Install Peertube with Docker and Docker Compose

https://www.youtube.com/embed/lw9XQmlY0gU

Step 1: Set up Your Server

To host Peertube, you'll need a machine to act as a server. This can be a physical machine (e.g., an old laptop or desktop), a Virtual Machine (VM) or Container hosted on one of your machines, or a Virtual Private Server (VPS) from a provider like Digital Ocean or Linode.

Regardless of the option you choose, follow these steps to set up your server:

Install Updates

Update and upgrade your server's packages using the following commands:

- For Ubuntu/Debian: `sudo apt update && sudo apt upgrade -y`
- For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf update -y`

Create a Non-Root User

Create a non-root user with superuser (sudo) privileges:

1. Add a new user using `adduser <username>`
2. Set the password for this user
3. Fill any relevant details (optional)
4. Enter 'Y' to accept your entries, and press Enter.
5. Add the user to the sudo group:
   - For Ubuntu/Debian: `usermod -aG sudo <username>`
   - For RedHat/CentOS/Fedora/Alma/Rocky: `usermod -aG wheel <username>`

Ubuntu / Debian

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

RedHat / CentOS / Fedora / Alma / Rocky

`usermod -aG wheel <username>`

Let's add a few tools to make sure we have what we need for the upcoming steps.

Ubuntu / Debian

`sudo apt install nano git wget curl openssh-server -y`

RedHat / CentOS / Fedora / Alma / Rocky

`sudo dnf install nano git wget curl openssh-server -y`

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

Step 2: Install Docker and Docker Compose

Install Docker and Docker Compose on your server:

1. Run the command to install Docker and Docker Compose: `curl https://get.docker.com | sh`

Add Your User to the Docker Group

Add your non-root user to the docker group so you can use Docker commands without sudo:

`sudo usermod -aG docker <username>`

Step 3: Peertube with Docker Compose

Create a new directory for Peertube and create the necessary files:

1. Create a new folder: `mkdir -p docker/peertube`
2. Move into the folder: `cd docker/peertube`
3. Create a new file called `compose.yaml` using the command `nano compose.yaml`

Copy and paste the following YAML snippet into `compose.yaml`:

```
services:


  peertube:
    image: chocobozzz/peertube:production-bookworm
    networks:
      default:
        ipv4_address: 172.18.0.42
```

```
        ipv6_address: fdab:e4b3:21a2:ef1b::42
    env_file:
      - .env
    ports:
     - "1935:1935" # Comment if you don't want the live feature
     - "9000:9000" # Uncomment if you use another webserver/proxy
    volumes:
      - assets:/app/client/dist
      - ./docker-volume/data:/data
      - ./docker-volume/config:/config
    depends_on:
      - postgres
      - redis
      - postfix
    restart: "always"

  postgres:
    image: postgres:13-alpine
    env_file:
      - .env
    volumes:
      - ./docker-volume/db:/var/lib/postgresql/data
    restart: "always"

  redis:
    image: redis:6-alpine
    volumes:
      - ./docker-volume/redis:/data
    restart: "always"

  postfix:
    image: mwader/postfix-relay
    env_file:
      - .env
    volumes:
      - ./docker-volume/opendkim/keys:/etc/opendkim/keys
    restart: "always"

networks:
  default:
```

```
    enable_ipv6: true
    ipam:
      driver: default
      config:
      - subnet: 172.18.0.0/16
      - subnet: fdab:e4b3:21a2:ef1b::/64

volumes:
  assets:
  certbot-www:
```

Make any necessary changes to the configuration. Essentially, determine if you want to use port 9000 to access the UI (if 9000 is free on the host leave it as is, if not, change the left side of the port mapping to some other port that is free).

Step 4: Set up the .env File

Create a new file called `.env` using the command `nano .env`. Copy and paste the following content into `.env`:

plaintext

```
# Database / Postgres service configuration
POSTGRES_USER=peertube
POSTGRES_PASSWORD=< a-L0n6-sTR0ng-PasswoRd-4-You5-d8 >
# Postgres database name "peertube"
POSTGRES_DB=peertube
# The database name used by PeerTube will be PEERTUBE_DB_NAME (only if set) *OR*
'peertube'+PEERTUBE_DB_SUFFIX
PEERTUBE_DB_NAME=peertube
PEERTUBE_DB_SUFFIX=_prod
# Database username and password used by PeerTube must match Postgres', so they are copied:
PEERTUBE_DB_USERNAME=$POSTGRES_USER
PEERTUBE_DB_PASSWORD=$POSTGRES_PASSWORD
PEERTUBE_DB_SSL=false
# Default to Postgres service name "postgres" in docker-compose.yml
PEERTUBE_DB_HOSTNAME=postgres

# PeerTube server configuration
# If you test PeerTube in local: use "peertube.localhost" and add this domain to your host
```

```
file resolving on 127.0.0.1
PEERTUBE_WEBSERVER_HOSTNAME=<peertube.yourgreatdomain.com>
# If you just want to test PeerTube on local
#PEERTUBE_WEBSERVER_PORT=9000
#PEERTUBE_WEBSERVER_HTTPS=true
# If you need more than one IP as trust_proxy
# pass them as a comma separated array:
PEERTUBE_TRUST_PROXY=["<192.168.1.21>", "172.18.0.0/24", "<202.244.12.221>"]

# Generate one using `openssl rand -hex 32`
PEERTUBE_SECRET=< use the command `openssl rand -hex 32` >

# E-mail configuration
# If you use a Custom SMTP server
PEERTUBE_SMTP_USERNAME=< noreply@yourgreatdomain.com >
PEERTUBE_SMTP_PASSWORD=< your-email-password-for-the-above-act >
# Default to Postfix service name "postfix" in docker-compose.yml
# May be the hostname of your Custom SMTP server
PEERTUBE_SMTP_HOSTNAME=< smtp.yourmail.com >
PEERTUBE_SMTP_PORT=< 25,465,587 >
PEERTUBE_SMTP_FROM=< noreply@yourgreatdomain.com >
PEERTUBE_SMTP_TLS=< true, false >
PEERTUBE_SMTP_DISABLE_STARTTLS=< true, false >
PEERTUBE_ADMIN_EMAIL=< you@yourgreatdomain.com >

# Postfix service configuration
#POSTFIX_myhostname=<MY DOMAIN>
# If you need to generate a list of sub/DOMAIN keys
# pass them as a whitespace separated string <DOMAIN>=<selector>
#OPENDKIM_DOMAINS=<MY DOMAIN>=peertube
# see https://github.com/wader/postfix-relay/pull/18
#OPENDKIM_RequireSafeKeys=no

# If you want to enable object storage for PeerTube, set the following variables.
#PEERTUBE_OBJECT_STORAGE_ENABLED=
#PEERTUBE_OBJECT_STORAGE_ENDPOINT=
#PEERTUBE_OBJECT_STORAGE_REGION=
#PEERTUBE_OBJECT_STORAGE_CREDENTIALS_ACCESS_KEY_ID=
#PEERTUBE_OBJECT_STORAGE_CREDENTIALS_SECRET_ACCESS_KEY=
#PEERTUBE_OBJECT_STORAGE_STREAMING_PLAYLISTS_BUCKET_NAME=
```

```
#PEERTUBE_OBJECT_STORAGE_STREAMING_PLAYLISTS_PREFIX=
#PEERTUBE_OBJECT_STORAGE_STREAMING_PLAYLISTS_BASE_URL=
#PEERTUBE_OBJECT_STORAGE_WEB_VIDEOS_BUCKET_NAME=
#PEERTUBE_OBJECT_STORAGE_WEB_VIDEOS_PREFIX=
#PEERTUBE_OBJECT_STORAGE_WEB_VIDEOS_BASE_URL=
#PEERTUBE_OBJECT_STORAGE_USER_EXPORTS_BUCKET_NAME=
#PEERTUBE_OBJECT_STORAGE_USER_EXPORTS_PREFIX=
#PEERTUBE_OBJECT_STORAGE_USER_EXPORTS_BASE_URL=
#PEERTUBE_OBJECT_STORAGE_ORIGINAL_VIDEO_FILES_BUCKET_NAME=
#PEERTUBE_OBJECT_STORAGE_ORIGINAL_VIDEO_FILES_PREFIX=
#PEERTUBE_OBJECT_STORAGE_ORIGINAL_VIDEO_FILES_BASE_URL=
#PEERTUBE_OBJECT_STORAGE_CAPTIONS_BUCKET_NAME=
#PEERTUBE_OBJECT_STORAGE_CAPTIONS_PREFIX=
#PEERTUBE_OBJECT_STORAGE_CAPTIONS_BASE_URL=

# Comment these variables if your S3 provider does not support object ACL
#PEERTUBE_OBJECT_STORAGE_UPLOAD_ACL_PUBLIC="public-read"
#PEERTUBE_OBJECT_STORAGE_UPLOAD_ACL_PRIVATE="private"

#PEERTUBE_LOG_LEVEL=info

# /!\ Prefer to use the PeerTube admin interface to set the following configurations /!\
#PEERTUBE_SIGNUP_ENABLED=true
#PEERTUBE_TRANSCODING_ENABLED=true
#PEERTUBE_CONTACT_FORM_ENABLED=true
```

Replace placeholders with your own values, such as:

- `POSTGRES_PASSWORD`: a strong password for the PostgreSQL database
- `PEERTUBE_SECRET`: a secret string generated using `openssl rand -hex 32`
- `PEERTUBE_TRUST_PROXY`: the IP addresses that should be trusted as reverse proxies
- `PEERTUBE_WEBSERVER_HOSTNAME` : (e.g. peertube.yourgreatdomain.com)

Step 4: Pull and Start Containers

Pull the Peertube images and start the containers:

1. Run `docker compose pull` to ensure your YAML is valid and pull down the necessary images.
2. Run `docker compose up -d && docker compose logs -f` to start the containers and view their log output

If you see a repeating error like `exit code 1` or `exit code 255` over and over, then you need to bring down your containers, and make changes. What changes need to be made will depend on what you see in the logs for error messages. But the logs are your friend.

Hopefully you are not seeing issues, and will land on a message that shows your url and port number. Above that there will be some credentials.

> IMPORTANT
>
> Let's grab our root credentials now.
>
> First stop following the logs with CTRL + C (sometimes you have to do this twice).
>
> Now enter `docker compose logs peertube | grep -A1 root` This should output a user (root) and a password for that user. This is what you'll use to login the first time.
>
> Make note of these somewhere else, as they will not be in the logs forever.

Reverse Proxy

Setting up a reverse proxy is not overly complicated, but it really is a video on it's own. If you already have a reverse proxy setup, then you should know how to configure it to proxy your sites for you. If you don't I have some videos on setting them up. I'll link to them here, as it will be a useful tool for you to have for almost all self hosted projects.

Install and Setup the Pangolin Reverse Proxy / Secure Tunnel

Install and Setup the NginX Proxy Manager Reverse Proxy

Time to Login

Presuming you've setup your reverse proxy to point to your instance, you are ready to head to the URL of your instance.

Hoepfully you are presented wit your new PeerTube instance. You can click the 'Login' button in the upper right. Enter 'root' for the username, and the password you retrieved in the steps above.

You should be successfully logged in. Now you can run through all the setup for your new site.

IMPORTANT

It is highly recommended to do the following:

1. Change your root user password to something stronger than the auto-generated password. You can find users by going to 'Overview' in the left menu, then the 'Users' tab.
2. Setup a new user under which to create your channel(s) and post your content.
3. Logout, and back in as the new user.
4. Go through Channel setup.

You can find my peertube instance at https://pt.opensourceisawesome.com.

Support my Channel and Content

You can support this product by purchasing my book, buying some great Awesome Open Source merchandise, or through Patreon and / or Paypal.

Patreon: https://www.patreon.com/awesomeopensource

Paypal: https://paypal.me/BrianMcGonagill

Get the merch:

https://osia.printify.me/

Revision #2
Created 2025-11-16 12:50:56 UTC by Brian McGonagill
Updated 2025-11-18 16:40:34 UTC by Brian McGonagill