# CVS and git repository

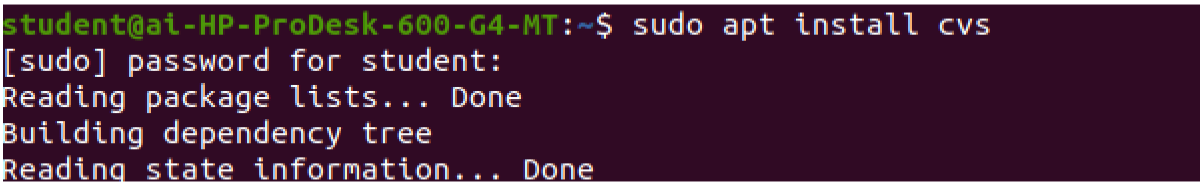
**Use Git** because it’s **faster, more flexible, works offline, and supports easy branching & merging**.

**CVS is outdated** and not recommended for modern projects.

**CVS**: Developers work on a single central server. They check out files, edit them, and commit directly to the main repo. **Git**: Developers clone a full copy of the repository, work locally, commit changes, and later push them to a remote server.

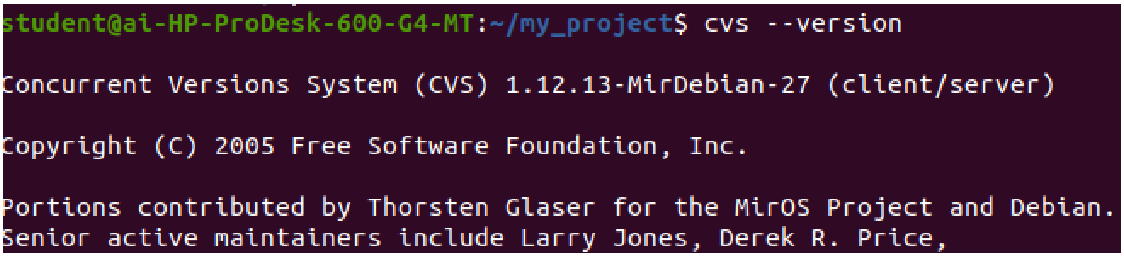
But Both are version control systems.

1.CVS must be installed on your system before using it.



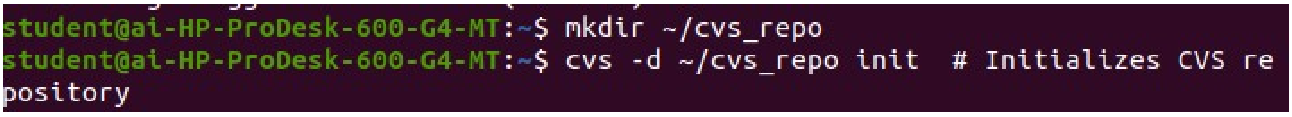
```
student@ai-HP-ProDesk-600-G4-MT:~$ sudo apt install cvs
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Checking its version



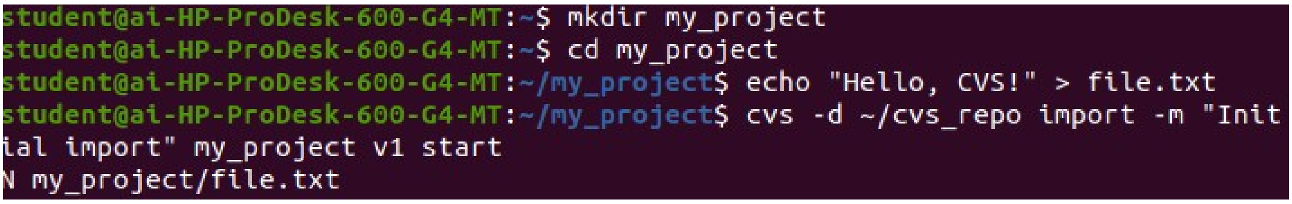
```
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs --version

Concurrent Versions System (CVS) 1.12.13-MirDebian-27 (client/server)

Copyright (C) 2005 Free Software Foundation, Inc.

Portions contributed by Thorsten Glaser for the MirOS Project and Debian.
Senior active maintainers include Larry Jones, Derek R. Price,
```

2.Initialize new repository



```
student@ai-HP-ProDesk-600-G4-MT:~$ mkdir ~/cvs_repo
student@ai-HP-ProDesk-600-G4-MT:~$ cvs -d ~/cvs_repo init  # Initializes CVS re
pository
```

3.Create and Import Project



```
student@ai-HP-ProDesk-600-G4-MT:~$ mkdir my_project
student@ai-HP-ProDesk-600-G4-MT:~$ cd my_project
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ echo "Hello, CVS!" > file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs -d ~/cvs_repo import -m "Init
ial import" my_project v1 start
N my_project/file.txt
```

make a directory,change ,create a file .

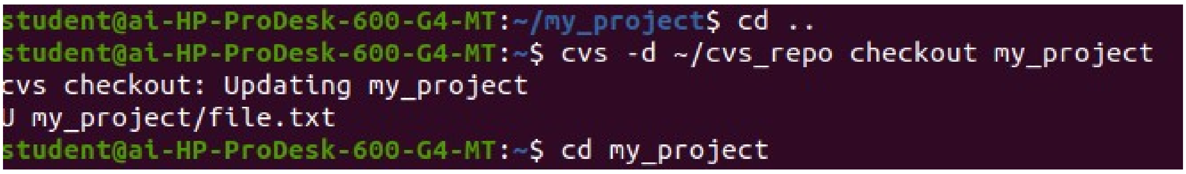
Import the project into CVS:

`my_project`: Project name

`v1`: Vendor tag (can be any label)

`start`: Release tag (initial version)

4.Checkout (Retrieve) the Project

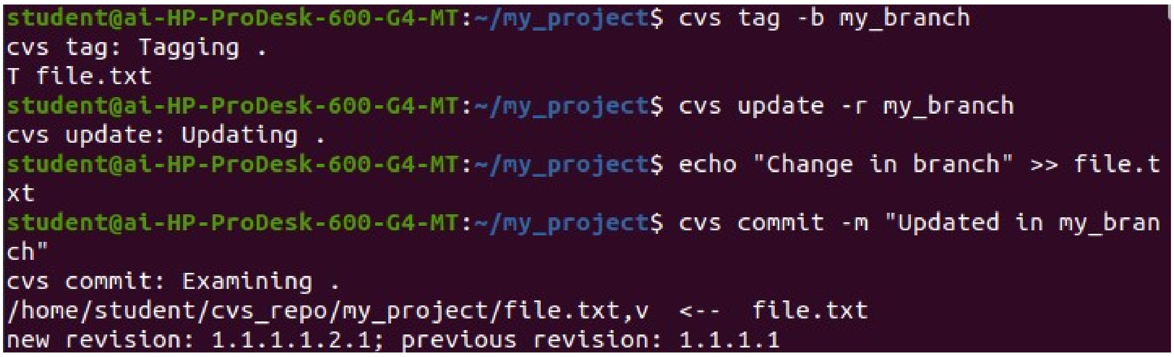


```
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cd ..
student@ai-HP-ProDesk-600-G4-MT:~$ cvs -d ~/cvs_repo checkout my_project
cvs checkout: Updating my_project
U my_project/file.txt
student@ai-HP-ProDesk-600-G4-MT:~$ cd my_project
```

Move out of the project directory , Check out the project from CVS , Move into the project folder

-d?:

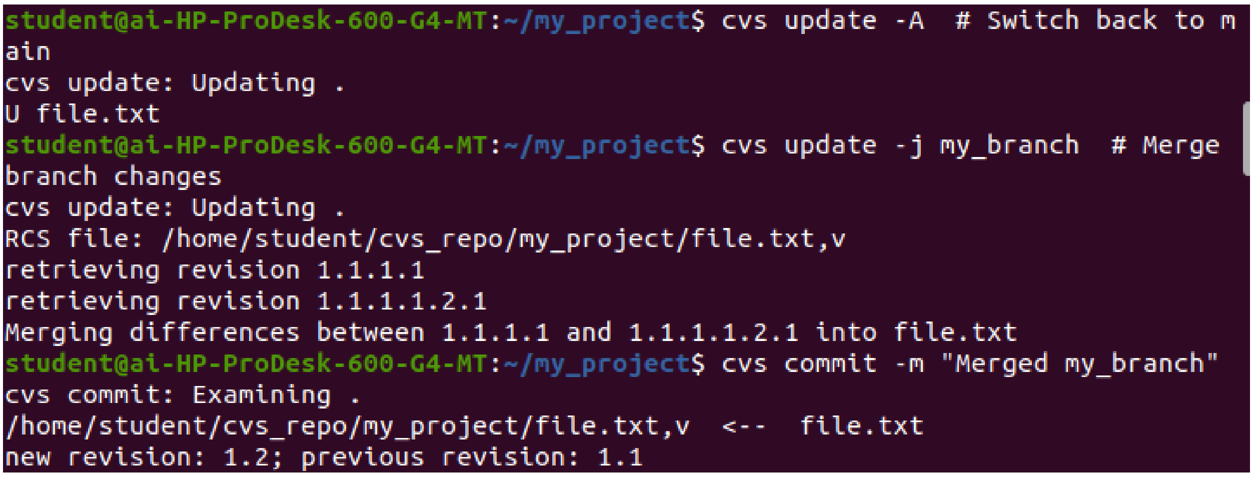
5.Create and Switch to a Branch



```
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs tag -b my_branch
cvs tag: Tagging .
T file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs update -r my_branch
cvs update: Updating .
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ echo "Change in branch" >> file.t
xt
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs commit -m "Updated in my_bran
ch"
cvs commit: Examining .
/home/student/cvs_repo/my_project/file.txt,v  <--  file.txt
new revision: 1.1.1.1.2.1; previous revision: 1.1.1.1
```

Create a new branchSwitch to the new branch:Modify a file and commit:

6.Merge the Branch into Main

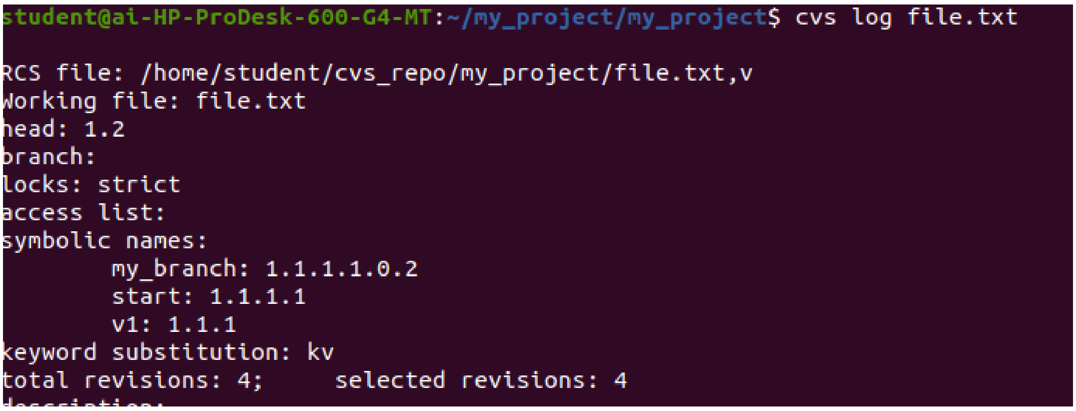


```
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs update -A  # Switch back to m
ain
cvs update: Updating .
U file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs update -j my_branch  # Merge
branch changes
cvs update: Updating .
RCS file: /home/student/cvs_repo/my_project/file.txt,v
retrieving revision 1.1.1.1
retrieving revision 1.1.1.1.2.1
Merging differences between 1.1.1.1 and 1.1.1.1.2.1 into file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_project$ cvs commit -m "Merged my_branch"
cvs commit: Examining .
/home/student/cvs_repo/my_project/file.txt,v  <--  file.txt
new revision: 1.2; previous revision: 1.1
```

Switch back to the main branch:Merge changes from my_branch Commit the merge

7.Final Check

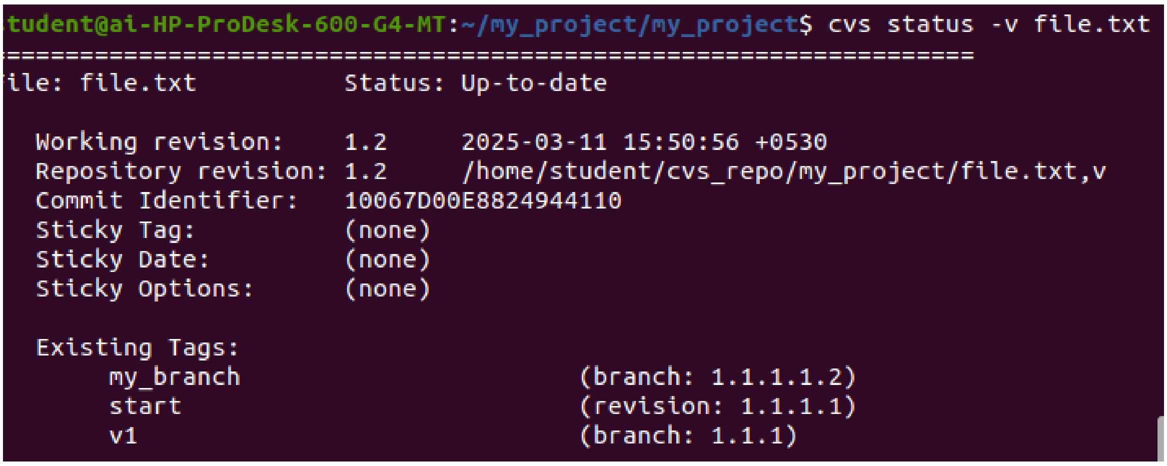
This will show the file’s version history.



```
student@ai-HP-ProDesk-600-G4-MT:~/my_project/my_project$ cvs log file.txt

RCS file: /home/student/cvs_repo/my_project/file.txt,v
Working file: file.txt
head: 1.2
branch:
locks: strict
access list:
symbolic names:
        my_branch: 1.1.1.1.0.2
        start: 1.1.1.1
        v1: 1.1.1
keyword substitution: kv
total revisions: 4;     selected revisions: 4
description:
```

8.to list all files or directories and hidden files if any.
   Ls -a

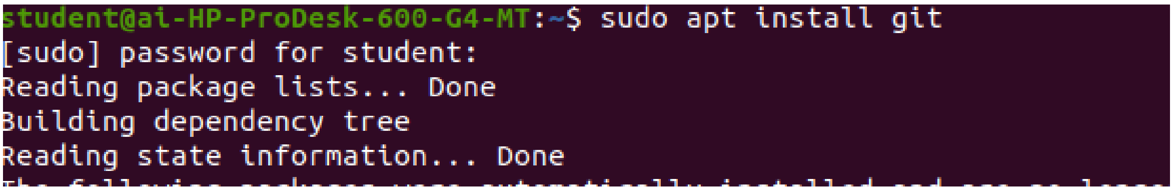
9. to list all the branches.



```
tudent@ai-HP-ProDesk-600-G4-MT:~/my_project/my_project$ cvs status -v file.txt
===================================================================
ile: file.txt           Status: Up-to-date

   Working revision:    1.2     2025-03-11 15:50:56 +0530
   Repository revision: 1.2     /home/student/cvs_repo/my_project/file.txt,v
   Commit Identifier:   10067D00E8824944110
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)

   Existing Tags:
        my_branch                       (branch: 1.1.1.1.2)
        start                           (revision: 1.1.1.1)
        v1                              (branch: 1.1.1)
```

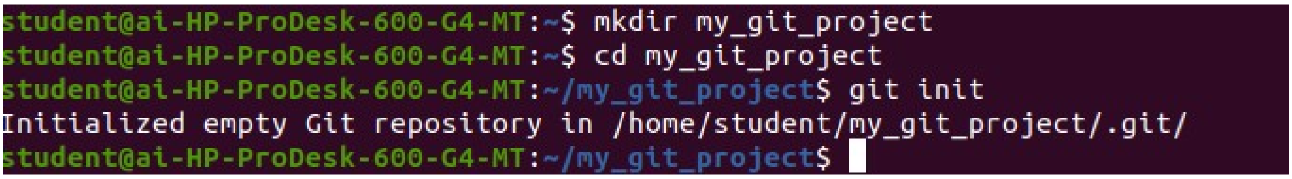
GIT REPOSITORY Setting Up Git

## 1.Install Git



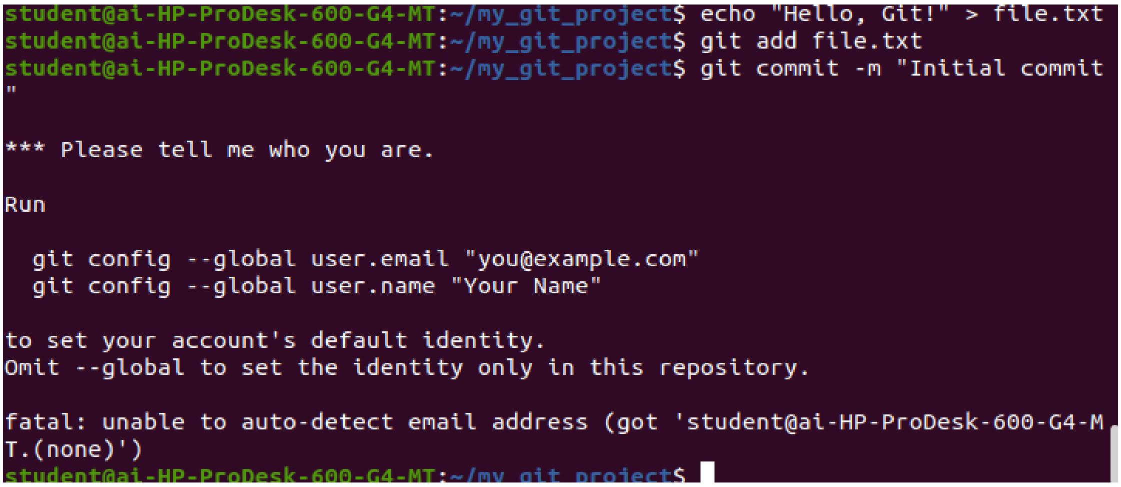
```
student@ai-HP-ProDesk-600-G4-MT:~$ sudo apt install git
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

## 2. Initialize a Git Repository



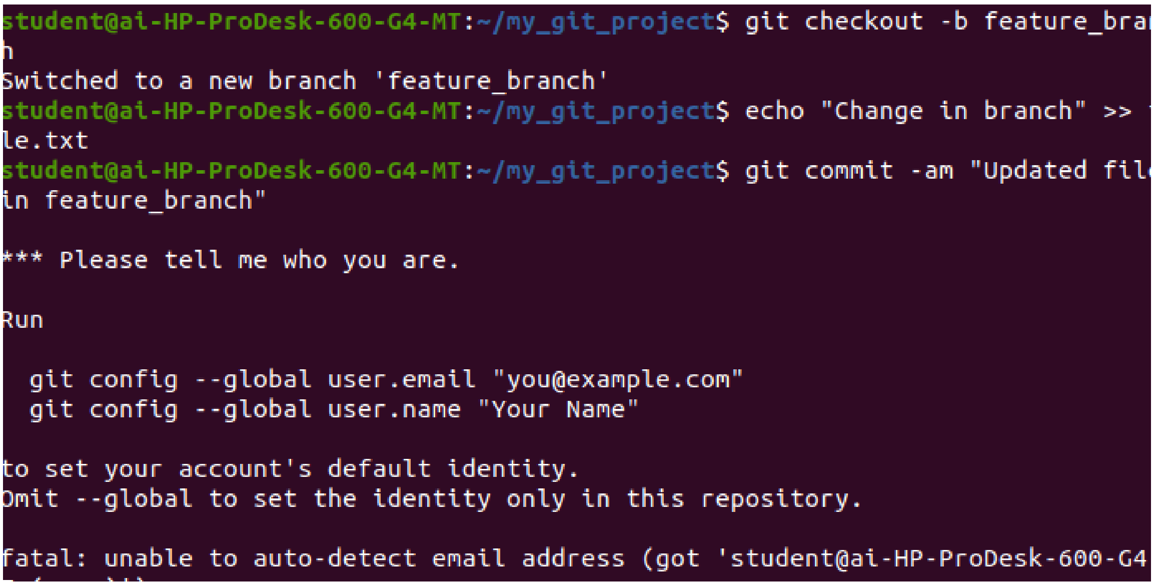
```
student@ai-HP-ProDesk-600-G4-MT:~$ mkdir my_git_project
student@ai-HP-ProDesk-600-G4-MT:~$ cd my_git_project
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git init
Initialized empty Git repository in /home/student/my_git_project/.git/
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$
```

## 3. Add a File and Commit



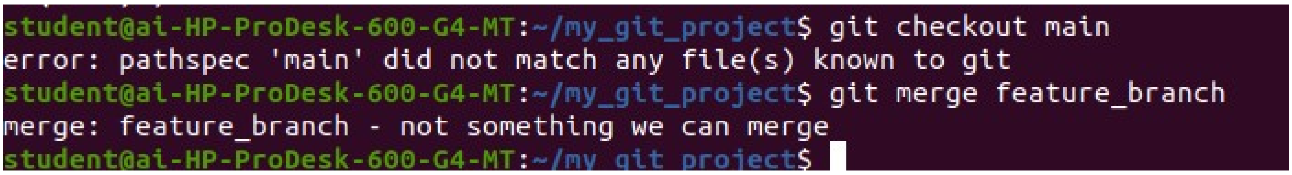
```
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ echo "Hello, Git!" > file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git add file.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git commit -m "Initial commit
"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'student@ai-HP-ProDesk-600-G4-M
T.(none)')
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$
```

## 4. Create and Switch to a Branch



```
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git checkout -b feature_bra
h
Switched to a new branch 'feature_branch'
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ echo "Change in branch" >>
le.txt
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git commit -am "Updated fil
in feature_branch"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'student@ai-HP-ProDesk-600-G4
```

5. Merge the Branch into Main



```
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ git merge feature_branch
merge: feature_branch - not something we can merge
student@ai-HP-ProDesk-600-G4-MT:~/my_git_project$ 
```

6. If Git detects a conflict, manually edit `file.txt`.

7.Use `git log --graph --oneline --all` to visualize branches.