How to setup our own git server using gitlab

Introduction

If you're a Git user, you know that having local repositories that can be accessed via a local LAN (or external WAN) is a crucial element of the development process. You can certainly opt to go with *GitHub*, but that negates the ability to host locally. So when you want to host your own repositories, where do you turn?

In a word, *GitLab.*

GitLab allows you to host an on-premise Git repository that can be accessed from either your local LAN or (if you have an available public IP address) from outside your company. GitLab is fairly easy to install and incredibly simple to use. I'm going to walk you through the process of installing GitLab on *Ubuntu Server 16.04.* I will assume you already have Ubuntu Server up and running, and have access to an account with sudo rights.

Installation

In order to install GitLab, you will need a server with at least two cores and 4GB of RAM. If you're running this on a virtual machine, make sure the VM exceeds those resources.

The first thing you must do is run an update or upgrade. Please know that--should the kernel upgrade--a reboot will be necessary. If this is a production machine, make sure to run this process during off hours.

Open up a terminal window on the server and issue the commands:

```
sudo apt update
sudo apt upgrade
```

Installation

Once those commands run, you are ready to install GitLab (assuming you don't have to reboot the server).

The first thing that must be installed is the necessary dependencies. This can be handled with the following command:

```
sudo apt-get install ca-certificates curl openssh-server postfix
```

During the above installation, you will be asked how to configure Postfix. Select Internet site, and then enter either the domain or the IP address of the server. If you're users are familiar with Linux, you could always select a local-only Postfix configuration, knowing that all users would have to use the mail command on the server to check to see if they have any mail delivered by GitLab. If you do go that route, you'll need to install the mailutils package, like so:

```
sudo apt install mailutils
```

Installation

With the dependencies ready, we must install the necessary repository with the following commands:

```
curl -LO https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh
cd /tmp
sudo bash /tmp/script.deb.sh
```

Finally, install GitLab with the command:

```
sudo apt-get install gitlab-ce
```

Configuring GitLab

There is only one file to edit for configuration. Open the file for editing with the command sudo nano /etc/gitlab/gitlab.rb. Within that file, you're looking for the following line:

```
external_url 'https://yourdomain'
```

Modify that to reflect either your server domain or IP address. If you're using an IP address, make sure to drop the https in favor of http.

If you configure GitLab to use a domain, you'll have to enable SSL. To do that, locate the following two lines (around line 1519):

```
# letsencrypt['enable'] = false
# letsencrypt['contact_emails'] = [ ]
```

Configuring GitLab

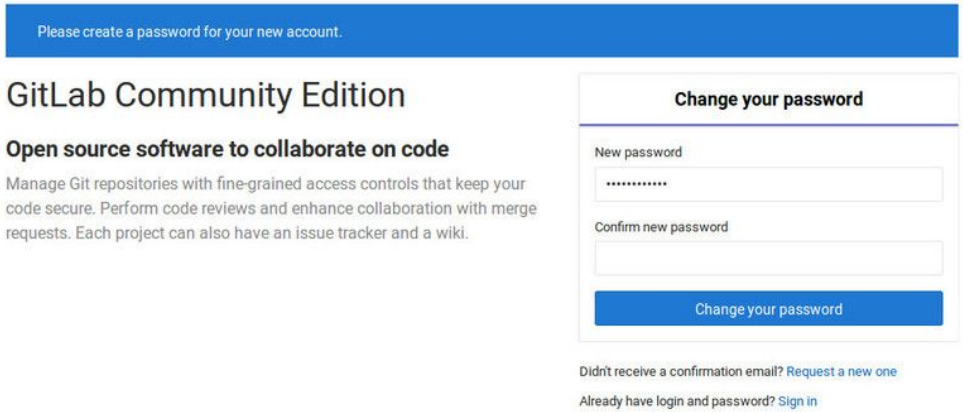
Uncomment those lines (remove the #) and then change false to true and enter a valid email address inside the empty [ ] characters.

Save and close that file. Reconfigure GitLab with the command:

```
sudo gitlab-ctl reconfigure
```

Configuring GitLab

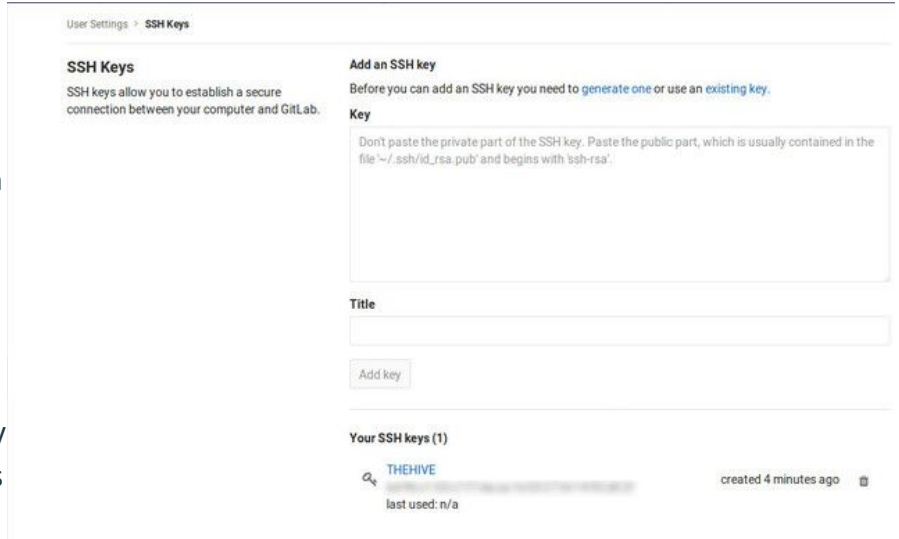
Once the reconfiguration happens, open a browser and point it to either **https://DOMAIN** or **http://IP** (Where DOMAIN is the domain of the server or IP is the IP address of the server). You will be prompted to change the administrator account password *(Figure A)*.



Configuring GitLab: **Adding ssh keys**

In order to push or pull to your projects, you must add remote machine account ssh-keys to your new GitLab account. To do this, find the ssh pubkey on the remote machine--on Linux you can do this by issuing the command cat ~/.ssh/id_rsa.pub and copying the output. Back on your GitLab account, click the Profile drop-down in the upper right corner and click Settings. In the Settings window, click SSH Keys *(Figure B)*.

Copy the contents of the SSH pubkey in the Key section, give the key a title, and click Add key. With the public key added, you should now be able to push and pull changes to the GitLab server.



Q&A