Ritesh Kumar Pandey
TY-IT-43

DevOps Lab
Experiment no. 01

Aim :

Create a Sample Web Page and Upload Source Code on GitHub and Perform different Git Operations.

Objective :

(i) To understand DevOps practices which aims to simplify Software Development Life Cycle.
(ii) To be aware of different Version Control tools like GIT, CVS or Mecurial.

Software Used : Ubuntu 20.04 LTS, GitHub, Git.

Theory :

* DevOps -

1) DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.
2) DevOps is a set of practices that combines software development (Dev) and IT operations (Ops).
3) It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

* Why DevOps :

DevOps is about removing the barriers between two tradionally siloed teams, development and operations. DevOps provide following benefits over traditional way :

(i) Speed.
(ii) Rapid Delivery.
(iii) Reliability.
(iv) Scale.
(v) Improved Collaboration
(vi) Security.

* Version Control :

(i) Version Control is the practice of tracking and managing changes to software code.

(ii) Version Control is also known as Source Control.

(iii) Version Control systems are software tools that help software teams manage changes to source code over time.

(iv) Version Control Systems (VCS) are sometimes known as SCM (Source Code Management) tools or RCS (Revision Control System).

* Availabe Version Control :

(i) The most popular VCS tools in use today is called Git.

(ii) Git is a Distributed VCS and is free and open source.

(iii) GitHub, GitLab, Beanstalk, Perforce, AWS CodeCommit, Mercurial, CVS (Concurrent version control), Bitbucket, etc. are some of the available Version Control Softwares.

* Git Commands :

1) git

Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

2) config

This is used to set configuration options. Used to define author name and email for all commits in current repository.

Syntax : git config user.name <name>
git config user.email <email>.

3) help

With no options and no command or Guide given, the synopsis of git command and a list of the most commonly used Git Commands are printed on the standard output.

Syntax : git help [-a|--all [--[no-]verbose]]
[-g|--guides][COMMAND|GUIDE]

4) bugreport

Collect information for user to file a bugreport.

Syntax : git bugreport [(-o|--output-directory]<path>]
[(-s|--suffix)<format>]

5) init

Create empty Git repository in the current directory.

syntax: git init

6) clone

Clone repository located at <repo> onto local machine.

Syntax : git clone <repo>

7) add

Stage all changes in <directory> for the next commit

Syntax : git add <directory>.

8) status

List which files are staged, unstaged and untracked.

Syntax : git status.

9) diff

Shows unstaged changes between your index and working directory.

Syntax : git diff.

10) commit

Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.

Syntax : git commit -m "<message>"

11) notes

Add or inspect object notes. without touching the objects themselves.

Syntax : git notes [list [<object>]]
git notes get-ref
git notes show [<object>]

12) restore

Restore specified paths in the working tree with some contents from a restore source. If a path is tracked but does not exist in the restore source, it will be removed to match the same

Syntax : git restore [<options>][--source=<tree>]
[--staged][--worktree][--] <pathspec>

13) reset

Reset staging area to match most recent commit, but leave the working directory unchanged.

Syntax : git reset

14) rm

The git rm command can be used to remove individual files or a collection of files.

Syntax : git rm <file>

15) mv

(i) git mv [-v] [-f] [-n] [-k] <source><destination>

(ii) git mv [-v][-f][-n][-k] <source> .... <destination>

In the first form, it renames <source>, which must exist and be either a file, symlink or directory, to <destination>. In second form, the last argument has to be an existing directory; the given source will be moved into this directory.

16) **branch**

Lists all the branches in your repository.

Syntax : git branch.

17) **checkout**

create and check out a new branch named <branch>. Drop the -b flag to checkout an existing branch.

Syntax : git checkout -b <branch>.

18) **pull**

Fetches the remote's copy of current branch and rebases it into the local copy. Uses git rebase instead of merge to integrate the branches

Syntax : git pull --rebase <remote>

19) **push**

git push <remote> <branch>

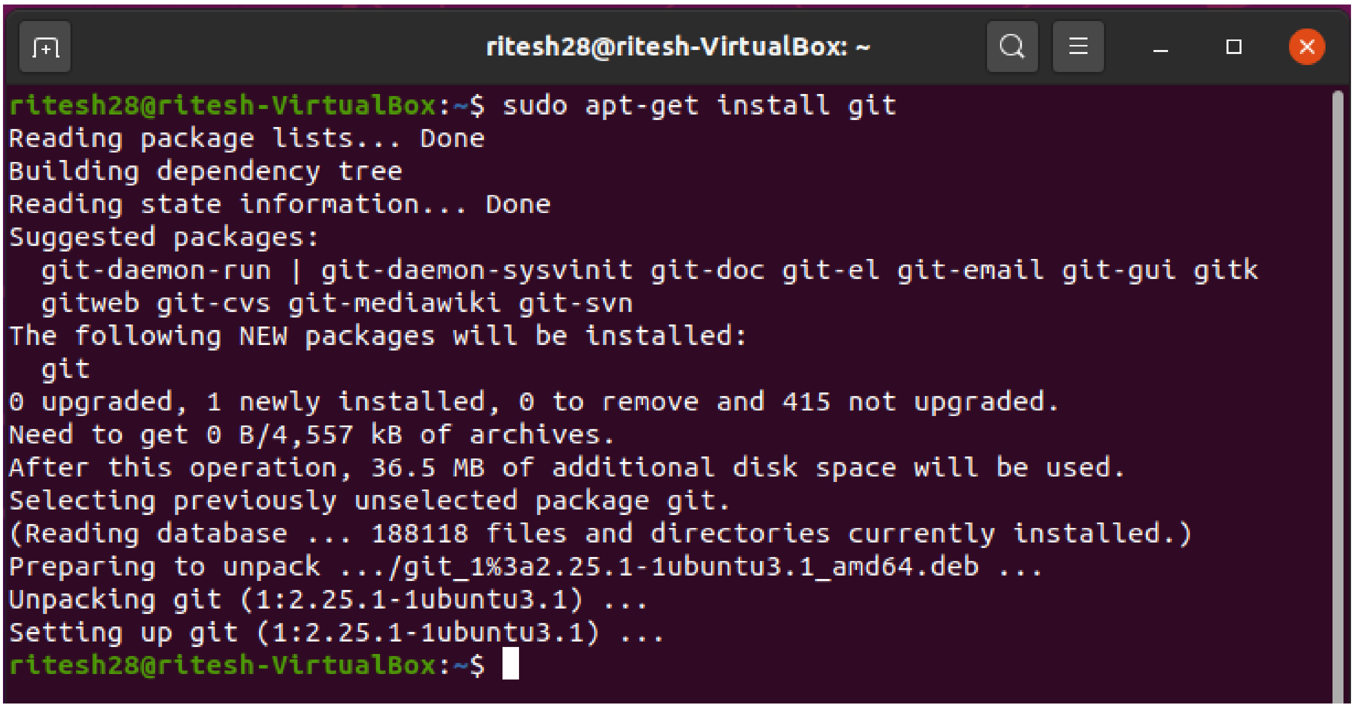
pushes all the commited files in the specified branch of the remote.

20) **remote**

creates a new connection to a remote repository. After adding a remote, we can use <name> as a shortcut for <url> in other commands.

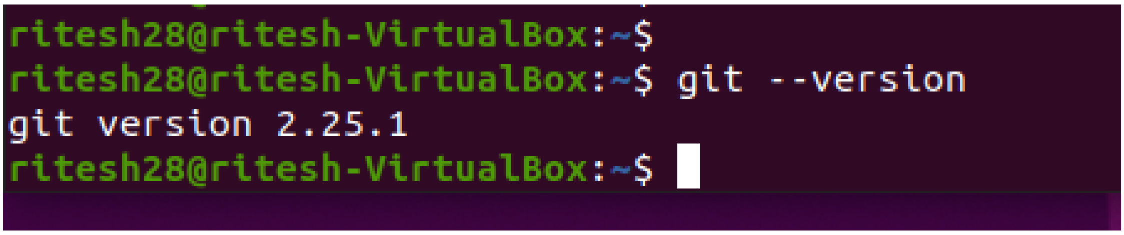
Syntax : git remote add <name> <url>

Output:



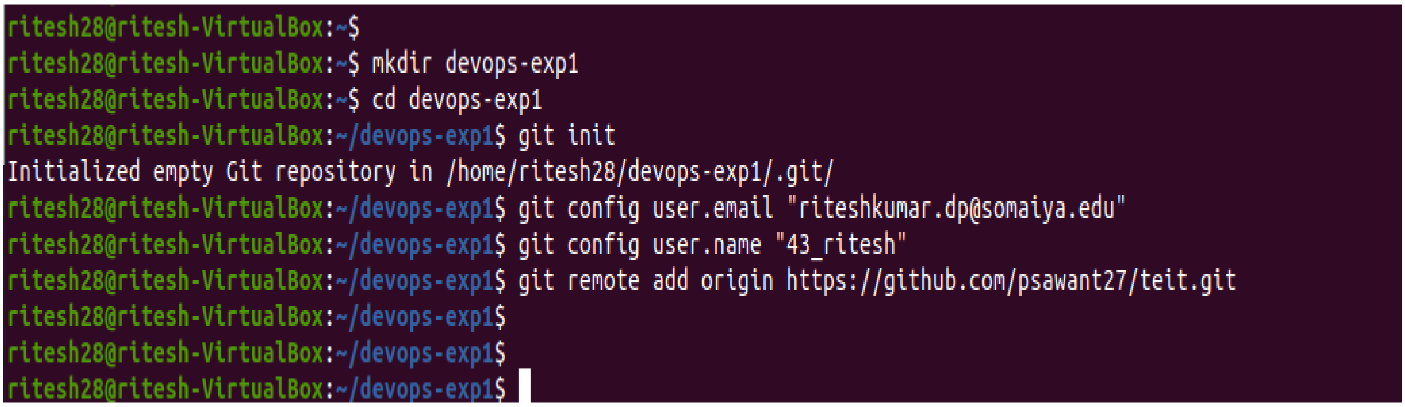
```
ritesh28@ritesh-VirtualBox:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 415 not upgraded.
Need to get 0 B/4,557 kB of archives.
After this operation, 36.5 MB of additional disk space will be used.
Selecting previously unselected package git.
(Reading database ... 188118 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.25.1-1ubuntu3.1_amd64.deb ...
Unpacking git (1:2.25.1-1ubuntu3.1) ...
Setting up git (1:2.25.1-1ubuntu3.1) ...
ritesh28@ritesh-VirtualBox:~$
```

Figure 1: Git Installation



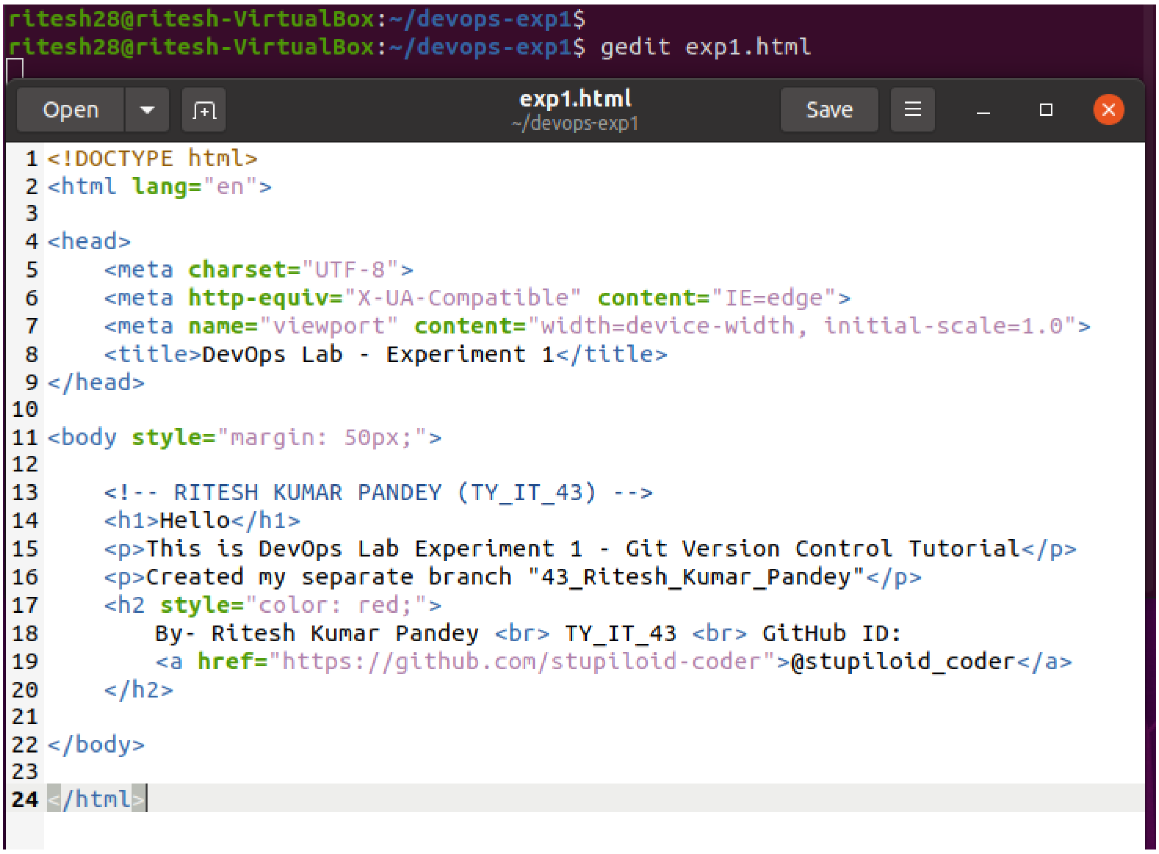
```
ritesh28@ritesh-VirtualBox:~$
ritesh28@ritesh-VirtualBox:~$ git --version
git version 2.25.1
ritesh28@ritesh-VirtualBox:~$
```

Figure 2: Git version



```
ritesh28@ritesh-VirtualBox:~$
ritesh28@ritesh-VirtualBox:~$ mkdir devops-exp1
ritesh28@ritesh-VirtualBox:~$ cd devops-exp1
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git init
Initialized empty Git repository in /home/ritesh28/devops-exp1/.git/
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git config user.email "riteshkumar.dp@somaiya.edu"
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git config user.name "43_ritesh"
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git remote add origin https://github.com/psawant27/teit.git
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$
```

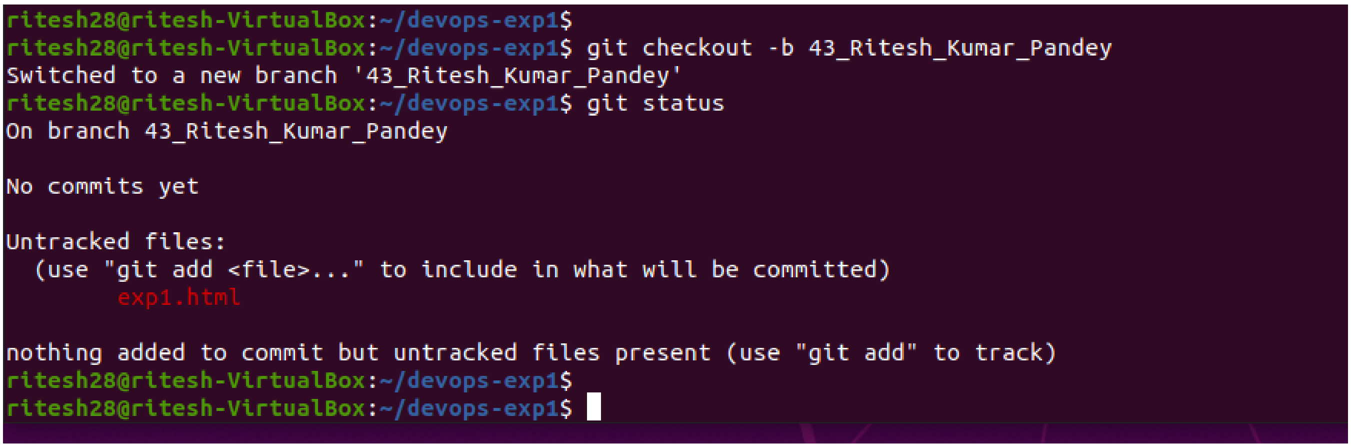
Figure 3: Git init, config & remote



```
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$ gedit exp1.html
```

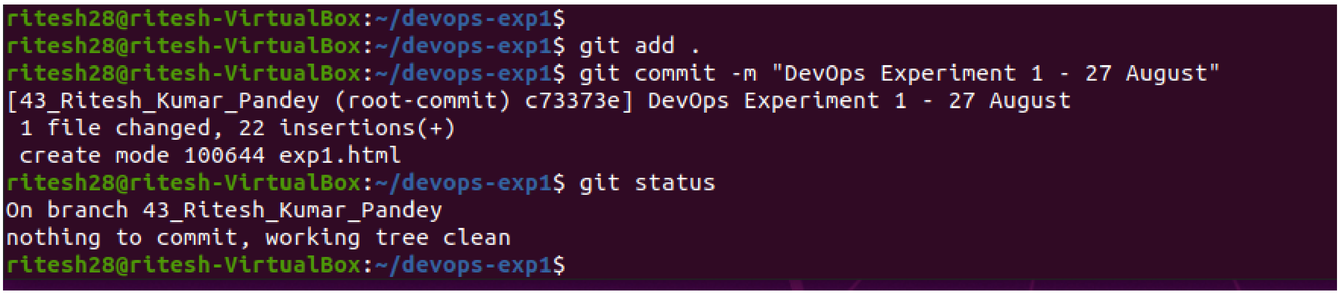
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DevOps Lab - Experiment 1</title>
</head>

<body style="margin: 50px;">

    <!-- RITESH KUMAR PANDEY (TY_IT_43) -->
    <h1>Hello</h1>
    <p>This is DevOps Lab Experiment 1 - Git Version Control Tutorial</p>
    <p>Created my separate branch "43_Ritesh_Kumar_Pandey"</p>
    <h2 style="color: red;">
        By- Ritesh Kumar Pandey <br> TY_IT_43 <br> GitHub ID:
        <a href="https://github.com/stupiloid-coder">@stupiloid_coder</a>
    </h2>

</body>

</html>
```

Figure 4: Website source code



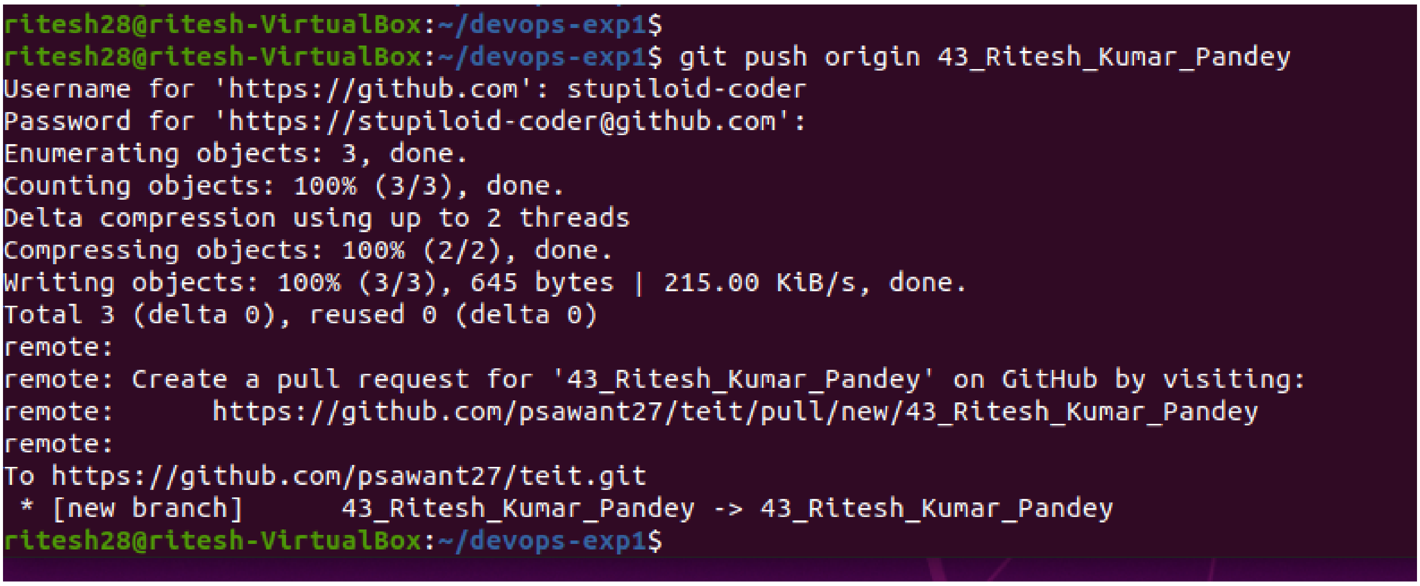
```
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git checkout -b 43_Ritesh_Kumar_Pandey
Switched to a new branch '43_Ritesh_Kumar_Pandey'
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git status
On branch 43_Ritesh_Kumar_Pandey

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        exp1.html

nothing added to commit but untracked files present (use "git add" to track)
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$ 
```

Figure 5: git checkout (creation of new branch) and git status



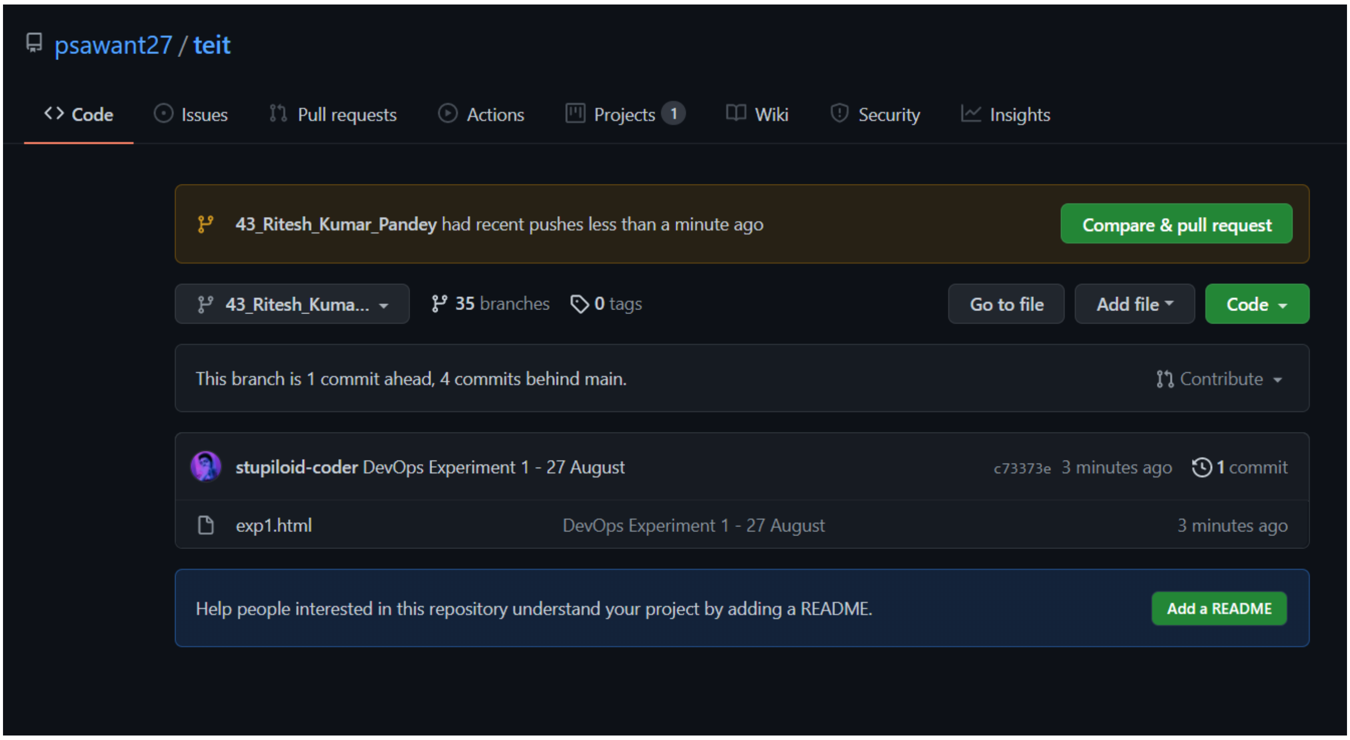
```
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git add .
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git commit -m "DevOps Experiment 1 - 27 August"
[43_Ritesh_Kumar_Pandey (root-commit) c73373e] DevOps Experiment 1 - 27 August
 1 file changed, 22 insertions(+)
 create mode 100644 exp1.html
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git status
On branch 43_Ritesh_Kumar_Pandey
nothing to commit, working tree clean
ritesh28@ritesh-VirtualBox:~/devops-exp1$
```

Figure 6: git add & commit



```
ritesh28@ritesh-VirtualBox:~/devops-exp1$
ritesh28@ritesh-VirtualBox:~/devops-exp1$ git push origin 43_Ritesh_Kumar_Pandey
Username for 'https://github.com': stupiloid-coder
Password for 'https://stupiloid-coder@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 645 bytes | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for '43_Ritesh_Kumar_Pandey' on GitHub by visiting:
remote:      https://github.com/psawant27/teit/pull/new/43_Ritesh_Kumar_Pandey
remote:
To https://github.com/psawant27/teit.git
 * [new branch]      43_Ritesh_Kumar_Pandey -> 43_Ritesh_Kumar_Pandey
ritesh28@ritesh-VirtualBox:~/devops-exp1$
```

Figure 7: git push



43_Ritesh_Kumar_Pandey had recent pushes less than a minute ago

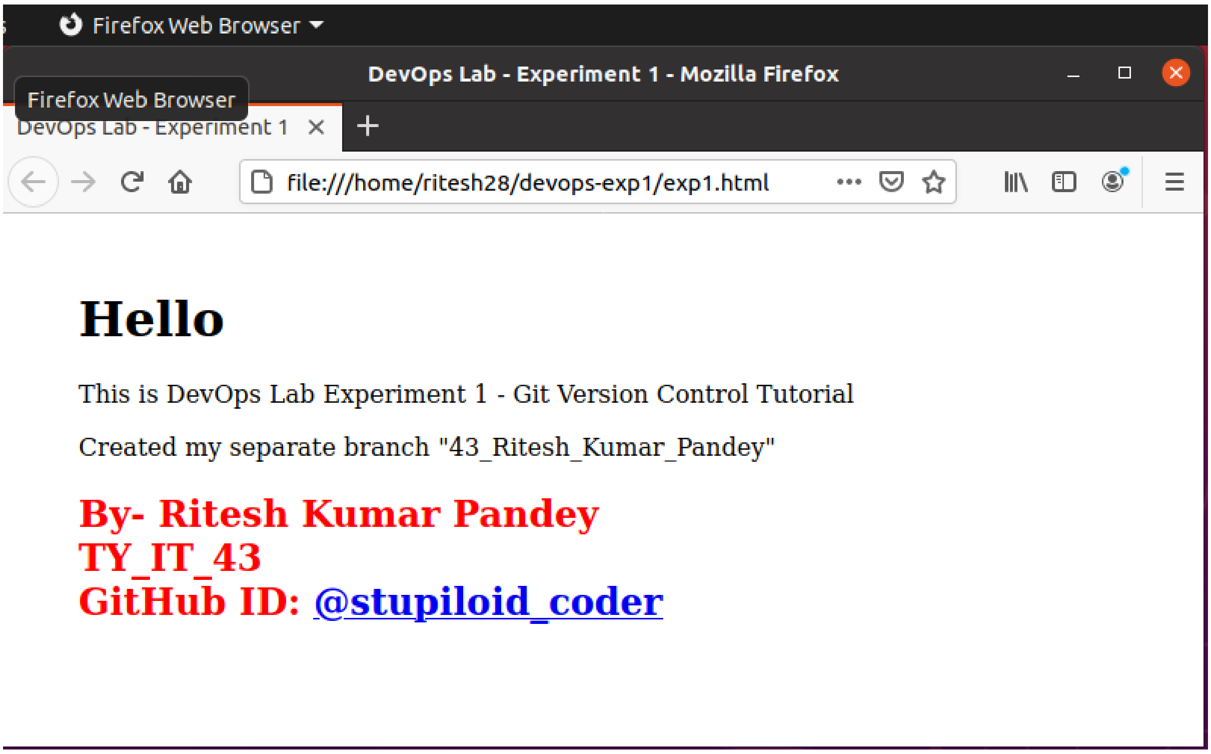
This branch is 1 commit ahead, 4 commits behind main.

stupiloid-coder DevOps Experiment 1 - 27 August c73373e 3 minutes ago 1 commit

exp1.html DevOps Experiment 1 - 27 August 3 minutes ago

Help people interested in this repository understand your project by adding a README.

Figure 8: changes in repository



Firefox Web Browser
DevOps Lab - Experiment 1 - Mozilla Firefox
Firefox Web Browser
DevOps Lab - Experiment 1
file:///home/ritesh28/devops-exp1/exp1.html

Hello

This is DevOps Lab Experiment 1 - Git Version Control Tutorial

Created my separate branch "43_Ritesh_Kumar_Pandey"

By- Ritesh Kumar Pandey
TY_IT_43
GitHub ID: @stupiloid_coder

Figure 9: Website in browser

Conclusion:

DevOps is a set of process that combines software development (Dev) and IT Operations (Ops). It aims to shorten the System Development Life Cycle and provide continuous delivery with high software quality. Version Control System are used to track and manage changes in a software code. It is used to collaborate with other developers. Git is free open-source Version Control Software. Git has various inbuilt utility commands such as push, pull, add, commit, etc. which helps collaboration and tracking easy. In this experiment, installation of git is done along with performing some basic utility command. A sample website is created and pushed into the git repository from local ubuntu terminal. A separate branch is also created using the git checkout command and the files are pushed into the newly created branch.

Outcome:

Ability to use Version Control for developing application.

Git repository link:

https://github.com/psawant27/teit/tree/43_Ritesh_Kumar_Pandey