

Lab Exercise 2- Working with Git Reset

Lab Exercise: Git Reset

This lab exercise will guide you through the usage of the git reset command in various scenarios. The git reset command is used to undo changes in the Git history, working directory, or staging area. There are three main modes: **soft**, **mixed**, and **hard**.

Objective

- Learn how to use git reset to modify the commit history, unstage files, or discard changes.
 - Understand the differences between --soft, --mixed, and --hard reset modes.
-

Prerequisites

1. Install Git on your system.
2. Set up a Git repository:

```
git init git-reset-lab
```

```
cd git-reset-lab
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp (Vbranch)
$ git init git-reset-lab
Initialized empty Git repository in C:/Users/admin/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab/.git/

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp (Vbranch)
$ cd git-reset-lab

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ |
```

Steps

1. Set Up the Repository

1. Create and commit an initial file:

```
echo "Line 1" > file.txt
```

```
git add file.txt
```

```
git commit -m "Initial commit: Add Line 1"
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ echo "DevOps" > Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git add Exp1.txt
warning: in the working copy of 'Exp1.txt', LF will be replaced by CRLF the next time Git touches it

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -m "Added content to Exp1.txt"
[master (root-commit) b023a9a] Added content to Exp1.txt
1 file changed, 1 insertion(+)
create mode 100644 Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ |
```

2. Add a second change:

```
echo "Line 2" >> file.txt
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ echo "DevOps Tools" >> Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ cat Exp1.txt
DevOps
DevOps Tools
```

```
git commit -am "Add Line 2"
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git add Exp1.txt
warning: in the working copy of 'Exp1.txt', LF will be replaced by CRLF the next
time Git touches it

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -m "Added line 2"
[master e3b5dd4] Added line 2
1 file changed, 1 insertion(+)

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ |
```

3. Add a third change:

```
echo "Line 3" >> file.txt
```

```
git commit -am "Add Line 3"
```

```

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ echo "DevOps Career" >> Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git add Exp1.txt
warning: in the working copy of 'Exp1.txt', LF will be replaced by CRLF the next
time Git touches it

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -am "Added third line"
[master 3abb783] Added third line
1 file changed, 1 insertion(+)

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)

```

4. Check the commit history:

```
git log --oneline
```

Example output:

```
c3f6b6b (HEAD -> main) Add Line 3
```

```
8b2a1c1 Add Line 2
```

```
4d5f8e9 Initial commit: Add Line 1
```

```

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git log --oneline
3abb783 (HEAD -> master) Added third line
e3b5dd4 Added line 2
b023a9a Added content to Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)

```

2. Use git reset --soft

This mode moves the HEAD pointer to an earlier commit but keeps the changes in the staging area.

1. Reset to the second commit:

```
git reset --soft HEAD~1
```

2. Check the commit history:

```
git log --oneline
```

Output:

```
8b2a1c1 (HEAD -> main) Add Line 2
```

```
4d5f8e9 Initial commit: Add Line 1
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git reset --soft HEAD~1

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git log --oneline
e3b5dd4 (HEAD -> master) Added line 2
b023a9a Added content to Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ |
```

3. Verify the staged changes:

```
git status
```

Output:

Changes to be committed:

modified: file.txt

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Expl.txt
```

4. If needed, re-commit the changes:

git commit -m "Recommit Line 3"

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -m "Recommited the line 3"
[master c95e805] Recommited the line 3
1 file changed, 1 insertion(+)
```

3. Use git reset --mixed

This mode moves the HEAD pointer and unstages the changes but keeps them in the working directory.

1. Reset to the first commit:

git reset --mixed HEAD~1

2. Check the commit history:

```
git log --oneline
```

Output:

4d5f8e9 (HEAD -> main) Initial commit: Add Line 1

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git reset --mixed HEAD~1
Unstaged changes after reset:
M      Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git log --oneline
e3b5dd4 (HEAD -> master) Added line 2
b023a9a Added content to Exp1.txt
```

3. Verify the changes in the working directory:

```
git status
```

Output:

Changes not staged for commit:

modified: file.txt

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Exp1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

3. If needed, stage and re-commit:

```
git add file.txt
```

```
git commit -m "Recommit Line 2 and Line 3"
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git add Exp1.txt
warning: in the working copy of 'Exp1.txt', LF will be replaced by CRLF the next
time Git touches it

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -m "Recommitted the line 2 and 3"
[master a063e4d] Recommitted the line 2 and 3
1 file changed, 1 insertion(+)
```

4. Use git reset --hard

This mode moves the HEAD pointer and discards all changes in the staging area and working directory.

1. Reset to the initial commit:

```
git reset --hard HEAD~1
```

2. Check the commit history:

```
git log --oneline
```

Output:

```
4d5f8e9 (HEAD -> main) Initial commit: Add Line 1
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git reset --hard HEAD~1
HEAD is now at e3b5dd4 Added line 2

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git log --oneline
e3b5dd4 (HEAD -> master) Added line 2
```

3. Verify the working directory:

```
cat file.txt
```

Output:

Line 1

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ cat Exp1.txt
DevOps
```

5. Use git reset with a Commit Hash

1. Add some changes for demonstration:

```
echo "Line 2" >> file.txt
```

```
git commit -am "Add Line 2"
```

```
echo "Line 3" >> file.txt
```

```
git commit -am "Add Line 3"
```

2. Get the commit hash for the initial commit:

```
git log --oneline
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git commit -am "Added another line"
warning: in the working copy of 'Exp1.txt', LF will be replaced by CRLF the next
time Git touches it
[master 63b89b7] Added another line
1 file changed, 1 insertion(+)

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git log --oneline
63b89b7 (HEAD -> master) Added another line
e3b5dd4 Added line 2
b023a9a Added content to Exp1.txt
```

3. Reset to the initial commit using the hash:

```
git reset --hard <commit-hash>
```

4. Verify the working directory and commit history:

```
git log --oneline
```

```
cat file.txt
```

```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ git reset --hard e3b5dd4
HEAD is now at e3b5dd4 Added line 2

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-reset-lab (master)
$ cat Exp1.txt
DevOps
```

Summary of Commands

Mode	Effect	Command Example
--soft	Moves HEAD, keeps changes staged.	git reset --soft HEAD~1
--mixed	Moves HEAD, unstages changes, keeps them in working dir.	git reset --mixed HEAD~1
--hard	Moves HEAD, discards all changes in staging and working dir.	git reset --hard HEAD~1