**Technical Report - Git Fundamentals**

Author: Om Luitel

9th September, 2023

In this comprehensive report, I have embarked on elucidating foundational Git operations, encompassing:

1. **Understanding the Mechanics of git init**: This segment delves into the inner workings of the `git init` command, which is responsible for initializing a new Git repository.
2. **Staging Files and Monitoring Status via Terminal**: This section delves deeply into the process of staging files for commits and monitoring the status of files within a Git repository through command-line interfaces.
3. **Committing Changes in Git**: Within this section, we delve into the intricacies of committing changes in Git, underscoring the significance of commit messages and the principles of version control.
4. **Exploring Git Logs and Deciphering Commit IDs**: This segment furnishes valuable insights into examining Git logs to review commit histories, along with a comprehensive understanding of the significance of commit IDs or SHA-1 hashes.
5. **Snapshotting and Temporal Navigation in Git**: Finally, we dissect the concept of creating snapshots of your project's states and expound upon how Git facilitates temporal navigation by enabling the revisitation of various commit points.

This technical exploration aims to equip you with a robust understanding of Git's fundamental operations, enabling effective version control and collaborative software development.

Steps:

1. Establish a fresh directory named "myapp03."
2. Access the "myapp03" directory by executing the "cd" command.
3. Initialize a Git repository within this directory using "git init."

```
om@om-linu:~/projects/git-project$ mkdir myapp03
om@om-linu:~/projects/git-project$ cd myapp03
om@om-linu:~/projects/git-project/myapp03$ git init
hint: Using 'master' as the name for the initial branch. This default br
anch name
hint: is subject to change. To configure the initial branch name to use
in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this com
mand:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/om/projects/git-project/myapp0
3/.git/
om@om-linu:~/projects/git-project/myapp03$
```

4. file structure after git init

```
om@om-linu:~/projects/git-project/myapp03$ ls -al
total 12
drwxrwxr-x 3 om om 4096 Sep  9 15:37 .
drwxrwxr-x 5 om om 4096 Sep  9 15:35 ..
drwxrwxr-x 7 om om 4096 Sep  9 15:37 .git
om@om-linu:~/projects/git-project/myapp03$
```

5. Create a file called "one.txt," add it to the staging area, and commit it with the comment "Feature One." → view git status on each steps.

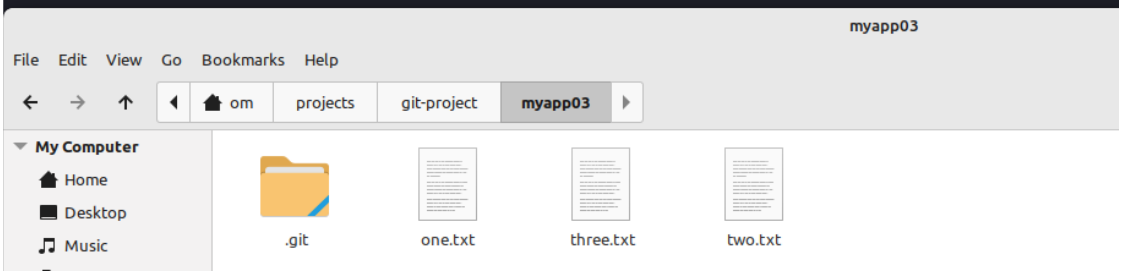
```
om@om-linu:~/projects/git-project/myapp03$ touch one.txt
om@om-linu:~/projects/git-project/myapp03$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        one.txt

nothing added to commit but untracked files present (use "git add" to tr
ack)
om@om-linu:~/projects/git-project/myapp03$ git add one.txt
om@om-linu:~/projects/git-project/myapp03$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   one.txt

om@om-linu:~/projects/git-project/myapp03$ git commit -m "Feature One" o
ne.txt
[master (root-commit) 18e28c8] Feature One
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 one.txt
om@om-linu:~/projects/git-project/myapp03$ git status
On branch master
nothing to commit, working tree clean
om@om-linu:~/projects/git-project/myapp03$
```

6. Follow same process, create another file called two.txt, commit with “Feature Two”

7. Follow same process, create another file called three.txt, commit with “Feature Three”

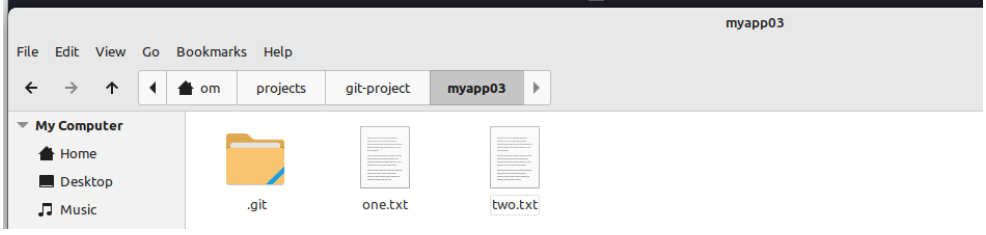
8. Review the Git logs, which should look similar to the screenshot below.

```
om@om-linu:~/projects/git-project/myapp03$ git log
commit 8e8eb4d3a9cac1f226d716ae437f118bd743219e (HEAD -> master)
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:55:03 2023 +1000

    Feature Three

commit bc5a94e379a750df9d11b3843f66f8b355c8e990
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:54:14 2023 +1000

    Feature Two

commit 18e28c8fd6cd5d05644e483e923137d531db28a4
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:45:58 2023 +1000

    Feature One
om@om-linu:~/projects/git-project/myapp03$
```




9. I am now reverting changes to the second commit. If I switch or checkout to 'bc5a94e379a750d,' I will have only two files, 'one.txt' and 'two.txt,' in the 'myapp03' directory.

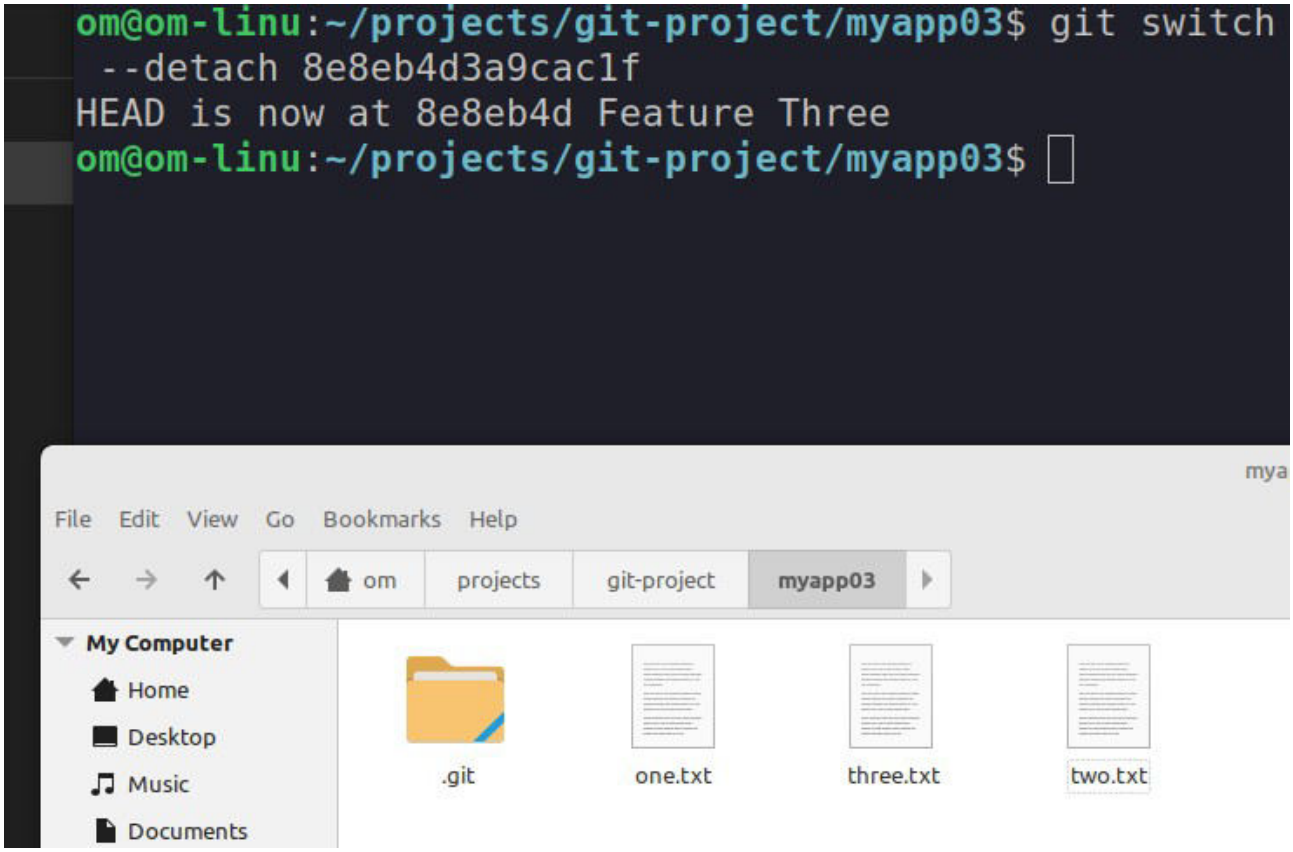
```
om@om-linu:~/projects/git-project/myapp03$ git log
commit 8e8eb4d3a9cac1f226d716ae437f118bd743219e (HEAD -> master)
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:55:03 2023 +1000

    Feature Three

commit bc5a94e379a750df9d11b3843f66f8b355c8e990
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:54:14 2023 +1000

    Feature Two

commit 18e28c8fd6cd5d05644e483e923137d531db28a4
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:45:58 2023 +1000

    Feature One
om@om-linu:~/projects/git-project/myapp03$ git switch --detach bc5a94e37
9a750d
HEAD is now at bc5a94e Feature Two
om@om-linu:~/projects/git-project/myapp03$
```



10. Now check git log, and it will just show you 2 commits, however if you do git log –all it will show you entire commits logs.

```
om@om-linu:~/projects/git-project/myapp03$ git log
commit bc5a94e379a750df9d11b3843f66f8b355c8e990 (HEAD)
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:54:14 2023 +1000

    Feature Two

commit 18e28c8fd6cd5d05644e483e923137d531db28a4
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:45:58 2023 +1000

    Feature One
om@om-linu:~/projects/git-project/myapp03$ git log --all
commit 8e8eb4d3a9cac1f226d716ae437f118bd743219e (master)
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:55:03 2023 +1000

    Feature Three

commit bc5a94e379a750df9d11b3843f66f8b355c8e990 (HEAD)
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:54:14 2023 +1000

    Feature Two

commit 18e28c8fd6cd5d05644e483e923137d531db28a4
Author: omluitel <omprakashluitelssm@gmail.com>
Date:   Sat Sep 9 15:45:58 2023 +1000

    Feature One
om@om-linu:~/projects/git-project/myapp03$
```

11. Now I am going back to feature 3 commit. | reverting changes




**Please note that Git checkout and switch operate identically. However, in this demonstration, the switch command was employed to revert changes. It's essential to emphasize that this approach is intended solely for demonstration purposes and is not recommended for use in a production environment.**