ownCloud Quick Start

ownCloud allows users to securely sync files with a shared team space. This space is available on all platforms with real-time updates. For more information on the ownCloud offerings, see the ownCloud product page here.

Table of Contents

- Prerequisites - To install ownCloud, admins must have the necessary PHP extensions. This topic goes into what specific modules are needed to use ownCloud.
- Manually Installing ownCloud with the Linux Command Line - This document details the manual installation path for ownCloud. Installing with third-party options is highlighted in the Alternative Methods to Installing ownCloud.
- Admin & User Guide - This guide covers some administrative tasks as well as a high-level of the available user tasks.
- Troubleshooting and Support - This document covers some of the more troublesome aspects of installing and using ownCloud # Prerequisites for Installing ownCloud on Linux

The downloaded archive of the ownCloud server will contain all of the necessary dependencies for installation and use. This document will guide users through the process for installing the PHP extensions that ownCloud requires.

Note

This guide focuses on the required extensions for ownCloud. For additional but recommended extensions and prerequisites, see the Recommended Prerequisites section of the Prerequisites for Manual Installation on Linux document.

This Quick Start Guide assumes users are comfortable using a Linux command line. With Linux, the following command can be used to detect if any of the extensions are already installed. Replace `<module_name>` with the extension name to check.

```
php -m | grep -i <module_name>
```

List of Required Extensions

The following is a list of the required PHP extensions. The installed PHP version should be version 7.2, 7.3, or 7.4. ### PHP Extensions —-

- **Ctype**: For character type checking
- **cURL**: Used for aspects of HTTP user authentication
- **DOM**: For operating on XML documents through the DOM API
- **GD**: For creating and manipulating image files in a variety of different image formats, including GIF, PNG, JPEG, WBMP, and XPM.
- **HASH Message Digest Framework**: For working with message digests (hash).
- **iconv**: For working with the iconv character set conversion facility.
- **intl**: Increases language translation performance and fixes sorting of non-ASCII characters
- **JSON**: For working with the JSON data-interchange format.
- **libxml**: This is required for the DOM, libxml, SimpleXML, and XMLWriter extensions to work. It requires that libxml2, version 2.7.0 or higher, is installed.
- **Multibyte String**: For working with multibyte character encoding schemes.
- **OpenSSL**: For symmetric and asymmetric encryption and decryption, PBKDF2, PKCS7, PKCS12, X509 and other crypto operations.
- **PDO**: This is required for the pdo_msql function to work.
- **Phar**: For working with PHP Archives (.phar files).
- **POSIX**: For working with UNIX POSIX functionality.
- **SimpleXML**: For working with XML files as objects.
- **XMLWriter**: For generating streams or files of XML data.
- **Zip**: For reading and writing ZIP compressed archives and the files inside them.
- **Zlib**: For reading and writing gzip (.gz) compressed files.

Database Extensions

- **pdo_mysql**: For working with MySQL & MariaDB.
- **pgsql**: For working with PostgreSQL. It requires PostgreSQL 9.0 or above.

- **sqlite**: For working with SQLite. It requires SQLite 3 or above. This is, usually, not recommended for performance reasons.

Required For Specific Apps

- **ftp**: For working with FTP storage
- **sftp**: For working with SFTP storage
- **imap**: For IMAP integration
- **ldap**: For LDAP integration
- **smbclient**: For SMB/CIFS integration

Back to Home. # Manually Installing ownCloud with the Linux Command Line

This guide details how to manually install ownCloud. This guide assumes that users have the required system configurations and permissions as detailed on the **Prerequisites** topic. This process is a manual path using the command line. For other methods of installing ownCloud, see the **Alternative Installation Methods** topic.

1 - Prepare the Server

Using an Ubuntu 18.04 LTS Server, ensure that the Apache Web Server and PHP are installed with the following command:

```
sudo apt install php libapache2-mod-php apache2
```

Further considerations such as having multiple concurrent PHP versions as well as best practices can be found on the **Server Preparation for Ubuntu 18.04** document of the ownCloud Documentation.

2 - Install MySQL/MariaDB

While a few options are supported for the choice of databases, the recommended option is MariaDB. For other database install options, see the **Manual Installation Databases** guide.

For Ubuntu 18.04 LTS, add the MariaDB repository with the commands:

```
sudo apt-get install software-properties-common

sudo apt-key adv --fetch-keys 'https://mariadb.org/mariadb_release_signing_key.asc'

sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el] https://ftp.osuosl.org/pub/mariadb/1
```

Once completed, update the package manager and install MariaDB with:

```
sudo apt update

sudo apt install mariadb-server
```

For other Linux distributions and versions of MariaDB, see the **Setting up MariaDB Repositories** page on the MariaDB site.

The following is an example of a configured MySQL/MariaDB database with all possible parameters.

```
<?php
    $AUTOCONFIG = [
        "dbtype"        => "mysql",
        "dbname"        => "owncloud",
        "dbuser"        => "username",
        "dbpass"        => "password",
        "dbhost"        => "localhost:8080",
        "dbtableprefix" => "",
        "adminlogin"    => "root",
        "adminpass"     => "root-password",
        "directory"     => "/www/htdocs/owncloud/data",

];
```

IMPORTANT

The port of localhost, which is set on the `dbhost` property, must be available for traffic. In this example, `8080` is used, and this port is unblocked and able to send and receive traffic.

---

3 - Configure the Apache Web Server

To configure on Debian and related distributions, simply create a owncloud.conf file at /etc/apache/sites-available with the following commands

```
touch /etc/apache/sites-available/owncloud.conf
```

NOTE: This command assumes that the needed hierarchy is already created.

In this configuration file, add the following content:

```
Alias /owncloud "/var/www/owncloud/"

<Directory /var/www/owncloud/>

    Options +FollowSymlinks

     AllowOverride All

     <IfModule mod_dav.c>

         Dav off

     </IfModule>

</Directory>
```

In the command line, create a symlink to the sites-enabled directory with:

```
ln -s /etc/apache2/sites-available/owncloud.conf /etc/apache2/sites-enabled/owncloud.conf
```

For more information on optional Apache Configurations, see the Configure Apache for Manual Installation on Linux document.

4 - Downloading ownCloud

Before downloading the software, it is advised to change the command line directory to a temporary folder such as `/tmp`.

```
cd /tmp
```

Go to the **ownCloud download page** and download the latest `.tar` file and the desired security hash file like `md5` as well as the `asc` PGP signature file.

Once downloaded, run the md5sum command using the downloaded `md5` file with the `owncloud.tar.bz2` file. In the following commands, change the `yyyymmdd` value to match the year, month, and date of the downloaded files.

```
sudo md5sum -c owncloud-yyyymmdd.tar.bz2.md5 < owncloud-yyyymmdd.tar.bz2
```

If the sha256 has file was downloaded instead, use this command to verify the file.

```
sudo sha256sum -c owncloud-yyyymmdd.tar.bz2.sha256 < owncloud-yyyymmdd.tar.bz2
```

Next, verify the PGP signature with this command:

```
gpg --verify owncloud-complete-yyyymmdd.tar.bz2.asc owncloud-complete-yyyymmdd.tar.bz2
```

With the software downloaded and verified there are two paths to installation. The recommended path is using a script to install. This solution provides all the necessary permissions and component setups. To use the scripted installation, visit the **Script Guided Installation** document.

5 - Manual Installation with the Command Line

In the /tmp folder where the ownCloud download as well as the validation files are stored from the last step, extract the oWnCloud files with the following command:

```
tar -xjf owncloud-complete-yyyymmdd.tar.bz2
```

Ensure that the `-xjf` extension is included since the tar file type is `bz2`.

This extraction operation creates a single directory called `owncloud` containing the necessary files to run ownCloud. Copy this directory from the `/tmp` directory created in the previous step to the Apache web server document root. The default location for this root is `/var/www`, so the command is:

```
cp -r owncloud /var/www
```

Before completing the installation, ownership roles and user permissions must be set. It is advised to use the scripted method to set these. This process is documented on the **Script Guided Installation** document.

6 - Finalizing the Installation via Command Line

Continuing with the command line process, the following command will add all the necessary credentials for the deployment.

```
cd /var/www/owncloud/
sudo -u www-data php occ maintenance:install \
   --database "mysql" \
   --database-name "owncloud" \
   --database-user "root"\
   --database-pass "password" \
   --admin-user "admin" \
   --admin-pass "password"
```

With this step, ownCloud is installed and configured. There are aditional recommended configurations that users may wish to configure. For inforamtion on these items, see the **Post Installation Configuration** section of the Detailed Installation Guide document.

Back to Home. # Alternative Methods to Installing ownCloud The following methods are supported as well. These provide a bit less configurations, but each requires their own dependencies and processes.

Docker

Docker is a Product-as-a-service product that allows virtualization containers to run various applications. To use the Docker image, download it from the **Docker Hub ownCloud site**.

For a guide of all the configurations required for the Docker method, see the **Installing with Docker** document.

Linux Package Manager

With many package managers, ownCloud can be installed by using the `owncloud-files` package.

NOTE

This only installs the bare-minimum ownCloud files. Users will need to configure Apache, their databases, and fulfill the PHP dependencies. For more information on what is required for ownCloud, see the Prerequisites topic. For the commands for different

Back to Home. # Administrative and User Guide

Admins will be in charge of configuring the ownCloud server and creating users with the needed permissions. For the process to install and configure the ownCloud server see the **Manually Installing ownCloud with the Linux Command Line** topic.

Admin Task: Creating a User Account

As an Administrator user, from the 'Users' page, there will be a table showing all created users. At the top of this table there are three fields with a 'Create' button.

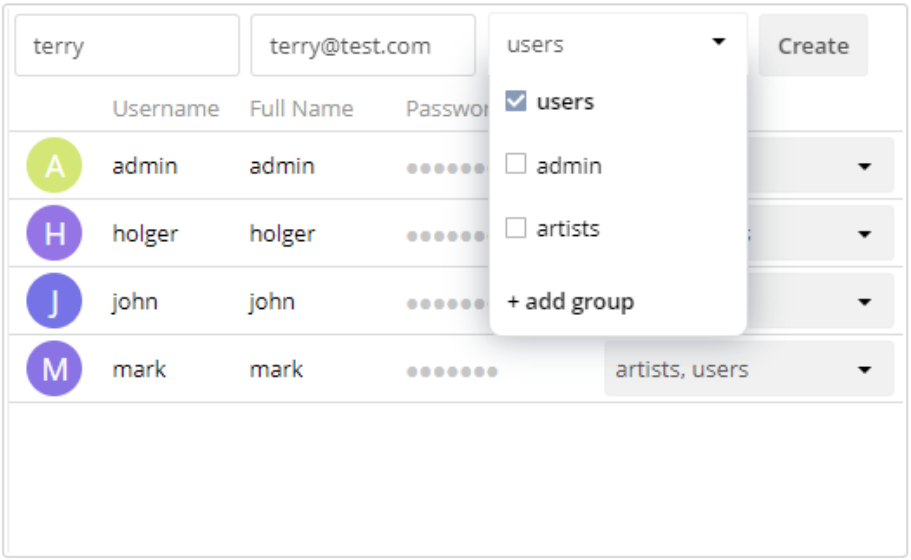

Figure 1: Options to Create a User

1. **Username** - This value will be the user name that your created user will need to login. The username may contain letters (a-z, A-Z), numbers (0-9), dashes (-), underscores (_), periods (.), and at signs (@).
2. **E-Mail** - The user's email where their generated first-use password will be sent.
3. **Groups** - This dropdown field allows the admin to move users to created groups. By default, users are unassigned.

When completed as desired, select the Create button to add the user. Admins may select additional fields to show on the table of users. For information on

these configurations, see the **User Configuration** section of the Admin Manual.

User Task: Connecting to ownCloud

ownCloud allows users to connect via a Desktop client, an Android app, or an iOS app.

For Desktop Users

Download the desktop application from **the ownCloud downloads page**.

Once downloaded, open the application and you will be prompted to enter your ownCloud server URL.

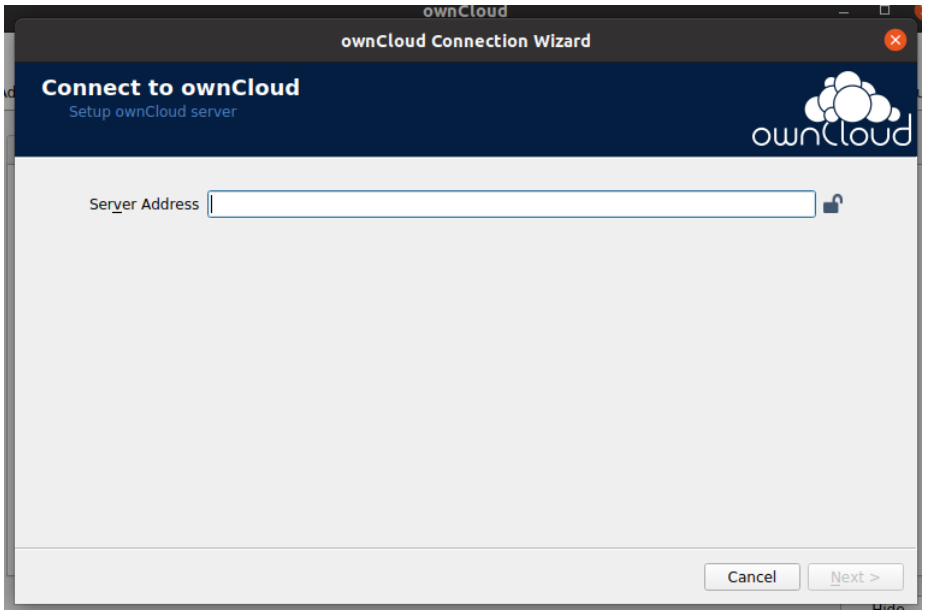

Figure 2: Ubuntu ownCloud app

Next, enter your credentials to log into the server.

After the correct credentials are entered, you may choose to connect files to ownCloud from a local drive, or simply skip this step and connect to the server.

Once approved, you are ready to use the ownCloud desktop client. For more information on using desktop client, see the **Using the Synchronization Client** topic in the Desktop Client manual.

For Android Users

Navigate to the Google Play Store and download the ownCloud app.

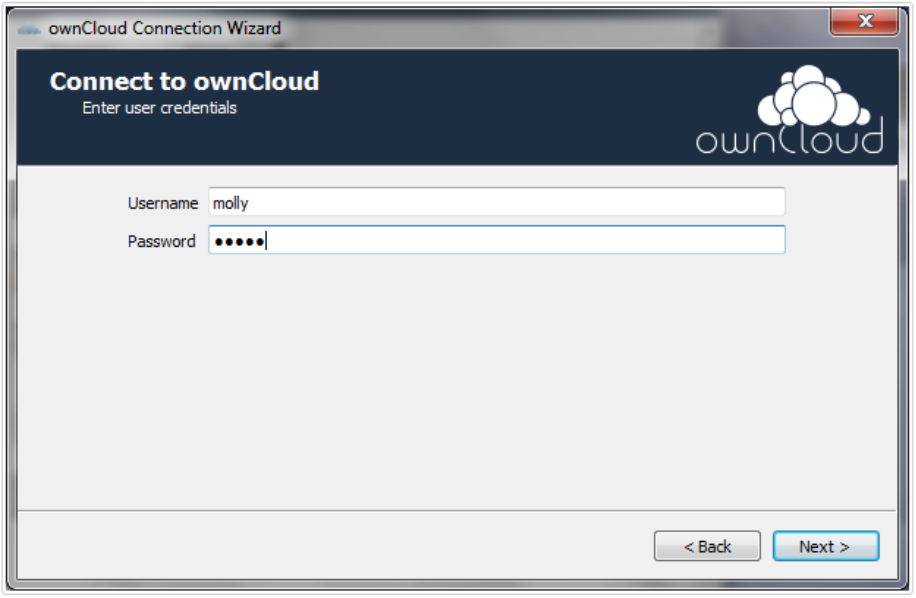

Figure 3: Ubuntu Login Information

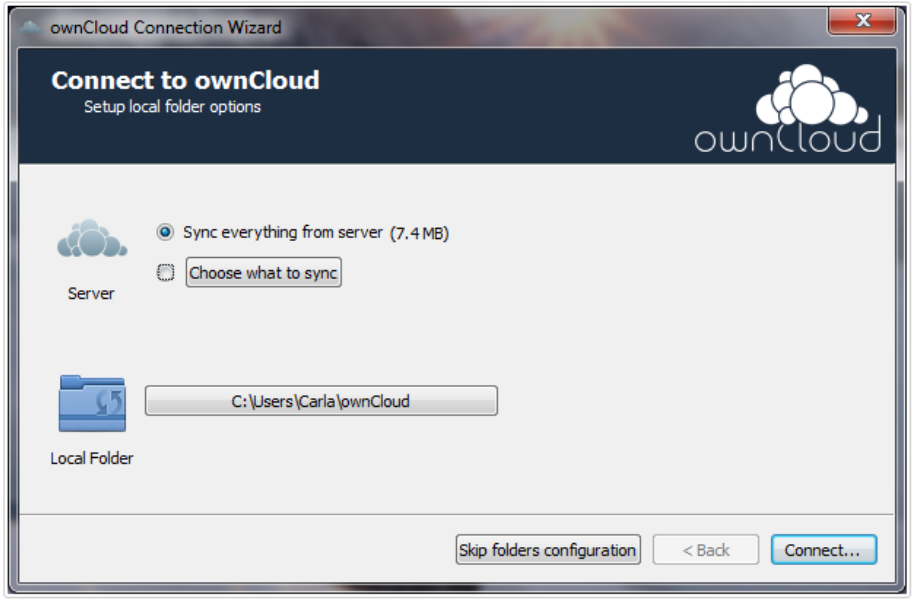

Figure 4: Choose what, if any, to sync

Open the app once it is installed. Enter your credentials into the login screen:

Figure 5: This information is available from your admin

Once logged in, if the ownCloud server is using self-signed SSL certificates, select 'Yes' when asked if you wish to trust this unverified certificate.

The app is ready for use. For more information on using your Android app, see the **Using the Android App** topic in the Mobile App for Android document.

For iOS Users

The iOS ownCloud application is available in the iOS store.

Once installed, enter your credentials in the login screen.

Once the user account is added and logged in, you are ready to use the ownCloud iOS app. For more information on gaining access to the ownCloud server with the iOS app, see the **Managing User Accounts** topic in the Mobile App for iOS document.

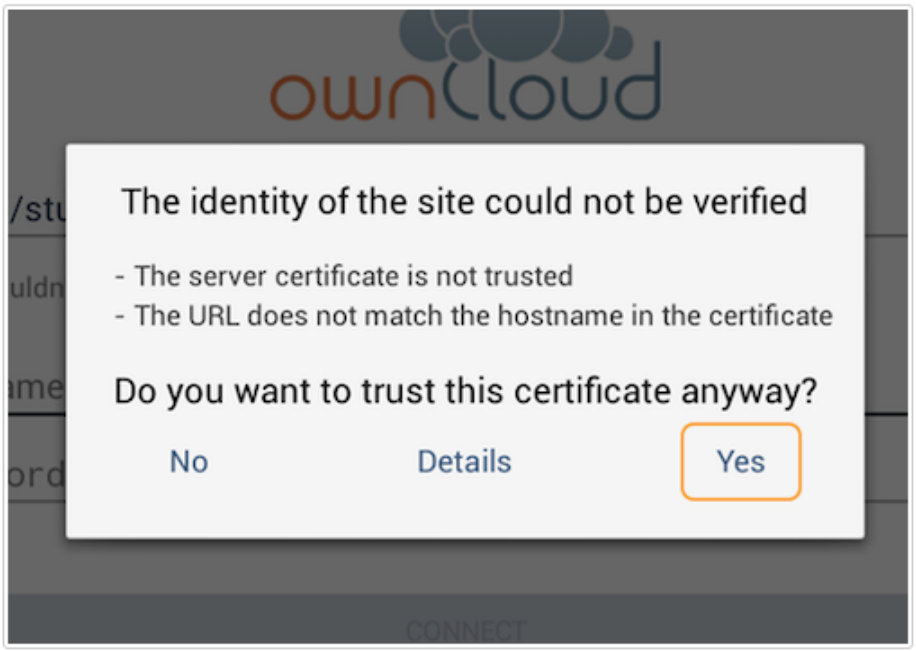

Figure 6: For users with self-signed SSLs

User Task: Basic User Actions

When using the ownCloud Web interface, users willl see the Files page. This is where the user may upload new files, remove files, and share any files from the respository. Users may also make changes based on how their permissions are set by the administrator.

For more information on using the Web Application, see the **Navigating the Main User Interface** section of the User Manual.

Back to Home. # Troubleshooting and Support

The following sections are some common errors that users and admins may experience while using ownCloud.

Installation Failure Error

The following error may occur if the installation fails:

```
An unhandled exception has been thrown: exception ‘PDOException’ with message 'SQLSTATE[HY0
```

To avoid data loss when many users are working in parallel, creating a large load on the system, ownCloud uses the `TRANSACTION_READ_COMMITTED` transaction in isolation. This requires that admins configure binary logging correctly when

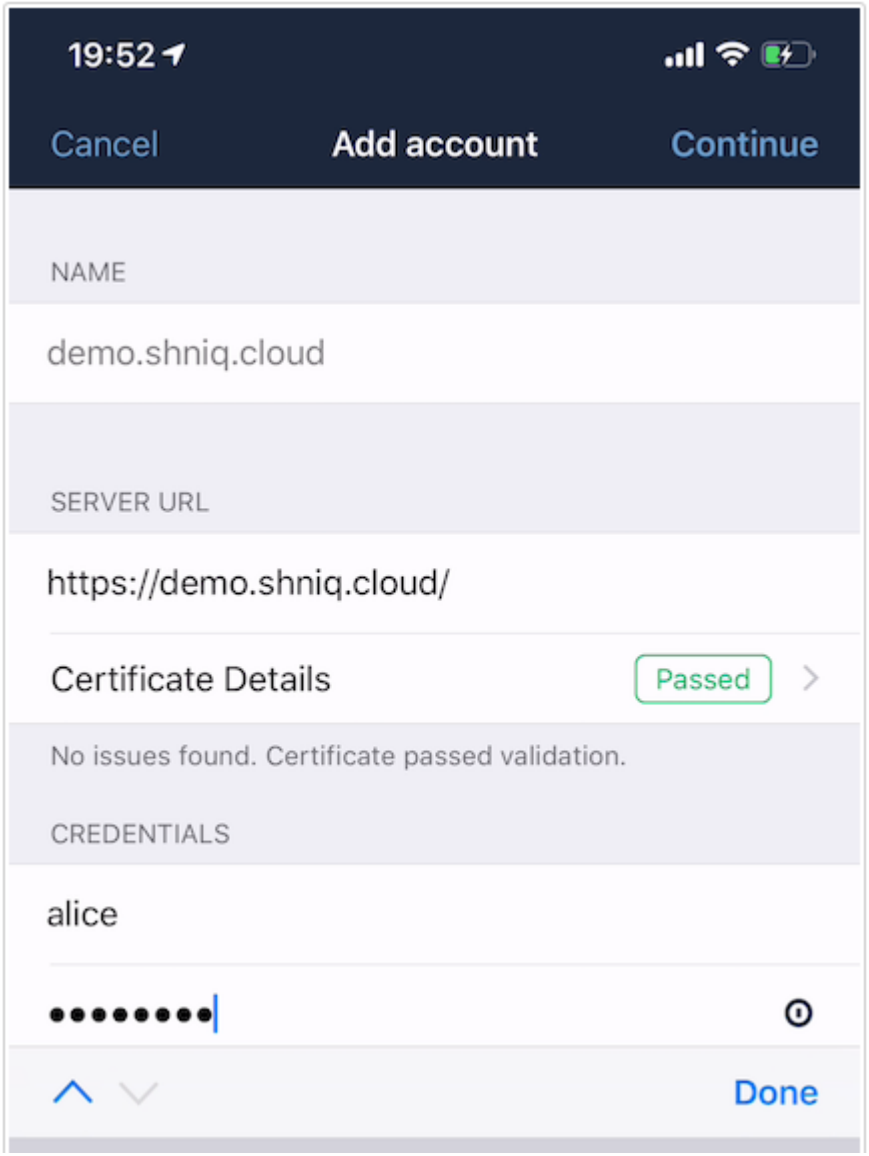

Figure 7: iOSLogin

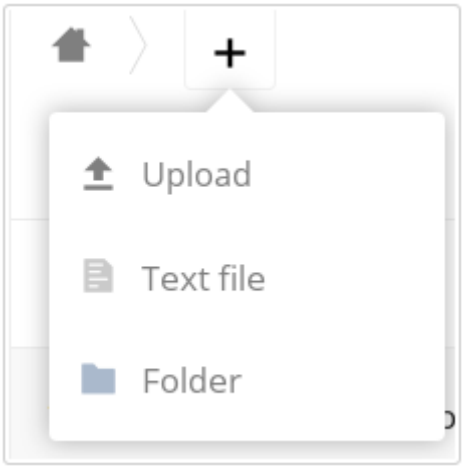

Figure 8: The Create button allows users to create a text file or a folder or upload a file

using MySQL or MariaDB, as this guide uses, or to disable it completely.

Binary logging records all the changes to the database and how long each change took to complete. This method enables replication and support backup operations. If admins wish to enable binary logging, then the following database change must be done.

In the database configuration file, change `BINLOG_FORMAT = STATEMENT` to either `BINLOG_FORMAT = MIXED` or `BINLOG_FORMAT = ROW`. These values may reside in the database startup script.

For more infromation on setting the `TRANSACTION_READ_COMMITTED` transaction isolation level, see the MySQL manual.

Log Files in the Desktop Application

Without a log file, any issues that arise will be difficult to troubleshoot. One solution to gaining log files is to create a temporary directory where log files are stored.

From the ownCloud client, press the `F12`, `Ctrl+L`, or `Cmd+L` buttons, depending on your operating system. The Log Output dialog will display.

On this dialog, select the 'Enable logging to temporary folder' checkbox. Now, to attain log files for the desktop client, click the 'Open Folder' to navigate to the directory with the log files.

Back to Home.

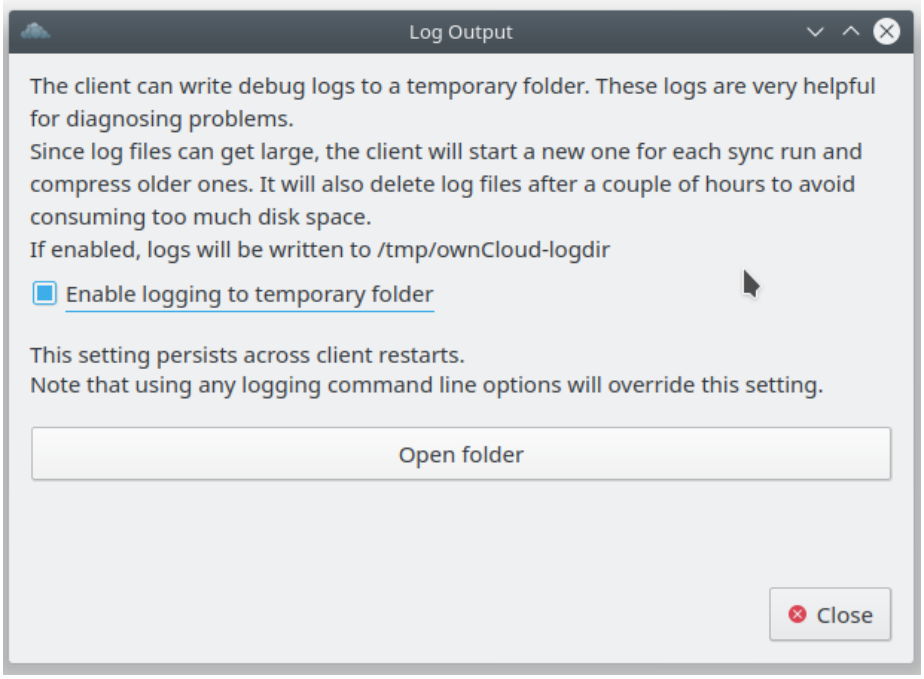

Figure 9: Log Output Dialog