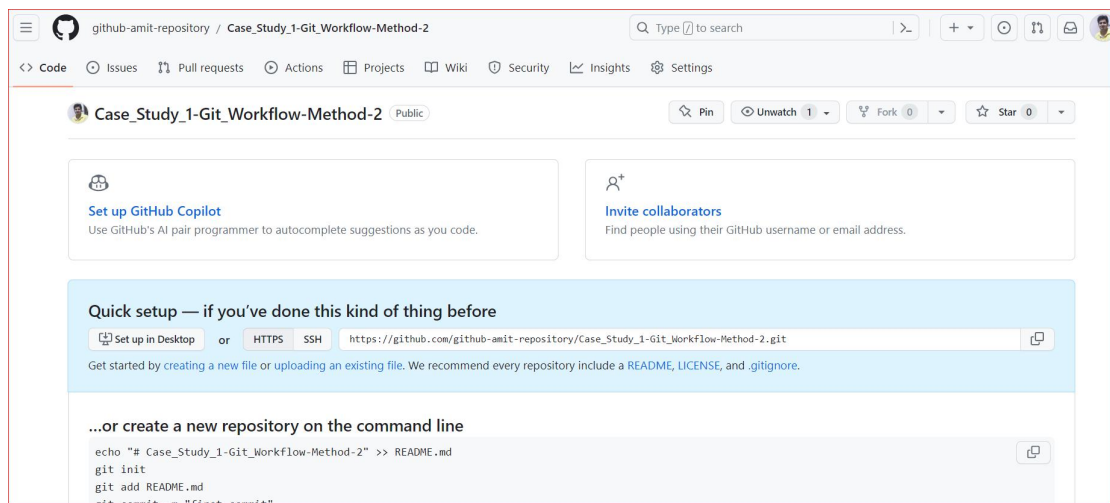
☐ Add a README file

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.](#)



The screenshot shows the GitHub repository creation page for a repository named 'Case_Study_1-Git_Workflow-Method-2' under the owner 'github-amit-repository'. The repository is public. The page includes options to initialize the repository with a README file and a .gitignore template. It also provides quick setup instructions for desktop, HTTPS, and SSH, and a command line setup for creating a new repository.

github-amit-repository / Case_Study_1-Git_Workflow-Method-2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Case_Study_1-Git_Workflow-Method-2 Public

Pin Unwatch 1 Fork 0 Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Invite collaborators
Find people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/github-amit-repository/Case_Study_1-Git_Workflow-Method-2.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Case_Study_1-Git_Workflow-Method-2" >> README.md
git init
git add README.md
git commit -m "First commit"
```

```
# git remote add origin "https://github.com/github-amit-repository/Case_Study_1-Git_Workflow-Method-2.git"
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git remote add origin "https://github.com/github-amit-repository/Case_Study_1-Git_Workflow-Method-2.git"
```



```
# git push origin master
```

```
root@Git-CaseStudy-1-Mtd-2:/home/ubuntu/git-flow_workflow# git push origin master
Username for 'https://github.com': github-amit-repository
Password for 'https://github-amit-repository@github.com':
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 956 bytes | 956.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/github-amit-repository/Case_Study_1-Git_Workflow-Method-2.git
* [new branch]      master -> master
```

The screenshot shows the GitHub interface for a repository named 'Case_Study_1-Git_Workflow-Method-2'. The repository is public and has 1 branch and 0 tags. The main branch is 'master'. The repository owner is 'github-amit-repository'. The commit history shows a recent commit 'c0d3e05' made 8 minutes ago, which is a merge of the 'hotfix/1.0' branch. The commit message is 'hostfix finish.Merge branch 'hotfix/1.0''. The commit includes three files: 'feature1.txt', 'feature2.txt', and 'h.txt'. The commit history also shows two previous commits: 'feature1.txt file commit in feature1 branch.' (21 minutes ago) and 'feature2.txt file commit in feature2 branch.' (16 minutes ago). At the bottom, there is a prompt to 'Add a README' to help people understand the project.

File	Commit Message	Time Ago
feature1.txt	feature1.txt file commit in feature1 branch.	21 minutes ago
feature2.txt	feature2.txt file commit in feature2 branch.	16 minutes ago
h.txt	last minute h.txt file changes commit in hotfix/1.0 branch.	9 minutes ago

Shared link of GitHub repository:-

https://github.com/github-amit-repository/Case_Study_1-Git_Workflow-Method-2.git