

Day 10 Task:

Advance Git & GitHub for DevOps Engineers

Task 1: Add a text file called version01.txt inside the Devops/Git/ with "This is first feature of our application" written inside. This should be in a branch coming from master, [hint try git checkout -b dev], switch to dev branch (Make sure your commit message will reflect as "Added new feature"). [Hint use your knowledge of creating branches and Git commit command]

version01.txt should reflect at local repo first followed by Remote repo for review. [Hint use your knowledge of Git push and git pull commands here]

Add new commit in dev branch after adding below mentioned content in Devops/Git/version01.txt: While writing the file make sure you write these lines

1st line>> This is the bug fix in development branch. Commit this with message " Added feature2 in development branch"

2nd line>> This is gadbad code. Commit this with message " Added feature3 in development branch"

3rd line>> This feature will gadbad everything from now. Commit with message " Added feature4 in development branch"

Restore the file to a previous version where the content should be "This is the bug fix in development branch" [Hint use git revert or reset according to your knowledge]

```
G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git push origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 414 bytes | 414.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/RishikeshOps/devops.git
 50404fd..1e48fe5 dev -> dev

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git log --oneline
1e48fe5 (HEAD -> dev, origin/dev) Added feature4 in development branch
50404fd Added feature3 in development branch
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 (origin/main, main) added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ cat version01.txt
This is first feature of our application
This is the bug fix in development branch
This is gadbad code
This feature will gadbad everything from now

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ |
```

dev devops / Git /

This branch is 3 commits ahead of main.

RishikeshOps Restore the file to a previous version

e6fa064 1 minute ago History

Day-01.txt

added Day-01.txt

4 hours ago

version01.txt

Restore the file to a previous version

1 minute ago

```

MINGW64:/c/Users/G15/Desktop/git/task/devops/Git

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git
$ git add version01.txt

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git
$ git commit -m "Restore the file to a previous version"
[dev e6fa064] Restore the file to a previous version
1 file changed, 1 deletion(-)

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git
$ git push origin dev
To https://github.com/RishikeshOps/devops.git
! [rejected]        dev -> dev (non-fast-forward)
error: failed to push some refs to 'https://github.com/RishikeshOps/devops.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git
$ git push -f origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 381 bytes | 381.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RishikeshOps/devops.git
+ 1e48fe5...e6fa064 dev -> dev (forced update)

```

```

MINGW64:/c/Users/G15/Desktop/git/task/devops/Git

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git add version01.txt

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git commit -m "Restore the file to a previous version"
[dev e6fa064] Restore the file to a previous version
1 file changed, 1 deletion(-)

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git push origin dev
To https://github.com/RishikeshOps/devops.git
! [rejected]        dev -> dev (non-fast-forward)
error: failed to push some refs to 'https://github.com/RishikeshOps/devops.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git push -f origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 381 bytes | 381.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RishikeshOps/devops.git
+ 1e48fe5...e6fa064 dev -> dev (forced update)

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$

```

Task 2:

Demonstrate the concept of branches with 2 or more branches with screenshot.

➔ In a version control system like Git, a branch is a separate line of development that allows multiple people to work on the same codebase simultaneously without interfering with each other. Each branch has

its own set of commits (i.e., changes to the code), and branches can be merged together to bring their changes together.

add some changes to dev branch and merge that branch in master

```
G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git push origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 414 bytes | 414.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/RishikeshOps/devops.git
50404fd..1e48fe5 dev -> dev

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git log --oneline
1e48fe5 (HEAD -> dev, origin/dev) Added feature4 in development branch
50404fd Added feature3 in development branch
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 (origin/main, main) added Day-01.txt
370a0a2 added day01 file

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ cat version01.txt
This is first feature of our application
This is the bug fix in development branch
This is gadbad code
This feature will gadbad everything from now

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ |
```

```
G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (main)
$ git merge dev
Updating 1ac93b6..e6fa064
Fast-forward
 Git/version01.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Git/version01.txt

G15@DESKTOP-0026P56 MINGW64 ~/Desktop/git/task/devops/Git (main)
$ git log --oneline
e6fa064 (HEAD -> main, origin/dev, prod, dev) Restore the file to a previous version
4e7f3bb Added feature2 in development branch
3e9c372 Added new feature
1ac93b6 (origin/main) added Day-01.txt
370a0a2 added day01 file
```

as a practice try git rebase too, see what difference you get.

➔ Git rebase is another way to integrate changes from one branch into another, similar to merge. However, the key difference between the two is how the changes are represented in the history of the branch.

When you merge two branches, a new commit is created that has two parents: one for the tip of the branch you are merging in and one for the current commit on the branch you are merging into. This creates a merge commit which has multiple parents, which shows that it is a merge commit.

On the other hand, when you rebase one branch onto another, the commits from the branch you are rebasing are "moved" to be based on the tip of the branch you are rebasing onto. This effectively replaces the original commits on the branch you are rebasing with new commits that have the same changes, but different SHA-1 identifiers,