# ownCloud Server Quick Start Guide

Katie Tothill

June 9, 2022

# Table of Contents

# Chapter 1. Installing ownCloud Server

Use the subsequent steps for a successful install of ownCloud Server in your on-premises environment.

## 1.1. System Requirements

The following minimum requirements must be met in order to install ownCloud Server:

| platform | options |
| --- | --- |
| operating system | Ubuntu 20.04 LTS |
| Database | MariaDB 10.5+ |
| Web server | Apache 2.4 with prefork and mod_php |
| PHP Runtime | 7.4 |
| openSSL | 1.1.x |

Table 1.1

| | |
| --- | --- |
| **IMPORTANT** | • MariaDB 10.6 is only supported with ownCloud release 10.9+. See Manual Installation Databases for more information.<br><br>• Some ciphers in version 1.x are no longer valid. You must verify the version regardless of the selected operating system. See openSSL Version for more information. |

## 1.2. Installation

This section describes the steps to install ownCloud on Ubuntu 20.04 LTS with SSH enabled and root privileges. Docker images and specific Linux Distribution packages are available at ownCloud Downloads.

### 1.2.1. Preparing Your System

1. Verify the `apt` installation and its required packages are installed and up to date on your system:

```
$ apt update && \
    apt upgrade -y
```

```
$ apt install -y \
    apache2 libapache2-mod-php \
    mariadb-server openssl redis-server wget php-imagick \
    php-common php-curl php-gd php-gmp php-bcmath php-imap \
    php-intl php-json php-mbstring php-mysql php-ssh2 php-xml \
```

```
    php-zip php-apcu php-redis php-ldap php-phpseclib
```

2. (Optional) Install additional tools helpful for debugging:

```
$ apt install -y \
    unzip bzip2 rsync curl jq \
    inetutils-ping  ldap-utils\
    smbclient
```

## 1.2.2. Configuring Apache

1. Update the Apache configuration and restart:

```
$ sed -i "s#html#owncloud#" /etc/apache2/sites-available/000-default.conf
service apache2 restart
```

2. Create and enable your ownCloud virtual host configuration:

```
FILE="/etc/apache2/sites-available/owncloud.conf"
cat <<EOM >$FILE
<VirtualHost *:80>
# uncommment the line below if variable was set
#ServerName $my_domain
DirectoryIndex index.php index.html
DocumentRoot /var/www/owncloud
<Directory /var/www/owncloud>
  Options +FollowSymlinks -Indexes
  AllowOverride All
  Require all granted

 <IfModule mod_dav.c>
  Dav off
 </IfModule>

 SetEnv HOME /var/www/owncloud
 SetEnv HTTP_HOME /var/www/owncloud
</Directory>
</VirtualHost>
EOM
```

```
$ a2dissite 000-default
```

```
$ a2ensite owncloud.conf
```

### 1.2.3. Configuring the Database

1. Start MySQL and grant privileges for ownCloud server:

```
$ service MySQL start
MySQL -u root -e "CREATE DATABASE IF NONE EXISTS owncloud; \
GRANT ALL PRIVILEGES ON owncloud.* \
   TO owncloud@localhost \
   IDENTIFIED BY 'password'";
```

2. Enable the requisite modules for your database and reload Apache:

```
$ a2enmod dir env headers mime rewrite setenvif
service apache2 reload
```

### 1.2.4. Installing ownCloud

1. Download and set up ownCloud. See Using the occ Command for a list of ownCloud Console (occ) commands.

```
$ cd /var/www/
```

```
$ wget https://download.owncloud.org/community/owncloud-10.6.0.tar.bz2 && #Download
ownCloud \
tar -xjf owncloud-10.6.0.tar.bz2 && #Extract ownCloud Archive \
```

```
$ chown -R www-data. owncloud
```

```
$ occ maintenance:install #ownCloud Installation \
    --database "MySQL" \
    --database-name "owncloud" \
    --database-user "owncloud" \
    --database-pass "password" \
    --admin-user "admin" \
    --admin-pass "admin"
myip=$(hostname -I|cut -f1 -d ' ') #ownCloud Trusted Domains Configuration \
```

```
$ occ config:system:set trusted_domains 1 --value="$myip"
```

| NOTE | To connect to ownCloud on a given port (e.g.: 8080), edit /var/www/owncloud/config/config.php. Users can then append the port number to |

the URL as `http://<your_owncloud_ip_address>:8080`.

> To use a host name in addition to an IP address, configure DNS for your ownCloud server domain and update the trusted domains to include this host name.

2. Configure Cron jobs to enable necessary file and user account synchronization:

```
$ occ background:cron

#Synchronize files between ownCloud server and authorized clients
echo "*/15  *  *  *  *  /var/www/owncloud/occ system:cron" \
  > /var/spool/cron/crontabs/www-data
chown www-data.crontab /var/spool/cron/crontabs/www-data
chmod 0600 /var/spool/cron/crontabs/www-data

#Synchronize users between LDAP or Active Directory Server and ownCloud server
echo "*/15  *  *  *  *  /var/www/owncloud/occ user:sync 'OCA\User_LDAP\User_Proxy'
-m disable -vvv >> /var/log/ldap-sync/user-sync.log 2>&1" \ >
/var/spool/cron/crontabs/www-data
chown www-data.crontab  /var/spool/cron/crontabs/www-data
chmod 0600  /var/spool/cron/crontabs/www-data
mkdir -p /var/log/ldap-sync
touch /var/log/ldap-sync/user-sync.log
chown www-data. /var/log/ldap-sync/user-sync.log
```

3. Configure caching for ownCloud and enable locking while transfers are in progress:

```
$ occ config:system:set #Set caching \
   memcache.local \
   --value '\OC\Memcache\APCu'
```

```
$ occ config:system:set \
   memcache.locking #Set file locking \
   --value '\OC\Memcache\Redis'
```

```
$ service redis-server start
```

```
$ occ config:system:set \
   redis \
   --value '{"host": "127.0.0.1", "port": "6379"}' \
   --type json
```

4. Configure log rotation for efficient management of large files:

```
FILE="/etc/logrotate.d/owncloud" #Set log rotation
sudo /bin/cat <<EOM >$FILE
/var/www/owncloud/data/owncloud.log {
  size 10M
  rotate 12
  copytruncate
  missingok
  compress
  compresscmd /bin/gzip
}
EOM
```

5. Take ownership of your ownCloud installation:

```
$ cd /var/www/
```

```
$ chown -R www-data. owncloud
```

# 1.3. Additional Resources

- Apache HTTP Server Version 2.4 Documentation
- MariaDB Repository Downloads
- PHP Runtime Downloads
- Installing ownCloud Server with Docker