



DevSecOps: Integrating Security in DevOps Practices

Submitted by:

Jaydev Swain
500124303
Batch 2 DevOps

Submitted to:

Dr. Hitesh Sharma

Lab Exercise 1 – Working With Git Revert

1. Lab Exercise: Git Revert

Command:

1. git init git-revert-lab
2. Cd git-revert-lab

```
jayde@ASUS-16x MINGW64 ~  
$ git init git-revert-lab  
Initialized empty Git repository in C:/Users/jayde/git-revert-lab/.git/
```

```
jayde@ASUS-16x MINGW64 ~  
$ cd git-revert-lab
```

```
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$
```

3. Echo "Line 1" > file.txt
4. Git add file.txt
5. Git commit -m "Initial commit: Add file.txt with Line 1"

```
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ echo "Line 1" > file.txt  
  
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ git add file.txt  
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
```

6. Echo "Line 2" >> file.txt
7. Git commit -am "Add file.txt with Line 2"

```
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ echo "Line 2" >> file.txt  
  
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ git commit -am "Add Line 2 to file.txt"  
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it  
[main 5b703d9] Add Line 2 to file.txt  
1 file changed, 1 insertion(+)
```

8. Echo "Line 3" >> file.txt
9. Git commit -am "Add Line 3 to file.txt"
10. Git log --oneline

```
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ echo "Line 3" >> file.txt  
  
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ git commit -am "Add Line 3 to file.txt"  
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it  
[main 35dfe4d] Add Line 3 to file.txt  
1 file changed, 1 insertion(+)  
  
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$ git log --oneline  
fatal: ambiguous argument '--oneline': unknown revision or path not in the working tree.  
Use '--' to separate paths from revisions, like this:  
'git <command> [<revision>...] -- [<file>...]'  
  
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)  
$
```

2. Revert the last Commit

Commands:

- ## 1. Git revert HEAD

```
MINGW64:/c/Users/jayde/git-revert-lab
Revert "Add Line 3 to file.txt"

This reverts commit 35dfe4d4f8daae4df54e6b22505e62349e4ad0ca.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Changes to be committed:
#   modified:   file.txt
#
#
#
#
#
```

- ## 1. Cat file.txt

```
[main e037d2b] Revert "Add Line 3 to file.txt"
1 file changed, 1 deletion(-)

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ cat file.txt
Line 1
Line 2

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$
```

- ## 2. Git log --oneline

```

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ git log -oneline
fatal: ambiguous argument '-oneline': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

```

3. Handle Merge Conflicts During Revert

Commands

1. `Echo "Conflict Line" >> file.txt`
2. `Git commit -am "Add Conflict Line"`

```

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ echo "Conflict Line" >> file.txt

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ git commit -am "Add Conflict Line"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main c98157f] Add Conflict Line
1 file changed, 1 insertion(+)

```

3. Git revert 5b703d9

```

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ git revert 5b703d9
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
error: could not revert 5b703d9... Add Line 2 to file.txt
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git revert --continue".
hint: You can instead skip this commit with "git revert --skip".
hint: To abort and get back to the state before "git revert",
hint: run "git revert --abort".
hint: Disable this message with "git config advice.mergeConflict false"
jayde@ASUS-16x MINGW64 ~/git-revert-lab (main|REVERTING)
$ |

```

4. Git add file.txt

5. Git revert -- continue

```

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main|REVERTING)
$ git revert --continue
U      file.txt
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main|REVERTING)
$

```

```

[main 50bddb2] Revert "Add Line 5 to file.txt"
1 file changed, 1 deletion(-)

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$ git log --oneline -n 3
cat file.txt
50bddb2 (HEAD -> main) Revert "Add Line 5 to file.txt"
c61eb84 Add Conflict Line
8c95e0e =====
Line 1
Line 2
Conflict Line

jayde@ASUS-16x MINGW64 ~/git-revert-lab (main)
$

```

Lab Exercise 2 – Working with Git Reset

Lab Exercise: Git Reset

Commands:

1. Git init git-reset-lab
2. Cd git -reset-lab
3. Echo "Line 1" >file.txt

```
MINGW64/c:/Users/jayde/git-reset-lab
jayde@ASUS-16x MINGW64 ~
$ mkdir git-reset-lab
cd git-reset-lab
git init

Initialized empty Git repository in C:/Users/jayde/git-reset-lab/.git/
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$
```

4. Git add file.txt
5. Git commit -m "Initial commit: Add Line 1"

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ echo "Line 1" > file.txt
git add file.txt
git commit -m "Initial commit: Add Line 1"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main (root-commit) 63a717d] Initial commit: Add Line 1
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

6. Echo "Line 2" >> file.txt
7. Git commit -am "Add Line 2"

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ echo "Line 2" >> file.txt
git commit -am "Add Line 2"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main b9e0f09] Add Line 2
1 file changed, 1 insertion(+)
```

8. Git log --oneline

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ echo "Line 3" >> file.txt
git commit -am "Add Line 3"
warning: in the working copy of 'file.txt', LF will
[main 1564bf6] Add Line 3
1 file changed, 1 insertion(+)

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git log --oneline
1564bf6 (HEAD -> main) Add Line 3
b9e0f09 Add Line 2
63a717d Initial commit: Add Line 1
```

Use Git Reset --Soft

Commands:

1. git reset --soft HEAD~1

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git reset --soft HEAD~1
```

2. Git log --oneline

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git log --oneline
b9e0f09 (HEAD -> main) Add Line 2
63a717d Initial commit: Add Line 1
```

3. Git status
4. Git commit -m "Reconnect Line 3"

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.txt
```

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git commit -m "Recommit Line 3"
[main 71c813b] Recommit Line 3
1 file changed, 2 insertions(+)
```

Use git reset --mixed

Commands:

1. Git reset --mixed HEAD~1
2. Git log --oneline


```
dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git reset --mixed HEAD~1
Unstaged changes after reset:
M   file.txt

dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git log --oneline
cc3fadb (HEAD -> main) Add Line 2
3235bd0 Initial commit: Add Line 1
40985fc Reapply "Add Line 3 to file.txt"
c9736e6 Revert "Add Line 3 to file.txt"
e669055 Add Line 3 to file.txt
0fffb951 Add Line 2 to file.txt
5b536f0 Initial commit: Add file.txt with Line 1
```

3. Git status

```
dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git status
On branch main
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:   file.txt
```

4. Git add file.txt

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

dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
```

5. Git commit -m "Reconnect Line 2 and Line 3"

```
dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git commit -m "Recommit Line 2 and Line 3"
[main d635918] Recommit Line 2 and Line 3
 1 file changed, 1 insertion(+)
```

Use Git Reset –Hard

Commands

1. git reset – hard HEAD~1
2. git log –online
3. cat file.txt

```
dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git reset --hard HEAD~1
HEAD is now at cc3fadb Add Line 2

dell@abhinav MINGW64 ~/git-reset-lab (main)
$ git log --oneline
cc3fadb (HEAD -> main) Add Line 2
3235bd0 Initial commit: Add Line 1
40985fc Reapply "Add Line 3 to file.txt"
c9736e6 Revert "Add Line 3 to file.txt"
e669055 Add Line 3 to file.txt
0ffb951 Add Line 2 to file.txt
5b536f0 Initial commit: Add file.txt with Line 1

dell@abhinav MINGW64 ~/git-reset-lab (main)
$ cat file.txt
Line 1
Line 2
```

Use Git Reset with a Commit Hash Commands:

1. echo "Line 2 ">> file.txt
2. Git commit -am "Add Line 2"
3. Echo "Line 3" >> file.txt
4. Git commit -am "Add Line 3"
5. Git log --oneline

6. Git reset --hard <commit-dhash>

```
MINGW64:/c/Users/jayde/git-reset-lab
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ echo "Line 2" >> file.txt
git commit -am "Add Line 2"

echo "Line 3" >> file.txt
git commit -am "Add Line 3"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main 715d97c] Add Line 2
1 file changed, 1 insertion(+)
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main a9751a9] Add Line 3
1 file changed, 1 insertion(+)
```

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git log --oneline
a9751a9 (HEAD -> main) Add Line 3
715d97c Add Line 2
71c813b Recommit Line 3
b9e0f09 Add Line 2
63a717d Initial commit: Add Line 1

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git reset --hard 63a717d
HEAD is now at 63a717d Initial commit: Add Line 1

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git log --oneline
63a717d (HEAD -> main) Initial commit: Add Line 1

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ 63a717d (HEAD -> main) Initial commit: Add Line 1
bash: syntax error near unexpected token `HEAD'

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ git reset --hard 63a717d
HEAD is now at 63a717d Initial commit: Add Line 1

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$
```

Lab Exercise 3 – Working with Git Rebase

Commands:

1. git init git-rebase-lab
2. cd git-rebase-lab
3. echo "Line 1 from main branch" > file.txt

```

MINGW64:/c/Users/jayde/git-reset-lab/git-rebase-lab

jayde@ASUS-16x MINGW64 ~/git-reset-lab (main)
$ mkdir git-rebase-lab
cd git-rebase-lab
git init
Initialized empty Git repository in C:/Users/jayde/git-reset-lab/git-rebase-lab/.git/

```

4. git add file.txt
5. git checkout -b feature-branch
6. echo "Line 2 from feature branch" >> file.txt

```

jayde@ASUS-16x MINGW64 ~/git-reset-lab/git-rebase-lab (main)
$ echo "Line 1 from main branch" > file.txt
git add file.txt
git commit -m "Initial commit: Add Line 1 from main branch"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[main (root-commit) 833d215] Initial commit: Add Line 1 from main branch
1 file changed, 1 insertion(+)
create mode 100644 file.txt

```

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

```

$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

```

9. git checkout main
10. echo "Line 3 from main branch" >> file.txt

```

dell@abhinav MINGW64 ~/git-rebase-lab (feature-branch)
$ git checkout master
error: pathspec 'master' did not match any file(s) known to git

dell@abhinav MINGW64 ~/git-rebase-lab (feature-branch)
$ echo "Line 3 from main branch" >> file.txt

dell@abhinav MINGW64 ~/git-rebase-lab (feature-branch)
$ git checkout feature-branch
M      file.txt
Already on 'feature-branch'

```

11. git add file.txt
12. git commit -m "Add Line 3 from main branch"

```

$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

```

Rebase Feature Branch Onto Main

Commands:

1. git checkout feature-branch
2. git rebase main

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab/git-rebase-lab (feature-branch)
$ git log --oneline --graph
cat file.txt
* c229303 (HEAD -> feature-branch) Add Line 2 from feature branch
* eec62a8 (main) Add Line 3 from main branch
* 833d215 Initial commit: Add Line 1 from main branch
Line 1 from main branch
Line 3 from main branch
Line 2 from feature branch
```

3. git log --oneline
4. git log --oneline --graph
5. cat file.txt

```
jayde@ASUS-16x MINGW64 ~/git-reset-lab/git-rebase-lab (feature-branch)
$ git checkout main
git merge feature-branch
Switched to branch 'main'
Updating eec62a8..c229303
Fast-forward
 file.txt | 5 +++++
 1 file changed, 5 insertions(+)

jayde@ASUS-16x MINGW64 ~/git-reset-lab/git-rebase-lab (main)
$ |
```