**Installing GITLAB on Ubuntu 18.04**

The [published GitLab hardware requirements](http://docs.gitlab.com/ce/install/requirements.html#hardware-requirements) recommend using a server with:

* 2 cores
* 8GB of RAM
* **Step 1 — Installing the Dependencies**

sudo apt update

sudo apt install ca-certificates curl openssh-server postfix

## Step 2 — Installing GitLab

cd /tmp

curl -LO <https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh>

Feel free to examine the downloaded script to ensure that you are comfortable with the actions it will take. You can also find a hosted version of the script [here](https://packages.gitlab.com/gitlab/gitlab-ce/install):

* less /tmp/script.deb.sh

Once you are satisfied with the safety of the script, run the installer:

* sudo bash /tmp/script.deb.sh

The script will set up your server to use the GitLab maintained repositories. This lets you manage GitLab with the same package management tools you use for your other system packages. Once this is complete, you can install the actual GitLab application with apt:

* sudo apt install gitlab-ce

## Step 3 — Adjusting the Firewall Rules

## sudo ufw enable

## sudo ufw status

The protocol to port mapping for HTTP and HTTPS are available in the /etc/services file, so we can allow that traffic in by name. If you didn’t already have OpenSSH traffic enabled, you should allow that traffic now too:

sudo ufw allow http

sudo ufw allow https

sudo ufw allow OpenSSH

Check the ufw status again; you should see access configured to at least these two services:

* sudo ufw status

## Step 4 — Editing the GitLab Configuration File

Before you can use the application, you need to update the configuration file and run a reconfiguration command. First, open Gitlab’s configuration file:

* sudo nano /etc/gitlab/gitlab.rb
* Near the top is the external\_url configuration line. Update it to match your domain. Change http to https so that GitLab will automatically redirect users to the site protected by the Let’s Encrypt certificate:
* /etc/gitlab/gitlab.rb
* ##! For more details on configuring external\_url see:
* ##! https://docs.gitlab.com/omnibus/settings/configuration.html#configuring-the-external-url-for-gitlab
* external\_url 'https://example.com'

Next, look for the letsencrypt['contact\_emails'] setting. This setting defines a list of email addresses that the Let’s Encrypt project can use to contact you if there are problems with your domain. It’s a good idea to uncomment and fill this out so that you will know of any issues:

/etc/gitlab/gitlab.rb

letsencrypt['contact\_emails'] = ['sammy@example.com']

Save and close the file. Run the following command to reconfigure Gitlab:

* sudo gitlab-ctl reconfigure

## Step 5 — Performing Initial Configuration Through the Web Interface

With GitLab running and access permitted, we can perform some initial configuration of the application through the web interface.

### Logging In for the First Time

Visit the domain name of your GitLab server in your web browser:

<https://example.com> Or With ur Instance public ip or instance DNS