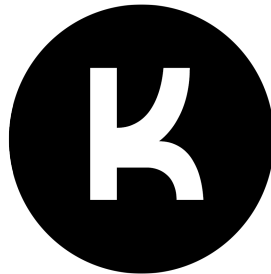


Kollaborate Server



Installation Guide

Kollaborate Server is a local implementation of the [Kollaborate](#) cloud workflow system that allows you to run the service in-house on your own server and storage.

There are two ways to run Kollaborate Server: 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 Server on macOS or Windows.

Features

All of the features of the cloud version including:

- Video review and feedback
- Comprehensive permissions system with department-level sandboxing
- Task tracking
- Team management
- Integration with [Digital Rebellion products](#)
- [More features...](#)

Kollaborate Server also features no storage or quota limitations in all editions and no per-project user limits on the Unlimited Edition.

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)
- [PHP 8.1](#)
- [MySQL 8.x](#) or [MariaDB 11.x](#)
- [Redis Stack Server 7+](#)
- [IonCube Loader](#)
- [ImageMagick](#)
- [FFMPEG](#)
- [LibreOffice](#)
- [Node.js](#)

Contents

Prerequisites	4
Installing	5
Docker setup	5
SSL certificates	7
Custom SSL Certificates	7
Useful Docker commands	8
Platform-specific Docker considerations	9
Manual Ubuntu installation	10
Sending Emails	13
Recommendations	14
Security software compatibility	15
Upgrading	16
Docker installation (version 3.7 or higher)	16
Docker installation pre-3.7	17
Upgrading a pre-Docker installation	17
Manual Ubuntu installation (version 3.7 or higher)	18
Manual Ubuntu installation pre-3.7	18
Configuration	20
Debug	20
PHP FPM (FastCGI Process Manager)	20
Tuning	20
Security	21
Frequently Asked Questions	22
Troubleshooting	23

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

If you have an existing installation of Kollaborate Server, please see the **Upgrading** section.

First setup a free account on <https://www.kollaborate.tv> and login to the **Kollaborate Server** page. Purchase an appropriate license for Kollaborate Server based on the number of users you will require per project, then download the latest version of Kollaborate Server.

The following guides provide the simplest method of installing the software and getting it up and running quickly. This is not recommended for a production environment so please see the **Recommendations** section for a list of security and performance tips.

You will need to be logged into an admin user account on your computer to install Kollaborate Server.

Warning

Kollaborate Server encrypts certain information in the database for privacy and security reasons. These encryption keys are stored in `config/keys.php`. It is important that you back this file up because Kollaborate needs it to decrypt data in the database. If you reinstall the software without this file, Kollaborate won't be able to read the database.

Docker setup

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

Note

As Kollaborate Server requires Linux, running through Docker is the only option if your server is running macOS or Windows.

1. Unzip the Kollaborate Server zip file to a convenient location on your computer.
2. If you're on Mac or Windows, install Docker Desktop from <https://www.docker.com> . If you're on Linux, install both docker and docker-compose via your package manager.

On Ubuntu it would be: `sudo apt install docker docker-compose`

3. Open the file **KollaborateServer/config.env** in a text editor.
4. Change the `MYSQL_USER` and `MYSQL_PASSWORD` values to something other than the defaults.
5. Change `KOLLAB_INSTALLER_PASSWORD` to something different from the default and make a note of what you changed it to, then save the file.
6. Open the file **KollaborateServer/docker-compose.yml** in a text editor.
7. Look for the following line in the `volumes` section under `webserver`:

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

Change the right-hand side of this line (after `source:`) to the path of your storage volume where files uploaded to Kollaborate Server will be stored. This should be a path on your host OS as your host OS sees it. For example, if you're on macOS, it would be the path to your storage drive that begins with `/Volumes/`. Now save the file.

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

For example: `cd ~/Documents/KollaborateServer`

9. Type `docker compose up` and press Enter. This will begin building and setting up the Docker containers.
- 10 Once setup has finished, run the Kollaborate Server installer by visiting <http://yoursite.com/install> in your web browser (this doesn't need to be on the same computer that you installed Kollaborate Server on). Replace `yoursite.com` with the domain name or IP address that points to the server - don't use `localhost`. The installer will ask you for the installer password. Enter the password you set earlier in `config.env` under `KOLLAB_INSTALLER_PASSWORD`. The installer will then guide you through the remaining steps to get everything set up.

Installation notes: Under Database Setup, keep the hostname as-is but change the MySQL username and password to whatever you set them to in the `config.env` file.

- 11 Once the installer has finished, the server is ready to use. Use the Super Admin account you created during installation to log into the admin area at <http://yoursite.com/admin> where you can add other admins, change site colors and customize other areas of the site, or visit <http://yoursite.com> to start creating projects.
- 12 For security, delete the "install" folder from the `KollaborateServer/web/html` directory.

SSL certificates

Kollaborate Server can request free SSL certificates from LetsEncrypt in order to encrypt connections to your site. This option is switched off by default and we strongly recommend you enable it. Because LetsEncrypt communicates over port 80, you will need to make sure this port is open on your firewall and forwarding correctly to the server.

Edit the `config.env` file and set `USE_LETS_ENCRYPT` to `true`. Make sure the `DOMAIN` field is set to the domain or subdomain you wish to use for Kollaborate Server and set an email address under `LETS_ENCRYPT_EMAIL`. LetsEncrypt will use this address to send important information such as notifications when your certificates are about to expire. (Kollaborate Server will automatically renew certificates so these emails will inform you if something has gone wrong and the certificate wasn't renewed for some reason.)

By default, `LETS_ENCRYPT_STAGING` is set to `true`. This is a test mode that doesn't communicate with LetsEncrypt's servers, so it can't generate valid certificates. LetsEncrypt limits the number of times your IP or domain can request certificates per week so we switched on staging mode by default to prevent your IP getting potentially banned for a week while you're in the process of setting things up.

Note

You must set `LETS_ENCRYPT_STAGING` to `false` to use it in a production environment.

If you previously had it in staging mode and after switching to production LetsEncrypt isn't letting you overwrite the staging certificates, perform the following steps to remove the certificates:

1. Stop the servers with `docker compose stop`.
2. Switch off LetsEncrypt completely by setting `USE_LETS_ENCRYPT` to `false` in `config.env`.
3. Start the servers with `docker compose start` (this deletes the old certificates).
4. After initialization has finished and the servers are all running, stop them with `docker compose stop` again.
5. Edit `config.env` and set `USE_LETS_ENCRYPT` to `true`.
6. Run `docker compose build` followed by `docker compose up`.

Custom SSL Certificates

Instead of LetsEncrypt, you can use a custom SSL certificate.

1. Name the certificate `main.pem` and place it in the **KollaborateServer/certificates** folder.
2. Edit `config.env` and make sure `USE_LETS_ENCRYPT` is set to `false` and `USE_THIRD_PARTY_SSL` is set to `true`.
3. Run `docker compose up`.

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 server

`docker compose stop` - Stop the server running

`docker compose restart` - Restarts the server

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

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

`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] bash`. This restores everything to the out-of-the-box Docker image configuration. This does not change your Kollaborate Server configuration, remove the database or delete any files.

Warning

Using `docker compose down` could cause your instance's MAC address to change which may invalidate your Kollaborate Server license. Licenses can be released at <https://www.kollaborate.tv/server> after logging in.)

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 Server 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.

We recommend using a WSL2 backend for Docker on Windows.

Manual Ubuntu installation

Note

We **strongly** recommend you run Kollaborate Server 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.

The following installation guide is for **Ubuntu 24.04** but it should be similar on other Linux distributions.

1. Make sure the required PHP dependencies are installed with the following commands:

```
sudo apt update && \  
  sudo apt -y install dirmngr ca-certificates \  
  software-properties-common apt-transport-https curl \  
  lsb-release gpg  
  
sudo add-apt-repository ppa:ondrej/php && sudo apt-get update  
  
sudo apt-get install -y php8.1 php8.1-dev \  
  php8.1-mysql php8.1-gd php8.1-curl php8.1-igbinary\  
  php8.1-mbstring php8.1-xml php8.1-redis php8.1-fpm \  
  ffmpeg imagemagick libreoffice nodejs npm
```

2. Install MariaDB 11.4:

```
curl -fsSL http://mirror.mariadb.org/PublicKey_v2 | \  
  sudo gpg --dearmor | \  
  sudo tee /usr/share/keyrings/mariadb.gpg > /dev/null  
  
MARIADB_REPO_URL="http://mirror.mariadb.org/repo/11.4/ubuntu/"  
  
echo "deb [arch=amd64,arm64 signed-by=/usr/share/keyrings/mariadb.gpg] \  
  ${MARIADB_REPO_URL} $(lsb_release -cs) main" | \  
  sudo tee /etc/apt/sources.list.d/mariadb.list  
  
sudo apt-get update && \  
  sudo apt-get -y install mariadb-server mariadb-client
```

3. (Optional) Secure MariaDB by running this script and following the prompts:

```
sudo mysql_secure_installation
```

4. (Optional) If you need to upgrade your MariaDB database from an earlier version, run the following command:

```
sudo mariadb-upgrade
```

5. Add the following line to `/etc/mysql/my.cnf`:

```
[mysqld]
sql_mode = ""
```

6. Install Redis Stack Server:

```
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
sudo chmod 644 /usr/share/keyrings/redis-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/redis.list
sudo apt update
sudo apt install redis-stack-server
```

7. Start Redis Stack Server:

```
sudo systemctl enable redis-stack-server
sudo systemctl start redis-stack-server
```

- Copy the file **KollaborateServer/webserver/www.conf** to **/etc/php/8.1/fpm/pool.d/**, overwriting the original file there.

9. Add your current user to the www-data group:

```
sudo usermod -a -G www-data $USER
```

- Move the file **KollaborateServer/webserver/server_config.toml** to **. KollaborateServer/webserver/bin.**

- Open **KollaborateServer/webserver/bin/server_config.toml** in a text editor. Set the port to 80 (or whatever your desired port is) and **web_root** to the full path to your **KollaborateServer/web/html** folder. Save the file.

- Type the following to get the server running (adjust to suit your Kollaborate Server path):

```
sudo service mariadb start
chmod +x KollaborateServer/webserver/bin/KollaborateServer_x86_64
sudo ./KollaborateServer/webserver/bin/KollaborateServer_x86_64
```

- Visit **http://[your site]/install**, where [your site] is the IP address or domain name of your site. Do not use localhost.

- The installer will guide you step-by-step through the process. You will only be able to go to the next step when the previous step is successful. The installer will try to do everything automatically for you, but if it can't it will give you exact instructions on what to do manually.

- After completing all the steps, you'll be presented with the a login screen. Login with the admin email address and password you specified during the installation process.

- In the Admin Area, go to **Configure** and setup any additional settings such as color schemes.

- Delete the install directory from the **/var/www/html** folder.

- Navigate to your site's main URL or domain name in your web browser and login. You can now set up projects and invite users.

See the **Sending Emails** section for a guide on configuring the server to send emails.

Sending Emails

Kollaborate Server has built-in SMTP email sending. You will need to already have access to an SMTP server, which could come from your business email provider, a free service like Gmail or a paid service like Amazon SES. This can be configured directly in the installer or later on from the Configure page of the Admin Area.

Your provider should offer SMTP details such as hostname, encryption type, etc on their site, which you should combine with the username and password you normally use to login.

Note

If you use two-factor authentication with your email provider, you may have to setup an App-Specific Password, which you should use instead of your normal login password. Your email provider will provide more details on this if it is needed.

Once you have set up your email configuration and saved it, go to Tools > Test Email in the Admin Area and send a test email to check that it's working. Note that the test email may take a few minutes to arrive.

Recommendations

The following recommendations are advised to improve performance and security in a production environment.

1. Create a new database user with a secure password, then update the site configuration page (located in the admin console) and set the database user to the new user. Then delete or disable the root account.
2. Give your server a fixed IP address on your local network. If you need access outside of your company, consider a dynamic DNS service or purchase a fixed IP address from your ISP. For simplicity we recommend giving Kollaborate Server its own domain or subdomain, such as kollaborate.mycompany.com.
3. Setup an SSL certificate for your domain to encrypt all communications with the server. You can get a free certificate from <https://letsencrypt.org> . The Docker installation can be configured to set this up for you automatically.
4. Use strong passwords for database users, admins and other logins. Remove or rename default users, or at the very least change their passwords.
5. Configure a firewall to block all incoming ports except 80 and/or 443.
6. Setting the environment variable `KOLLAB_INSTALLER_PASSWORD` to a non-empty value requires anyone trying to use the installer or upgrader to enter this password first. We recommend deleting the "install" folder for production use but this password prevents malicious users from modifying or discovering settings while you're still in the process of setting things up.

This password is automatically set up for Docker installations via the `config.env` file. Manual installations can set this up either via the shell (`export KOLLAB_INSTALLER_PASSWORD="1234"`) or by adding a line to the `/etc/environment` file: `KOLLAB_INSTALLER_PASSWORD="1234"`

You will most likely need to log out or reboot after setting this in order for the change to be recognized.

Security software compatibility

Security software like `mod_security` and `mod_evasive` are designed to protect your server from attackers who may try to post binary data or flood the server with requests that occur far too frequently to be from legitimate users.

However, Digital Rebellion applications legitimately using the Kollaborate API may be incorrectly considered to be attacking the server if the security settings are too strict.

Most security products will have a monitoring mode where errors are logged but no blocking takes place. We recommend testing Kollaborate Server in this mode to decide the best security settings to choose.

We recommend keeping security settings lenient so persistent attackers will be blocked but legitimately heavy users will not.

Upgrading

If you already have an existing Kollaborate Server installation, use these steps to upgrade to the latest version.

Docker installation (version 3.7 or higher)

Note

Upgrading versions might automatically upgrade your database to a new format, making it difficult to downgrade to an earlier version. We recommend downloading your existing database as a .sql file (which you can do from the Tools section of the Admin Area) before installing the new version.

If your installation is running via Docker and is Kollaborate Server 3.7 or higher, use the following steps:

1. Backup the following files and folders by moving them to a different location:

`KollaborateServer/web/config/`

`KollaborateServer/web/html/custom/`

`KollaborateServer/config.env`

`KollaborateServer/docker-compose.yml`

`KollaborateServer/kollab_server.lnc`

2. Overwrite the old KollaborateServer folder with the new one inside the installation zip file.
3. Copy the folders back to KollaborateServer/web. Compare the old config.env file with the new one, copying back your old settings while ensuring any new settings still exist in the file.
4. Edit the file **KollaborateServer/docker-compose.yml** in a text editor. Look for the following line in the `volumes` section under `webserver`:

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

Change the right-hand side of this line (after `source:`) to the path of your storage volume where files uploaded to Kollaborate Server are stored. This should be a path on your host OS as your host OS sees it. For example, if you're on macOS, it would be the path to your storage drive that begins with `/Volumes/`. Use the backed-up version of `docker-compose.yml` to refer back to if necessary.

5. In a terminal window, type `cd [path to KollaborateServer directory]`.
6. Type `docker compose build` followed by `docker compose up -d`.
7. Kollaborate Server will start and automatically upgrade the database for you. Note that this process could take several minutes, especially if you are upgrading from a version that is old. Type `docker compose logs webserver` to monitor this process and check for any errors.

8. Delete the “install” folder from the KollaborateServer/web/html folder.

Docker installation pre-3.7

If your installation is running via Docker and is lower than version 3.7, use the following steps:

1. Backup the following files and folders by moving them to a different location:

KollaborateServer/html/config/

KollaborateServer/html/custom/

KollaborateServer/config.env

KollaborateServer/docker-compose.yml

KollaborateServer/kollab_server.inc

2. Overwrite the old KollaborateServer folder with the new one inside the installation zip file.
3. Copy the folders back to KollaborateServer/web instead of KollaborateServer/html on the old version. Compare the old config.env file with the new one, copying back your old settings while ensuring any new settings still exist in the file.
4. Edit the file **KollaborateServer/docker-compose.yml** in a text editor. Look for the following line in the `volumes` section under `webserver`:

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

Change the right-hand side of this line (after `source:`) to the path of your storage volume where files uploaded to Kollaborate Server are stored. This should be a path on your host OS as your host OS sees it. For example, if you're on macOS, it would be the path to your storage drive that begins with `/Volumes/`. Use the backed-up version of `docker-compose.yml` to refer back to if necessary.

5. In a terminal window, type `cd [path to KollaborateServer directory]`.
6. Type `docker compose build` followed by `docker compose up -d`.
7. Kollaborate Server will start and automatically upgrade the database for you. Note that this process could take several minutes, especially if you are upgrading from a version that is old. Type `docker compose logs webserver` to monitor this process and check for any errors.
8. Delete the “install” folder from the KollaborateServer/web/html folder.

Upgrading a pre-Docker installation

Kollaborate Server used to support installation on macOS and Windows via MAMP and Uniform Server. This is no longer supported: Docker is now the only way to run Kollaborate Server on these platforms.

To move an older installation to Docker, perform the following steps:

1. Use a database tool like phpMyAdmin or Sequel Pro to backup your current database to a .sql file.

2. Follow the steps in this installation guide to install a Docker installation from scratch. When specifying the path to your storage volume in the Docker configuration files, choose the same storage location you were using with the previous version of Kollaborate Server.
3. Use a database tool like phpMyAdmin or Sequel Pro to connect to the database on the Docker instance and restore your backed up .sql file, overwriting the current database.
4. The restored database is older than the current version of the software, so it will need to be upgraded. Kollaborate will do this automatically when it starts.
5. Delete the install folder from the KollaborateServer/web/html folder.

Manual Ubuntu installation (version 3.7 or higher)

If your existing installation is a manual installation and is Kollaborate Server 3.7 or higher, use the following steps:

1. Backup the following files from your existing installation:

`/var/www/config/`

`/var/www/custom/`

`/var/www/kollab_server.inc`

2. Select all files in the KollaborateServer/web folder in the zip file and copy them to /var/www, overwriting the existing files.
3. Type `killall KollaborateServer_x86_64` to terminate the Kollaborate Server process. Then type `./[Kollaborate Server path]/webserver/bin/KollaborateServer_x86_64` to relaunch it.
4. Kollaborate Server will start and automatically upgrade the database for you. Note that this process could take several minutes, especially if you are upgrading from a version that is old. Type `docker compose logs webserver` to monitor this process and check for any errors.
5. Delete the "install" folder from the KollaborateServer/web/html folder.

Manual Ubuntu installation pre-3.7

If your installation is manually installed and is lower than version 3.7, use the following steps:

1. Backup the following files from your existing installation:

`/var/www/html/config/`

`/var/www/html/custom/`

`/var/www/kollab_server.inc` or `/var/www/html/kollab_server.inc`

2. Select all files in the KollaborateServer/web folder in the zip file and copy them to /var/www, overwriting the existing files.

3. Now copy the files back, in the following new locations:

`/var/www/config/`

`/var/www/custom/`

`/var/www/kollab_server.lnc`

4. Type `killall KollaborateServer_x86_64` to terminate the Kollaborate Server process. Then type `./[Kollaborate Server path]/webserver/bin/KollaborateServer_x86_64` to relaunch it.
5. Kollaborate Server will start and automatically upgrade the database for you. Note that this process could take several minutes, especially if you are upgrading from a version that is old. Type `docker compose logs webserver` to monitor this process and check for any errors.
6. Delete the "install" folder from the `KollaborateServer/web/html` folder.

Configuration

server_config.toml supports the following configuration parameters:

version - The version of the configuration file. If you have an old version, Kollaborate will automatically upgrade it for you.

port (optional) - The port on which to run the web server. Defaults to 80 if not specified.

web_root - The location of the web/html folder. If running inside a Docker container, it should be the path as Docker sees it, not as your host operating system sees it.

php_command - The command used to run PHP scripts. For example, it could be `"/usr/bin/php -c /etc/php/8.1/cli/php.ini"`

max_upload_size_gb (optional) - The maximum file upload size, in gigabytes. Defaults to 60.

scheduled_task_time_offset_from_utc (optional) - The number of whole hours to offset from UTC when running scheduled tasks. This can make a task scheduled to run at midnight actually run at midnight in your timezone.

check_for_updates (optional) - Set to false to stop Kollaborate checking for updates and notifying you within the interface.

Debug

To add the debug section, add the line `[debug]`, followed by any of the following options:

log_level (optional) - The verbosity of logging. Available options are (in order of verbosity): info, error, warn, debug, trace. Defaults to "debug".

PHP FPM (FastCGI Process Manager)

To add the FPM section, add the line `[php_fpm]`, followed by any of the following options:

php_fpm_command - The command to launch the PHP FPM process. This should look something like `"php-fpm8.1"`.

socket_address - The address on which to connect to the PHP FPM process. It should look something like `"/var/run/php/php8.1-fpm.sock"`.

Tuning

To add the tuning section, add the line `[tuning]`, followed by any of the following options:

num_php_fpm_connections (optional) - The number of concurrent connections to the PHP FPM process for executing PHP commands. Setting this too low could result in slowdowns as threads have to wait for a free connection.

num_db_readers (optional) - Number of read connections to open to the database. More connections means less potential for slowdowns.

num_db_writers (optional) - Number of write connections to open to the database. More connections means less potential for slowdowns.

request_timeout (optional) - The default time before a connection to the server times out, in seconds. Note that some operations, like uploads, override this setting.

rate_limit_multiplier (optional) - A multiplier to apply to the rate limiter to increase or decrease it proportionally. If you find yourself hitting the rate limiter a lot, you could set it to 1.2 to increase all rate limits by 20%.

max_concurrent_requests (optional) - The maximum number of simultaneous requests before load shedding will start. After this point, users will receive an error message when trying to access the site. This is intended to avoid the site going down in times of high traffic or when under DDOS attack.

max_db_backups (optional) - The maximum number of database backups to keep. Defaults to 10.

Security

To add the security section, add the line [security], followed by any of the following options:

auto_ban_malicious_users (optional) - Set to false to disable automatic banning of users who visit certain non-existent URLs (such as /.env) to probe for vulnerabilities. Defaults to true.

Frequently Asked Questions

Q. Can I store media files on a different drive / server?

Yes, as long as the drive is mounted on the server. Specify the full path on the Configure page of the Admin Area.

Q. How do I get Digital Rebellion apps to communicate with Kollaborate Server?

- A. Digital Rebellion apps will have options in their preferences for specifying the URL to Kollaborate Server. Point this URL to the URL of the server, then login and they will communicate with the local server instead of the cloud.

Q: I updated the configuration incorrectly and locked myself out. How do I change it back?

- A. If you can't access the Admin Area to change the configuration, you can manually edit the configuration file **config.php** inside the **config** folder in your web directory.

Q: Which ports should I open / forward on my firewall?

- A. Ports 80 (unencrypted) and/or 443 (encrypted) should be open for traffic.

Q: How do I unregister the trial version so I can register my full license?

A: If you deleted the install folder (which is recommended for production installations), download Kollaborate Server again and copy the install folder back (do not copy any other folders).

Visit **[http://\[yoursite\]/install](http://[yoursite]/install)** and keep clicking **Continue** until you get to the Licensing page. Click the option to relicense the server, then login with your Kollaborate cloud email address and password, then you will be automatically licensed to the full version of Kollaborate Server.

Troubleshooting

Problem: Upon visiting the site you are presented with a blank white screen.

Solution: The most common cause of this is an incorrect license. Make sure you followed the steps of the Installation section correctly to generate and install your license. If you view your PHP log it should tell you the exact reason for the failure.

Problem: Emails don't arrive straight away.

Solution: The mail queue runs on a two minute timer, so emails will arrive at most two minutes late. This delay is doubled if using email routing.

Problem: Emails end up in recipients' spam boxes.

Solution: Do not send emails from a dynamic IP address. For best results, choose a fixed IP address that is linked to a domain and generate an SPF record for that domain.

Problem: Cut Notes sync / live comments don't work.

Solution: You may need to type the following:

```
cd *[my Kollaborate Server installation]*/websocket (e.g. cd /var/www/websocket)
```

```
npm install websocket.io
```

Then reboot the server.

Uploads fail or appear to work but no file exists on the server

Uploads not working can be one of three things:

1. The `max_upload_size_gb` parameter in `server_config.toml` may be set too low.
2. Permissions are preventing PHP from writing to your storage location. On Ubuntu the default PHP FPM user is `www-data` so you can type **`sudo chown -R www-data:www-data [path to storage directory]`**
3. You will have better results from a directly-attached drive (e.g. via USB) than a network volume.