Kollaborate Encoder

Installation Guide

Kollaborate Encoder is an add-on to Kollaborate Server that allows you to automatically encode video files uploaded to your local server. Multiple servers can be used for high-traffic websites (additional server licenses required).

There are two ways to run Kollaborate Encoder: via Docker or directly on a Linux server. Docker is by far the simplest way and it is therefore our recommended method.

> ***Note***
>
> Manual installations are only available for Linux and so Docker is therefore the only way to host Kollaborate Encoder on macOS or Windows.

Requirements

For Docker installation:

- Docker (included with Docker Desktop for Mac/Win)
- Docker Compose (included with Docker Desktop for Mac/Win)

For manual installation:

- Linux computer (preferably Ubuntu)
- ImageMagick
- FFMPEG
- LibreOffice
- Node.js

Contents

Prerequisites

This guide assumes you are familiar with the following topics or tasks in your operating system:

- Managing files
- Using the command line
- Editing configuration files with a text editor
- Modifying file and user permissions
- Executing command line scripts
- Familiarity with basic database and network concepts

If you are not familiar with these topics, please research them before continuing.

Installing

To ensure maximum performance on Kollaborate Server we recommend installing Kollaborate Encoder on a separate server to the web server. Kollaborate Encoder should be assigned a fixed local IP address and should be able toaccess both Kollaborate Server and the Kollaborate Server storage volume over the local network. The storage volume should be a network volume that's mounted on both the web and encoding servers

Docker setup

The easiest and fastest way to get the server up and running is to run it via Docker.

> ***Note***
>
> As Kollaborate Encoder requires Linux, running through Docker is the only option if your server is running macOS or Windows.

1. Download the latest version of Kollaborate Encoder from https://www.kollaborate.tv/server after logging in.

2. Unzip the zip file to a convenient location on your computer.

3. Copy the file KollaborateEncoder/kollab_encoder.default.conf and rename the copy to kollab_encoder.conf.

4. Open kollab_encoder.conf in a text editor.

5. Change the site_url field to the URL of your Kollaborate Server web server. You may also choose to specify a password the web server will use when issuing commands to the encoder.

6. Docker installations require you to set a UUID as well. This helps the web server determine which encoder it is talking to if it can't from the IP address alone. The UUID can be any text string, but an easy way to generate a unique string is with the uuidgen command in a terminal window.

7. Install Docker Desktop for Mac or Windows from http://www.docker.com .

   (Note that if your host OS is Linux, Docker and Docker Compose are two separate downloads and you will need both of them. You should be able to install them from your OS's package manager.)

8. Open the file KollaborateEncoder/docker-compose.yml in a text editor.

9. **Look for the following line in the volumes section under volumes:**

   ```
   - source: ~/Documents/KollaborateServer/Storage
   ```

   Change this to the path of your storage volume where files uploaded to Kollaborate Server are stored. This should be the path as your host operating system sees it.

   Now save the file.

10. Open up a terminal window, then type `cd` followed by the path to the KollaborateEncoder folder that contains docker-compose.yml.

    For example:

    ```
    cd ~/Documents/KollaborateEncoder
    ```

11. Type `docker-compose up` and press Enter. This will begin setting up the Docker image.

12. Visit http://[your site]/admin and log into the Admin Area on the web server.

13. Under the Servers section, add the URL and port (default port is 8001) of the encoder and the password if you specified it, for example `http://1.2.3.4:8001`. Enter the same UUID you specified in the config file if you didn't leave it blank.

14. The Admin Dashboard page will show you the status of your encoder(s) and whether Kollaborate Server is able to communicate with them.

Useful Docker commands

These commands assume your current working directory is the folder that contains the docker-compose.yml file.

`docker-compose start` - Start the encoder

`docker-compose stop` - Stop the encoder running

`docker-compose restart` - Restarts the encoder

`docker-compose logs` - View error logs from the encoder (very helpful for troubleshooting)

`docker-compose exec [service] sh` - Access a Terminal to allow you to execute commands on the encoder's operating system. [service] should be replaced with the name of the service in docker-compose.yml, such as `encoder`.

`docker builder prune -a` - Removes the Docker build cache. This is useful for forcing the `docker-compose build` command to rebuild everything without using the cache. It can also save significant disk space once the images are built.

`docker-compose down`, followed by `docker-compose up --detach` - Recreates the images and removes any custom changes you may have made with `docker-compose exec [service] sh`. This restores everything to the out-of-the-box Docker image configuration. This does not change your Kollaborate Encoder configuration or delete any files.

Platform-specific Docker considerations

Mac

Disk read-write performance when hosting on macOS can be slower than for Windows or Linux. This can be sped up by adding **:delegated** to the volume reference in the docker-compose.yml file e.g. - ~/Documents/KollaborateServer/Storage:/kollabserver/storage:delegated

This feature delays syncing file system changes made in the container back to the volume in macOS. While this can significantly improve file system performance, it could lead to data loss in the event that the container exits before the data is synced back to macOS. You should therefore ensure that you only use "docker-compose stop" or pressing Ctrl-C only once in Terminal, to gracefully shut down the containers instead of killing them immediately.

Additionally, another problem that may occur is a mismatch between the encoder and web server, where the encoder cannot see a file that has been uploaded because it has not yet been synced back to the volume. This will cause encoding failures at first but the encoder will keep retrying and encoding will eventually start once the data gets synced.

If you find disk performance to be too slow when hosting on macOS, we recommend using Linux as the host OS, where disk performance differences between OS and container are essentially negligible. Docker is working on improving macOS disk performance so hopefully these differences will disappear over time.

Windows

You may need to enable access to the drive Kollaborate Encoder is stored on and also the storage volume in Docker Desktop’s preferences before you can run the container.

Ensure that Linux containers are enabled in Docker Desktop’s preferences.

Manual Ubuntu installation

Note

We **strongly** recommend you run Kollaborate Encoder via Docker. Docker handles most of the setup for you automatically and requires far less configuration than a manual installation. Only install manually if you have a specific reason for doing so.

Note

Important: Be careful when copying and pasting from this user guide as your PDF viewer may add extra whitespace around the commands. If you receive errors after running a command, double-check that there are no errant spaces in the middle of the command. Also check that you are entering double dashes (--) where appropriate instead of single dashes (-)

The following installation guide is for **Ubuntu 22.04** but it should be similar on other Linux distributions. We are installing to the folder `/kollabencoder` but feel free to change it to your preferred location.

1. **First enter these commands to make sure you have the required prerequisites to build from source:**

```
sudo su -c "echo 'deb https://us.archive.ubuntu.com/ubuntu jammy main multiverse' \
    >> /etc/apt/sources.list"

sudo apt-get update && sudo apt-get -y install autoconf automake \
build-essential libass-dev libfreetype6-dev libharfbuzz-dev libsdl2-dev \
libtheora-dev libtool libva-dev libvdpau-dev libvorbis-dev libxcb1-dev \
libxcb-shm0-dev libxcb-xfixes0-dev pkg-config texinfo zlib1g-dev yasm \
nasm make gcc-multilib libx264-dev libfdk-aac-dev libmp3lame-dev \
libopus-dev libvpx-dev libnode-dev node-gyp libssl-dev openssl \
wget webp librsvg2-bin
```

2. **Build NASM:**

```
mkdir /kollabencoder
mkdir /kollabencoder/ffmpeg_sources
cd /kollabencoder/ffmpeg_sources
wget https://www.nasm.us/pub/nasm/releasebuilds/2.14.02/nasm-2.14.02.tar.bz2
tar xjvf nasm-*.tar.bz2
cd /kollabencoder/ffmpeg_sources/nasm-2.14.02
./configure
make
make install
make distclean
```

3. **Enter the following terminal commands to build x264:**

```
cd /kollabencoder/ffmpeg_sources
wget https://code.videolan.org/videolan/x264/-/archive/stable/x264-stable.tar.bz2
tar xjvf x264-stable.tar.bz2
```

```
cd /kollabencoder/ffmpeg_sources/x264-stable
PATH="/kollabencoder/bin:$PATH"
./configure --prefix="/kollabencoder/ffmpeg_build" \
--extra-cflags="-march=native" --bindir="/kollabencoder/bin" \
--enable-static --enable-pic --disable-opencl --enable-shared
PATH="/kollabencoder/bin:$PATH"
make
make install
sudo cp libx264.so.* /lib/x86_64-linux-gnu
```

4. **Then build ffmpeg:**

```
FFMPEG_VERSION="6.1.1"
cd /kollabencoder/ffmpeg_sources
wget https://ffmpeg.org/releases/ffmpeg-${FFMPEG_VERSION}.tar.bz2
tar xjvf ffmpeg-${FFMPEG_VERSION}.tar.bz2
cd ffmpeg-${FFMPEG_VERSION}
PATH="/kollabencoder/bin:$PATH" \
    PKG_CONFIG_PATH="/kollabencoder/ffmpeg_build/lib/pkgconfig" ./configure \
    --prefix="/kollabencoder/ffmpeg_build" \
    --pkg-config-flags="--static" \
    --extra-cflags="-I/kollabencoder/ffmpeg_build/include -march=native" \
    --extra-ldflags="-L/kollabencoder/ffmpeg_build/lib" \
    --bindir="/kollabencoder/bin" \
    --enable-gpl \
    --enable-libass \
    --enable-libfdk-aac \
    --enable-libfreetype \
    --enable-libharfbuzz \
    --enable-libmp3lame \
    --enable-libopus \
    --enable-libtheora \
    --enable-libvorbis \
    --enable-libvpx \
    --enable-libx264 \
    --enable-nonfree \
    --enable-openssl \
    --enable-postproc \
    --disable-doc
PATH="/kollabencoder/bin:$PATH" make
make install
make distclean
```

5. **Install nodejs and npm:**

```
sudo apt-get install nodejs npm
```

6. **Install ImageMagick:**

```
sudo apt-get install imagemagick
```

7. **Install LibreOffice (for converting documents to PDF):**

```
sudo apt-get install libreoffice
```

8. Now copy the kollab_server.js, kollab_server.default.conf, package.json and the encoder_supporting_files folder from the KollaborateEncoder/encoder directory to a location of your choice on your hard drive.

9. **In Terminal, type** `cd [Kollaborate Encoder location]` **where the location is the folder you just copied the files to. Now type the following to install JavaScript dependencies:**

   ```
   npm install
   ```

10. Now that everything is installed, see the Configuring and Running Kollaborate Encoder section of this guide.

Configuring and Running Kollaborate Encoder

> ***Note***
>
> **Note:** this section only applies to manual installations. Docker installations can be run by typing "docker-compose up".

Rename the `kollab_encoder.default.conf` file to `kollab_encoder.conf`. Now open it in a text editor. Every line in the configuration file should end with a semicolon (;) and strings should not have quotation marks around them.

Specify the following to configure Kollaborate Encoder:

port - This specifies the port to listen on. Set this to any port that is accessible by Kollaborate Server over your network.

site_url - The URL with which to access Kollaborate Server. It is best to use a local URL such as http://192.168.0.1/mysite rather than a remote URL like http://www.mysite.com. The URL should not redirect.

ffmpeg_path - Specify the path to the ffmpeg executable. You can obtain this by typing `which ffmpeg` at the command prompt.

imagick_path - Specify the path to ImageMagick's `convert` executable. You can obtain this by typing `which convert` at the command prompt.

soffice_path - Specify the path to LibreOffice's `soffice` executable. You can obtain this by typing `which soffice` at the command prompt.

temp_storage_path - A local path on the Kollaborate Encoder server to store temporary files.

storage_path - The path to your Kollaborate Server storage volume.

password - Optional. This makes the encoder require a password in order to accept commands sent to it. Leave this field empty to specify no password. Remember to set the same password for the encoder on Kollaborate Server's Servers page in the Admin Area (http://[your_site_url]/admin).

uuid - Optional. This is a unique text string by which to identify the encoder to the web server. This can help the web server determine which encoder is talking to it if it can't from IP address alone. It can be any text string but you can easily generate a UUID with the `uuidgen` command in Terminal. Remember to set the same UUID for the encoder on Kollaborate Server's Servers page in the Admin Area (http://[your_site_url]/admin).

ffmpeg_extra_params - Optional. Append additional commands to the ffmpeg executable to customize encoding further. Separated into `ffmpeg_extra_params_video` and `ffmpeg_extra_params_audio` to separate video and audio file commands.

After saving the file, launch Kollaborate Encoder with the following command:

```
nodejs [path to kollab_encoder.js]
```

See Kollaborate Server Configuration below for information on pointing Kollaborate Server to your Kollaborate Encoder installation.

Kollaborate Server Configuration

If Kollaborate Server has not yet been installed, you can set up encoding servers from the installer, otherwise they can be set up from the Servers page in the Admin Area (you must be a superadmin to see this section).

When adding a server, add the local IP address of the encoding server as a URL. If you are using a port other than 80 or 443 (default is 8001), specify the port after a colon like so: `http://1.2.3.4:8001`. Specify a password and UUID if you set them in the config file. Then click Add Server.

An unlimited number of servers can be added but additional encoding licenses need to be purchased to unlock them and send jobs to them.

Click Continue in the installer or Save Settings on the admin page to save the new server information.

Testing

Visit the Dashboard page in the Admin Area of Kollaborate Server (http://[your_site_url]/admin). The encoders will be listed on this page, along with colored icons showing their status. Red means the server cannot access the encoder and green means it is accessible.

To test out the encoding functionality, upload a video file to the site by dragging it into the Files page. The encoding will begin after the file has successfully uploaded. If you try to immediately view the file you may receive a message saying it is still in the encoding queue but you can still download it to watch it back locally on your system.

After a few minutes (depending on the duration of the file) a green tick will appear next to the name of the video in the web player. Click on the tick and the page will reload and play the encoded version of the video. `docker-compose logs` or viewing `kollab_encoder.log` will show information about the encoding process and any errors that occurred.