7. Git Tags and Releases

Write the command to create a lightweight Git tag named “v1.0” for a commit in your local repository.

- In Git, every time we make a commit, it’s like saving a new version of our project.
- But sometimes, we want to **mark an important version** — like ‘Version 1.0’ or ‘Final Submission’ so that later we can easily come back to it.
- That’s what **tags** are for — they are **labels** we attach to specific commits."
- Tags are used to mark **important commits or releases** (v1.0, v2.0, release-candidate, etc.)

<u>Execution Steps</u>

Step 1: Create a new project folder and initialize git

```
mkdir tag-demo
 cd tag-demo
 git init
```

Step 2: Create a file and make your first commit

```
touch  file1.txt   add some contents to the file : example - This is  my project version
git add .
git commit –m “initial commit version 1”
```

Step 3: Create a lightweight tag named v1.0 for the current commit

```
git tag v1.0
```

This creates a lightweight tag — just a simple label for that commit.

To Verify the tags:

- git tag
- git show v1.0

Step 4: Make new changes to the file

```
echo "added features" >> file1.txt
git add file1.txt
git commit -m "updated version 1"
```

Repeat another commit:

```
echo "bugfix" >> file1.txt
git add file1.txt
git commit -m " new  version 2"
```

Repeat another commit:

```
echo "finalized features added" >> file1.txt
git add file1.txt
git commit -m " final updated version 3"
```

These are normal commits. We didn’t tag them because they are incremental changes.

To verify : `git log  --oneline`

Step 5 : When ready to mark the next release, tag again (v2.0)

- After you’ve made multiple commits and reached the next milestone:
- make any final commit for v2.0

```
echo "final changes for v2" >> file1.txt
git add file1.txt
git commit -m "Prepare for version 2.0"
```

Create another tag named v2.0

```
git tag v2.0
```

Verify tags: `git tag`

Now we have two tags : v1.0 and v2.0.”

(Optional)

Step 6: Push your project and tags to GitHub

- Create an empty repository in GitHub (name it **tag-demo**)
- Copy the repo URL and run:

```
git remote add origin https://github.com/your-username/tag-demo.git
git branch -M main
git push -u origin main
```

Then push your tags:

```
git push origin - -tags
```

Verify on GitHub :

Go to your repo → Click on **"Tags"**
You'll see both `v1.0` and `v2.0`.

- Every commit is like a new version
- A tag is a name or label for a specific commit
- Developers use tags to mark major versions — like v1.0, v2.0, etc.
- When we push tags to GitHub, they appear as **Releases**, which are often used when deploying real software.