Install and Setup Forgejo

https://www.youtube.com/embed/FPVpKCvFQr8

Where did Forgejo come from?

- A project called Gitea was pretty popular, and the main developers / maintainers were taking Gitea in a direction that the community members weren't thrilled about. Thus Forgejo is a fork of Gitea, but now maintained in it's own repository.

What is it?

- Forges are also known as places to store your repository of code, text, markdown, instructions. While this forge is based on Git, thee are lots out there. They were originally designed for making code more easy to maintain over time, and to track where changes happened, and who made them, and so on.

What's it useful for?

- While originally intended for code repositories, you can use Forges for so many other purposes.
  - Notes and revision tracking
  - Tutorials
  - Blog posts
  - literally hosting websites
  - Ansible or other scripted playbooks
  - Community contributions to documentation
  - the list goes on and on.

Setup a Server

In most self hosted applications these days, you'll need a machine to act as a server. This can be a machine in your home / business (such as an old laptop or desktop, a Raspberry PI or Single Board Computer (SBC), or even the computer you're reading this article from), a VM / Container hosted on one of your machines, or a VPS (Virtual Private Server) hosted by companies like RackNerd, Digital Ocean, Linode, Vultr, and so many more. Regardless of which option you choose, you'll want to do a few things to get the server setup properly.

Install updates to our Server

Ubuntu / Debian

```
sudo apt update && sudo apt upgrade -y
```

RedHat / CentOS / Fedora / Alma / Rocky

```
sudo dnf update -y
```

Add a non-root / sudo user on the server

Generally, when you setup a new server, the VPS (Virtual Private Server) service sets up a default "root" user for you. It's considered unsafe to do everything as "root", so let's setup a non-root user who has super user (sudo) privileges.

`adduser <username>` You'll be prompted to enter and confirm a password for this user. You'll also be asked for some user information like Name, etc, but this is not required information. At the end, confirm the entries, and you'll have your new user.

Next, we need to add the user to the super user group.

Ubuntu / Debian

```
usermod -aG sudo <username>
```

RedHat / CentOS / Fedora / Alma / Rocky

```
usermod -aG wheel <username>
```

Now, you can log out of the system, and log back in as your new non-root super user.

Install Docker and Docker Compose

Fortunately, there is a single line command that will install both Docker and Docker Compose for us on most Linux based distributions.

We need the 'curl' utility to get this to work, so if you don't have it, you'll want to install it first with

Ubuntu / Debian

```
sudo apt install curl -y
```

RedHat / CentOS / Fedora / Alma / Rocky

```
sudo dnf install curl -y
```

Next, we'll run the command to install Docker and Docker Compose:

`curl https://get.docker.com | sh`

You may be prompted to enter your super user password, so be ready for it. Once you do, the install should proceed.

Once complete. we want to add our user to the 'docker' group so we can do `docker` and `docker compose` commands without having to type in `sudo` each time.

`sudo usermod -aG docker <username>`

Now we'll logout / exit the session, and log right back in so the updated group will take effect.

Install Forgejo

Let's create a folder structure four our installation. The great thing about docker is you can create a parent folder, then create separate application and service folders inside of it. When it's time to create backups, just zip up that parent folder and back it up to any place you store your regular backups.

`mkdir -p docker/forgejo/forgejo-data`

Let's move into the first level 'forgejo' folder:

`cd docker/forgejo`

Now we'll create a file called "compose.yaml", and put our docker compose yaml syntax configuration in that file.

`nano compose.yaml`

Next, copy the yaml block below, and paste it into the file we just created:

```
networks:
  forgejo:
    external: false

services:
  server:
    image: codeberg.org/forgejo/forgejo:12
    container_name: forgejo
    environment:
      - USER_UID=1001
      - USER_GID=1001
      - FORGEJO__database__DB_TYPE=mysql
```

```
      - FORGEJO__database__HOST=db:3306
      - FORGEJO__database__NAME=forgejo
      - FORGEJO__database__USER=forgejo
      - FORGEJO__database__PASSWD=< a long strong password here >
    restart: unless-stopped
    networks:
      - forgejo
    volumes:
      - ./forgejo-data:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "222:22"
    depends_on:
      - db


  db:
    image: mysql:8
    restart: unless-stopped
    environment:
      - MYSQL_ROOT_PASSWORD=< a long strong different password here >
      - MYSQL_USER=forgejo
      - MYSQL_PASSWORD=< a long strong password here > # must match the one from the first
section
      - MYSQL_DATABASE=forgejo
    networks:
      - forgejo
    volumes:
      - ./mysql:/var/lib/mysql
```

In the file, make sure to change the password item values surrounded by less than "<" and greater than ">" signs to actual long, strong, passwords. If you want, you can open another terminal window, and run:

`openssl rand -base64 32`

Re-run it for the root password as well in the second section. Make sure the value for "FORGEJO__database__PASSWD" in the first section, and "MYSQL_PASSWORD" in the second section match exactly, or the application won't function.

Use CTRL + O to save your changes, press Enter to confirm, and exit the nano editor with CTRL + X.

Pull the Docker Images for Forgejo

Now we'll pull the docker images with the command:

```
docker compose pull
```

Once that's done, you're ready to start Forgejo with the commands:

```
docker compose up -d && docker compose logs -f
```

As long as you don't see any errors as the logs start to scroll, once the scrolling has stopped you'll be ready to access your Forgejo web interface.

At this point, you may want to setup a Reverse Proxy so you can reach your Forgejo site from outside your Local Network, in which case, I have a section on that further down. It's important that you make this decision before you go through the setup screen, as you want to set that up first, otherwise you'll have to find and change a configuration file later on. it's not hard, but over time, as this article ages, I can't guarantee it will be configured from the same place.

Access your Forgejo Site

If you want to setup Forgejo to be accessed outside your LAN via a fully qualified domain name, i have a section on the basics of Reverse Proxy setup further down in this article. You'll want to own the domain for which you are setting up a sub-domain (e.g. forge.yourcooldomain.com). Also, you'll want to setup all the necessary plumbing before going through the initial configuration page of your Forgejo site.

That said, you can test that the site is up and running by going to the IP address of your host machine, and the port you mapped in the compose.yaml document. If you didn't change the port on the left side of the mapping, then the web port will be 3000 by default.

NOTE: The SSH port for connecting and pushing / pulling code is defaulted to port 222, not 22 as SSH is going to need to be proxied.

I visited the IPv4 address of my server on port 3000 in my favorite modern browser, and you should do the same. In my case I went to http://192.168.1.21:3000. If you've set things up properly, you should be greeted by the Forgejo setup / configuration screen.

There's a good bit to fill out on this screen so pay attention. A few sections are optional, but in my opinion important, and they are collapsible sections that are not expanded by default.

I have an overview of what to fill out on this screen after the 'Reverse Proxy Setup' section below. If you are just running Forgejo locally, and don't intend to run it with a domain name on the internet, then feel free to skip down to the configuration overview.

Reverse Proxy Setup

A reverse proxy is a tool used to allow you to make your applications and services available through the use of a domain name / subdomain. Insted of going to 192.168.1.21:3000 you'd be able to go to `forge.yourgreatdomain.com", and access your application / service from anywhere with an internet connection.

What you'll need

- A domain name you own, or at least have access to setup A / CNAME records for.
- The public IPv4 IP address of the server / location where you're running your Reverse Proxy
- The private (LAN / local) IPv4 address of the host machine where you're running your application or service
- The port number where your application or service is running
- A Reverse Proxy Application / Service setup and Running

1. Create the subdomain you want for this application / service (e.g. forgejo.mycooldomain.com) in your DNS provider's setup (usually with your domain name registrar).
2. Associate this record to an A-record or CNAME record (sometimes all one step with step 1 above), and enter the public IPv4 address of your reverse proxy application server.
3. Set the TTL (Time To Live) to be 300 seconds, or 5 minutes if possible. Choose the shortest time they'll allow.
4. In the Reverse Proxy you need to ensure any request for the subdomain you just created your A / CNAME DNS record for will be forwarded to the correct local (internal / LAN) IPv4 address where your application / service is running.
5. Make sure you forward the subdomain to the correct port for the application / service.
6. Make sure you are setting up LetsEncrypt to request valid SSL Certificates for your new entry. Most reverse proxies offer this as a 'checkbox' type option, or do it by default.
    1. If you don't want to use LetsEncrypt certificates, you are welcome to create / buy your own, but you're responsible for making sure they are protected and valid.
7. Now, you can go through the Forgejo start screen, keeping in mind the subdomain you've setup.

Forgejo Start Screen Setup / Configuration

1. You can leave the database section as-is except for the Password field. You need to copy the MYSQL_PASSWORD from your compose file (not the root password) and paste it into the Password field to ensure you have the right password here.

2. Feel free to adjust the Site Title and Description to meet your desires.
3. Consider the Email and Login settings for the site. This can be important to a good experience for you, and / or for other users of your site.
4. If you want email to work, you should setup the Email section. Make sure you know the SMTP host, port number, username, and password for the email service you want to use.
5. Definitely expand the last section and create your initial admin user. make a really long strong password for this user as well.

Forgejo Features to Setup

Organization(s)

Under your Avatar (top right) > Settings > Organizations

You can create Organizations under which you wish to work, have repositories and projects, and more.

Built in Authentication (TOTP for 2FA)

1. Avatar (top right) > Settings > Authentication
    1. Enable 2 Factor Authentication for tOTP
    2. Alternatively, you can set up WebAuthN Keys

External Authentication

Go to your avatar in top right > Site Administration > Identity and Access > Authentication Sources

1. LDAP
2. FreeIPA
3. OAuth2 / OIDC
    1. Authentik
        1. Create an Application called Forgejo
        2. Create an OAuth2 / OIDC provider for Forgejo
        3. Bind it to a group your user (and any other users you want to have access to the application) belongs to.
        4. In Forgejo, create a new OAuth2/OIDC authentication source
        5. Call it 'Authentik' to easily identify it at login time.
        6. Copy the Client ID (auto-generated) from Authentik to ForgeJo
        7. Copy the Client Secret (auto-generated) from Authentik to Forgejo
        8. Save in Authentik
        9. Access the Provider you just Created in Authentik by clicking on it's name link.
        10. Copy the Configuration URL from Authentik to the Autodiscovery URL in Forgejo
        11. Save in Forgejo.
        12. Log out of Forgejo, and log back in, this time using the 'Login with Authentik' option.

13. Once logged in, you'll be asked if you want to register as a new user, or link to an existing account. I linked mine, but it's up to you.
14. Voila! You are now logged in using yoru self-hosted OIDC provider.

SMTP / Email (available from start up screen)

1. Enter your SMTP Host server (e.g. smtp.purelymail.com, or smtp.google.com, etc)
2. Enter the send from email (I recommend creating a specific email for this purpose if you have that option.
3. Enter the port (generally 587 for StartTLS, or 465 for SSL)
4. Enter the Username for the email (usually the email address)
5. Enter the Password for the email user.

SSH Keys

SSH Keys are extremely secure, and provide a really easy way to commit changes to anything you're working on within a repository. I cannot recommend highly enough that you set this up and use these keys for your commits.

Revision #2
Created 2025-08-12 11:39:19 UTC by Brian McGonagill
Updated 2025-08-12 12:37:15 UTC by Brian McGonagill