Install Gitlab with SSL/TLS Certificate on Ubuntu 20.04

By

[**gen\_too**](https://kifarunix.com/author/gen_too/)

-

November 20, 2020

[0](https://kifarunix.com/install-gitlab-with-ssl-tls-certificate-on-ubuntu-20-04/#respond)

1387

In this tutorial, you will learn how to install Gitlab with SSL/TLS certificate on Ubuntu 20.04. [GitLab](https://gitlab.com/gitlab-org/gitlab) *is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD,* etc. *It is is a complete DevOps platform, delivered as a single application*.

Read more about what Gitlab offers on [Gitlab Features page](https://about.gitlab.com/features/).

Install Gitlab with SSL/TLS Certificate on Ubuntu 20.04

Gitlab Installation Requirements on Ubuntu 20.04

Below are the bare minimums to install Gitlab with SSL/TLS certificates on Ubuntu 20.04.

Hardware Requirements

Ensure the system you want to host with Gitlab has met the following minimum hardware requirements;

* **Storage**: Enough storage depending on the size of the repositories you want to store in GitLab.
* **CPU**: At least 4 cores. (supports up to 500 users).
* **Memory**: At least 4GB RAM (supports up to 500 users).
* **Swap**: At least 2GB of swap memory.

Software Requirements

* GitLab requires Ruby (MRI) 2.6.
* The minimum required Go version is 1.13.
* Node.js 10.13.0 or higher is required. Node 12.x is recommended.
* GitLab 13.0 and later requires Redis version 4.0 or higher.

Database Requirements

Gitlab supports PostgreSQL database only. With the following being table outlining required version of PostgreSQL database for a specific version of Gitlab;

| **GitLab version** | **Minimum PostgreSQL version** |
| --- | --- |
| 10.0 | 9.6 |
| 13.0 | 11 |
| 13.6 | 12 |

Read more about the requirements of installing [Gitlab on Requirements page](https://docs.gitlab.com/ee/install/requirements.html).

Install Gitlab with SSL/TLS Certificate on Ubuntu 20.04

Run System Update

Run system update;

apt update

Install Required package dependencies

Run the command below to install some required package dependencies.

apt install curl tzdata ca-certificates openssh-server

[Install Gitlab on Ubuntu 20.04](https://kifarunix.com/install-gitlab-with-ssl-tls-certificate-on-ubuntu-20-04/#install-gitlab-on-ubuntu-20.04)

In this tutorial, we are installing Gitlab Community Edition, CE. Therefore, you can install it using the ready DEB binary package or directly from Gitlab repositories;

To install Gitlab CE on Ubuntu 20.04, you need to install the Gitlab CE repo by running the command below;

curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash

Once the Gitlab package repo is done, install Gitlab on Ubuntu 20.04;

apt install gitlab-ce

To install Gitlab using DEB binary package, download the binary from the [Gitlab packages page](https://packages.gitlab.com/gitlab/gitlab-ce) and install it manually using apt or dpkg commands;

wget -O gitlab-ce\_13.5.4-ce.0\_arm64.deb https://packages.gitlab.com/gitlab/gitlab-ce/packages/ubuntu/focal/gitlab-ce\_13.5.4-ce.0\_arm64.deb/download.deb

apt install ./gitlab-ce\_13.5.4-ce.0\_arm64.deb

Configure Gitlab with SSL/TLS Certificate on Ubuntu 20.04

Once the Gitlab CE package is installed, you can install SSL/TLS certificates. There are two ways in which you can configure Gitlab with SSL/TLS certificates;

1. Using free and automated HTTPS with Let’s Encrypt
2. Manually configuring HTTPS with your own certificates

In this demo, we are going to manually setup Gitlab with SSL certs using our own self signed certificates. If you are going to expose your gitlab server to public internet, you can consider using the commercial TLS certs.

Create a directory to store the SSL certs;

mkdir /etc/gitlab/ssl

Next, generate the self signed SSL certs by running the command below. Be sure to replace the certificates details accordingly in the command below;

openssl req -newkey rsa:4096 -x509 -sha512 -days 3650 -nodes -out /etc/gitlab/ssl/kifarunix-demo.crt -keyout /etc/gitlab/ssl/kifarunix-demo.key -subj "/C=US/ST=California/L=San Francisco/O=Kifarunix-demo Ltd/CN=\*.kifarunix-demo.com/"

Copy the public certificate file only into the **/etc/gitlab/trusted-certs** directory.

mkdir /etc/gitlab/trusted-certs

cp /etc/gitlab/ssl/kifarunix-demo.crt /etc/gitlab/trusted-certs/

Want to use Let’s Encrypt instead? Check this [link](https://docs.gitlab.com/omnibus/settings/ssl.html#lets-encrypt-integration).

Configure a URL for GitLab Server on Ubuntu 20.04

Once the certificates are generated, edit the **/etc/gitlab/gitlab.rb** configuration file and replace the value of the **external\_url** parameter to your Gitlab CE server URL to enable HTTPS for the domain. Replace the domain name appropriately.

vim /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

##!

##! Note: During installation/upgrades, the value of the environment variable

##! EXTERNAL\_URL will be used to populate/replace this value.

##! On AWS EC2 instances, we also attempt to fetch the public hostname/IP

##! address from AWS. For more details, see:

##! https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html

#external\_url 'http://gitlab.example.com'

external\_url 'https://gitlab.kifarunix-demo.com'

#### Enable Gitlab SSL Settings

Next, scroll down to Gitlab NGINX section and make the following adjustments (as per your setup).

################################################################################

## GitLab NGINX

##! Docs: https://docs.gitlab.com/omnibus/settings/nginx.html

################################################################################

nginx['enable'] = true

nginx['client\_max\_body\_size'] = '250m'

nginx['redirect\_http\_to\_https'] = true

...

...

nginx['ssl\_certificate'] = "/etc/gitlab/ssl/kifarunix-demo.crt"

nginx['ssl\_certificate\_key'] = "/etc/gitlab/ssl/kifarunix-demo.key"

...

...

nginx['ssl\_protocols'] = "TLSv1.2 TLSv1.3"