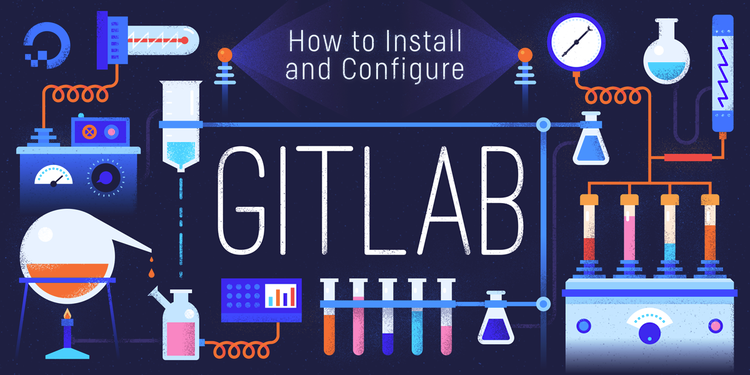
**Git Lab**

1. GitLab is a web-based [Git](https://en.wikipedia.org/wiki/Git_(software)) [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) manager with [wiki](https://en.wikipedia.org/wiki/Wiki) and [issue tracking](https://en.wikipedia.org/wiki/Issue_tracking_system) features, using an [open source](https://en.wikipedia.org/wiki/Open-source_software) license, developed by GitLab Inc.
   1. Wiki: A wiki is a [website](https://en.wikipedia.org/wiki/Website) on which users [collaboratively](https://en.wikipedia.org/wiki/Collaborative_software) modify content and structure directly from the [web browser](https://en.wikipedia.org/wiki/Web_browser). In a typical wiki, text is written using a simplified [markup language](https://en.wikipedia.org/wiki/Markup_language) and often edited with the help of a [rich-text editor](https://en.wikipedia.org/wiki/Online_rich-text_editor).[[1]](https://en.wikipedia.org/wiki/Wiki" \l "cite_note-Britannica-1)
   2. Issue Tracking: An issue tracking system (also ITS, trouble ticket system, support ticket, request management or incident ticket system) is a [computer software](https://en.wikipedia.org/wiki/Computer_software) package that manages and maintains lists of [issues](https://en.wikipedia.org/wiki/Issue_(computers)),[[1]](https://en.wikipedia.org/wiki/Issue_tracking_system" \l "cite_note-1) as needed by an organization.
2. The software was written by Dmitriy Zaporozhets and Valery Sizov from [Ukraine](https://en.wikipedia.org/wiki/Ukraine).
3. The code is written in [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)).
4. Later, some parts have been rewritten in [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).
5. It is used by organisations such as [IBM](https://en.wikipedia.org/wiki/IBM), [Sony](https://en.wikipedia.org/wiki/Sony), [Jülich Research Center](https://en.wikipedia.org/wiki/Jülich_Research_Center), [NASA](https://en.wikipedia.org/wiki/NASA), [Alibaba](https://en.wikipedia.org/wiki/Alibaba_Group), [Invincea](https://en.wikipedia.org/wiki/Invincea), [O’Reilly Media](https://en.wikipedia.org/wiki/O’Reilly_Media), [Leibniz-Rechenzentrum (LRZ)](https://en.wikipedia.org/wiki/Leibniz-Rechenzentrum), [CERN](https://en.wikipedia.org/wiki/CERN)[[7]](https://en.wikipedia.org/wiki/GitLab" \l "cite_note-7)[[8]](https://en.wikipedia.org/wiki/GitLab" \l "cite_note-8)[[9]](https://en.wikipedia.org/wiki/GitLab" \l "cite_note-9), and [SpaceX](https://en.wikipedia.org/wiki/SpaceX).[[10]](https://en.wikipedia.org/wiki/GitLab" \l "cite_note-10)
6. Originally, the product was named GitLab and was fully [free and open source software](https://en.wikipedia.org/wiki/Free_and_open_source_software) distributed under the [MIT License](https://en.wikipedia.org/wiki/MIT_License).
   1. MIT License: The **MIT License** is a [permissive free software license](https://en.wikipedia.org/wiki/Permissive_free_software_license) originating at the [Massachusetts Institute of Technology](https://en.wikipedia.org/wiki/Massachusetts_Institute_of_Technology) (MIT).
7. n July 2013,[[12]](https://en.wikipedia.org/wiki/GitLab" \l "cite_note-12) the product was split:
   1. GitLab CE: Community Edition
   2. GitLab EE: Enterprise Edition

**Installation Steps**

****

**1. Pre-requistics**

1. 2 cores
2. 4GB of RAM

**2.** **Installing the Dependencies**

1. sudo apt-get update
2. sudo apt-get install ca-certificates curl openssh-server postfix

**3.** **Install GitLab**

1. cd /tmp
2. curl -LO https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh
3. less /tmp/script.deb.sh
4. sudo bash /tmp/script.deb.sh
5. sudo apt-get install gitlab-ce
6. sudo gitlab-ctl reconfigure
7. sudo gitlab-ctl restart

**4.** **Adjusting the Firewall Rules**

1. **sudo ufw status &** If it is inactive **sudo ufw enable**
2. **sudo ufw allow http**
3. **sudo ufw allow OpenSSH**
4. **sudo ufw status**

**5. Create a GitLab account**

1. [**https://gitlab.com**](https://gitlab.com/)
2. **Add a ssk key**

### **Command line instructions**

##### **Git global setup**

git config --global user.name "Votarytech"

git config --global user.email "vitadmtest@gmail.com"

##### **Create a new repository**

git clone https://gitlab.com/Votarytech/DEMO\_Project.git

cd DEMO\_Project

touch README.md

git add README.md

git commit -m "add README"

git push -u origin master

##### **Existing folder**

cd existing\_folder

git init

git remote add origin https://gitlab.com/Votarytech/DEMO\_Project.git

git add .

git commit -m "Initial commit"

git push -u origin master

##### **Existing Git repository**

cd existing\_repo

git remote rename origin old-origin

git remote add origin https://gitlab.com/Votarytech/DEMO\_Project.git

git push -u origin --all

git push -u origin --tags

**if you are pushing without pull the changes from remote server repository:**

