Lab Exercise 3- Working with Git Rebase

Lab Exercise: Git Rebase

This exercise demonstrates the use of git rebase in a scenario where no conflicts occur.

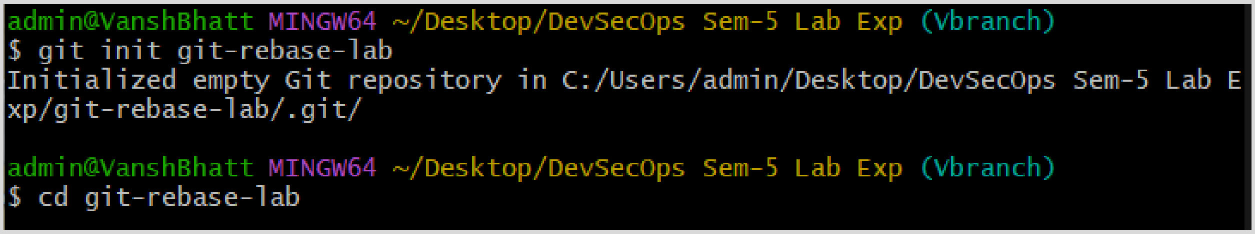
Objective

1. Learn how to rebase branches when there are no conflicting changes.
2. Understand the clean, linear history created by git rebase.

Prerequisites

1. Install Git on your system.
2. Initialize a Git repository:

```
git init git-rebase-lab

cd git-rebase-lab
```



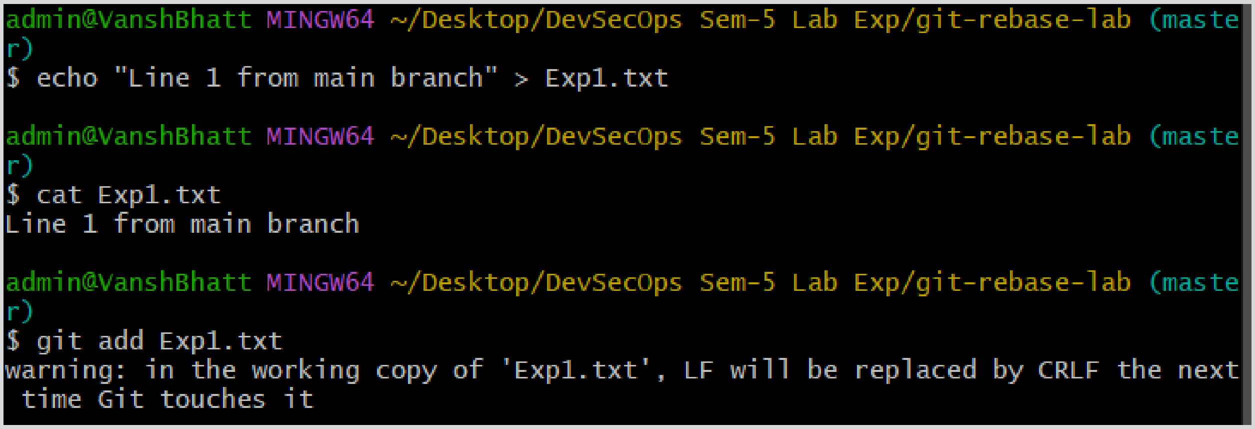
```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp (Vbranch)
$ git init git-rebase-lab
Initialized empty Git repository in C:/Users/admin/Desktop/DevSecOps Sem-5 Lab E
xp/git-rebase-lab/.git/

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

Step-by-Step Instructions

1. Set Up the Repository

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

```
echo "Line 1 from main branch" > file.txt

git add file.txt
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ echo "Line 1 from main branch" > Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ cat Exp1.txt
Line 1 from main branch

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

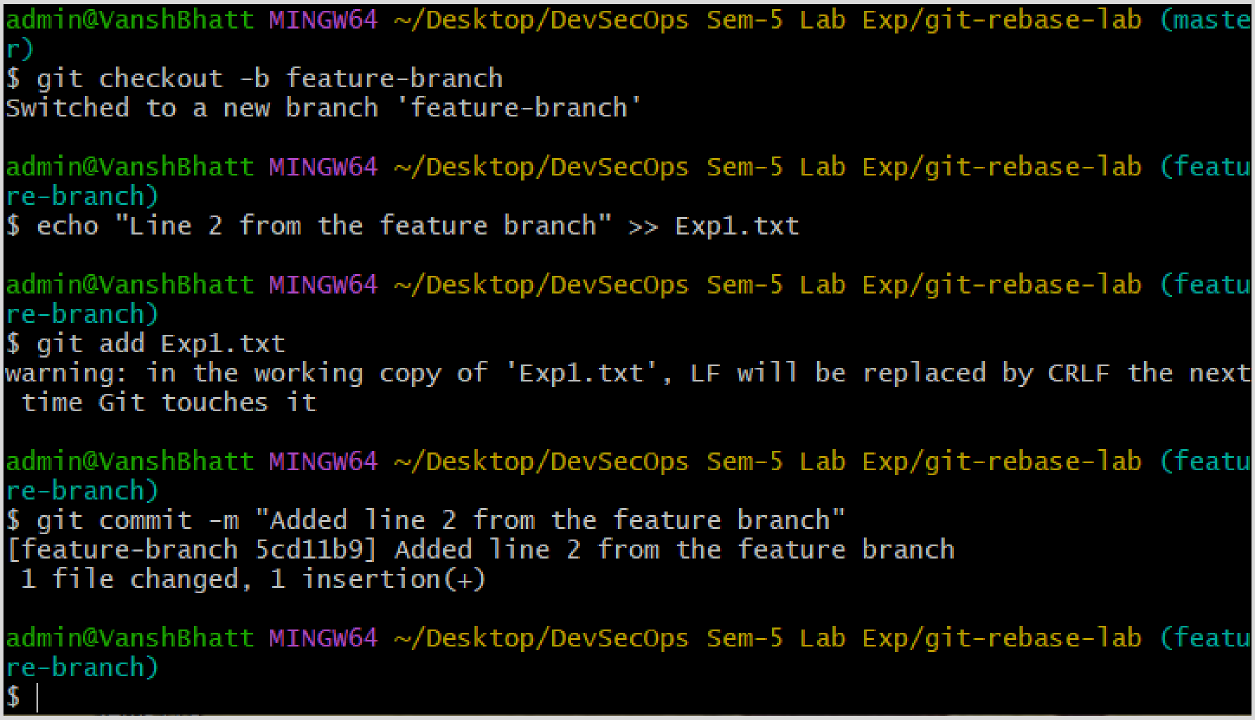
```
git commit -m "Initial commit: Add Line 1 from main branch"
```

2. Create a new branch feature-branch:

```
git checkout -b feature-branch
```

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

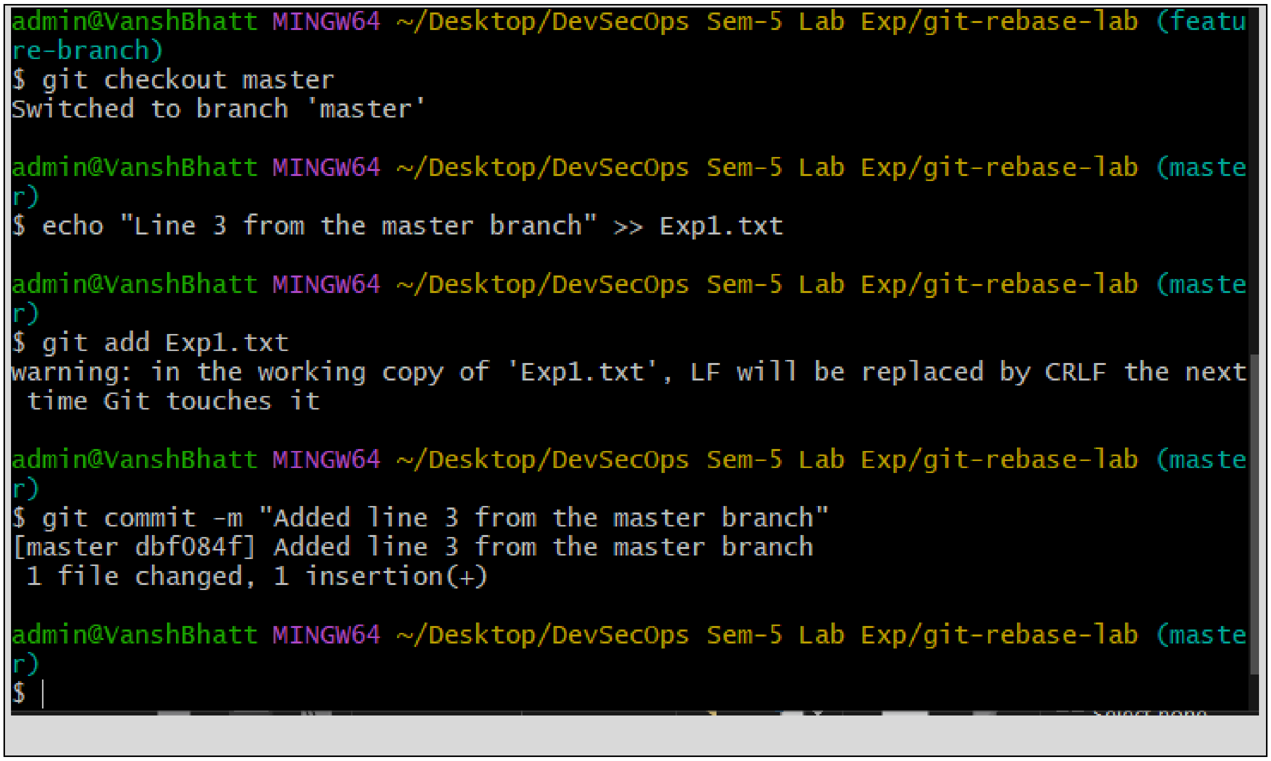
```
echo "Line 2 from feature branch" >> file.txt

git add file.txt
git commit -m "Add Line 2 from feature branch"
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch)
$ echo "Line 2 from the feature branch" >> Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch)
$ 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-rebase-lab (featu
re-branch)
$ git commit -m "Added line 2 from the feature branch"
[feature-branch 5cd11b9] Added line 2 from the feature branch
 1 file changed, 1 insertion(+)

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch)
$ |
```

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

```
git checkout main

echo "Line 3 from main branch" >> file.txt

git add file.txt

git commit -m "Add Line 3 from main branch"
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch)
$ git checkout master
Switched to branch 'master'

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ echo "Line 3 from the master branch" >> Exp1.txt

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ 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-rebase-lab (maste
r)
$ git commit -m "Added line 3 from the master branch"
[master dbf084f] Added line 3 from the master branch
 1 file changed, 1 insertion(+)

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

---

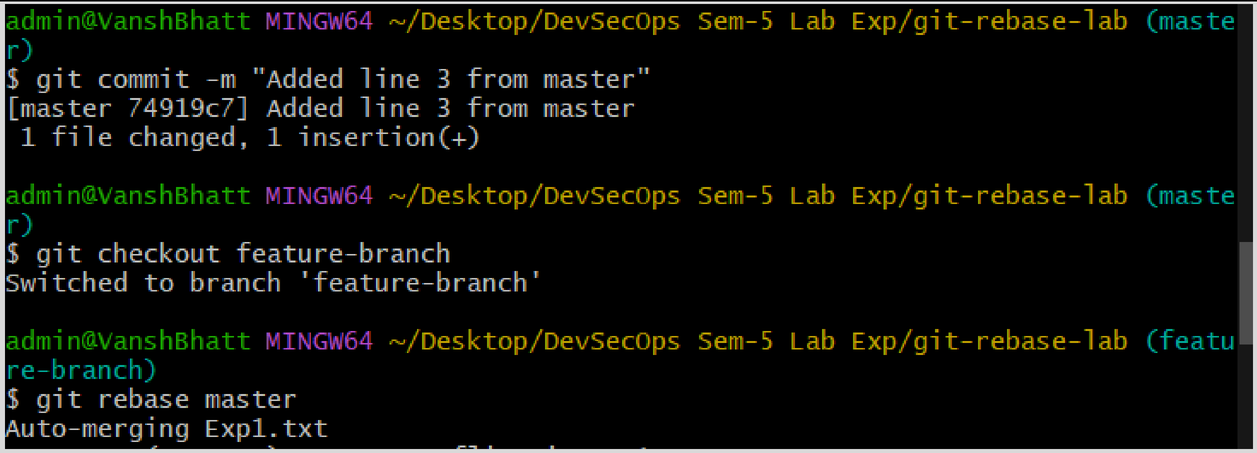
2. Rebase feature-branch onto main

1. Switch to feature-branch:

```
git checkout feature-branch
```

2. Rebase feature-branch onto main:

```
git rebase main
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ git commit -m "Added line 3 from master"
[master 74919c7] Added line 3 from master
 1 file changed, 1 insertion(+)

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (maste
r)
$ git checkout feature-branch
Switched to branch 'feature-branch'

admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch)
$ git rebase master
Auto-merging Exp1.txt
```

3. Git will replay the commit from feature-branch onto the main branch. Since there are no conflicts, the rebase completes automatically.

---

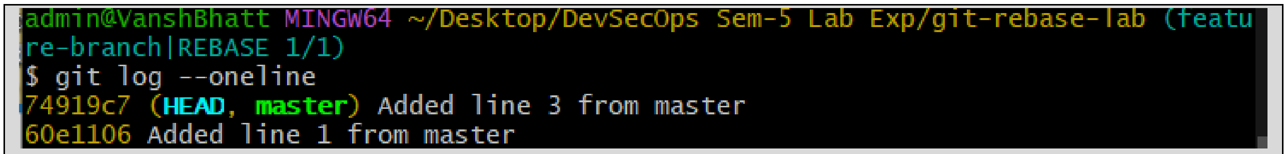
3. Verify the Rebase

1. View the commit history:

```
git log --oneline

git log --oneline --graph
```

Example output:

```
* Add Line 2 from feature branch

* Add Line 3 from main branch

* Initial commit: Add Line 1 from main branch
```



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch|REBASE 1/1)
$ git log --oneline
74919c7 (HEAD, master) Added line 3 from master
60e1106 Added line 1 from master
```

2. Check the contents of file.txt:

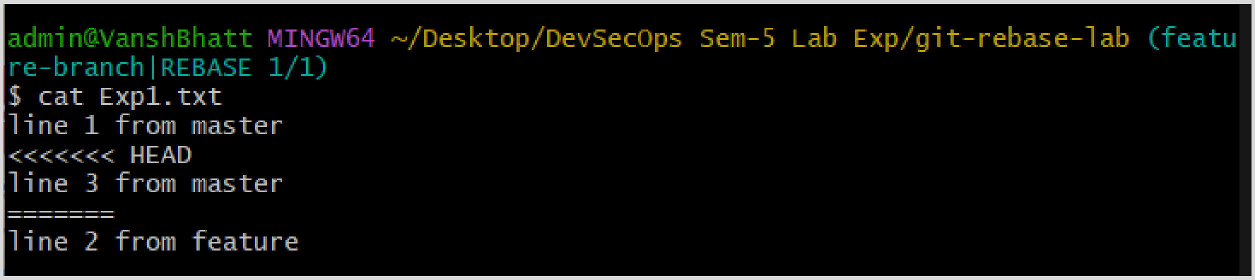
```
cat file.txt
```

Output:

Line 1 from main branch

Line 3 from main branch

Line 2 from feature branch



```
admin@VanshBhatt MINGW64 ~/Desktop/DevSecOps Sem-5 Lab Exp/git-rebase-lab (featu
re-branch|REBASE 1/1)
$ cat Exp1.txt
line 1 from master
<<<<<<< HEAD
line 3 from master
=======
line 2 from feature
```