

How to Install Gitea Code Hosting Platform with HTTPS on Debian 10

Gitea is a code hosting web application written in Go and forked from Gogs. As its name suggests, it is designed to be used with the popular source control program Git, similarly to Gitlab and Github. This guide will explain how to install Gitea on Debian 10 behind an HTTPS reverse proxy (Nginx).

Requirements

- A Debian 10 system on which you have root privileges.
- A registered domain name pointing to your server.
- The \$EDITOR environment variable should be set.
- Access to an SMTP server for email notifications (optional).

Step 1: Preparing the system

Start by updating your package index and install any available updates:

```
apt update
apt upgrade -y
reboot
```

For this setup, several software packages are required:

- Git, a dependency of Gitea.
- PostgreSQL, as Gitea requires a database.
- Nginx, which will be used as a reverse proxy.
- Certbot, a utility for obtaining Let's Encrypt SSL certificates.
- Sudo, to run commands as the postgres system user.

Install them as follows:

```
apt install -y git nginx certbot postgresql sudo
```

Next, create a user to run Gitea:

```
adduser --system --disabled-password --group --shell /bin/bash --home /home/gitea gitea
```

Then create the directory structure for Gitea:

```
mkdir -p /var/lib/gitea/{data,log} /etc/gitea /run/gitea
```

And set ownerships and permissions as follows:

```
chown -R gitea:gitea /var/lib/gitea
chown -R gitea:gitea /run/gitea
chown -R root:gitea /etc/gitea
chmod -R 750 /var/lib/gitea
chmod 770 /etc/gitea
```

The permissions on /etc/gitea are temporary and will be tightened after running the web installer.

Step 2: Database Setup

Make sure Postgres is enabled and running:

```
systemctl enable --now postgresql@11-main.service
```

Then create a user role and database to be used by Gitea:

```
sudo -u postgres psql
postgres=# CREATE ROLE gitea LOGIN ENCRYPTED PASSWORD 'your_password';
postgres=# CREATE DATABASE gitea;
postgres=# GRANT ALL PRIVILEGES ON DATABASE gitea TO gitea;
postgres=# exit;
```

Step 3: Installing Gitea

Download the latest linux-amd64 binary from [Gitea's download page](#). For example:

```
wget https://dl.gitea.io/gitea/master/gitea-master-linux-amd64 -O
/usr/local/bin/gitea
chmod 755 /usr/local/bin/gitea
```

Next, create a systemd unit file for Gitea:

```
$EDITOR /etc/systemd/system/gitea.service
```

And enter the following:

```
[Unit]
Description=Gitea (Git with a cup of tea)
After=syslog.target
After=network.target
Requires=postgresql.service
[Service]
Type=simple
User=gitea
Group=gitea
WorkingDirectory=/var/lib/gitea/
```

```
RuntimeDirectory=gitea
ExecStart=/usr/local/bin/gitea web -c /etc/gitea/app.ini
Restart=always
Environment=USER=gitea HOME=/home/gitea GITEA_WORK_DIR=/var/lib/gitea
[Install]
WantedBy=multi-user.target
```

Make sure the new unit is loaded:

```
systemctl daemon-reload
```

Then instruct systemd to start Gitea at system startup:

```
systemctl enable gitea.service
```

Step 4: Configuring Gitea

For the initial configuration, we'll use the included web install script. First, start Gitea:

```
systemctl start gitea.service
```

Then navigate to http://your_domain:3000/install and fill in the required parameters as follows:

- Database Type: PostgreSQL
- Host: 127.0.0.1:5432
- Username: gitea
- Password: Enter the password you chose during Postgres role creation.
- Database Name: gitea
- SSL: Disable
- Site Title: Title of your choice.
- Repository Root Path: /var/lib/gitea/data/repositories
- Git LFS Root Path: /var/lib/gitea/data/lfs
- Run As Username: gitea
- SSH Server Domain: your_domain
- SSH Server Port: 22
- Gitea HTTP Listen Post: 3000
- Gitea Base URL: https://your_domain/
- Log Path: /var/lib/gitea/log

Configure email and the remaining settings as deemed fit, then click „Install Gitea“. You will be redirected to a faulty URL. This is normal, as we haven't configured Nginx or HTTPS yet. For performance reasons, we will now configure Gitea to listen on a unix socket instead of the default TCP port.

Stop Gitea before proceeding:

```
systemctl stop gitea.service
```

Tighten permissions on /etc/gitea as shown below. This prevents anyone not in the gitea group from

reading app.ini, which contains sensitive information, including database credentials.

```
chmod 750 /etc/gitea
chown root:gitea /etc/gitea/app.ini
chmod 640 /etc/gitea/app.ini
```

Open its configuration file:

```
$EDITOR /etc/gitea/app.ini
```

Remove the following line from the [server] section:

```
HTTP_PORT = 3000
```

And add the following lines to the [server] section:

```
HTTP_ADDR      = /run/gitea/gitea.sock
PROTOCOL       = unix
UNIX_SOCKET_PERMISSION = 666
```

From:

<https://blog.cooltux.net/> - **TuxNet DokuWiki**

Permanent link:

<https://blog.cooltux.net/doku.php?id=it-wiki:git:gitea&rev=1580989114>

Last update: **2020/02/06 11:38**

