**Setting Up GitLab Community Edition**

GitLab has been an integral part of the infrastructure at LearnLinuxTV for several years, and its functionality has exceeded expectations. GitLab is a version control system based on Git, similar to GitHub, but it offers more than just a Git interface. It includes a wide range of features for developer collaboration, software delivery, and much more. Additionally, the GNOME project uses GitLab for its own development, showcasing its scalability and reliability.

### **Prerequisites**

Before starting the GitLab installation process, there are a few essential requirements to consider:

1. **Linux Instance**:  
   For this guide, we’ll be using Ubuntu 22.04. While GitLab itself can run on various Linux distributions, the instructions here are tailored for Ubuntu 22.04. If you’re using a different distribution, the steps may differ slightly.
2. **Recommended Hardware**:  
   GitLab runs best on servers with:
   * Minimum of **4 CPU cores**
   * At least **4 GB of RAM** (8 GB is ideal for better performance)
   * Sufficient **storage** for your repositories (160 GB storage is a good starting point for a cloud instance)
3. **Domain**:  
   You will also need a domain name for accessing GitLab. If you don’t already have one, you can purchase it from a registrar of your choice. Hover is a highly recommended registrar due to its excellent customer service and transparent pricing. Once you have the domain, make sure to point it to the IP address of your GitLab server.
4. **Preliminary Setup**:  
   Before diving into the GitLab setup, ensure that your server is up-to-date, you have created a non-root user for administration, and SSH is properly secured. I have a separate video for setting up a secure server environment, which I recommend reviewing if you haven’t already completed these steps.

### **Step 1: Installing the GitLab Repository**

To begin the GitLab installation, you’ll first need to install the official GitLab repository. This will allow you to access the GitLab Community Edition (CE).

Run the following command to add the GitLab repository:

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

**Security Tip**: Always inspect scripts fetched from the internet before running them. I have verified this script prior to creating this guide, but it’s good practice to review the script yourself before execution.

### **Step 2: Installing GitLab CE**

Now that the repository is set up, you can install GitLab by running the following command:

sudo apt install gitlab-ce

While GitLab will be installed, it is not yet configured. To configure GitLab and make it accessible, follow the next steps.

### **Step 3: Configuring GitLab**

1. **Edit GitLab Configuration**:  
   The configuration for GitLab resides in the /etc/gitlab/gitlab.rb file. Open this file for editing:

* sudo nano /etc/gitlab/gitlab.rb

1. **Update the External URL**:  
   Locate the line beginning with external\_url and update it to match the domain or subdomain you have configured for GitLab. For example:

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

1. **Reconfigure GitLab**:  
   After making this change, run the following command to apply the configuration:

* sudo gitlab-ctl reconfigure
* This process may take a few minutes to complete.

### **Step 4: Retrieving the Root Password**

Once GitLab has been configured, you will need to retrieve the initial root password, which is stored in a file on the server. To access this password, run the following command:

cat /etc/gitlab/initial\_root\_password

Make sure to retrieve this password promptly, as this file will delete itself within 24 hours.

With the root password, you can now log into GitLab through a web browser using the username root and the password retrieved above.

### **Step 5: Securing GitLab with HTTPS**

At this stage, GitLab is installed and accessible, but the connection is not secure. To secure your GitLab server with HTTPS, follow these steps:

1. **Edit GitLab Configuration Again**:

* sudo nano /etc/gitlab/gitlab.rb

1. **Enable HTTPS**:  
   Modify the external\_url line to use https instead of http:

* external\_url 'https://gitlab.example.com'

1. **Enable Let’s Encrypt for SSL Certificates**:  
   In the same configuration file, search for the Let’s Encrypt section and enable it:

* letsencrypt['enable'] = true  
  letsencrypt['contact\_emails'] = ['your\_email\_address\_here']  
  letsencrypt['auto\_renew'] = true

1. **Reconfigure GitLab**:  
   After making these changes, reconfigure GitLab once again:

* sudo gitlab-ctl reconfigure

### **Conclusion**

At this point, your GitLab instance should be fully set up and accessible over HTTPS with a valid SSL certificate. You are now ready to start using GitLab for your development and collaboration needs.