

Lab Exercise 1- Working with Git Revert

Lab Exercise: Git Revert

This exercise will guide you through reverting changes in Git. The git revert command is used to create a new commit that undoes the changes introduced by a previous commit without modifying the history.

Objective

- Learn how to use git revert to undo changes from specific commits.
 - Practice handling merge conflicts during a revert.
-

Prerequisites

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

```
git init git-revert-lab
```

```
cd git-revert-lab
```

Steps

1. Set Up the Repository

1. Create a file:

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

2. Stage and commit the file:

```
git add file.txt  
  
git commit -m "Initial commit: Add file.txt with Line 1"
```

3. Add more changes:

```
echo "Line 2" >> file.txt  
  
git commit -am "Add Line 2 to file.txt"
```

4. Add another change:

```
echo "Line 3" >> file.txt  
  
git commit -am "Add Line 3 to file.txt"
```

5. Verify the commit history:

```
git log --oneline
```

Example output:

```
shagu@Shagun MINGW64 ~/Desktop/Notes/DevSecOps-Lab/devsecops-lab-1 (master)
$ git log --oneline
60ee011 (HEAD -> master) Add Line 3 to file.txt
9175c1f Add Line 2 to file.txt
9123e81 Initial commit: Add file.txt with Line 1
```

2. Revert the Last Commit

1. Revert the most recent commit:

```
git revert HEAD
```

2. Git will open a text editor for the commit message. Save and close the editor to complete the revert.
3. Verify the contents of file.txt:

```
cat file.txt
```

Output:

```
shagu@Shagun MINGW64 ~/Desktop/Notes/DevSecOps-Lab/devsecops-lab-1 (master)
$ cat file.txt
Line1
Line2
```

4. Check the commit history:

```
git log --oneline
```

Example output:

```
shagu@Shagun MINGW64 ~/Desktop/Notes/DevSecOps-Lab/devsecops-lab-1 (master)
$ git log --oneline
2b7bfa5 (HEAD -> master) Revert "Add Line 3 to file.txt"
60ee011 Add Line 3 to file.txt
9175c1f Add Line 2 to file.txt
9123e81 Initial commit: Add file.txt with Line 1
```

3. Handle Merge Conflicts During Revert

1. Modify file.txt:

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

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

2. Revert the second commit (8b2a1c1 again) to trigger a conflict:

```
git revert 8b2a1c1
```

3. Git will indicate a conflict. Resolve it:
 - Open file.txt and remove conflict markers.
 - Keep the desired lines.
4. Stage the resolved file:

```
git add file.txt
```

5. Complete the revert:

```
git revert --continue
```

Output:

```
shagu@Shagun MINGW64 ~/Desktop/Notes/DevSecOps-Lab/devsecops-lab-1 (master)
$ git log --oneline
feb8a3a (HEAD -> master) Revert "Add Line 2 to file.txt"
f93f370 Add Conflict Line
2b7bfa5 Revert "Add Line 3 to file.txt"
60ee011 Add Line 3 to file.txt
9175c1f Add Line 2 to file.txt
9123e81 Initial commit: Add file.txt with Line 1
```