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

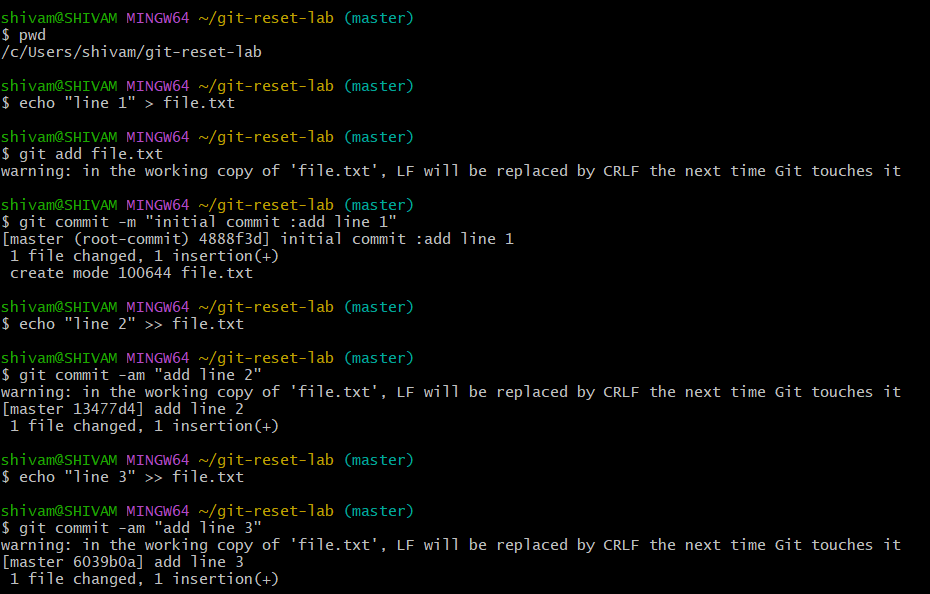
**Prerequisites**

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

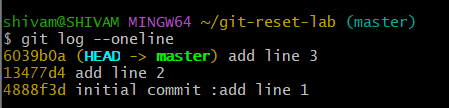
**Steps**

**1. Set Up the Repository**

1. Create and commit an initial file:
2. Add a second change:
3. Add a third change:
4. Check the commit history:



Example output:

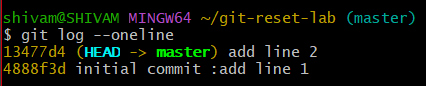


**2. Use git reset --soft**

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

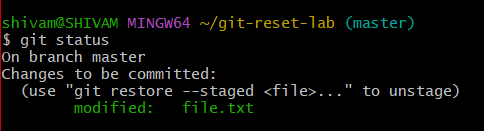
1. Reset to the second commit:
2. Check the commit history:

Output:



1. Verify the staged changes:

Output:



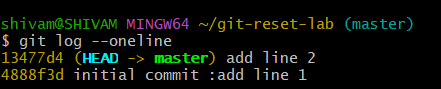
1. If needed, re-commit the changes:

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

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

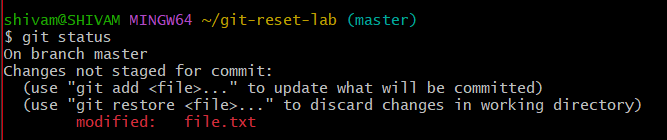
1. Reset to the first commit:
2. Check the commit history:

Output:



1. Verify the changes in the working directory:

Output:



1. If needed, stage and re-commit:

git add file.txt

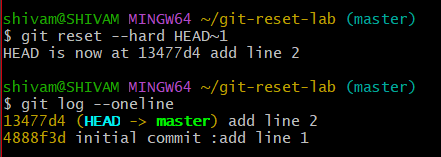
git commit -m "Recommit Line 2 and Line 3"

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

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

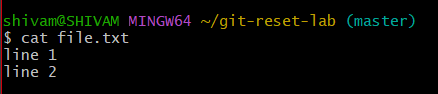
1. Reset to the initial commit:
2. Check the commit history:

Output:



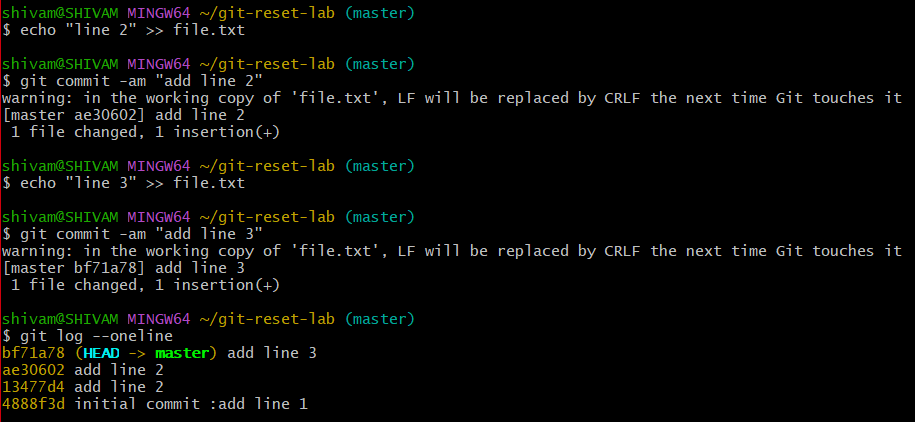
1. Verify the working directory:

Output:



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

1. Add some changes for demonstration:
2. Get the commit hash for the initial commit:



1. Reset to the initial commit using the hash:
2. Verify the working directory and commit history:

