

1o- Experiment 1: Git Installation and Basic Git Commands

Aim: Perform installation of Git Bash and explore usage of basic Git Commands.

Steps:

1. Download Git from git-scm.com and install it.
2. Configure Git with username and email using:
`git config --global user.name "Your Name"`
`git config --global user.email "your.email@example.com"`
3. Initialize a local repository:
`git init`
4. Add a file and make a commit:
`git add filename`
`git commit -m "Initial commit"`
5. Explore branching:
`git branch feature-1`
`git checkout feature-1`
6. Merge changes and review logs:
`git merge feature-1`
`git log --oneline --graph`

Conclusion: This experiment gave hands-on experience with basic Git commands. We understood how to set up Git, create repositories, manage versions through commits and branches, and track history using logs. These skills are essential for collaborative and efficient software development.

Experiment-1 Git installation and basic Git Commands

```

MINGW64:/c/Users/vaibh/EXP_1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git --version
git version 2.45.0.windows.1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ ^[[200~
bash: $'\E[200~': command not found

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ 

git config --global user.name "Vaibhav_Chavan"

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git config --global user.email "vai.cha.rt22@dypatil.edu"

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ mkdir EXP_1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ cd EXP_1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1
$ ^[[200~
bash: $'\E[200~': command not found

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1
$ git init
Initialized empty Git repository in C:/Users/vaibh/EXP_1/.git/

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ touch a1.txt a2.txt a3.txt a4.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a1.txt
This is a1.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a2.txt
This is a2.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a3.txt
This is a3.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a4.txt
This is a4.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    a1.txt
    a2.txt
    a3.txt
    a4.txt

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git add .
warning: in the working copy of 'a1.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'a3.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'a4.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git commit -m "initial commit: added a1,a2,a3,a4.txt"
[master (root-commit) 390daa5] initial commit: added a1,a2,a3,a4.txt

```

MINGW64:/c/Users/vaibh/EXP_1

```
warning: in the working copy of 'a4.txt', LF will be replaced by CRLF the next time Git touches it
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
```

```
$ git commit -m "initial commit: added a1,a2,a3,a4.txt"
[master (root-commit) 390daa5] initial commit: added a1,a2,a3,a4.txt
 4 files changed, 4 insertions(+)
 create mode 100644 a1.txt
 create mode 100644 a2.txt
 create mode 100644 a3.txt
 create mode 100644 a4.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ nano a1.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat a1.txt
is a1.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat *.txt
is a1.txt
This is a2.txt
This a4.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat a3.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a3.txt
This is a3.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat a3.txt
This is a3.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat *.txt
is a1.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat *.txt
is a1.txt
This is a2.txt
This is a3.txt
This a4.txt
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > *.txt
bash: *.txt: ambiguous redirect
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a1.txt
Test1
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a2.txt
Test2
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a3.txt
Test3
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a4.txt
Test4
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > *.txt
bash: *.txt: ambiguous redirect
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat *.txt
Test1
Test2
Test3
```

MINGW64:/c/Users/vaibh/EXP_1

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git add .
warning: in the working copy of 'a1.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'a2.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'a3.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'a4.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git commit -m "commit1: added a1,a2,a3,a4.txt"
[master cf30dcc] commit1: added a1,a2,a3,a4.txt
 4 files changed, 4 insertions(+), 4 deletions(-)

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git status
On branch master
nothing to commit, working tree clean

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat > a4.txt
Yes test4

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ 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:   a4.txt

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git add .
warning: in the working copy of 'a4.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
```

```
MINGW64:/c/Users/vaibh/EXP_1
no changes added to commit (use "git add" and/or "git commit -a")

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git add .
warning: in the working copy of 'a4.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git commit -m "initial commit: added a1,a2,a3,a4.txt"
[master 17da96a] initial commit: added a1,a2,a3,a4.txt
 1 file changed, 1 insertion(+), 1 deletion(-)

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git status
On branch master
nothing to commit, working tree clean

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git branch feature-branch

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ cat a3.txt
Test3

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ git add a1.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ git commit -m "new feature branch for a1"
On branch feature-branch
nothing to commit, working tree clean

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ cat a1.txt
Test1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ cat > a1.txt
```

```
MINGW64:/c/Users/vaibh/EXP_1
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ git checkout master
Switched to branch 'master'
M     a1.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat a1.txt
MODIFICATION is done in feature 1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'
M     a1.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ cat a1.txt
MODIFICATION is done in feature 1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ ls
a1.txt  a2.txt  a3.txt  a4.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ add a5.txt
bash: add: command not found

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ touch a5.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ cat > a5.txt
New txt in this branch

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (feature-branch)
$ ls
a1.txt  a2.txt  a3.txt  a4.txt  a5.txt
```

```
MINGW64:/c/Users/vaibh/EXP_1
$ git checkout master
Switched to branch 'master'
M     a1.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ 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:   a1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    a5.txt

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ cat a5.txt
New txt in this branch

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ ls
a1.txt  a2.txt  a3.txt  a4.txt  a5.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git merge new-feature
merge: new-feature - not something we can merge

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git merge feature branch
merge: feature - not something we can merge

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git merge feature-branch
Already up to date.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/EXP_1 (master)
$ git log --oneline --graph --decorate
* 17da96a (HEAD -> master, feature-branch) initial commit: added a1,a2,a3,a4.txt
* cf30dcc commit1: added a1,a2,a3,a4.txt
```

2o- Experiment 2: Create and Fork Repositories in GitHub

Aim: Create and fork repositories in GitHub.

Steps:

1. Sign up on [GitHub](#).
2. Create a new repository using the GitHub UI.
3. Clone the repository to local machine:
4. `git clone https://github.com/yourusername/repo-name.git`
5. Fork a repository and make changes in your fork.
6. Sync changes between local and remote repositories.
7. Tag a commit:
8. `git tag v1.0`
9. Revert a commit using:
10. `git revert <commit_id>`

Conclusion: This experiment demonstrated how GitHub facilitates collaborative development. We practiced forking, cloning, syncing, tagging, and reverting changes — all critical for maintaining clean, versioned codebases in a team environment.

```
MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local
vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git --version
git version 2.45.0.windows.1

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git config --global user.name "Vaichavan"

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git config --global user.email "vaibhavschavan26@gmail.com"

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttpspath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Vaichavan
user.email=vaibhavschavan26@gmail.com

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ mkdir Gitdemo

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ cd Gitdemo

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo
$ git clone https://github.com/Vaichavan/Exp_1_local.git
Cloning into 'Exp_1_local'...
warning: You appear to have cloned an empty repository.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo
$ cd Exp_1_local

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
```

```
MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    beta.txt

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git add .
warning: in the working copy of 'alpha.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'beta.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_Local (main)
$ git commit -m "alpha gamma"
[main 7820f0a] alpha gamma
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 beta.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_Local (main)
$ git remote add origin https://github.com/Vaichavan/Exp_1_local.git
error: remote origin already exists.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git remote -v
origin  https://github.com/Vaichavan/Exp_1_local.git (fetch)
origin  https://github.com/Vaichavan/Exp_1_local.git (push)

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/Vaichavan/Exp_1_local.git'

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 525 bytes | 525.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vaichavan/Exp_1_local.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ A|
```

```
MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Vaichavan
user.email=vaibhavschavan26@gmail.com

vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ mkdir Gitdemo
vaibh@LAPTOP-TQNHF5AE MINGW64 ~
$ cd Gitdemo

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo
$ git clone https://github.com/Vaichavan/Exp_1_local.git
Cloning into 'Exp_1_local'...
warning: You appear to have cloned an empty repository.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo
$ cd Exp_1_local

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ touch alpha.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ notepad alpha.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git status
On branch main

No commits yet

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

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git add .

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git status
```

```
MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local
No commits yet

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

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git add .

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   alpha.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git commit -m "beta gamma"
[main (root-commit) d90614d] beta gamma
 1 file changed, 1 insertion(+)
 create mode 100644 alpha.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git log
commit d90614d2a7c1f39f916ac9390ba47c2e5aac4025 (HEAD -> main)
Author: Vaichavan <vaibhavschavan26@gmail.com>
Date:   Tue Feb 4 15:29:28 2025 +0530

  beta gamma

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ touch beta.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ cat > beta.txt
THIS IS Vaibhav_beta

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ cat > alpha.txt
This is Vaibhav_alpha
```

MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ touch beta.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ cat > beta.txt
THIS IS Vaibhav_beta

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ cat > alpha.txt
This is Vaibhav_alpha

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

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:   alpha.txt

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

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git add .
warning: in the working copy of 'alpha.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'beta.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git commit -m "alpha gamma"
[main 7820f0a] alpha gamma
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 beta.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git remote add origin https://github.com/Vaichavan/Exp_1_local.git
error: remote origin already exists.
```

MINGW64:/c/Users/vaibh/Gitdemo/Exp_1_local

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

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

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git add .
warning: in the working copy of 'alpha.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'beta.txt', LF will be replaced by CRLF the next time Git touches it

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git commit -m "alpha gamma"
[main 7820f0a] alpha gamma
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 beta.txt

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git remote add origin https://github.com/Vaichavan/Exp_1_local.git
error: remote origin already exists.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git remote -v
origin  https://github.com/Vaichavan/Exp_1_local.git (fetch)
origin  https://github.com/Vaichavan/Exp_1_local.git (push)

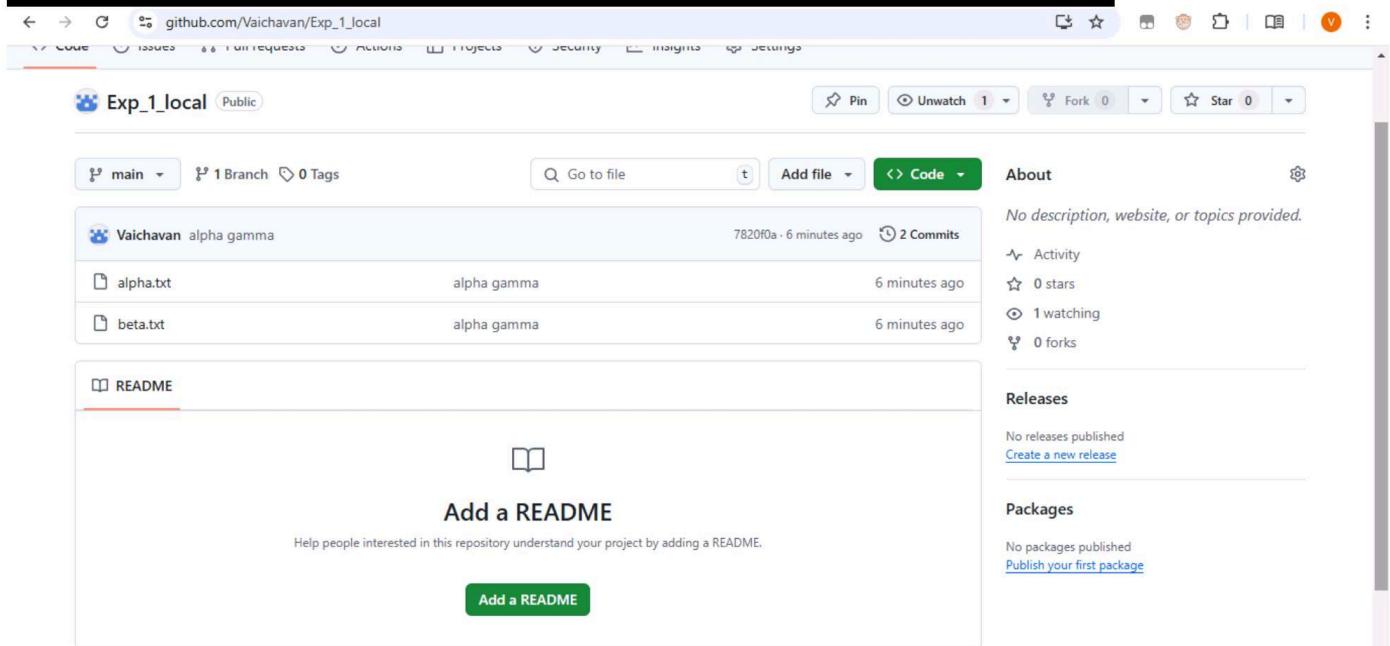
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/Vaichavan/Exp_1_local.git'

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 525 bytes | 525.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vaichavan/Exp_1_local.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ A
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ git pull origin main
From https://github.com/Vaichavan/Exp_1_local
 * branch            main      -> FETCH_HEAD
Already up to date.
```

```
vaibh@LAPTOP-TQNHF5AE MINGW64 ~/Gitdemo/Exp_1_local (main)
$ |
```



The screenshot shows a GitHub repository page for 'Exp_1_local'. At the top, there's a terminal window showing the command 'git pull origin main' was run and it was already up to date. Below the terminal is the GitHub repository interface. The repository name is 'Exp_1_local' and it's public. It has 1 branch ('main') and 0 tags. The 'Code' tab is selected. The repository contains two commits by user 'Vaichavan': one commit 'alpha gamma' (7820f0a) and another commit 'alpha gamma' (6 minutes ago). There are also two files: 'alpha.txt' and 'beta.txt', both last modified 6 minutes ago. On the right side of the repository page, there are sections for 'About', 'Activity', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Activity' section shows 0 stars, 1 watching, and 0 forks. The 'Releases' and 'Packages' sections both note 'No releases published' and 'Publish your first package'.

3o- Experiment 3: Installation and Configuration of Jenkins

Aim: To install and configure Jenkins to set up a build job.

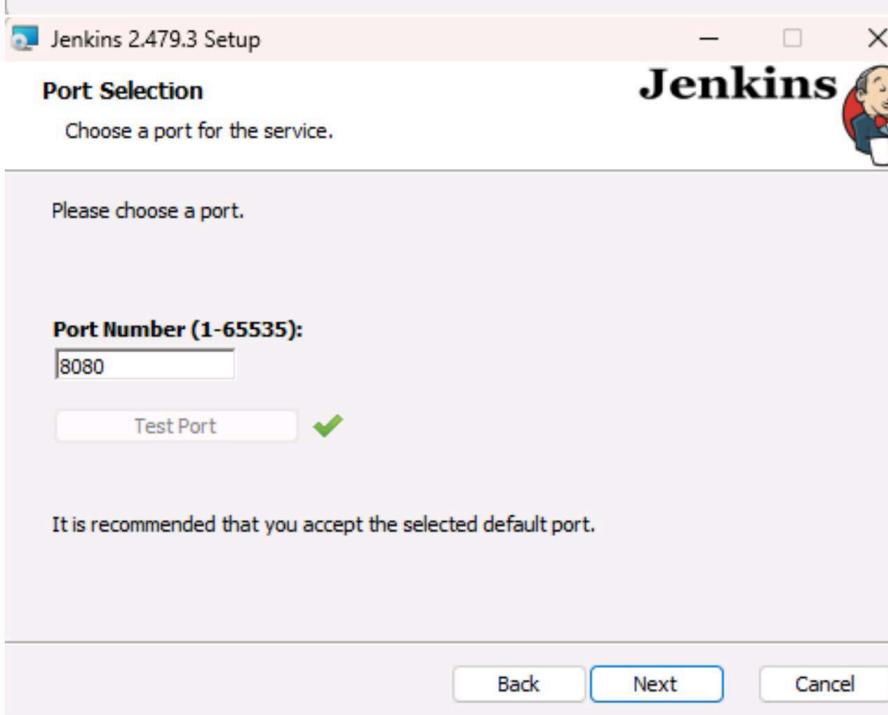
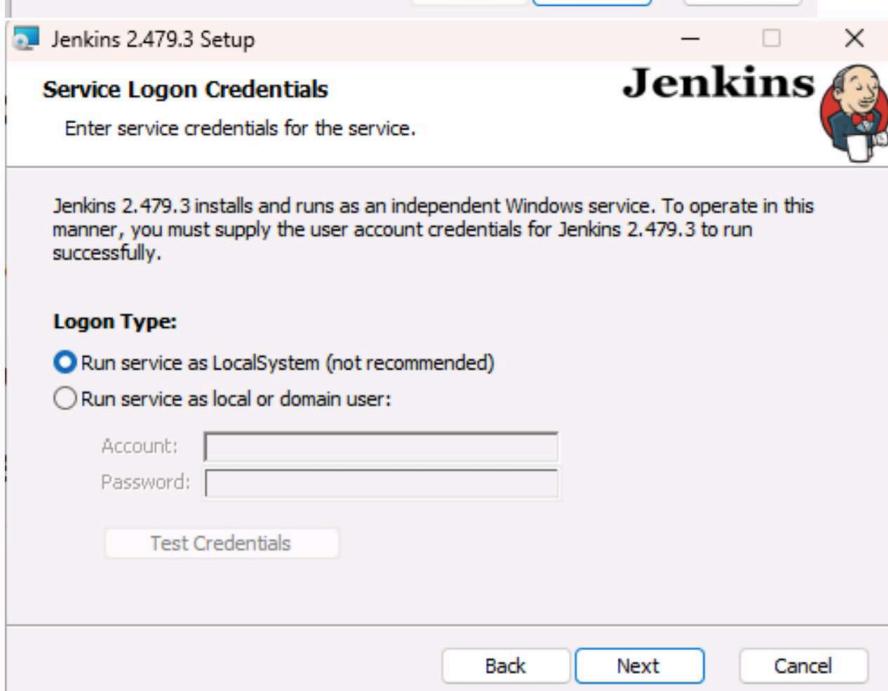
Steps:

1. Download and install Jenkins from jenkins.io.
2. Start Jenkins and unlock it using the initialAdminPassword.
3. Install suggested plugins.
4. Create the first admin user and configure the instance.
5. Create a new freestyle project.
6. Configure source code management and build steps.
7. Run the job and observe the console output.

Conclusion: We learned how to install and configure Jenkins and set up a build job. Jenkins automates repetitive tasks such as building and testing code, which helps teams implement continuous integration efficiently.

Would you like me to complete the rest of the experiments (4–10) in the same format?

Here are the structured details for **Experiments 4 to 6** from your DevOps lab manual:



ORACLE Products Industries Resources Customers Partners Developers Company   View Accounts 

Tools and resources Java downloads Java archive

JDK 23 JDK 21 GraalVM for JDK 23 GraalVM for JDK 21

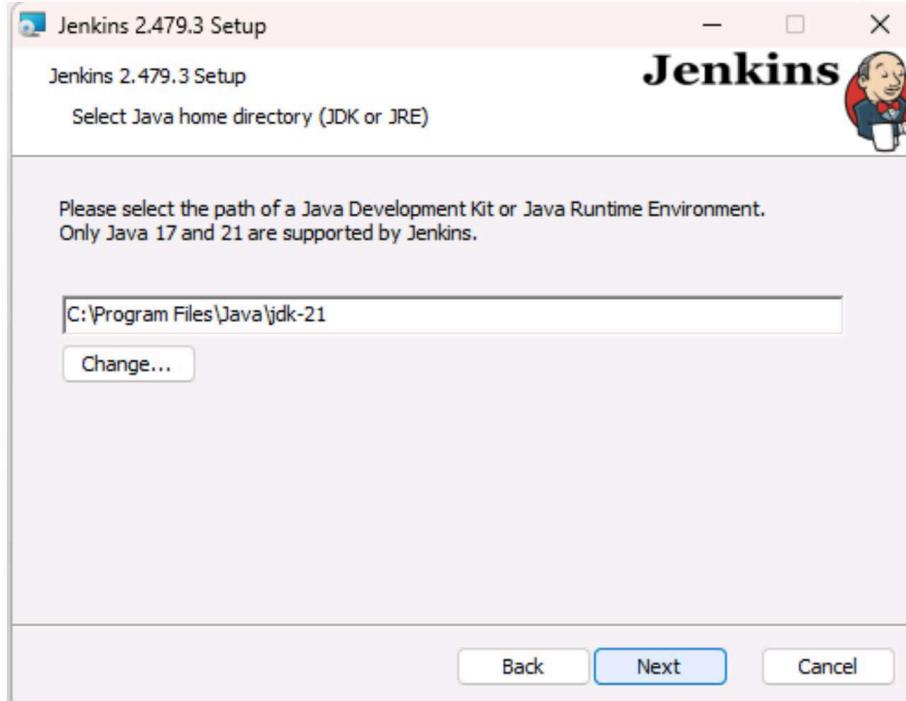
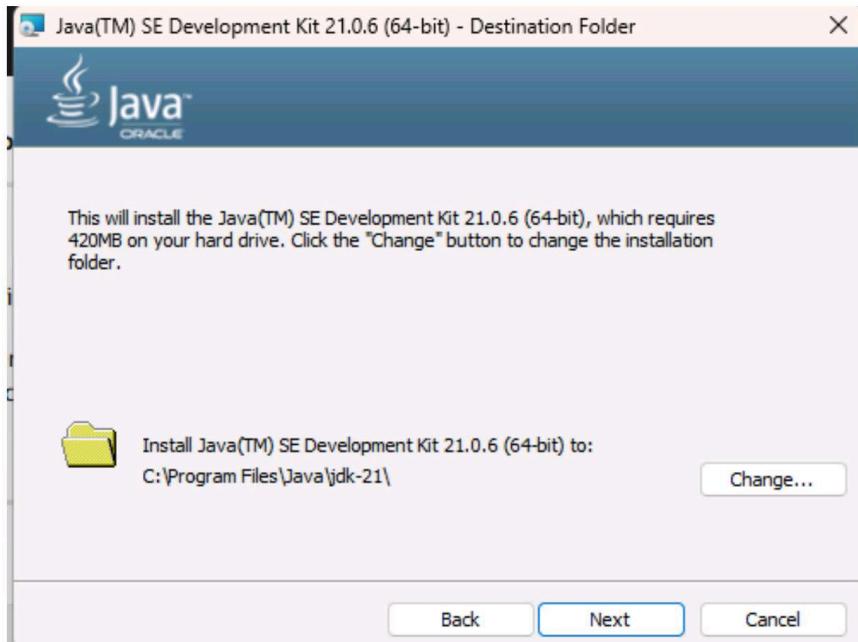
Java SE Development Kit 21.0.6 downloads

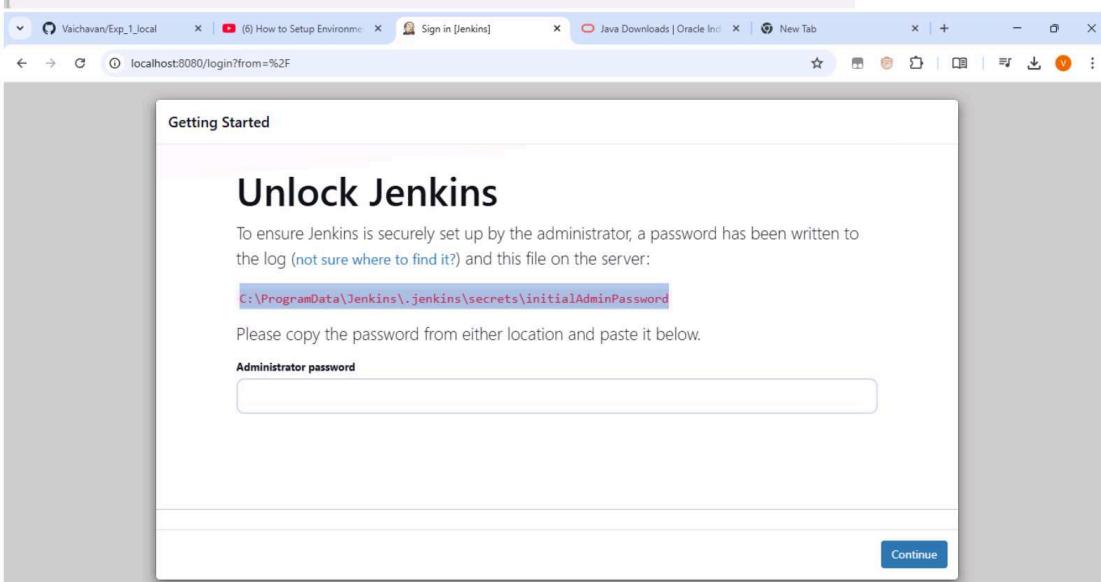
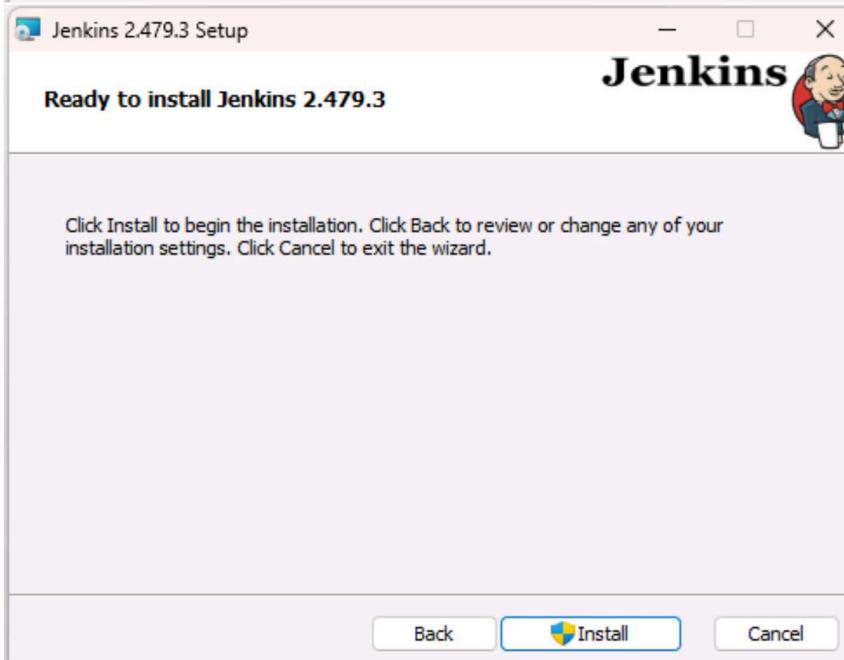
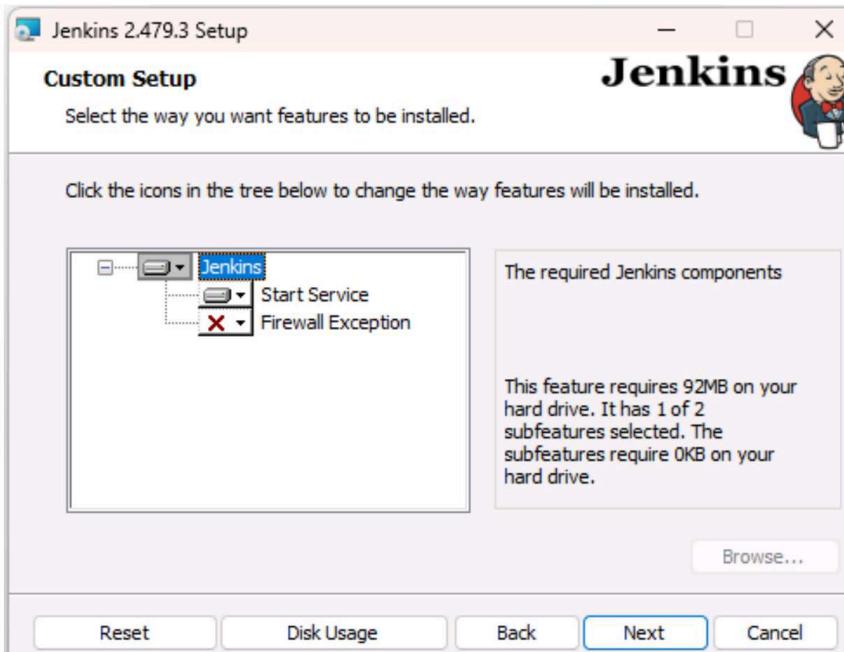
JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions \(NFTC\)](#).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License \(OTN\)](#) and production use beyond the limited free grants of the OTN license will [require a fee](#).

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	185.92 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)
x64 Installer	164.31 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.06 MB	https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)





localhost:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up, copy the password from the log (not sure where to find it?)

C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword

Please copy the password from:

Administrator password

.....

File Edit View

833826535ddf4bf58242acd8c619313d

ProgramData

C:\ProgramData

C:\ProgramData\Jenkins\secrets\initialAdminPassword

Name	Date modified	Type	Size
Alteryx	03-10-2024 11:59	File folder	
Autodesk	19-03-2024 19:36	File folder	

Vaichavan/Exp_1_local | (6) How to Setup Environment | Setup Wizard [Jenkins] | Java Downloads | Oracle Ind. | New Tab

localhost:8080

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.479.3

Getting Started

Getting Started

st8080

(6) How to Setup Environment x Setup Wizard [Jenkins] x Java Downloads | Oracle Inc x New Tab x +

Getting Started

Folders	✓ OWASP Markup Formatter	✓ Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	Dark Theme

** Ionicons API
 Folders
 OWASP Markup Formatter
 ** ASM API
 ** JSON Path API
 ** Structs
 ** Pipeline: Step API
 Build Timeout
 ** bouncycastle API
 ** Credentials
 ** Plain Credentials

** - required dependency

Jenkins 2.479.3

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

Jenkins 2.479.3

[Skip and continue as admin](#) [Save and Continue](#)

Getting Started

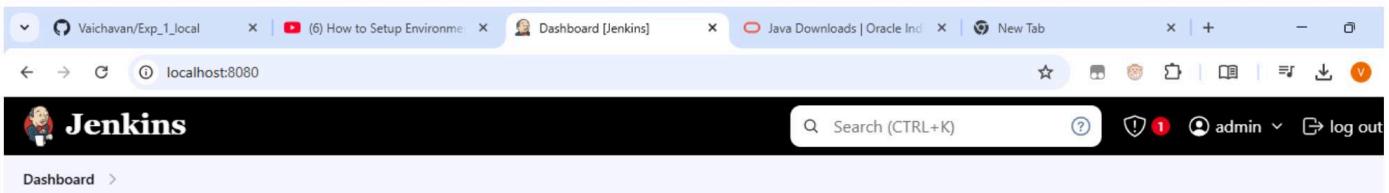
Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

[Start using Jenkins](#)



Vaichayan/Exp_1.local | (6) How to Setup Environment | Dashboard [Jenkins] | Java Downloads | Oracle Ind... | New Tab

localhost:8080

Jenkins

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent



Vaichayan | (6) How | Dashboard [Jenkins] | Java D... | New Tab

localhost:8080

Jenkins

Dashboard >

+ New Item

Build History

Manage Jenkins

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent



Vaichayan | (6) How | Dashboard [Jenkins] | Java D... | New Tab

localhost:8080

Jenkins

Dashboard >

+ New Item

Build History

Manage Jenkins

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

+ New Item

Build History

Manage Jenkins

General

Configure

General

Description

First job

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

This project is parameterized [?](#)

Throttle builds [?](#)

Execute concurrent builds if necessary [?](#)

Advanced ▾

Source Code Management

None

Git [?](#)

Configure

Build Steps

Execute Windows batch command [?](#)

Command

See [the list of available environment variables](#)

```
echo "This is my first Jenkins Vaibhav" %date% %time%
```

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save

Apply

Configure

-  General
-  Source Code Management
-  **Build Triggers**
-  Build Environment
-  Build Steps
-  Post-build Actions

Branch Specifier (blank for 'any') [?](#)

*/main

Add Branch

Repository browser [?](#)

(Auto)

Additional Behaviours

Add ▾



Dashboard > FirstJobs >

-  Status
-  Changes
-  Workspace
-  Build Now
-  Configure
-  Delete Project
-  Rename

FirstJobs

First job

Permalinks

Builds
No builds		



Jenkins

Dashboard > FirstJobs >

Status

</> Changes

FirstJobs

First job

Workspace

Permalinks

▷ Build Now

⚙ Configure

🗑 Delete Project

✍ Rename

Builds

... ↴

Today

✓ #2 4:39 PM

✓ #1 4:39 PM

le

🔍 ⭐



Search (CTRL+K)



ut



Console Output

Download

Copy

View as plain text

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\FirstJobs
[FirstJobs] $ cmd /c call C:\WINDOWS\TEMP\jenkins11649757884822622145.bat

C:\ProgramData\Jenkins\.jenkins\workspace\FirstJobs>echo "This is my first Jenkins Vaibhav" 04-02-2025
16:39:09.51
"This is my first Jenkins Vaibhav" 04-02-2025 16:39:09.51

C:\ProgramData\Jenkins\.jenkins\workspace\FirstJobs>exit 0
Finished: SUCCESS
```

4o- Experiment 4: Jenkins Pipeline for Application Deployment

Aim: To build the pipeline of jobs in Jenkins and create a pipeline script to deploy an application over the server.

Steps:

1. Open Jenkins Dashboard and click on "New Item".
2. Enter the pipeline name and select "**Pipeline**" as the type, then click **OK**.
3. Under the **Pipeline** tab, choose either:
 - **Pipeline script** to write inline code, or
 - **Pipeline script from SCM** to fetch a Jenkinsfile from a Git repository.
4. Write a pipeline script in Groovy syntax:

```

5. pipeline {
6.   agent any
7.   stages {
8.     stage('Build') {
9.       steps {
10.        echo 'Building...'
11.      }
12.    }
13.    stage('Test') {
14.      steps {
15.        echo 'Testing...'
16.      }
17.    }
18.    stage('Deploy') {
19.      steps {
20.        echo 'Deploying...'
21.      }
22.    }
23.  }
24. }
```

25. Save the pipeline job and click **Build Now**.
26. Monitor build stages in **Stage View**.

Conclusion: This experiment demonstrated how Jenkins pipelines automate the software delivery process. By defining steps in code, we created a CI/CD pipeline to build, test, and deploy applications reliably and repeatedly.

Dashboard [Jenkins] localhost:8080

Jenkins

Search (CTRL+K)

Dashboard >

+ New Item

All +

Build History

Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

Icon: S M L

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀️	FirstJobs	27 days #2	N/A	0.12 sec

Add description

localhost:8080/newJob

Jenkins

Dashboard > New Item

New Item

Enter an item name

Select an item type

 Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps like archiving artifacts and sending email notifications.

Jenkins

Dashboard > All > New Item

New Item

Enter an item name

Select an item type

 Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps like archiving artifacts and sending email notifications.

localhost:8080/view/all/newJob

Dashboard > All > New Item

Enter an item name

Select an item type

-  Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
-  Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

localhost:8080/job/Test/

Dashboard > Test > Configuration

- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- Trigger/call builds on other projects

Add build step

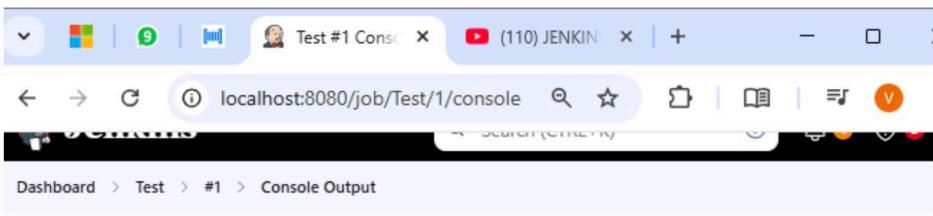
localhost:8080/view/all/newJob

Dashboard > All > New Item

Enter an item name

All pipeline +

S	W	Name ↴	Last Success	Last Failure	Last Duration
...	☀️	Build	N/A	N/A	N/A
...	☀️	Deploy	N/A	N/A	N/A
...	☀️	Test	N/A	N/A	N/A



The screenshot shows a browser window with the address bar set to `localhost:8080/job/Test/1/console`. The page displays the Jenkins dashboard with the 'Test' job selected. Under the job details, the 'Console Output' tab is active, showing the build log. The log output includes standard Jenkins startup messages and a command-line echo test.

```

Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Test
[Test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2267628309746816016.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Test>echo "Testing . . . ."
"Testing . . . ."

C:\ProgramData\Jenkins\.jenkins\workspace\Test>exit 0
Finished: SUCCESS

```

Console Output

View as plain text

```

Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Test
[Test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2267628309746816016.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Test>echo "Testing . . . ."
"Testing . . . ."

C:\ProgramData\Jenkins\.jenkins\workspace\Test>exit 0
Finished: SUCCESS

```

Dashboard > Build >

</> Changes

Console Output

[Download](#)

```
Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Build
[Build] $ cmd /c call C:\WINDOWS\TEMP\jenkins3562887581620601650.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Build>echo "Building ....."
"Building ....."

C:\ProgramData\Jenkins\.jenkins\workspace\Build>exit 0
Finished: SUCCESS
```

[All](#) [pipeline](#) [+](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	Build	29 sec #1	N/A	0.14 sec	▶
✓	☀	Deploy	11 sec #1	N/A	0.13 sec	▶
✓	☀	Test	1 min 54 sec #1	N/A	0.3 sec	▶

[Dashboard](#) > [Build](#) > [Configuration](#)

- [Dashboard](#) > [Build](#) >
- [Status](#)
- [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Configure](#)
- [Delete Project](#)
- [Rename](#)

≡ Build other projects ?

Projects to build

! No project specified

- Trigger only if build is stable
- Trigger even if the build is unstable
- Trigger even if the build fails

Add post-build action ▾

[Save](#)
[Apply](#)

Post-build Actions

≡ **Build other projects** ?

Projects to build

Test

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Add post-build action ▾

Save

Apply

▼  |  |   Test [Jenkins] X + - □

← → C ⓘ localhost:8080/job/Test/

Dashboard > Test >

Filter

Today

#1 7:27 PM

Test

 Add description

Upstream Projects

 [Build](#)

Permalinks

- [Last build \(#1\), 4 min 9 sec ago](#)
- [Last stable build \(#1\), 4 min 9 sec ago](#)
- [Last successful build \(#1\), 4 min 9 sec ago](#)
- [Last completed build \(#1\), 4 min 9 sec ago](#)

Post-build Actions

Build other projects [?](#)

Projects to build

Deploy,

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Add post-build action [▼](#)

[Save](#)

[Apply](#)

localhost:8080/job/Deploy/

Dashboard > Deploy >

Filter

Today

#1 7:28 PM

Deploy

Upstream Projects

[Test](#)

Permalinks

- Last build (#1), 3 min 38 sec ago
- Last stable build (#1), 3 min 38 sec ago
- Last successful build (#1), 3 min 38 sec ago
- Last completed build (#1), 3 min 38 sec ago

Enter an item name

Pipeline_Scripted

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Dashboard > All > New Item

Enter an item name

Pipeline_Scripted_Vaibhav1261

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Dashboard > All > New Item

Enter an item name

Pipeline_Scripted_Vaibhav1261

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



OK

Dashboard > Pipeline_Scripted_Vaibhav1261 > Configuration

Pipeline script

Script ?

```

1> pipeline {
2>     agent any
3>     stages{
4>         stage( "Build_stage" ) {
5>             steps{
6>                 echo "Building Project. . ."
7>             }
8>         stage ( "Test_stage" ) {
9>             steps{
10>                 echo "Testing Project. . ."
11>             }
12>         stage( "Deploy _stage" ) {
13>             steps{
14>                 echo "Deploying Project. . ."
15>             }
16>         }
17>         post{
18>             always{
19>                 echo "Thank You"
20>             }
21>         }
22>     }
23> }
```



Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced ▾

Save

Apply

Build Pipeline_Scripted_Vaibhav1261

id	pipeline
#1	<p>Start Build_stage Test_stage Deploy_stage Post Actions End</p> 

```
1 - pipeline {
2     agent any
3     stages{
4         stage( "Build_stage" ) {
5             steps{
6                 echo "Building Project. . ."
7             }
8         }
9         stage( "Test_stage" ) {
10            steps{
11                echo "Testing Project. . ."
12            }
13        }
14        stage( "Deploy _ stage" ) {
15            steps{
16                echo "Deploying Project. . ."
17            }
18        }
19        post{
20            failure{
21                echo "Thank You"
22            }
23        }
24    }
25}
```



Use Groovy Sandbox 

[Pipeline Syntax](#)

Advanced

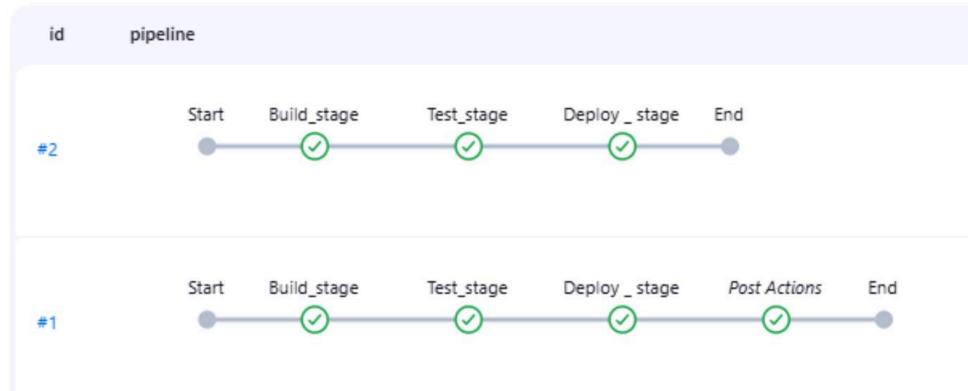
[Save](#)

[Apply](#)

Build Pipeline_Scripted_Vaibhav1261

Build

Configure



← → ⌂ github.com/Vaichavan/exp4_vaibhav_jenpipe/blob/main/jenkinsfile ↻

Files
exp4_vaibhav_jenpipe/jenkinsfile

main
+ Q

Go to file
t

README.md
hello.c
jenkinsfile

Vaichavan Create jenkinsfile

Code Blame 20 lines (20 loc) · 428 Bytes Code 55% faster with GitHub Copilot

```

1 pipeline {
2     agent any
3     stages{
4         stage( "Build_stage" ) {
5             steps{
6                 echo "Building Project. . ."
7             }
8         stage ( "Test_stage" ) {
9             steps{
10                 echo "Testing Project. . ."
11             }
12         stage( "Deploy _stage" ) {
13             steps{
14                 echo "Deploying Project. . ."
15             }
16         }
17         post{
18             failure{
19                 echo "Thank You"
20             }
21         }
22     }
23 }
```

Enter an item name

Pipeline_Declarative_Vaibhav1261

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Definition

Pipeline script

Pipeline script

Pipeline script from SCM

| Try Sample Pipeline

Go to file



<> Code ▾

Local

Codespaces

Clone



HTTPS SSH GitHub CLI

Copy url to clipboard

https://github.com/Vaichavan/exp4_vaibhav_jenpipe.git



Clone using the web URL.

Open with GitHub Desktop

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Vaichavan/exp4_vaibhav_jenpipe.git

```
⚠ Failed to connect to repository : Command "git.exe ls-remote -h -- https://github.com/Vaichavan/exp4_vaibhav_jenpipe.git HEAD" returned status code 128:
stdout:
stderr: fatal: Cannot prompt because user interactivity has been disabled.
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/Vaichavan/exp4_vaibhav_jenpipe.git/'
```

Credentials ?

System variables

Variable	Value
gt	"C:\Users\vaibh\OneDrive\Desktop\roll\new2.c"
NUMBER_OF_PROCESSORS	8
OnlineServices	Online Services
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
platformcode	KV

Edit environment variable

```
C:\Program Files\dotnet\
C:\MinGW\bin
C:\Users\vaibh\OneDrive\Desktop\helloworld.c
C:\Users\vaibh\anaconda3
C:\Users\vaibh\anaconda3\Scripts
C:\Users\vaibh\anaconda3\Library\bin
C:\Users\vaibh\anaconda3\Library\mingw-w64\bin
C:\Users\vaibh\anaconda3\Library\usr\bin
C:\Program Files\MySQL\MySQL Server 5.5\bin
C:\xampp\php
C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\Program Files\HP\HP One Agent
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Java\jdk-21\bin
C:\Program Files\Git\bin
```

≡  Settings



Vaichavan (Vaichavan)

Your personal account

 Public profile

 Account

Pu

Now



Settings / Developer Settings

 Type ↗

 GitHub Apps

 OAuth Apps

 Personal access tokens

Fine-grained tokens

 Preview

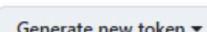
Tokens (classic)

Personal access tokens (classic)



No personal access token

Need an API token for scripts or testing? Generate a persona GitHub API.

 Generate new token ▾

Generate new token  Beta

Fine-grained, repo-scoped

Generate new token (classic)

For general use

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of : over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

generate-classic-token

What's this token for?

Expiration *

30 days ▾ The token will expire on Wed, Apr 2 2025

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows

Jenkins Credentials Provider: Jenkins

jenkins

Global credentials (unrestricted)

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

vaibhavschavan26@gmail.com

Treat username as secret ?

Password ?

.....

ID ?

webs123

Jenkins Credentials Provider: Jenkins

vaibhavschavan26@gmail.com

Treat username as secret ?

Password ?

.....

ID ?

webs123

An internal unique ID by which these credentials are identified by Jenkins. Useful to specify explicitly when creating jobs using visual forms.

Description ?

webs123

≡  Settings / Developer Settings

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

 GitHub Apps

 OAuth Apps

 Personal access tokens

Fine-grained tokens

Preview

Tokens (classic)

Personal access tokens (classic)

Tokens you have generated that can be used to access the [GitHub API](#)

 Make sure to copy your personal access token now. You won't see it again!

 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 

← → C ⓘ localhost:8080/job/Pipeline... 🔒 🔍 ⚡ 🌐 📁 📄 📂 📈 🎯 ⋮

Dashboard > Pipeline_Declarative_Vaibhav1261 > Configuration

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?



Credentials ?

webs123



+ Add

Advanced ▾

Add Repository

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main



Add Branch

Repository browser ?

(Auto)



Additional Behaviours

Add ▾

Script Path ?

Jenkinsfile

Lightweight checkout ?

Pipeline Syntax

 Status

 Changes

 Build Now

 Configure

 Delete Pipeline

 Stages

 Rename

 Pipeline Syntax

Builds

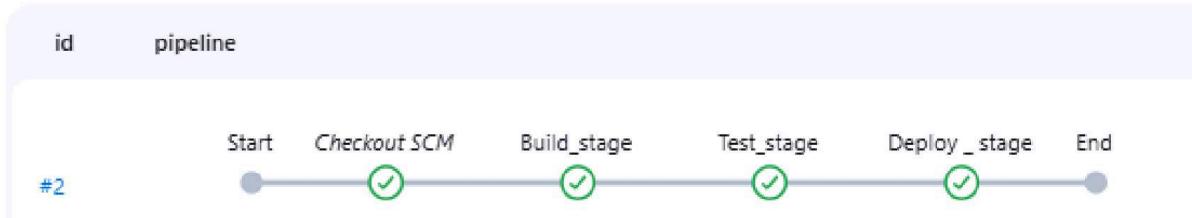
... ▾

Today

 #1 8:26 PM



Build Pipeline_Declarative_Vaibhav1261



localhost:8080

Dashboard >

All pipeline +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀️	Build	1 hr 18 min #1	N/A	0.14 sec
✓	☀️	Deploy	1 hr 18 min #1	N/A	0.13 sec
✓	☀️	Pipeline_Declarative_Vaibhav1261	17 min #2	N/A	7.7 sec
✓	☀️	Pipeline_Scripted_Vaibhav1261	58 min #2	N/A	1.5 sec
✓	☀️	Test	1 hr 20 min #1	N/A	0.3 sec

Icon: S M L

... more

5o- Experiment 5: Jenkins on Tomcat Server

Aim: To install Tomcat server on Windows and run Jenkins over the Tomcat server.

Steps:

1. Download **Apache Tomcat** from tomcat.apache.org and extract it.
2. Run startup.bat from the **Tomcat/bin** directory.
3. Verify the server is running at <http://localhost:8080>.
4. Download the **Jenkins WAR file** from jenkins.io.
5. Place jenkins.war in the **Tomcat/webapps** folder.
6. Restart Tomcat server; Jenkins will deploy as a web app.
7. Access Jenkins at <http://localhost:8080/jenkins> and complete the initial setup.
8. Create and test a job to verify Jenkins is running under Tomcat.

Conclusion: Running Jenkins on a Tomcat server allowed us to understand how web applications are deployed in real-world DevOps setups. We learned to host Jenkins as a web app and integrate it with Tomcat for better scalability and centralized management.

Dashboard [Jenkins] Apache Tomcat® - Apache Tomcat

tomcat.apache.org/download-90.cgi

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)

Downloads

Copying 748 items from apache-tomcat-9.0.1... to apache-tomcat-9.0.10...
61% complete

Speed: 383 KB/s

Name: OpenSans700.woff
Time remaining: About 20 seconds
Items remaining: 488 (1.47 MB)

Fewer details

conf

Downloads > apache-tomcat-9.0.102-windows-x64 > apache-tomcat-9.0.102 > conf

Name	Date modified	Type	Size
catalina.policy	11-03-2025 10:55	POLICY File	13 KB
catalina	11-03-2025 10:55	Properties Source ...	8 KB
context	11-03-2025 10:55	Microsoft Edge H...	2 KB
jaspic-providers	11-03-2025 10:55	Microsoft Edge H...	2 KB
jaspic-providers.xsd	11-03-2025 10:55	XML Schema File	3 KB
logging	11-03-2025 10:55	Properties Source ...	4 KB
server	11-03-2025 10:55	Microsoft Edge H...	9 KB
tomcat-users	11-03-2025 10:55	Microsoft Edge H...	3 KB



File Edit Selection View Go Run ... ← → Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

tomcat-users.xml

```
C: > Users > vaibh > Downloads > apache-tomcat-9.0.102-windows-x64 > apache-tomcat-9.0.102 > conf > tomcat-users.xml
49  <!--
52  |   <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
53  |   <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
54  |   <user username="role1" password="<must-be-changed>" roles="role1"/>
55  |-->
56  |   <role rolename="manager-gui"/>
57  |   <role rolename="manager-script"/>
58  |   <role rolename="manager-jmx"/>
59  |   <role rolename="manager-status"/>
60  |   <user username="admin" password="admin#123" roles="manager-status,manager-gui,manager-script"/>
61  |</tomcat-users>
62
```

bin

Downloads > apache-tomcat-9.0.102-windows-x64 > apache-tomcat-9.0.102 > bin

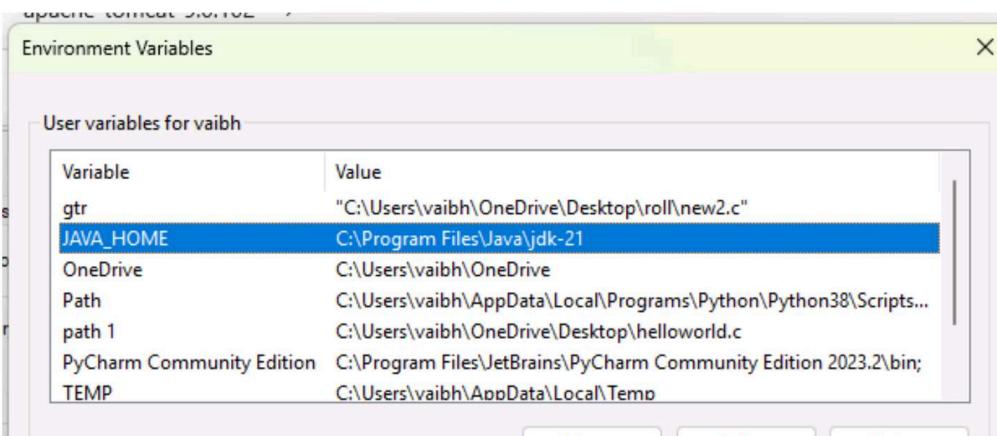
Name	Date modified	Type	Size
startup	11-03-2025 10:55	Windows Batch File	2 KB
startu	11-03-2025 10:55	sh_auto_file	2 KB

File Edit Selection View Go Run ... ← →

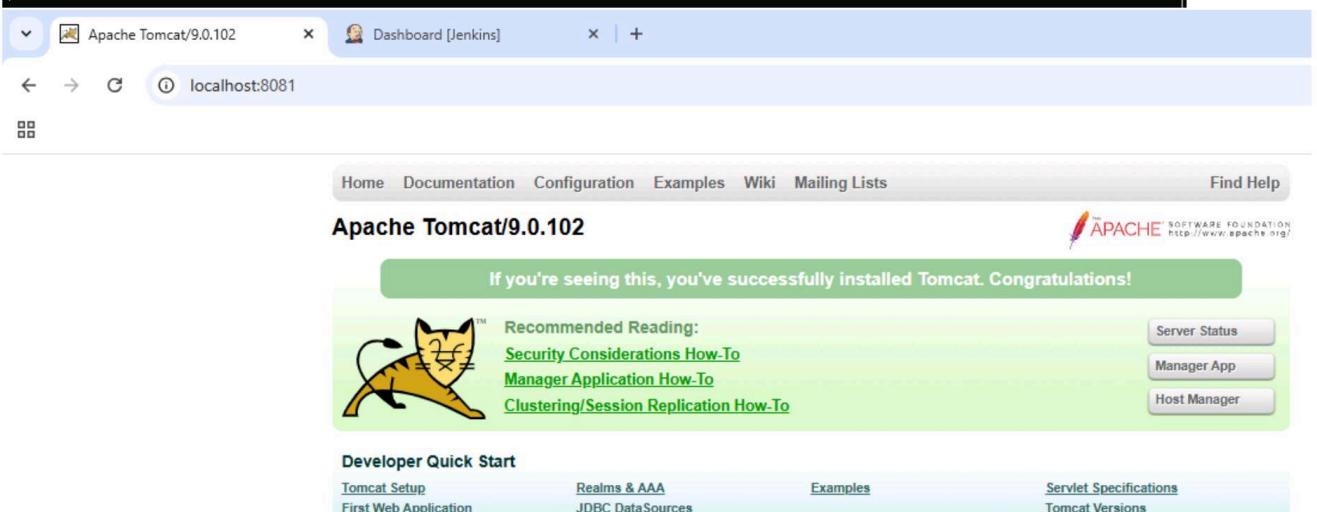
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

server.xml

```
C: > Users > vaibh > Downloads > apache-tomcat-9.0.102-windows-x64 > apache-tomcat-9.0.102 > conf > server.xml
22  <Server port="8005" shutdown="SHUTDOWN">
23
24      <Service name="Catalina">
25
26          <!--The connectors can use a shared executor, you can
27          |<!--
28          |   <Executor name="tomcatThreadPool" namePrefix="catalina-"
29          |       maxThreads="150" minSpareThreads="4"/>
30          |-->
31
32
33          <!-- A "Connector" represents an endpoint by which requests
34          |    and responses are returned. Documentation at :
35          |        Java HTTP Connector: /docs/config/http.html
36          |        Java AJP Connector: /docs/config/ajp.html
37          |        APR (HTTP/AJP) Connector: /docs/apr.html
38          |        Define a non-SSL/TLS HTTP/1.1 Connector on port
39          |-->
40
41          <Connector port="8081" protocol="HTTP/1.1"
42              connectionTimeout="20000"
43              redirectPort="8443"
44              maxParameterCount="1000"
45              />
```



```
Tomcat -> 11-Mar-2025 11:41:28.200 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-8081"]
11-Mar-2025 11:41:28.242 INFO [main] org.apache.catalina.startup.Catalina.load Server initialization in [920] milliseconds
11-Mar-2025 11:41:28.309 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
11-Mar-2025 11:41:28.315 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apache Tomcat/9.0.102]
11-Mar-2025 11:41:28.333 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/docs]
11-Mar-2025 11:41:28.760 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/docs] has finished in [425] ms
11-Mar-2025 11:41:28.760 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/examples]
11-Mar-2025 11:41:29.620 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/examples] has finished in [860] ms
11-Mar-2025 11:41:29.622 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/host-manager]
11-Mar-2025 11:41:29.719 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/host-manager] has finished in [99] ms
11-Mar-2025 11:41:29.719 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/manager]
11-Mar-2025 11:41:29.816 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/manager] has finished in [97] ms
11-Mar-2025 11:41:29.818 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/ROOT]
11-Mar-2025 11:41:29.882 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:/Users/vaibh/Downloads/apache-tomcat-9.0.102/webapps/ROOT] has finished in [64] ms
11-Mar-2025 11:41:29.893 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8081"]
11-Mar-2025 11:41:29.955 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [1709] milliseconds
```



Apache Tomcat/9.0.102

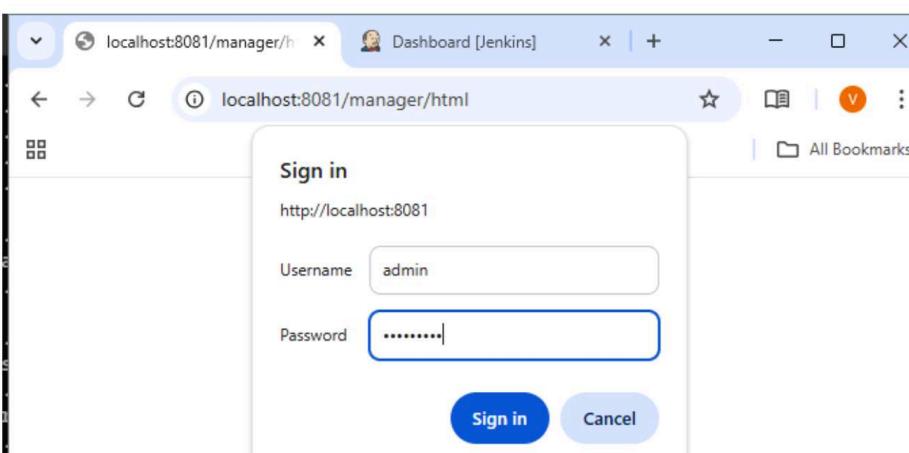
If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)



Apache Tomcat/9.0.102 Dashboard [Jenkins] Download Apache Maven – M

maven.apache.org/download.cgi

User Centre

- Plugin Developer Centre
- Maven Repository Centre
- Maven Developer Centre
- Books and Resources
- Security

archive if you intend to build Maven yourself.
In order to guard against corrupted downloads/installations, it is highly recommended by Apache Maven developers.

Link

Binary tar.gz archive	apache-maven-3.9.9-bin.tar.gz
Binary zip archive	apache-maven-3.9.9-bin.zip

Installed plugins - Plugins [Jenk] localhost:8080/manage/pluginManager/installed

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Plugins

Plugins

Updates 56

Available plugins

Installed plugins

Advanced settings

Search: mav

Name ↓	Enabled
Maven Integration plugin 3.25	<input checked="" type="checkbox"/> 3.25 X

This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.
[Report an issue with this plugin](#)

Installed plugins - Plugins [Jenk] localhost:8080/manage/pluginManager/installed

Dashboard > Manage Jenkins > Plugins

Plugins

Updates 56

Available plugins

Installed plugins

Advanced settings

Search: cor

Report an issue with this plugin

Deploy to container Plugin 1.16	<input checked="" type="checkbox"/> X
Email Extension Plugin 1876.v28d8d38315b_d	<input checked="" type="checkbox"/> X

This plugin allows you to deploy a war to a container after a successful build.
Glassfish 3.x remote deployment
[Report an issue with this plugin](#)

This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says
[Report an issue with this plugin](#)

Apache Tomcat/9.0.102 warfileweb_1261 Config [Jenkins] jenkins plugin man localhost:8080/view/expt/job/warfileweb_1261/configure

Dashboard > expt > warfileweb_1261 > Configuration

Configure

Source Code Management

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

Credentials [?](#)

[+ Add](#)

[Advanced](#)

Environment Variables

User variables for vaibh

Variable	Value
gtr	"C:\Users\vaibh\OneDrive\Desktop\roll\new2.c"
JAVA_HOME	C:\Program Files\Java\jdk-21
MAVEN_HOME	C:\Users\vaibh\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9
OneDrive	C:\Users\vaibh\OneDrive
Path	C:\Users\vaibh\AppData\Local\Programs\Python\Python38\Scripts;C:\Users\vaibh\Ap...
path 1	C:\Users\vaibh\OneDrive\Desktop\helloworld.c
PvCharm Community Edition	C:\Program Files\JetBrains\PvCharm Community Edition 2023.2\bin:

New... Edit... Delete

Maven installations

Maven installations  Edited

[Add Maven](#)

Maven

Name

LocalMaven

MAVEN_HOME

C:\Users\vaibh\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9

Install automatically 

[Add Maven](#)

[Save](#)

[Apply](#)

Add pre-build step 

Configure

 General

 Source Code Management

 Build Triggers

 Build Environment

 Pre Steps

 Build

 Post Steps

 Build Settings

 Post-build Actions

Build

Root POM 

pom.xml

Goals and options 

clean package

Advanced 

Post Steps

Run only if build succeeds

Run only if build succeeds or is unstable

Run regardless of build result

[Save](#)

[Apply](#)

Dashboard > expt > warfileweb_1261 > Configuration

Post Steps

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps**
- Build Settings
- Post-build Actions

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Deploy artifacts to Maven repository
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container**
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects
- Delete workspace when build is done

Add post-build action ^

Save **Apply**

Context path ? warfileweb_1261 Config [Jenkins] +

localhost:8080/job/warfileweb_1261/configure

Dashboard > warfileweb_1261 > Configuration

Containers

Add Container ^

Filter

- GlassFish 2.x
- GlassFish 3.x
- GlassFish 4.x
- JBoss AS 3.x
- JBoss AS 4.x
- JBoss AS 5.x**
- JBoss AS 6.x
- JBoss AS 7.x
- Tomcat 4.x Remote
- Tomcat 5.x Remote
- Tomcat 6.x Remote
- Tomcat 7.x Remote
- Tomcat 8.x Remote
- Tomcat 9.x Remote

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions**

WAR/EAR files ?

Context path ?

Containers

Tomcat 9.x Remote

Credentials

+ Add

Tomcat URL ?

Advanced ▾

Save **Apply**

localhost:8080/view/expt/job/warfileweb_1261/configure

Dashboard > expt > warfileweb_1261 > Configuration

Configure

Jenkins Credentials Provider: Jenkins

General

Source Code Management

Build Triggers

Build Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: admin

Treat username as secret

Password:

ID: vaibhav_cat

Downloads > apache-tomcat-9.0.102 > webapps >

New |剪切|复制|粘贴|||| Sort | View |...

Name	Date modified	Type
Desktop		
Downloads		
Pictures		
Documents		
Music		
host-manager	11-03-2025 10:56	File folder
manager	11-03-2025 10:56	File folder
examples	11-03-2025 10:56	File folder
docs	11-03-2025 10:55	File folder
ROOT	11-03-2025 10:55	File folder



localhost:8080/job/warfileweb_1261/

Jenkins

Dashboard > warfileweb_1261 >

- [Status](#)
- </> [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Configure](#)
- [Delete Maven project](#)
- [Modules](#)
- [Rename](#)

Maven project warfileweb_1261

Permalinks

- [Last build \(#2\), 6 hr 0 min ago](#)
- [Last stable build \(#2\), 6 hr 0 min ago](#)
- [Last successful build \(#2\), 6 hr 0 min ago](#)
- [Last failed build \(#1\), 6 hr 2 min ago](#)
- [Last unsuccessful build \(#1\), 6 hr 2 min ago](#)
- [Last completed build \(#2\), 6 hr 0 min ago](#)

Builds

#	Build Number	Last Published
<input checked="" type="checkbox"/>	#2	12:55PM
<input type="checkbox"/>	#1	12:53PM

File Explorer

webapps

Name	Date modified	Type
ROOT	11-03-2025 10:55	File folder
docs	11-03-2025 10:55	File folder
examples	11-03-2025 10:56	File folder
host-manager	11-03-2025 10:56	File folder
manager	11-03-2025 10:56	File folder
webAppExample	11-03-2025 12:55	File folder
webAppExample.war	11-03-2025 12:55	WAR File

localhost:8081/webAppExample/

Welcome to KK JavaTutorials..

Person 1

V

Vaibhav Chavan
vaibhavschavan26@gmail.com

Dashboard > expt > warfileweb_1261 > #2 > Console Output

```
[INFO] Processing war project
[INFO] Copying webapp resources [C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\src\main\webapp]
[INFO] Building war: C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\target\webappExample.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.458 s
[INFO] Finished at: 2025-03-11T12:55:53+05:30
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\pom.xml to com.javarticles.webapp/webappExample/0.0.1-SNAPSHOT/webappExample-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\target\webappExample.war to com.javarticles.webapp/webappExample/0.0.1-SNAPSHOT/webappExample-0.0.1-SNAPSHOT.war
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\target\webappExample.war to container Tomcat 9.x Remote with context webappExample
[C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\target\webappExample.war] is not deployed. Doing a fresh deployment.
Deploying [C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\target\webappExample.war]
Finished: SUCCESS
```

Jenkins

Dashboard > expt > warfileweb_1261 > #2 > Console Output

Status Changes Console Output Edit Build Information Delete build '#2' Timings Git Build Data Redeploy Artifacts See Fingerprints Previous Build

Console Output

Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\warfileweb_1261\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/kishancs2020/webAppExample.git # timeout=10
Fetching upstream changes from https://github.com/kishancs2020/webAppExample.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/kishancs2020/webAppExample.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision b676ca23a034d3818c8b863734f5d6a3aa9b7ee4 (refs/remotes/origin/master)
> git.exe config core.sparseCheckout # timeout=10
> git.exe checkout -f b676ca23a034d3818c8b863734f5d6a3aa9b7ee4 # timeout=10
Commit message: "Just Modified JSP.."

6o- Experiment 6: Selenium Testing in Jenkins Using Maven

Aim: To set up and run Selenium tests in Jenkins using Maven.

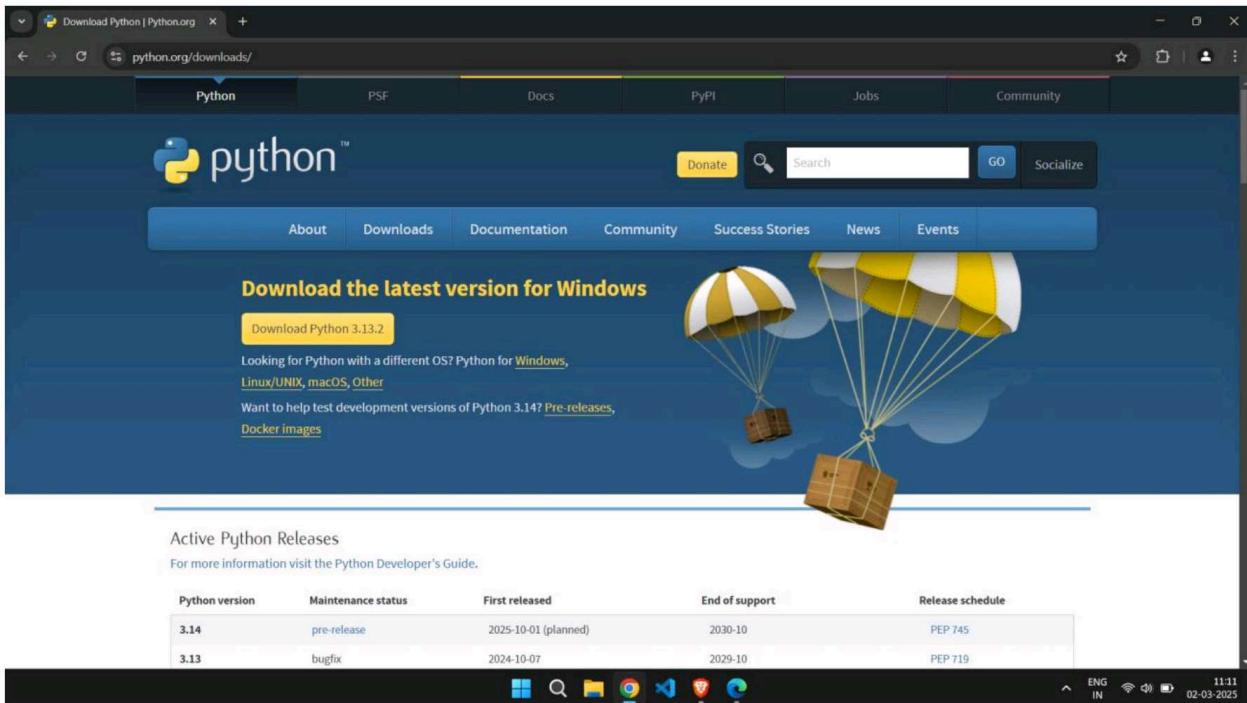
Steps:

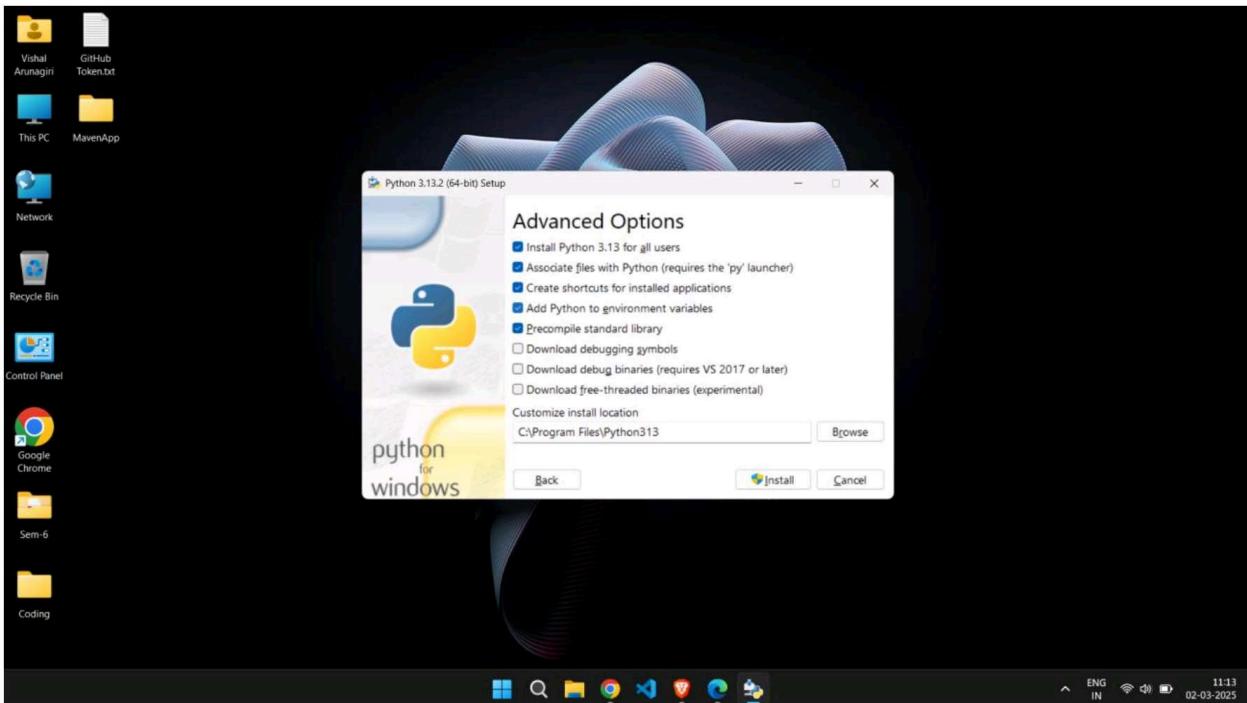
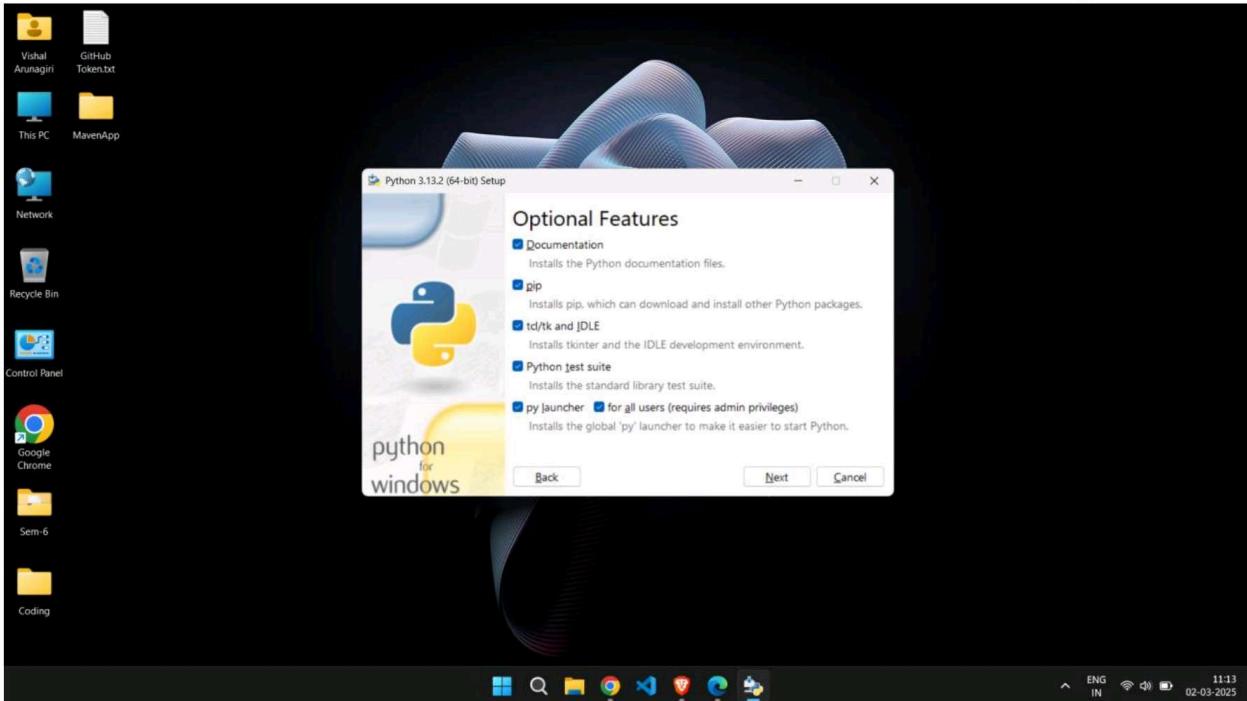
1. Install **Java, Python, Selenium, and Maven**.
2. Set up a Python virtual environment and install Selenium:
 3. python -m venv venv
 4. source venv/bin/activate
5. pip install selenium
6. Write a Selenium test in Python (e.g., search on Google):
 7. from selenium import webdriver
 8. from selenium.webdriver.common.keys import Keys
 9. driver = webdriver.Chrome()
 10. driver.get("https://www.google.com")
 11. driver.find_element_by_name("q").send_keys("DevOps" + Keys.ENTER)
 12. driver.close()
13. Save the test and ensure ChromeDriver is in the project path.
14. In Jenkins, create a **Freestyle Project**.
15. In the **Build** section, add an **Execute shell** build step:
 16. pip install selenium
 17. python sample_test.py
18. Save and **Build Now**.

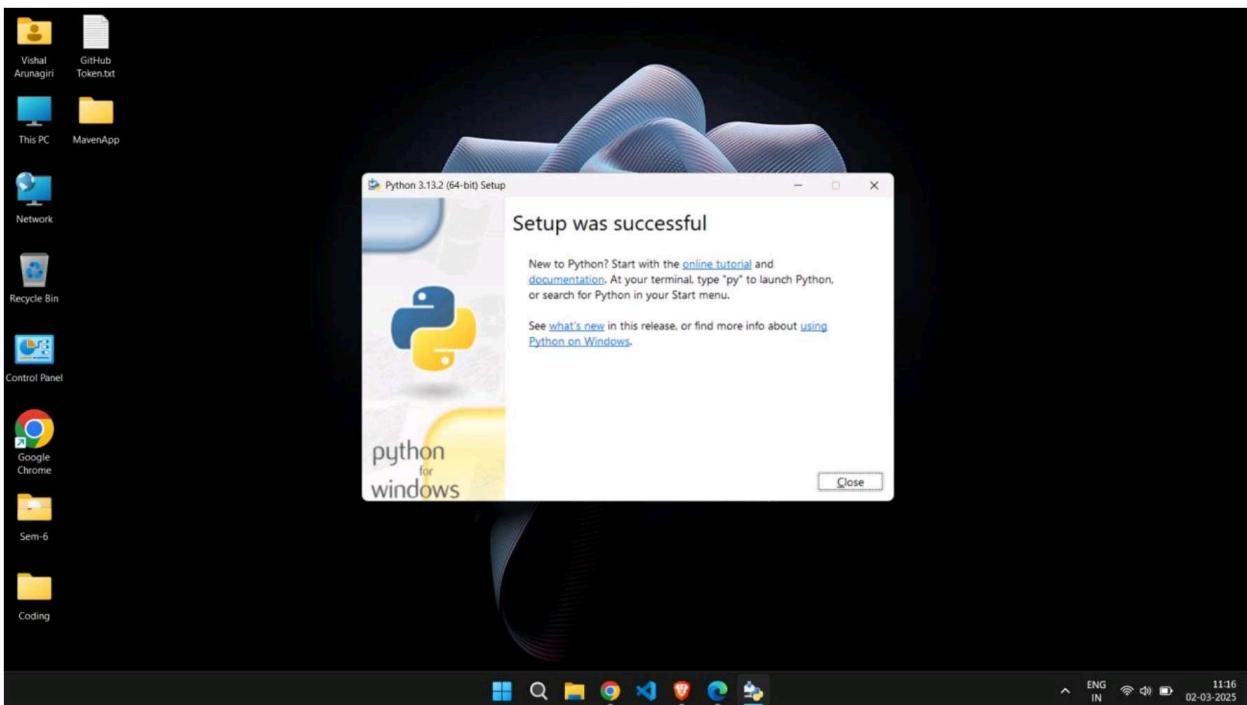
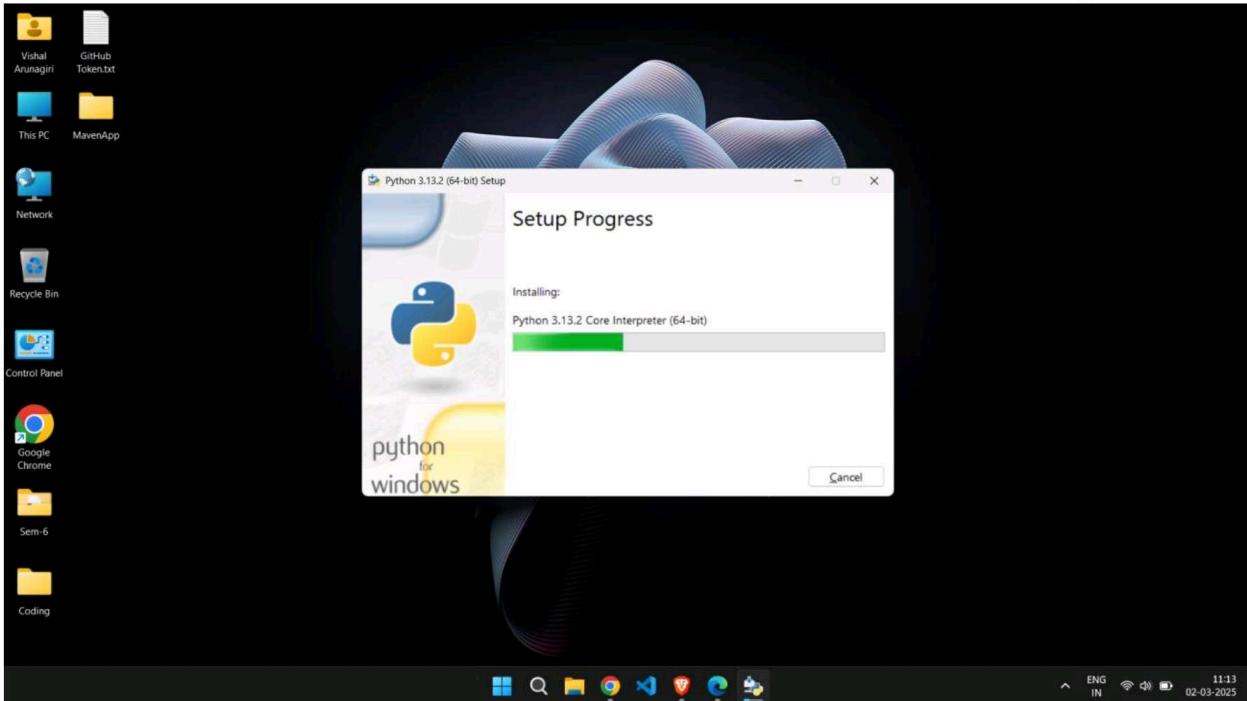
Conclusion: This experiment provided practical knowledge of integrating Selenium with Jenkins to automate web application testing. Using Maven and Python, we ran browser-based UI tests, supporting continuous testing within a CI/CD pipeline.

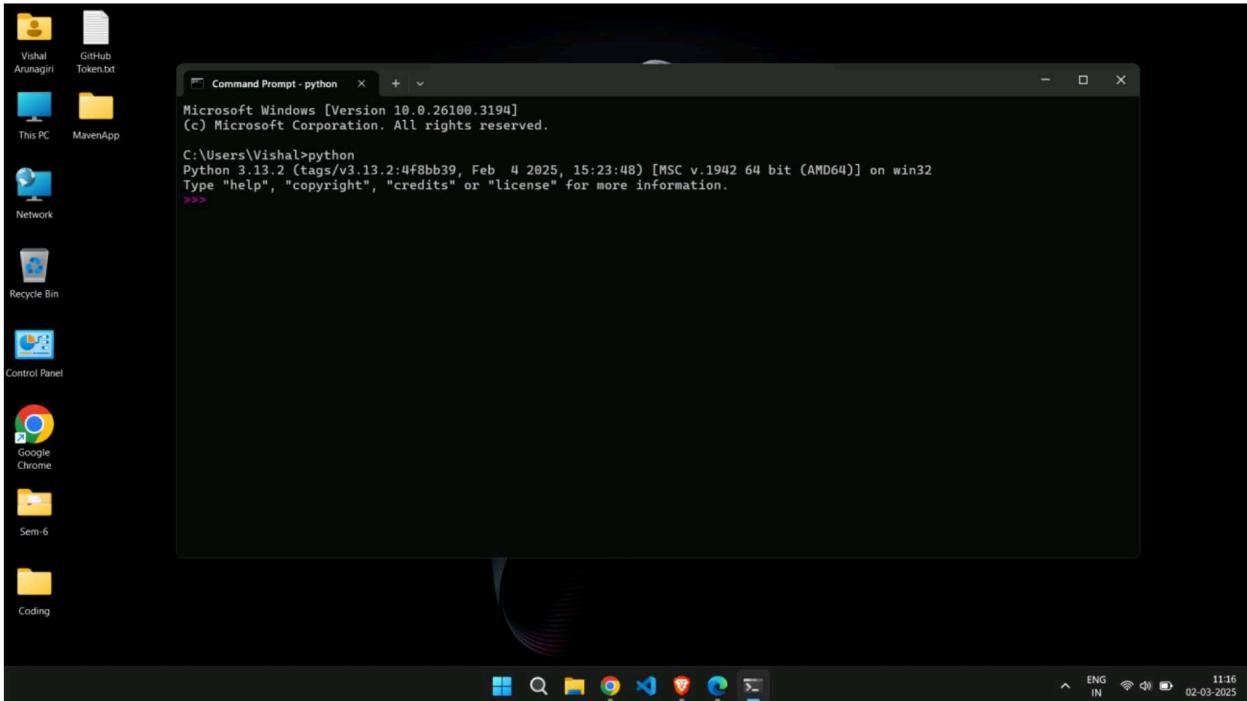
Experiment - 6

Download & Install Python

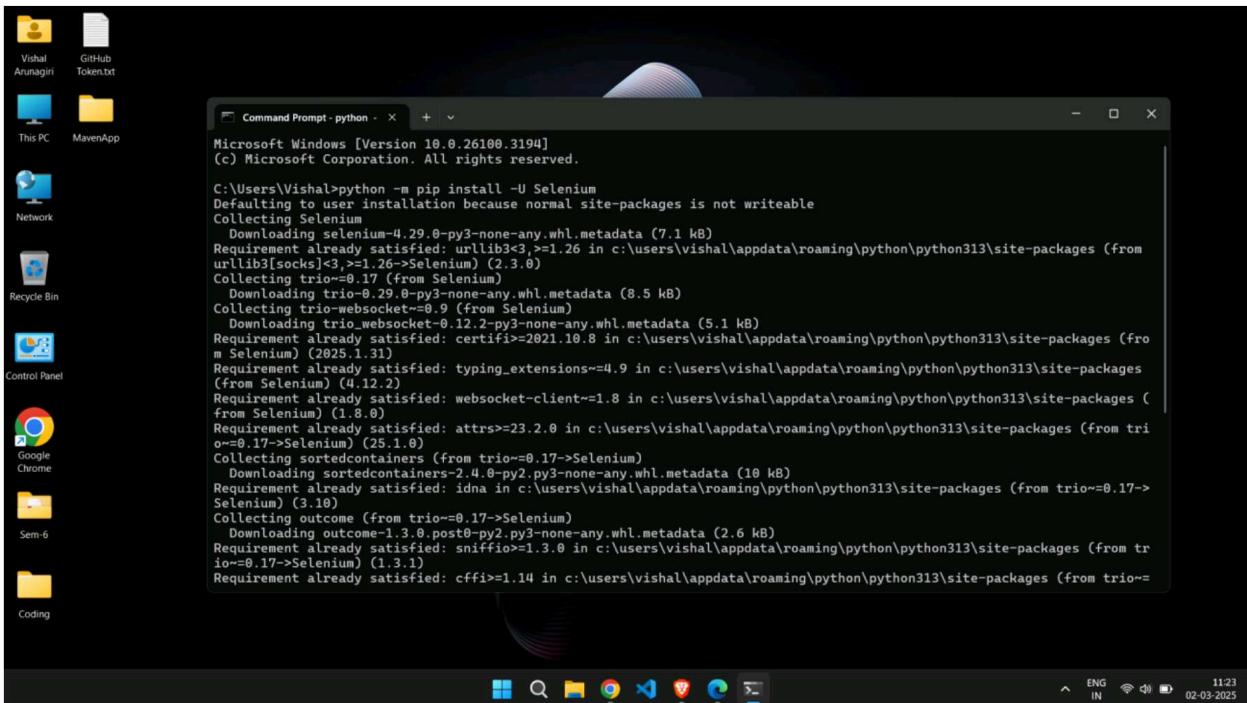


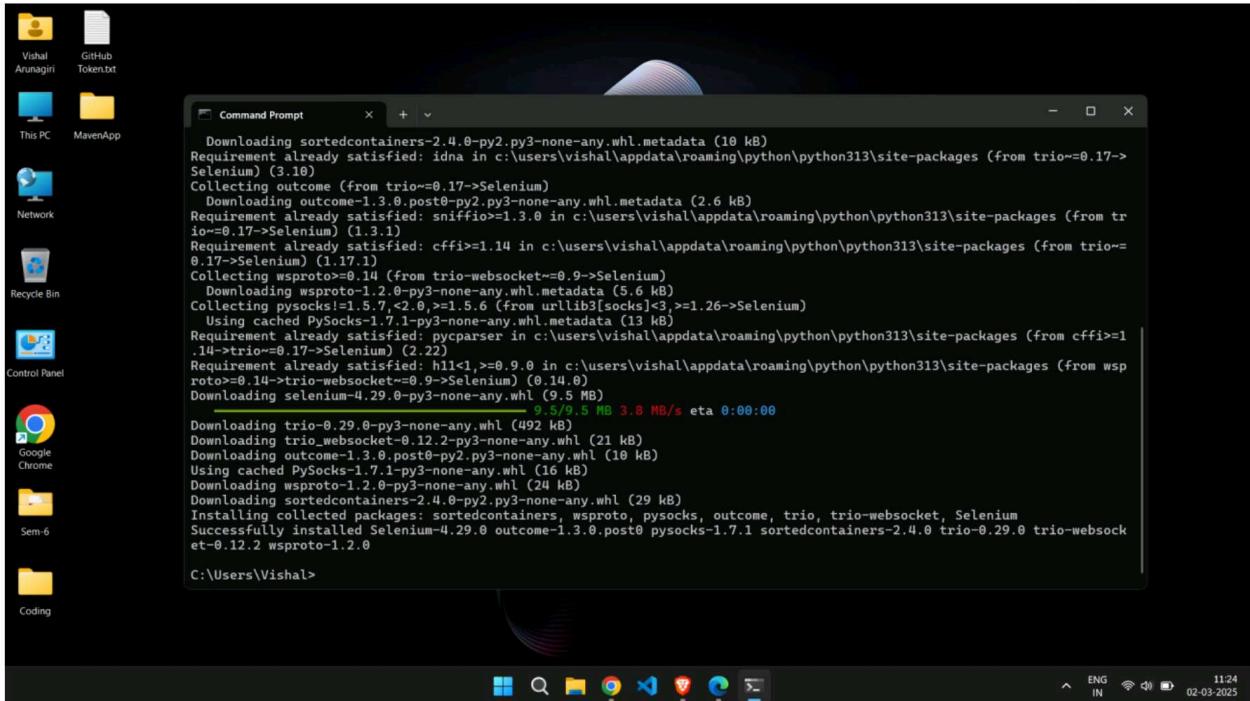






Installing Selenium libraries in Python





Download & Installation of Pycharm

Windows macOS Linux

PyCharm Professional

The Python IDE for data science and web development

[Download .exe \(Windows\)](#)

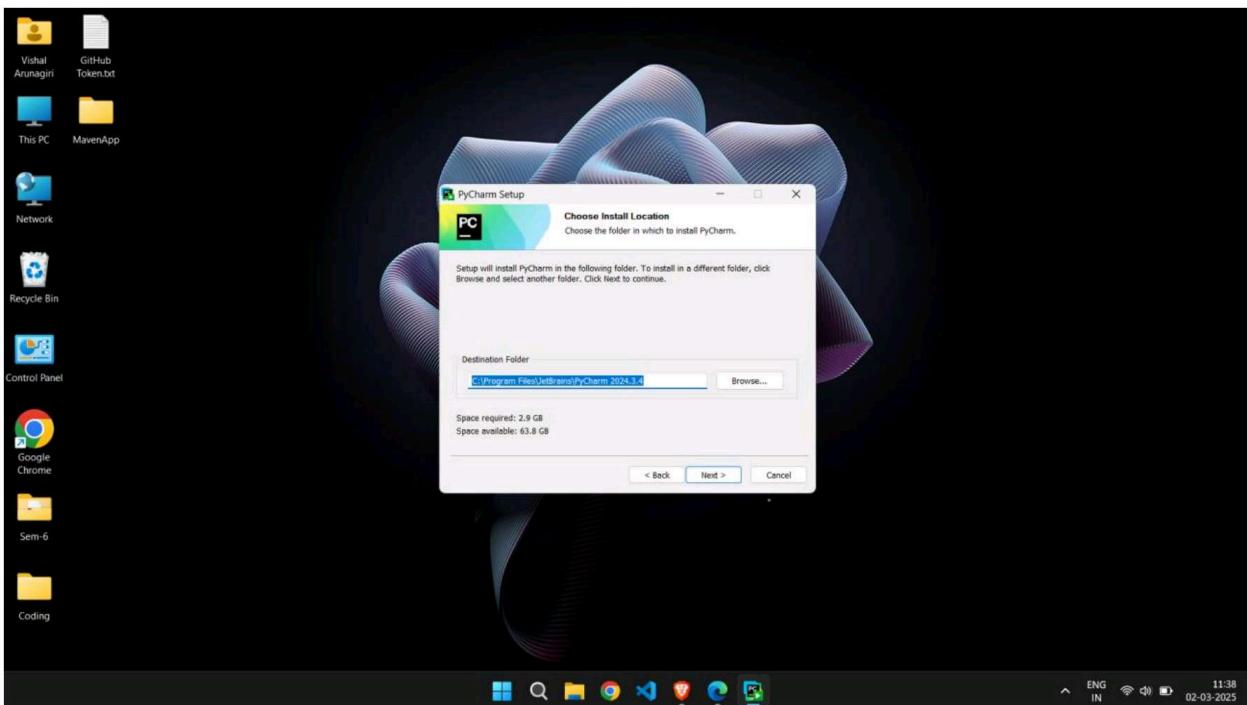
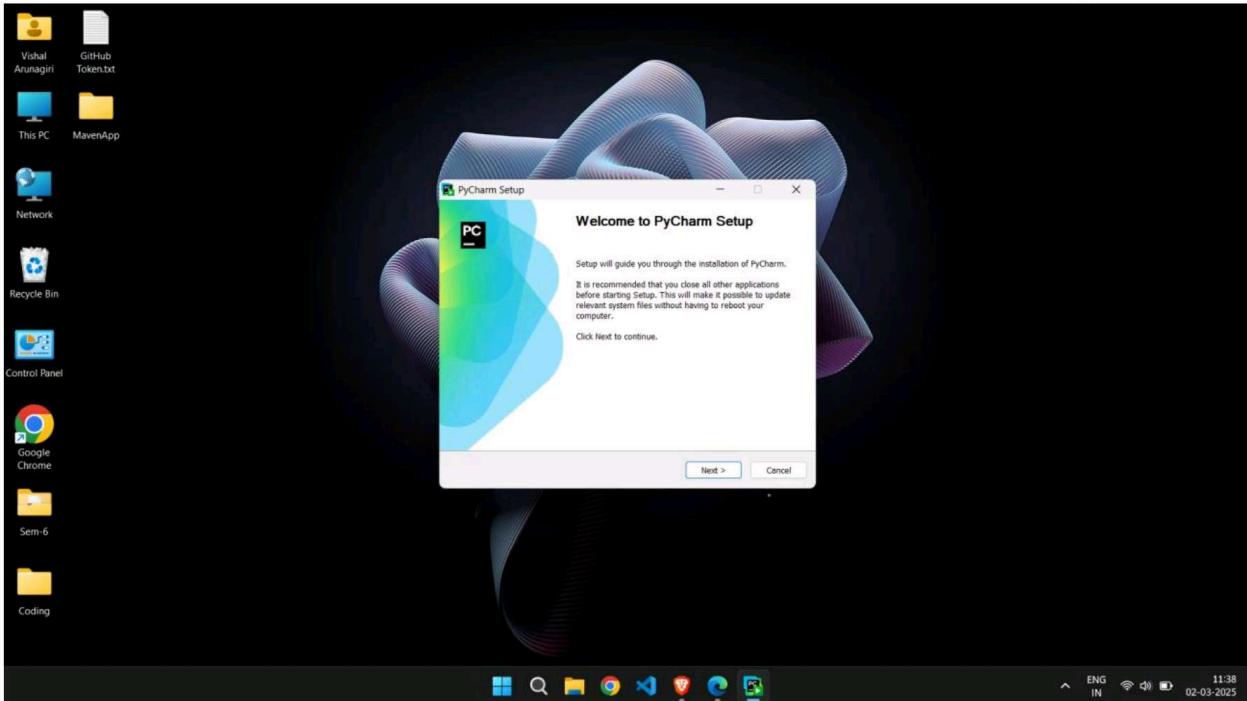
Free 30-day trial

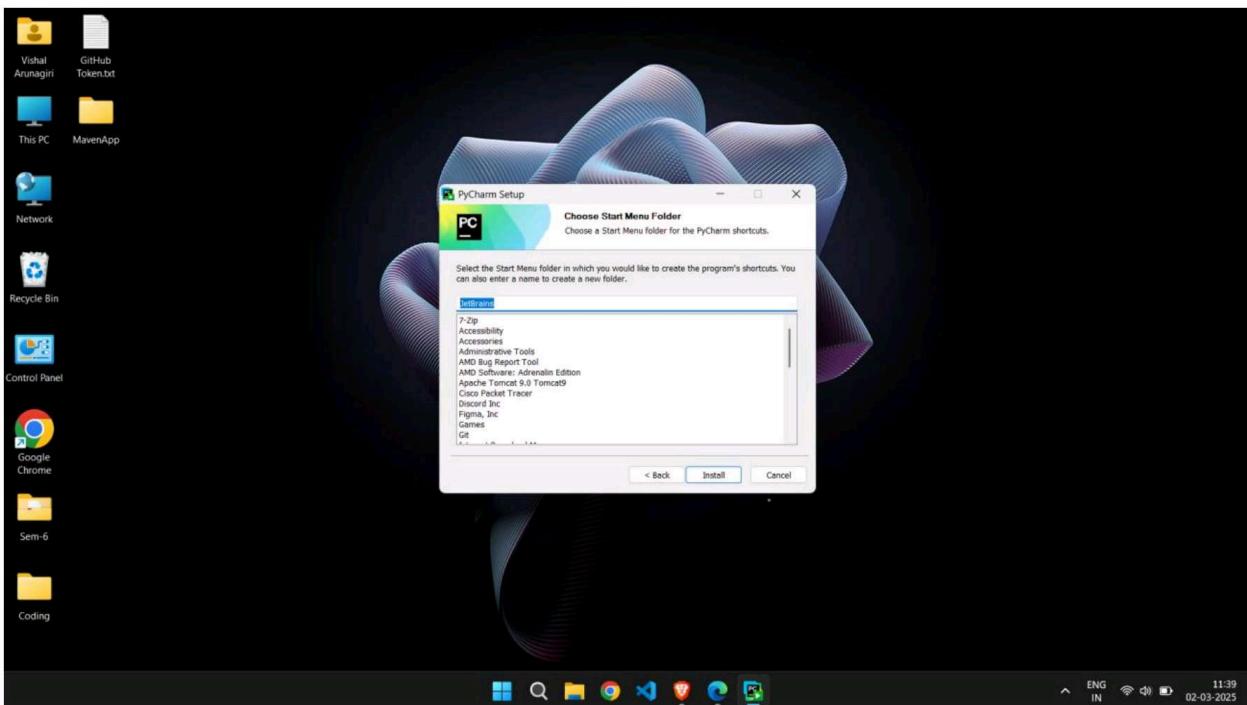
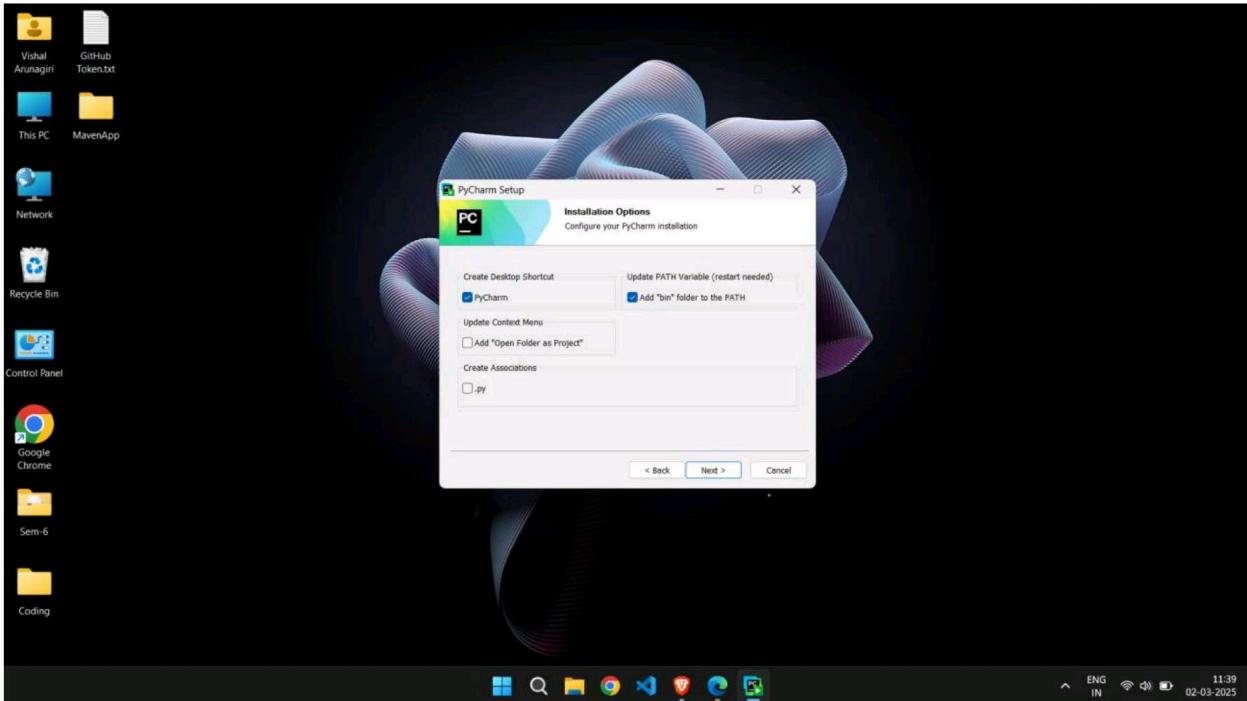
Version: 2024.3.4 System requirements Other versions

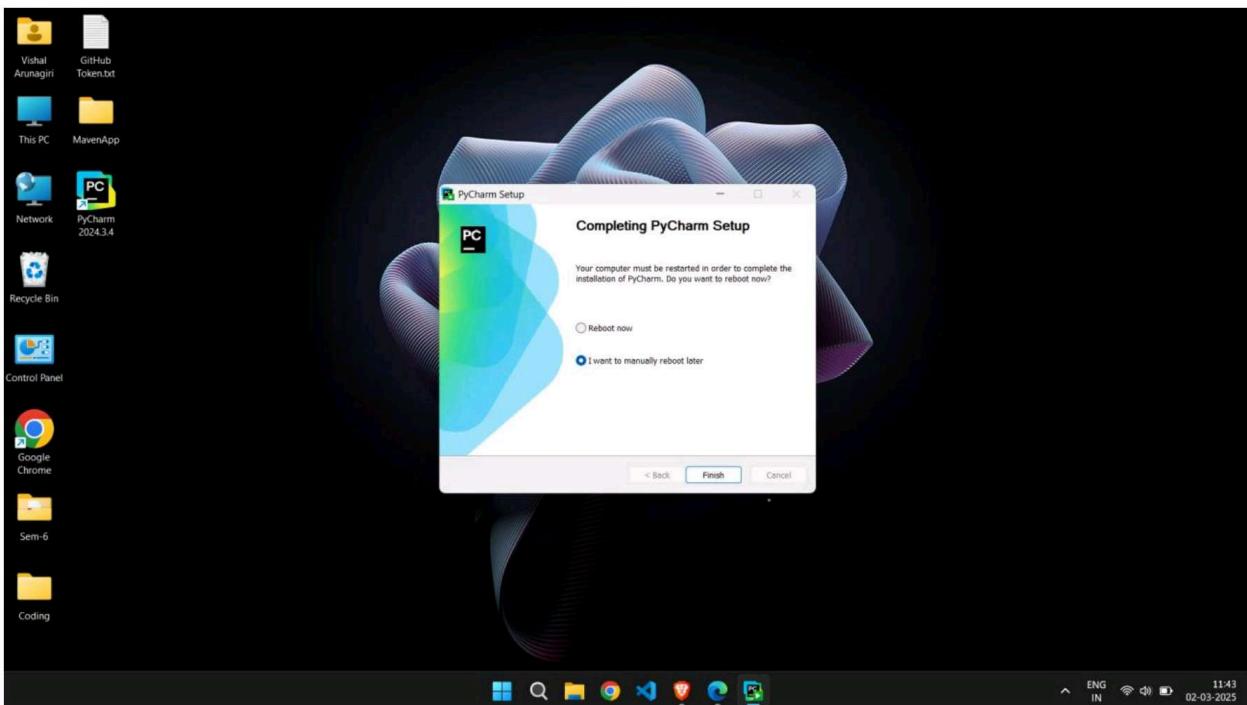
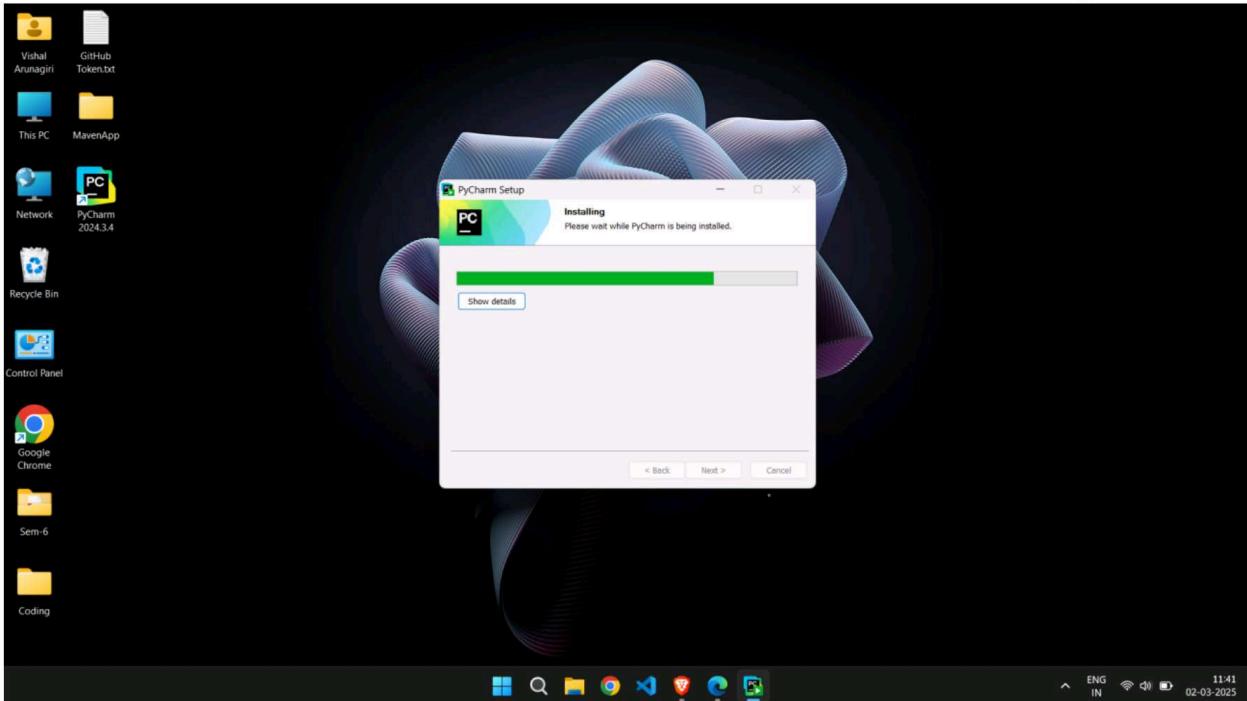
Build: 243.25659.43 Installation instructions Third-party software

27 February 2025

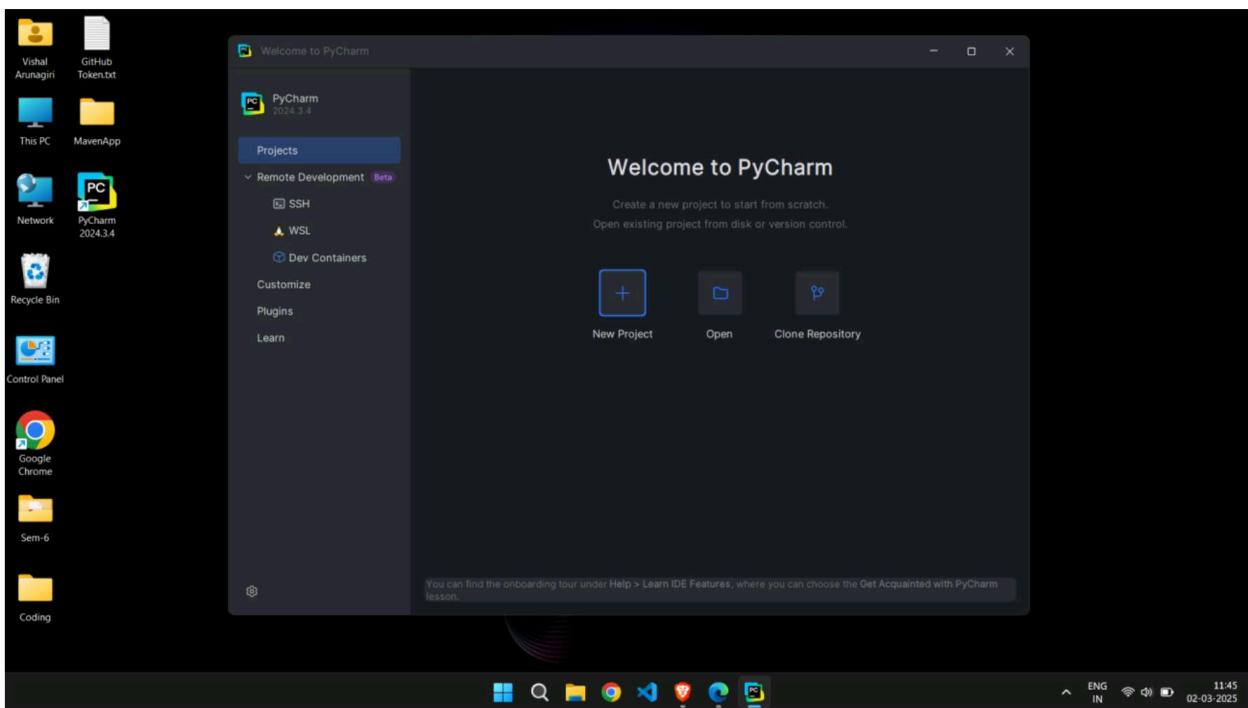
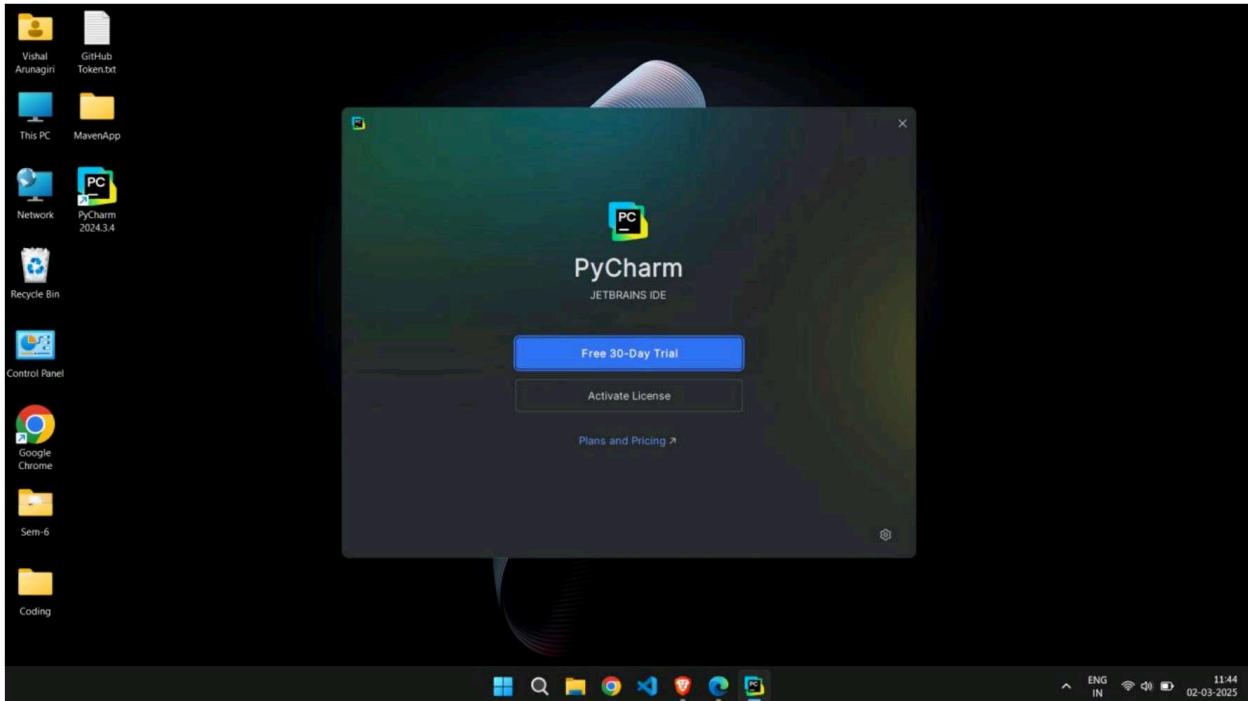


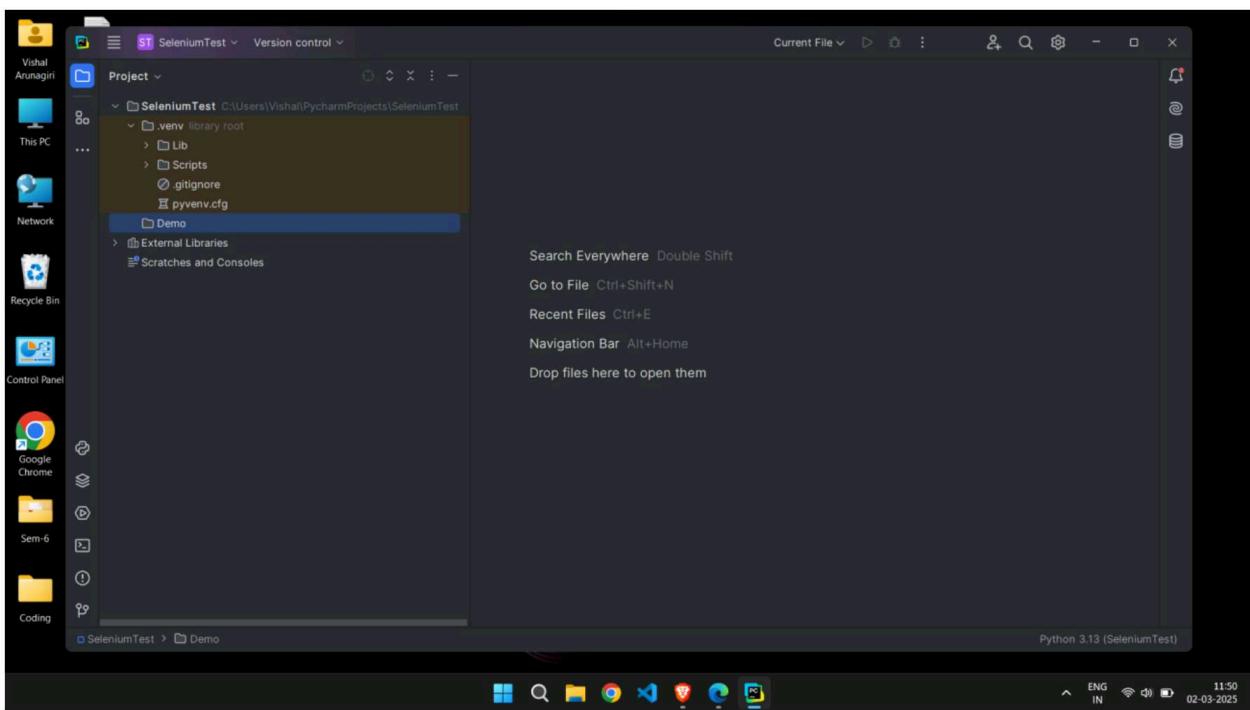
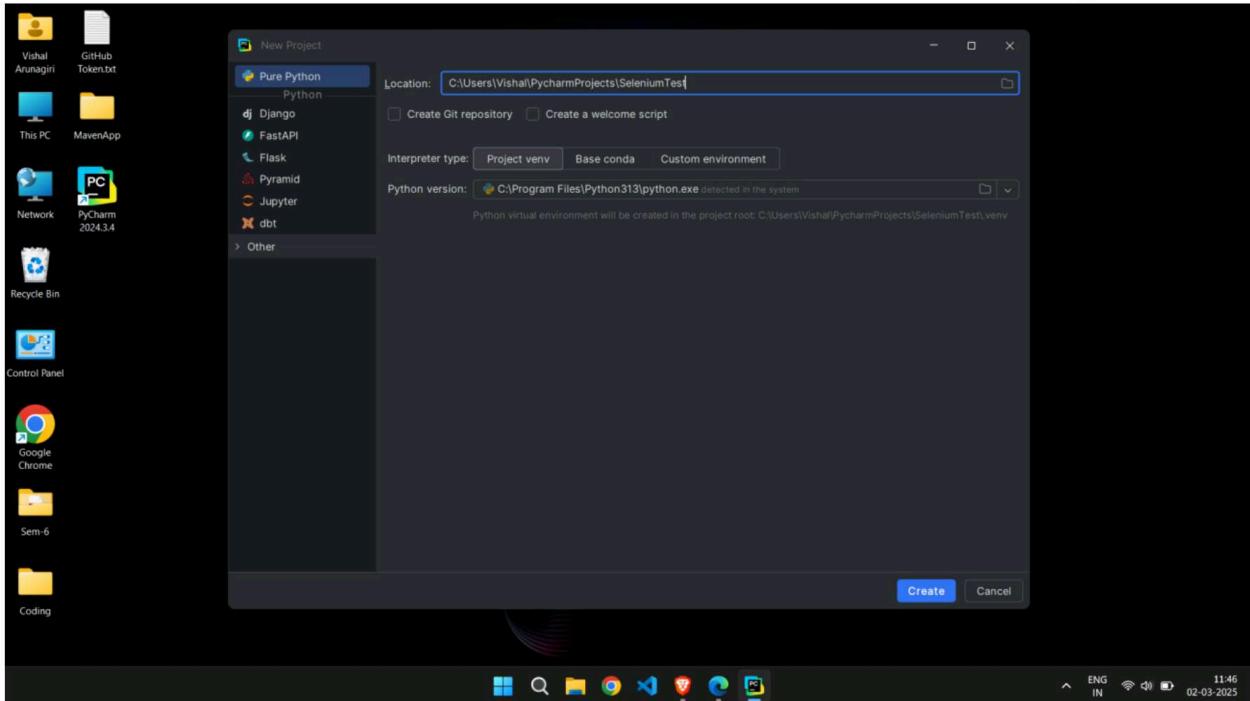


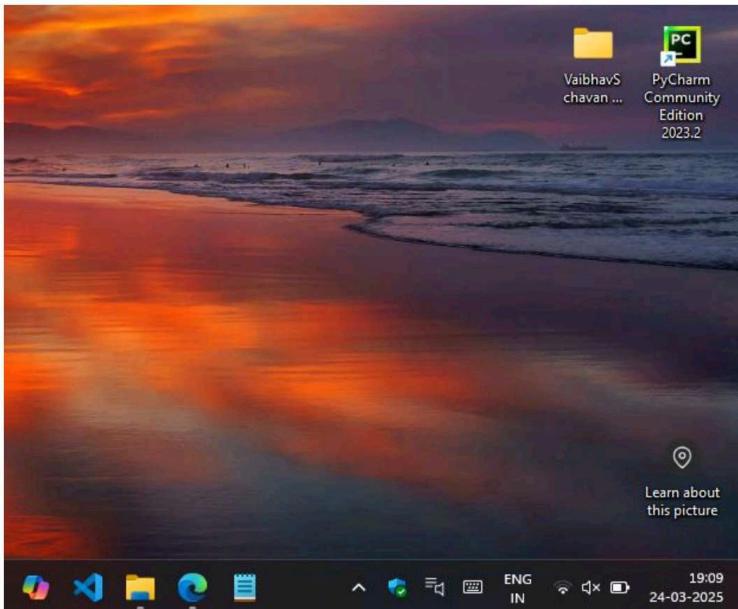




Create a New Project and write the Selenium Test Script



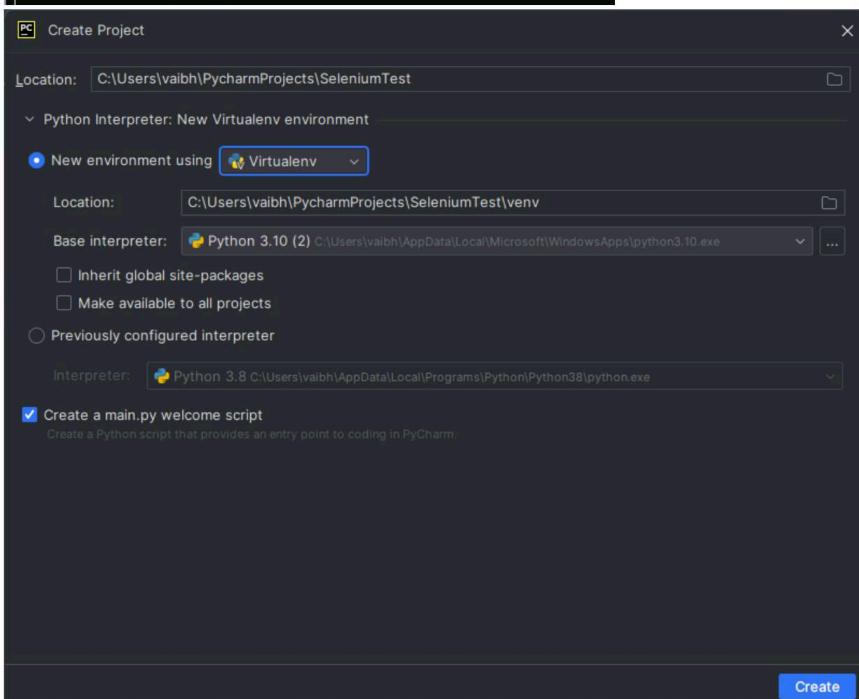




```
C:\Users\vaibh>python
Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

```
Command Prompt - python - + - ×
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vaibh>python -m pip install -U Selenium
Requirement already satisfied: Selenium in c:\users\vaibh\ana
conda3\lib\site-packages (4.30.0)
```



Project SeleniumTest Version control Current File

```
Sample1.py
1 from selenium import webdriver
2 import time
3 from selenium.webdriver.common.keys import Keys
4
5 print("Sample test case started")
6
7 driver = webdriver.Chrome()
8 driver.maximize_window()
9 driver.get("https://www.google.com/")
10
11 driver.find_element(by="name", value="q").send_keys("javatpoint")
12 time.sleep(3)
13
14 driver.find_element(by="name", value="btnK").send_keys(Keys.ENTER)
15 time.sleep(3)
16
```

Terminal Local +

```
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> pip install selenium
Collecting selenium
  Using cached selenium-4.30.0-py3-none-any.whl (9.4 MB)
Collecting websocket-client==1.8
  Using cached websocket_client-1.8.0-py3-none-any.whl (58 KB)
Collecting typing_extensions~=4.9
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 KB)
```

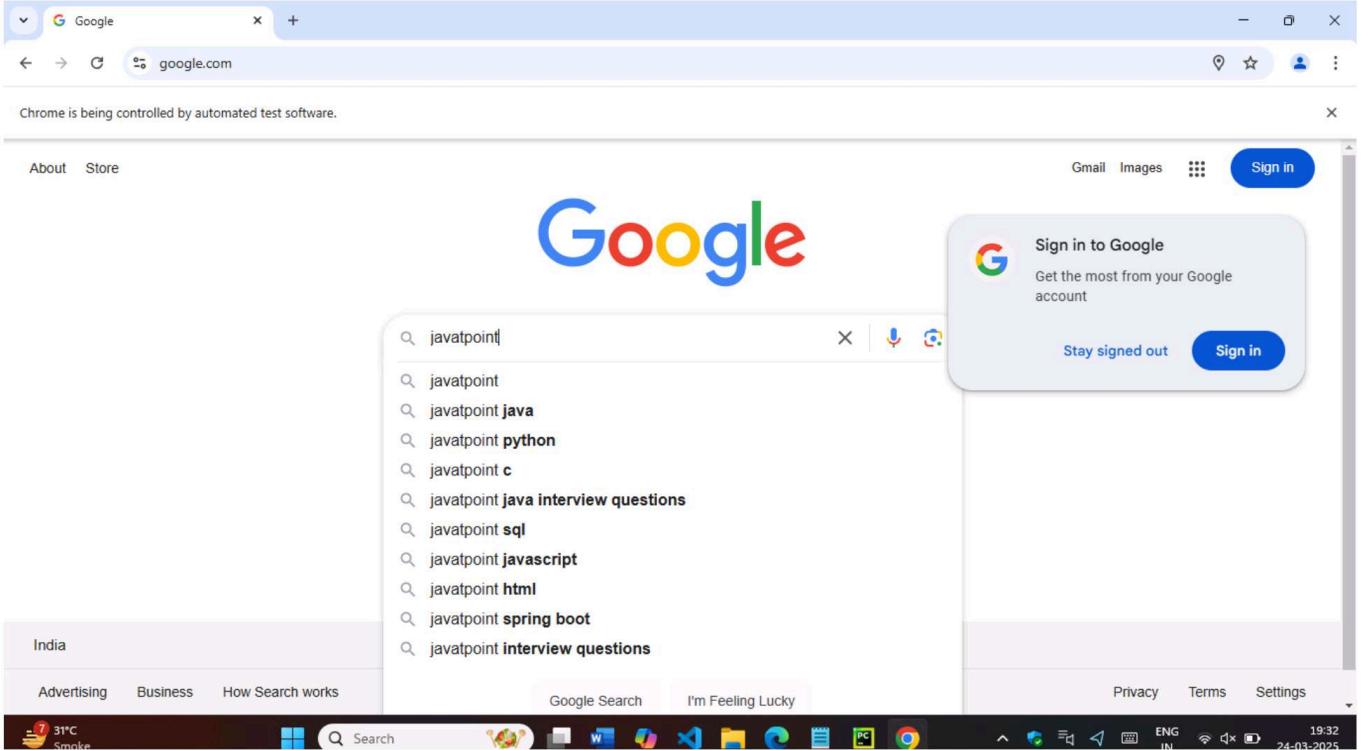
SeleniumTest > Demo1 > Sample1.py 7:28 CRLF UTF-8 4 spaces

Project SeleniumTest Version control Current File

```
Sample1.py
1 from selenium import webdriver
2 import time
3 from selenium.webdriver.common.keys import Keys
4
5 print("Sample test case started")
6
7 driver = webdriver.Chrome()
8 driver.maximize_window()
9 driver.get("https://www.google.com/")
10
11 driver.find_element(by="name", value="q").send_keys("javatpoint")
12 time.sleep(3)
13
14 driver.find_element(by="name", value="btnK").send_keys(Keys.ENTER)
15 time.sleep(3)
16
```

Run Sample1

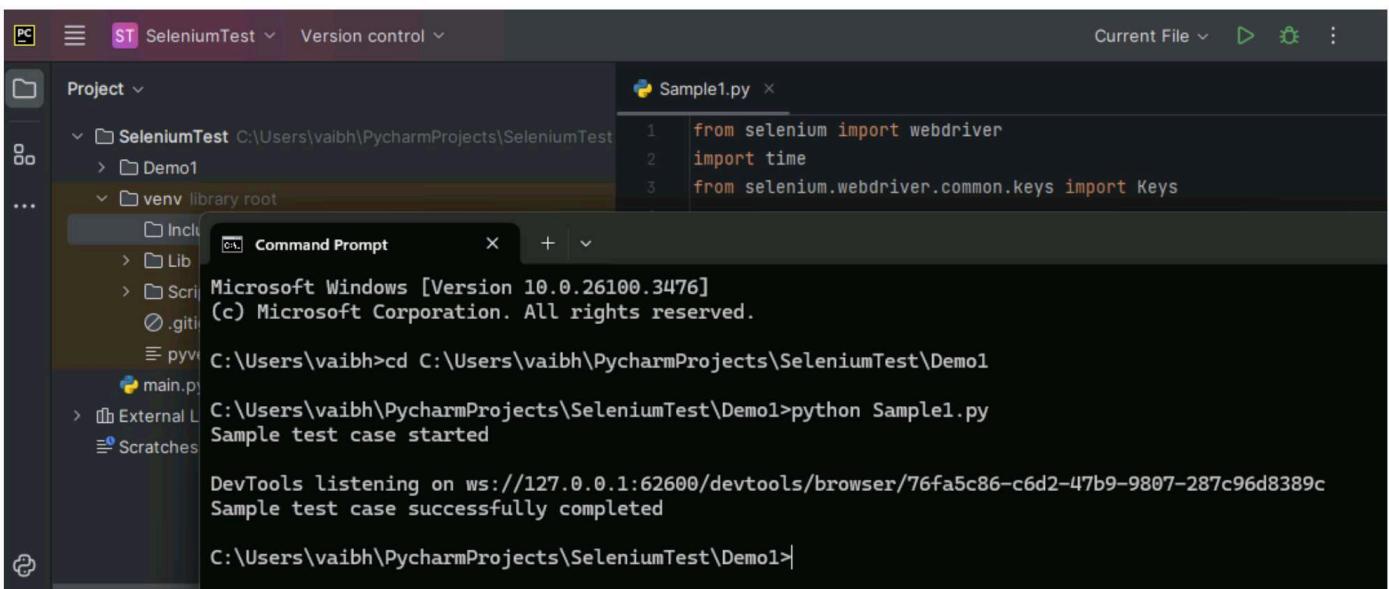
```
C:\Users\vaibh\PycharmProjects\SeleniumTest\venv\Scripts\python.exe C:\Users\vaibh\PycharmProjects\SeleniumTest\Demo1\Sample1.py
Sample test case started
Sample test case successfully completed
Process finished with exit code 0
```



Google search results for "javatpoint":

- javatpoint
- javatpoint java
- javatpoint python
- javatpoint c
- javatpoint java interview questions
- javatpoint sql
- javatpoint javascript
- javatpoint html
- javatpoint spring boot
- javatpoint interview questions

Sign in to Google button is visible on the right.



PyCharm IDE interface:

- Project: SeleniumTest
- File: Sample1.py
- Code content:

```
from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
```

- Terminal output:

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vaibh>cd C:\Users\vaibh\PycharmProjects\SeleniumTest\Demo1
C:\Users\vaibh\PycharmProjects\SeleniumTest\Demo1>python Sample1.py
Sample test case started

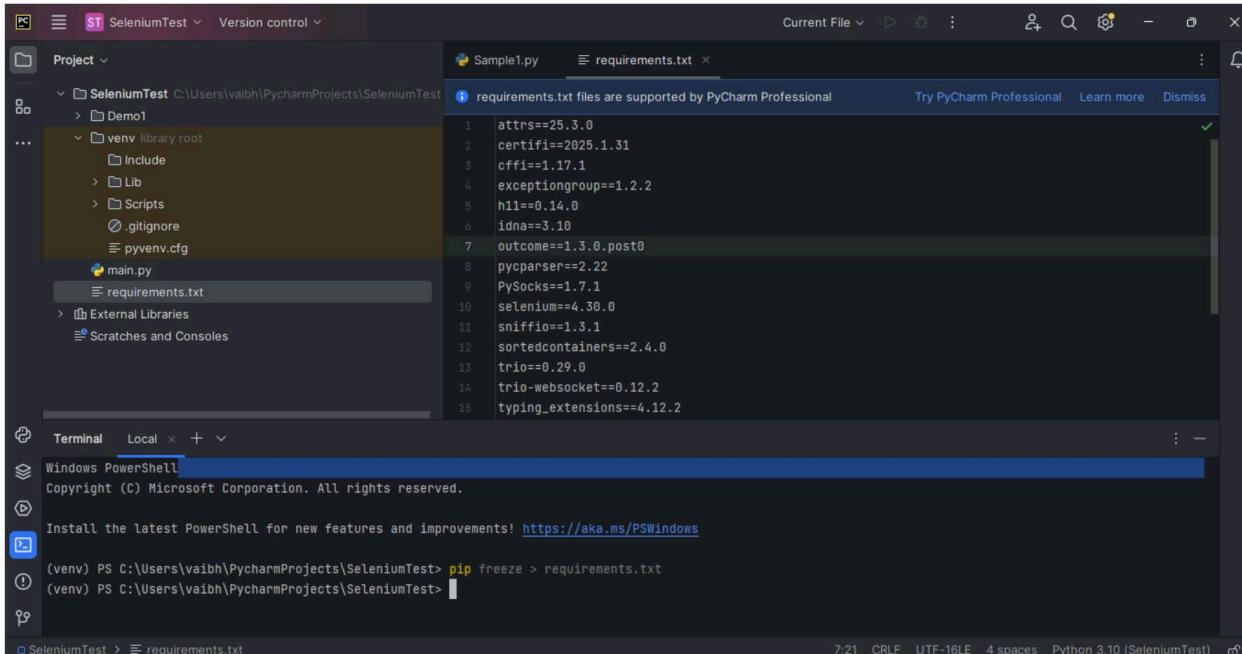
DevTools listening on ws://127.0.0.1:62600/devtools/browser/76fa5c86-c6d2-47b9-9807-287c96d8389c
Sample test case successfully completed

C:\Users\vaibh\PycharmProjects\SeleniumTest\Demo1>
```



The screenshot shows the Jenkins 'Installed plugins' page. A search bar at the top contains the text 'pipeline'. Two plugins are listed:

- Pipeline** (version 600.vb_57cdd26fd7) - A suite of plugins that lets you orchestrate automation, simple or complex. Status: Enabled.
- Pipeline Graph Analysis** (version 216vfd8b_ece330ca_) - Provides a REST API to access pipeline and pipeline run data. Status: Enabled.



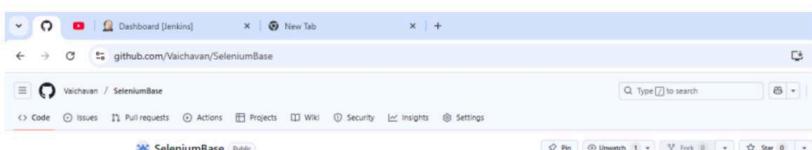
The screenshot shows a PyCharm Professional interface for a project named 'SeleniumTest'. The project structure includes a 'venv' directory with 'Demo1' and 'Scripts' sub-directories, and files like 'main.py', 'requirements.txt', and 'pyvenv.cfg'. The 'requirements.txt' file is open in the editor, showing dependencies:

```

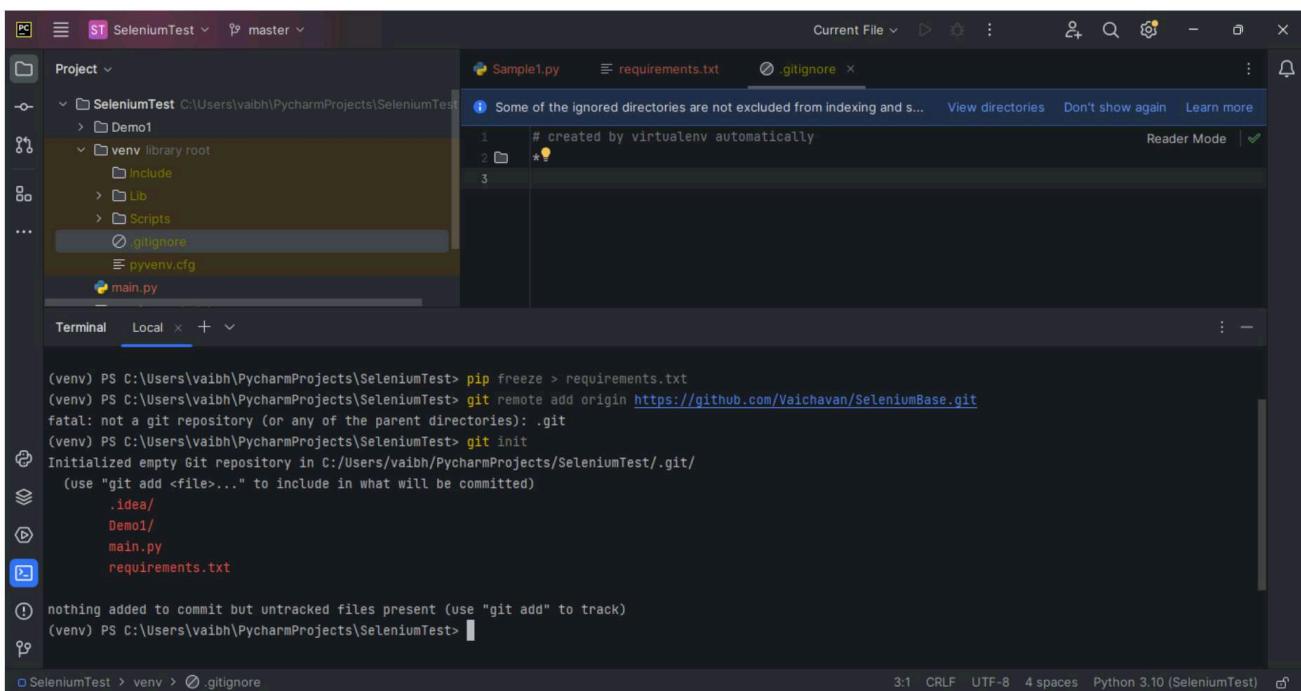
1 attrs==25.3.0
2 certifi==2025.1.31
3 cffi==1.17.1
4 exceptiongroup==1.2.2
5 h11==0.14.0
6 idna==3.10
7 outcome==1.3.0.post0
8 pycparser==2.22
9 PySocks==1.7.1
10 selenium==4.30.0
11 sniffio==1.3.1
12 sortedcontainers==2.4.0
13 trio==0.29.0
14 trio-websocket==0.12.2
15 typing_extensions==4.12.2

```

The terminal window shows the command `pip freeze > requirements.txt` being run in a Windows PowerShell environment.



The screenshot shows a GitHub repository page for 'SeleniumBase'. The repository is public and has 0 stars. It includes tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The code tab shows the repository's contents.



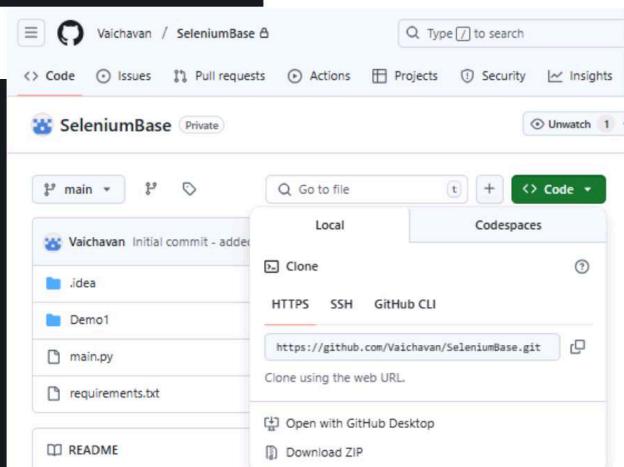
The screenshot shows a PyCharm Professional interface for the same 'SeleniumTest' project. The terminal window shows the command `git init` being run, initializing an empty Git repository. The message indicates that '.git' was created by virtualenv automatically. The terminal also shows the addition of files to the repository and a warning about untracked files.

```
(use "git add <file>..." to include in what will be committed)
.idea/
Demo1/
main.py
requirements.txt

nothing added to commit but untracked files present (use "git add" to track)
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git add .
warning: in the working copy of '.idea/inspectionProfiles/profiles_settings.xml', LF will be replaced by CRLF
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git status
On branch master
```

```
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git commit -m "Initial commit - added Selenium test files"
[master (root-commit) 895fdd9] Initial commit - added Selenium test files
9 files changed, 72 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/SeleniumTest.iml
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 Demo1/Sample1.py
create mode 100644 main.py
create mode 100644 requirements.txt
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest>
```

```
create mode 100644 Demo1/Sample1.py
create mode 100644 main.py
create mode 100644 requirements.txt
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git branch -M main
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git push -u origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 2.37 KiB | 607.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vaichavan/SeleniumBase.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
(venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest>
```



The screenshot shows the Jenkins configuration for the 'SeleniumBase' job. Under the 'Source Code Management' section, 'Git' is selected. The 'Repository URL' is set to <https://github.com/Vaichavan/SeleniumBase.git>. The 'Credentials' dropdown is set to 'webs123'. There are buttons for '+ Add' and 'Advanced'. At the bottom, there are 'Save' and 'Apply' buttons.

localhost:8080/manage/pluginManager/available

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Search: shini

Install	Name	Released
	ShiningPanda 0.24	6 yr 8 mo ago
	Build Tools	
This plugin adds Python support to Jenkins with some useful builders (standard Python builder, virtualenv builder, ...) and the ability to use a Python axis in multi-configuration projects (for testing on multiple versions of Python).		

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vaibh> where python
C:\Users\vaibh\anaconda3\python.exe
C:\Users\vaibh\AppData\Local\Programs\Python\Python38\python.exe
C:\Users\vaibh\AppData\Local\Microsoft\WindowsApps\python.exe

C:\Users\vaibh>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\vaibh\anaconda3\lib\site-packages (22.3.1)
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 1.9 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
Successfully installed pip-25.0.1

C:\Users\vaibh>pip install seleniumbase
Collecting seleniumbase
  Downloading seleniumbase-4.36.1-py3-none-any.whl.metadata (86 kB)
Requirement already satisfied: pip>=25.0.1 in c:\users\vaibh\anaconda3\lib\site-packages (from seleniumbase)

C:\Users\vaibh>sbase --version
SeleniumBase 4.36.1
```

```
C:\Users\vaibh>pytest --version
pytest 8.3.5
```

Dashboard > expt > SeleniumBase > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Home ? C:\Users\vaibh\anaconda3\python.exe

Nature ? Shell

Command ?

```
python -m venv env
call ./env/Scripts/activate.bat
pip install -r requirements.txt
sbase install chromedriver latest
python Demo1/Sample1.py
```

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾

Save

Apply

SeleniumBase #6 Console [Jen...]

localhost:8080/view/expt/job/SeleniumBase/6/console

Jenkins

Dashboard > expt > SeleniumBase > #6 > Console Output

Console Output

Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\workspace\SeleniumBase
The recommended git tool is: NONE
using credential webs123
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\SeleniumBase.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/Vaibhav/SeleniumBase.git # timeout=10
Fetching upstream changes from https://github.com/Vaibhav/SeleniumBase.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.0.windows.1'

Download Copy View as plain

[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

*** chromedriver to download = 134.0.6998.165 (Latest Stable)

Downloading chromedriver-win64.zip from:
<https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165>.
Download Complete!

Extracting ['chromedriver.exe'] from chromedriver-win64.zip ...
Unzip Complete!

The file ['chromedriver.exe'] was saved to:
C:\Users\vaibh\anaconda3\lib\site-packages\seleniumbase\drivers\chromedriver.exe

Making [chromedriver.exe 134.0.6998.165] executable ...
[chromedriver.exe 134.0.6998.165] is now ready for use!

Sample test case started
Sample test case successfully completed
Finished: SUCCESS

```

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

print("Sample test case started")

# Configure Chrome options
chrome_options = Options()
chrome_options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36")
chrome_options.add_argument("--disable-blink-features=AutomationControlled") # Prevent detection

# Initialize WebDriver
driver = webdriver.Chrome(options=chrome_options)
driver.maximize_window()
driver.get("https://www.google.com/")

# Search for 'simplilearn'
search_box = driver.find_element(By.NAME, "q")
search_box.send_keys("simplilearn")
time.sleep(2)
search_box.send_keys(Keys.ENTER)
time.sleep(3)

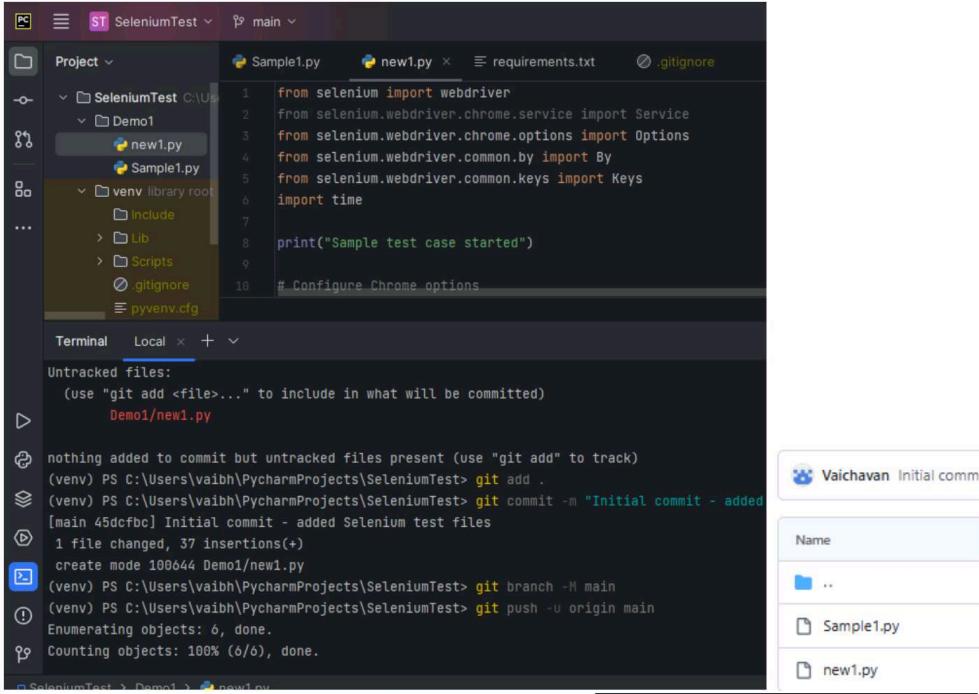
# Get search result snippets
results = driver.find_elements(By.CSS_SELECTOR, ".tF2Cxc .VwiC3b")

# Print the first few snippets
for i, result in enumerate(results[:5]): # Adjust the number as needed
    print(f"Result {i+1}: {result.text}\n")

# Close the browser
driver.quit()

print("Sample test case successfully completed")

```



```

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

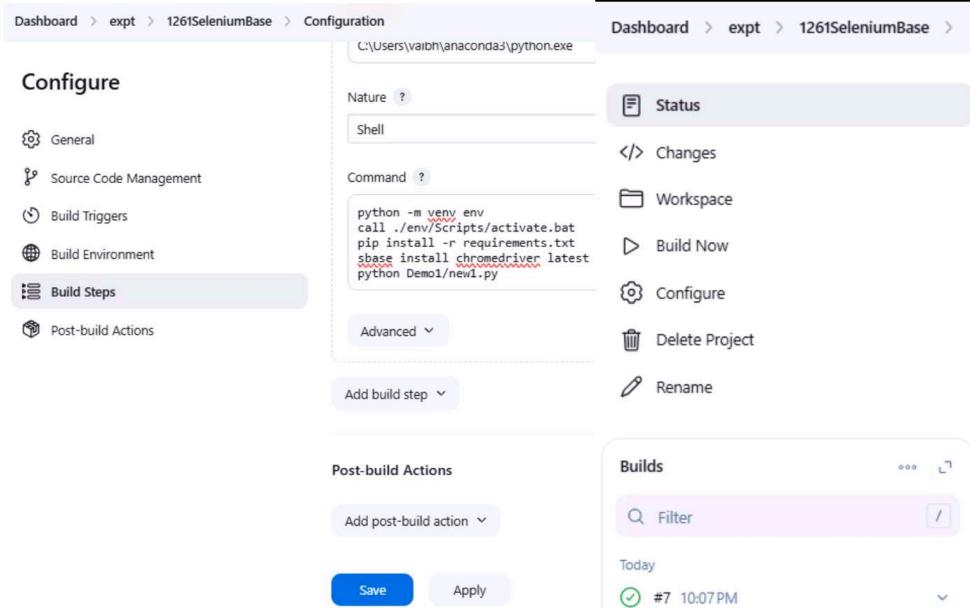
print("Sample test case started")

# Configure Chrome options

```

Untracked files:
 (use "git add <file>..." to include in what will be committed)
 Demo1/new1.py

nothing added to commit but untracked files present (use "git add" to track)
 (venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git add .
 (venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git commit -m "Initial commit - added [main 45dcfbc] Initial commit - added Selenium test files"
 ⏴ 1 file changed, 37 insertions(+)
 create mode 100644 Demo1/new1.py
 (venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git branch -M main
 (venv) PS C:\Users\vaibh\PycharmProjects\SeleniumTest> git push -u origin main
 Enumerating objects: 6, done.
 Counting objects: 100% (6/6), done.



Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Nature ? Shell

Command ?

```

python -m venv env
call ./env/Scripts/activate.bat
pip install -r requirements.txt
base install chromedriver latest
python Demo1/new1.py

```

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Rename

Add build step ▾

Post-build Actions

Add post-build action ▾

Save Apply

Builds

Filter

Today

#7 10:07PM

Dashboard > expt > 1261SeleniumBase > #7 > Console Output

C:\Users\vaibh\anaconda3\Lib\site-packages\seleniumbase\drivers\
chromedriver.exe

Making [chromedriver.exe 134.0.6998.165] executable ...
[chromedriver.exe 134.0.6998.165] is now ready for use!

Sample test case started

Result 1: Get Certified. Get Ahead. 8 Million+ Careers advanced; 1,500 Live classes every month ♦ Be a Leader in Your FieldChange, Adapt and Build with AI. All our top ...

Result 2: Simplilearn is the world's leading online training provider and has helped over 8 million professionals, and corporations acquire the skills they need to ...

Result 3: World's leading digital skills platform providing digital skills training to help individuals acquire the skills they need to thrive in the ...

Result 4: Our online learning platform offers an unparalleled selection of over 400 certificate courses, taught by a talented team of 2000+ expert trainers.

Result 5: Simplilearn is a global leader in digital upskilling, offering highly specialized training in emerging technologies and processes shaping the digital economy's ...

Sample test case successfully completed

Finished: SUCCESS

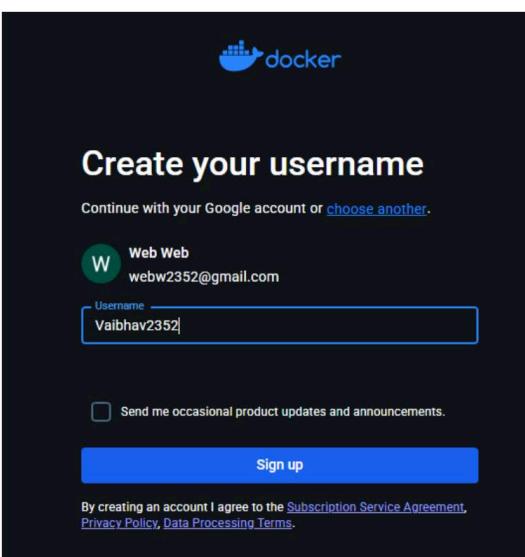
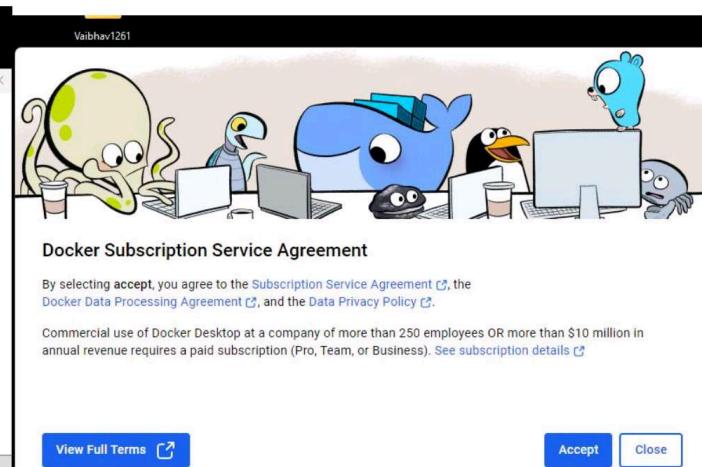
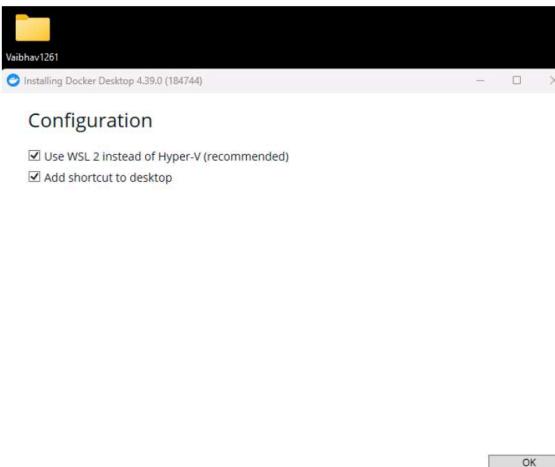
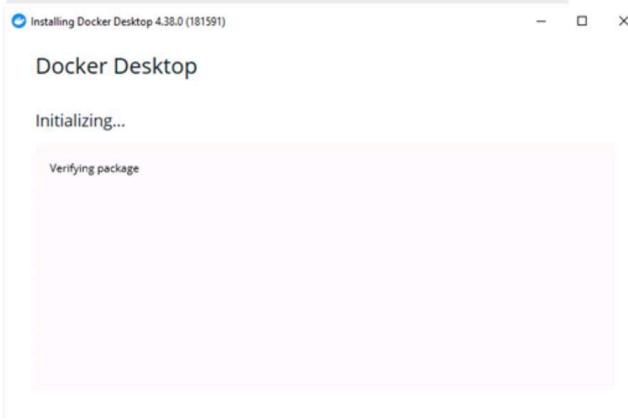
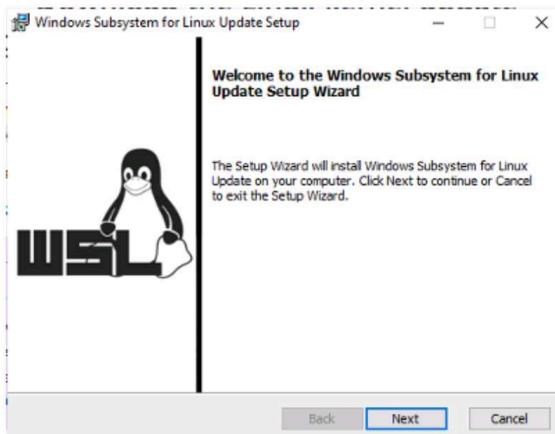
7o- ✓ Experiment 7: Docker Architecture and Basic Commands

Aim: To understand Docker architecture, install Docker, and execute Docker commands to manage and interact with containers.

Steps:

1. Download and install **Docker Desktop** from [Docker Hub](#).
2. Create a **Docker Hub** account.
3. Verify Docker installation:
`docker --version`
5. Login to Docker:
`docker login`
7. Pull a public image from Docker Hub:
`docker pull ubuntu`
9. Run a container interactively:
`docker run -it ubuntu /bin/bash`
11. List running and stopped containers:
`docker ps -a`
13. Stop and remove containers:
`docker stop <container_id>`
`docker rm <container_id>`
16. List all images:
`docker images`
18. Remove images:
`docker rmi <image_id>`

Conclusion: In this experiment, we explored Docker's architecture, including its image, container, and registry components. We learned to install Docker, manage containers, and interact with Docker Hub, laying the foundation for container-based application deployment in DevOps.



```
PS C:\Users\Admin> docker --version
Docker version 28.0.1, build 668a01e
PS C:\Users\Admin> docker pull node
Using default tag: latest
latest: Pulling from library/node
b6ccb5f2452c: Pull complete
255774ee0027: Pull complete
2d0f5d4eac65: Pull complete
353e14e5cc47: Pull complete
7cd785773db4: Pull complete
baf255479525: Pull complete
891eb8249475: Pull complete
fe9c928bbad8: Pull complete
Digest: sha256:990d0ab35ae15d8a322ee1eeaf4f7cf14e367d3d0ee2f472704b7b3df4c9e7c1
Status: Downloaded newer image for node:latest
docker.io/library/node:latest
PS C:\Users\Admin> docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Common Commands:
  run      Create and run a new container from an image
  exec    Execute a command in a running container
  ps      List containers
```

```
PS C:\Users\Admin\vai> docker search ubuntu
Run 'docker buildx build --help' for more information
PS C:\Users\Admin\vai> docker search ubuntu
NAME          DESCRIPTION                           STARS      OFFICIAL
ubuntu        Ubuntu is a Debian-based Linux operating sys... 17524      [OK]
ubuntu/squid   Squid is a caching proxy for the Web. Long-t... 108
ubuntu/nginx   Nginx, a high-performance reverse proxy & we... 128
ubuntu/cortex   Cortex provides storage for Prometheus. Long-... 4
ubuntu/kafka    Apache Kafka, a distributed event streaming ... 53
ubuntu/apache2  Apache, a secure & extensible open-source HT... 90
ubuntu/bind9    BIND 9 is a very flexible, full-featured DNS... 102
ubuntu/prometheus Prometheus is a systems and service monitori... 71
ubuntu/zookeeper ZooKeeper maintains configuration informatio... 13
ubuntu/mysql    MySQL open source fast, stable, multi-thread... 67
ubuntu/postgres PostgreSQL is an open source object-relatio... 41
ubuntu/redis    Redis, an open source key-value store. Long-... 23
ubuntu/jre     Distroless Java runtime based on Ubuntu. Lon... 19
ubuntu/dotnet-aspnet Chiseled Ubuntu runtime image for ASP.NET a... 26
ubuntu/grafana  Grafana, a feature rich metrics dashboard & ... 12
ubuntu/cassandra Cassandra, an open source NoSQL distributed ... 2
ubuntu/nemcached Memcached, in-memory keyvalue store for smal... 5
ubuntu/python   A chiseled Ubuntu rock with the Python runt... 21
ubuntu/dotnet-runtime Chiseled Ubuntu runtime image for .NET apps... 20
ubuntu/dotnet-deps Chiseled Ubuntu for self-contained .NET & A... 16
ubuntu/prometheus-alertmanager Alertmanager handles client alerts from Prom... 9
ubuntu/mlflow    MLFlow: for managing the machine learning li... 5
ubuntu/telegraf  Telegraf collects, processes, aggregates & w... 4
ubuntu/loki     Grafana Loki, a log aggregation system like ... 2
ubuntu/chiseled-jre [MOVED TO ubuntu/jre] Chiseled JRE: distroL... 3
PS C:\Users\Admin\vai> docker search ubuntu --filter stars=1000
NAME          DESCRIPTION                           STARS      OFFICIAL
```

```
PS C:\Users\Admin\vai> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
node            latest   990d0ab35ae1  2 weeks ago   1.61GB
ubuntu          latest   72297848456d  2 months ago  117MB
mysql           latest   0596fa224cdf   2 months ago  1.09GB
PS C:\Users\Admin\vai> |
```

```
PS C:\Users\Admin\vai> docker pull ubuntu
ubuntu        Ubuntu is a Debian-based Linux operating sys... 17524      [OK]
PS C:\Users\Admin\vai> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\Admin\vai> docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
79f239a40e62: Pull complete
daac2c594bdd: Pull complete
769c3ac51f88: Pull complete
ceal172a6e83b: Pull complete
548990e33276: Pull complete
fae51f7de1fb: Pull complete
c11056354384: Pull complete
cb8acbf2440c: Pull complete
49978e7ccddf: Pull complete
b2ead3e96e6b: Pull complete
Digest: sha256:0596fa224cdf3b3355ce3ddbf7ce77be27ec9e51841dfc5d2e1c8b81eea69d2
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
PS C:\Users\Admin\vai> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
node            latest   990d0ab35ae1  2 weeks ago   1.61GB
ubuntu          latest   72297848456d  2 months ago  117MB
mysql           latest   0596fa224cdf   2 months ago  1.09GB
PS C:\Users\Admin\vai> |
```

```
PS C:\Users\Admin\vai> docker login
Authenticating with existing credentials... [Username: vaibhav2352]
```

```
[ Info → To login with a different account, run 'docker logout' followed by 'docker login'
```

```
Login_Succeeded
PS C:\Users\Admin\vai> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:7e1a4e2d11e2ac7a8c3f768d4166c2defeb09d2a750b010412b6ea13de1efb19
Status: Downloaded newer image for hello-world:latest

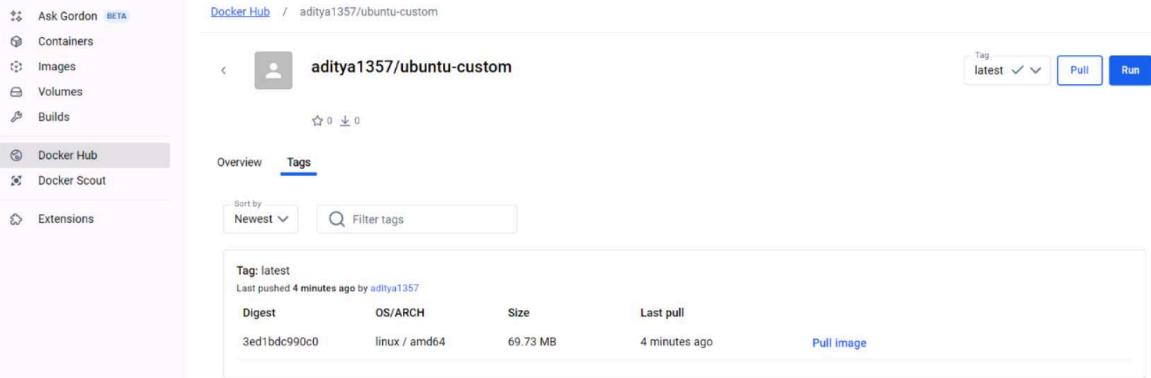
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

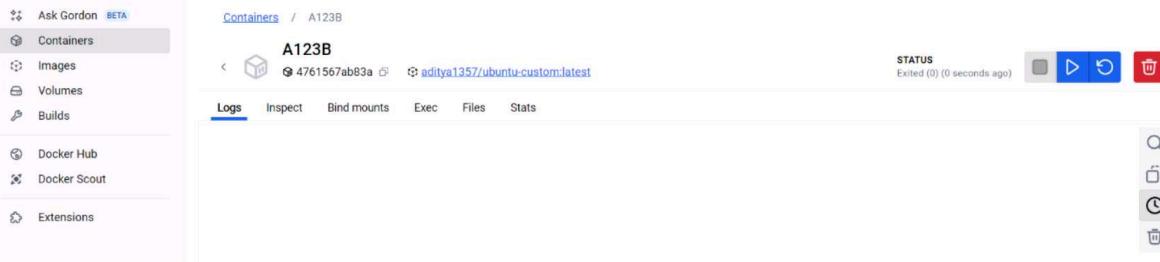
For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

The screenshot shows the Docker Hub interface for the repository `aditya1357/ubuntu-custom`. The `Tags` tab is selected, showing one tag: `latest`. The tag was last pushed 4 minutes ago by `aditya1357`. The table provides details about the image:

Digest	OS/ARCH	Size	Last pull
<code>3ed1bdc990c0</code>	linux / amd64	69.73 MB	4 minutes ago

A `Pull image` button is visible.



The screenshot shows Docker Desktop with a running container named `A123B`. The container ID is `4761567ab83a`. The status bar indicates it exited 0 seconds ago. The `Logs` tab is selected, showing the container's logs. Other tabs include `Inspect`, `Bind mounts`, `Exec`, `Files`, and `Stats`.

```
PS C:\Users\adity> docker pull aditya1357/ubuntu-custom
Using default tag: latest
latest: Pulling from aditya1357/ubuntu-custom
Digest: sha256:3ed1bdc990c0a7ff7321616d4502c0da631d6453e0658af16b76c6adb7774541
Status: Image is up to date for aditya1357/ubuntu-custom:latest
docker.io/aditya1357/ubuntu-custom:latest
PS C:\Users\adity> docker run -it ubuntu-custom:latest
Unable to find image 'ubuntu-custom:latest' locally
docker: Error response from daemon: pull access denied for ubuntu-custom, repository does not exist or may require 'docker login': denied: requested access to the resource is denied

Run 'docker run --help' for more information
PS C:\Users\adity> docker pull aditya1357/ubuntu-custom
Using default tag: latest
latest: Pulling from aditya1357/ubuntu-custom
Digest: sha256:3ed1bdc990c0a7ff7321616d4502c0da631d6453e0658af16b76c6adb7774541
Status: Image is up to date for aditya1357/ubuntu-custom:latest
docker.io/aditya1357/ubuntu-custom:latest
PS C:\Users\adity> docker run -it ubuntu-custom bash
Unable to find image 'ubuntu-custom:latest' locally
docker: Error response from daemon: pull access denied for ubuntu-custom, repository does not exist or may require 'docker login': denied: requested access to the resource is denied

Run 'docker run --help' for more information
PS C:\Users\adity> docker run -it aditya1357/ubuntu-custom bash
root@159f8d04cac2:/# |
```

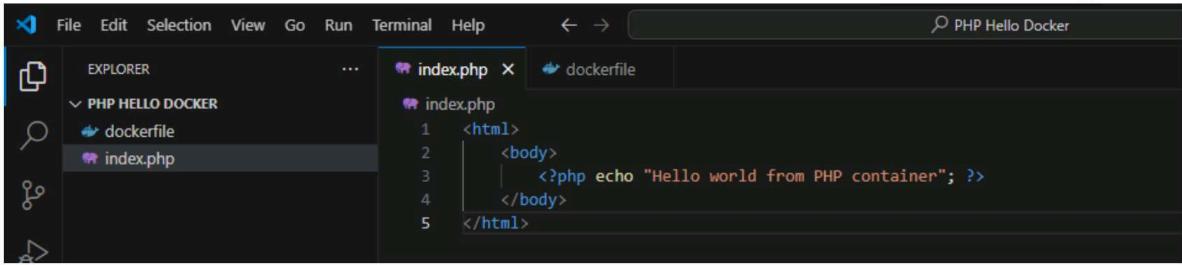
8o- Experiment 8: Building a Docker Image

Aim: To learn Dockerfile instructions and build an image for a sample web application on Docker Engine.

Steps:

1. Create a simple static web app (index.html) inside a project folder.
2. Write a Dockerfile in the same folder:
3. FROM nginx
4. COPY ./usr/share/nginx/html
5. EXPOSE 80
6. Build the Docker image:
7. docker build -t my-web-app .
8. Verify the image using:
9. docker images
10. Run a container from the built image:
11. docker run -d -p 8080:80 my-web-app
12. Open a browser and visit <http://localhost:8080> to see the app running.
13. Push the image to Docker Hub:
14. docker tag my-web-app yourusername/my-web-app
15. docker push yourusername/my-web-app

Conclusion: This experiment taught us how to build Docker images using Dockerfiles. We containerized a static web app, ran it on Docker Engine, and pushed the image to Docker Hub, reinforcing key DevOps practices like reproducibility, versioning, and portability.



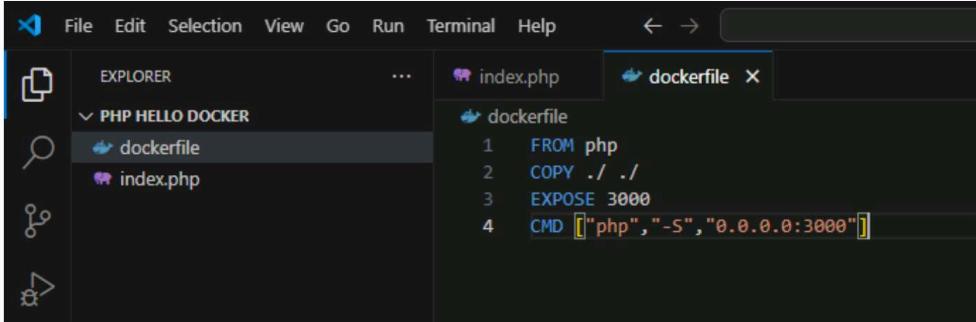
File Edit Selection View Go Run Terminal Help

EXPLORER PHP HELLO DOCKER index.php dockerfile

```

index.php x
index.php
1 <html>
2   <body>
3     <?php echo "Hello world from PHP container"; ?>
4   </body>
5 </html>

```



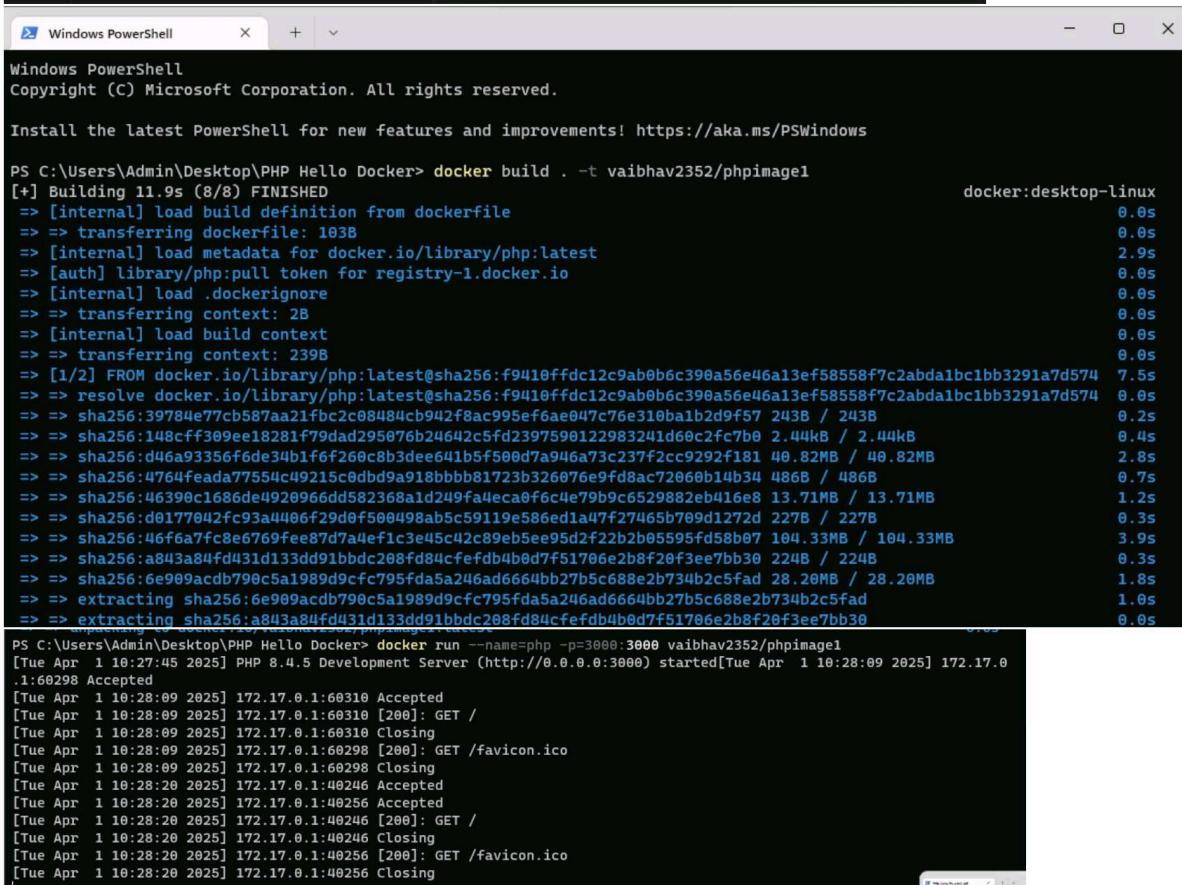
File Edit Selection View Go Run Terminal Help

EXPLORER PHP HELLO DOCKER dockerfile index.php

```

dockerfile
1 FROM php
2 COPY ./ .
3 EXPOSE 3000
4 CMD ["php", "-S", "0.0.0.0:3000"]

```



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

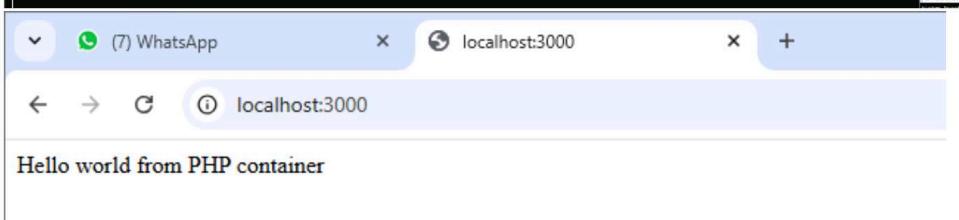
PS C:\Users\Admin\Desktop\PHP Hello Docker> docker build . -t vaibhav2352/phpimage1
[+] Building 11.9s (8/8) FINISHED                                            docker:desktop-linux
--> [internal] load build definition from dockerfile                      0.0s
--> => transferring dockerfile: 103B                                         0.0s
--> [internal] load metadata for docker.io/library/php:latest             2.9s
--> [auth] library/php:pull token for registry-1.docker.io                 0.0s
--> [internal] load .dockignore                                           0.0s
--> => transferring context: 2B                                         0.0s
--> [internal] load build context                                         0.0s
--> => transferring context: 239B                                         0.0s
--> [1/2] FROM docker.io/library/php:latest@sha256:f9410ffdc12c9ab0b6c390a56e46a13ef58558f7c2abda1bc1bb3291a7d574 7.5s
--> => resolve docker.io/library/php:latest@sha256:f9410ffdc12c9ab0b6c390a56e46a13ef58558f7c2abda1bc1bb3291a7d574 0.0s
--> => sha256:39784e77cb587aa21fbc2c08484cb942f8ac995ef6ae047c76e310ba1b2d9f57 243B / 243B 0.2s
--> => sha256:148cff309ee18281f79dad295076b24642c5fd2397590122983241d60c2fc7b0 2.44kB / 2.44kB 0.4s
--> => sha256:d46a93356f6de34b1f6f260c8b3dee641b5f500d7a946a73c237f2cc9292f181 40.82MB / 40.82MB 2.8s
--> => sha256:4764feada77554c49215c0dbd9a918bbb81723b326076e9fd8ac72060b14b34 486B 0.7s
--> => sha256:46390c1686de4920966dd582368a1d249fa4eca0f6c4e79b9c6529882eb416e8 13.71MB / 13.71MB 1.2s
--> => sha256:d0177042fc93a4406f29d0f500498ab5c59119e586ed1a47f27465b709d1272d 227B / 227B 0.3s
--> => sha256:46f647fc8e6769fee87d7a4ef1c3e45c42c89eb5ee95d2f2b2b05595fd58b07 104.33MB / 104.33MB 3.9s
--> => sha256:a843aa4fd431d133dd91bbdc208fd84cefdb4b0d7f51706e2b8f20f3ee7bb30 224B / 224B 0.3s
--> => sha256:6e909acdb790c5a1989d9fcf795fda5a246ad6664bb27b5c688e2b734b2c5fad 28.20MB / 28.20MB 1.8s
--> => extracting sha256:6e909acdb790c5a1989d9fcf795fda5a246ad6664bb27b5c688e2b734b2c5fad 1.0s
--> => extracting sha256:a843aa4fd431d133dd91bbdc208fd84cefdb4b0d7f51706e2b8f20f3ee7bb30 0.0s

```

```

PS C:\Users\Admin\Desktop\PHP Hello Docker> docker run --name=php -p=3000:3000 vaibhav2352/phpimage1
[Tue Apr  1 10:27:45 2025] 8.4.5 Development Server (http://0.0.0.0:3000) started[Tue Apr  1 10:28:09 2025] 172.17.0.1:60298 Accepted
[Tue Apr  1 10:28:09 2025] 172.17.0.1:60310 [200]: GET /
[Tue Apr  1 10:28:09 2025] 172.17.0.1:60310 Closing
[Tue Apr  1 10:28:09 2025] 172.17.0.1:60298 [200]: GET /favicon.ico
[Tue Apr  1 10:28:09 2025] 172.17.0.1:40246 Accepted
[Tue Apr  1 10:28:20 2025] 172.17.0.1:40256 Accepted
[Tue Apr  1 10:28:20 2025] 172.17.0.1:40246 [200]: GET /
[Tue Apr  1 10:28:20 2025] 172.17.0.1:40246 Closing
[Tue Apr  1 10:28:20 2025] 172.17.0.1:40256 [200]: GET /favicon.ico
[Tue Apr  1 10:28:20 2025] 172.17.0.1:40256 Closing

```



More Docker. Easy Access. New Streamlined Plans. [Learn more](#)

Ask Gordon BETA

[Containers](#) [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Images

Volumes

Builds

Docker Hub

Docker Scout

Extensions

Ask Gordon BETA

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container memory usage 🕒
8.98MB / 7.49GB

Container CPU usage 🕒
0.00% / 1200% (12 CPUs available)

Only show running containers

Search

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
magical_stern	260ade8ddcb7c	ubuntu		0%	3 hours ago	⋮
musing_habit	40d5b3e94d44	ubuntu		0%	3 hours ago	⋮
agitated_goodall	e3e6a8c58a88	hello-world		0%	3 hours ago	⋮
php	e32e442ecc60	yahibhav2352/phpimage1	3000:30000	0%	3 minutes ago	⋮

Show 4 items

Walkthroughs

 Containerize your application
\$ docker init
3 mins

 Multi-container applications
8 mins

[View more in the Learning center](#)

Engine running : RAM 0.97 GB CPU 0.08% Disk: 4.58 GB used (limit 1006.85 GB)

> Terminal ⟳ New version available

DEVOPS

EXPERIMENT NO 8

22CE1307

Creating the Project Folder:

```
PS C:\Users\adity\Desktop\code\Dockers> mkdir PHP_Hello_Docker
```

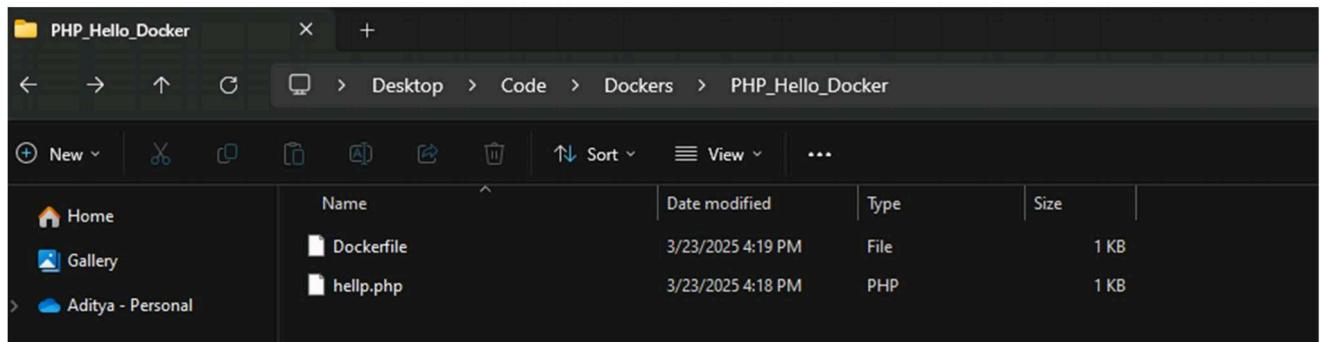
```
Directory: C:\Users\adity\Desktop\code\Dockers
```

Mode	LastWriteTime	Length	Name
d----	3/23/2025 4:17 PM		PHP_Hello_Docker

```
PS C:\Users\adity\Desktop\code\Dockers> cd PHP_Hello_Docker
```

```
PS C:\Users\adity\Desktop\code\Dockers\PHP_Hello_Docker> |
```

Create a Web Application File:



```
PS C:\Users\adity\Desktop\code\Dockers\PHP_Hello_Docker> ls
```

```
Directory: C:\Users\adity\Desktop\code\Dockers\PHP_Hello_Docker
```

Mode	LastWriteTime	Length	Name
-a----	3/23/2025 4:19 PM	245	Dockerfile
-a----	3/23/2025 4:18 PM	92	hellp.php

DockerFile:

hello.php

```
FROM php
COPY ./hello.php /var/www/html/
EXPOSE 3000
CMD ["php", "-S", "0.0.0.0:3000", "-t", "/var/www/html"]
```

```
<html>
<body>
    <?php echo "Hello world from PHP container!"; ?>
</body>
</html>
```

Building the Docker Image:

```
PS C:\Users\adity\Desktop\code\Dockers\PHP_Hello_Docker> docker build . -t adityasync/myfirstphp_image
[+] Building 24.1s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 284B
=> [internal] load metadata for docker.io/library/php:latest
=> [auth] library/php:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> transferring context: 2B
=> [internal] load build context
=> transferring context: 414B
=> [1/2] FROM docker.io/library/php:latest@sha256:f9410ffdc12c9ab0b6c390a56e46a13ef58558f7c2abda1bc1bb3291a7d57452
=>  resolve docker.io/library/php@latest@sha256:f9410ffdc12c9ab0b6c390a56e46a13ef58558f7c2abda1bc1bb3291a7d57452
=> sha256:39784e77cb587aa21fb2c09484c6942f8ac995efae047c76e310ba1b2d9f57 243B
=> sha256:46390c49215c0dd9a18bbb2b1723b326976e9fd8ac7206b01b4b34 486B
=> sha256:4764feada77554c49215c0dd9a18bbb2b1723b326976e9fd8ac7206b01b4b34 486B
=> sha256:d146a9356f6de34b1f6f250c8b3dee641b5f5f9007a946a73c237f2c929f181 40.82MB
=> sha256:148cff309ee13281f79dad295876b24642c5fd2397590122983241d69c2fc7b6 2.44kB
=> sha256:d0177842fc93a4406f29d0f508498a5c59119e586ed1a47f27465b709d1272d 227B
=> sha256:a843a84fd431d133d9d1b2bd208f84c3efdb1b0d7f51706e2b8f28f3ee7b5838 224B
=> sha256:46f6a7fc8e6769fee87d74ef1c3e45c2c89eb5ee95d2f2b2b65595fd58b07 104.33MB
=> sha256:6e999acd790c5a198949fc795fda5a246ad6664b0b7b5c688e2b734b2c5fad 28.20MB
=> extracting sha256:6e999acd790c5a198949fc795fda5a246ad6664b0b7b5c688e2b734b2c5fad
=> extracting sha256:a843a84fd431d133d9d1b2bd208f84c3efdb1b0d7f51706e2b8f20f3ee7b5830
=> extracting sha256:46fa7fc8e6769fee87d74ef1c3e45c2c89eb5ee95d2f2b2b65595fd58b07
=> extracting sha256:d0177842fc93a4406f29d0f508498a5c59119e586ed1a47f27465b709d1272d
=> extracting sha256:46390c49215c0dd9a18bbb2b1723b326976e9fd8ac7206b01b4b34
=> extracting sha256:4764feada77554c49215c0dd9a18bbb2b1723b326976e9fd8ac7206b01b4b34
=> extracting sha256:d146a9356f6de34b1f6f250c8b3dee641b5f5f9007a946a73c237f2c929f181
=> extracting sha256:148cff309ee13281f79dad295876b24642c5fd2397590122983241d69c2fc7b6
=> extracting sha256:a843a84fd431d133d9d1b2bd208f84c3efdb1b0d7f51706e2b8f28f3ee7b5838
=> extracting sha256:39784e77cb587aa21fb2c08484c6942f8ac995efae047c76e310ba1b2d9f57
=> [2/2] COPY ./ .
=> exporting to image
=> exporting layers
=> exporting manifest sha256:48af5a14a8a182c8d48907a8c2cfaf3cab11ff7e0224bf406d2d37d34c2ae367
=> exporting config sha256:21d97a55e569a75c7a59f97be5d18d0e4ed9ced17b38e2a9a0a740c44e2
=> exporting attestation manifest sha256:74a151b9396f8184a5c734c6feff3c7f43ffff0f07704beeb58c1f78abef8b8e
=> exporting manifest list sha256:320996d117b0f0f72266bac4d2b97668978172ceabdb91ea93e68774adacc15c
=> naming to docker.io/adityasync/myfirstphp_image:latest
=> unpacking to docker.io/adityasync/myfirstphp_image:latest
0.1s
```

Checking if the image is created:

PS C:\Users\adity\Desktop\code\Dockers\PHP_Hello_Docker> docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
adityasync/myfirstphp_image	latest	32096d117b40	About a minute ago	762MB
adityasync/my_custom_image	latest	5d221809610c	41 minutes ago	198MB
my_custom_image	latest	5d221809610c	41 minutes ago	198MB
ubuntu	latest	72297848456d	7 weeks ago	117MB

Running the Docker Container:

```
PS C:\Users\adity\Desktop\code\dockers\PHP_Hello_Docker> docker run --name=php -p 3000:3000 adityasync/myfirstphp_image
[Sun Mar 23 11:20:21 2025] PHP 8.4.5 Development Server (http://0.0.0.0:3000) started
[Sun Mar 23 11:20:25 2025] 172.17.0.1:56222 Accepted
[Sun Mar 23 11:20:25 2025] 172.17.0.1:56222 [200]: GET /hello.php
[Sun Mar 23 11:20:25 2025] 172.17.0.1:56222 Closing
```

Viewing the Web Application:



Hello world from PHP container!

Stopping & Removing the Container:

```
PS C:\Users\adity\Desktop\code\dockers\PHP_Hello_Docker> docker stop php
php
PS C:\Users\adity\Desktop\code\dockers\PHP_Hello_Docker> docker rm php
php
PS C:\Users\adity\Desktop\code\dockers\PHP_Hello_Docker> |
```

Push the Image to Docker Hub:

```
PS C:\Users\adity\Desktop\code\dockers\PHP_Hello_Docker> docker push adityasync/myfirstphp_image
Using default tag: latest
The push refers to repository [docker.io/adityasync/myfirstphp_image]
ab096cc6ec91: Pushed
d46a93356f6d: Pushed
d0177042fc93: Pushed
a843a84fd431: Pushed
148cff309ee1: Pushed
46f6a7fc8e67: Pushed
57078ad008e2: Pushed
6e909acdb790: Pushed
4764feada775: Pushed
46390c1686de: Pushed
39784e77cb58: Pushed
latest: digest: sha256:900151cadad9d0455c5347e77a460b9ce2c4ab614afe28de40339197e04c32cb size: 856
```

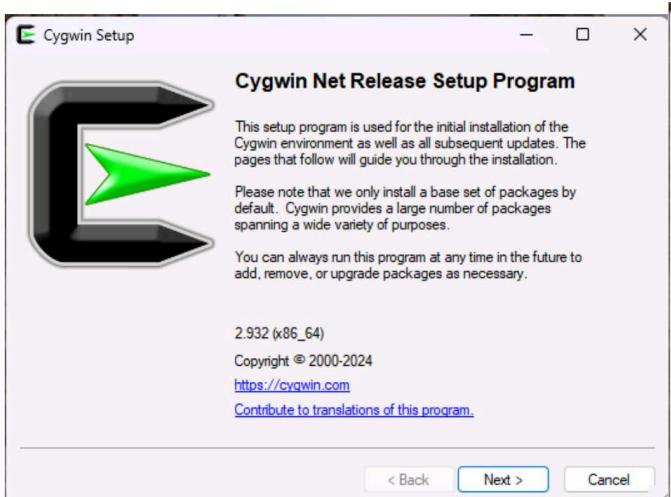
9o- Experiment 9: Configuration Management with Ansible/Chef/Puppet

Aim: To install and configure pull-based software configuration management and provisioning tools using Ansible, Chef, or Puppet.

Steps (Ansible-focused):

1. Install **Cygwin** or WSL on Windows and enable Ansible (or use Linux terminal).
2. Install Ansible:
3. sudo apt update
4. sudo apt install ansible
5. Create an **inventory file** listing target machines (e.g., localhost).
6. Create a **playbook** (YAML file) to install packages or configure services:
7. - name: Install and start Apache
8. hosts: localhost
9. become: yes
10. tasks:
11. - name: Install Apache
12. apt:
13. name: apache2
14. state: present
15. - name: Start Apache
16. service:
17. name: apache2
18. state: started
19. Run the playbook:
20. ansible-playbook playbook.yml
21. Verify that services were installed and started correctly.

Conclusion: In this experiment, we practiced Infrastructure as Code (IaC) using Ansible. We automated system configuration and software provisioning, which is essential for consistent environment setup, efficient scaling, and reducing human errors in DevOps pipelines.

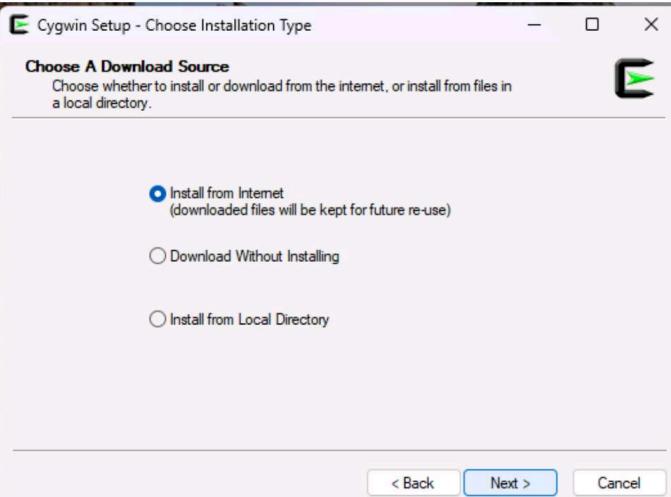


This setup program is used for the initial installation of the Cygwin environment as well as all subsequent updates. The pages that follow will guide you through the installation.

Please note that we only install a base set of packages by default. Cygwin provides a large number of packages spanning a wide variety of purposes.

You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.

2.932 (x86_64)
Copyright © 2000-2024
<https://cygwin.com>
[Contribute to translations of this program.](#)

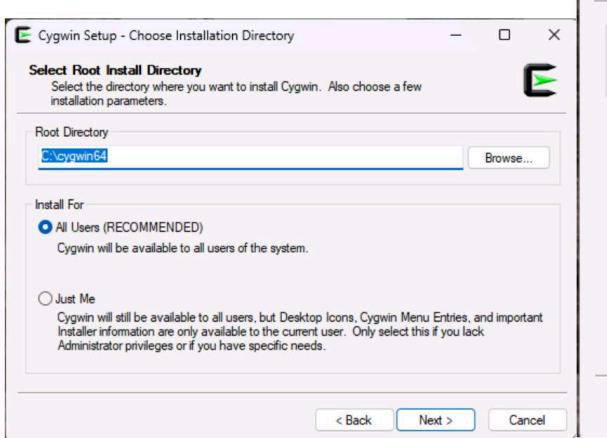


Choose A Download Source
Choose whether to install or download from the internet, or install from files in a local directory.

Install from Internet
(downloaded files will be kept for future re-use)

Download Without Installing

Install from Local Directory

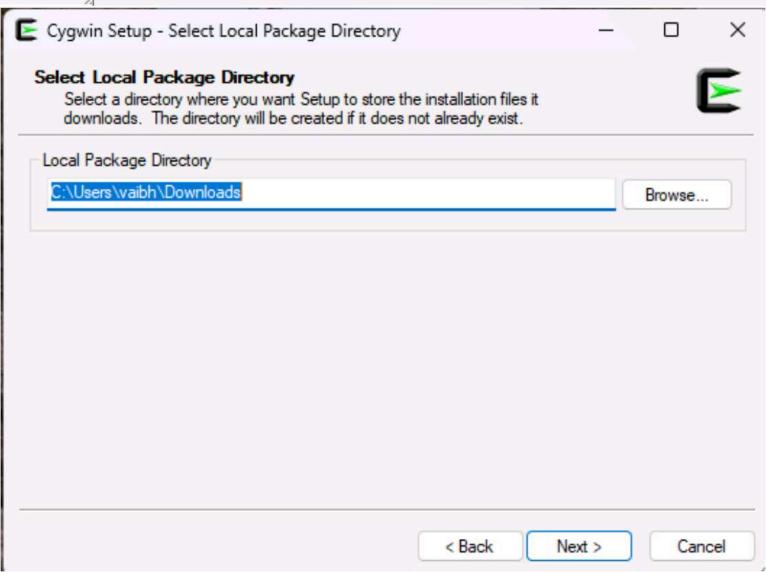


Select Root Install Directory
Select the directory where you want to install Cygwin. Also choose a few installation parameters.

Root Directory:

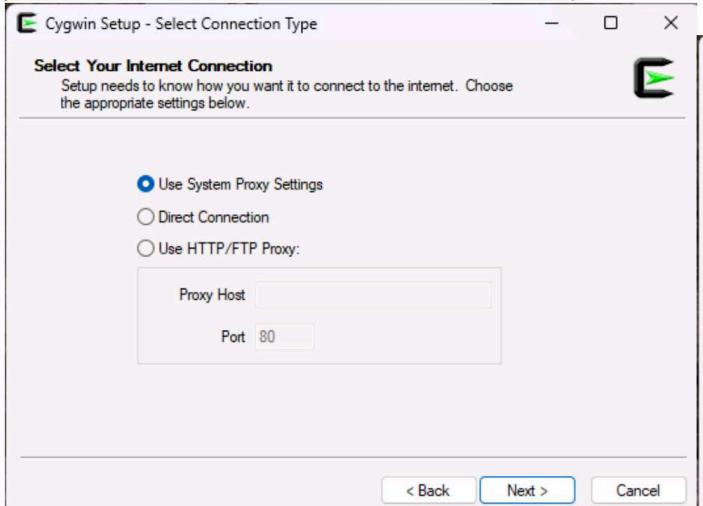
Install For:

- All Users (RECOMMENDED)
Cygwin will be available to all users of the system.
- Just Me
Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.



Select Local Package Directory
Select a directory where you want Setup to store the installation files it downloads. The directory will be created if it does not already exist.

Local Package Directory:



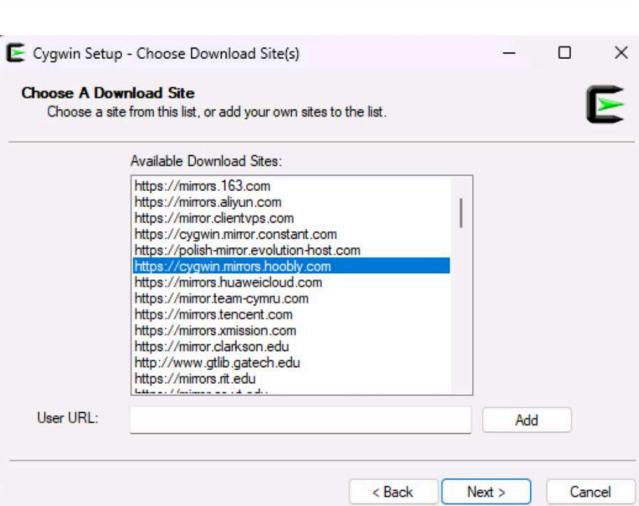
Select Your Internet Connection
Setup needs to know how you want it to connect to the internet. Choose the appropriate settings below.

Use System Proxy Settings

Direct Connection

Use HTTP/FTP Proxy:

Proxy Host:
Port:



Choose A Download Site
Choose a site from this list, or add your own sites to the list.

Available Download Sites:

```
Https://mirrors.163.com
Https://mirrors.aliyun.com
Https://mirror.clientvps.com
Https://cygwin.mirror.constant.com
Https://polish-mirror.evolution-host.com
Https://cygwin.mirror.hoobly.com
Https://mirrors.huaweicloud.com
Https://mirror.team-cymru.com
Https://mirrors.tencent.com
Https://mirrors.xmission.com
Https://mirror.clarkson.edu
Http://www.gtlb.gatech.edu
Https://mirrors.rt.edu
```

User URL:

Cygwin Setup - Select Packages

Select Packages
Select packages to install.

Package	Current	New	Src?	Categories	Size
ansible		2.8.4-1	<input type="checkbox"/>	Admin, Unmaintained	23,901
ansible-doc		2.8.4-1	<input checked="" type="checkbox"/>	Doc, Unmaintained	19,752

Hide obsolete packages

Cygwin Setup - Progress

This page displays the progress of the download or installation.

Installing...
ansible 2.8.4-1
/usr/lib/python3.7/site-packages/ansible/modules/database/...

Progress:
Total: 
Disk: 

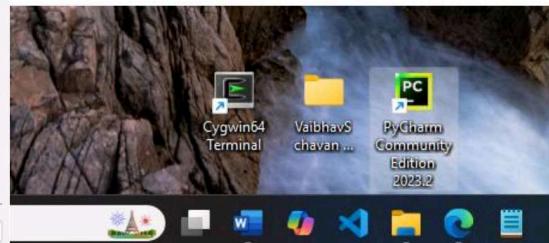
vaibh@LAPTOP-TQNHF5AE ~
\$ which ansible
/usr/bin/ansible

Cygwin Setup - Review and confirm changes

Review and confirm changes

Install _autorebase 001091-1
Install alternatives 1.31-1
Install base-cygwin 3.8-2
Install base-files 4.3-3
Install bash 5.2.21-1
Install bzip2 1.0.8-1
Install ca-certificates 2024.2.69_v8.0.401-1
Install coreutils 9.0-1
Install crypto-policies 20190218-1
Install cygutils 1.4.17-1
Install cygwin 3.6.0-1
Install dash 0.5.12-5

< Back Next > Cancel



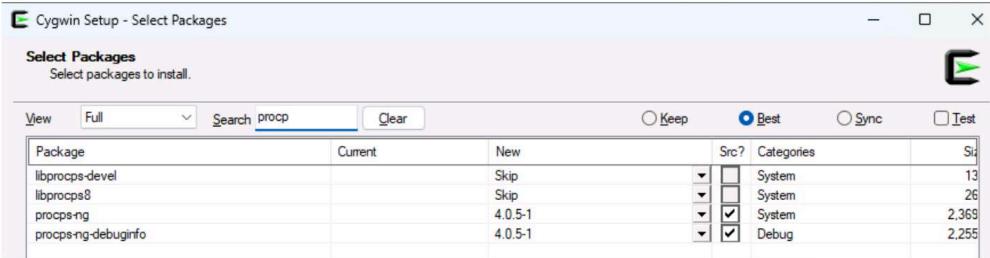
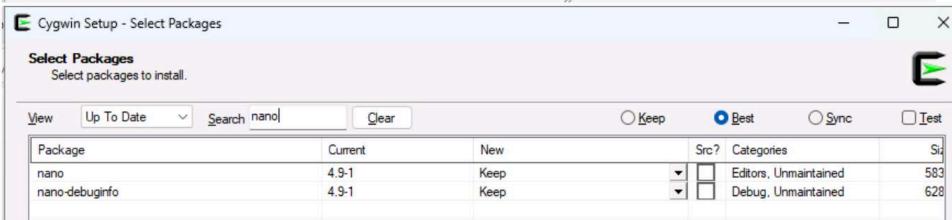
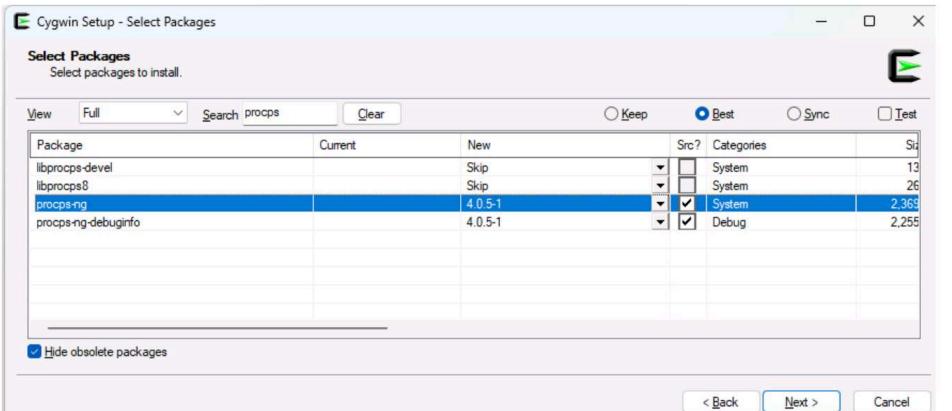
localhost:8080/manage/pluginManager/installed

Dashboard > Manage Jenkins > Plugins

Plugins

Search installed plugins

Name	Enabled
Ansible plugin 524.v9fa_a_4c989224	<input checked="" type="checkbox"/>
Invoke Ansible Ad-Hoc commands and playbooks. Report an issue with this plugin	<input checked="" type="checkbox"/>
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	
Ansible Tower Plugin 0.17.0	<input checked="" type="checkbox"/>
This plugin connects Jenkins with Ansible Tower Report an issue with this plugin	<input checked="" type="checkbox"/>



```

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

vaibh@LAPTOP-TQNHF5AE ~
$ ansible --version
ansible 2.8.4
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vaibh/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.7.12 (default, Nov 23 2021, 18:58:07) [GCC 11.2.0]

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m command -a "uptime"
localhost | CHANGED | rc=0 >>
11:57:50 up 5 days, 14:36, 0 users,  load average: 3.04, 3.04, 3.04

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m file -a "dest=/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir state=directory"
localhost | CHANGED => {
    "changed": true,
    "gid": 197609,
    "group": "vaibh",
    "mode": "0775",
    "owner": "vaibh",
    "path": "/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir",
    "size": 0,
    "state": "directory",
    "uid": 197609
}

```

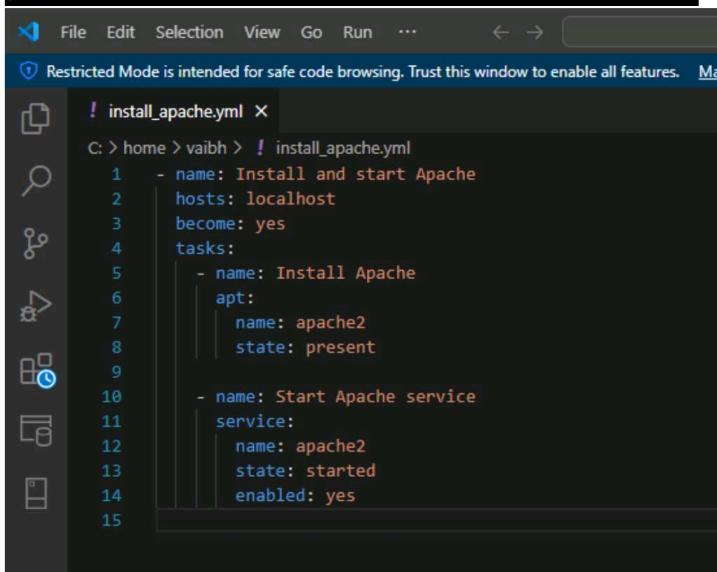
```

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m command -a "uptime"
localhost | CHANGED | rc=0 =>
11:57:50 up 5 days, 14:36, 0 users, load average: 3.04, 3.04, 3.04

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m file -a "dest=/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir state=directory"
localhost | CHANGED => {
  "changed": true,
  "gid": 197609,
  "group": "vaibh",
  "mode": "0775",
  "owner": "vaibh",
  "path": "/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir",
  "size": 0,
  "state": "directory",
  "uid": 197609
}

vaibh@LAPTOP-TQNHF5AE ~
$ ansible localhost -m file -a "dest=/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir state=directory"
localhost | SUCCESS => {
  "changed": false,
  "gid": 197609,
  "group": "vaibh",
  "mode": "0775",
  "owner": "vaibh",
  "path": "/cygdrive/c/ProgramData/Jenkins/.jenkins/workspace/testdir",
  "size": 0,
  "state": "directory",
  "uid": 197609
}

vaibh@LAPTOP-TQNHF5AE ~
$
```



```

! install_apache.yml ×
C: > home > vaibh > ! install_apache.yml
1   - name: Install and start Apache
2     hosts: localhost
3     become: yes
4     tasks:
5       - name: Install Apache
6         apt:
7           name: apache2
8           state: present
9
10      - name: Start Apache service
11        service:
12          name: apache2
13          state: started
14          enabled: yes
15
```

Dashboard > ansible_vaibhav_1261 > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Build Steps

Invoke Ansible Playbook

Ansible installation

Playbook path ?

Inventory

Do not specify Inventory

File or host list

Inline content

Host subset ?

Save **Apply**

Dashboard > ansible_vaibhav_1261 >

Permalinks

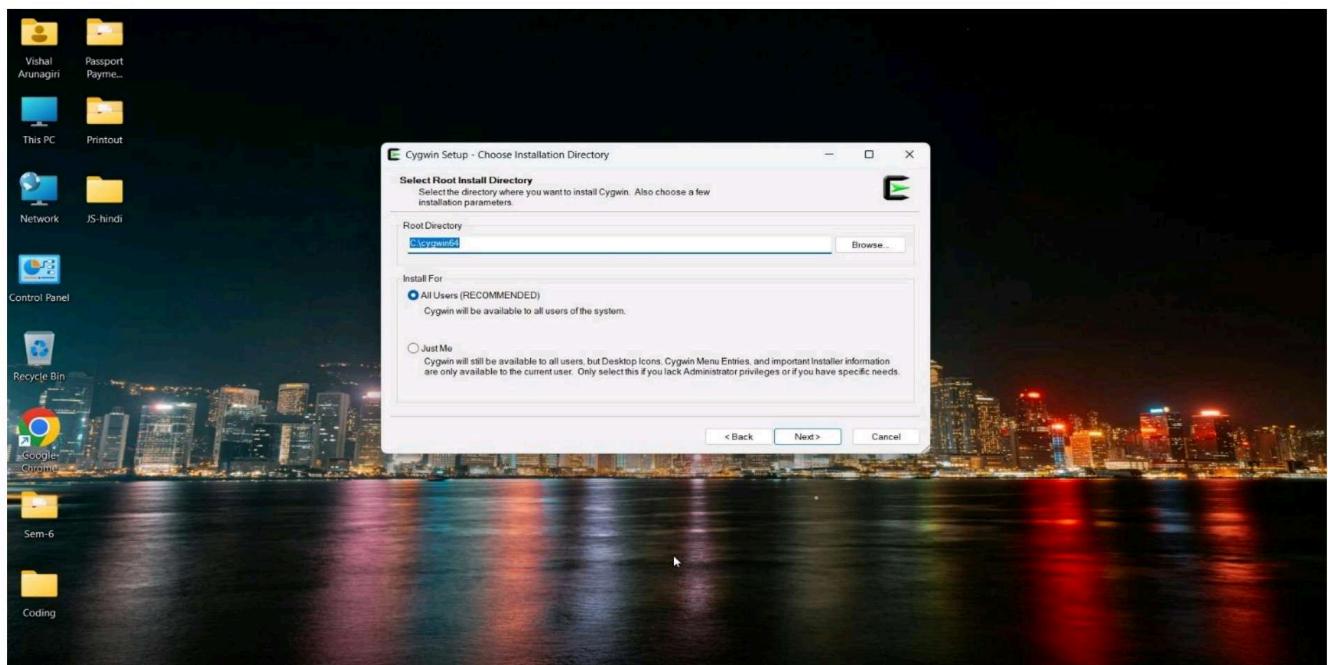
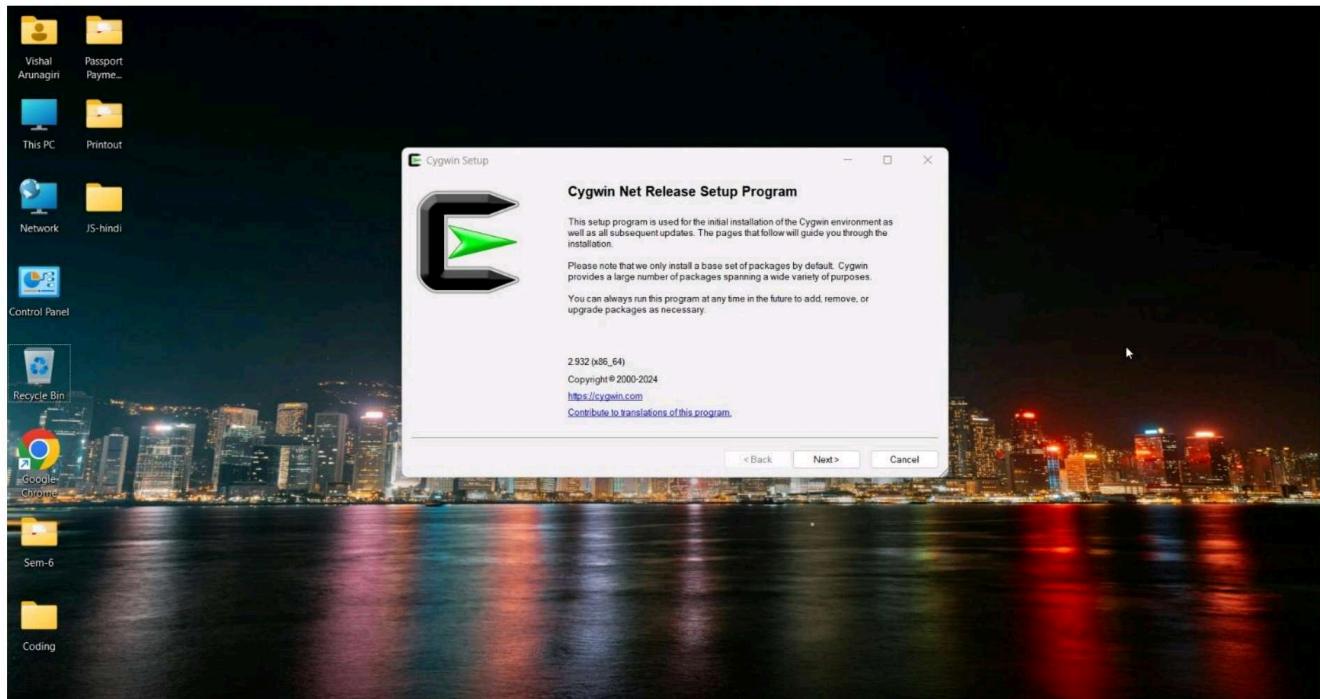
- Last build (#4), 3 min 36 sec ago
- Last failed build (#4), 3 min 36 sec ago
- Last unsuccessful build (#4), 3 min 36 sec ago
- Last completed build (#4), 3 min 36 sec ago

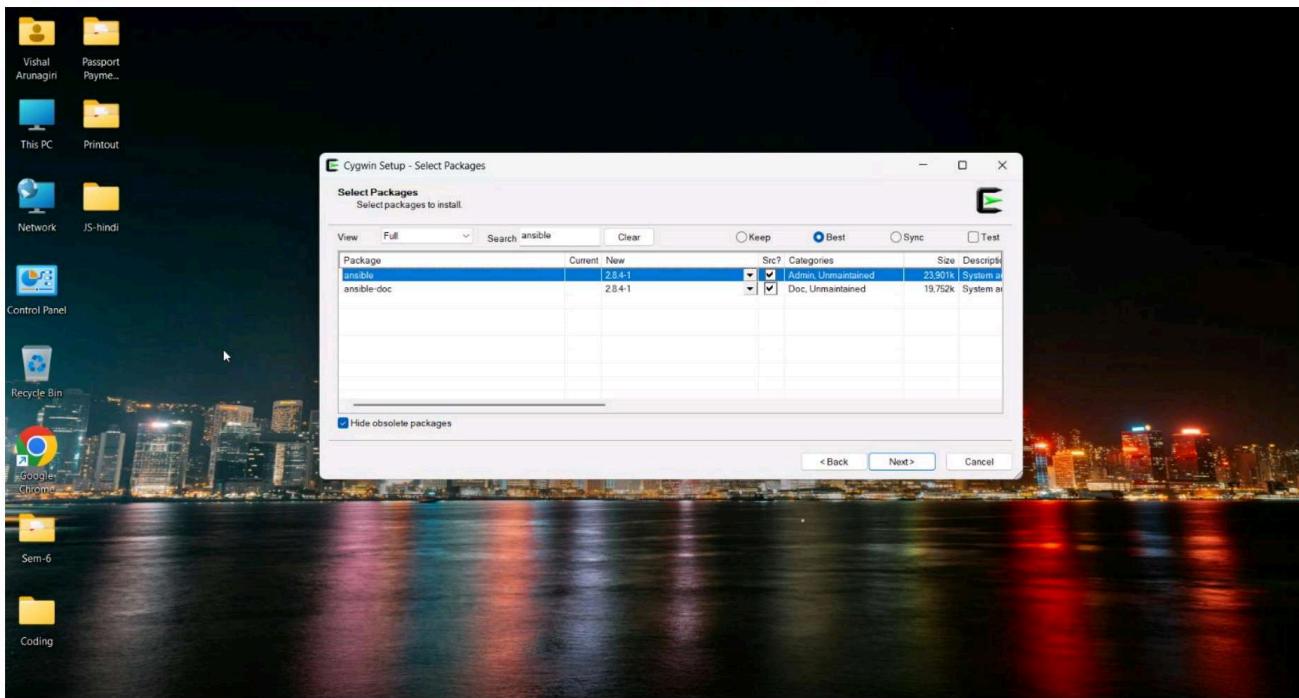
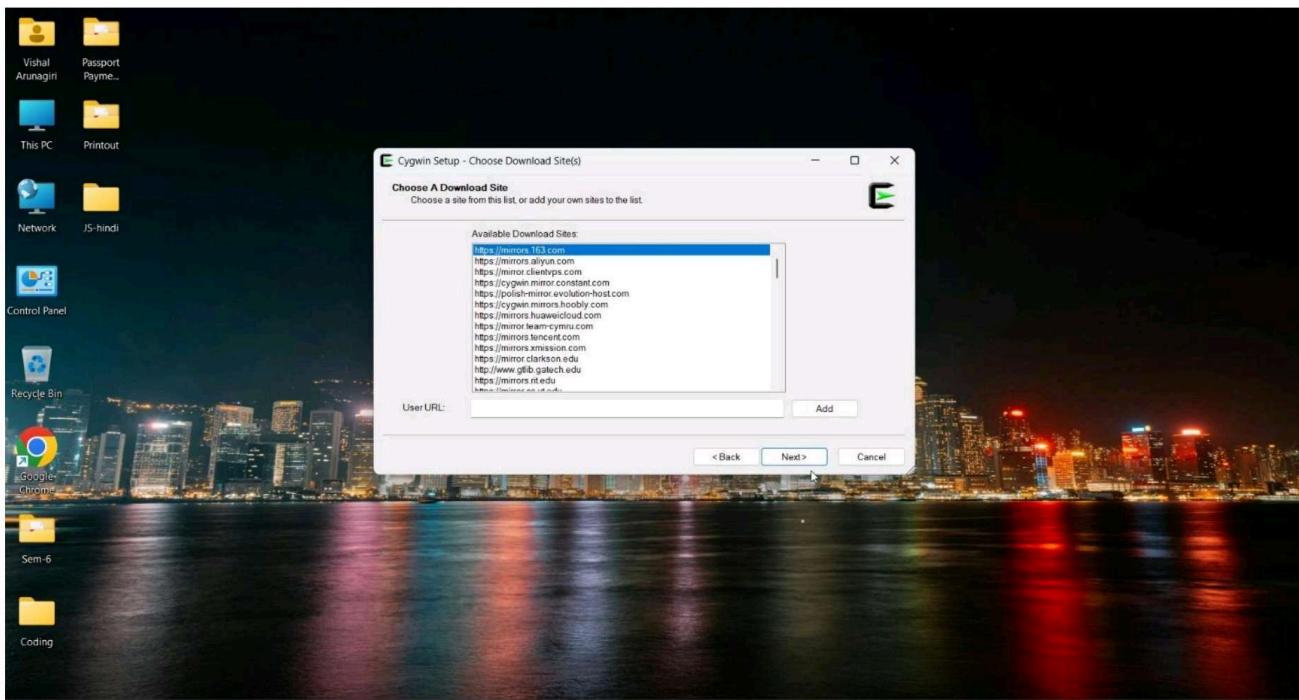
Builds

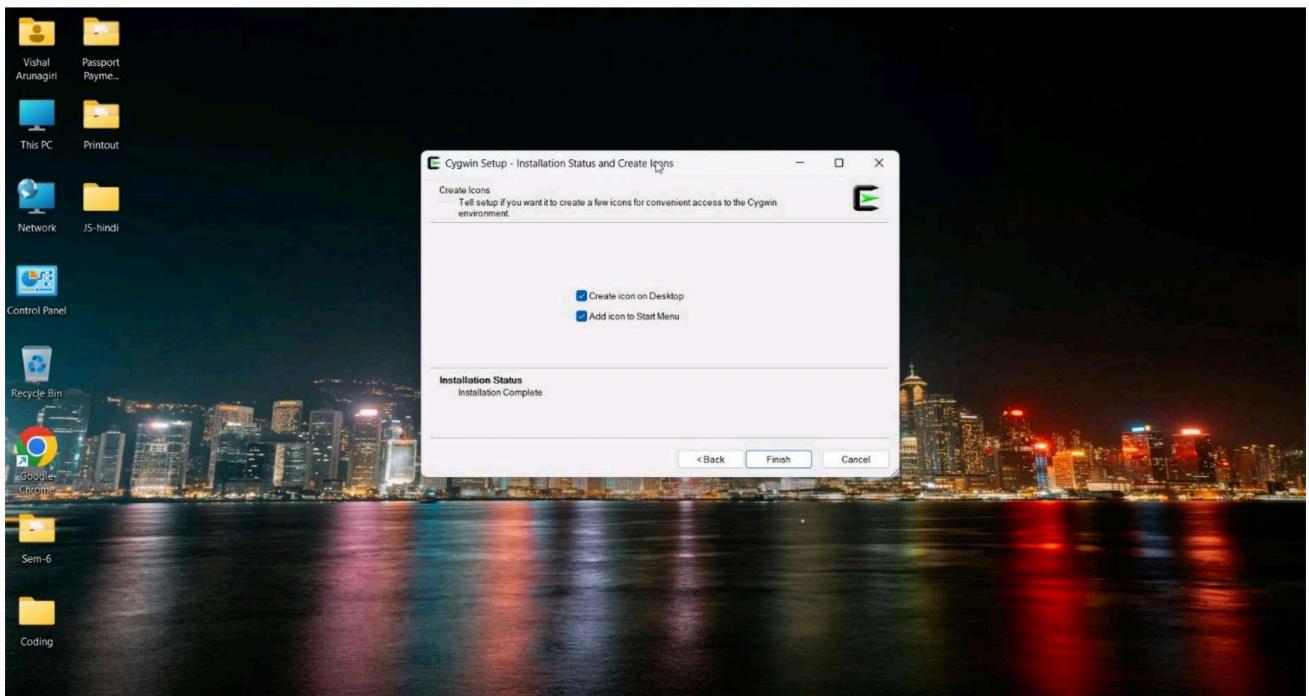
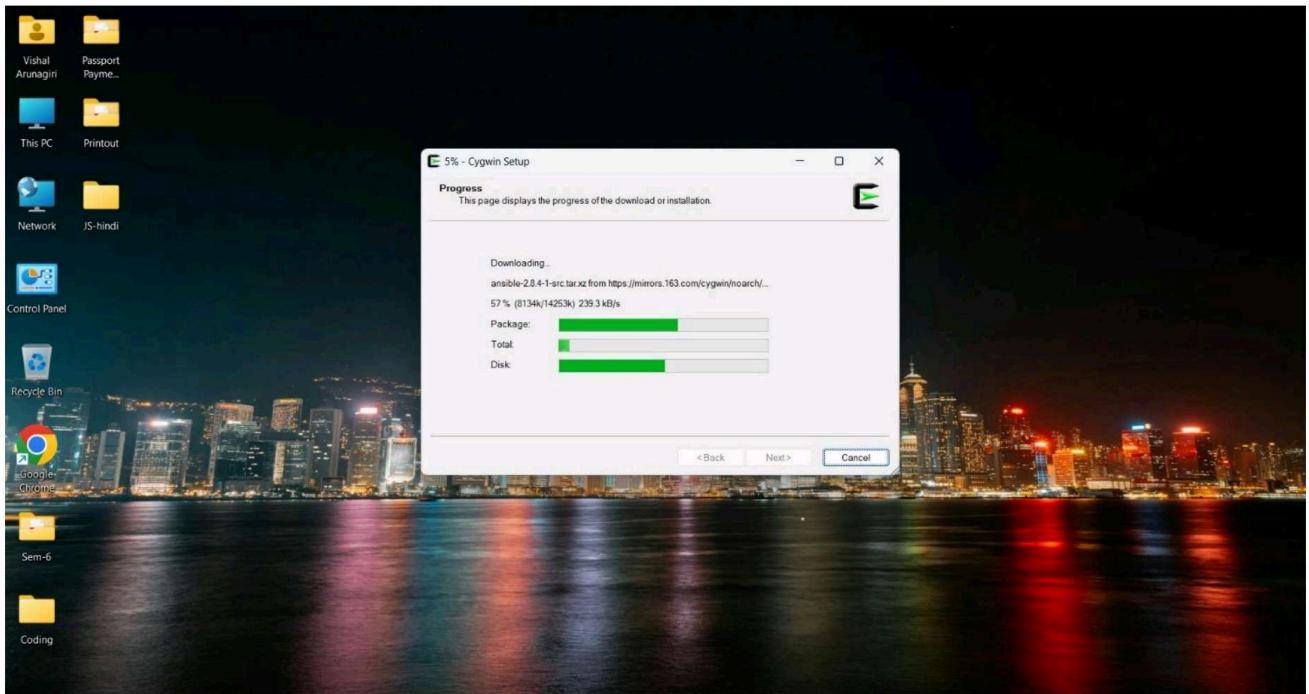
	Filter	/
Today		
✖	#4	1:14 PM
✖	#3	12:59 PM
✖	#2	12:58 PM
✖	#1	12:56 PM

Experiment – 9

Installation of Cygwin







Installation of Ansible Plugin in Jenkins

The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text 'ansible'. Below the search bar, there are two tabs: 'Available plugins' (which is selected) and 'Installed plugins'. A sidebar on the left includes links for 'Updates', 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main content area displays a table of available plugins. The first row in the table is for the 'Ansible' plugin, which is checked for installation. The table has columns for 'Install', 'Name', and 'Released'. The 'Ansible' entry includes a brief description: 'Invoke Ansible Ad-Hoc commands and playbooks.' The second row is for the 'Ansible Tower' plugin, which is not selected for installation. The Jenkins version '2.492.1' is visible at the bottom right.

The screenshot shows the Jenkins plugin manager interface with the 'Download progress' tab selected. On the left, there is a sidebar with links for 'Updates', 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main content area is titled 'Download progress' and shows the status of the 'Ansible' plugin download. It indicates that the 'Preparation' step is complete with three success items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. The 'Ansible' plugin itself is shown with a green checkmark and the status 'Success'. Below this, it says 'Loading plugin extensions' with another green checkmark and 'Success'. At the bottom, there are two options: a link to 'Go back to the top page' and a checkbox to 'Restart Jenkins when installation is complete and no jobs are running'. The Jenkins version '2.492.1' is visible at the bottom right.

Creating a Freestyle Job in Jenkins

The screenshot shows the Jenkins interface for creating a new item. The title bar says "New Item [Jenkins]". The URL in the address bar is "localhost:8080/view/all/newJob". The main heading is "New Item". A text input field says "Enter an item name" with "Ansible" typed in. Below it, a section titled "Select an item type" lists four options: "Freestyle project" (selected), "Maven project", "Pipeline", and "Multi-configuration project". Each option has a small icon and a brief description. At the bottom is a blue "OK" button.

Code of playbook.yml

```
playbook.yml
C: > cygwin64 > home > Vishal > playbook.yml
1 ---
2   - name: Check Ansible Version and Ping Hosts
3     hosts: localhost
4
5     tasks:
6       - name: Run 'ansible --version' on localhost
7         command: ansible --version
8         register: ansible_version_output
9
10      - name: Show Ansible version
11        debug:
12          var: ansible_version_output.stdout_lines
13
14      - name: Ping remote hosts
15        ping:
16
```

Invoke Playbook Ansible in Jenkins

The screenshot shows the Jenkins configuration page for a job named "Ansible Config". The left sidebar includes options like General, Source Code Management, Triggers, Environment, Build Steps (which is selected), and Post-build Actions. The main content area is titled "Build Steps" and contains a section for "Execute Windows batch command". The command entered is: `C:\cygwin64\bin\bash.exe -l -c "ansible-playbook /home/Vishal/playbook.yml"`. There is also an "Advanced" button and a "Add build step" link.

The screenshot shows the Jenkins dashboard for the "Ansible" project. The top navigation bar includes links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area displays the "Ansible" job status with a green checkmark icon. It lists the last four builds: Last build (#1), Last stable build (#1), Last successful build (#1), and Last completed build (#1), all from 52 seconds ago. A "Permalinks" section provides links to these builds. Below this is a "Builds" section with a "Filter" input field and a list of builds from today, including "#1 11:00 PM". The bottom right corner shows links for REST API and Jenkins 2.492.1.

An Ansible job named #1 is running on Jenkins. The console output shows the execution of an Ansible playbook.

```
Started by user Vishal Arunagiri
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Ansible
[Ansible] $ cmd /c call C:\WINDOWS\TEMP\jenkins12391470061126731616.bat
[WARNING]: provided hosts list is empty, only localhost is available. Note
that the implicit localhost does not match 'all'

PLAY [Check Ansible Version and Ping Hosts] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Run 'ansible --version' on localhost] ****
changed: [localhost]

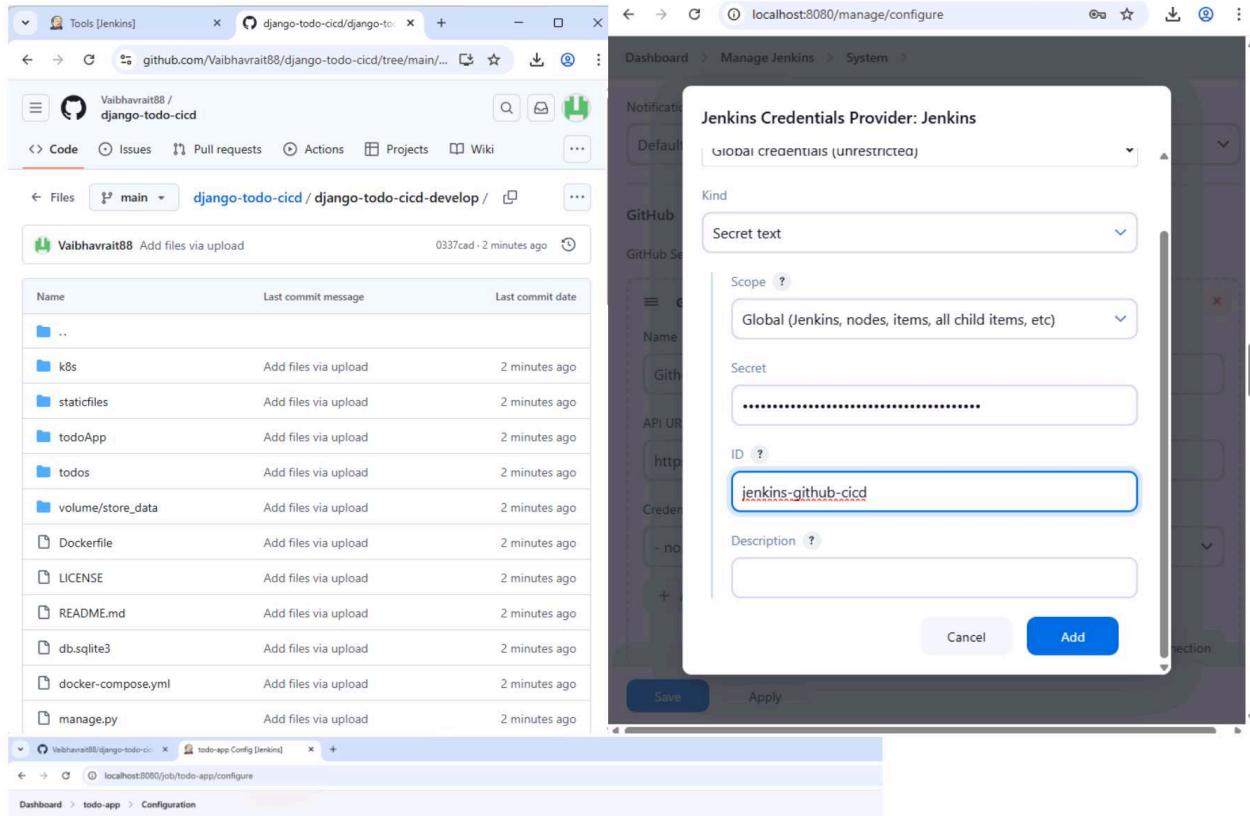
TASK [Show Ansible version] ****
ok: [localhost] => {
  "ansible_version_output.stdout_lines": [
    "ansible 2.8.4",
    "  config file = /etc/ansible/ansible.cfg",
    "  configured module search path = ['/home/SVSTBH/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']",
    "  ansible python module location = /usr/lib/python3.7/site-packages/ansible",
    "  executable location = /usr/bin/ansible",
    "  Python version = 3.7.12 (default, Nov 23 2021, 18:58:07) [GCC 11.2.0]"
  ]
}

TASK [Ping remote hosts] ****
ok: [localhost]

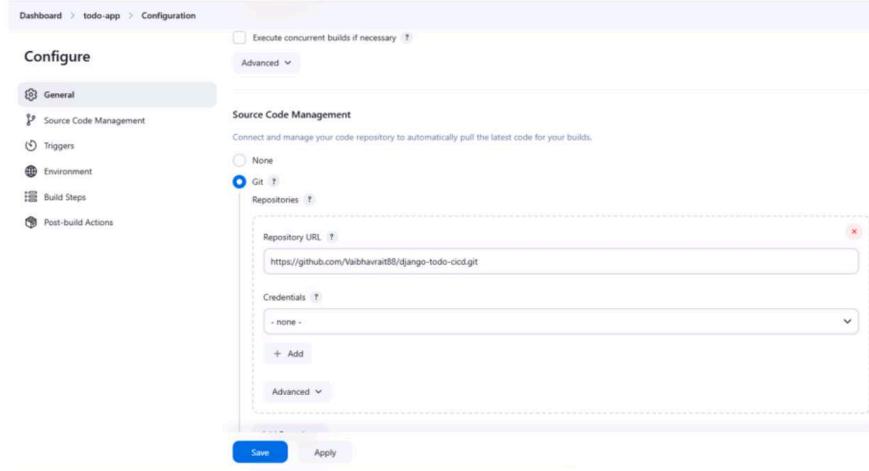
PLAY RECAP ****
localhost : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

C:\ProgramData\Jenkins\jenkins\workspace\Ansible>exit 0
Finished: SUCCESS
```

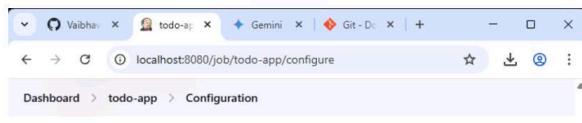
REST API Jenkins 2.492.1



The Jenkins dashboard shows a GitHub repository named "Vaibhavrait88/django-todo-cicd". A modal window titled "Jenkins Credentials Provider: Jenkins" is open, showing the configuration for a "Secret text" credential named "jenkins-github-cicd". The credential has a global scope and contains a secret value.

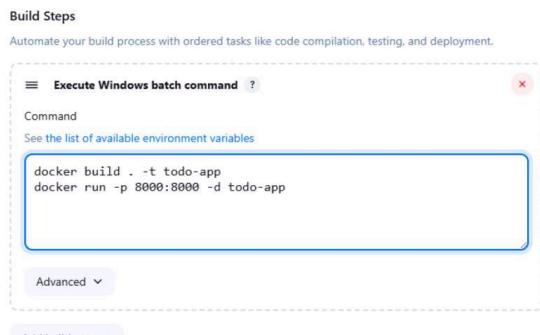


The Jenkins configuration page for the "todo-app" job shows the "Source Code Management" section. The "Git" connector is selected, and the repository URL is set to "https://github.com/Vaibhavrait88/django-todo-cicd.git". No credentials are currently assigned to this repository.

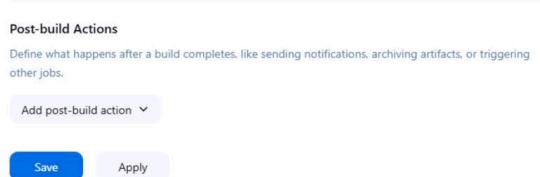


The Jenkins configuration page for the "todo-app" job shows the "Build Steps" section. It contains a single step: "Execute Windows batch command". The command entered is:

```
docker build . -t todo-app
docker run -p 8000:8000 -d todo-app
```



The Jenkins configuration page for the "todo-app" job shows the "Post-build Actions" section. It contains a single action: "Add post-build action".



The Jenkins configuration page for the "todo-app" job shows the "Post-build Actions" section. It contains a single action: "Add post-build action".



django-todo-cicd/Dockerfile at todo-app #12 Console [Jenkins] +

localhost:8080/job/todo-app/12/console

Jenkins

Dashboard > todo-app > #12 > Console Output

Console Output

Console Output

Started by user unknown or anonymous
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\workspace\todo-app
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\todo-app\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/Vaibhavrait88/django-todo-cicd # timeout=10
Fetching upstream changes from https://github.com/Vaibhavrait88/django-todo-cicd
> git.exe --version # timeout=10
> git --version # git version 2.48.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/Vaibhavrait88/django-todo-cicd +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision be9b99977a722a4794a02a99fa2c296daaf01e4fa (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f be9b99977a722a4794a02a90fac206daaf01e4fa # timeout=10
Commit message: "Update Dockerfile"
> git.exe rev-list --no-walk 989abb0180348e467df1f65129da4510dd141f0c2 # timeout=10
[todo-app] \$ cmd /c call C:\WINDOWS\TEMP\jenkins12995650107660579484.bat

C:\ProgramData\Jenkins\workspace\todo-app>docker build -t todo-app .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile

django-todo-cicd/Dockerfile at todo-app #12 Console [Jenkins] +

localhost:8080/job/todo-app/12/console

Jenkins

Dashboard > todo-app > #12 > Console Output

Console Output

```
#12 0.000      HINT: Configure the DEFAULT_AUTO_FIELD setting or the TodosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
#12 0.645  Operations to perform:
#12 0.645    Apply all migrations: admin, auth, contenttypes, sessions, todos
#12 0.645  Running migrations:
#12 0.645    No migrations to apply.
#12 DONE 0.7s

#13 exporting to image
#13 exporting layers
#13 exporting layers 3.6s done
#11 exporting manifest sha256:5bb55c8dd46268999604f098b69c7a9b5fd0c43e06b5eef1d367f68273e3ae71 done
#11 exporting config sha256:17eb9a15f349e4582664a175d493b421baa17df04287641995dc29b2e0df done
#13 exporting attestation manifest sha256:65bf908e667a5e2d485a75642e0b5a6554d39ba7a701d00b62c5d81ec9cad81 0.0s done
#13 exporting manifest list sha256:7fe39acae8749ccca866802766bf7azd91bf02c1de3a20bbc82623644c2212f3e 0.0s done
#13 naming to docker.io/library/todo-app:latest
#13 unpacking to docker.io/library/todo-app:latest
#13 unpacking to docker.io/library/todo-app:latest 2.5s done
#13 DONE 6.1s

C:\ProgramData\Jenkins\workspace\todo-app>docker run -p 8000:8000 -d todo-app
3580cedf0824f744b328431b1615715e094c5287cd5daeca78006cb019d3bb53d

C:\ProgramData\Jenkins\workspace\todo-app>exit
Finished: SUCCESS
```

35°C Hazel

Search

docker:desktop PERSONAL

More Docker. Easy Access. New Streamlined Plans. Learn more ↗

Containers

Containers [Give feedback ↗](#)

View all your running containers and applications. [Learn more ↗](#)

Container CPU usage: 0.45% / 1200% (12 CPUs available)

Container memory usage: 92.14MB / 7.49GB

Container memory usage: 92.14MB / 7.49GB

Container CPU usage: 0.45% / 1200% (12 CPUs available)

Container memory usage: 92.14MB / 7.49GB

	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	php	e32e442ecc60	vaibhav/23s	3000:30000	
<input type="checkbox"/>	magical_shtern	260ade8dcbb7c	ubuntu		
<input type="checkbox"/>	musing_habit	40d5b3e94d44	ubuntu		
<input type="checkbox"/>	agitated_goodal	e3e6a8c58a88	hello-world		
<input checked="" type="checkbox"/>	happy_chaum	3580cedf0824	todo-app	8000:8000	

Showing 5 items

REST API Jenkins 2.492.2

Sign in - Google Todo list

localhost:8000/todos/

Todo List - for batch 5

Do laundry

Add

<input type="checkbox"/> submit expt10 to maam vaibhav before 3pm	
<input type="checkbox"/> Docker file banao	
<input type="checkbox"/> Send Resume Google now !	
<input type="checkbox"/> Hacktoberfest Updates	

Git Tutorial

09 May 2023 11:54

What is Git?

Git is a popular **version control system**. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

What does Git do?

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

Working with Git

- Initialize Git on a folder, making it a **Repository**
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered **modified**
- You select the modified files you want to **Stage**
- The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

Git Install

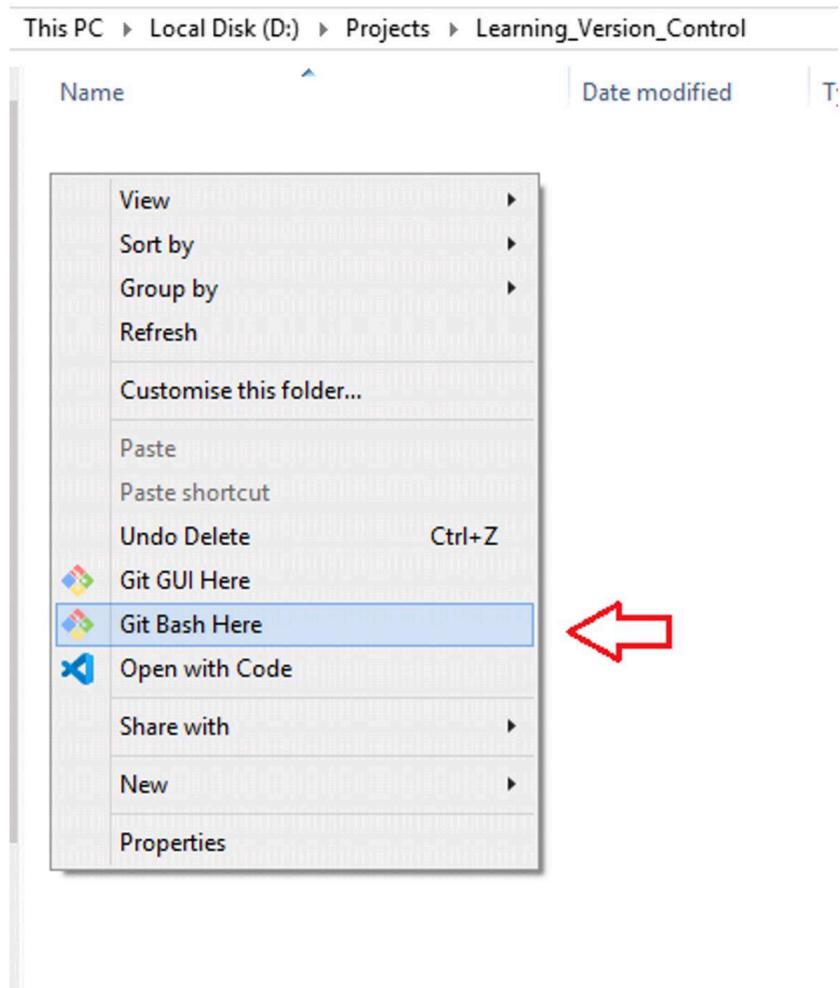
You can download Git for free from the following website: <https://www.git-scm.com/>

Using Git with Command Line

To start using Git, we are first going to open up our Command shell. For Mac and Linux you can use the built-in terminal.

For Windows, you can use Git bash, which comes included in Git for Windows.

Once you have navigated to the correct folder -> Right click -> Select "Git Bash Here"



The first thing we need to do, is to check if Git is properly installed:

Command: `git --version`

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control
$ git --version
git version 2.33.0.windows.2
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control
$ |
```

Configure Git

Now let Git know who you are.

This is important for version control systems, as each Git commit uses this information:

Command: `git config --global user.name "b-sachin"`
`git config --global user.email "sachinbhopiofficial@gmail.com"`

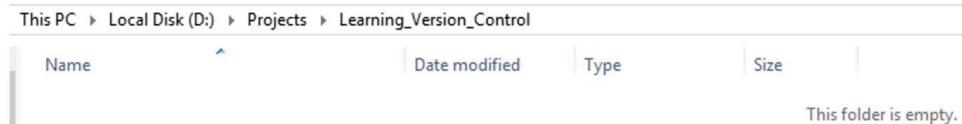
```
MINGW64:/d/Projects/Learning_Version_Control
(base)
$ git config --global user.name "b-sachin"
(base)
$ git config --global user.email "sachinbhopiofficial@gmail.com"
(base)
$ |
```

Change the user name and e-mail address to your own. You will probably also want to use this when registering to GitHub later on.

Initialize Git

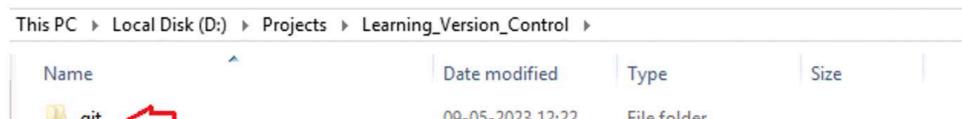
Command: `git init`

Before:



```
MINGW64:/d/Projects/Learning_Version_Control
$ git config user.name "b-sachin"
fatal: not in a git directory
(base)
$ git config -- user.name "b-sachin"
fatal: not in a git directory
(base)
$ |
```

After:



```
MINGW64:/d/Projects/Learning_Version_Control
$ git config -- user.name "b-sachin"
fatal: not in a git directory
(base)
$ git init
Initialized empty Git repository in D:/Projects/Learning_Version_Control/.git/
(base)
$ |
```

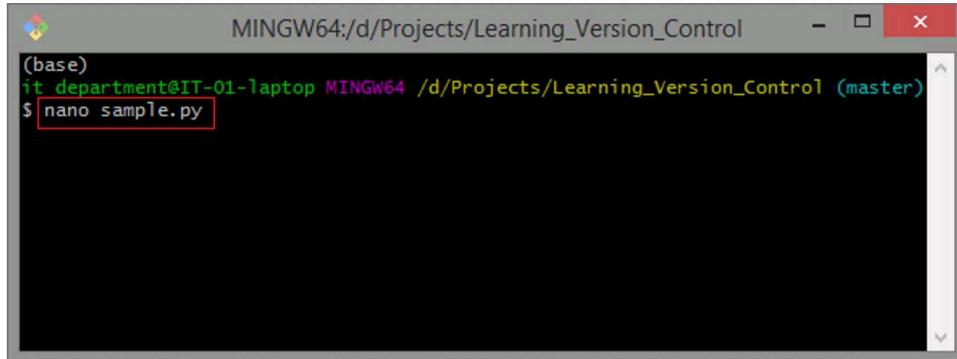
Git Adding New Files

You just created your first local Git repo. But it is empty.
So let's add some files,

To use **nano editor** inside bash prompt to create python file

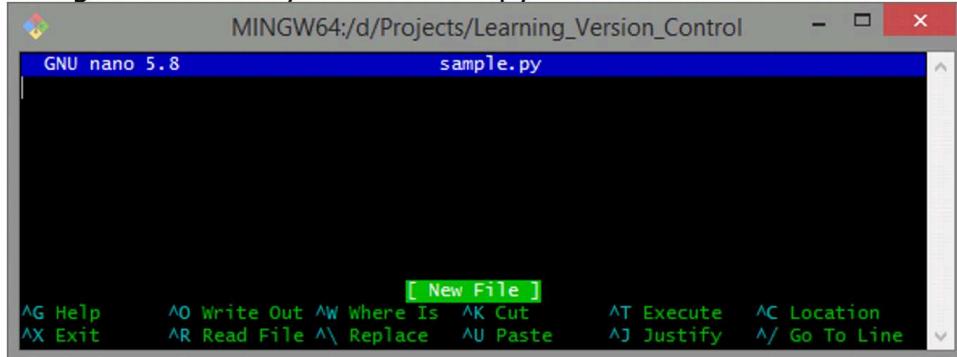
Command: **nano <file_name>**

```
nano sample.py
```



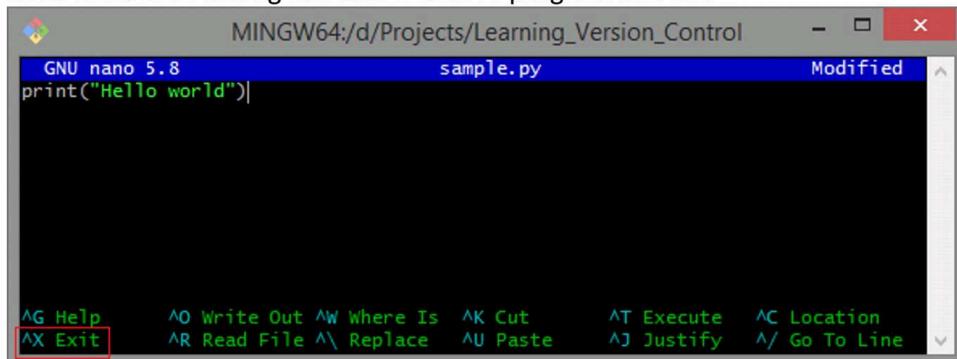
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
\$ nano sample.py

Using nano editor you can write python code



GNU nano 5.8 sample.py [New File]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Once done with coding use **Ctrl+X** to save program and exit



GNU nano 5.8 sample.py Modified
print("Hello world")
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Press **Y** to save code

Save modified buffer? |
Y Yes
N No ^C Cancel

MINGW64:/d/Projects/Learning_Version_Control

GNU nano 5.8 sample.py Modified

Press **Enter** to come out to bash prompt

File Name to Write: sample.py|
^G Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T Browse

MINGW64:/d/Projects/Learning_Version_Control

GNU nano 5.8 sample.py Modified

Here you can see new python file with name sample.py has been created.

Name	Date modified	Type	Size
.git	09-05-2023 12:22	File folder	
sample	09-05-2023 12:37	PY File	1 KB

(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
\$ nano sample.py
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
\$ |

MINGW64:/d/Projects/Learning_Version_Control

To use **VS Code editor**

MINGW64:/d/Version_Control

```
(base)
it_department@IT-01-laptop MINGW64 /d/Version_Control (master)
$ code sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Version_Control (master)
$
```

sample.py - Visual Studio Code

sample.py

```
1 print("Hello world")
2
```

Restricted Mode

To keep VS Code as default editor:

MINGW64:/d/Projects/Learning_Version_Control

```
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git config --global core.editor "code --wait"
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

You can also use **ls** command to see all files and folder inside working directory.

MINGW64:/d/Projects/Learning_Version_Control

```
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ nano sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ ls ←
sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Then we check the Git **status** and see if it is a part of our repo:

MINGW64:/d/Projects/Learning_Version_Control

```
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Now Git is **aware** of the file, but has not **added** it to our repository!

Files in your Git repository folder can be in one of 2 states:

- Tracked - files that Git knows about and are added to the repository
- Untracked - files that are in your working directory, but not added to the

repository

When you first add files to an empty repository, they are all untracked. To get Git to track them, you need to stage them, or add them to the staging environment.

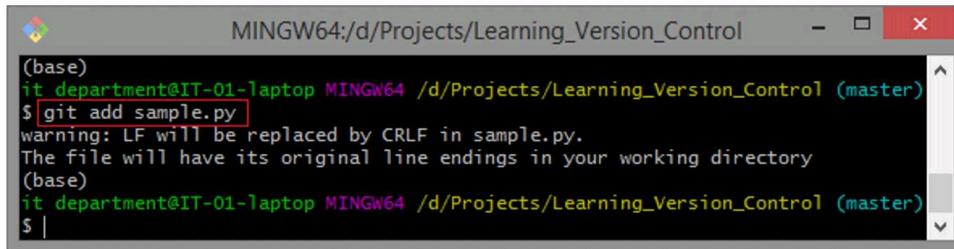
Git Staging Environment

One of the core functions of Git is the concepts of the Staging Environment, and the Commit.

As you are working, you may be adding, editing and removing files. But whenever you hit a milestone or finish a part of the work, you should add the files to a Staging Environment.

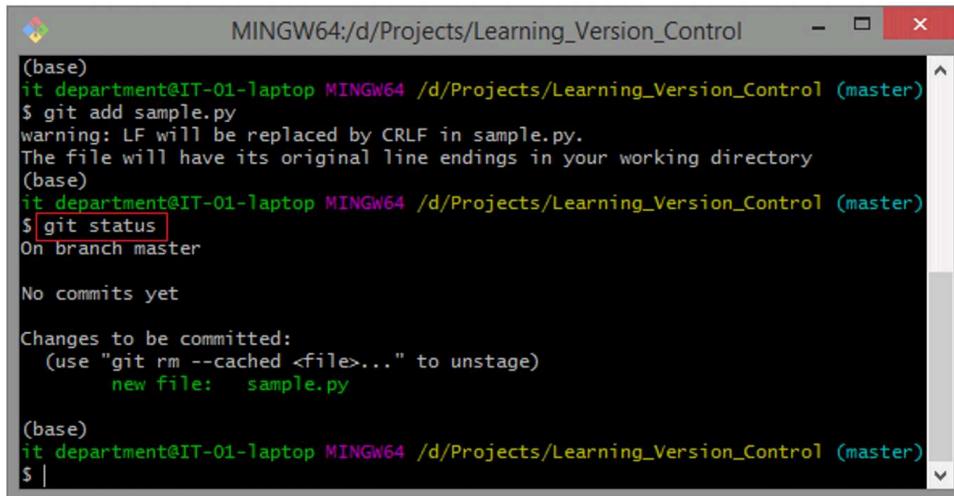
Staged files are files that are ready to be **committed** to the repository you are working on.

For now, we are done working with `sample.py`. So we can add it to the Staging Environment:



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git add sample.py
warning: LF will be replaced by CRLF in sample.py.
The file will have its original line endings in your working directory
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

The file should be **Staged**. Let's check the status:



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git add sample.py
warning: LF will be replaced by CRLF in sample.py.
The file will have its original line endings in your working directory
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   sample.py

(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

To **Unstage**:

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git rm --cache sample.py
rm 'sample.py'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status
On branch master

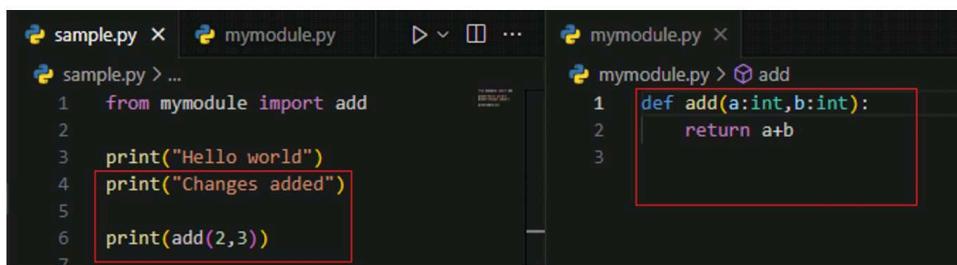
No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

git add for More than One File

Lets add one more file mymodule.py in project and make changes in sample.py file accordingly



```
sample.py X mymodule.py
sample.py > ...
1   from mymodule import add
2
3   print("Hello world")
4   print("Changes added")
5
6   print(add(2,3))

mymodule.py X
mymodule.py > ⌂ add
1 def add(a:int,b:int):
2     return a+b
```

1. By adding multiple files seperated by space

Command: `git add sample.py mymodule.py`

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git add sample.py mymodule.py
warning: LF will be replaced by CRLF in mymodule.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in sample.py.
The file will have its original line endings in your working directory
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status
On branch master

No commits yet

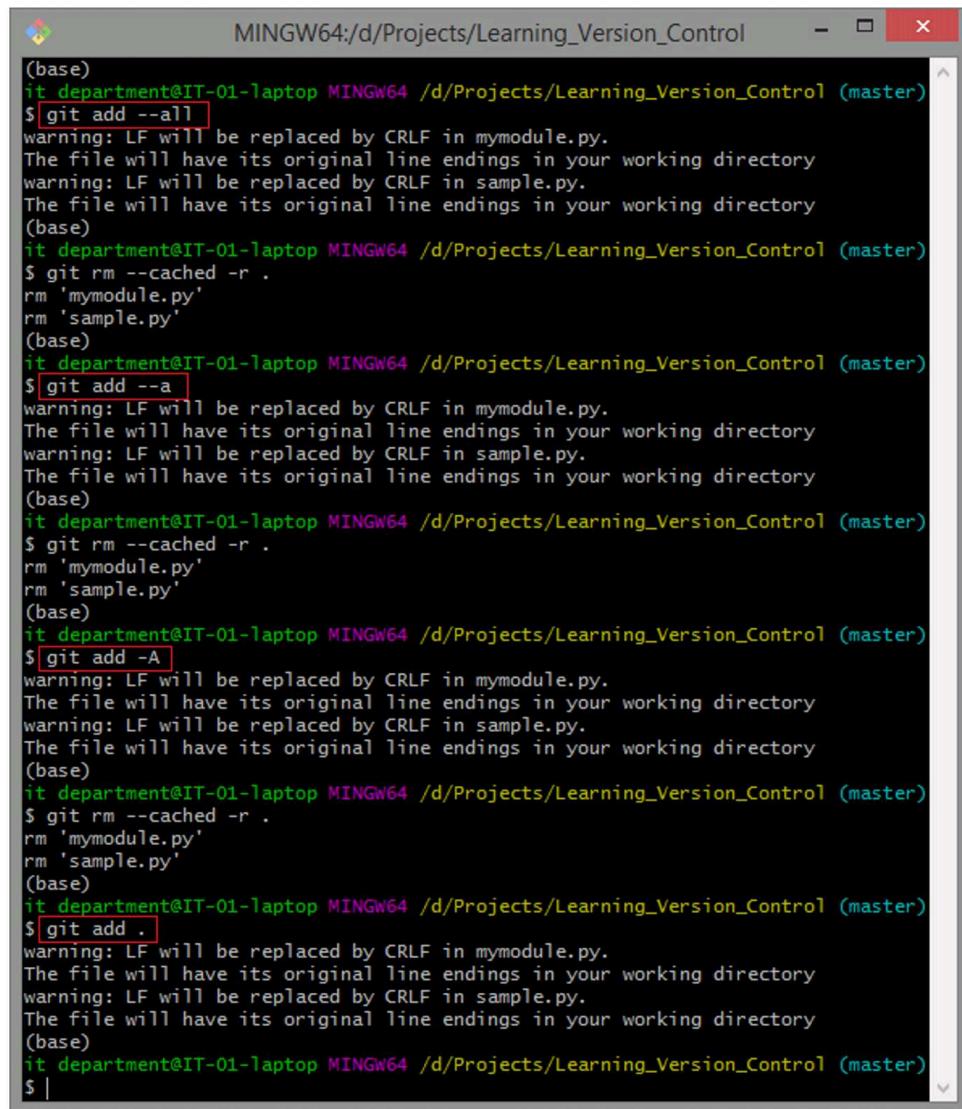
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:   mymodule.py
  new file:   sample.py

(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git rm --cached mymodule.py sample.py
rm 'mymodule.py'
rm 'sample.py'
```

2. By adding multiple files seperated by using one of the following ways

Command: `git add --all`

```
git add --a  
git add -A  
git add .
```



The screenshot shows a terminal window titled "MINGW64:/d/Projects/Learning_Version_Control". It displays a sequence of git commands and their outputs. The commands include `git add --all`, `git rm --cached -r`, `git add --a`, `git add -A`, and `git add .`. Each command is followed by a warning message about LF being replaced by CRLF in files like `mymodule.py` and `sample.py`, and a note that the file will have its original line endings in the working directory.

```
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git add --all  
warning: LF will be replaced by CRLF in mymodule.py.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in sample.py.  
The file will have its original line endings in your working directory  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git rm --cached -r .  
rm 'mymodule.py'  
rm 'sample.py'  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git add --a  
warning: LF will be replaced by CRLF in mymodule.py.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in sample.py.  
The file will have its original line endings in your working directory  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git rm --cached -r .  
rm 'mymodule.py'  
rm 'sample.py'  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git add -A  
warning: LF will be replaced by CRLF in mymodule.py.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in sample.py.  
The file will have its original line endings in your working directory  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ git add .  
warning: LF will be replaced by CRLF in mymodule.py.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in sample.py.  
The file will have its original line endings in your working directory  
(base)  
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)  
$ |
```

Git Commit

Since we have finished our work, we are ready move from **stage** to **commit** for our repo.

Adding commits keep track of our progress and changes as we work.

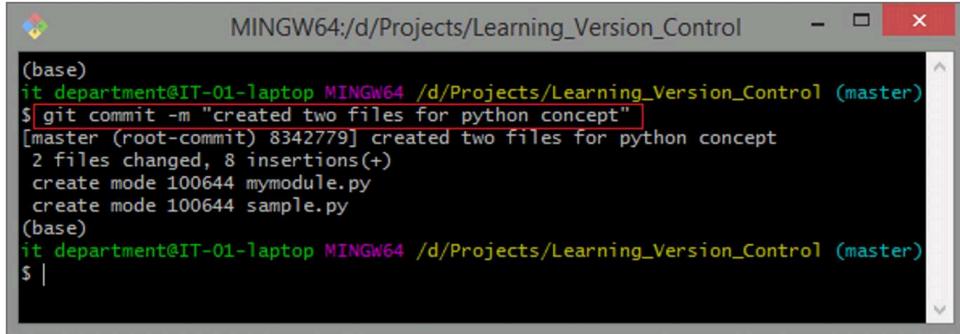
Git considers each **commit** change point or "save point".

It is a point in the project you can go back to if you find a bug, or want to make a change.

When we **commit**, we should **always** include a **message**.

By adding clear messages to each **commit**, it is easy for yourself (and others) to see what has changed and when.

Command: `git commit -m "created two files for python concept"`



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git commit -m "created two files for python concept"
[master (root-commit) 8342779] created two files for python concept
 2 files changed, 8 insertions(+)
  create mode 100644 mymodule.py
  create mode 100644 sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Git Commit without Stage

Sometimes, when you make small changes, using the staging environment seems like a waste of time.

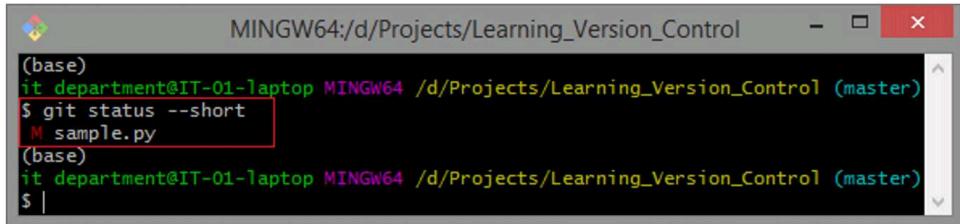
It is possible to commit changes directly, skipping the staging environment.

The `-a` option will automatically stage every changed, already tracked file.

Let's add a small update to `sample.py` file: (one print statement added in `sample.py` file)

And check the status of our repository. But this time, we will use the `--short` option to see the changes in a more compact way:

Command: `git status --short`



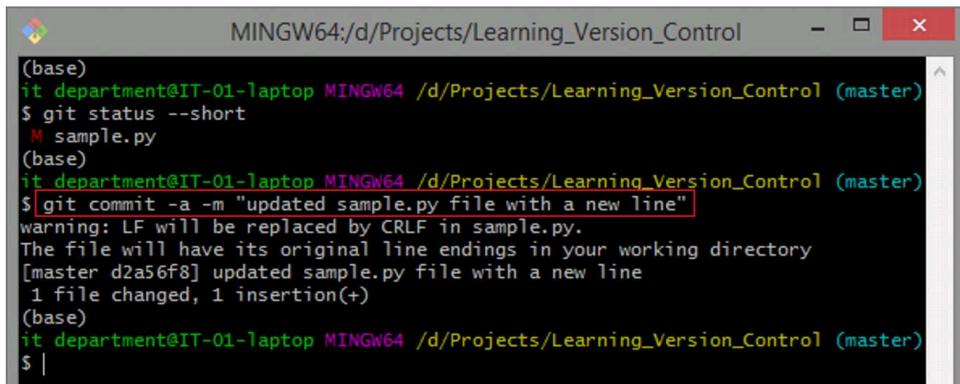
```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status --short
 M sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Note: Short status flags are:

- o ?? - Untracked files
- o A - Files added to stage
- o M - Modified files
- o D - Deleted files

We see the file we expected is modified. So let's commit it directly without '`git add .`' command:

Command: `git commit -a -m "updated sample.py file with a new line"`



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git status --short
 M sample.py
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git commit -a -m "updated sample.py file with a new line"
warning: LF will be replaced by CRLF in sample.py.
The file will have its original line endings in your working directory
[master d2a56f8] updated sample.py file with a new line
 1 file changed, 1 insertion(+)
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

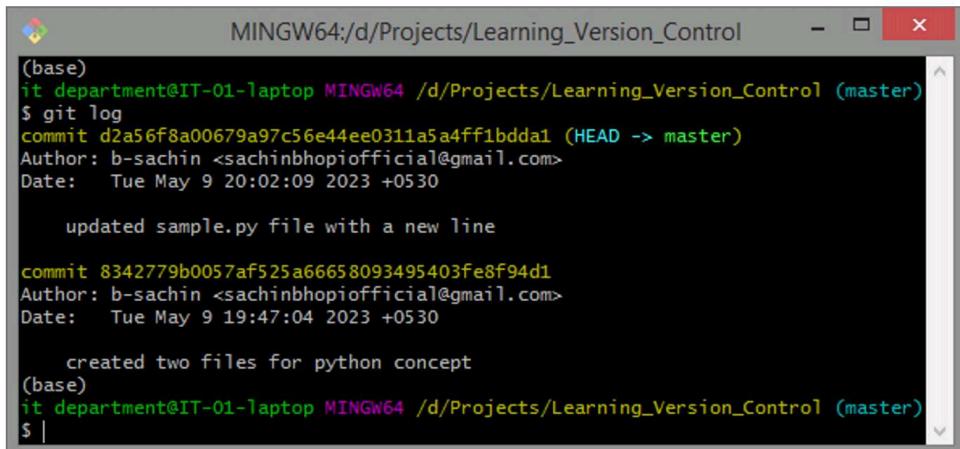
Warning: Skipping the Staging Environment is not generally recommended.

Skipping the stage step can sometimes make you include unwanted changes.

Git Commit Log

To view the history of commits for a repository, you can use the `log` command:

Command: `git log`



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git log
commit d2a56f8a00679a97c56e44ee0311a5a4ff1bdda1 (HEAD -> master)
Author: b-sachin <sachinbhopiofficial@gmail.com>
Date:   Tue May 9 20:02:09 2023 +0530

    updated sample.py file with a new line

commit 8342779b0057af525a66658093495403fe8f94d1
Author: b-sachin <sachinbhopiofficial@gmail.com>
Date:   Tue May 9 19:47:04 2023 +0530

    created two files for python concept
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Git Help

If you are having trouble remembering commands or options for commands, you can use Git `help`.

There are a couple of different ways you can use the `help` command in command line:

- `git command -help` - See all the available options for the specific command
- `git help --all` - See all possible commands

Any time you need some help remembering the specific option for a command, you can use `git command -help`:

Example Command: `git commit -help`



The screenshot shows a terminal window titled "MINGW64:/d/Projects/Learning_Version_Control". The command `git commit -help` is run, displaying a detailed help message for the `git commit` command. The message is organized into sections: Commit message options, Commit contents options, and a note about pathspecs. It lists various flags and their descriptions, such as `-q` for quiet mode and `--all` for committing all changes.

```
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git commit -help
usage: git commit [<options>] [--] <pathspec>...

-q, --quiet      suppress summary after successful commit
-v, --verbose     show diff in commit message template

Commit message options
  -F, --file <file>    read message from file
  --author <author>   override author for commit
  --date <date>        override date for commit
  -m, --message <message>
                      commit message
  -c, --reedit-message <commit>
                      reuse and edit message from specified commit
  -C, --reuse-message <commit>
                      reuse message from specified commit
  --fixup [(amend|reword):]<commit>
                      use autosquash formatted message to fixup or amend/reword specified commit
  --squash <commit>   use autosquash formatted message to squash specified commit
  --reset-author
                      the commit is authored by me now (used with -C/-c/--amend)
  --end)
  --trailer <trailer> add custom trailer(s)
  -s, --signoff      add a Signed-off-by trailer
  -t, --template <file>
                      use specified template file
  -e, --edit         force edit of commit
  --cleanup <mode>   how to strip spaces and #comments from message
  --status           include status in commit message template
  -S, --gpg-sign[=<key-id>]
                      GPG sign commit

Commit contents options
  -a, --all          commit all changed files
  -i, --include      add specified files to index for commit
  --interactive     interactively add files
  -p, --patch        interactively add changes
  -o, --only         commit only specified files
  -n, --no-verify    bypass pre-commit and commit-msg hooks
  --dry-run          show what would be committed
  --short            show status concisely
  --branch           show branch information
  --ahead-behind    compute full ahead/behind values
  --porcelain       machine-readable output
  --long             show status in long format (default)
  -z, --null         terminate entries with NUL
  --amend            amend previous commit
  --no-post-rewrite bypass post-rewrite hook
  -u, --untracked-files[=<mode>]
                      show untracked files, optional modes: all, normal, no.
(Default: all)
  --pathspec-from-file <file>
                      read pathspec from file
  --pathspec-file-nul
                      with --pathspec-from-file, pathspec elements are separated with NUL character

(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Git help --all See All Possible Commands

To list all possible commands, use the `help --all` command:

The screenshot shows a terminal window titled "MINGW64:/d/Projects/Learning_Version_Control". The command entered is "git help --all". The output lists various Git commands with their descriptions. Some commands like "branch", "checkout", and "diff" have their descriptions truncated by ellipses (...).

```
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git help --all
See 'git help <command>' to read about a specific subcommand

Main Porcelain Commands
  add          Add file contents to the index
  am           Apply a series of patches from a mailbox
  archive      Create an archive of files from a named tree
  bisect       Use binary search to find the commit that introduced a b
  bug
    branch     List, create, or delete branches
    bundle     Move objects and refs by archive
    checkout   Switch branches or restore working tree files
    cherry-pick Apply the changes introduced by some existing commits
    citool     Graphical alternative to git-commit
    clean      Remove untracked files from the working tree
    clone      Clone a repository into a new directory
    commit     Record changes to the repository
    describe   Give an object a human readable name based on an availab
  le ref
    diff       Show changes between commits, commit and working tree, e
  tc
    fetch     Download objects and refs from another repository
    format-patch Prepare patches for e-mail submission
    gc         Cleanup unnecessary files and optimize the local reposi
....skipping...
See 'git help <command>' to read about a specific subcommand
```

Note: If you find yourself stuck in the list view, **SHIFT + G** to jump to the end of the list, then **q** to exit the view.

Working with Git Branches

In Git, a **branch** is a new/separate version of the main repository.

Let's say you have a large project, and you need to update the design on it. How would that work without and with Git:

Without Git:

- Make copies of all the relevant files to avoid impacting the live version
- Start working with the design and find that code depend on code in other files, that also need to be changed!
- Make copies of the dependant files as well. Making sure that every file dependency references the correct file name
- EMERGENCY! There is an unrelated error somewhere else in the project that needs to be fixed ASAP!
- Save all your files, making a note of the names of the copies you were working on
- Work on the unrelated error and update the code to fix it
- Go back to the design, and finish the work there
- Copy the code or rename the files, so the updated design is on the live version
- (2 weeks later, you realize that the unrelated error was not fixed in the new design version because you copied the files before the fix)

With Git:

- With a new branch called new-design, edit the code directly without impacting the main branch
- EMERGENCY! There is an unrelated error somewhere else in the project that needs to be fixed ASAP!
- Create a new branch from the main project called small-error-fix
- Fix the unrelated error and merge the small-error-fix branch with the main branch
- You go back to the new-design branch, and finish the work there
- Merge the new-design branch with main (getting alerted to the small error fix that you were missing)

Branches allow you to work on different parts of a project without impacting the main branch.

When the work is complete, a branch can be merged with the main project.

You can even switch between branches and work on different projects without them interfering with each other.

Branching in Git is very lightweight and fast!

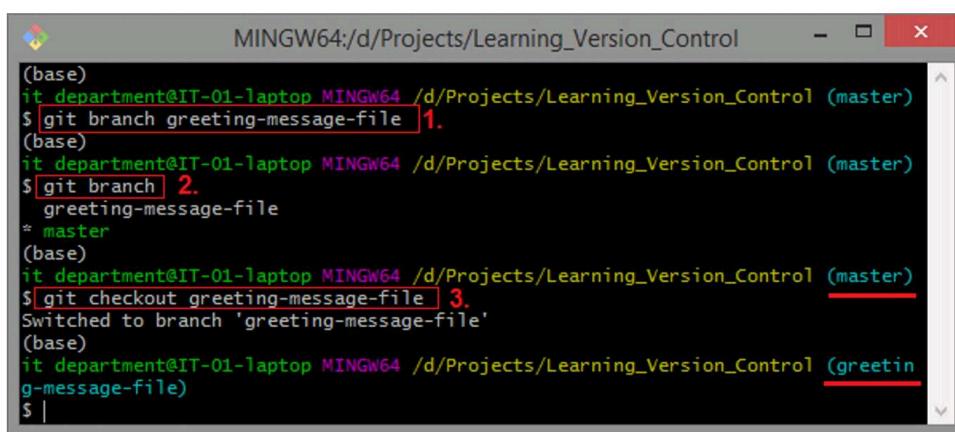
New Git Branch

Let add some new features to our `sample.py` page.

We are working in our local repository, and we do not want to disturb or possibly wreck the main project.

So we create a new **branch**:

Commands: `git branch greeting-message-file`
`git branch`
`git checkout greeting-message-file`



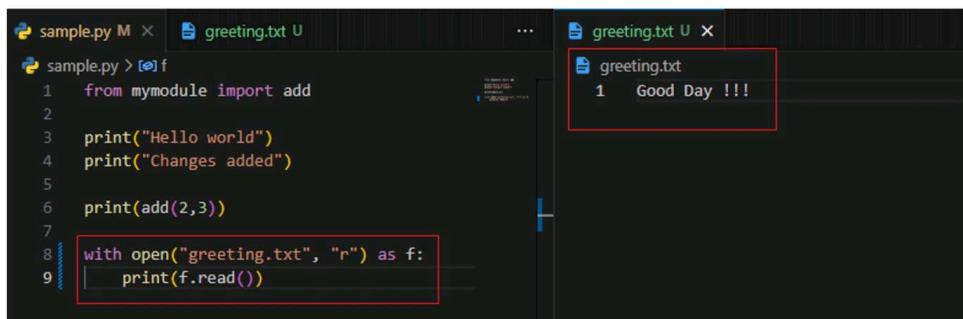
```

MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git branch greeting-message-file 1.
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git branch 2.
  greeting-message-file
* master
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout greeting-message-file 3.
Switched to branch 'greeting-message-file'
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greetin
g-message-file)
$ |

```

- git branch greeting-message-file** : for creating a new branch called 'greeting-message-file'
- git branch** : To see all available branches and current active branch is pointed using asterisk (*)
- git checkout greeting-message-file** : To move from current active branch i.e. 'master' to new branch i.e. 'greeting-message-file'

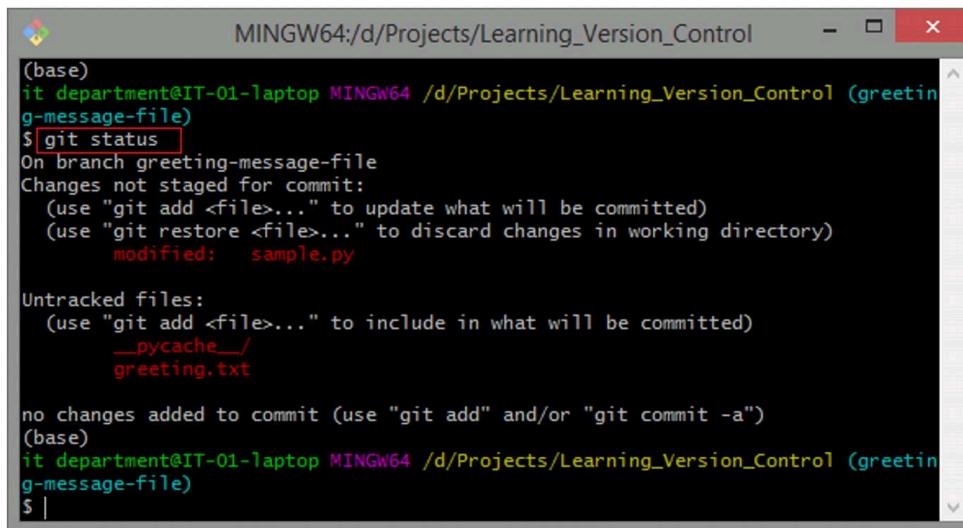
Let add new file `greeting.txt` which has some text like 'Good Day !!!'
and
Open `greeting.txt` file in `sample.py` file using python code and print 'Good Day !!!' message.



```
sample.py M x greeting.txt U ...
sample.py > [o] f
1 from mymodule import add
2
3 print("Hello world")
4 print("Changes added")
5
6 print(add(2,3))
7
8 with open("greeting.txt", "r") as f:
9     print(f.read())
...
greeting.txt U x
greeting.txt
1 Good Day !!!
```

We have made changes to a file and added a new file in the working directory (same directory as the **main branch**).

Now check the status of the current **branch**:



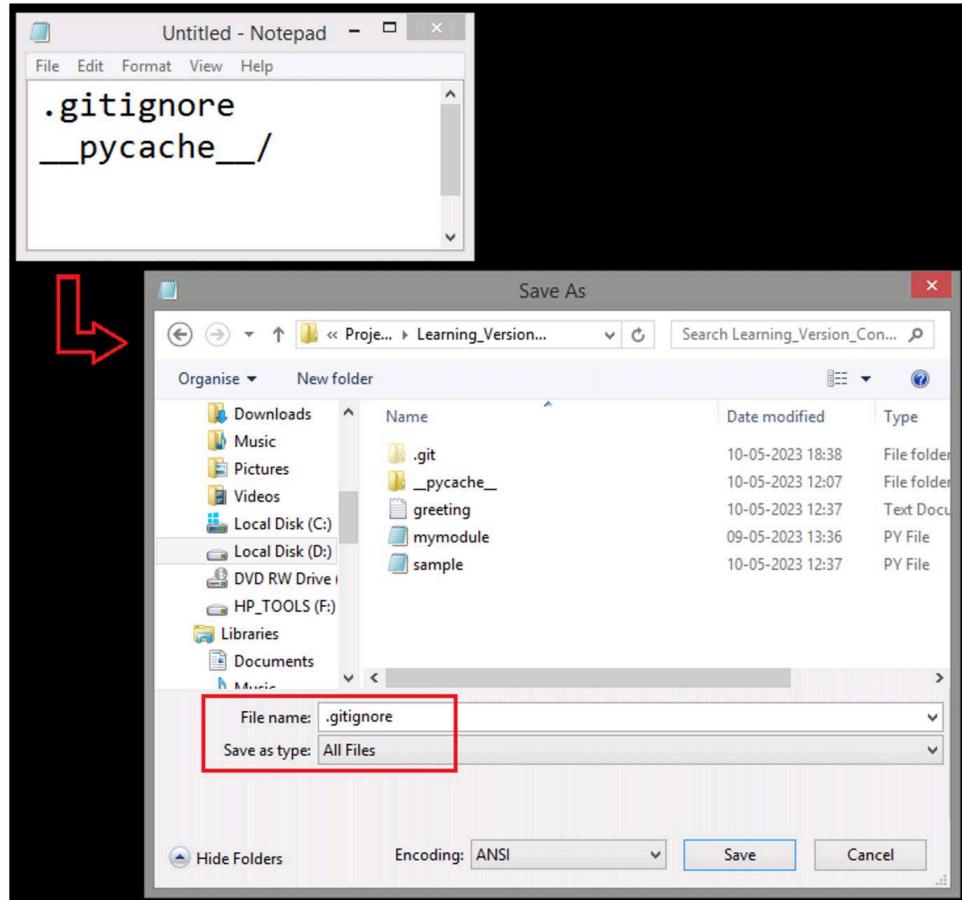
```
MINGW64:/d/Projects/Learning_Version_Control
(base) it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git status
On branch greeting-message-file
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   sample.py

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

no changes added to commit (use "git add" and/or "git commit -a")
(base) it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```

So let's go through what happens here:

- There are changes to our `sample.py`, but the file is not staged for **commit**
- `greeting.txt` is not **tracked**
- `__pycache__/` is a folder created by VS Code which we need to ignore thus we will add it to `.gitignore` file. [`.gitignore` is a separate concept we will learn later in detail.]
 - For now, Using notepad create `.gitignore` file and add `__pycache__/` and `.gitignore` in it.



Now again we can check the status of the current **branch**:

```
MINGW64:/d/Projects/Learning_Version_Control - x
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git status
On branch greeting-message-file
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:   sample.py

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

no changes added to commit (use "git add" and/or "git commit -a")
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```

So we need to add both files i.e. `sample.py` and `greeting.txt` to the Staging Environment for this **branch**:

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git add --all
warning: LF will be replaced by CRLF in sample.py.
The file will have its original line endings in your working directory
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git status
On branch greeting-message-file
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   greeting.txt
    modified:   sample.py

(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```

We are happy with our changes. So we will commit them to the **branch**:

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git commit -m "added greeting message to sample file"
[greeting-message-file 76834e5] added greeting message to sample file
 2 files changed, 3 insertions(+)
  create mode 100644 greeting.txt
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```

Now we have a new **branch**, that is different from the master **branch**.

Switching Between Branches

Now let's see just how quick and easy it is to work with different branches, and how well it works.

We are currently on the branch **greeting-message-file**. We added an text file to this branch

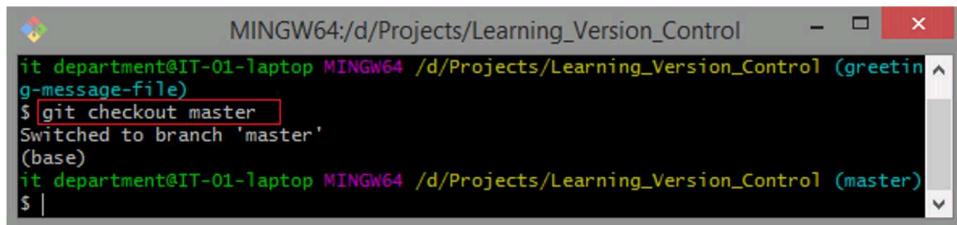
So let's list the files in the current directory:

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ ls
__pycache__/ greeting.txt mymodule.py sample.py
```

We can see the new file **greeting.txt**, and if we open the sample file, we can see the code has been altered.

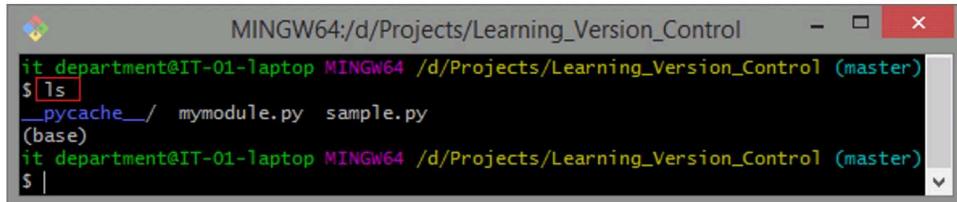
All is as it should be.

Now, let's see what happens when we change branch to **master**



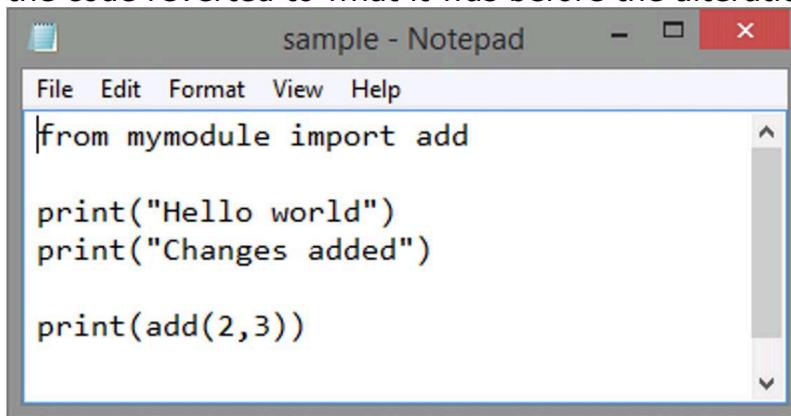
```
MINGW64:/d/Projects/Learning_Version_Control
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git checkout master
Switched to branch 'master'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

The new image is not a part of this branch. List the files in the current directory again:



```
MINGW64:/d/Projects/Learning_Version_Control
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ ls
__pycache__/ mymodule.py sample.py
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Greeting.txt is no longer there! And if we open the **sample.py** file, we can see the code reverted to what it was before the alteration.

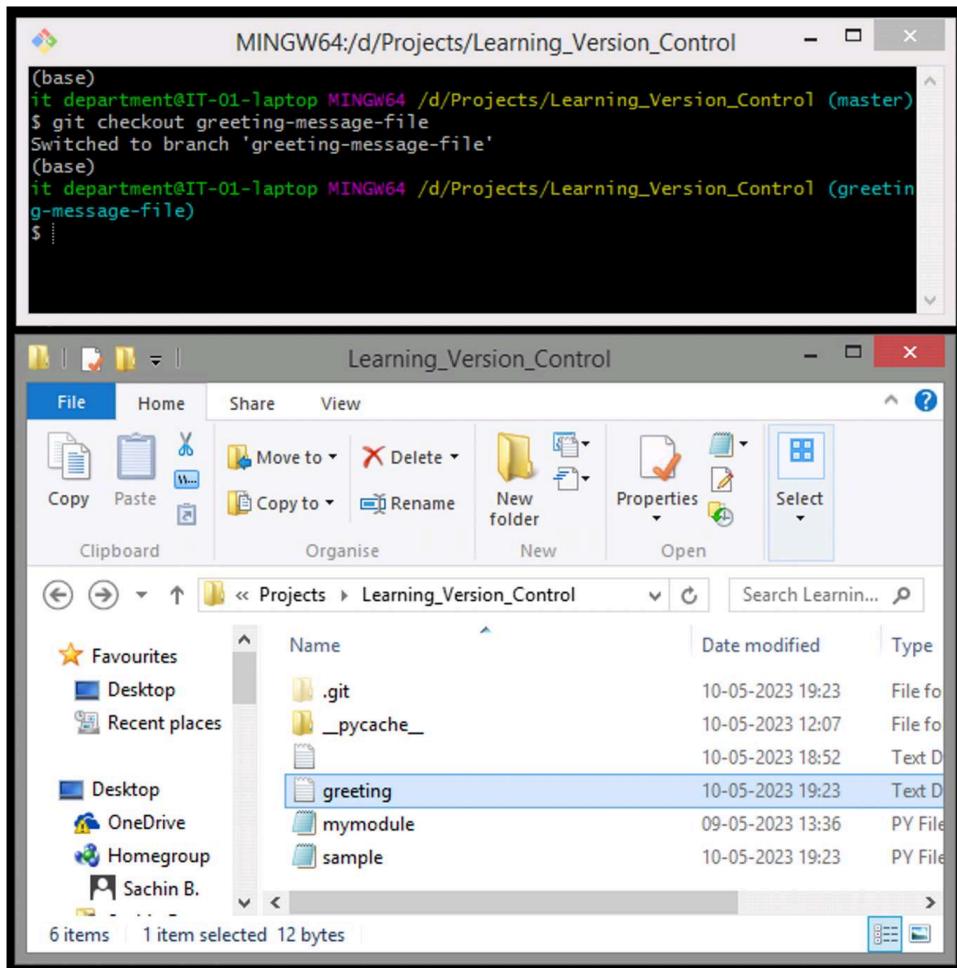


```
sample - Notepad
File Edit Format View Help
from mymodule import add

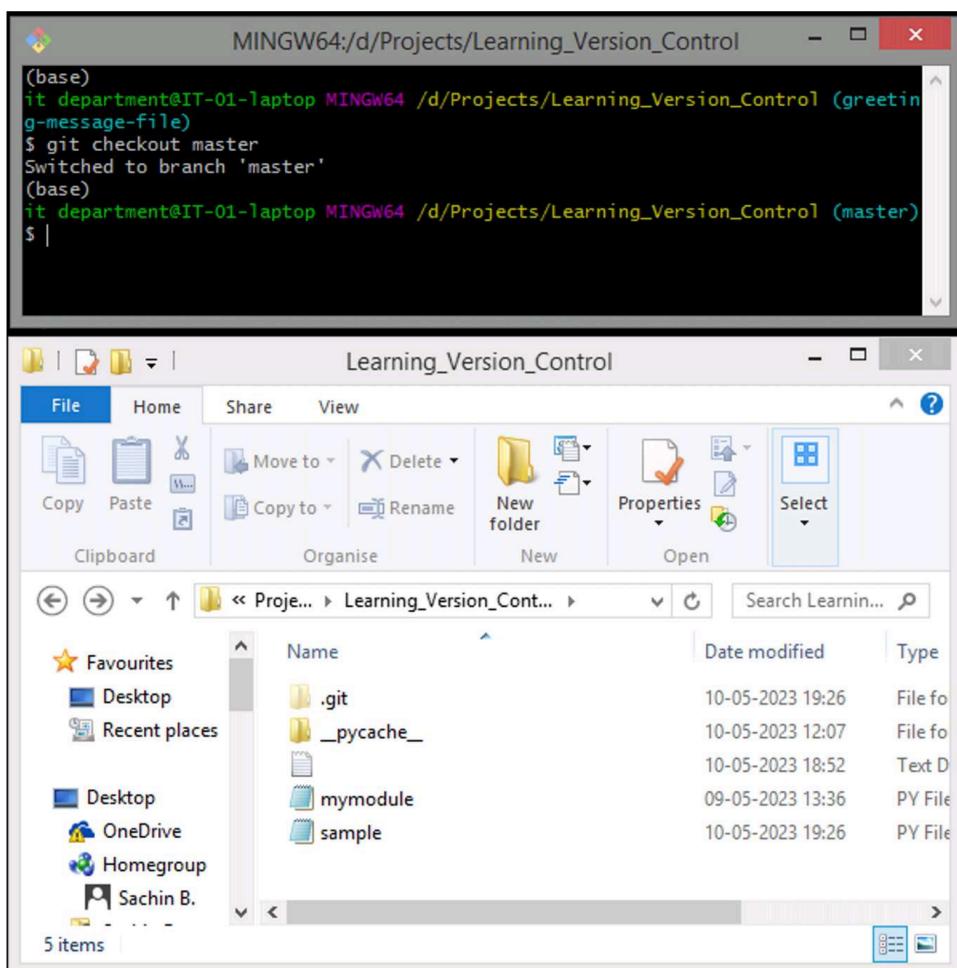
print("Hello world")
print("Changes added")

print(add(2,3))
```

See how easy it is to work with branches and how this allows you to work on different things



```
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout greeting-message-file
Switched to branch 'greeting-message-file'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$
```



```
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git checkout master
Switched to branch 'master'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$
```

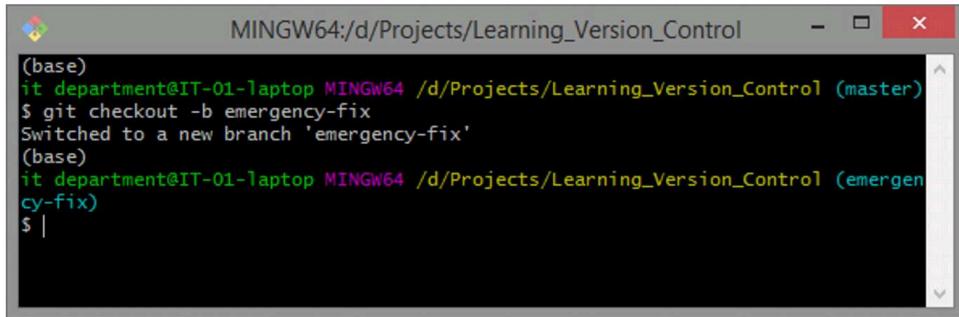
Emergency Branch

Now imagine that we are not yet done with `greeting-message-file`, but we need to fix an error on `master`.

I don't want to mess with `master` directly, and I do not want to mess with `greeting-message-file`, since it is not done yet.

So we create a new branch to deal with the emergency:

Command: `git checkout -b emergency-fix`

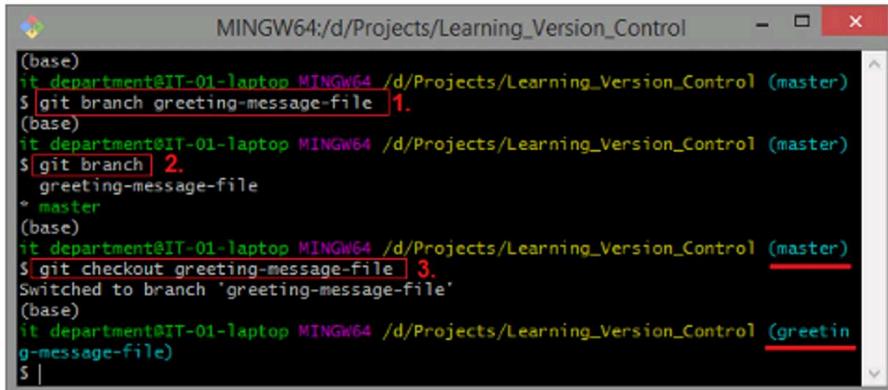


```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout -b emergency-fix
Switched to a new branch 'emergency-fix'
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ |
```

Now we have created a new branch from master, and changed to it. We can safely fix the error without disturbing the other branches.

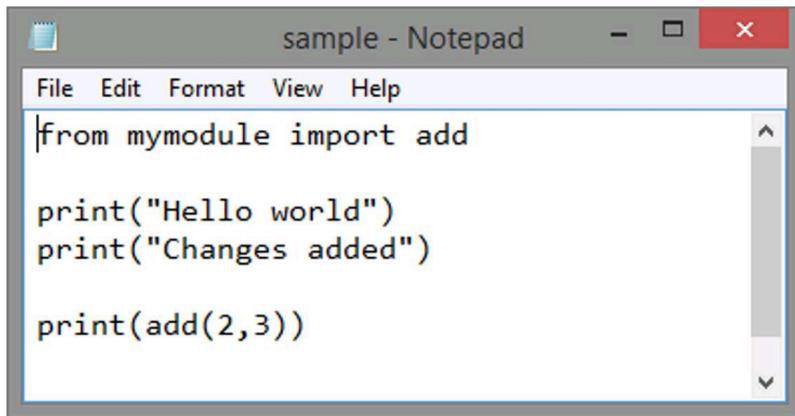
Following is the normal way that we have learned earlier:

Commands: `git branch greeting-message-file`
`git branch`
`git checkout greeting-message-file`



```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git branch greeting-message-file 1.
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git branch 2.
  greeting-message-file
* master
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout greeting-message-file 3.
Switched to branch 'greeting-message-file'
(base)
it_department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```

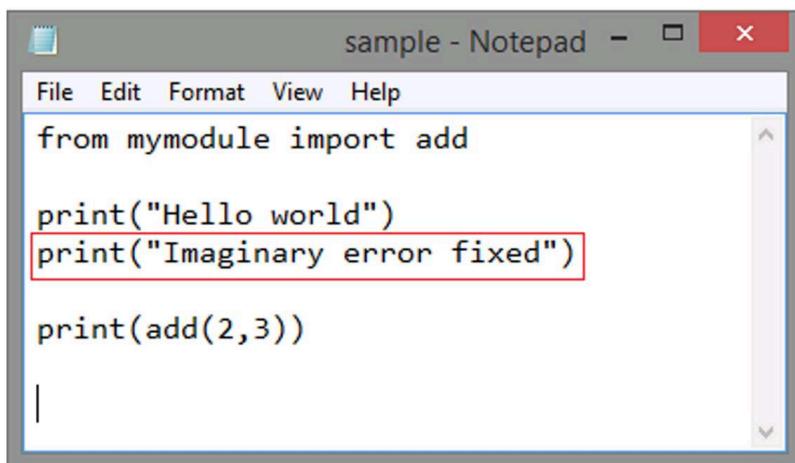
Let's fix our imaginary error:



```
sample - Notepad
File Edit Format View Help
from mymodule import add

print("Hello world")
print("Changes added")

print(add(2,3))
```

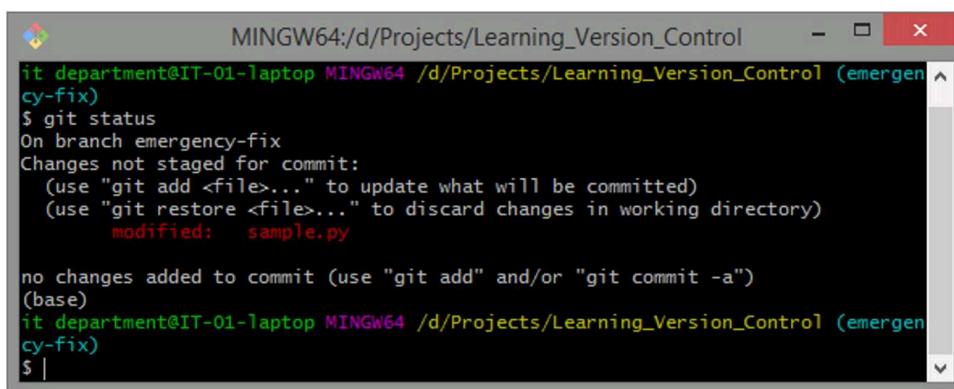


```
sample - Notepad
File Edit Format View Help
from mymodule import add

print("Hello world")
print("Imaginary error fixed")  
|  
print(add(2,3))
```

We have made changes in this file, and we need to get those changes to the master branch.

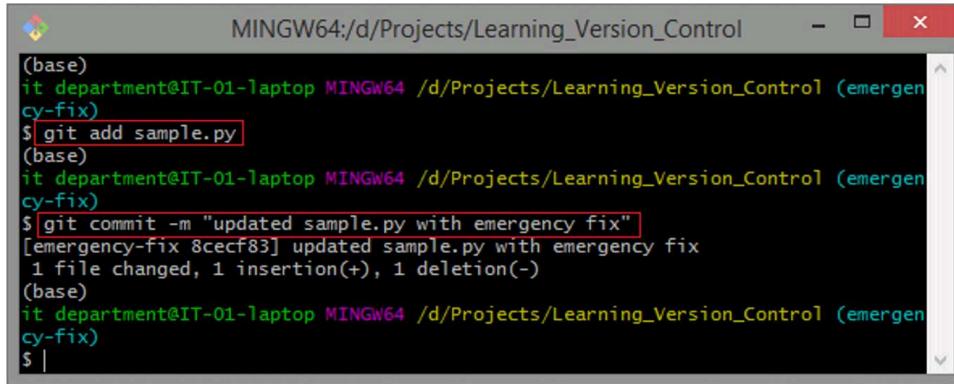
Check the status:



```
MINGW64:/d/Projects/Learning_Version_Control
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ git status
On branch emergency-fix
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:   sample.py

no changes added to commit (use "git add" and/or "git commit -a")
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ |
```

stage the file, and commit:



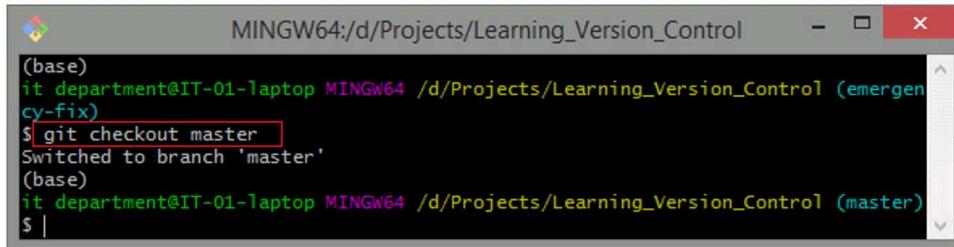
```
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ git add sample.py
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ git commit -m "updated sample.py with emergency fix"
[emergency-fix 8cecf83] updated sample.py with emergency fix
 1 file changed, 1 insertion(+), 1 deletion(-)
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ |
```

Now we have a fix ready for master, and we need to merge the two branches.

Merge Branches

We have the emergency fix ready, and so let's merge the master and emergency-fix branches.

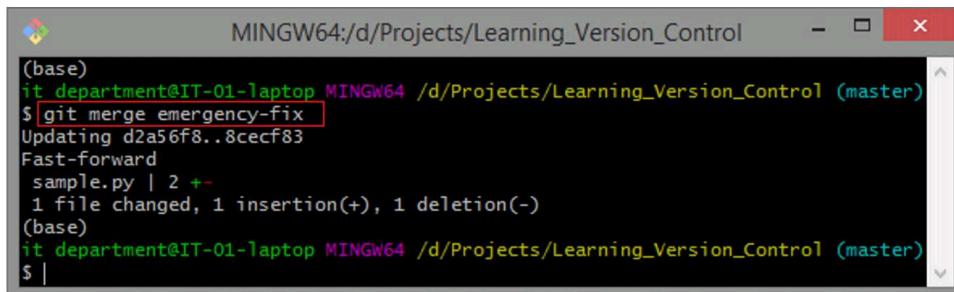
First, we need to change to the master branch:



```
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (emergency-fix)
$ git checkout master
Switched to branch 'master'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Now we merge the current branch (master) with emergency-fix:

Command: `git merge emergency-fix`



```
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git merge emergency-fix
Updating d2a56f8..8cecf83
Fast-forward
  sample.py | 2 ++
  1 file changed, 1 insertion(+), 1 deletion(-)
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Since the emergency-fix branch came directly from master, and no other changes had been made to master while we were working, Git sees this as a continuation of master.

So it can "Fast-forward", just pointing both master and emergency-fix to the same commit.

As master and emergency-fix are essentially the same now, we can **delete emergency-fix**, as it is no longer needed:

Command: `git branch -d emergency-fix`

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git branch -d emergency-fix
Deleted branch emergency-fix (was 8cecf83).
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ |
```

Learning:

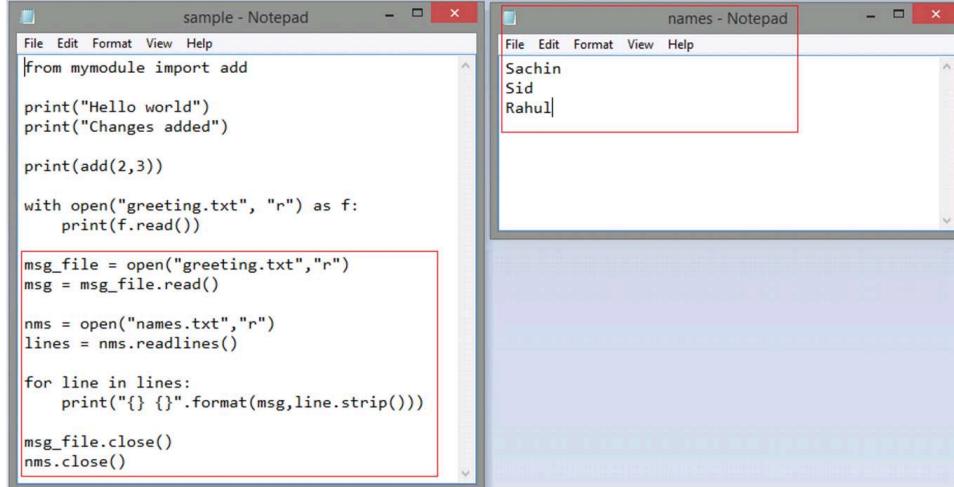
Command to create branch: `git branch greeting-message-file`

Command to delete branch: `git branch -d greeting-message-file`

Merge Conflict

Now we can move over to **greeting-message-file** and keep working. Add another text file (names.txt) and change sample.py, so it shows it:

```
MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout greeting-message-file
Switched to branch 'greeting-message-file'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |
```



The image shows two Notepad windows side-by-side. The left window, titled 'sample - Notepad', contains Python code. The right window, titled 'names - Notepad', contains a list of names.

sample - Notepad

```
File Edit Format View Help
from mymodule import add
print("Hello world")
print("Changes added")
print(add(2,3))
with open("greeting.txt", "r") as f:
    print(f.read())
msg_file = open("greeting.txt","r")
msg = msg_file.read()
nms = open("names.txt","r")
lines = nms.readlines()
for line in lines:
    print("{} {}".format(msg,line.strip()))
msg_file.close()
nms.close()
```

names - Notepad

```
File Edit Format View Help
Sachin
Sid
Rahul|
```

Now, we are done with our work here and can stage and commit for this branch:

```

MINGW64:/d/Projects/Learning_Version_Control
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git status
On branch greeting-message-file
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:   sample.py

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

no changes added to commit (use "git add" and/or "git commit -a")
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git add .
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ git commit -m "added names file"
[greeting-message-file de10030] added names file
 2 files changed, 16 insertions(+), 1 deletion(-)
 create mode 100644 names.txt
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (greeting-message-file)
$ |

```

We see that sample.py has been changed in both branches.

In 'greeting-message-file' branch

```

sample - Notepad
File Edit Format View Help
|from mymodule import add
|
print("Hello world")
print("Changes added")
|
print(add(2,3))
|
with open("greeting.txt", "r") as f:
    print(f.read())
|
msg_file = open("greeting.txt","r")
msg = msg_file.read()
|
nms = open("names.txt","r")
lines = nms.readlines()
|
for line in lines:
    print("{} {}".format(msg,line.strip()))
|
msg_file.close()
nms.close()

```

In 'master' branch

```

sample - Notepad
File Edit Format View Help
|from mymodule import add
|
print("Hello world")
print("Imaginary error fixed")
|
print(add(2,3))

```

Now we are ready to merge hello-world-images into master. But what will happen to the changes we recently made in master?

Command: `git checkout master`
`git merge greeting-message-file`

Note: It will open editor and will ask:

Please enter a commit message to explain why this merge is necessary, especially if it merges an updated upstream into a topic branch

Write: Commit message:

merge conflict completed

```
MINGW64:/d/Version_Control
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git checkout master
Switched to branch 'master'
(base)
it department@IT-01-laptop MINGW64 /d/Projects/Learning_Version_Control (master)
$ git merge greeting-message-file
Auto-merging sample.py
Merge made by the 'recursive' strategy.
greeting.txt |  1 +
names.txt     |  3 ++
sample.py     | 17 ++++++-----+
3 files changed, 20 insertions(+), 1 deletion(-)
create mode 100644 greeting.txt
create mode 100644 names.txt
```

```
MINGW64:/d/Version_Control
$ git log
commit 3fe8d15660d86e93c5483907f86a4f4d397344b0 (HEAD -> master)
Merge: 2366b3c 42a87e3
Author: b-sachin <sachinbhopiofficial@gmail.com>
Date:   Thu May 11 19:45:17 2023 +0530

    Merge branch 'greeting-message-file'
    merge conflict completed

commit 42a87e38ac2919b19b7ba7e87b97fbea90ff1dec (greeting-message-file)
Author: b-sachin <sachinbhopiofficial@gmail.com>
Date:   Thu May 11 19:43:17 2023 +0530

    added names file
```