# Cloud & Backup systems

## Overview

- [\*\*ownCloud - Your own personal cloud based backup/data storage system\*\*](#owncloud)

- [\*\*Nextcloud - Self-hosted productivity platform\*\*](#nextcloud)

- [\*\*Nextcloud Talk - Video calls via Nextcloud, including TURN server\*\*](#nextcloud-talk)

- [\*\*Pydio - Feature-rich backup and sync server with web interface\*\*](#pydio)

- [\*\*UrBackup Server - Full backups for systems on your network\*\*](#urbackup)

- [\*\*Gogs - GitHub style server, with web interface\*\*](#gogs)

- [\*\*Gitea - GitHub style server, with web interface\*\*](#gitea)

- [\*\*Syncthing - Backup and sync server with web interface\*\*](#syncthing)

- [\*\*MinIO - S3 compatible distributed object server\*\*](#minio)

- [\*\*Firefox Sync Server - Sync bookmarks, tabs, history and passwords\*\*](#firefox-sync-server)

- [\*\*vaultwarden - Unofficial Bitwarden password manager server written in Rust\*\*](#vaultwarden)

- [\*\*FuguHub - Your Own Personal Cloud Server\*\*](#fuguhub)

- [\*\*File Browser - Light web based file manager with sharing features\*\*](#file-browser)

??? info "How do I run \*\*DietPi-Software\*\* and install \*\*optimised software\*\* items?"

To install any of the \*\*DietPi optimised software items\*\* listed below run from the command line:

```sh

dietpi-software

```

Choose \*\*Browse Software\*\* and select one or more items. Finally select `Install`.

DietPi will do all the necessary steps to install and start these software items.

![DietPi-Software menu screenshot](../assets/images/dietpi-software.jpg){: width="643" height="365" loading="lazy"}

To see all the DietPi configurations options, review the [DietPi Tools](../../dietpi\_tools/) section.

[Return to the \*\*Optimised Software list\*\*](../../software/)

## ownCloud

The ownCloud package turns your DietPi system into your very own personal cloud based backup/data storage system (e.g.: Dropbox).

Also Installs:

- Webserver

- USB dedicated hard drive highly recommended

![ownCloud web interface screenshot](../assets/images/dietpi-software-cloud-owncloud.png){: width="400" height="218" loading="lazy"}

=== "Access to the web interface"

- URL = `http://<your.IP>/owncloud`

- Username = `admin`

- Password = `<your global password>` (default: `dietpi`)

If you may want to configure your ownCloud from command line via `occ` command see the [ownCloud admin manual](https://doc.owncloud.org/server/10.5/admin\_manual/configuration/server/occ\_command.html).

To simplify this configuration, DietPi has added a shortcut to the otherwise necessary `sudo -u www-data php /var/www/owncloud/occ`.

Just use inside your terminal:

```sh

occ list

```

=== "Update ownCloud to the latest version"

1. Option: Use the web-based updater from within the ownCloud web UI settings.

2. Option: Use the updater script from console (recommended):

```sh

sudo -u www-data php /var/www/owncloud/updater/application.php

1

```

3. Follow the official documentation for a manual upgrade process: <https://doc.owncloud.com/server/admin\_manual/maintenance/manual\_upgrade.html>

=== "FAQ"

\*\*Where is my data stored?\*\*

`/mnt/dietpi\_userdata/owncloud\_data` (or `dietpi.txt` choice)

\*\*Why am I limited to 2 GiB file size uploads?\*\*

DietPi will automatically apply the max supported upload size to the PHP and ownCloud configs.

- 32-bit systems can handle 2 GB

- 64-bit systems can handle 8796 PB, yep, in petabyte

- `echo -e "$(( $(php -r 'print(PHP\_INT\_MAX);') / 1024 / 1024))MB"`

\*\*Will my data be saved after deinstallation?\*\*

Your userdata directory will stay after deinstallation.

As well a database backup will be saved to your userdata directory. Thus you can easily restore your instance by reinstalling ownCloud and restore the database dump.

\*\*\*

Website: <https://owncloud.com>

Official documentation: <https://doc.owncloud.org/server/admin\_manual>

YouTube video tutorial: \*How to Install DietPi OwnCloud on Raspberry Pi\*.

<iframe src="https://www.youtube-nocookie.com/embed/-OatWtH1Z9c?rel=0" frameborder="0" allow="fullscreen" width="560" height="315" loading="lazy"></iframe>

## Nextcloud

Nextcloud gives you access to all your files wherever you are. Store your documents, calendar, contacts and photos on a server at home, at one of our providers or in a data center you trust.

![Nextcloud web interface screenshot](../assets/images/dietpi-software-cloud-nextcloud.jpg){: width="2048" height="1280" loading="lazy"}

=== "Quick start"

Nextcloud is accessible via regular HTTP/HTTPS port \*\*80\*\*/\*\*443\*\* below the `/nextcloud` path:

- URL = `http://<your.IP>/nextcloud/`

- Username = `admin`

- Password = `<your global password>` (default: `dietpi`)

To fast access the files, a dedicated USB hard drive is highly recommended.

=== "Advanced configuration"

For an advanced setup you could further configure your Nextcloud setup from the command line - see the [Nextcloud Admin guide](https://docs.nextcloud.com/server/latest/admin\_manual/configuration\_server/occ\_command.html).

To simplify this configuration, DietPi has added a shortcut to the otherwise necessary `sudo -u www-data php /var/www/nextcloud/occ`.

Just use inside your terminal:

```sh

ncc list

```

=== "Brute-force protection"

Nextcloud offers built-in brute force protection and additionally a plugin \*\*\*Brute-force settings\*\*\*.

This will delay your login rate in case of several failed login attempts.

This protection can be extended with Fail2Ban (see following tab).

See also:

- <https://apps.nextcloud.com/apps/bruteforcesettings>

- <https://docs.nextcloud.com/server/latest/admin\_manual/configuration\_server/bruteforce\_configuration.html>

=== "Fail2Ban integration"

Using Fail2Ban your can block users after failed login attempts. This hardens your system, e.g. against brute force attacks.

- Set options in the \*\*\*Nextcloud configuration file\*\*\* (typical `/var/www/nextcloud/config/config.php`):

- Add trusted domains if not already set via the `'trusted\_domains'` entry.

```ini

'trusted\_domains' =>

array (

0 => 'localhost',

1 => '<your\_trusted\_domain>',

),

```

The entry of the trusted domains is important, because one of the Fail2Ban regular expressions in the Fail2Ban filter file ("Trusted domain error", see below) deals with trusted domain login errors. By default, if you login via a non trusted domain, Nextcloud will show an error login dialog.

!!! attention

Take care, if you use this "Trusted domain error" `failregex` option and you then reload the page several times (more often than `maxretry` value in the Fail2Ban jail file) you lockout yourself also for logging in via a trusted domain from the IP address you are using.

- log file options: These are set to appropriate values by default (e.g. `log\_level`, `log\_type`) resp. DietPi defaults (`logfile` via `SOFTWARE\_NEXTCLOUD\_DATADIR` within `/boot/dietpi.txt`), so that they do not need to be set as sometimes otherwise described.

- Create new \*\*\*Fail2Ban filter\*\*\* (e.g. `/etc/fail2ban/filter.d/nextcloud.conf`):

```ini

# Fail2Ban filter for Nextcloud

[Definition]

\_groupsre = (?:(?:,?\s\*"\w+":(?:"[^"]+"|\w+))\*)

failregex = ^\{%(\_groupsre)s,?\s\*"remoteAddr":"<HOST>"%(\_groupsre)s,?\s\*"message":"Login failed:

^\{%(\_groupsre)s,?\s\*"remoteAddr":"<HOST>"%(\_groupsre)s,?\s\*"message":"Trusted domain error.

datepattern = ,?\s\*"time"\s\*:\s\*"%%Y-%%m-%%d[T ]%%H:%%M:%%S(%%z)?"

```

- Create new \*\*\*Fail2Ban jail file\*\*\* `/etc/fail2ban/jail.d/nextcloud.local`:

```ini

[nextcloud]

backend = auto

enabled = true

port = http,https

protocol = tcp

filter = nextcloud

maxretry = 5

bantime = 600

logpath = /mnt/dietpi\_userdata/nextcloud\_data/nextcloud.log

```

Check whether the `logpath` is identical to the value in the Nextcloud configuration file (`config.php`see above).

As not specified here, Fail2Ban uses properties like `maxretry`, `bantime`, etc. from `/etc/fail2ban/jail.conf` or `/etc/fail2ban/jail.local` (if present). Note the setting `backend = auto`. By default, `backend` is set to `systemd` in `/etc/fail2ban/jail.conf`. As a result, Fail2Ban ignores the `logpath` entry here in the jail `nextcloud.conf`, with the consequence, that Fail2Ban does not recognize an attack on Nextcloud (port 80, 443) even though attacks are listed in `/mnt/dietpi\_userdata/nextcloud\_data/nextcloud.log`.

- Restart Fail2Ban: `systemctl restart fail2ban`.

- Test your settings by trying to sign in multiple times from a remote PC with a wrong user or password. After `maxretry` attempts your IP must be banned for `bantime` seconds (DietPi does not respond anymore) as the default action by Fail2Ban is `route`, specified in `/etc/fail2ban/action.d/route.conf`.

- Check the current status on your DietPi with `fail2ban-client status nextcloud`.

- See also:

- [Fail2Ban](../system\_security/#fail2ban-protects-your-system-from-brute-force-attacks)

- <https://help.nextcloud.com/t/repeated-login-attempts-from-china/6510/11?u=michaing>

- <https://www.c-rieger.de/nextcloud-installationsanleitung/#c06>

=== "Update Nextcloud to the latest version"

1. Option: Use the web-based updater from within the Nextcloud web UI settings.

2. Option: Use the updater script from console (recommended):

```sh

phpenmod phar # The PHP Phar module is required

sudo -u www-data php /var/www/nextcloud/updater/updater.phar

y # Starts download and install of files

y # Starts the internal database upgrade and migration steps

N # Do not keep maintenance mode active

```

3. Follow the official documentation for a manual upgrade process: <https://docs.nextcloud.com/server/latest/admin\_manual/maintenance/manual\_upgrade.html>

=== "FAQ"

\*\*Where is my data stored?\*\*

`/mnt/dietpi\_userdata/nextcloud\_data` (or `dietpi.txt` choice)

\*\*Why am I limited to 2GB file size uploads?\*\*

DietPi will automatically apply the max supported upload size to the PHP and Nextcloud configs.

- 32bit systems can handle 2 GB

- 64bit systems can handle 8796 PB (petabytes)

\*\*Will my data be saved after deinstallation?\*\*

Your user data directory will stay after deinstallation. As well a database backup will be saved to your user data directory. Thus you can easily restore your instance by reinstalling Nextcloud and restore the database dump.

\*\*How can I check my OPcache status?\*\*

PHP uses a so called \*OPcache\* to store PHP scripts in optimised \*opcode\* format in RAM, which speeds up browser access as the raw PHP scripts do not need to be read from disk and parsed first.

The Nextcloud admin panel includes a check for sufficient OPcache settings and in case shows a recommendation for settings to apply. If such appears, we recommend to apply them with an own configuration file, e.g.:

```sh

echo -e '; Custom Nextcloud OPcache settings\n; priority=99\nopcache.memory\_consumption=128' > /etc/php/7.4/mods-available/custom.ini

phpenmod custom

```

You can watch the actual usage with the pre-installed [opcache-gui](https://github.com/amnuts/opcache-gui){:class="nospellcheck"} by @amnuts.

- URL: `http://<your.IP>/opcache.php`

![OPcache statistics page](../assets/images/OpCache\_php\_output.png){: width="700" height="976" loading="lazy"}

The dialog shows the cache status as well as the settings.

\*\*What shall I do about missing `bcmath`, `gmp` and `imagick` PHP modules?\*\*

After a fresh install via `dietpi-software`, Nextcloud shows a warning \*"This instance is missing some recommended PHP modules. For improved performance and better compatibility it is highly recommended to install them."\*, also described [here](https://dietpi.com/phpbb/viewtopic.php?p=27383#p27383). We propose to ignore them, `imagick` is not needed at all and a subject of discussion as of security issues, `bcmath` and `gmp` are required only if you want to use the WebAuthn passwordless authentication with Nextcloud.

If you must mute those warnings, you can install the modules manually:

```sh

sudo apt install php-bcmath php-gmp php-imagick libmagickcore-6.q16-6-extra

```

The `imagick` related warning can be solved alternatively by disabling the `Theming` app (in the Nextcloud apps settings), if no manual theming is done to the Nextcloud instance anyway.

\*\*\*

Website: <https://nextcloud.com/>

Official documentation: <https://docs.nextcloud.com/server/latest/admin\_manual/contents.html>

YouTube video tutorial #1: \*DietPi Nextcloud Setup on Raspberry Pi 3 B Plus\*.

<iframe src="https://www.youtube-nocookie.com/embed/Q3R2RqFSyE4?rel=0" frameborder="0" allow="fullscreen" width="560" height="315" loading="lazy"></iframe>

YouTube video tutorial #2: [DietPi Docker Nextcloud External Storage Setup with SAMBA SERVER on RPI3B](https://www.youtube.com/watch?v=NOb12BuNpZ8)

## Nextcloud Talk

Host video calls on your own Nextcloud instance. The TURN server \*\*\*Coturn\*\*\* will be installed and configured as well to allow reliable video calls through outside the local network, NAT and firewall setups.

Also installs:

- [\*\*Nextcloud\*\*](#nextcloud)

- \*\*Coturn\*\*

![Nextcloud Talk app screenshot](../assets/images/dietpi-software-cloud-nextcloudtalk.png){: width="2560" height="1440" loading="lazy"}

=== "Installation notes"

During installation you will be asked to enter the external server domain and a port, that you want to use for the Coturn TURN server. Note that you need to forward the chosen port and/or open it in your firewall.

If HTTPS was or is enabled via `dietpi-letsencrypt`, Coturn will be configured to use the LetsEncrypt certificates for TLS connections on the chosen TURN server port automatically.

Coturn by default will listen to non-TLS requests as well on the port configured in `/etc/turnserver.conf`. You can force TLS/control this by switching port forwarding in your router and/or opening/dropping ports in your firewall.

Coturn logging by default is disabled via `/etc/default/coturn` command arguments, since it is very verbose and produces much disk I/O. You can enable and configure logging via `/etc/turnserver.conf`, if required.

\*\*\*

Website: <https://nextcloud.com/talk>

## Pydio

Pydio is a feature-rich backup and sync server with web interface. Similar to ownCloud with vast configuration options to meet your "cloud" needs.

Also Installs:

- Webserver

![Pydio web interface screenshot](../assets/images/dietpi-software-cloud-pydio.png){: width="400" height="243" loading="lazy"}

=== "Access to the web interface"

URL = `http://<your.IP>/pydio`

=== "First time connect"

- Ignore the warnings and click the button titled `CLICK HERE TO CONTINUE TO PYDIO`.

Remark: If you require SSL access, please use LetsEncrypt to set this up.

- The wizard can now be started, click the `start wizard >` button to begin.

- Enter and create a new admin account for use with Pydio. Then click the `>>` button.

- Under database details, enter the following:

- Database type = `MySQL`

- Host = `localhost`

- Database = `pydio`

- User = `pydio`

- Password = `<your global password>` (default: `dietpi`)

- Use MySqli = No

- Click test connection, when successful, click the `>>` button.

- Under advanced options, use the default values, then click the `Install Pydio` button.

=== "Setup sync client on remote systems"

Once the server has been configured (as per above):

- Download the sync client for your system: <https://pydio.com/en/get-pydio/downloads/pydiosync-desktop-app>

- When configuring the remote server, use the following:

- Select HTTP option (unless you have setup an SSL cert)

- URL = `http://<your.IP>/pydio` (replace IP with your system IP)

- User = The "admin" user you setup in initial setup.

- Password = The "admin" password you setup in initial setup.

\*\*\*

Website: <https://pydio.com>

## UrBackup

UrBackup Server is an Open Source client/server backup system, that through a combination of image and file backups accomplishes both data safety and a fast restoration time.

Basically, it allows you to create a complete system backup, using a simple web interface, for systems on your network.

![UrBackup interface screenshot](../assets/images/dietpi-software-cloud-urbackup.png){: width="400" height="103" loading="lazy"}

=== "Access to the web interface"

The web interface is accessible via port \*\*55414\*\*:

URL = `http://<your.IP>:55414`

Remark: Change the IP address for your system.

=== "Backup storage location"

The location of the backups can be changed in the web interface:

- Select `Settings`.

- Change the Backup Storage Path: `/mnt/dietpi\_userdata/urbackup` is recommended.

- Click `Save`.

- Restart service with `systemctl restart urbackupsrv`.

=== "Download the client"

Install the appropriate client on the systems you wish to backup from <https://www.urbackup.org/download.html#client\_windows>.

\*\*\*

Website: <https://www.urbackup.org/index.html>

## Gogs

Your very own GitHub style server, with web interface.

![Gogs web interface screenshot](../assets/images/dietpi-software-cloud-gogs.png){: width="400" height="175" loading="lazy"}

=== "Access to the web interface"

The web interface is accessible via port \*\*3000\*\*:

- URL: `http://<your.IP>:3000`

=== "First run setup"

Has to be done once, when connected to the web interface:

- Change the following values only:

- Database Type: `MySQL`

- Host: `/run/mysqld/mysqld.sock`

- Password: `<your global password>` (default: `dietpi`)

- Repository Root Path: `/mnt/dietpi\_userdata/gogs-repo`

- Run User: `gogs`

- Domain: `<your.domain/IP>`

- Application URL: `http://<your.domain/IP>:3000/`

- Log Path: `/var/log/gogs`

- Scroll to the bottom of page and select "Install Gogs".

- Depending on whether the application URL above was entered correctly/is accessible by the connected browser, you may need to reconnect to the web page using the IP address, e.g.: `http://<your.IP>:3000`

- Once the page has reloaded, you will need to click `Register` to create the admin account.

=== "External access"

If you wish to allow external access to your Gogs server, you will need to setup port forwarding on your router, pointing to the IP address of your DietPi device.

- Port: 3000

- Protocol: TCP+UDP

If an external access is used, HTTPS is strongly recommended to increase your system security. You can get a free certificate e.g. via [dietpi-letsencrypt](../../dietpi\_tools/#dietpi-letsencrypt){:class="nospellcheck"}.

=== "View logs"

Initial service logs can be viewed via:

```sh

journalctl -u gogs

```

Daemon logs can be found in the following directory:

```

/var/log/gogs

```

=== "Update to latest version"

You can easily update Gogs by reinstalling it. Your settings and data are preserved by this:

```sh

dietpi-software reinstall 49

```

\*\*\*

Official website: <https://gogs.io/>

Official documentation: <https://gogs.io/docs>

Source code: <https://github.com/gogs/gogs>

License: [MIT](https://github.com/gogs/gogs/blob/main/LICENSE)

## Gitea

Your very own GitHub style server, with web interface.

![Gitea logo](../assets/images/dietpi-software-cloud-gitea.jpg){: width="320" height="200" loading="lazy"}

=== "Access to the web interface"

The web interface is accessible via port \*\*3000\*\*:

- URL: `http://<your.IP>:3000`

=== "First run setup"

Has to be done once, when connected to the web interface:

- Change the following values only:

- Host: `/run/mysqld/mysqld.sock`

- Password: `<your global password>` (default: `dietpi`)

- SSH Server Domain: `<your.domain/IP>`

- Gitea Base URL: `http://<your.domain/IP>:3000/`

- Log Path: `/var/log/gitea` (However, file logging is disabled by default.)

- Scroll to the bottom of page and select "Install Gitea".

- Depending on whether the base URL above was entered correctly/is accessible by the connected browser, you may need to reconnect to the web page using the IP address, e.g.: `http://<your.IP>:3000`

- Once the page has reloaded, you will need to click `Register` to create the admin account.

=== "External access"

If you wish to allow external access to your Gitea server, you will need to setup port forwarding on your router, pointing to the IP address of your DietPi device.

- Port: 3000

- Protocol: TCP+UDP

If an external access is used, HTTPS is strongly recommended to increase your system security. You can get a free certificate e.g. via [dietpi-letsencrypt](../../dietpi\_tools/#dietpi-letsencrypt){:class="nospellcheck"}.

=== "Fail2Ban integration"

Using Fail2Ban your can block IP addresses after failed login attempts. This hardens your system against e.g. brute-force attacks.

=== "When using journal logging (default)"

By default Gitea logs to the systemd journal (see "View logs" tab), in which case Fail2Ban protection can be enabled with the following steps:

Create a new filter `/etc/fail2ban/filter.d/gitea.conf`:

```ini

[Definition]

journalmatch = \_SYSTEMD\_UNIT=gitea.service

failregex = Failed authentication attempt for \x1b\[1m.+\x1b\[0m from \x1b\[1m<HOST>:\d+\x1b\[0m:

```

Create a new jail `/etc/fail2ban/jail.d/gitea.conf`:

```ini

[gitea]

enabled = true

backend = systemd

```

=== "When using file logging"

When file logging is enabled (see "View logs" tab), Fail2Ban protection can be enabled with the following steps:

Create a new filter `/etc/fail2ban/filter.d/gitea.conf`:

```ini

[Definition]

failregex = Failed authentication attempt for .+ from <HOST>:\d+:

```

Create a new jail `/etc/fail2ban/jail.d/gitea.conf`:

```ini

[gitea]

enabled = true

backend = auto

logpath = /var/log/gitea/gitea.log

```

You can specify other properties like `maxretry` or `bantime` to overwrite the defaults in `/etc/fail2ban/jail.conf` or `/etc/fail2ban/jail.local` if present. When done:

- Restart Fail2Ban: `systemctl restart fail2ban`

- Try to login to the Gitea web interface with a wrong password.

- Check whether the failed login has been detected: `fail2ban-client status gitea`

- When you further try to login `maxretry` times, your IP should be banned for `bantime` seconds, so that neither the Gitea web interface, nor SSH or any other network application will respond to requests from your client. When Fail2Ban was installed via `dietpi-software`, by default `route`/`blackhole` blocking is used, so that `ip r` on the server should show a `blackhole` route for your client's IP.

- See also:

- [Fail2Ban](../system\_security/#fail2ban)

- <https://docs.gitea.io/en-us/fail2ban-setup/>

=== "View logs"

By default, logs can be viewed with the following command:

```sh

journalctl -u gitea

```

File logging to `/var/log/gitea/gitea.log` can be enabled by editing `/mnt/dietpi\_userdata/gitea/custom/conf/app.ini` and changing `MODE = console` in the `[log]` section to `MODE = file`.

=== "Update to latest version"

You can easily update Gitea by reinstalling it. Your settings and data are preserved by this:

```sh

dietpi-software reinstall 165

```

\*\*\*

Official website: <https://gitea.io/>

Official documentation: <https://docs.gitea.io/>

Official forum: <https://discourse.gitea.io/>

Source code: <https://github.com/go-gitea/gitea>

License: [MIT](https://github.com/go-gitea/gitea/blob/main/LICENSE)

## Syncthing

Backup and sync server with web interface. Extremely lightweight and efficient as no webserver is required.

![Syncthing interface screenshot](../assets/images/dietpi-software-cloud-syncthing.png){: width="400" height="195" loading="lazy"}

=== "Access to the web interface"

The web interface is accessible via port \*\*8384\*\*:

URL = `http://<your.IP>:8384`

=== "First run setup"

Has to be done once, when connected to the web interface.

- When the `Danger! Please set a GUI Authentication User and Password in the Settings dialog.` box appears, click the `settings` button inside the box.

- Under `GUI Authentication User` and `GUI Authentication Password`: Enter new login details you would like to use for access to the web interface. Then click `save`.

DietPi will automatically setup your default folder share to `/mnt/dietpi\_userdata`.

=== "Setup a second system to sync with"

In this example we will use a Windows system. The goal is to "sync" the user data from your DietPi device with another system.

- Download, extract and run the Windows application `syncthing.exe`: <https://syncthing.net/downloads/>.

- Syncthing web interface will load automatically, if not, you can access it via `http://127.0.0.1:8384/`.

- Click `Actions` at the top right, then select `Show ID`. Copy the UUID code.

- On the DietPi device, open the web interface and click `Add remote device` (bottom right).

- Under `Device ID`, paste in the UUID we copied earlier.

- Under `Device name`, enter any name. e.g.: \*My Windows PC\*.

- Under `Share Folders With Device` tick/select DietPi user data, then click `save`.

- After a few seconds, go back to the Windows web interface `http://127.0.0.1:8384/`. You should receive a message asking you to confirm the new device, click `Add Device`.

- Under `Share Folders With Device` tick/select DietPi user data, then click `save`.

You devices should now duplicate the user data from your DietPi device to your Windows PC.

\*\*\*

Website: <https://syncthing.net>

## MinIO

It is an open source Kubernetes Native, High Performance Object Storage (S3 Compatible). It helps building cloud-native data infrastructure for machine learning, analytics and application data workloads.

![MinIO setup diagram](../assets/images/dietpi-software-cloud-minio.jpg){: width="417" height="443" loading="lazy"}

=== "Quick start"

The web interface is accessible via port \*\*9000\*\*:

- URL: `http://<your.IP>:9000`

- Username: `adminadmin`

- Password: `adminadmin`

- [MinIO Server Quick Start Guide](https://docs.min.io/docs/minio-quickstart-guide.html)

- [Python Client Quick Start Guide - MinIO](https://docs.min.io/docs/python-client-quickstart-guide.html)

- [JavaScript Client Quick Start Guide - MinIO](https://docs.min.io/docs/javascript-client-quickstart-guide.html)

\*\*\*

Website: <https://min.io/product/overview>

Official documentation: <https://docs.min.io>

## Firefox Sync Server

This is Mozilla's Firefox Sync Server which manages syncing Firefox instance bookmarks, history, tabs and passwords across devices. Out of the box it runs on a Python server for small loads and can be configured to run behind Nginx or Apache.

![Firefox Sync logo](../assets/images/dietpi-software-cloud-firefoxsyncserver.png){: width="300" height="95" loading="lazy"}

=== "Configure Firefox"

- Open `about:config` to access advanced settings.

- Search for: `identity.sync.tokenserver.uri`.

- Set value to: `http://<your.IP>:5002/token/1.0/sync/1.5`.

- We recommend to access your Firefox Sync Server only from local network or via VPN, keeping the default listening port \*\*5002\*\* closed for access from outside of your LAN.

- If you need to access it remotely without VPN, adjust the `public\_url` setting inside the config file `/mnt/dietpi\_userdata/firefox-sync/syncserver.ini` to contain your public IP or domain and desired port.

=== "Directories"

- Install directory: `/opt/firefox-sync`

- Data directory: `/mnt/dietpi\_userdata/firefox-sync`

- Config file: `/mnt/dietpi\_userdata/firefox-sync/syncserver.ini`

=== "View logs"

View the logs by executing:

```sh

journalctl -u firefox-sync

```

=== "Update to latest version"

You can easily update the Firefox Sync Server by reinstalling it. Your settings and data are preserved by this:

```sh

dietpi-software reinstall 177

```

\*\*\*

Official documentation: <https://mozilla-services.readthedocs.io/en/latest/howtos/run-sync-1.5.html>

Source code: <https://github.com/mozilla-services/syncserver>

License: [MPL2.0](https://github.com/mozilla-services/syncserver/blob/master/LICENSE)

Credits: This software title has been added to DietPi-Software by [CedArctic](https://github.com/CedArctic), many thanks! :D

## vaultwarden

vaultwarden is an unofficial Bitwarden password manager server with web interface, written in Rust.

![vaultwarden web vault screenshot](../assets/images/dietpi-software-vaultwarden.jpg){: width="600" height="247" loading="lazy"}

=== "First access"

- During install, a self-signed 4096-bit RSA TLS certificate is created to allow encrypted HTTPS access, which is required for access with most Bitwarden clients and reasonable as of the sensitivity of the data a password manager handles.

- Most web browsers will warn you on access that the certificate is not trusted, although usually you can choose to ignore that and still access the web vault.

- Most Bitwarden clients on the other hand will deny to access your server, as long as the certificate is not trusted.

- As far as you have a public domain name for your DietPi server, we recommend to request an official trusted CA certificate, e.g. via `dietpi-letsencrypt` and setup either a reverse proxy, or configure vaultwarden to use the retrieved key and certificate directly via ROCKET\_TLS setting in the config file (see "Directories" tab).

??? info "How do I add a self-signed certificate to the OS' Trusted Root Certification Authorities store?"

=== "Windows 10"

1. In your browser, next to the address bar, select the warning or lock icon.

Then select the certificate button to open Windows' Certificate view.

2. Switch to the "Details" tab.

![Import certificate on Windows 10, screenshot 1](../assets/images/import\_cert\_windows\_1.png)

3. Select "Save to file".

4. In the newly opened window, select "Continue".

![Import certificate on Windows 10, screenshot 2](../assets/images/import\_cert\_windows\_2.png)

5. Leave default DER coding and select "Continue".

6. Select "Browse" to chose a target file location.

![Import certificate on Windows 10, screenshot 3](../assets/images/import\_cert\_windows\_3.png)

7. Choose a target file location and name, it is only required temporarily.

8. Select "Continue".

9. Select "Finish".

![Import certificate on Windows 10, screenshot 4](../assets/images/import\_cert\_windows\_4.png)

10. Double-click the created certificate file and select "Install certificate".

11. Select "Local system".

12. Select "Continue", which requires administrator permissions.

![Import certificate on Windows 10, screenshot 5](../assets/images/import\_cert\_windows\_5.png)

13. Choose "Save all certificates to the following store".

14. Select "Browse".

15. Select "Trusted Root Certification Authorities".

16. Select "Ok".

17. Select "Continue".

18. Select "Finish".

![Import certificate on Windows 10, screenshot 6](../assets/images/import\_cert\_windows\_6.png)

=== "macOS"

1. In your browser (note that this cannot be done in Safari), next to the address bar, select the warning or lock icon.

Then select the "Certificate (Invalid)" button.

![Import certificate on macOS, screenshot 1](../assets/images/import\_cert\_mac\_1.png){: width="250px"}

2. Drag the certificate icon to your desktop, it is only required temporarily.

3. Double-click on the certificate file.

![Import certificate on macOS, screenshot 2](../assets/images/import\_cert\_mac\_2.png)

4. On the "Keychain" dropdown, select "System".

5. Select "Add".

6. Enter an administrator username and password.

7. Select "Modify Keychain".

![Import certificate on macOS, screenshot 3](../assets/images/import\_cert\_mac\_3.png)

8. Double-click on the certificate in the list.

9. Select "Trust".

![Import certificate on macOS, screenshot 4](../assets/images/import\_cert\_mac\_4.png)

10. On the "Secure Sockets Layer (SSL)" dropdown, select "Always Trust".

11. Click the red button in the top left corner of the window.

12. Enter an administrator username and password.

13. Select "Update Settings".

![Import certificate on macOS, screenshot 5](../assets/images/import\_cert\_mac\_5.png)

=== "Web access"

The web interface is accessible via port \*\*8001\*\*:

- URL = `https://<your.IP>:8001`

- On first access, you need to create an account, either via web UI or via client (see "Client access" tab).

=== "Client access"

Any official Bitwarden client will work: <https://bitwarden.com/download>

1. Select the settings cog at the top left of the window.

2. Add `https://<your.IP>:8001` into the custom server field.

3. Create a new account, which will be created on your own server only.

=== "Directories"

- Install directory: `/opt/vaultwarden`

- Data directory: `/mnt/dietpi\_userdata/vaultwarden`

- Config file: `/mnt/dietpi\_userdata/vaultwarden/vaultwarden.env`

=== "View logs"

```sh

journalctl -u vaultwarden

```

=== "Update to latest version"

```sh

dietpi-software reinstall 183

```

\*\*\*

Official documentation: <https://github.com/dani-garcia/vaultwarden/wiki>

Forum: <https://vaultwarden.discourse.group>

Source code: <https://github.com/dani-garcia/vaultwarden>

Open-source license: [GPLv3](https://github.com/dani-garcia/vaultwarden/blob/master/LICENSE.txt)

Credits: This software title has been added to DietPi-Software by @CactiChameleon9. Thank you!

## FuguHub

FuguHub transforms your DietPi device into a secure online storage system, letting you access and share files from any connected computer or device.

![FuguHub logo](../assets/images/dietpi-software-cloud-fuguhub.png){: width="149" height="140" loading="lazy"}

=== "Web interface"

The web interface is accessible via regular HTTP and HTTPS ports \*\*80\*\* and \*\*443\*\*:

- URL: `http://<your.IP>` or `https://<your.IP>` (When using HTTPS, you may ignore the browser warning because of the self-signed certificate which is used by default.)

- Username: `dietpi`

- Password: `<your global password>` (default: `dietpi`)

!!! warning "The fact that FuguHub runs on ports 80 and 443 by default makes it incompatible with regular web servers when using the default setup."

![FuguHub web interface screenshot](https://user-images.githubusercontent.com/28480705/99921345-12aaec80-2d2a-11eb-8503-1687b4997db1.png){: width="1920" height="1088" loading="lazy"}

=== "Setup details"

- Install directory: `/home/bd`

- Config file: `/home/bd/bdd.conf`

- File server directory: `/mnt/dietpi\_userdata/fuguhub-data`

=== "Logs"

- Service: `journalctl -u bdd`

- Trace: `/home/bd/trace/`

It contains an info about the database creation only, even after playing around with the web UI a bit.

=== "Update"

To update FuguHub, simply reinstall it:

```sh

dietpi-software reinstall 161

```

All your settings and data will be preserved.

\*\*\*

Official website: <https://fuguhub.com/>

## File Browser

Access and manage your data from anywhere via browser with this lightweight remote file manager. Other than ownCloud and Nextcloud, it accesses the raw data on your filesystem, based on a chosen root directory, which makes it similar to Syncthing. You can setup multiple users with their own root directory and also sharing files and directories via password-protected link is possible.

![File Browser logo](../assets/images/dietpi-software-cloud-filebrowser.svg){: width="150" height="150" loading="lazy"}

=== "Access to the web interface"

The web interface is accessible via port \*\*8084\*\*:

- URL = `http://<your.IP>:8084`

- Username = `dietpi`

- Password = `<your global password>`

=== "Directories"

- Install directory: `/opt/filebrowser`

- Config directory: `/mnt/dietpi\_userdata/filebrowser`

- Default data directory: `/mnt`

=== "View logs"

View the logs by executing:

```sh

journalctl -u filebrowser

```

=== "Update to latest version"

You can easily update File Browser by reinstalling it. Your settings and data are preserved by this:

```sh

dietpi-software reinstall 198

```

\*\*\*

Official documentation: <https://filebrowser.org/>

Source code: <https://github.com/filebrowser/filebrowser>

License: [Apache 2.0](https://github.com/filebrowser/filebrowser/blob/master/LICENSE)

[Return to the \*\*Optimised Software list\*\*](../../software/)