

Lab Exercise 3- Working with Git Rebase

1. Initialize a Git repository

```
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % mkdir git-rebase-lab
mohdanas@Mohds-MacBook-Air DEVSECOPS_LAB % git init git-rebase-lab
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 /Users/mohdanas/Desktop/DEVSECOPS_LAB/git-rebase-lab/.git/
```

2. Create the main branch and make the initial commit:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % echo "Line 1 from main branch" > file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git add file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git commit -m "Initial commit: Add Line 1 from main branch"
[master (root-commit) 1f32e41] Initial commit: Add Line 1 from main branch
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

3. Create a new branch feature-branch:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

4. Add a new line to file.txt in feature-branch

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % echo "Line 2 from feature branch" >> file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git add file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git commit -m "Add Line 2 from feature branch"
[feature-branch (root-commit) c08476e] Add Line 2 from feature branch
1 file changed, 2 insertions(+)
create mode 100644 file.txt
```

5. Switch back to the main branch and add another line:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % git checkout master
Switched to branch 'master'
mohdanas@Mohds-MacBook-Air git-rebase-lab % echo "Line 3 from main branch" >> file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git add file.txt
mohdanas@Mohds-MacBook-Air git-rebase-lab % git commit -m "Add Line 3 from main branch"
[master 75e8767] Add Line 3 from main branch
1 file changed, 1 insertion(+)
```

2. Rebase feature-branch onto main

1. Switch to feature-branch

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % git checkout feature-branch
Already on 'feature-branch'
```

2. Rebase feature-branch onto main:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % git rebase master
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
error: could not apply e926a71... Add Line 2 from feature branch
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config set advice.mergeConflict false"
Could not apply e926a71... Add Line 2 from feature branch
```

Solved conflict.

3. Verify the Rebase

1. View the commit history:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % git log --oneline
75e8767 (HEAD, master) Add Line 3 from main branch
44a842b Initial commit: Add Line 1 from main branch
mohdanas@Mohds-MacBook-Air git-rebase-lab % git log --oneline --graph
* 75e8767 (HEAD, master) Add Line 3 from main branch
* 44a842b Initial commit: Add Line 1 from main branch
```

2. Check the contents of file.txt:

```
mohdanas@Mohds-MacBook-Air git-rebase-lab % cat file.txt
Line 1 from main branch
Line 3 from main branch
Line 2 from feature branch
```