**Assignment 1**

**Attempt all questions, each question carries 5 marks, there is no negative marking.  
Upload your answer file with your name in repository “/Assignment1” location.**

Q.1) How user can push files on GitHub?

Ans. The git push command is used to transfer or push the commit, which is made on a local branch in your computer to a remote repository like GitHub. The command used for pushing to GitHub is given below:

git push ‘remote\_name’ ‘branch\_name’

**Using Command Line to PUSH to GitHub**

**Step 1:** Creating a new repository.

**Step 2:** Open your Git Bash.

**Step 3:** Create your local project in your desktop directed towards a current working directory.

**Step 4:** Initialize the git repository.

**Step 5:** Add the file to the new local repository.

**Step 6:** Commit the files staged in your local repository by writing a commit message.

**Step 7:** Copy your remote repository’s URL from GitHub.

**Step 8:** Add the URL copied, which is your remote repository to where your local content from your repository is pushed.

**Step 9:** Push the code in your local repository to GitHub.

**Step 10:** View your files in your repository hosted on GitHub.

**Using GitHub Desktop to PUSH to your local content to GitHub**

**Step 1:** Click “Set up in a Desktop”.

**Step 2:** Cloning in a GitHub Desktop.

**Step 3:** Copy all the required files from your local computer into the clone folder on your computer.

**Step 4:** Move to GitHub Desktop and commit to master.

**Step 5:** Publish branch in GitHub Desktop to upload your all files to GitHub.

Q.2) Which command are used to check GIT Version?

Ans. You can check your current version of Git by running the git --version command in a terminal (Linux, Mac OS X) or command prompt (Windows). The versions of Git supported by Crucible.

git --version

C:\Users\Yogesh Maindargi\Desktop\Django Internship\Django-Internship\_3March2021>git --version

git version 2.30.1.windows.1

If you don't see a supported version of Git, you'll need to either upgrade Git or perform a fresh install.

Q.3) How you can collaborate with others in github?

### Ans.

### TWO COMMON COLLABORATIVE WORK FLOWS

**SHARED REPOSITORY MODEL**

* For small projects where you are basically in the same physical space (e.g., lab with offices near each other).
* Be careful! You are cloning the main repository.
* Everyone has push and pull access to the central repo, so be careful and:
  + Never commit to the master directly.
  + Always do your work on a different branch from master.

**BASIC SHARED REPOSITORY WORKFLOW**

* update your local repo with git pull origin master,
* create a working branch with git checkout -b MyNewBranch
* make your changes on your branch and stage them with git add,
* commit your changes locally with git commit -m "description of your commit", and
* upload the changes (including your new branch) to GitHub with git push origin MyNewBranch
* Go to the main repo on GitHub where you should now see your new branch
* click on your branch name
* click on “Pull Request” button (URC)
* click on “Send Pull Request”

**FORK AND PULL MODEL**

* This is the model used by U of T Coders on its own website and repos.
* The “owner”/”Project Leader” of the upstream repo assigns rights to “Collaborators”
* Collaborators do not have push access to main (upstream) repo
* Project Lead accepts Pull Requests (PRs) fro collaborators, reviews them, then merges them into main repo.

### TWO PERSON COLLABORATION VIA THE CLI - SHARED REPO WORKFLOW (WITHOUT BRANCHES)

One of you will be the “Owner” and one of you will be the “Collaborator.”

#### A. OWNER GIVES COLLABORATOR ACCESS TO THEIR REPO.

1. Go to your GitHub repo
2. Add a file called “tenlines.txt” and put the text from the etherpad into the file. Commit your changes.
3. Click on **Settings** tab.
4. Click **Collaborators**
5. Enter collaborataors username

#### B. COLLABORATOR CLONES OWNER’S REPO

1. Go to https://github.com/notifications and accept access to Owner’s repo.
2. On the CLI, clone the owner’s repo but issuing the commmand:  
   $ git clone URL-of-Origin-Repo Directory-Address-of-Local-Repo

#### C. COLLABORATOR WORKS ON CLONE OF OWNER’S REPO

Go to your cloned repo:  
$ cd ~/.../yourClone

Open editor and revise working file:  
atom tenlines.txt

Commit your changes to your local repo:  
$ git add tenlines.txt  
$ git commit -m "your commit message"

Push your changes to the Owner’s repo on GitHub:  
$ git push origin master

#### D. OWNER REVIEW AND ACCEPTS CHANGES FROM COLLABORATOR

Look at Owner’s GitHub repo and see new commit(s) from Collaborator.

Download (pull) Collaborators changes to Owner’s local repo:  
$ git pull origin master

Q.4) How you can push updates on your repository?

Ans.

While pushing updates on the repository use following commands:

**Step 1:** Go to the location of working directory where we have to add or update the files in the local repository on your system. Then use following command.

git add .

**Step 2:** then, make a local commit by.

git commit -m "your commit message"

**Step 3:** once you make your local commit, you can then push it to your remote GitHub fork.

git push origin master

But if you want to pull your changes from the forked repository in github

To make your source code pulled by original fork, you have to send a pull request to the project owner.

1. Go to the web page of your forked project on GitHub.
2. Hit the pull request button on the top right of page
3. Select the commits that you want to submit by change commits button.
4. Write some description of your changes, comments, or etc.
5. Send pull request and wait for the owner reply.

Q.5) What is alternative of github & what is Full form of GIT?

Ans.

## **Best GitHub alternatives**

The focus here is on GitHub alternatives that have at least some sort of free service. Because that was the main attraction of GitHub. There are several Git repository hosting services but not all of them provide a free option in their package.

### Migrate GitHub to GitLabGitLab

[GitLab](https://about.gitlab.com/) is the number one choice to replace GitHub. It is the closest to GitHub in terms of use and feel. Best of all, GitLab is an open-source software. You can download and install it on your own server.

### BitBucket

### 

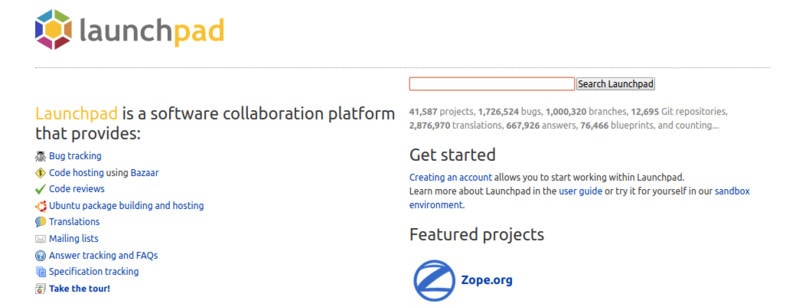
[BitBucket](https://bitbucket.org/product) is a version control repository hosting service from Atlassian. It is tightly integrated with other Atlassian project management tools like Jira, HipChat, and Confluence. This makes it a preferred choice for big enterprises.

**SourceForge**



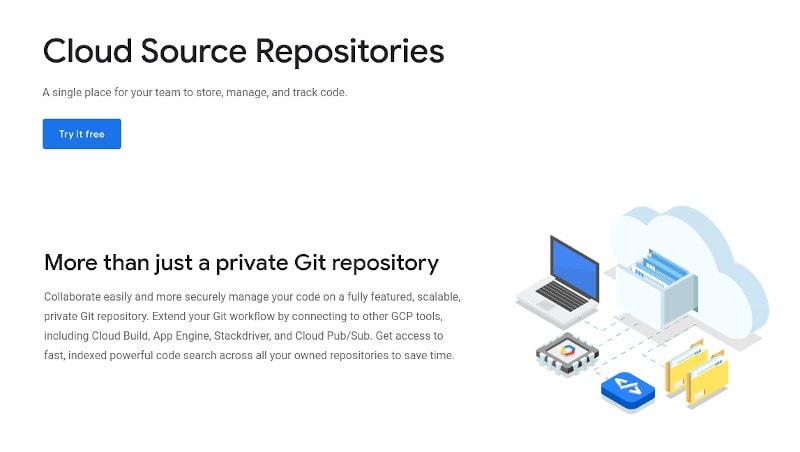
[SourceForge](https://sourceforge.net/) is another big name on this list of GitHub alternatives. SourceForge has been popular among open-source projects. Many Linux distributions and projects provide their downloads through SourceForge. It enables developers to create open-source projects by providing all the necessary tools.

**Launchpad**



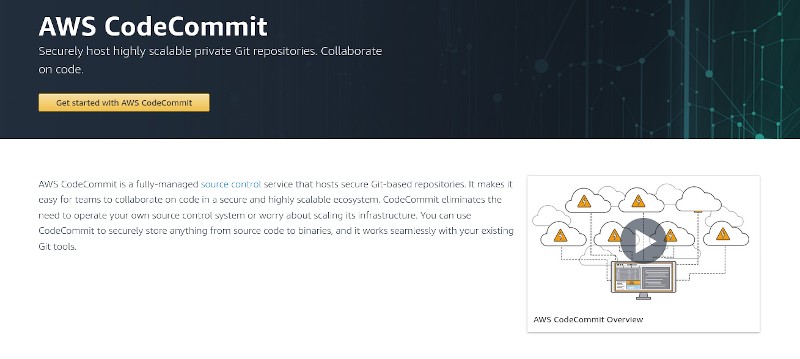
[Launchpad](https://launchpad.net/) is a software collaboration platform from [Canonical](https://www.canonical.com/), the parent company of Ubuntu. Launchpad has been extensively used by Canonical and projects around Ubuntu. It has been instrumental in providing the PPA and bug tracking for Ubuntu related projects.

**Google Cloud Source Repositories**

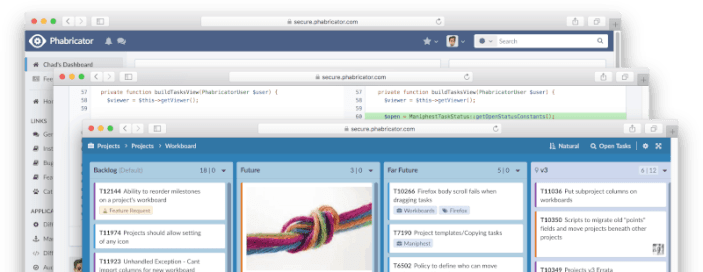


[Google Cloud Source Repositories](https://cloud.google.com/source-repositories) can be a good alternative for private repositories. You can get started for free with a limit of 5 users and 50 GB storage. To start with, you get a 12-months trial period.

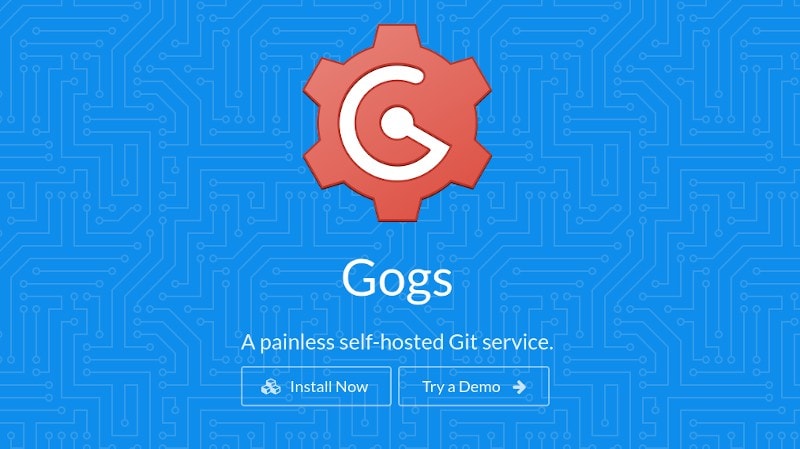
**AWS CodeCommit**



[AWS CodeCommit](https://aws.amazon.com/codecommit/) is a similar alternative to Google Cloud Source Repositories. Just like the Google Cloud Platform, AWS also provides a free tier that does not end when the trial ends. So, it’s free forever if your usage is within the free tier limits as mentioned in their [official documentation](https://aws.amazon.com/free/).

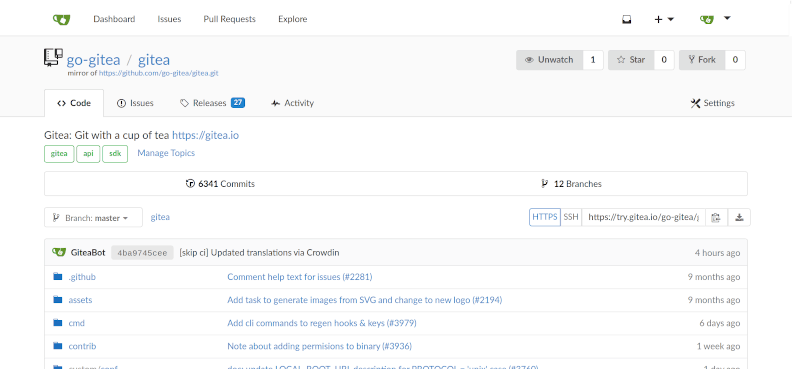
**Phabricator**

[Phabricator by Phacility](https://www.phacility.com/phabricator/) is an all-in-one tool that lets you host code and discuss/plan to keep working on a project without needing to utilize separate applications for communication/collaboration.

**Gogs (Self-Hosted)**

Unlike some of the ones mentioned above, [Gogs](https://gogs.io/) is a completely self-hosted solution to host your code. Also, it is a very lightweight option that can also run on a Raspberry Pi. Of course, you can also utilize a $5/month cloud hosting solution like Linode or Digital Ocean.

**Gitea (Self-Hosted)**



Yet another self-hosting solution to host your code is a community fork of Gogs (which I’ve mentioned above). Similar to the above, it is a lightweight option and can even run-on Raspberry Pi.

**Apache Allura (Self-Hosted)**

Apache Allura was originally built by SourceForge as an open-source implementation that utilized [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [MongoDB](https://en.wikipedia.org/wiki/MongoDB) while offering REST APIs. It is indeed a decent GitHub alternative.

**Full form of GIT (Version Control System)**

Git is software for tracking changes in any set of [files](https://en.wikipedia.org/wiki/Computer_file), usually used for coordinating work among [programmers](https://en.wikipedia.org/wiki/Programmer) collaboratively developing [source code](https://en.wikipedia.org/wiki/Source_code) during [software development](https://en.wikipedia.org/wiki/Software_development). Its goals include speed, [data integrity](https://en.wikipedia.org/wiki/Data_integrity), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems). 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.

"git" can mean anything, depending on your mood.

* random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
* stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
* "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.