

# DevOps { PW Assignment }

## Linux Assignment Questions and Answer

### 1) Tasks to Be Performed:

#### 1. Based on what you have learnt in the class, do the following steps:

- a. Create a new folder
- b. Put the following files in the folder → Code.txt Log.txt Output.txt
- c. Stage the Code.txt and Output.txt files
- d. Commit them
- e. And finally push them to GitHub

Ans:-

```
SUMIT@Sumit-Dhal MINGW64 ~ (main)
$ mkdir my-project
$ cd my-project/
$ cd my-project/my-to-do-cicd/
SUMIT@Sumit-Dhal MINGW64 ~ (main)
$ cd my-project/my-to-do-cicd/
SUMIT@Sumit-Dhal MINGW64 ~ (main)
$ cd my-project/
SUMIT@Sumit-Dhal MINGW64 ~ (my-project (main))
$ touch Code.txt Log.txt Output.txt
SUMIT@Sumit-Dhal MINGW64 ~ (my-project (main))
$ ls
Code.txt  Log.txt  Output.txt
SUMIT@Sumit-Dhal MINGW64 ~ (my-project (main))
$ git init
Initialized empty Git repository in C:/Users/SUMIT/my-project/.git/
SUMIT@Sumit-Dhal MINGW64 ~/my-project (master)
$ git add Code.txt Output.txt
SUMIT@Sumit-Dhal MINGW64 ~/my-project (master)
$ git commit -m "add Code.txt and Output.txt"
[master (root-commit) ce73efaf] add Code.txt and Output.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Code.txt
create mode 100644 Output.txt
SUMIT@Sumit-Dhal MINGW64 ~/my-project (master)
$ git remote add origin https://github.com/ssvillan/Pw_task.git
SUMIT@Sumit-Dhal MINGW64 ~/my-project (master)
$ git branch -M master
SUMIT@Sumit-Dhal MINGW64 ~/my-project (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 236 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

#### 2. Please share the commands for the above points

```
mkdir my-project

cd my-project/

touch Code.txt Log.txt Output.txt

ls
Code.txt  Log.txt  Output.txt

git init
Initialized empty Git repository in C:/Users/SUMIT/my-project/.git/

git add Code.txt Output.txt

git commit -m "add Code.txt and Output.txt"
[master (root-commit) ce73efaf] add Code.txt and Output.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Code.txt
create mode 100644 Output.txt

git remote add origin https://github.com/ssvillan/Pw\_task.git

git branch -M master

git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
```

```

Writing objects: 100% (3/3), 236 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ssvillan/Pw_task.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

## 2) Tasks to Be Performed:

1. Create a Git working directory with feature1.txt and feature2.txt in the master branch
2. Create 3 branches develop, feature1 and feature2
3. In develop branch create develop.txt, do not stage or commit it
4. Stash this file and check out to feature1 branch
5. Create new.txt file in feature1 branch, stage and commit this file
6. Checkout to develop, unstash this file and commit

Ans:-

```

SUMIT@Sumit-Dhal MINGW64 ~ (main)
$ mkdir git-demo
SUMIT@Sumit-Dhal MINGW64 ~ (main)
$ cd git-demo
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (main)
$ git init
Initialized empty Git repository in C:/Users/SUMIT/git-demo/.git/
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ touch feature1.txt feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ ls
feature1.txt  feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git add feature1.txt feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git commit -m "Add feature1.txt and feature2.txt to master"
[master (root-commit) 088ad5e] Add feature1.txt and feature2.txt to master
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature1.txt
 create mode 100644 feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git branch develop
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git branch feature1
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git branch feature2
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (master)
$ git checkout develop
Switched to branch 'develop'
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ touch develop.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ ls
develop.txt  feature1.txt  feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ git stash push -m "Stash develop.txt"
No local changes to save

SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ ls
develop.txt  feature1.txt  feature2.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ git stash push -m "Stash develop.txt"
git checkout feature1
No local changes to save
Switched to branch 'feature1'
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (Feature1)
$ 
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (Feature1)
$ touch new.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (Feature1)
$ git add new.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (Feature1)
$ git commit -m "add new.txt in feature1 branch"
[feature1 b4c2d05] add new.txt in feature1 branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (Feature1)
$ git checkout develop
Switched to branch 'develop'
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ git stash pop
No stash entries found.
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ git add develop.txt
SUMIT@Sumit-Dhal MINGW64 ~/git-demo (develop)
$ git commit -m "add develop.txt after unstashing"
[develop 0ce2e4b] add develop.txt after unstashing
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 develop.txt

```

**7. Please submit all the Git commands used to do the above step>**

```
mkdir git-demo
cd git-demo
git init
Initialized empty Git repository in C:/Users/SUMIT/git-demo/.git/
touch feature1.txt feature2.txt
git add feature1.txt feature2.txt

git commit -m "Add feature1.txt and feature2.txt to master"
[master (root-commit) 088ad5e] Add feature1.txt and feature2.txt to master
 2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 feature1.txt
  create mode 100644 feature2.txt

git branch develop
git branch feature1
git branch feature2

git checkout develop
Switched to branch 'develop'
touch develop.txt

git stash push -m "Stash develop.txt"
git checkout feature1
touch new.txt

git add new.txt

git commit -m "add new.txt in feature1 branch"
[feature1 b4c2d05] add new.txt in feature1 branch
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 new.txt

git checkout develop
Switched to branch 'develop'

git stash pop
No stash entries found.

git add develop.txt

git commit -m "add develop.txt after unstashing"
[develop 0ce2e4b] add develop.txt after unstashing
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 develop.txt
```

### 3) Tasks to Be Performed:

1. Create a Git working directory, with the following branches:
  - Develop
  - F1
  - f2
2. In the master branch, commit main.txt file
3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively
4. Push all these branches to GitHub
5. On local delete f2 branch
6. Delete the same branch on GitHub as well!

Ans:-

```
SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (main)
$ git checkout -b master
Switched to a new branch 'master'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ touch main.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git add main.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git commit -m "add main.txt to master branch"
On branch master
nothing to commit, working tree clean

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/ssvillan/PW-New-Project/pull/new/master
remote:
To https://github.com/ssvillan/PW-New-Project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout -b develop
Switched to a new branch 'develop'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (develop)
$ touch develop.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (develop)
$ git add develop.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (develop)
$ git commit -m "add develop.txt to develop branch"
[develop e234545] add develop.txt to develop branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 develop.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (develop)
$ git push -u origin develop
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
```

```
SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (develop)
$ git checkout -b f1
Switched to a new branch 'f1'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f1)
$ touch f1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f1)
$ git add f1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f1)
$ git commit -m "add f1.txt to f1 branch"
[f1 ec6a48a] add f1.txt to f1 branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f1)
$ git push -u origin f1
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 256 bytes | 256.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'f1' on GitHub by visiting:
remote:   https://github.com/ssvillan/PW-New-Project/pull/new/f1
remote:
To https://github.com/ssvillan/PW-New-Project.git
 * [new branch]      f1 -> f1
branch 'f1' set up to track 'origin/f1'.

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f1)
$ git checkout -b f2
Switched to a new branch 'f2'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ touch f2.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ git add f2.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ git commit -m "add f2.txt to f2 branch"
[f2 94154ec] add f2.txt to f2 branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f2.txt
```

```

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ git push -u origin f2
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 233 bytes | 233.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'f2' on GitHub by visiting:
remote:     https://github.com/ssvillan/PW-New-Project/pull/new/f2
remote:
To https://github.com/ssvillan/PW-New-Project.git
 * [new branch]      f2 -> f2
branch 'f2' set up to track 'origin/f2'.

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ git branch -D f2
error: cannot delete branch 'f2' used by worktree at 'C:/Users/SUMIT/git-new-project'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (f2)
$ git push origin --delete f2
To https://github.com/ssvillan/PW-New-Project.git
 - [deleted]          f2

```

```

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git branch -D f2
Deleted branch f2 (was 94154ec).

```

#### 4. Tasks to Be Performed:

1. Put master.txt on master branch, stage and commit
2. Create 3 branches: public 1, public 2 and private
3. Put public1.txt on public 1 branch, stage and commit
4. Merge public 1 on master branch
5. Merge public 2 on master branch
6. Edit master.txt on private branch, stage and commit
7. Now update branch public 1 and public 2 with new master code in private
8. Also update new master code on master
9. Finally update all the code on the private branch

**Ans:-**

```

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout master
Already on 'master'
Your branch is up to date with 'origin/master'.

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ touch master.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git add master.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git commit -m "add master.txt to master branch"
[master 25d2675] add master.txt to master branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 master.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git branch public1

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git branch public2

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git branch private

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout public1
Switched to branch 'public1'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ touch public1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git add public1.txt
fatal: pathspec 'public1.txt' did not match any files

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ touch public1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git add public1.txt
fatal: pathspec 'public1.txt' did not match any files

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git add public1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git commit -m "add public1.txt to public1 branch"
[public1 d54605f] add public1.txt to public1 branch

```

```

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git merge public1 -m "merge public1 branch into master"
Updating 25d2675..d54605f
Fast-forward (no commit created; -m option ignored)
  public1.txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 public1.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git merge public2 -m "merge public2 branch into master"
Already up to date.

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout public2
Switched to branch 'public2'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ touch public2.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ git add public2.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ git commit -m "add public2.txt to public2 branch"
[public2 4558da5] add public2.txt to public2 branch
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 public2.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git merge public2 -m "merge public2 branch into master"
Merge made by the 'ort' strategy.
  public2.txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 public2.txt

```

```

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout private
Switched to branch 'private'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (private)
$ echo "Edit the content file from private branch" >> master.txt

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (private)
$ git add master.txt
warning: in the working copy of 'master.txt', LF will be replaced by CRLF the next time Git touches it

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (private)
$ git commit -m "update the master.txt on private branch"
[private f3aad88] update the master.txt on private branch
  1 file changed, 1 insertion(+)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (private)
$ git checkout public1
Switched to branch 'public1'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git merge private -m "merge private branch into public1"
Merge made by the 'ort' strategy.
  master.txt | 1 +
  1 file changed, 1 insertion(+)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public1)
$ git checkout public2
Switched to branch 'public2'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ git merge private -m "merge private branch into public2"
Merge made by the 'ort' strategy.
  master.txt | 1 +
  1 file changed, 1 insertion(+)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (public2)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 4 commits.
  (use "git push" to publish your local commits)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git merge private -m "merge private branch into master"
Merge made by the 'ort' strategy.
  master.txt | 1 +
  1 file changed, 1 insertion(+)

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (master)
$ git checkout private
Switched to branch 'private'

SUMIT@Sumit-Dhal MINGW64 ~/git-new-project (private)
$ git merge master -m "Sync private branch with all final updates from master"
Updating f3aad88..7e72dc9
Fast-forward (no commit created; -m option ignored)
  public1.txt | 0
  public2.txt | 0
  2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 public1.txt
  create mode 100644 public2.txt

```

## 5. Tasks to Be Performed:

1. Create a Git Flow workflow architecture on Git
2. Create all the required branches
3. starting from e feature branch, push the branch to the master, following the architecture
4. Push an urgent.txt on master using hotfix

Ans:-

```
SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (master)
$ git checkout -b master
Switched to a new branch 'master'

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (master)
$ touch master.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (master)
$ git add master.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (master)
$ git commit -m "add master.txt to master branch"
[master (root-commit) d95e034] add master.txt to master branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 master.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (master)
$ git checkout -b develop
Switched to a new branch 'develop'

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (develop)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (feature-branch)
$ touch file.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (feature-branch)
$ git add file.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (feature-branch)
$ git commit -m "add file.txt to feature-branch"
[feature-branch e2bc9ff] add file.txt to feature-branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (feature-branch)
$ git merge feature-branch -m "merge feature-branch into master"
Already up to date.

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (feature-branch)
$ git checkout develop
Switched to branch 'develop'

SUMIT@Sumit-Dhal MINGW64 ~/gitflow-demo (develop)
$ git merge --no-ff feature-branch -m "Merge feature-branch into develop"
Merge made by the 'ort' strategy.
 file.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
```