|  |  |  |
| --- | --- | --- |
| **No** | **Version Control Systems** | **Brief intro** |
| 1 | [**Aegis**](http://aegis.sourceforge.net/) | Aegis supports multiple developers on a project, and includes an intranet web interface. It is designed for repository security, and supports distributed and multiple repositories. It runs on “almost any flavor of Unix.” Source code is provided. |
| 2 | [**Bazaar-NG**](http://bazaar-vcs.org/) | Bazaar-NG is a decentralized revision control system that allows users to commit their own branches of the source code for particular software without requiring special permissions. |
| 3 | [**/BriefCase 3 Toolkit**](http://www.applied-cs-inc.com/bcintro.html) | Like other revision control systems, /BriefCase allows you to manage multiple versions of your software. It runs on Unix and Linux systems with the Korn Shell andawk. It uses a client/server model, witha RCS-based central repository. |
| 4 | [**CvsGui, WinCVS, MacCVS, gCVS – GUI Front-end for CVS**](http://www.wincvs.org/) | WinCVS is afrontend for the CVS package (see elsewhere on this page) for Windows, while MacCVS runs on the Macintosh andgCVS on Unix systems. It has a native system look-and-feel, realtime sandbox view with a visual indication of the local state of files, filters to monitor any folder in a flat view, file revision history graphs, support for text, binary and Unicode files, etc. It is distributed under LGPL. |
| 5 | [**CVS (Concurrent Version System)**](http://www.nongnu.org/cvs/) | CVS provides version control on entire directory trees. It supports client/server operations where developers may be scattered all over the globe. CVS is supported on most Unices and clones, Windows NT/95, OS/2, and VMS. It is probably one of the most widely used free version control systems around and has the advantage that it is not only free, but is open source. |

Git-is a [version control system](https://en.wikipedia.org/wiki/Version_control_system) that is widely used for [software development](https://en.wikipedia.org/wiki/Software_development) and other version control tasks. It is a [distributed revision control](https://en.wikipedia.org/wiki/Distributed_revision_control) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was created by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) in 2005 for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with other kernel developers contributing to its initial development.

As with most other distributed version control systems, and unlike most [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) systems, every Git [working directory](https://en.wikipedia.org/wiki/Working_directory) is a full-fledged[repository](https://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is [free software](https://en.wikipedia.org/wiki/Free_software) distributed under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) version 2.

Github- is a web-based [Git](https://en.wikipedia.org/wiki/Git_(software)" \o "Git (software)) repository [hosting service](https://en.wikipedia.org/wiki/Hosting_service). It offers all of the [distributed revision control](https://en.wikipedia.org/wiki/Distributed_revision_control) and [source code management](https://en.wikipedia.org/wiki/Source_code_management) (SCM) functionality of [Git](https://en.wikipedia.org/wiki/Git_(software)" \o "Git (software)) as well as adding its own features. Unlike Git, which is strictly a [command-line](https://en.wikipedia.org/wiki/Command-line) tool, GitHub provides a [Web-based graphical interface](https://en.wikipedia.org/wiki/Web_application) and desktop as well as mobile integration. It also provides [access control](https://en.wikipedia.org/wiki/Access_control) and several collaboration features such as [bug tracking](https://en.wikipedia.org/wiki/Bug_tracking_system),[feature requests](https://en.wikipedia.org/wiki/Software_feature), [task management](https://en.wikipedia.org/wiki/Task_management), and [wikis](https://en.wikipedia.org/wiki/Wiki) for every project.

gitHub offers both plans for private [repositories](https://en.wikipedia.org/wiki/Repository_(revision_control)) and free accounts, which are usually used to host [open-source](https://en.wikipedia.org/wiki/Open-source) software projects. As of April 2016, GitHub reports having more than 14 million users and more than 35 million repositories, making it the largest host of source code in the world.

What are the difference between Git and Github

Git = Local (on you computer), GitHub = Remote (web).

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 while

Github- is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

Github provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

Things that you can do in github

1. Share your repositories with others.
2. Access other user's repositories.
3. Store remote copies of your repositories (github servers) as backup of your local

Terms

**Repository**- is a somewhat general term used to refer to a destination designated for data storage. However, many IT experts use the term more specifically to refer to a particular kind of setup within an overall IT structure, such as a group of databases, where an enterprise or organization has chosen to keep various kinds of data.   
  
Some experts refer to a data repository as a partitioning of data, where partitioned data types are stored together. It is also commonly called data warehousing.

**Commit**-a change simply means recording the change in the central system. Other programmers can then see this change. They can also pull down the change, and the version control tool will automatically update the contents of any files that were changed

**Fork-**   happens when developers take a copy of [source code](https://en.wikipedia.org/wiki/Source_code) from one [software package](https://en.wikipedia.org/wiki/Computer_software) and start independent development on it, creating a distinct and separate piece of software. The term often implies not merely a [development branch](https://en.wikipedia.org/wiki/Branching_(revision_control)), but a split in the developer community, a form of [schism](https://en.wikipedia.org/wiki/Schism_(religion)).

**Pull-** The act of getting new changes from a repository is usually called.

**Push**-The act of moving your own changes to a repository .