Introduction to DevOps **(SEZG514)**

Assignment - 1

Submitted by: Soumya Prakash Datta
Roll: 2022mt93539
Email: 2022mt93539@wilp.bits-pilani.ac.in

Problem statement:

ABC Organization would like to opt for the distributed version control system to upgrade their environment, where Git has been selected as the solution. You been assigned as a consultant to educate the migration process to move their Source Code from Centralized to Distributed systems. As a phase one, you would like to go ahead with a workshop to demonstrate below operation to make the ABC team comfortable.

1. Create a Repository
2. Add two directories and some raw code files to the repository
3. Move code from one directory to another directory
4. Update one source code file and display the difference
5. Create a Branch
6. Add some raw code to the branch
7. Merge the Branch with Main line

And at the end provide the Summary of advantages of moving from Centralized Source Code to Distributed Version Control.

Setup:

We need to setup `git` locally and for that we can follow official guide from https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

After installing `git` we can set our **username** and **email** detail globally or this can be setup later on a repo-by-repo basis.

Global setup can be done as follows -

```
git config --global user.name "username"
git config --global user.email "email@domain"
```

To create `git` repo remotely there are many choices, for this project we are going to use github

Create a repository:

Repository can be created using `git init` command.

Let's create a directory

```
mkdir test_repo
cd test_repo
```

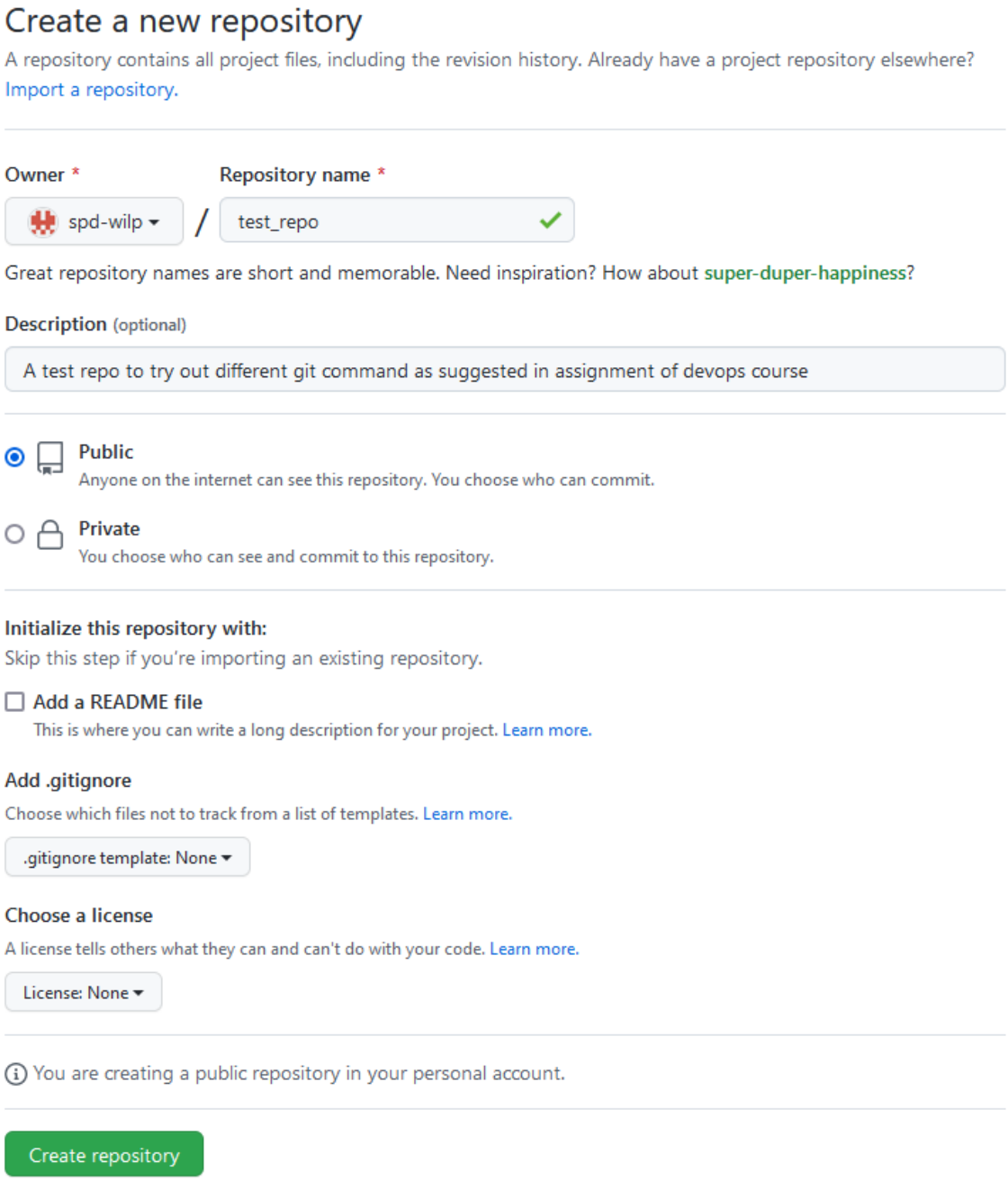
Initializing git repo

```
git init
```

At this point we have initialized an empty git repository.

```
C:\Users\datta\workspace\bits-resources\devops
λ mkdir test_repo

C:\Users\datta\workspace\bits-resources\devops
λ cd test_repo\

C:\Users\datta\workspace\bits-resources\devops\test_repo
λ 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 C:/Users/datta/workspace/bits-resources/devops/test_repo/.git/

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -la
total 4
drwxr-xr-x 1 datta 197609 0 Mar  1 00:33 ./
drwxr-xr-x 1 datta 197609 0 Mar  1 00:33 ../
drwxr-xr-x 1 datta 197609 0 Mar  1 00:33 .git/

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ
```

Now let's setup remote repo in github and link our local repo with it. We can just go to github and click on the click on the **new** button and fill in the details to create a new repo, sample details are added below



Now we need to link our local repo to this remote one, for that we can use `git remote add <name> <link>` command, afterwards we can use `git remote -v` to validate. Let's add our newly created github repo with the name `origin`

```
git remote add origin https://github.com/spd-wilp/test_repo.git
git remote -v
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git remote add origin https://github.com/spd-wilp/test_repo.git

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git remote -v
origin  https://github.com/spd-wilp/test_repo.git (fetch)
origin  https://github.com/spd-wilp/test_repo.git (push)

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ
```

Just to keep things simple, will skip github specific details till the very end of this project, but in general, github provides workflows around managine branches, raising pull-request, code review and many more things.

Add two directories and some raw code files to the repository

Let's create two directories, `dir1` and `dir2` and create some files inside them

```
mkdir dir1 dir2
touch dir1/f1 dir1/f2 dir2/f3 dir2/f4 dir2/f5
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ mkdir dir1 dir2

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ touch dir1/f1 dir1/f2 dir2/f3 dir2/f4 dir2/f5

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -la
total 4
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 ./
drwxr-xr-x 1 datta 197609 0 Mar  1 00:36 ../
drwxr-xr-x 1 datta 197609 0 Mar  1 00:33 .git/
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 dir1/
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 dir2/

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -la dir1
total 0
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 ./
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 ../
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f1
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f2

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -la dir2
total 0
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 ./
drwxr-xr-x 1 datta 197609 0 Mar  1 00:45 ../
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f3
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f4
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f5
```

Now using `git status` we can see that git is not yet tracking those newly created directories/files

```
git status
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dir1/
        dir2/

nothing added to commit but untracked files present (use "git add" to track)
```

To add those files to repository, we can use `git add` and `git commit`

```
git add --all
git commit -m "added initial files"
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git add --all

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   dir1/f1
        new file:   dir1/f2
        new file:   dir2/f3
        new file:   dir2/f4
        new file:   dir2/f5


C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git commit -m "added initial files"
[master (root-commit) 5495e08] added initial files
 5 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dir1/f1
 create mode 100644 dir1/f2
 create mode 100644 dir2/f3
 create mode 100644 dir2/f4
 create mode 100644 dir2/f5

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master
nothing to commit, working tree clean

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git log
commit 5495e08ec1bd32d9566cbe9bd21038383cbc3883 (HEAD -> master)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 00:49:23 2023 +0800

    added initial files
```

At this point all files are added to our local git repo

Move code from one directory to another directory

We can move file from one directory to other directory in git repository using `git mv`

For example lets move `f2` file from `dir1` to `dir2` and then commit this change in git

```
git mv dir1/f2 dir2
git status
git commit -m "moved file"
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git mv dir1\f2 dir2\

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    dir1/f2 -> dir2/f2


C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git commit -m "moved file"
[master 2f0c690] moved file
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename {dir1 => dir2}/f2 (100%)

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git log
commit 2f0c690129f08c41fd72a014d9fb057767d4bc31 (HEAD -> master)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:00:15 2023 +0800

    moved file

commit 5495e08ec1bd32d9566cbe9bd21038383cbc3883
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 00:49:23 2023 +0800

    added initial files
```

Update one source code file and display the difference

Right now the files we have created are empty, lets add some content into one of those files. We can check difference using `git diff` command

```
echo 'hello world' > dir1/f1
git status
git diff
```

Lets now commit this change in repo

```
git add --all
git commit -m "added content in a file"
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ echo 'hello world' > dir1/f1

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ cat dir1/f1
'hello world'

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master
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:   dir1/f1

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

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git diff
diff --git a/dir1/f1 b/dir1/f1
index e69de29..06227be 100644
--- a/dir1/f1
+++ b/dir1/f1
@@ -0,0 +1 @@
+'hello world' ^M

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git add --all

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   dir1/f1


C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git commit -m "added content in a file"
[master 82afadb] added content in a file
 1 file changed, 1 insertion(+)

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git log
commit 82afadb080c8c5d65576337f697cd125ff2087b3 (HEAD -> master)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:13:06 2023 +0800

    added content in a file

commit 2f0c690129f08c41fd72a014d9fb057767d4bc31
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:00:15 2023 +0800

    moved file

commit 5495e08ec1bd32d9566cbe9bd21038383cbc3883
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 00:49:23 2023 +0800

    added initial files
```

Create a branch

So far we have been working from the `master` branch. Lets create a `dev` branch using `git branch` command. After that we can using `git checkout` command to switch to the newly created branch.

```
git branch dev
git checkout dev
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git branch dev

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git branch
  dev
* master

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git checkout dev
Switched to branch 'dev'

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git branch
* dev
  master
```

Add some code to the branch

To change content of a branch, we can first use `git checkout` to switch to the branch, then we can make changes and then perform `git add` and `git commit` to save our changes to the branch

```
git checkout dev
touch dir1/f6
echo "some content" > dir1/f6
git add --all
git commit -m "added a new file"
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git checkout dev
Switched to branch 'dev'

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git branch
* dev
  master

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ touch dir1/f6

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ echo "some content" > dir1\f6

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dir1/f6

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git add --all

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git status
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   dir1/f6


C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git commit -m "added a new file"
[dev 0d265eb] added a new file
 1 file changed, 1 insertion(+)
 create mode 100644 dir1/f6

C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git log
commit 0d265ebad6e95d816931ee5317e0d13c4b2b4733 (HEAD -> dev)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 08:49:10 2023 +0800

    added a new file

commit 82afadb080c8c5d65576337f697cd125ff2087b3 (master)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:13:06 2023 +0800

    added content in a file

commit 2f0c690129f08c41fd72a014d9fb057767d4bc31
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:00:15 2023 +0800

    moved file

commit 5495e08ec1bd32d9566cbe9bd21038383cbc3883
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 00:49:23 2023 +0800

    added initial files
```

Merge the Branch with Main line

Now we have added a file in `dev` branch in the privious step. To merge that update in `master` branch, we can use `git merge`.

```
git checkout master
git merge dev
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (dev)
λ git checkout master
Switched to branch 'master'

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git merge dev
Updating 82afadb..0d265eb
Fast-forward
 dir1/f6 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 dir1/f6

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git log
commit 0d265ebad6e95d816931ee5317e0d13c4b2b4733 (HEAD -> master, dev)
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 08:49:10 2023 +0800

    added a new file

commit 82afadb080c8c5d65576337f697cd125ff2087b3
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:13:06 2023 +0800

    added content in a file

commit 2f0c690129f08c41fd72a014d9fb057767d4bc31
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 01:00:15 2023 +0800

    moved file

commit 5495e08ec1bd32d9566cbe9bd21038383cbc3883
Author: Soumya Prakash Datta <dattasoumyaprakash@gmail.com>
Date:   Wed Mar 1 00:49:23 2023 +0800

    added initial files

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -l dir1\
total 2
-rw-r--r-- 1 datta 197609 16 Mar  1 01:11 f1
-rw-r--r-- 1 datta 197609 18 Mar  1 08:58 f6

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ ls -l dir2\
total 0
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f2
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f3
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f4
-rw-r--r-- 1 datta 197609 0 Mar  1 00:45 f5
```

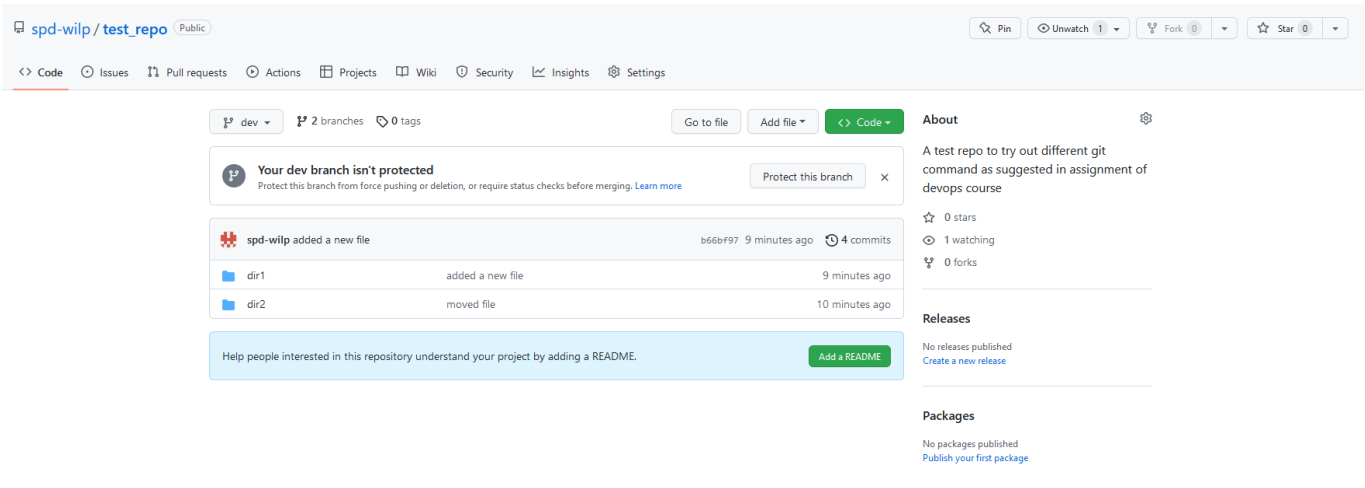
Now let's push all our changes to `github`. For this we can use `git push <remote_name> <branch_name>` command, for our case in the beginning we used `origin` as remote_name, and for now we can push all branches to remote, using `--all` option.

One thing to note here, as we are using `https` in this example, we won't be able to use password as authentication mechanism for github, we have to use `access_token`, it can be provided from github web interface and guide for the same is accessible at setup github access token

Another option is to use `ssh` to connect to github and in that case `ssh keys` will be used for authentication, setup guide for the same can be found at setup ssh key

```
# in case we have a different user in git global config, this will make sure
github to provide authentication prompt and not use the global user by default
git config --local credential.helper ""

git push --all origin
```

```
C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git config --local credential.helper ""

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ git push --all origin
Username for 'https://github.com': spd-wilp
Password for 'https://spd-wilp@github.com':
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (17/17), 1.22 KiB | 250.00 KiB/s, done.
Total 17 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/spd-wilp/test_repo.git
 * [new branch]      dev -> dev
 * [new branch]      master -> master

C:\Users\datta\workspace\bits-resources\devops\test_repo (master)
λ
```



Advantages of Distributed Version Control Systems (DVCS) to Centralized Version Control Systems (CVCS)

1. **Offline work:** One of the most significant advantages of DVCS is the ability to work offline. Developers can work on their local copy of the repository, make changes, and commit them locally without requiring any connection to the central server.
2. **Better collaboration:** In a DVCS, developers can collaborate better and more easily since they can clone the repository and work on their own copy of code, committing and merging changes when they're ready. This can be particularly useful for remote teams who may have limited or unreliable network connectivity.
3. **No single point of failure:** DVCS systems don't have a single point of failure like CVCS systems. If the central server goes down, developers can continue working on their local copies of the repository and then sync changes when the server comes back online. Also in case the central server loses its data, it can be restored easily from one of the local developer copies.
4. **Branching and merging:** DVCS systems are great at branching and merging, allowing developers to create and merge branches easily. This allows for more experimentation and faster iteration in development.

Resources

Github repo: https://github.com/spd-wilp/test_repo

Assignment source has been added inside the repo as well.