**Experiment No. 1**

**Aim: Installation of version control tool Git on Ubuntu 18.04/Windows/Mac**

**Lab Outcome No.: 9.ITL8003.2**

**Lab Outcome: Illustrate different version control strategies using Git**

**Date of Performance: 29/1/21 Date of Submission: 5/2/21**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Program formation/ Execution / Ethical practices (07 )** | **Documentation (02)** | **Timely Submission (03)** | **Viva Answer (03)** | **Experiment Marks (15)** | **Teacher Signature with date** |
|  |  |  |  |  |  |

**Experiment No 1**

**Aim :** Installation of version control tool Git on Ubuntu 18.04

**Lab Outcome No.:** ITL8003.2 -

**Lab Outcome:** Illustrate different Version Control strategies using Git.

**Theory:**

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

* **Frictionless Context Switching**. Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
* **Role-Based Codeines**. Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
* **Feature Based Workflow**. Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
* **Disposable Experimentation**. Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work—with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Some of the basic operations in Git are:

1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:

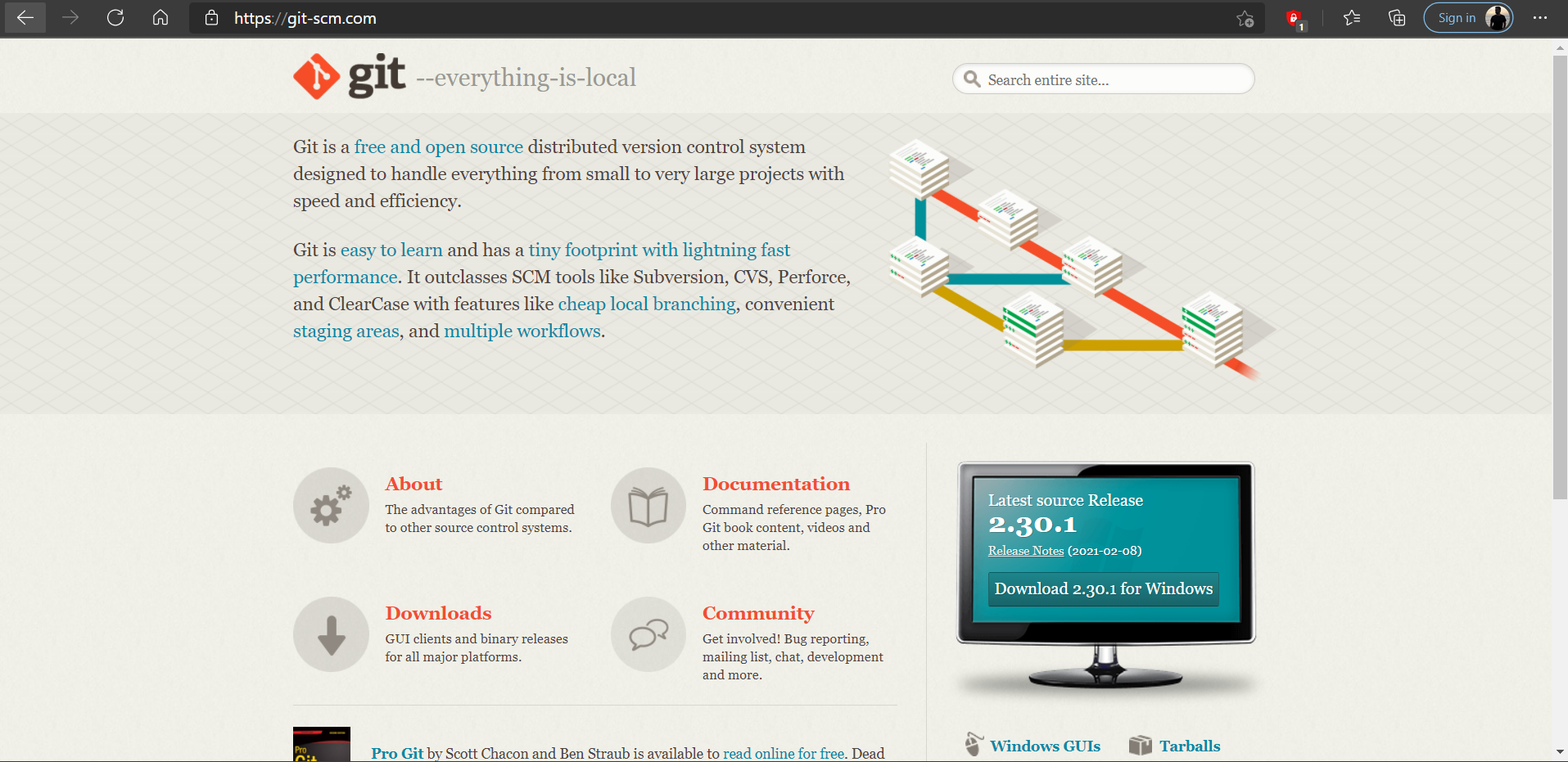
1. Branching
2. Merging
3. Rebasing

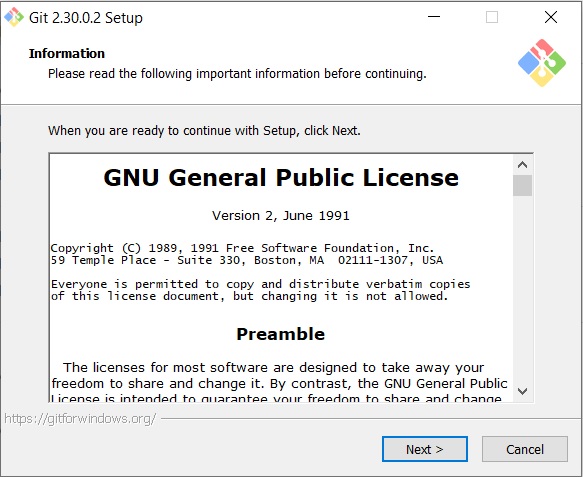
The following diagram depicts all supported operations in GIT:

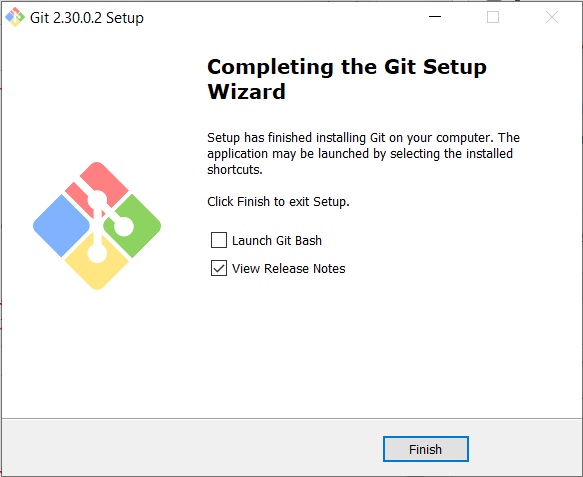


# Output :

1. In windows, download GIT from https://git-scm.com/ and perform the straightforward installation.







1. Confirm the version after installation using the command “git version”



# Conclusion :

Git radically changes the way how your team will create and deliver work to you. Various processes including designing, development, product management, marketing, customer support can be easily handled and maintained using Git in your organization. Git tool has been successfully installed and is ready to be used.