**Lab Exercise 2- Working with Git Reset**

**Lab Exercise: Git Reset**

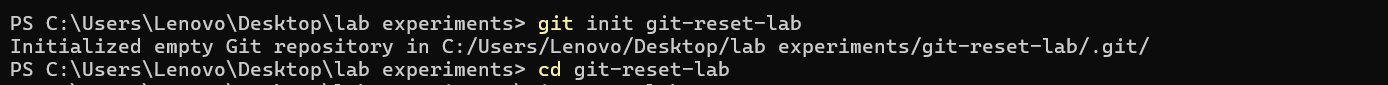
This lab exercise will guide you through the usage of the git reset command in various scenarios. The git reset command is used to undo changes in the Git history, working directory, or staging area. There are three main modes: **soft**, **mixed**, and **hard**.

**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**

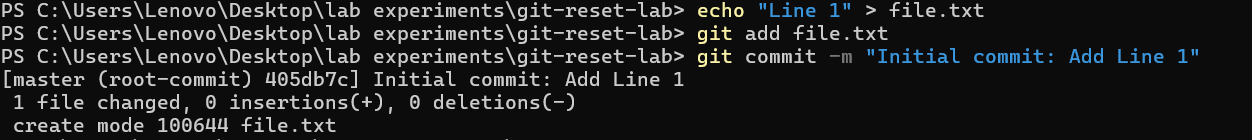
1. Install Git on your system.
2. Set up a Git repository:



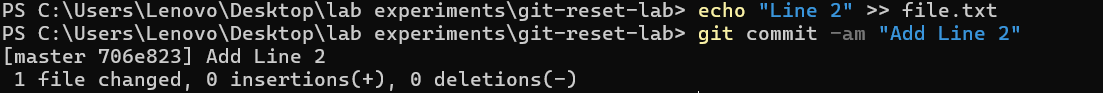
**Steps**

**1. Set Up the Repository**

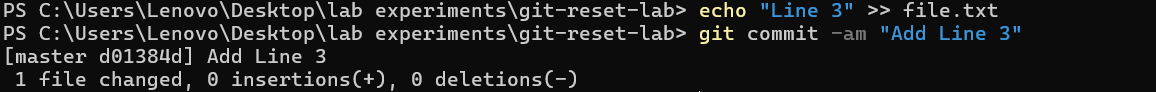
1. Create and commit an initial file:



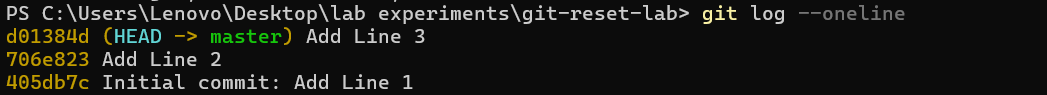
1. Add a second change:



1. Add a third change:



1. Check the commit history:



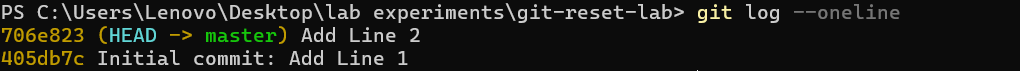
**2. Use git reset --soft**

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

1. Reset to the second commit:



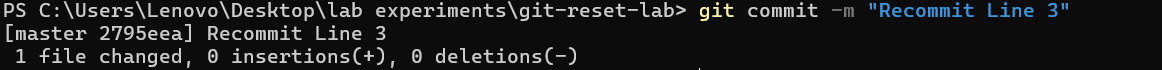
1. Check the commit history:



1. Verify the staged changes:

A black screen with white text

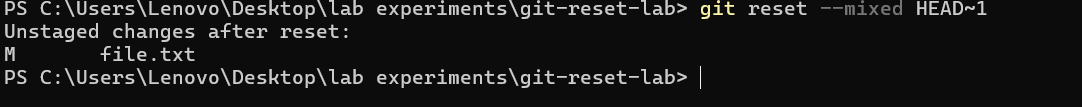
AI-generated content may be incorrect.

1. If needed, re-commit the changes: 

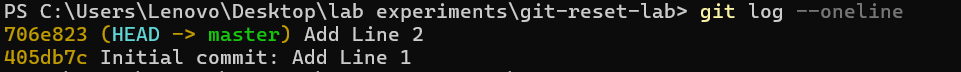
**3. Use git reset --mixed**

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

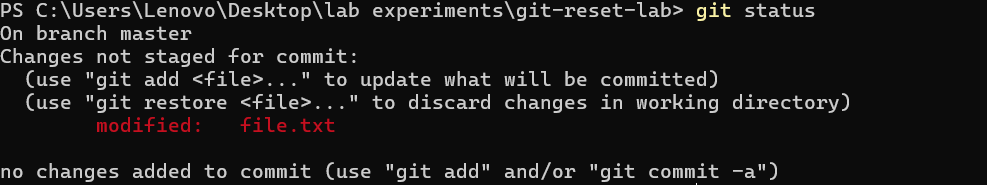
1. Reset to the first commit:



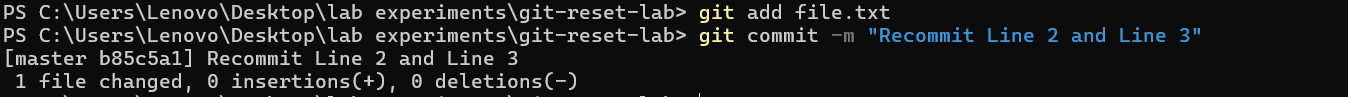
1. Check the commit history :



1. Verify the changes in the working directory:



1. If needed, stage and re-commit:



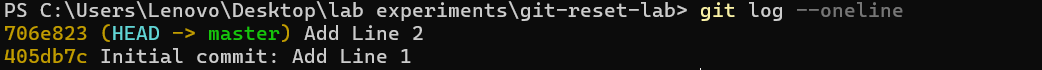
**4. Use git reset --hard**

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

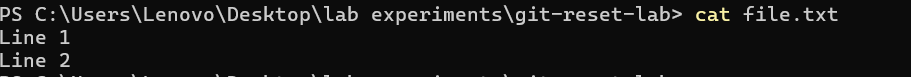
1. Reset to the initial commit:



1. Check the commit history:



1. Verify the working directory:



**5. Use git reset with a Commit Hash**

1. Add some changes for demonstration:

A screen shot of a computer screen

AI-generated content may be incorrect.

1. Get the commit hash for the initial commit:



1. Reset to the initial commit using the hash:



1. Verify the working directory and commit history:

A screen shot of a computer

AI-generated content may be incorrect.

**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 |