

Module 2: Git Assignment - 2

1. Create a Git working directory with feature1.txt and feature2.txt in the master branch.

Solution:-

```
$ sudo su  
# apt update -y
```

```
ubuntu@ip-172-31-81-98:~$ sudo su  
root@ip-172-31-81-98:/home/ubuntu# apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
9 kB]
```

```
# apt upgrade -y  
# apt install git  
# which git  
# git --version
```

```
root@ip-172-31-81-98:/home/ubuntu# apt upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
#  
# An OpenSSL vulnerability has recently been fixed with USN-6188-1 & 6119-1:  
# CVE-2023-2650: possible DoS translating ASN.1 object identifiers.  
# Ensure you have updated the package to its latest version.  
#  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@ip-172-31-81-98:/home/ubuntu# apt install git  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
git is already the newest version (1:2.34.1-1ubuntu1.9).  
git set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@ip-172-31-81-98:/home/ubuntu# which git  
/usr/bin/git  
root@ip-172-31-81-98:/home/ubuntu# git --version  
git version 2.34.1
```

```
# git config --global user.name "Amit Tiwari"
# git config --global user.email "redhat.amitiwari@gmail.com"
# git config --list
# mkdir gitdir
```

```
root@ip-172-31-81-98:/home/ubuntu# git config --global user.name "Amit Tiwari"
root@ip-172-31-81-98:/home/ubuntu# git config --global user.email "redhat.amitiwari@gmail.com"
root@ip-172-31-81-98:/home/ubuntu# git config --list
user.name="Amit
user.email="redhat.amitiwari@gmail.com"
root@ip-172-31-81-98:/home/ubuntu# mkdir gitdir
```

```
# cd gitdir
# git init
# ls -al
```

```
root@ip-172-31-81-98:/home/ubuntu# cd gitdir
root@ip-172-31-81-98:/home/ubuntu/gitdir# git init
hint: Using 'master' as the name for the initial branch. This default branch 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 command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/gitdir/.git/
root@ip-172-31-81-98:/home/ubuntu/gitdir# ls -al
total 12
drwxr-xr-x 3 root  root  4096 Jul 17 04:22 .
drwxr-x-- 5 ubuntu ubuntu 4096 Jul 17 04:22 ..
drwxr-xr-x 7 root  root  4096 Jul 17 04:22 .git
```

```
# git status
# touch feature1.txt feature2.txt
# git add .
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
root@ip-172-31-81-98:/home/ubuntu/gitdir# touch feature1.txt feature2.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# git add .
```

```
# git commit -m "My first commit in master."  
# git status
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git commit -m "My first commit in master."  
[master (root-commit) 1b2fa0f] My first commit in master.  
 2 files changed, 0 insertions(+), 0 deletions(-)  
 create mode 100644 feature1.txt  
 create mode 100644 feature2.txt  
root@ip-172-31-81-98:/home/ubuntu/gitdir# git status  
On branch master  
nothing to commit, working tree clean
```

2. Create 3 branches develop, feature1 and feature2.

Solution:-

```
# git branch  
# git branch develop && git branch feature1 && git branch feature2  
# git branch
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch  
* master  
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch develop && git branch feature1 && git branch feature2  
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch  
 develop  
 feature1  
 feature2  
* master
```

3. In develop branch create develop.txt, do not stage or commit it.

Solution:-

```
# git checkout develop
# git branch
# touch develop.txt
# vi develop.txt
l/p:- This file to be stash.
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git checkout develop
Already on 'develop'
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch
* develop
  feature1
  feature2
  master
root@ip-172-31-81-98:/home/ubuntu/gitdir# vi develop.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# cat develop.txt
This file to be stash.
```

4. Stash this file and check out to feature1 branch.

Solution:-

```
# git stash push -u -m "This is first stash without add and commit
the file." develop.txt
# git stash list
# git checkout feature1
# git branch
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git stash push -u -m "This is first st
ash without add and commit the file." develop.txt
Saved working directory and index state On develop: This is first stash without
add and commit the file.
root@ip-172-31-81-98:/home/ubuntu/gitdir# git stash list
stash@{0}: On develop: This is first stash without add and commit the file.
stash@{1}: On develop: "This
root@ip-172-31-81-98:/home/ubuntu/gitdir# git checkout feature1
Switched to branch 'feature1'
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch
  develop
* feature1
  feature2
  master
```

5. Create new.txt file in feature1 branch, stage and commit this file.

Solution:-

```
# touch new.txt
# git add new.txt
# git commit -m "This file new.txt is commit to feature1 branch."
new.txt
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# touch new.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# git add new.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# git commit -m "This file new.txt is co
mmmit to feature1 branch." new.txt
[feature1 47b645c] This file new.txt is commit to feature1 branch.
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 new.txt
```

6. Checkout to develop, unstash this file and commit.

Solution:-

```
# git branch
# git checkout develop
# git branch
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch
develop
* feature1
feature2
master
root@ip-172-31-81-98:/home/ubuntu/gitdir# git checkout develop
Switched to branch 'develop'
root@ip-172-31-81-98:/home/ubuntu/gitdir# git branch
* develop
feature1
feature2
master
```



```
# git stash list
# git stash apply stash@{0}
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git stash list
stash@{0}: On develop: This is first stash without add and commit the file.
stash@{1}: On develop: "This
root@ip-172-31-81-98:/home/ubuntu/gitdir# git stash apply stash@{0}
Already up to date.
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        develop.txt

nothing added to commit but untracked files present (use "git add" to track)
root@ip-172-31-81-98:/home/ubuntu/gitdir# cat develop.txt
This file to be stash.
```

```
# git add develop.txt
# git commit -m "This is first commit after unstash" develop.txt
```

```
root@ip-172-31-81-98:/home/ubuntu/gitdir# git add develop.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# git commit -m "This is first commit af
ter unstash" develop.txt
[develop c6bc273] This is first commit after unstash
 1 file changed, 1 insertion(+)
 create mode 100644 develop.txt
root@ip-172-31-81-98:/home/ubuntu/gitdir# git status
On branch develop
nothing to commit, working tree clean
```

7. Please submit all the Git commands used to do the above steps.

Solution:-

```
$ sudo su
# apt update -y
# apt upgrade -y
# apt install git
# which git
# git --version
# git config --global user.name "Amit Tiwari"
# git config --global user.email "redhat.amitiwari@gmail.com"
# git config --list
# mkdir gitdir
# cd gitdir
# git init
```

```
# ls -al
# git status
# touch feature1.txt feature2.txt
# git add .
# git commit -m "My first commit in master."
# git status
# git branch
# git branch develop && git branch feature1 && git branch
feature2
# git branch
# git checkout develop
# git branch
# touch develop.txt
# vi develop.txt
l/p:- This file to be stash.
# git stash push -u -m "This is first stash without add and commit
the file." develop.txt
# git stash list
# git checkout feature1
# git branch
# touch new.txt
# git add new.txt
# git commit -m "This file new.txt is commit to feature1 branch."
new.txt
# git branch
# git checkout develop
# git branch
# git stash list
# git stash apply stash@{0}
# git add develop.txt
# git commit -m "This is first commit after unstash" develop.txt
```