Cloudron Docs

The Cloudron Docs as a PDF

None

Table of contents

1. Welcome to Cloudron!

1.1 What is Cloudron?

Cloudron is a platform that makes it easy to install, manage and secure web apps on your server.

You can install Cloudron on your server (from say, AWS, Digital Ocean etc.), give it a domain name and start installing apps. Behind the scenes, Cloudron automates all the tasks around installation like configuring databases, DNS setup and certificate management.

Cloudron provides a centralized way to manage users and specify which apps they can access.

Cloudron has a backup solution that lets you back up and restore each app individually (compared to server snapshots). With Cloudron backups, you can even migrate your Cloudron in its entirety from one server provider to another.

1.2 The Cloudron App Store

The Cloudron App Store provides a mechanism for distribution and continuous update of apps. A good analogy for this is the Apple App Store for iOS or Google Play for Android. Anyone today can easily install apps on their phones and the apps are kept up-to-date. Cloudron does the same, but for servers. You can easily install apps and receive continuous updates for the apps via the Cloudron App Store.

Your browser does not support the video tag.

1.3 Motivation

Say you want to run a web application like WordPress, GitLab, Rocket.chat or even an email server. The first step is to start reading up installation manuals and configuring the server. Web apps today use multiple package managers, languages and frameworks, making this process very tedious and complicated.

Once the software is installed, DNS and SSL certificates have to be set up. If the server hosts multiple apps, one needs to make sure that apps do not interfere with each other, set up a reverse proxy and configure the firewall.

Installation is just one hurdle, though. The server and apps must be secured and backed up properly. Upstream releases must be tracked, and updates must be applied on time.

As you can see, self-hosting web applications is error-prone and time-consuming. 1-click installers and docker files automate some of the above tasks, but requires one to have the technical know how to complete the installation, and put in the effort to keep it up-to-date.

We, at Cloudron, want to fix just that!

2. Installation

2.1 Install

2.1.1 Install

Run the following commands on a fresh Ubuntu Noble 24.04 (x64) server:

```
wget https://cloudron.io/cloudron-setup
chmod +x cloudron-setup
./cloudron-setup
```

You can find referral links to get started on various cloud providers with free credits here.

Minimum Requirements

Cloudron requires at least 2GB RAM, 20GB Disk space. Make sure the firewall does not block port 80 (http) and 443 (https). Cloudron does not support running on ARM, LXC, Docker or OpenVZ (Open Virtuozzo).

Marketplace

Cloudron is available pre-installed as on various marketplaces - AWS, DigitalOcean, Hostinger, Linode, Time4VPS and Vultr.

Network requirements

Depending on your network, please read the Home Server or the Intranet installation considerations.

2.1.2 Setup

Once installation is complete, navigate to `https://<IP>` in your browser and accept the self-signed certificate.

In Chrome, you can accept the self-signed certificate by clicking on `Advanced` and then click `Proceed to <ip> (unsafe)`.

In Firefox, click on `Advanced`, then `Accept the Risk and Continue`.

2.1.3 Domain Setup

Provide a domain like `example.com`. The way Cloudron works is that the dashboard gets installed at `my.example.com`, and apps are installed under subdomains that you specify like `git.example.com`, `chat.example.com`, and so on.

It is perfectly safe to use a domain that is already in use as long as the `my` subdomain is available. When installing apps, Cloudron will never overwrite your existing DNS records and your existing subdomains will remain intact. It is also possible to use a subdomain like `cloudron.example.com`.

Next, select the DNS service in which the domain in hosted. If your service is not listed below, use the `Wildcard` or `Manual` option. See DNS Providers for the various providers and options.

Primary domain

The first domain added on Cloudron is called the `Primary Domain`. The dashboard is made available under the `my` subdomain of the primary domain. More domains can be added after installation in the in the Domains view. The Primary Domain can be changed post installation.

2.1.4 Admin Account

Once DNS is setup, Cloudron will redirect to `https://my.example.com`. The browser address bar will show a green lock to indicate that the connection to your Cloudron is now secure.

Enter the adminstrator username, email and password for Cloudron.

The email address is used for password resets and notifications. It is local to your Cloudron and not sent anywhere (including cloudron.io).

⚠ **Let's Encrypt requires a valid admin email**

Cloudron sets up a Let's Encrypt account with the administrator's email. If this email address is not valid, Let's Encrypt will not issue certificates and Cloudron will fall back to using self-signed certs.

2.1.5 App Store Account

You are now ready to start installing apps! When you click on the `Cloudron.io Account` menu in the UI, you will able to create a cloudron.io account by clicking `Set up account` button. This account is used to manage your subscription and billing.

2.1.6 Firewall Setup

Security is a core feature of Cloudron and the default installation is already setup to follow best practices. We do not recommend adding and modifying rules in `iptables` / `ip6tables` since Cloudron already does this. All unneeded ports are blocked. Ports are whitelisted as and when the apps you install require them.

To further harden security, we recommend:

- configuring the VPS Firewall
- securing SSH access

2.1.7 Email Setup

Given the proliferation of Email spam, many email providers block emails from VPS and Home network IPs. To ensure reliable delivery of email from apps and Cloudron, check the Email delivery status and possibly set up an Email relay.

Note that as part of app packaging, all apps are pre-configured to use Cloudron's internal mail server. You only need to configure the relay in the Cloudron dashboard and all apps will automatically use the relay.

2.2 Home Server

2.2.1 Prerequisites

Cloudron can be installed in a home network as long as the following prerequisites are met.

Public IPv4 / IPv6

If you require Cloudron to be accessible from outside your home, you need a public IPv4 or IPv6 address. This IP address does not need to be static. Post installation, you can use the Dynamic DNS feature to keep your DNS automatically up-to-date. You can visit this site to view your current public IP address.

If you do not require Cloudron to be accessible from outside your home, click on `Advanced Settings` in the Domain Setup UI. Then, choose `Static IP` and provide the internal IP of your server. If you decide to do this, you **must** use a Programmatic DNS provider (see below). With an internal IP and no programmatic DNS, Cloudron will not be able to get certificates from Let's Encrypt.

DNS Provider

Cloudron supports a variety of DNS providers to automatically configure the DNS. When using one of the programmatic providers, Cloudron can get Let's Encrypt certificates using DNS automation.

If you decide not to use one of those providers and instead use `Wildcard` or `Manual` DNS, then you must also forward port 80 from your router to the server. This is required to obtain Let's Encrypt certificates.

> ⚠ **Self-signed certificates**
>
> We discourage use of Cloudron with self-signed certificates. The issue is not of security but of usability. Most mobile apps do not work with self-signed certificates. Users keep seeing nagging scary screens on their browsers and the overall user experience is poor.

Port Forwarding

If you require Cloudron to be accessible from outside the home network, you must port forward 443 in your router's firewall to the Cloudron server. See this site for router specific instructions on how to setup port forwarding.

Some apps use custom TCP ports (for git, p2p, etc). You need to set up port forwarding for those as well when you install the apps.

> ⚠ **Port 443**
>
> Be sure to forward port 443 before you do the domain setup. Otherwise, you cannot reach the dashboard after the domain setup.

NAT Loopback

NAT loopback or Hairpinning is a feature of the router allowing internal services to access self or other services via the public IP. This feature allows an app on Cloudron to reach another app on Cloudron using the DNS name (which resolves to the public IP). This feature is crucial for OIDC login to work. Most modern routers support this.

2.3 Intranet

2.3.1 Prerequisites

Cloudron can be installed in an intranet as long as the following prerequisites are met. For this document, intranet is defined as a private network behind a corporate firewall. The server uses private IPs and users connect to the server after connecting to the corporate network with a VPN.

IPv4 / IPv6

By default, Cloudron uses the public IPv4 / IPv6 for configuring the DNS. For intranet setups, this will most likely be incorrect.

In the Domain Setup UI, click on `Advanced Settings` and choose `Network Interface` or the `Static IP` provider. Be sure to configure the Cloudron VM to have this static internal IP address across reboot (usually in your DHCP server).

DNS Provider

Cloudron supports a variety of DNS providers to automatically configure the DNS. When using one of the programmatic providers, Cloudron can get Let's Encrypt certificates using DNS automation.

In an intranet setup, Cloudron has no way to get Let's Encrypt certificates without a programmatic DNS provider. If you are unable to choose a programmatic DNS provider during installation time, choose `Self-signed` certificates in the `Advanced Settings` of the Domain Setup UI. Later, you can upload valid certificates in the `Domains` view or better yet switch to a supported DNS provider to get Let's Encrypt certificates.

> ⚠ **Self-signed certificates**
>
> We discourage use of Cloudron with self-signed certificates. The issue is not of security but of usability. Most mobile apps do not work with self-signed certificates. Users keep seeing nagging scary screens on their browsers and the overall user experience is poor.

3. Support

3.1 Unreachable Dashboard

Follow the instructions below if Cloudron dashboard is unreachable. If the dashboard is reachable, see the following sections to troubleshoot specific issues:

- App issues
- Mail Server issues
- Service issues

3.1.1 Troubleshooting Tool

If the Cloudron dashboard is unreachable, the first step is to SSH into the server and run the troubleshooting tool - `cloudron-support --troubleshoot`.

The output of the tool should give an idea of what is not working. If the checks fail, the tool will give further suggestions. Each of the tests are explained in the sections below.

```
cloudron-support --troubleshoot
```

```
Vendor: Hetzner Product: vServer
Linux: 6.8.0-50-generic
Ubuntu: noble 24.04
Processor: AMD EPYC Processor
BIOS NotSpecified  CPU @ 2.0GHz x 3
RAM: 3911360KB
Disk: /dev/sda1        57G
[OK]    node version is correct
[OK]    IPv6 is enabled and public IPv6 address is working
[OK]    docker is running
[OK]    docker version is correct
[OK]    MySQL is running
[OK]    nginx is running
[OK]    dashboard cert is valid
[OK]    dashboard is reachable via loopback
[OK]    box v8.2.0 is running
[OK]    netplan is good
[OK]    DNS is resolving via systemd-resolved
[OK]    Dashboard is reachable via domain name
[OK]    Domain smartserver.io is valid and has not expired
[OK]    unbound is running
```

3.1.2 Tests

Node version

Cloudron is developed and tested against a specific node version. This might have changed if you installed nodejs on the server. Cloudron does not support making changes to the base system. Follow the instructions provided in the output to revert the node version.

IPv6

On some VPS providers, IPv6 is enabled and an address is assigned but IPv6 routing does not work. As a result, your server cannot reach other IPv6 servers. The tool should provide instructions on disabling IPv6 as a temporary measure. You should contact your VPS provider's support to fix the issue long term.

Docker

When docker is not running, check `journalctl -u docker` and `systemctl status docker` for clues. Docker sometimes obsoletes options between versions. Check if you have a custom config at `/etc/systemd/system/docker.service.d` that requires fixing.

Incorrect Docker Version

Cloudron is developed and tested again a specific docker version. This might have changed if you installed docker or compose on the server. Cloudron does not support making changes to the base system. Follow the instructions provided in the output to revert the docker version.

MySQL

> **Two instances of MySQL**
>
> There are two instances of MySQL on Cloudron. One instance runs on the host and is used by the platform. Another instance is the MySQL addon which runs in a container named `mysql` and is shared by apps. This test is related to the host MySQL.

Check the systemd MySQL service for clues using `systemctl status mysql` . Check for errors and warnings in `/var/log/mysql/error.log`.

Recovering from MySQL corruption is a complex technical topic. You can try to restore the quick method below failing which it's best to restore your server entirely from the latest Cloudron backup.

1. Locate the latest box backup in your backup storage . This is a file named `box.tar.gz` (tgz) or `box` (rsync) . Extract or locate the file `box.mysqldump` inside it. Copy this file to some location in your server.
2. Stop box code:

```
systemctl stop box
```

3. Stop MySQL:

```
systemctl stop mysql
```

4. Recreate the existing MySQL installation:

```
mv /var/lib/mysql /var/lib/mysql.old
mkdir /var/lib/mysql
chown -R mysql:mysql /var/lib/mysql
mysqld --initialize   # This will dump the MySQL root password in /var/log/mysql/error.log
```

5. Start MySQL:

```
systemctl start mysql
```

6. Connect to the MySQL service

```
# there is no space between p and the password. e.g -pMySecurePassword
mysql -uroot -p<password from /var/log/mysql/error.log>
```

7. Change the root password to `password`:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

8. Import the database:

```
mysql -uroot -ppassword < /path/to/box.mysqldump
```

9. Start the platform code again:

```
systemctl restart box
```

nginx

nginx is the reverse proxy that serves up the Cloudron dashboard and the apps. Use `journalctl -u nginx` and check `/var/log/nginx/error.log` to investigate why it is not running.

Often, the issue is that some config is incorrect or corrupt. nginx configs are located at `/home/yellowtent/platformdata/nginx/` . Delete some files here and try to make `systemctl restart nginx` work. You can safely delete nginx certificates and config files and attempt to get it running. Cloudron can regenerate certificates and configs from it's database. Once you have access to the dashboard, go to App -> Repair to regenerate the nginx config.

Dashboard Certificate

Invalid certificates are often caused by invalid DNS API tokens (wildcard certs) or incoming port 80 to server is blocked by the firewall (manual or wildcard DNS.

To fix, visit the dashboard and accept the self-signed certificate in the browser. Then, go to Domains view and update the DNS API Token. Then, renew the certs using `Domains` -> `Renew Certs` .

> **API Token may be tied to IP**
>
> Some DNS API tokens (cloudflare, namecheap) can be restricted by IP. If your server's IP changed, then the token rules have to be fixed.

Loopback / Hairpin NAT

Cloudron configures the DNS to be the public IP address. If the router does not support Hairpin NAT, then you will have trouble reaching the dashboard and the apps from inside your network.

There are many ways to fix this and the solution depends on your use case. Here are some suggestions:

- If you intend to use Cloudron and the apps from only inside your network, change the network settings to use the local IP address. When doing so, Cloudron will use the local IP in the DNS instead of the public IP address.
- If you intend to use Cloudron from outside your network as well, the easiest fix is to purchase a new router that supports hairpin NAT.
- If purchasing a new router is not an option:
    - Configure your network's DNS server to return the Local VM IP for all the subdomain in use. This way when your PC/Laptop accesses a domain, it starts using the Local VM IP instead of the public IP to connect to Cloudron. Devices outside the network will continue to use the public IP address as expected.
    - Some apps use the domain name to connect with each other (For example, collabora app connects to nextcloud app with domain name). For such cases, configure Cloudron's DNS server `unbound` to use your network's DNS.

Netplan

Ubuntu uses netplan by default to configure networking. This is an optional component and your VPS provider could have disabled it. When enabled, the configs are usually located at `/etc/netplan/50-cloud-init.yaml` . This file contains information on how your server learns about it's IP and nameservers. netplan generates systemd (default) or NetworkManager configs (desktop Ubuntu) based on `network.renderer` .

DNS Resolution

`systemd-resolved` is the Ubuntu system DNS. It runs on `127.0.0.53` .

Use `resolvectl` to see the name servers being used. Usually, the system gets it's nameservers via netplan configs. You can also hardcode the nameservers in `/etc/systemd/resolved.conf`. Set `DNS=1.1.1.1`, for example.

- Check if `systemctl status systemd-resolved` is running.
- Check if `host www.cloudron.io 127.0.0.53` works.
- Check if `systemd-resolved` is the systemd DNS by checking if `/etc/resolv.conf` has `nameserver 127.0.0.53`

Domain issues & expiry

First, check if the domain points to your server.

Next, check if your domain has not expired. If the domain has expired and you renew it, you have to wait a bit for the DNS records to come back. This is because most registrars replace your DNS with parking pages until you renew. Some registrars, won't revert back your old DNS records. In such situations, add a DNS A record for `my.example.com` to point to the server's IP. Then, wait and a bit for the DNS to propagate and login to the dashboard. Go to `Domains` view and click the Sync DNS button to re-add Cloudron related DNS records.

If the domain has expired, but you don't plan to renew it, then edit your PC/Mac's `/etc/hosts` and add an entry (the old domain and not the new one) to point to the server.

```
1.2.3.4 my.example.com
```

- With the above entry in place, you can visit `https://my.example.com` in your browser. You may have to purge the site from your browser's history to get over HSTS certificate issues.
- Login and go to `Domains` view. Add the new domain.
- Change the dashboard domain to this new domain.
- Move all the apps one by one from the old domain to the new domain by changing the location.
- Delete the old domain and remove the entry we added earlier from `/etc/hosts`.

3.1.3 Failed upgrade

If the domain looks ok and nginx is running fine, check the box logs at `/home/yellowtent/platformdata/logs/box.log`.

To make sure, all migrations are up to date:

- Run `/home/yellowtent/box/setup/start.sh` on the server.
- Check `systemctl status box`. If it's not running, try `systemctl restart box`

If it still has errors, please contact support.

3.2 Support

3.2.1 Forum

The Forum is the best place to ask support questions. The Forum makes reported issues SEO friendly and helps others (and you) find resolutions to existing problems.

Open Forum

3.2.2 Email

For sensitive support questions, please write to support@cloudron.io. Please send emails from your Cloudron.io account so that we can associate your request with your subscription. If you have multiple subscriptions, please put in your Cloudron ID as well. You can find the Cloudron ID in the `Settings` page of the dashboard as well as https://console.cloudron.io/#/cloudrons .

For sales related enquiries, please write to sales@cloudron.io.

3.2.3 Remote Support

⚠ **Remote Support Disclaimer**

Please read our Remote Support Disclaimer carefully before giving us Remote Access.

To enable SSH access to your server for the Cloudron support team, you will have to SSH into your server and run the following command:

```
cloudron-support --enable-remote-support
```

You can also add the following SSH keys manually. Depending on your setup, the keys below can be added to the following file: `/home/cloudron-support/.ssh/authorized_keys` .

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGWS+930b8QdzbchGljt3KSljH9wRhYvht8srrtQHdzg support@cloudron.io
```

To disable support:

```
cloudron-support --disable-remote-support
```

Keep the key to a single line

SSH key must be added as a single line. Make sure it doesn't break across multiple lines.

4. Knowledge Base

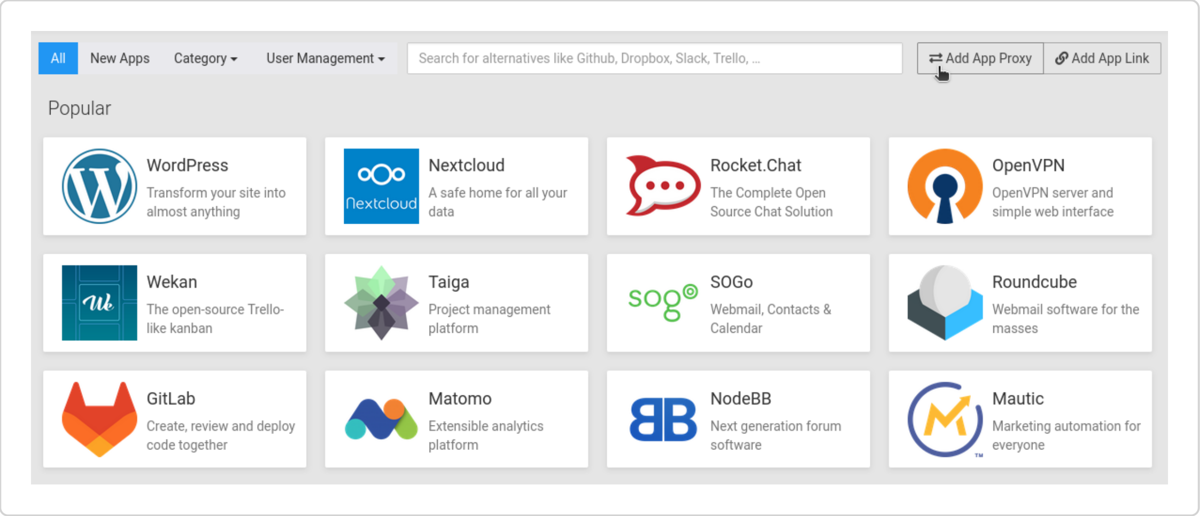
4.1 Apps

4.1.1 Installation

Three types of app icons can be added in the dashboard:

- App Store Apps
- App Links
- App Proxy

Icons can be added from the `Appstore` view:


All
New Apps
Category ▾
User Management ▾
Search for alternatives like Github, Dropbox, Slack, Trello, ...
⇄ Add App Proxy
Add App Link
Popular
WordPress
Transform your site into almost anything
Nextcloud
A safe home for all your data
Rocket.Chat
The Complete Open Source Chat Solution
OpenVPN
OpenVPN server and simple web interface
Wekan
The open-source Trello-like kanban
Taiga
Project management platform
SOGo
Webmail, Contacts & Calendar
Roundcube
Webmail software for the masses
GitLab
Create, review and deploy code together
Matomo
Extensible analytics platform
NodeBB
Next generation forum software
Mautic
Marketing automation for everyone

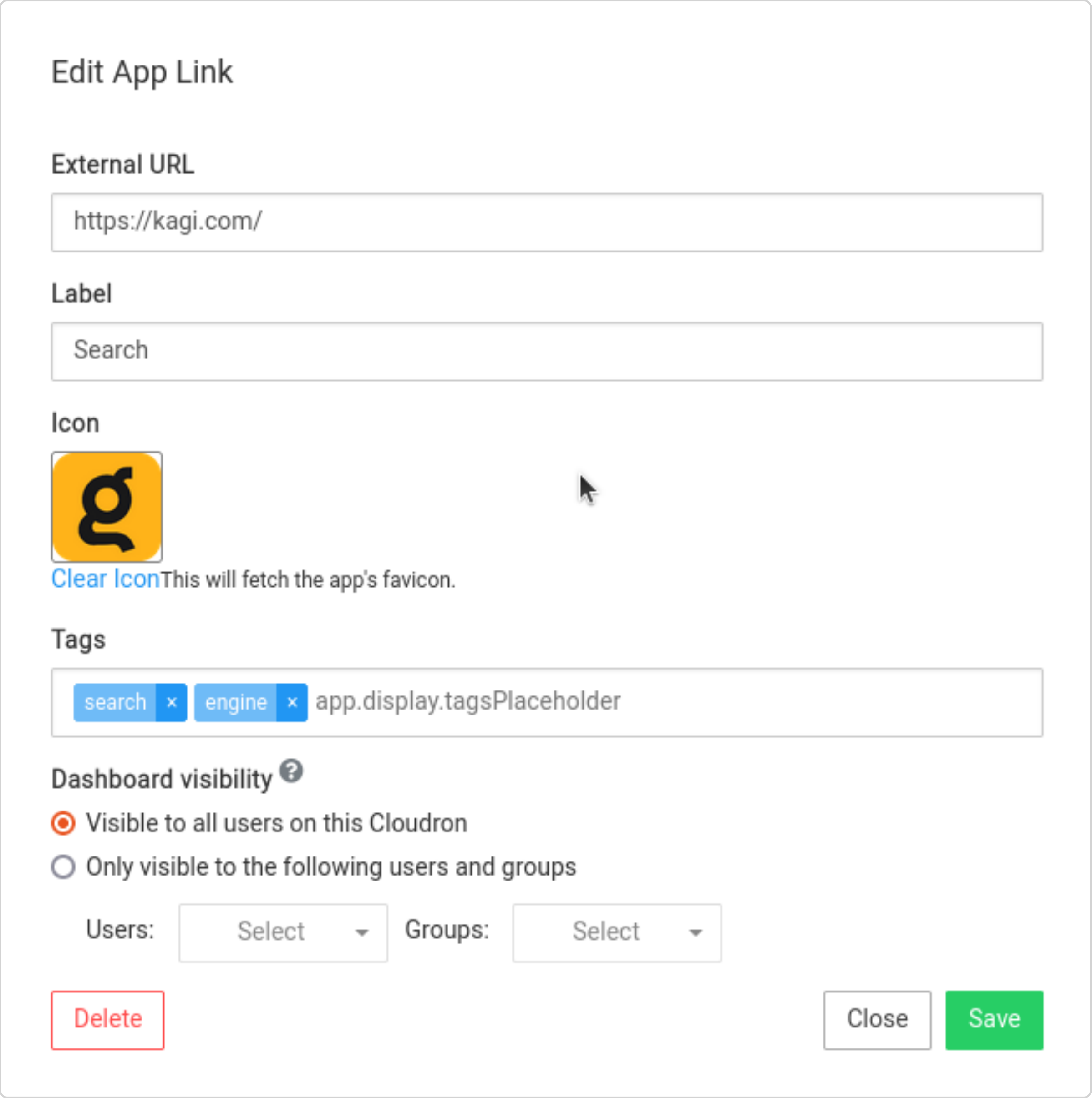
App Store

Apps can be installed from the `App Store` menu item. Clicking on an app will display information about the app.

Clicking the `Install` button will show an install dialog:

App Links

An App Link is a shortcut to an external web site. Clicking on a App Link simply opens up the website in a new tab on the user's browser.



Edit App Link

External URL

https://kagi.com/

Label

Search

Icon

Clear IconThis will fetch the app's favicon.

Tags

search × engine × app.display.tagsPlaceholder

Dashboard visibility

● Visible to all users on this Cloudron

○ Only visible to the following users and groups

Users: Select Groups: Select

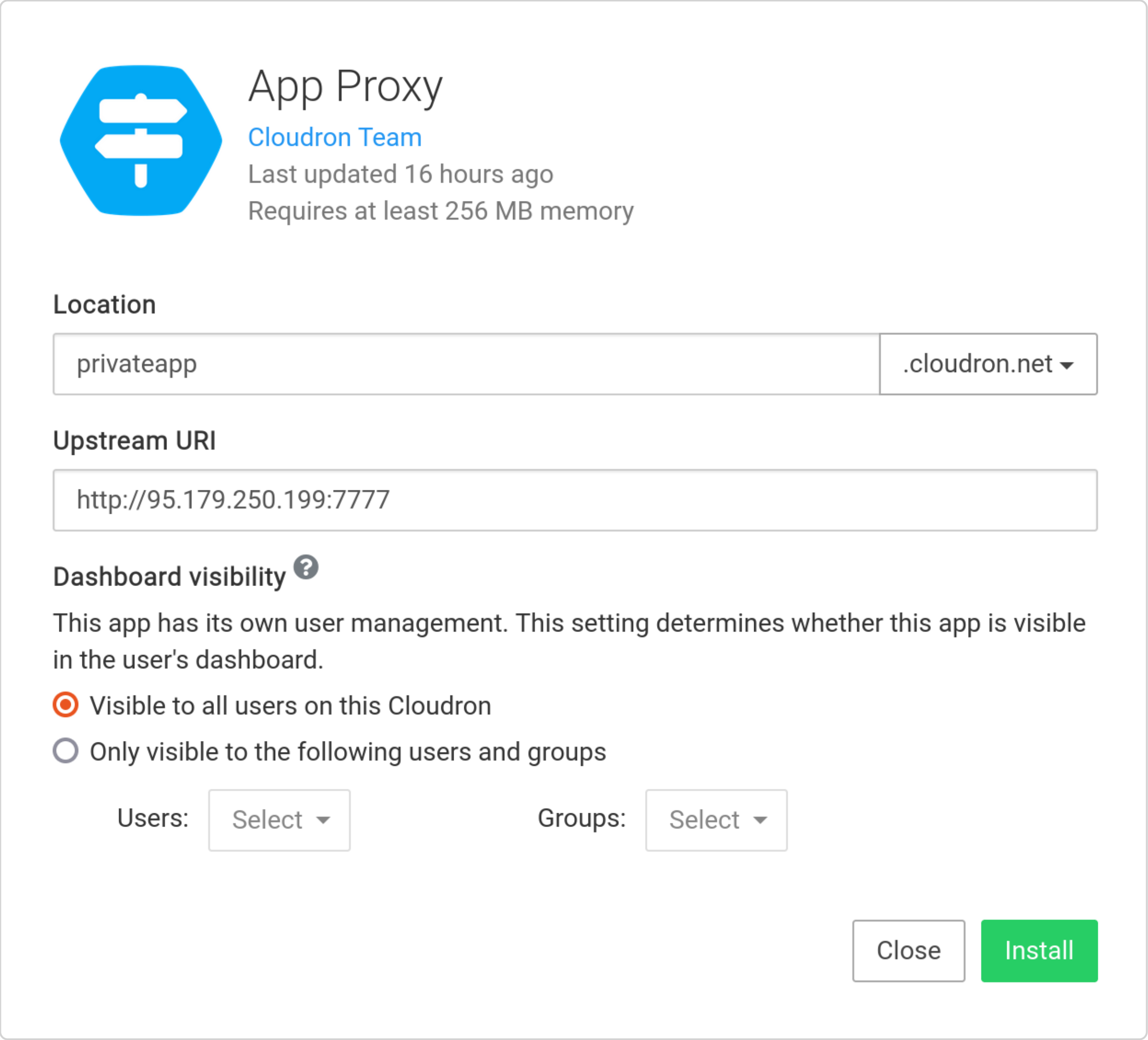
App Proxy

App Proxy is a service that lets one publish a public HTTPS URL endpoint for a non-Cloudron hosted application. When a user visits the public endpoint, App Proxy proxies requests to the hosted application.

When using HTTP proxying, you must ensure that the network between Cloudron and the internal application is secure.

Benefits of using the App Proxy include DNS management, Certificate management, Configurable domain aliases and redirections, setting a custom CSP, setting custom `robots.txt` and up/down notifications.

You can also specify which users & groups can see the proxy app icon using the `Dashboard Visibility` setting.



App Proxy

Cloudron Team

Last updated 16 hours ago

Requires at least 256 MB memory

Location: privateapp .cloudron.net

Upstream URI: http://95.179.250.199:7777

Dashboard visibility

This app has its own user management. This setting determines whether this app is visible in the user's dashboard.

● Visible to all users on this Cloudron

○ Only visible to the following users and groups

Users: Select Groups: Select

The Upstream URI can have one for the following formats:

- `http://ip:port` or `http://[ipv6]:port`
- `https://ip:port` or `https://ipv6]:port`
- `http://domain:port`
- `https://domain:port`

When using `https`, certificates are not verified.

⚠ **Updates & Backups**

As the app is hosted externally, managing app updates and backups of proxied apps are outside the scope of Cloudron.

⚠ **Cannot mirror public sites**

App Proxy has not been designed to mirror or proxy other people's public sites or domains. The App Proxy sets the `Host` header to the App Proxy's location when proxying and not the upstream/target domain. You will most likely see a 502 error if you try to mirror public sites.

Protect the upstream

It's a good idea to have a firewall rule in the upstream server to only accept requests from Cloudron server IP.

4.1.2 Configuration

Clicking on the gear button will open the app's configure view.

4.1.3 Location

Primary Domain

The `Location` field, also known as Primary Domain, is the subdomain into which the app will be installed. Use the drop down selector on the right to choose the domain into which the app will by installed. If the subdomain field is empty, the app will be installed in the bare/naked domain.

Cloudron packages are "relocatable" by design. Changing the location field in the `Location` section of the app's configure UI will move the app to another domain or subdomain:

Location field can be multi-level

The `Location` field can be any level deep. For example, you can specify location as `blog.dev` to make the app available at `blog.dev.smartserver.space`.

No data loss

Moving an app to a new location is a non-destructive action. Existing app data will be migrated to the new domain.

Secondary Domains

Some apps require more than one domain. For example, minio uses two separate domains - one for the UI and one it's API. Other examples include Loomio (websockets domain), CryptPad (sandbox domain) and Traccar (OsmAnd protocol).

Secondary domains can be specified at installation time. Like the Primary domain, they can be changed later in the `Location` section:

Aliases

Some apps can be reached via more than one domain. For example, WordPress multi-site can serve up websites based on the domain name. EspoCRM supports creating customers portals on custom domains.

Aliases can be setup from the `Location` section in the app's configure UI:

The alias feature is only enabled for select apps since it requires apps to support multiple domains.

Redirections

Redirections forward one or more domains to the primary domain with a HTTP 301. They can be setup from the `Location` section in the app's configure UI:

In the above example, anyone visiting `chat2.cloudron.ml` or `chat3.smartserver.io` will be automatically redirected to the main domain `chat.cloudron.ml` (with a HTTP 301).

The redirection feature preserves any URI components like subpaths in the original request.

`www` redirection

In DNS, the domains `example.com` and `www.example.com` are independent and can point to completely different websites. In practice, it is a good idea to forward one to the other. Do this, by adding `www` or the bare domain as a redirection.

Port Bindings

TCP and UDP port bindings can be configured in the `Location` section in the app's configure UI:

In the above example, the UDP Port of the VPN app is exposed at port 7194. The TCP Port is disabled.

When enabling ports, remember to also whitelist ports in any Cloud Firewall.

Ephemeral ports

Ports in the range 32768-60999 is used to temporarily connect to external servers (e.g email, backups, volumes). It is recommended to run apps outside this range to prevent conflicts.

Cloudflare proxying

If you proxy the domain via Cloudflare, port bindings will not work because Cloudflare only proxies HTTP and HTTPS.

4.1.4 Display

Label

`Label` is the text that is displayed for the app on the dashboard below the icon.

Tags

`Tags` are a mechanism to tag apps with labels. For example, you can mark specific apps with the customer name and filter apps by customer name.

Icon

A custom icon for the app can be set in the `Icon` section. When not set, the App package's icon is used.

4.1.5 Access Control

Access Restriction

Many apps in Cloudron are integrated with Cloudron's user management. For such apps, one or more groups or users can be assigned to an app to restrict login. For apps not integrated with Cloudron user management, see the section on controlling the visibility of app icon in dashboard.

Note that Cloudron only handles authentication. Assigning roles to users is done within the application itself. For example, changing a user to become a `commenter` or `author` inside WordPress has to be done within WordPress.

- `Allow all users from this Cloudron` - Any user in the Cloudron can access the app.
- `Only allow the following users and groups` - Only the users and groups can access the app.

Dashboard Visibility

The Dashboard of a Cloudron user displays the apps that the user can access. For apps that use Cloudron Single Sign-on, the dashboard only displays an app if the user has access to it.

For apps configured to not use the Cloudron Single Sign-on (for example, some public app like a Forum or Chat), the apps are displayed (by default) on the dashboard of all users. Admins can control if an app appears in a user's dashboard using the `Dashboard Visibility` section in the app's configure UI.

Operators

An admin can set user(s) & group(s) as the operators of an app. An app operator can perform configuration and maintanence tasks. Unlike an app admin, an operator cannot uninstall the app or change it's location. Operators cannot clone apps either because they do not have the permissions to install new apps.

An operator will see the gear icon on their dashboard:

On clicking the gear icon, they will see the operator UI:

4.1.6 Info

Various app information can be found in the `Info` section of the app:

- `App Title and Version` - This is the app's title and the upstream app version
- `App ID` - Unique ID of the app instance
- `Package Version` - Version of the Cloudron package. This is distinct from the (upstream) App Version
- `Installed At` - When the app was installed
- `Last Updated` - When the app was last updated

Admin Notes

App specific notes can be saved in markdown format. Notes are shared by admins. All Admins and App Operators can view and edit them.

Checklist

Checklists provide a mechanism to inform administrators of urgent and security related tasks that need to be carried out at the earliest. Examples include changing the default admin credentials, reviewing registration settings, etc.

Checklist appears in the `Info` section of the app.

When a Checklist item is marked as done, the username and date of completion is tracked for audit purposes.

4.1.7 Resources

Memory limit

All apps are run with a memory limit to ensure that no app can bring down the whole Cloudron. The default memory limit of an app is set by the app author at packaging time. This limit is usually the minimum amount of RAM required for the app. Cloudron admins are expected to tweak the memory limit of an app based on their usage.

When an app runs out of memory, Cloudron automatically restarts it and sends an OOM notification to Cloudron admins.

The memory limit can be set by adjusting the slider in the `Resources` section of the app's configure view.

> **Unlimited Swap**
>
> The memory limit specified above is just the RAM. All apps get unlimited swap.

Low Resource Warning

When you try to install a new app, a 'Low Resource Warning' message may be displayed based on the calculation of maximum memory limits of existing installed apps. This is a warning that the server will run out of memory, in case all apps are close to their set memory limit.

The warning is shown based on a conservative estimate, because more often than not, apps use well below their maximum memory limit.

CPU Limit

By default, all apps use as much CPU as they need. To constrain the maximum CPU usage, you can set a CPU Limit. If your server has 16 cores, then a setting of 50%, will limit the app to use a maximum of 8 cores at a time.

The CPU limit can be set by adjusting the slider in the `Resources` section of the app's configure view.

Devices

A list of host devices to be mounted into the app can be specified in the `Devices` section.

4.1.8 Storage

Data Directory

Apps store their data and assets in the `/home/yellowtent/appsdata/<appid>` directory. If the server is running out of disk space (in the root filesystem), you can move the app's storage directory to another location. In most cases, this is an external disk mounted on the server. For example, you can mount a DigitalOcean Block Storage or AWS Block Store and move the app's data to that disk.

For example, to move an app's data to an external disk location like `/mnt/seagate`:

- Add the external disk as a volume named `seagate`.
- Go to the app's `Storage` section and select the volume. An optional prefix may be specifed to store the data in a subdirectory.

> ⚠ **Volume Type**
>
> `cifs` and `sshfs` volumes are unsuitable as an app's data directory as they do not support file permissions and ownership. `mountpoint` and `nfs` volumes may or may not work depending on the destination filesystem.

> **App Data Directory is backed up**
>
> The external app data directory is part of the app's backup.

Mounts

Apps on Cloudron are containerized and do not have access to the server's file system. To provide an app access to a path on the server, one can create a Volume and then mount the volume into the app. Apps can access any mounted volumes via `/media/{volume name}` directory in their file system.

For example, to give an app access to an external disk `/mnt/music`:

- Create a volume in the `Volumes` view name `music`.
- Add an app mount.

The app can access the music files from `/media/music` (which corresponds to the host path `/mnt/songs`).

When the read only flag is checked, the `/media/music` directory is not writable.

Mounts are not backed up

Volumes are not backed up. Restoring an app will not restore the volume's content. Please make sure to have a suitable backup plan if you write to them.

4.1.9 Email

Mail FROM address

For apps that can send email, Cloudron automatically assigns an address of the form `<location>.app`. To change this name, go to the `Email` section in the app's configure UI.

Display name

Support for email address display name depends on the app. If the display name input box is missing, it means that the app doesn't support it (possibly because it uses a dynamic display name). If the display name is empty, the app package provides a suitable default (usually the app's title).

Disable Email Configuration

For select apps, you can also disable email auto-configuration using `Do not configure app's mail delivery settings`. When selected, Cloudron will not configure email delivery settings inside the app, you can set it up yourself.

This is not a mailbox, just an address

The app is simply configured to send mails with the above name. If you want to receive email with the address, be sure to create a mailbox. If a mailbox with the name does not exist, any replies to the email will bounce.

Inbox

For apps that can receive email, the inbox address for the app can be assigned in the `Email` section of the app's configure UI.

When an inbox address is assigned, Cloudron will configure the app to receive mails using that address. It will also generate a dynamic username and password for the app to use to access the inbox.

An inbox address can only be assigned, if the email server for the domain in hosted on Cloudron. If the email server is external to Cloudron, use the "Do not configure inbox" option and configure the app on your own.

Mailbox must be manually created

The app is simply configured to receive mails with the above address. You must create a mailbox for emails to be received by the mail server.

4.1.10 Security

robots.txt

The `Robots.txt` file is a file served from the root of a website to indicate which parts must be indexed by a search engine. The file follows the Robots Exclusion Standard. Google has an excellent document about the semantics.

The robots.txt contents of an app can be set in the `Security` section of the app's configure UI.

By default, Cloudron does not setup a robots.txt for apps. When unset, the app is free to provide it's own robots.txt.

In addition, the Cloudron admin page has a hardcoded robots.txt that disables indexing:

```
User-agent: *
Disallow: /
```

Custom CSP

The CSP HTTP header instructs the browser to only load scripts, media, images and other resources only from specific sites. Some apps set these headers to be overly restrictive and provide no way to customize them. For such apps, you can override the CSP headers set by the app.

For example, to embed Mattermost in another site, you can set the following CSP policy for Mattermost:

```
frame-ancestors site.example.com;
```

HSTS Preload

HSTS Preload is a list of sites that are hardcoded into Chrome as being HTTPS only. Most major browsers (Chrome, Firefox, Opera, Safari, IE 11 and Edge) also have HSTS preload lists based on the Chrome list.

Requirements and implications:

- Due to the size of the preload list, automated preload list submissions of whole registered domains (bare domain) are accepted.
- This will prevent all subdomains and nested subdomains being accessed without a valid HTTPS certificate.
- New entries are hardcoded into the Chrome source code and can take several months before they reach the stable version.

When enabled, Cloudron will serve the following HSTS headers:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
```

To enable HSTS Preload, enable it in the `Security` section of the app:

Submission

Cloudron does not automatically submit the domain to the HSTS Preload list. You must do that manually here.

4.1.11 Cron

Cron jobs required for the app to function are already integrated into the app package and no further configuration is required. If you want to run additional custom cron commands, you can add them in the `Cron` section.

Cron commands are run with the exact same context as the app (in a one-off container). This means that they have access to all the same environment and databases as the app itself. They also follow the life cycle states of the app. When an app is stopped, they don't run anymore. The log output of the cron commands can be viewed using the log viewer.

Cron times are specified in UTC.

The schedule pattern can also be one of the following cron extensions:

- `@service` : Run once on app restart or if app is already running.
- `@reboot` : Run once on app restart or if app is already running.
- `@yearly` : Run once a year, ie. `0 0 1 1 *`.
- `@annually` : Run once a year, ie. `0 0 1 1 *`.
- `@monthly` : Run once a month, ie. `0 0 1 * *`.
- `@weekly` : Run once a week, ie. `0 0 * * 0`.
- `@daily` : Run once a day, ie. `0 0 * * *`.
- `@hourly` : Run once an hour, ie. `0 * * * *`.

> **Chaining commands**
>
> The command can be chained using `&&` or '||' . For example, `echo "=> Doing job" && /app/data/do_job.sh`

4.1.12 Services

Redis

By default, apps requiring Redis for caching are set up to use a standalone internal redis database. If you want to conserve resources or your usage of the app doesn't necessitate caching, you can disable redis in the `Services` section.

TURN

By default, apps requiring STUN/TURN configuration are set up to use the built-in TURN service. If you prefer to use an external one, this can be disabled in the `Services` section.

4.1.13 Web Terminal

Cloudron provides a web terminal that gives access to the app's file system. The web terminal can be used to introspect and modify the app's files, access the app's database etc. Note that Cloudron runs apps as containers with a read-only file system. Only `/run` (dynamic data), `/app/data` (backup data) and `/tmp` (temporary files) are writable.

The web terminal can be accessed using the Web Terminal button:

Clicking the icon will pop up a new window. The terminal is essentially a shell into the app's file system.

Customizing Web Terminal

The bash terminal of each app will source `/app/data/.bashrc` if it exists.

Meaning, if you create a `/app/data/.bashrc` and add custom options like:

```
echo "Welcome to $CLOUDRON_APP_ORIGIN"
echo "APPID: $(uname -n)"
echo "Workdir: $(pwd)"
echo "Datadir: /app/data/"
echo "Time: $(date '+%d.%m.%Y %H:%M')"
echo "Use 'printenv | grep -i Cloudron' to get all Cloudron varaibles"
```

The Web Terminal and cloudron-cli will display this:

```
Welcome to https://docs.cloudron.io
APPID: 3d5ad25e-3741-4506-9be2-f18b482a7800
Workdir: /app/code
Datadir: /app/data
Time: 01.07.2025 10:25
Use 'printenv | grep -i Cloudron' to get all Cloudron varaibles
```

4.1.14 File Manager

Cloudron provides a File Manager that be used to modify the app's file system from the browser.

The File Manager can be accessed using the File Manager button:

Clicking the icon will pop up a new window. Note that there are action like Rename, Delete, Change Ownership in the context menu.

4.1.15 SFTP Access

Certain apps like WordPress, LAMP, Surfer support access to their data via SFTP. Files can be viewed and uploaded using any SFTP client. The FTP connection information can be displayed by clicking the `SFTP Access` menu item.

A SFTP client like FileZilla can be used to connect as follows:

- `Host` - `sftp://my.cloudron.space` (host is the same for SFTP access to all apps)
- `Username` - `girish@lamp.cloudron.space` (username is different for SFTP access to each app)
- `Password` - Cloudron password (password is the same for SFTP access to all apps)
- `Port` - 222

Only Cloudron admins have SFTP access. To give a specific user access to SFTP of a single app, make them an operator.

Port 222

SFTP service runs at port 222. The server firewall already has this port open. However, you will have to whitelist this port in the Cloud firewall (e.g EC2 Security Group or DigitalOcean Firewall). If the domain is fronted by Cloudflare, use the IP address of the server to connect via SFTP instead of `my.domain.com`.

4.1.16 Log Viewer

To view the logs of an app, click the logs button:

This will open up a popup dialog that display the logs:

Up to 10MB of current logs and one rotated log is retained per app. Logs older than 14 days are removed. The raw logs are located at `/home/yellowtent/platformdata/logs/<appid>/`.

4.1.17 Graphs

The Graphs view shows an overview of the CPU, disk, network and memory usage of the app.

4.1.18 Stop App

An app can be stopped using the `Stop` button from the app toolbar.

4.1.19 Restart App

An app can be stopped using the `Restart` button in the `Repair` section.

4.1.20 Recovery Mode

When an app is not working as expected or keeps crashing, it useful to have the app container in an introspectable state. The File manager is always available but the Web Terminal won't work because the app is continuously crashing. This method can also be used to disable any problematic extensions or plugins preventing the app from starting up.

For such situations, first check the app logs and then place the app in `Recovery Mode` for debugging. Apps in `Recovery Mode`:

- Start in a 'paused' state. The Web Terminal can be used to start the app manually. By convention, the startup script is located in `/app/pkg/start.sh`. Run this command to see where the app crashes.
- Use the database buttons on the top of Web Terminal to access the app databases.
- Cloudron runs containers in a readonly filesystem. In `Recovery Mode`, the entire filesystem is writable and you can make make changes to the application code. Note that all code changes will be lost when you leave the `Recovery Mode`!

4.1.21 Archive

An app can be archived by using the `Archive` button in the `Uninstall` view.

Archiving an app is similar to uninstalling an app. All data associated with the app is removed from the server and the app won't appear in the main dashboard. There are however, key differences:

- The latest app backup is kept forever and is not affected by the Cleanup policy. In contrast, when an app is uninstalled, the latest backup is removed based on the Cleanup policy.
- The app is moved to the `App Archive` in the `Backups` view. Apps in the `App Archive` can be easily restored. In contrast, when an app is uninstalled, you have to use the App Import mechanism which is more complicated.
- Deleting an entry from the `App Archive` will delete the associated backup based on the Cleanup policy.

4.1.22 Uninstall

An app can be uninstalled using the `Uninstall` button in the app's configure UI.

Uninstalling an app immediately removes all data associated with the app from the server.

App backups are not removed immediately when an app is uninstalled. They are cleaned up based on the Cleanup policy. Provided you still have the backups, an app can be restored using App Import.

> **Archiving is recommended**
>
> If there is the remotest chance that the app might come in handy later, consider Archiving the app instead. This approach will not only free up space on your server but also make it possible to easily restore the app later when you need it.

4.1.23 Version

There are two independent versions associated with an app. These are shown in the info section.

- The `Package version`. Cloudron uses semver for it's app packages.
- The `App version` or `Upstream version`. App version format varies wildly. It can be date based, semver, git commit based, numbers etc.

Install Specific Version

When you install an app from the `App Store` view, it installs the latest package. It is possible to install older packages and older versions of the app, as long as the Cloudron version supports it.

To determine the version, you have to look into the package's source code. All Cloudron app packages are open source and available at https://git.cloudron.io/cloudron . Each app package is in it's own repository and has the `-app` suffix.

For example, Managed WordPress, GitLab, Espo CRM and so on.

To install a specific package version:

- First, determine the desired package version. In each git repo, there is a `CHANGELOG` . Let's say we want to install WordPress 6.4.3 . For this, we locate this line. The line above says that Package Version 3.6.1 has WordPress 6.4.3.
- Go to `App Store` view and click on the app.
- Change the URL bar to `?version=3.6.1` parameter as above to the desired package version and press enter.
- Install the app

Automatic update

If you want to stick to the installed version, be sure to disable automatic app updates .

4.1.24 Troubleshooting

As a first step, check the app logs to look for errors.

- For transient errors like database connectivity error, a simple restart will sort out the issue.
- Check if the services used by the app are running in the `Services` view. If one or more are not running, see the Services docs.
- Use the recovery mode for debugging.
- If the issue is related to a mis-configuration or an error in some plugin, theme or extension, the approach to fix depends on the app and the tooling the app provides. For example, WordPress plugins can be disabled using WP CLI and Nextcloud can be disabled using occ. Please look into the Cloudron's app docs for hints (Look under `Apps` in the side bar).
- To test if the app starts, run the `start.sh` script using the Web Terminal. This script is usually located under `/app/pkg` or `/app/code` (this is a Cloudron packaging convention).
- Once fixed, disable recovery mode. This will start the app.

4.2 App Store

4.2.1 Overview

The Cloudron App Store is a repository of apps hosted at cloudron.io. The App Store provides app packages that can be installed on a Cloudron. A Cloudron installation periodically polls the App Store for updates.

4.2.2 Account

A Cloudron App Store account (cloudron.io account) is used to manage your subscription & billing. Before installing apps, you must set up the Cloudron with your App Store Account information. By doing so, the Cloudron will register itself and get an unique Cloudron ID.

You can view this information in the `Settings` page.

4.2.3 Password reset

The password of the App Store account can be reset here.

4.2.4 Account Change

To switch the App Store account (cloudron.io account) to a new email address, you can simply change the email address in your cloudron.io profile.

It's not possible to transfer subscriptions to another existing cloudron.io account. If really needed, the workflow for account transfer is as follows:

- Send the Cloudron ID(s) to transfer with the email address of your old account to support@cloudron.io
- We will cancel your existing subscription and refund the pro-rated amount
- You can then delete the subscription from your old account.
- When you delete a subscription, Cloudron Dashboard's App Store view will log you put. Login with new cloudron.io account and re-set up the subscription.

4.3 Backups

4.3.1 Overview

The Cloudron backup system creates a portable snapshot of the platform data and application data. Each app is backed up independently allowing them to be restored, cloned or migrated independently.

Unlike VM snapshots, these backups contain only the necessary information for reinstallation of Cloudron or app. For example, application code and system libraries are not part of a backup because Cloudron packages are read-only and can never change. Runtime files (lock files, logs) and temporary files generated by apps are not backed up either. Only the database and app user data is backed up. This design significantly reduces the size of backups.

4.3.2 Storage providers

Amazon S3

To get started:

- Create a bucket in S3.

> ⚠ **Lifecycle rules**
>
> S3 buckets can have lifecycle rules to automatically remove objects after a certain age. When using the `rsync` format, these lifecycle rules may remove files from the `snapshot` directory and will cause the backups to be corrupt. For this reason, we recommend not setting any lifecycle rules that delete objects after a certain age. Cloudron will periodically clean up old backups based on the retention period.

- AWS has two forms of security credentials - root and IAM. When using root credentials, follow the instructions here to create access keys. When using IAM, follow the instructions here to create a user and use the following policy to give the user access to the bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::<your bucket name>",
                "arn:aws:s3:::<your bucket name>/*"
            ]
        }
    ]
}
```

- In the Cloudron dashboard, choose `Amazon S3` from the drop down.

Backblaze B2

To get started:

- Create a Backblaze B2 bucket

⚠ **Lifecycle rules**

Versioning is enabled by default in Backblaze. This means that despite Cloudron periodically deleting old backups, a copy of them is still retained by Backblaze. Over time, these copies tend to add up and can result in a significant cost. We recommend changing the `Lifecycle Settings` to `Keep only the last version of the file`. Given that the Cloudron backups are already versioned by date, you won't need any other copies.

- Create Access key and Secret Access Key from the `Application Keys` section in Backblaze. Be sure to provide read and write access to the bucket. You should restrict access of the key to just the backup bucket.
- Make a note of the `keyID` and `applicationKey`. As noted in their docs:

```
    Access Key  <your-application-key-id>
    Secret Key  <your-application-key>
```

- In the Cloudron dashboard, choose `Backblaze B2` from the drop down. The Endpoint URL has the form `s3.<region>.backblazeb2.com`, where is similar to `us-west-004`.

CIFS

To get started:

- Hosting providers like Hetzner and OVH provider storage boxes that be mounted using Samba/CIFS.
- In the Cloudron dashboard, choose `CIFS Mount` from the drop down.

Hetzner Storage Box

We recommend using SSHFS for Hetzner Storage Box since it is much faster and efficient storage wise compared to CIFS. When using Hetzner Storage Box with CIFS, the Remote Directory is `/backup` for the main account. For sub accounts, the Remote Directory is `/subaccount`.

Cloudflare R2

To get started:

- Create a Cloudflare R2 bucket.
- Generate S3 auth tokens for the bucket.
- In the Cloudron dashboard, choose `Cloudflare R2` from the down down. The S3 endpoint in shown in the Cloudflare dashboard.

⚠ **Remove the bucket name in Cloudflare URL**

Cloudflare dashboard shows a URL that contains the the bucket name in the end. On Cloudron, you should set the `Endpoint` **without** the bucket name in the end.

Contabo Object Storage

To get started:

- Create a Contabot Storage bucket.
- Obtain S3 credentials for the storage.
- In the Cloudron dashboard, choose `Contabo Object Storage` from the drop down.

DigitalOcean Spaces

To get started:

- Create a DigitalOcean Spaces bucket in your preferred region following this guide.
- Create DigitalOcean Spaces access key and secret key following this guide.
- In the Cloudron dashboard, choose `DigitalOcean Spaces` from the drop down.

> **Rate limits**
>
> In our tests, we hit a few issues including missing implementation for copying large files (> 5GB), severe rate limits and poor performance when deleting objects. If you plan on using this provider, keep an eye on your backups. Cloudron will notify admins by email when backups fail.

Exoscale SOS

To get started:

- Create a Exoscale SOS bucket
- Create Access key and Secret Access Key from the Exoscale dashboard
- In the Cloudron dashboard, choose `Exoscale SOS` from the drop down.

EXT4

To get started:

- Attach an external EXT4 hard disk to the server. Depending on where your server is located, this can be a DigitalOcean Block Storage, AWS Elastic Block Store, Linode Block Storage.
- If required, format it using `mkfs.ext4 /dev/<device>`. Then, run `blkid` or `lsblk` to get the UUID of the disk.
- In the Cloudron dashboard, choose `EXT4 Mount` from the drop down.

> **Do not add `/etc/fstab` entry**
>
> When choosing this storage provider, do not add an `/etc/fstab` entry for the mount point. Cloudron will add and manage a systemd mount point.

Filesystem

To get started:

- Create a directory on the server where backups will be stored.

> ⚠ **External Disk**
>
> Having backups reside in the same physical disk as the Cloudron server is dangerous. For this reason, Cloudron will show a warning when you use this provider.

- In the Cloudron dashboard, choose `Filesystem` from the drop down.

The `Use hardlinks` option can be checked to make the Cloudron use hardlinks 'same' files across backups to conserve space. This option has little to no effect when using the `tgz` format.

Filesystem (mountpoint)

Use this provider, when the built-in providers (EXT4, CIFS, NFS, SSHFS) don't work for you.

To get started:

- Setup a mount point manually on the server.
- In the Cloudron dashboard, choose `Filesystem (mountpoint)` from the drop down. This option differs from the `Filesystem` provider in that it checks if the backup directory is mounted before a backup. This check ensure that if the mount is down, Cloudron is not backing up to the local hard disk.

Google Cloud Storage

To get started:

- Create a Cloud Storage bucket following this guide.
- Create a service account key in JSON format.
- In the Cloudron dashboard, choose `Google Cloud Storage` from the drop down.

Hetzner Object Storage

To get started:

- Create a Object Storage bucket following this guide.
- Create S3 API keys
- In the Cloudron dashboard, choose `Hetzner Object Storage` from the drop down.

IDrive e2

To get started:

- Create a IDrive e2 Storage bucket
- Create Access key and Secret Access Key from the IDrive e2 Dashboard
- In the Cloudron dashboard, choose `IDrive e2` from the drop down.

IONOS (Profitbricks)

To get started:

- Create a bucket in the S3 Web Console
- Create Object Storage Keys in the S3 Key Management
- In the Cloudron dashboard, choose `IONOS (Profitbricks)` from the drop down.

Linode Object Storage

To get started:

- Create a Linode Object Storage bucket
- Create Access key and Secret Access Key from the Linode dashboard
- In the Cloudron dashboard, choose `Linode Object Storage` from the drop down.

Minio

To get started:

- Install Minio following the installation instructions.

⚠ Install Minio on another server

Do not setup Minio on the same server as the Cloudron! Using the same server will inevitably result in data loss if something goes wrong with the server's disk. The minio app on Cloudron is meant for storing assets and not backups.

- Create a bucket on Minio using the Minio CLI or the web interface.
- In the Cloudron dashboard, choose `Minio` from the drop down.
    - The `Endpoint` field can also contain a custom port. For example, `http://192.168.10.113:9000` .
    - For HTTPS installations using a self-signed certificate, select the `Accept Self-Signed certificate` option.

NFS

To get started:

- Setup an external NFSv4 server. If you need help setting up a NFSv4 server, see this article or this guide.
- In the Cloudron dashboard, choose `NFS mount` from the drop down.

⚠ Insecure traffic

Please note that NFS traffic is unencrypted and can be tampered. For this reason, you must use NFS mounts only on secure private networks. For backups, we recommend using encryption to make the setup secure.

OVH Object Storage

OVH Public Cloud has OpenStack Swift Object Storage and supports S3 API. Getting S3 credentials is a bit convoluted, but possible as follows:

- Download the OpenStack RC file from horizon interface
- `source openrc.sh` and then `openstack ec2 credentials create` to get the access key and secret
- In the Cloudron dashboard, choose `OVH Object Storage` from the drop down.

Scaleway Object Storage

To get started:

- Create a Scaleway Object Storage bucket.
- Create access key and secret key from the credentials section
- In the Cloudron dashboard, choose `Scaleway Object Storage` from the drop down.

⚠ Storage Class

The Storage Class must be set to `STANDARD` . Setting it to `GLACIER` will result in an error because server side copy operation is not supported in that mode.

SSHFS

To get started:

- Setup an external server and make sure SFTP is enabled in the sshd configuration of the server.
- In the Cloudron dashboard, choose `SSHFS mount` from the drop down.
- Make sure the private ssh-key does not have a password.

Hetzner Storage Box

When using Hetzner Storage Box, the Remote Directory is `/home` for the main account. We have found sub accounts to be unreliable with SSHFS. We recommend using CIFS instead if you want to use subaccounts.

UpCloud Object Storage

To get started:

- Create a UpCloud Object Storage.
- Create a bucket inside the Object Storage.
- Click the S3 API Access link to get access credentials.
- In the Cloudron dashboard, choose `UpCloud Object Storage` from the down down.

⚠ **Multipart copy limitation**

Some regions of UpCloud, like NYC and CHI, do not implement the multipart copy operation. This restriction prevents large files (5GB) from being copied. For `tgz` format, if the backup is more than 5GB, the backup will fail. For `rsync` format, files greater than 5GB will not backup properly.

Vultr Object Storage

To get started:

- Create a Vultr Object Storage bucket
- Make a note of the access key and secret key listed in the bucket management UI.
- In the Cloudron dashboard, choose `Vultr Object Storage` from the drop down.

Wasabi

To get started:

- Create a Wasabi bucket
- Create Access key and Secret Access Key from the Wasabi dashboard
- In the Cloudron dashboard, choose `Wasabi` from the drop down.

XFS

To get started:

- Attach an external XFS hard disk to the server. Depending on where your server is located, this can be a DigitalOcean Block Storage, AWS Elastic Block Store, Linode Block Storage.
- If required, format it using `mkfs.xfs /dev/<device>`. Then, run `blkid` or `lsblk` to get the UUID of the disk.
- In the Cloudron dashboard, choose `XFS Mount` from the drop down.

> **Do not add `/etc/fstab` entry**
>
> When choosing this storage provider, do not add an `/etc/fstab` entry for the mount point. Cloudron will add and manage a systemd mount point.

No-op

This storage backend disables backups. When backups are disabled, updates to apps cannot be rolled back and result in data loss. This backend only exists for testing purposes.

4.3.3 Backup formats

Cloudron supports two backup formats - `tgz` (default) and `rsync`. The `tgz` format stores all the backup information in a single tarball whereas the `rsync` format stores all backup information as files inside a directory.

> **Both formats have the same content**
>
> The contents of the `tgz` file when extracted to disk will be the exact same as the contents of the `rsync` directory. Both the formats are complete and portable.

tgz format

The `tgz` format uploads an app's backup as a gzipped tarball. This format is very efficient when having a large number of small number files.

This format has the following caveats:

- Most Cloud storage API require the content length to be known in advance before uploading data. For this reason, Cloudron uploads big backups in chunks. However, chunked (multi-part) uploads cannot be parallelized and also take up as much RAM as the chunk size.
- `tgz` backup uploads are not incremental. This means that if an app generated 10GB of data, Cloudron has to upload 10GB every time it makes a new backup.

rsync format

The `rsync` format uploads individual files to the backup storage. It keeps track of what was copied the last time around, detects what changed locally and uploads only the changed files on every backup. Note that despite uploading 'incrementally', tgz format can be significantly faster when uploading a large number of small files (like source code repositories) because of the large number HTTP requests that need to be made for each file.

This format has the following caveats:

- By tracking the files that were uploaded the last time around, Cloudron minimizes uploads when using the rsync format. To make sure that each backup directory is "self contained" (i.e can be simply copied without additional tools), Cloudron issues a 'remote copy' request for each file.
- File uploads and remote copies are parallelized.
- When using backends like Filesystem, CIFS, EXT4, NFS & SSHFS, the rsync format can hardlink 'same' files across backups to conserve space. Note that while the protocols themselves support hardlinks, support for hardlinks depends ultimately on the remote file system.
- When encryption is enabled, file names are optionally encrypted.

4.3.4 Encryption

Backups can be optionally encrypted (AES-256-CBC) with a secret key. When encryption is enabled, Cloudron will encrypt both the filename and it's contents.

There are some limitations to lengths of filenames when encryption is enabled:

- File names can be max 156 bytes. See this comment for an explanation. If backups are failing because of `KeyTooLong` errors, you can run the following command in Web Terminal to detect the offending file and rename it to something shorter:

```
cd /app/data
find . -type f -printf "%f\n" | awk '{ print length(), $0 | "sort -rn" }' | less
```

- Backup backends like S3 have max object path length as 1024. There is an overhead of around 20 bytes per file name in a path. So, if you have a directory which is 10 level deep, there is a 200 byte overhead. Filename encryption can be optionally turned off.

Keep password safe

Cloudron does not save a copy of the password in the database. If you lose the password, there is no way to decrypt the backups.

Filenames

When using encryption with the `rsync` format, file names can be optionally encrypted.

Maximum encrypted filename length

Linux file system has a maximum path size of 4096. However, most storage backends have a max key size which is far less. For example, the max size of keys in S3 is 1024. If you have long file names (full path), then you can turn off encryption of file names. In addition, when backing up to a filesystem like EXT4, each path segment is individually encrypted and the length of each encrypted segment must be less than 255.

File format

The Cloudron CLI tool has subcommands like `backup encrypt`, `backup decrypt`, `backup encrypt-filename` and `backup decrypt-filename` that can help inspect encrypted files. See the Decrypt backups for more information.

Four 32 byte keys are derived from the password via scrypt with a hardcoded salt:

- Key for encrypting files
- Key for the HMAC digest of encrypted file
- Key for encrypting file names
- Key for the HMAC digest of the file name for deriving it's IV (see below)

Each encrypted file has:

- A 4 byte magic `CBV2` (Cloudron Backup v2)
- A 16 byte IV. This IV is completely random per file.
- File encrypted used AES-256-CBC
- A 32 byte HMAC of the IV and encrypted blocks

Each encrypted filename has:

- A 16 byte IV. This IV is derived from HMAC of the filename. This is done this way because the sync algorithm requires the encryption to be deterministic to locate the file upstream.
- Filename encrypted using AES-256-CBC

4.3.5 Schedule

The backup schedule & retention policy can be set in the `Backups` view.

The `Backup Interval` determines how often you want the backups to be created. If a backup fails (because say the external service is down or some network error), Cloudron will retry sooner than the backup interval. This way Cloudron tries to ensure that a backup is created for every interval duration.

- The backup process runis with a nice of 15. This makes sure that it gets low priority if the Cloudron is doing other things.
- The backup task runs with a configurable memory limit. This memory limit is configured in `Backups` -> `Configure` -> `Advanced`.
- There is currently a timeout of 12 hours for the backup to complete.

4.3.6 Retention Policy

The `Retention Policy` determines how backups are retained. For example, a retention policy of 1 week means that all backups older than a week are deleted. The policy `7 daily` means to keep a single backup for each day for the last 7 days. So, if 5 backups were created today, Cloudron will remove 4 of them. It does **not** mean to keep 7 backups a day. Similarly, the term `4 weekly` means to keep a single backup for each week for the last 4 weeks.

The following are some of the important rules used to determine if a backup should be retained:

- For installed apps and box backups, the latest backup is always retained regardless of the policy. This ensures that even if all the backups are outside of the retention policy, there is still at least one backup preserved. This change also ensure that the latest backup of stopped apps is preserved when not referenced by any box backup.
- An App backup that was created right before an app updates is also marked as special and persisted for 3 weeks. The rationale is that sometimes while the app itself is working fine, some errors/bugs only get noticed after a couple of weeks.
- For uninstalled apps, the latest backup is removed as per the policy.
- If the latest backup is already part of the policy, it is not counted twice.
- Errored and partial backups are removed immediately.

4.3.7 Cleanup Backups

The `Backup Cleaner` runs every night and removes backups based on the Retention Policy.

Cloudron also keeps track of the backups in it's database. The `Backup Cleaner` checks if entries in the database exist in the storage backend and removes stale entries from the database automatically.

You can trigger the `Backup Cleaner` using the `Cleanup Backups` button:

If you click on the Logs button after triggering `Cleanup Backups`, you will see the exact reason why each individual backup is retained. In the logs, a `box_` prefix indicates that it is a full Cloudron backup where `app_` prefix indicates that it is an app backup.

- `keepWithinSecs` means the backup is kept because of the retention policy.
- `reference` means that this backup is being referenced by another backup. When you make a full Cloudron backup, it takes the backup of each app as well. In this case, each app backup is "referenced" by the parent "box" backup.
- `preserveSecs` means the backup is kept because it is the backup of a previous version of the app before an app update. We keep these backups for 3 weeks in case an update broke something and it took you some time to figure that something broke.

Preserve specific backups

See backup labels section on how to preserve specific backups regardless of the retention policy.

Old Local Backups

By default, Cloudron stores backups in the filesystem at `/var/backups`. If you move backups to an external location, previous backups have to be deleted manually by SSHing into the server.

- SSH into the server.
- Run `cd /var/backups` to change directory.
- There may be several timestamped directories. You can delete them using `rm -rf /var/backups/<timestamped-directory>`.
- The `snapshot` subdirectory can be removed using `rm -rf /var/backups/snapshot`.

4.3.8 Backup Labels

App Backups can be tagged with a label for readability. Use the `Edit` button to change a backup's label.

In addition, specific backups can be preserved for posterity using the preserve checkbox:

4.3.9 Snapshot App

To take a backup of a single, click the `Create Backup` button in the `Backups` section of the app's configure UI.

4.3.10 Concurrency Settings

When using one of the cloud storage providers (S3, GCS), the upload, download and copy concurrency can be configured to speed up backup and restore operations.

- Upload concurrency - the number of file uploads to be done in parallel.
- Download concurrency - the number of file downloads to be done in parallel.
- Copy concurrency - the number of remote file copies to be done in parallel. Cloudron conserves bandwidth by not re-uploading unchanged files and instead issues a remote file copy request.

There are some caveats that you should be aware of when tuning these values.

- Concurrency values are highly dependent on the storage service. These values change from time to time and as such it's not possible to give a standard recommendation for these values. In general, it's best to be conservative since backup is just a background task. Some services like Digital Ocean Spaces can only handle 20 copies in parallel before you hit rate limits. Other provides like AWS S3, can comfortably handle 500 copies in parallel.
- Higher upload concurrency necessarily means you have to increase the memory limit for the backup.

4.3.11 Snapshot Cloudron

To take a backup of Cloudron and all the apps, click the `Backup now` button in the `Settings` page:

⚠ **Warning**

When using the `no-op` backend, no backups are taken. If backups are stored on the same server, be sure to download them before making changes in the server.

4.3.12 Disable automatic backups

An app can be excluded from automatic backups from the 'Advanced settings' in the Configure UI:

Note that the Cloudron will still create backups before an app or Cloudron update. This is required so that it can be reverted to a sane state should the update fail.

⚠ **Warning**

Disabling automatic backup for an app puts the onus on the Cloudron adminstrator to backup the app's files regularly. This can be done using the Cloudron CLI tool's `cloudron backup create` command.

4.3.13 Clone app

To clone an app i.e an exact replica of the app onto another domain, first create an app backup and click the clone button of the corresponding backup:

This will bring up a dialog in which you can enter the location of the new cloned app:

4.3.14 Restore app

Apps can be restored to a previous backup by clicking on the `Restore` button.

✎ **Both data and code are reverted**

Restoring will also revert the code to the version that was running when the backup was created. This is because the current version of the app may not be able to handle old data.

4.3.15 Import App Backup

Migrating apps or moving apps from one Cloudron to another works by first creating a backup of the app on the old Cloudron, optionally copying the backup to the new Cloudron's server and importing the backup on the new Cloudron. You can also use this approach to resurrect an uninstalled app from it's backup.

Use the following steps will migrate an app:

- First, create a backup of the app on the old Cloudron
- If the old Cloudron is backing up to the filesystem, copy the backup of this app to the new server. You can determine the backup id using the `Copy to Clipboard` action in the Backups view. You can use a variety of tools like scp, rclone, rsync to copy over the backup depending on your backup configuration.
- If the old Cloudron is **not** backing up to the filesystem, download the backup configuration of this backup. This is simply a file that helps copy/paste the backup configuration setting to the new server.

- Install a new app on the new Cloudron. When doing so, make sure that the version of the app on the old cloudron and new cloudron is the same.
- Go to the `Backups` view and click on `Import`.
- You can now upload the backup configuration which you downloaded from the previous step to auto-fill the import dialog.
- Alternately, enter the credentials to access the backup.

> **Backup Path**
>
> Backup path is the **relative path** to the backup. It's usually of the form `path/to/<timestamp>/app_xx`.

4.3.16 Restore Email

There is currently no built-in way to restore specific emails or email folders, but this can be done manually using the process below. Mail is backed up in this directory when viewed uncompressed (is using tgz, otherwise is always uncompressed if rsync): `<backupMount>/snapshot/box/mail/vmail/<mailbox>/mail/*`

The example scenario: A user deleted a folder of emails in their mailbox and needs the folder (and it's emails) restored, the folder is called "CriticalEmails".

1. SCP to the backup disk or service
2. Locate the mail folder that the user may have deleted. In the example above, we would find this missing folder located at `<backupMount>/snapshot/box/mail/vmail/<mailbox>/mail/.CriticalEmails`
3. Copy that folder (replacing with the actual email address needed) to this location on your Cloudron disk: `/home/yellowtent/boxdata/mail/vmail/<mailbox>/mail/.CriticalEmails` and ensure the permissions match that of the other folders (should be `drwxr--r-- ubuntu:ubuntu`)
4. Restart the mail service in Cloudron

The user should now be able to see the mail folder named "CriticalEmails" (in this example) and all the emails associated with that folder.

4.3.17 Restore Cloudron

To restore from a backup:

- If you have the old Cloudron around, go to `Backups` and download the latest backup configuration. If you don't have access to the old Cloudron, then you have to first determine the Box Backup ID. Get the timestamp and filename by following this guide. The Backup ID is of the form `<timestamp>/<box backup filename>` e.g. `2024-08-21-010015-708/box_v8.0.3.tar.gz` (or without .tar.gz for rsync).
- Install Cloudron on a new server with Ubuntu LTS (20.04/22.04/24.04):

```
wget https://cloudron.io/cloudron-setup
chmod +x cloudron-setup
./cloudron-setup --version x.y.z # version must match your backup version
```

Backup & Cloudron version

The Cloudron version and backup version must match for the restore to work. If you installed a wrong version by mistake, it's easiest to just start over with a fresh Ubuntu installation and re-install Cloudron.

- If your domains use Wildcard, Manual or No-op DNS provider, you should manually switch the DNS of the domains to the new server's IP. At the minimum, you have to change the IP of the dashboard domain (i.e `my.domain.com`). Note that if you do not switch the IP for the app domains, the restore of those apps will fail and you have to trigger the restore of the app from the Backups section when you are ready to switch the IP.
- Navigate to `http://<ip>` and click on `Looking to restore` located at the bottom:
- Provide the backup information to restore from. If you downloaded the backup configuration from your previous installation, you can upload it here to fill up the fields.

Alternately, you can just fill up the form by hand:

Warning

When using the filesystem provider, ensure the backups are owned by the `yellowtent` user. Also, ensure that the backups are in the same file system location as the old Cloudron.

- Cloudron will download the backup and start restoring:

The new Cloudron server is an exact clone of the old one - all your users, groups, email, apps, DNS settings, certificates, will be exactly as-is before. The main exception is the backup settings which is not restored and will be set to the config that was provided in the restore UI. For this, reason, be sure to re-verify your backup configuration after restore.

Graphs and performance data is also not persisted across a migration because the new server characteristics are most likely totally different from the old server.

Dry Run

When you restore Cloudron, Cloudron will automatically update the DNS to point to the new server. Using the `Dry run` feature you can skip the DNS setup. This allows you to test your backups or get a feel of how your apps might perform if you switch the server, without affecting your current installation.

To do a dry run of Cloudron restore:

- Create an entry in your `/etc/hosts` file. The `/etc/hosts` overrides DNS and these entries will direct your machine to go this new server, instead of your existing server when you visit the dashboard domain. Note that this entry has to made on your PC/Mac and not on the Cloudron server. In addition, these entries only affect your PC/Mac and not any other device. Assuming, `1.2.3.4` is your new server IP, add an entry like this:

```
# this makes the dashboard accessible
1.2.3.4 my.cloudrondomain.com

# add this for every app you have/want to test.
```

```
1.2.3.4 app1.cloudrondomain.com
1.2.3.4 app2.cloudrondomain.com
```

- Follow the steps in Restore cloudron. Check the `Dry run` checkbox:
- Once restored, the browser will automatically navigate to `https://my.cloudrondomain.com`. It is important to check that this is indeed the new server and not your old server! The easiest way to verify this is to go the `Network` view and check the IP Address. If the IP is still of the old server, this is most likely a browser DNS caching issue. Generally, restarting the browser or trying to access the browser in anonymous mode resolves the issue.
- If you want to make the "switch" to the new server, go to the `Domains` view and click on `Sync DNS`.
- You can now remove the entries in `/etc/hosts` and shutdown your old server.

4.3.18 Move Cloudron to another server

If the existing server's cpu/disk/memory can be resized (as is the case with most server providers), then simply resize it and reboot the server. Cloudron will automatically adapt to the available resources after a server resize.

To migrate to a different server or move Cloudron to a different server provider:

- Take a complete backup of the existing Cloudron. Click the `Backup now` button in the `Settings` page.

Backup location

We recommend backing up to an external service like S3 or Digital Ocean Spaces. This is because the backups become immediately available for the new server to restore from. If you use the filesystem for backups, you have to copy the backup files manually to the new server using rsync or scp.

- Download the backup configuration of the newly made backup. This is simply a file that helps copy/paste the backup configuration setting to the new server.
- Follow the steps to restore Cloudron on the new server. It is recommended to not delete the old server until migration to new server is complete and you have verified that all data is intact (instead, just power it off).

Backup & Cloudron version

The Cloudron version and backup version must match for the restore to work. To install a specific version of Cloudron, pass the `version` option to `cloudron-setup`. For example, `cloudron-setup --version 3.3.1`.

4.4 Branding

4.4.1 Overview

The `Branding` view can be used to customize various aspects of the Cloudron like it's name, logo and footer. The `Branding` view is only accessible by a Cloudron superadmin.

4.4.2 Cloudron Name

The Cloudron name in the `Branding` view. The name is used in the following places:

- Email templates - user invitations and notifications
- Dashboard header/navbar
- Login page
- OIDC login button of apps. Support for this depends on the app.

4.4.3 Cloudron Logo

The Cloudron logo can be changed by clicking on the logo in the `Branding` view. The avatar is used in the following places:

- Email templates - user invitations and notifications
- Dashboard header/navbar
- Login page

4.4.4 Background Image

The background image for the login pages can be set in the `Branding` view.

4.4.5 Custom Pages

Custom pages can be setup under `/home/yellowtent/boxdata/custom_pages`.

File	Description
`app_not_responding.html`	When any app becomes unresponsive
`notfound.html`	Used when the user navigates with the server's IP address. This is also used if you set the DNS of an arbitrary domain to the server with no associated app.

4.4.6 Footer

The footer can be branded with Markdown from the `Branding` page.

The footer can be templated using the following variables:

- `%YEAR%` - the current year
- `%VERSION%` - the current Cloudron version

4.5 Certificates

4.5.1 Overview

Cloudron integrates with Let's Encrypt to install certificates for apps. Certificates are renewed automatically.

4.5.2 Certificate Providers

Cloudron supports the following certificate providers:

- `Let's Encrypt Prod` - Obtain individual certs for each domain. This provider uses HTTP automation and requires inbound port 80 to be open. This provider will list your individual domain names in the Certificate transparency project.
- `Let's Encrypt Prod - Wildcard` (default) - Obtain wildcard certs for each domain. This provider uses DNS automation and can only be used with programmatic DNS API providers.
- `Let's Encrypt Staging` - Obtain individual certs for each domain from Let's Encrypt staging endpoint. These certs are for testing and not trusted by the browser. This provider uses HTTP automation and requires inbound port 80 to be open.
- `Let's Encrypt Staging - Wildcard` - Obtain wildcard certs for each domain from Let's Encrypt staging endpoint. These certs are for testing and not trusted by the browser. This provider uses DNS automation and can only be used with programmatic DNS API providers.
- `Self-Signed` / `Custom Certificate` - Disable Let's Encrypt integration and use a custom wildcard certificate instead.

Certificate provider can be set per-domain from the `Domains` view under the domain's Advanced settings.

4.5.3 Custom certificates

Wildcard certificate

A custom wildcard certificate can be provided per domain in advanced settings of a domain in the `Domains` view. When setting such a certificate, make sure to add both the bare domain and the wildcard domain as part of the certificate.

Follow this tutorial for instructions on how to generate a custom wildcard certificate that has both the bare domain and the wildcard domain.

Intermediate certs

You can upload a certificate chain by simply appending all the intermediate certs in the same cert file.

App certificate

Custom certificates can also be set for each installed application using the REST API. This can be used to set an Extended Validation (EV) certificate for an app. For example, assuming we have the PEM encoded files `cert.pem` and `key.pem`:

```
# first encode the newlines to send as JSON
key=$(perl -pe 's/\n/\\\n/' key.pem)
cert=$(perl -pe 's/\n/\\n/' cert.pem)

curl -X POST -H "Content-Type: application/json" -d "{ \"cert\": \"${cert}\", \"key\": \"${key}\" }" https://my.cloudron.xyz/api/v1/apps/5555f553-96ad-46c9-
ba42-13d08ecb86a0/configure?access_token=3f1e6d8e5ece3f3dbdefd88679fdd270b00223b58ce6781990cf95e444b7c7f3
```

In the example above, `my.example.com` is the Cloudron domain. `5555f553-96ad-46c9-ba42-13d08ecb86a0` is the app id obtained from the `Updates` section of the app. API tokens can be created in the profile view.

Intermediate certs

You can upload a certificate chain by simply appending all the intermediate certs in the same cert file.

4.5.4 Certificate transparency

Let's Encrypt participates in Certificate transparency. This means that your apps and subdomains are discoverable via the Certificate transparency project (crt.sh and Google's website). Some hackers take advantage of this to hack web applications before they are in installed.

For this reason, we recommend that you use Wildcard certificates. When using Wildcard certificates, the subdomain information is not 'leaked'. Note that Let's Encrypt only allows obtaining wildcard certificates using DNS automation. Cloudron will default to obtaining wildcard certificates when using one of the programmatic DNS API providers.

4.5.5 Renewal

Automatic renewal

Cloudron attempts to start renewing certificates automatically 1 month before expiry of the certificate. If renewal fails, a notification email will be sent to the Cloudron administrators. When the certificate expires, Cloudron will start using fallback certificates for the app.

Manual renewal

To instantly trigger renewal of Let's encrypt certificate, click the `Renew All` button on the domains page.

4.5.6 Revokation

Cloudron does not revoke certificates when an app is uninstalled. Instead, it retains the certificate, so that it can be reused if another app is installed in the same subdomain. This allows you to install apps for testing in the same location, say `test`, and not have to worry about running over the Let's Encrypt rate limit.

4.5.7 CAA records

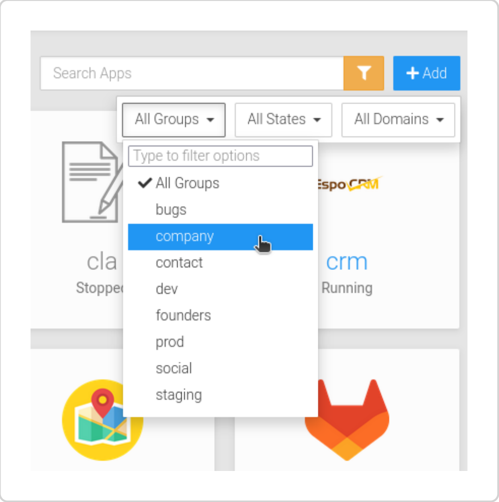
Starting Sep 2017, Let's Encrypt will check for CAA records to validate if the domain owner has authorized the CA to issue certificates for the domain. For this reason, make sure that either the CAA record for the domain is empty OR setup a CAA record allowing `letsencrypt.org`.

4.6 Dashboard

4.6.1 Filter

Apps in the dashboard can be filtered using one or more of the following filters:

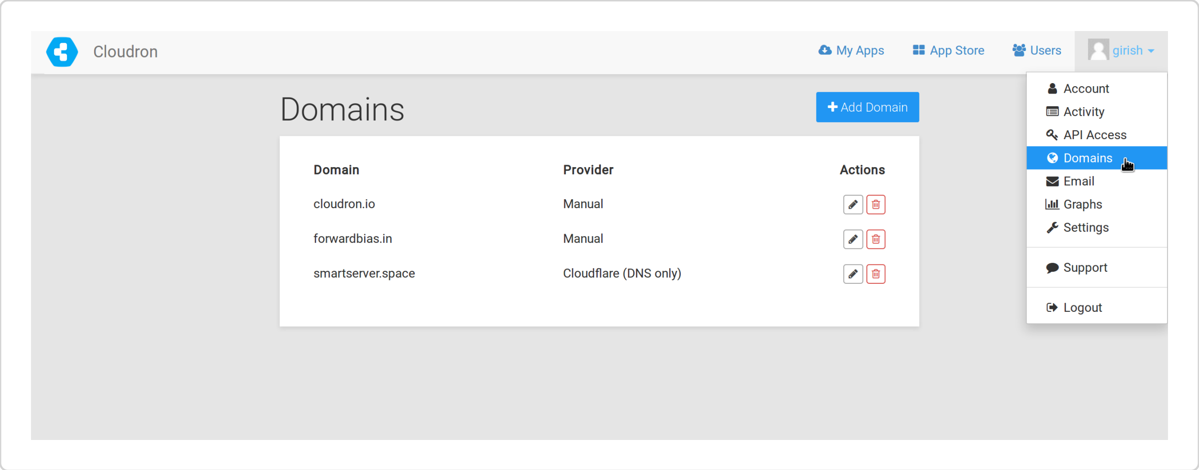
- Group name
- App State
- Domain name



4.7 Domains

4.7.1 Overview

The `Domains` view can be used to add a domain for use by Cloudron. Once added, apps can be installed as subdomains of the added domain. The Cloudron Email Server can also be enabled on a per-domain basis.



Domains

Domain	Provider	Actions
cloudron.io	Manual	
forwardbias.in	Manual	
smartserver.space	Cloudflare (DNS only)	

Cloudron integrates with various DNS service APIs to automate DNS setup. Using the API, Cloudron can also get Wildcard certificates via Let's Encrypt. This approach helps in hiding an app's domain from the Certificate Transparency Log.

If your DNS provider is not supported yet, we recommend using the Wildcard provider.

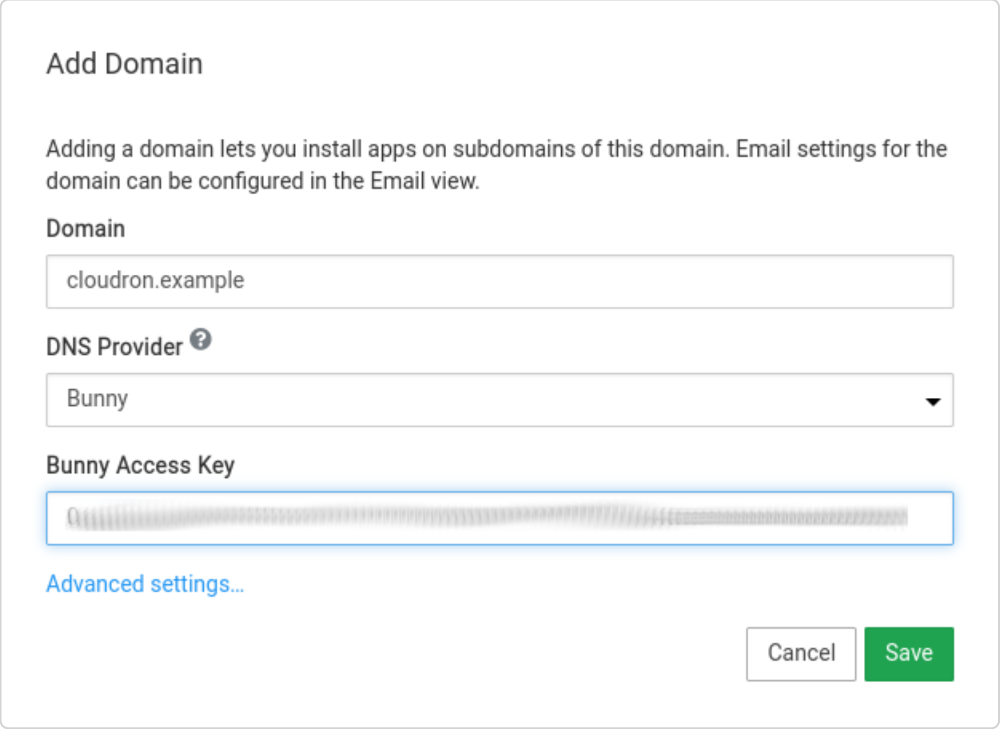
It is safe to add the same domain across multiple Cloudrons. This way, apps using the same top level domain can be installed on separate servers.

4.7.2 DNS providers

Bunny DNS

To get started:

- Ensure that your domain is hosted on Bunny.
- Once you domain is on Bunny, get the Access Key
- In the Cloudron dashboard, choose Bunny from the drop down and provide the Access key.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron.example

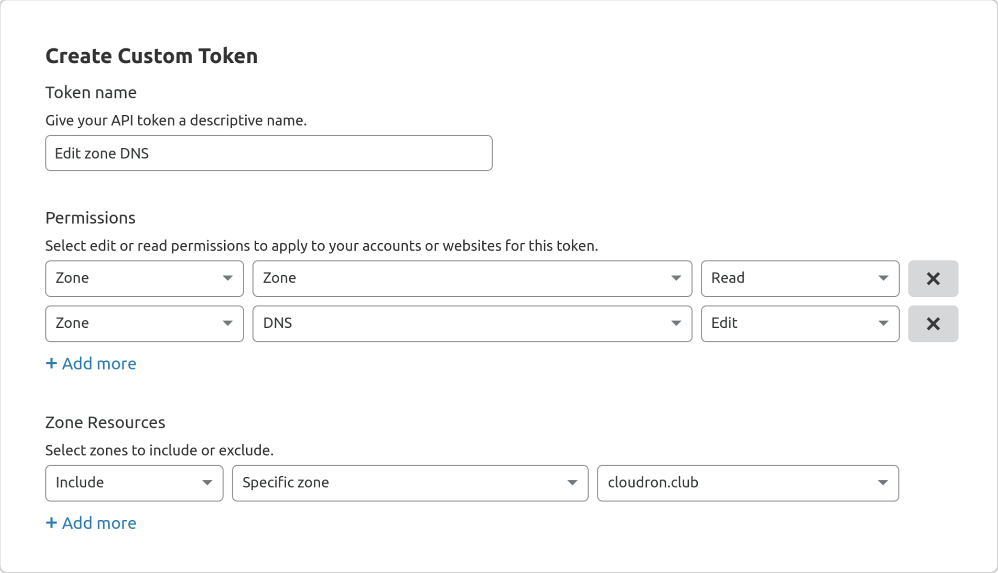
DNS Provider: Bunny

Bunny Access Key

Cloudflare DNS

To get started:

- Ensure that your domain is hosted on Cloudflare. If your domain is not hosted on Cloudflare, you can follow the Cloudflare 101 guide.
- Once your domain is on Cloudflare, get the Global API Key or API Token available in the profile section of your account.
- Choose the `Edit zone DNS` template to create a token.
- When using the API token, it must include the `Zone:Read` and `DNS:Edit` permission. The token only needs permission for the specific zone as shown below:



Create Custom Token

Token name

Give your API token a descriptive name.

Edit zone DNS

Permissions

Select edit or read permissions to apply to your accounts or websites for this token.

Zone	Zone	Read	✕
Zone	DNS	Edit	✕

+ Add more

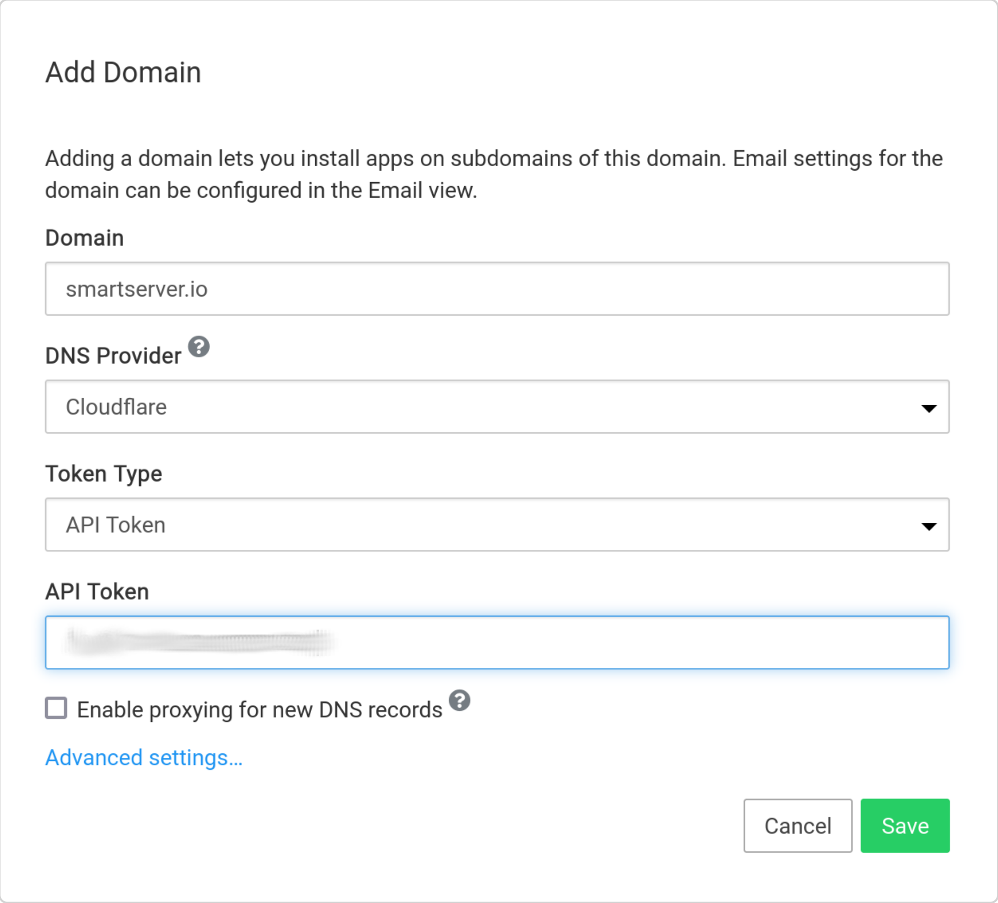
Zone Resources

Select zones to include or exclude.

Include	Specific zone	cloudron.club

+ Add more

- In the Cloudron dashboard, choose Cloudflare from the drop down and provide the API key.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: smartserver.io

DNS Provider: Cloudflare

Token Type: API Token

API Token: [redacted]

☐ Enable proxying for new DNS records

Advanced settings...

- In Cloudflare crypto configuration, set SSL to "Full SSL (Strict) mode". Users have often reported redirect loops without this setting.

Proxying

New DNS records are configured for proxying HTTP requests based on the `Enable proxying  for new DNS records` checkbox. Please note that all your traffic is readably by Cloudflare when proxying is enabled. When enabled, we recommend also setting up Full SSL (Strict) mode.

Email and HTTP Proxy

If you use Cloudflare for your primary domain and enable Cloudron email for any domain, Cloudflare proxying must be disabled for the `my` subdomain. This is because Cloudflare will only proxy HTTP and not email protocol.

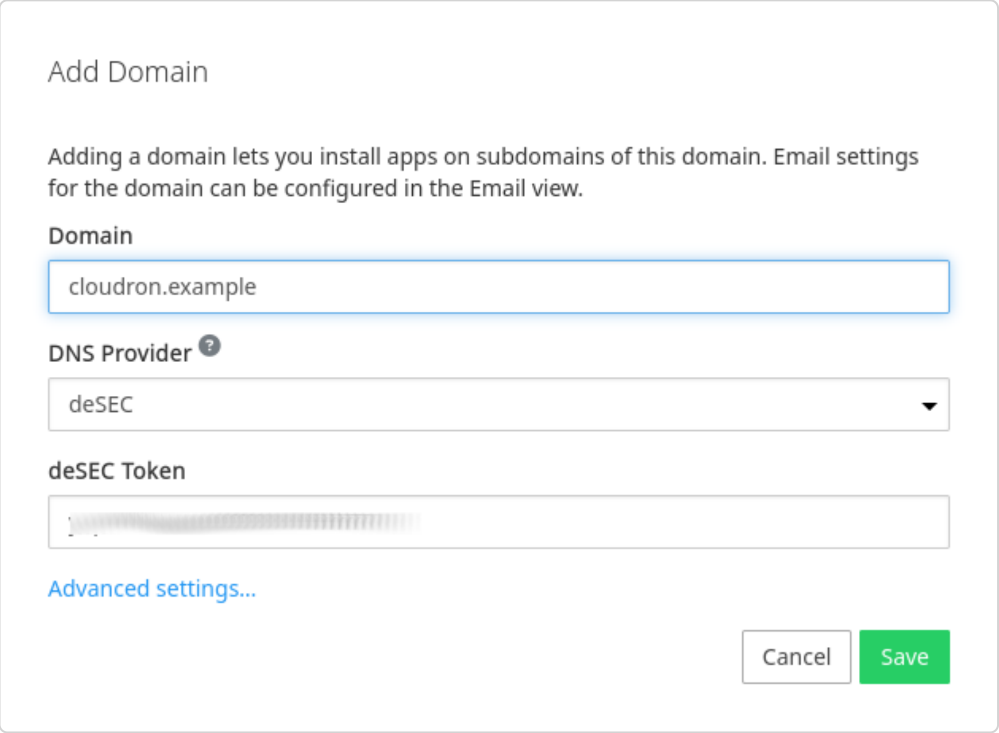
Subdomains with HTTPS Proxy

Cloudflare universal certificates only support one level of subdomain with HTTPS proxying on the free plan. See this forum thread for more information.

deSEC DNS

To get started:

- Ensure that your domain is hosted on deSEC.
- Once your domain is on deSEC DNS, create an token.
- In the Cloudron dashboard, choose deSEC from the drop down and provide the token.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron.example

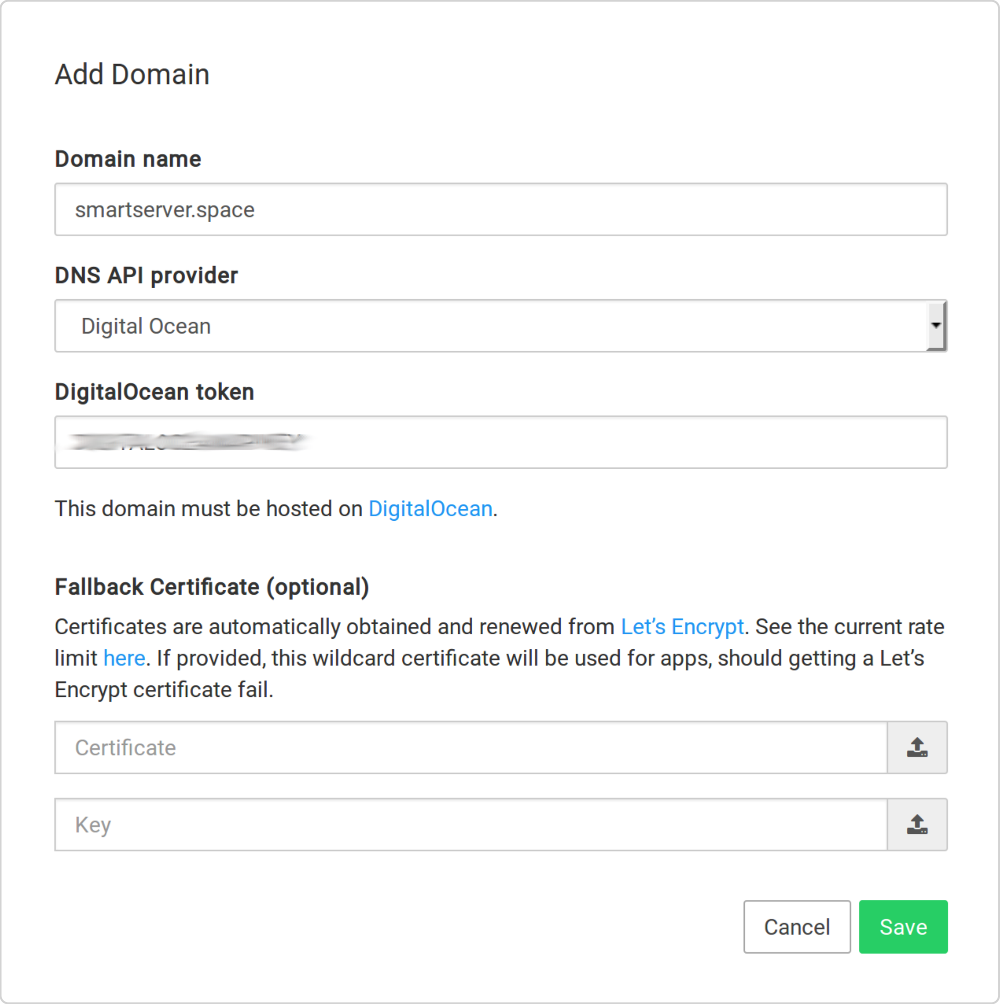
DNS Provider: deSEC

deSEC Token

DigitalOcean DNS

To get started:

- Ensure that your domain is hosted on DigitalOcean. If your domain is not hosted in DigitalOcean, you can follow this tutorial to point your domain's nameservers to DigitalOcean nameservers.
- Once your domain is on DigitalOcean DNS, create an APIv2 token with read+write access.
- In the Cloudron dashboard, choose DigitalOcean from the drop down and provide the API key.



Add Domain

Domain name

smartserver.space

DNS API provider

Digital Ocean

DigitalOcean token

[redacted]

This domain must be hosted on DigitalOcean.

Fallback Certificate (optional)

Certificates are automatically obtained and renewed from Let's Encrypt. See the current rate limit here. If provided, this wildcard certificate will be used for apps, should getting a Let's Encrypt certificate fail.

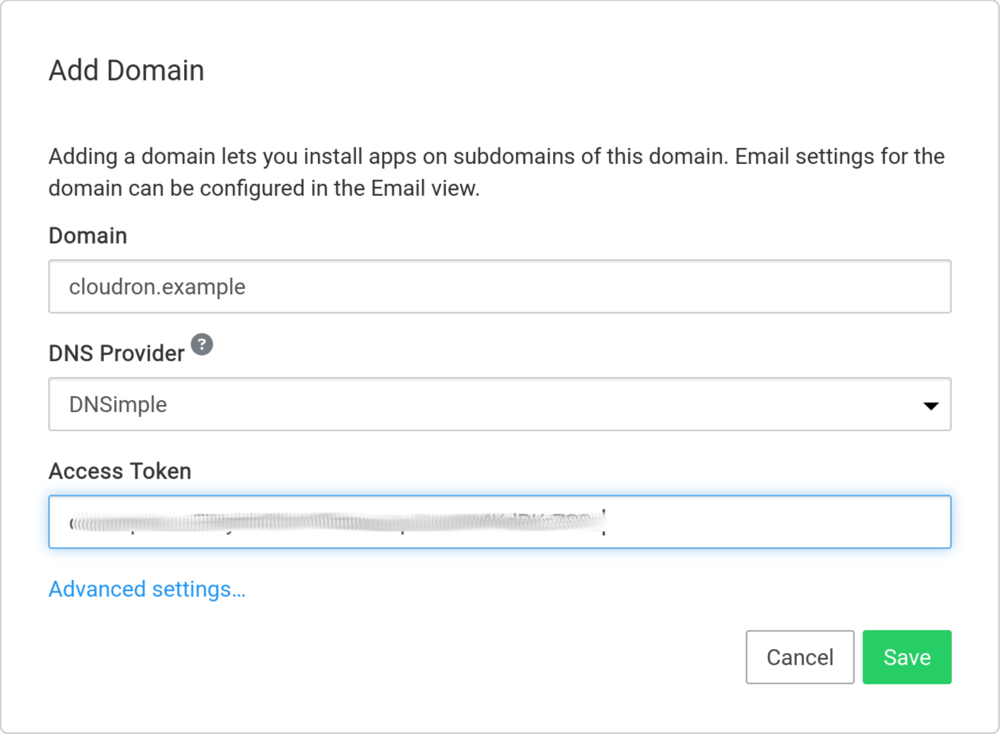
Certificate

Key

DNSimple DNS

To get started:

- Ensure that your domain is hosted on DNSimple. If your domain is not hosted in DNSimlple, you can follow these articles.
- Once your domain is on DNSimple DNS, create an API token
- In the Cloudron dashboard, choose DNSimple from the drop down and provide the API key.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain

cloudron.example

DNS Provider ?

DNSimple

Access Token

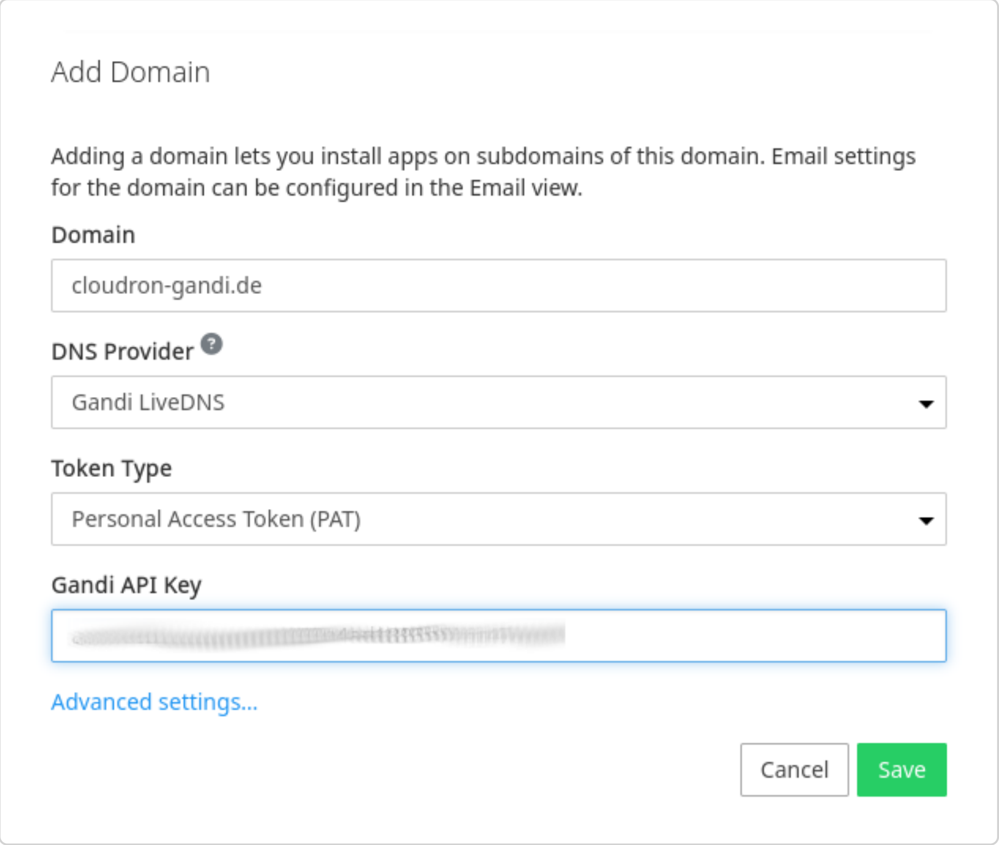
[redacted]

Gandi LiveDNS

If you purchased a domain from Gandi, you can use Gandi LiveDNS for DNS integration.

To get started:

- Ensure that your domain is using LiveDNS. Go the `DNS Records` page in Gandi to verify that you are using LiveDNS. Old domains might see a `Switch to LiveDNS` banner. If you see this, click on `Gandi's DNS Records` button and wait for 2-3 hours for Gandi to migrate the domain.
- Next, create an LiveDNS API key from the security section. Choose `Authentication options` and scroll down to generate `Personal Access Token (PAT)`. When generating the token, you can provide a specific domain name and select `Manage domain name technical configurations` permission.
- In the Cloudron dashboard, choose Gandi LiveDNS from the drop down and provide the PAT.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron-gandi.de

DNS Provider: Gandi LiveDNS

Token Type: Personal Access Token (PAT)

Gandi API Key:

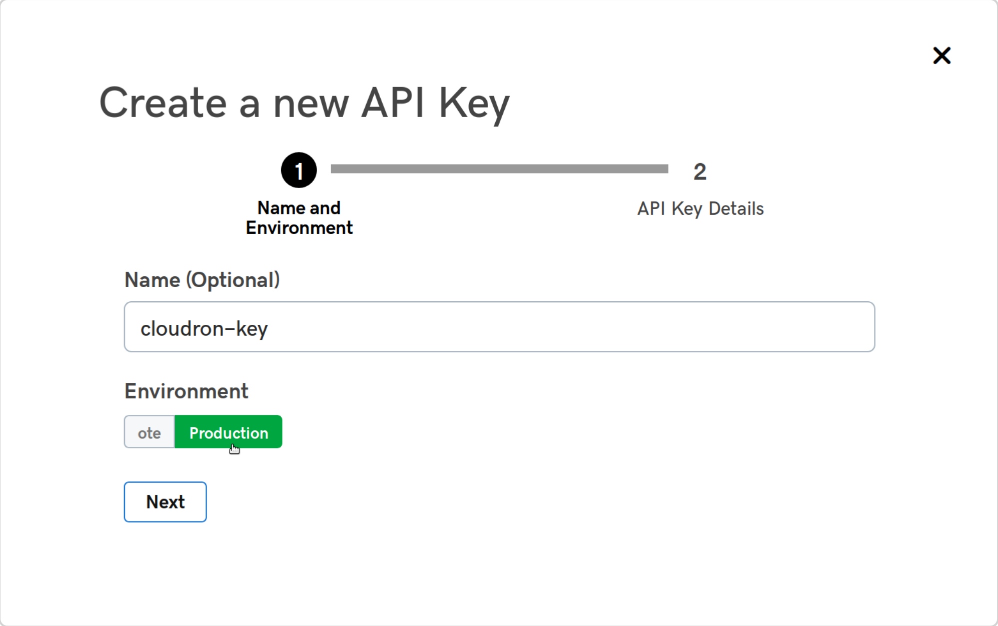
GoDaddy

> ⚠ **Restricted API access**
>
> As of 2024, access to parts of the Domains API is limited to accounts with 50 or domains.

If your domain is registered with GoDaddy, you can use Cloudron's GoDaddy DNS backend to manage the DNS.

To get started:

- Create a GoDaddy API Key at their developer portal. When creating a new key **select production environment**.



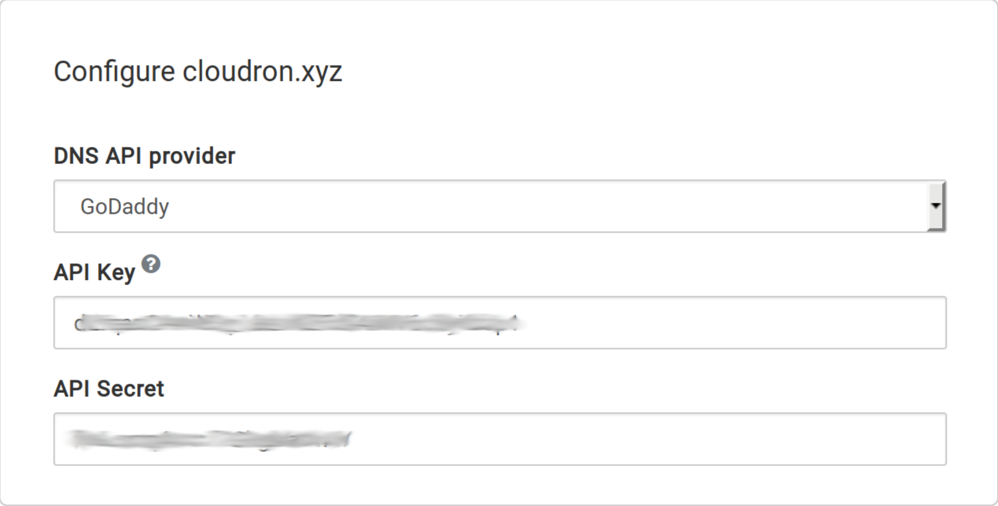
Create a new API Key

Name (Optional)

cloudron-key

Environment

- In the Cloudron dashboard, choose GoDaddy from the drop down and provide the key and secret.



Configure cloudron.xyz

DNS API provider

GoDaddy

API Key

[redacted]

API Secret

[redacted]

No Delete Record API

GoDaddy does not have an API to delete records. For this reason, Cloudron sets deleted records to `0.0.0.0`.

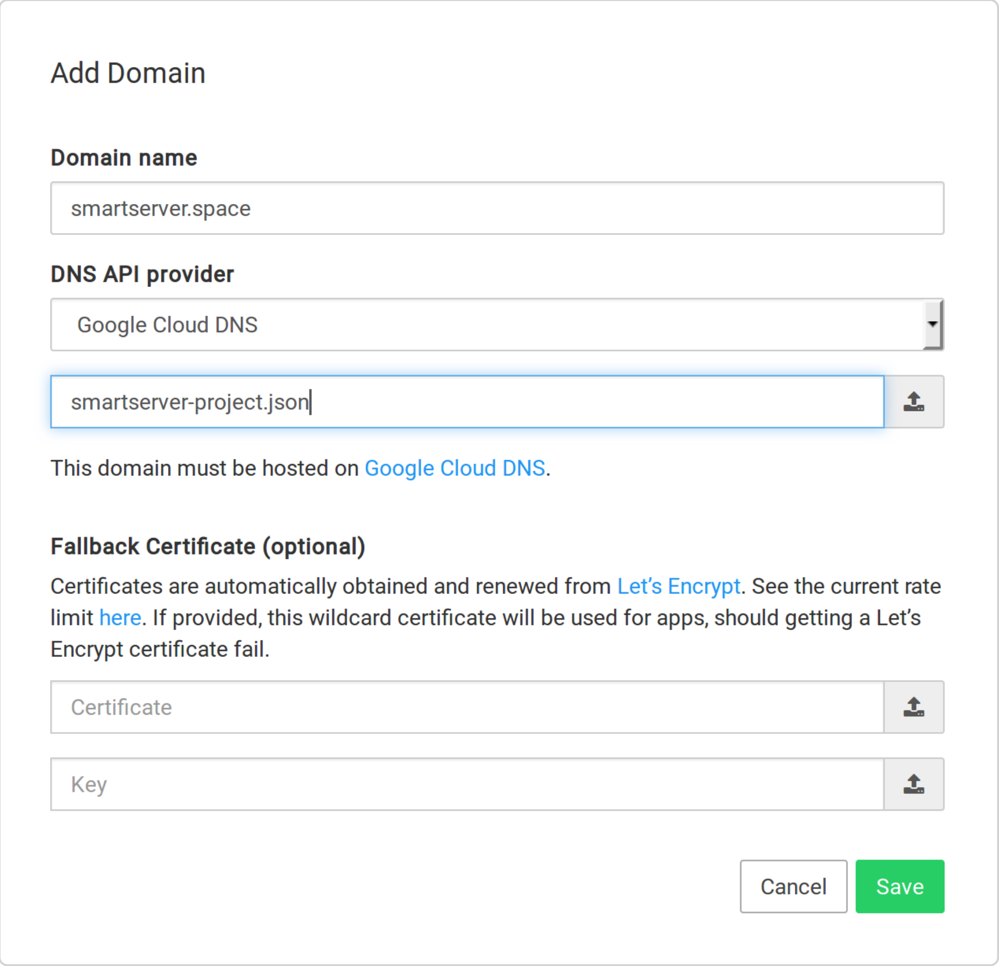
Google Cloud DNS

To get started:

- Ensure that your domain is hosted on Google Cloud DNS. You can move your existing domain to use the Cloud DNS by following this guide.

> ⚠ **Google Domains**
>
> Google Domains is a different product than Google Cloud DNS. The above guide gives directions on how to make a Google Domains hosted domain use the Google Cloud DNS

- Create a service account key in JSON format.
- In the Cloudron dashboard, choose Google Cloud DNS from the drop down.



Add Domain

Domain name

smartserver.space

DNS API provider

Google Cloud DNS

smartserver-project.json

This domain must be hosted on Google Cloud DNS.

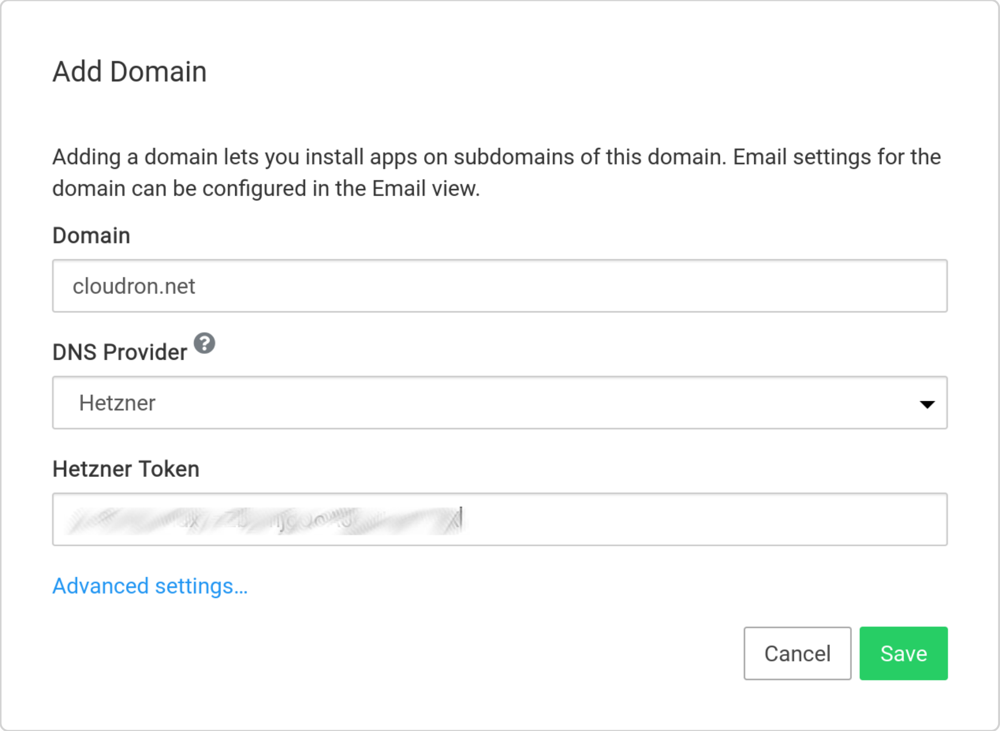
Fallback Certificate (optional)

Certificates are automatically obtained and renewed from Let's Encrypt. See the current rate limit here. If provided, this wildcard certificate will be used for apps, should getting a Let's Encrypt certificate fail.

Hetzner DNS

To get started:

- Ensure the domain is hosted using Hetzner DNS. If not, you can follow this guide.
- Create a Hetzner API token.
- In the Cloudron dashboard, choose Hetzner DNS from the drop down.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron.net

DNS Provider: Hetzner

Hetzner Token

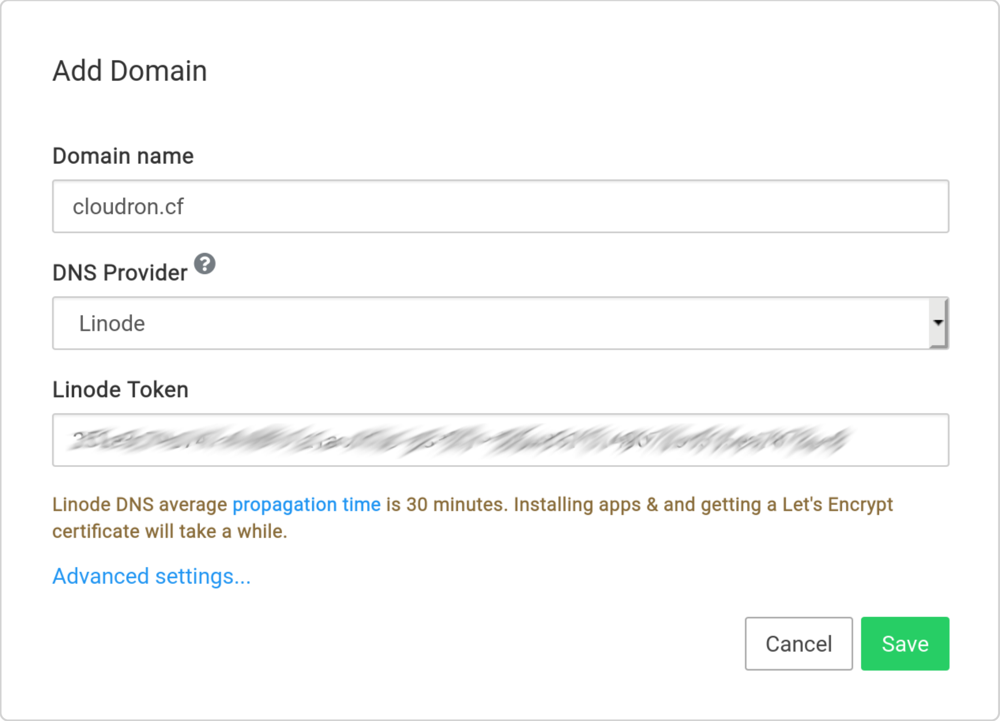
Robot vs KonsoleH

Cloudron only supports domains that have their name servers set to Robot (`*.ns.hetzner.com`). Domains registered via KonsoleH tend to have name severs like `*.your-server.de` . Please switch over to Robot name servers for the integration to work.

Linode DNS

To get started:

- Ensure the domain is hosted using Linode DNS. If not, you can follow this guide to migrate an existing domain to use the service.
- Create a Linode API token. The Personal Access Token must have `Domains` access. Set the `Expiry` to `Never`.
- In the Cloudron dashboard, choose Linode from the dropdown.



Add Domain

Domain name: cloudron.cf

DNS Provider: Linode

Linode Token: [redacted]

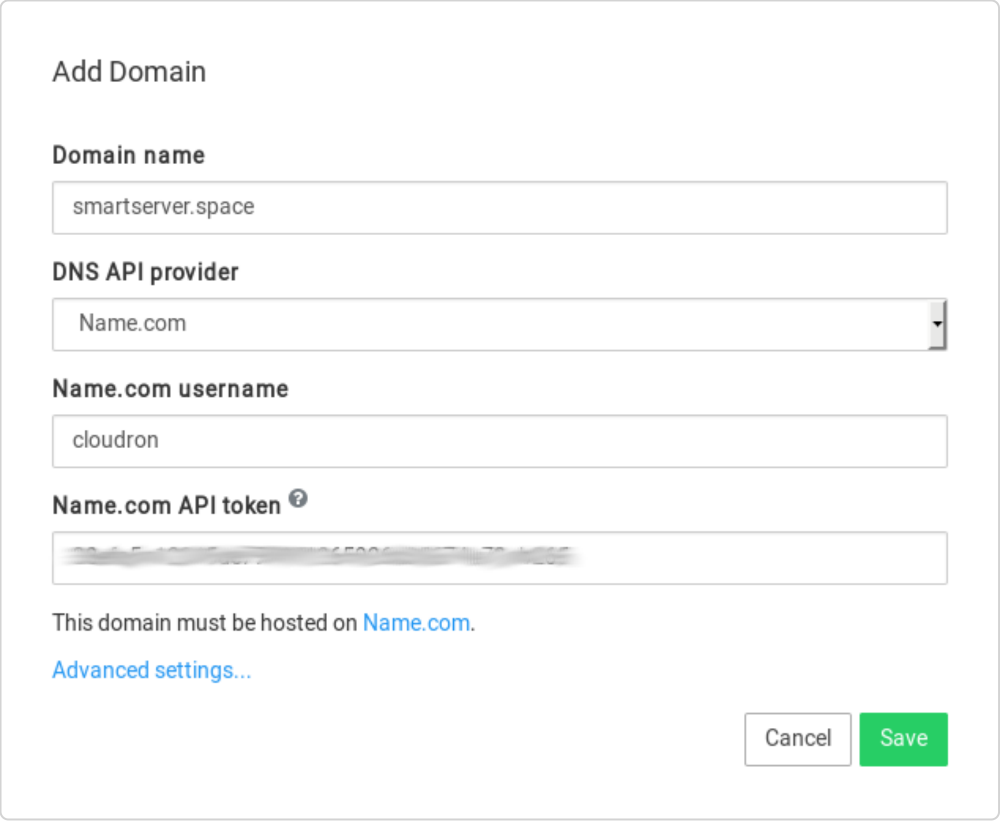
Linode DNS average propagation time is 30 minutes. Installing apps & and getting a Let's Encrypt certificate will take a while.

Name.com DNS

If your domain is registered with name.com, you can use Cloudron's Name.com DNS backend to manage the DNS.

To get started:

- Create a name.com API token
- In the Cloudron dashboard, choose Name.com from the dropdown.



Add Domain

Domain name

smartserver.space

DNS API provider

Name.com

Name.com username

cloudron

Name.com API token

[redacted]

This domain must be hosted on Name.com.

Namecheap DNS

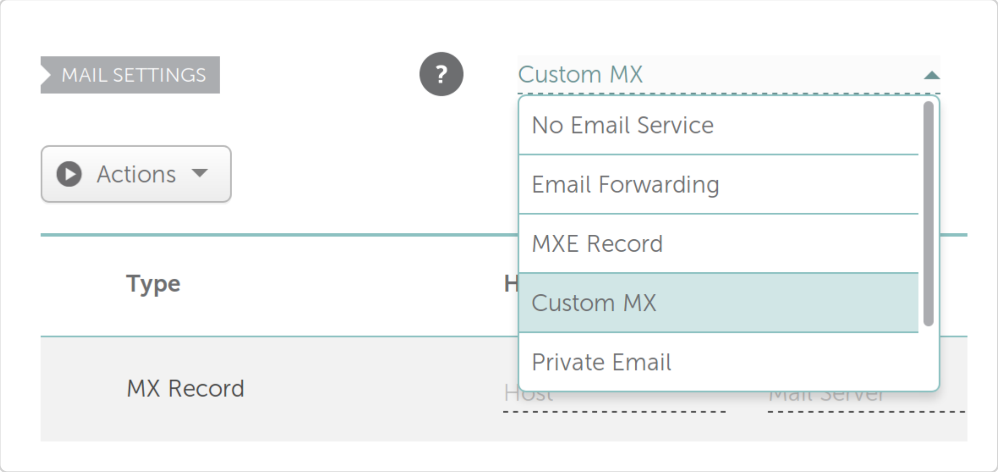
If your domain is registered with Namecheap, you can use Cloudron's Namecheap DNS backend to manage the DNS.

To get started:

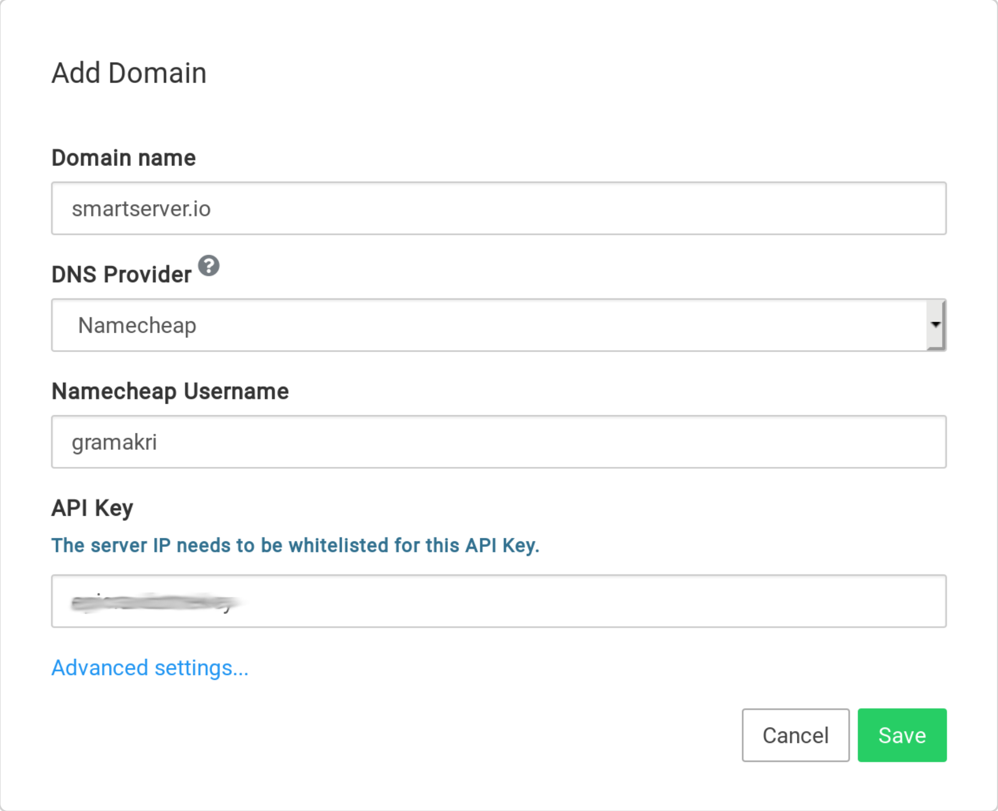
- Enable API access for the Namecheap account (this is disabled by default). To do so, create an API key (Profile -> Tools) and whitelist your Cloudron's IP in the Namecheap dashboard.

> ⚠ **Access error**
>
> It takes a few minutes for the IP whitelisting to take effect. If adding the domain in Cloudron gives an access error, wait a bit.

- If you intend to enable Cloudron Email for this domain, select `Custom MX` in the `MAIL SETTINGS` and set the MX record to be `my.example.com` (i.e the primary domain). Once set, Cloudron will keep this record updated accordingly.



- In the Cloudron dashboard, choose Namecheap from the drop down.



Add Domain

Domain name

smartserver.io

DNS Provider ?

Namecheap

Namecheap Username

gramakri

API Key

The server IP needs to be whitelisted for this API Key.

[redacted]

Advanced settings...

Netcup DNS

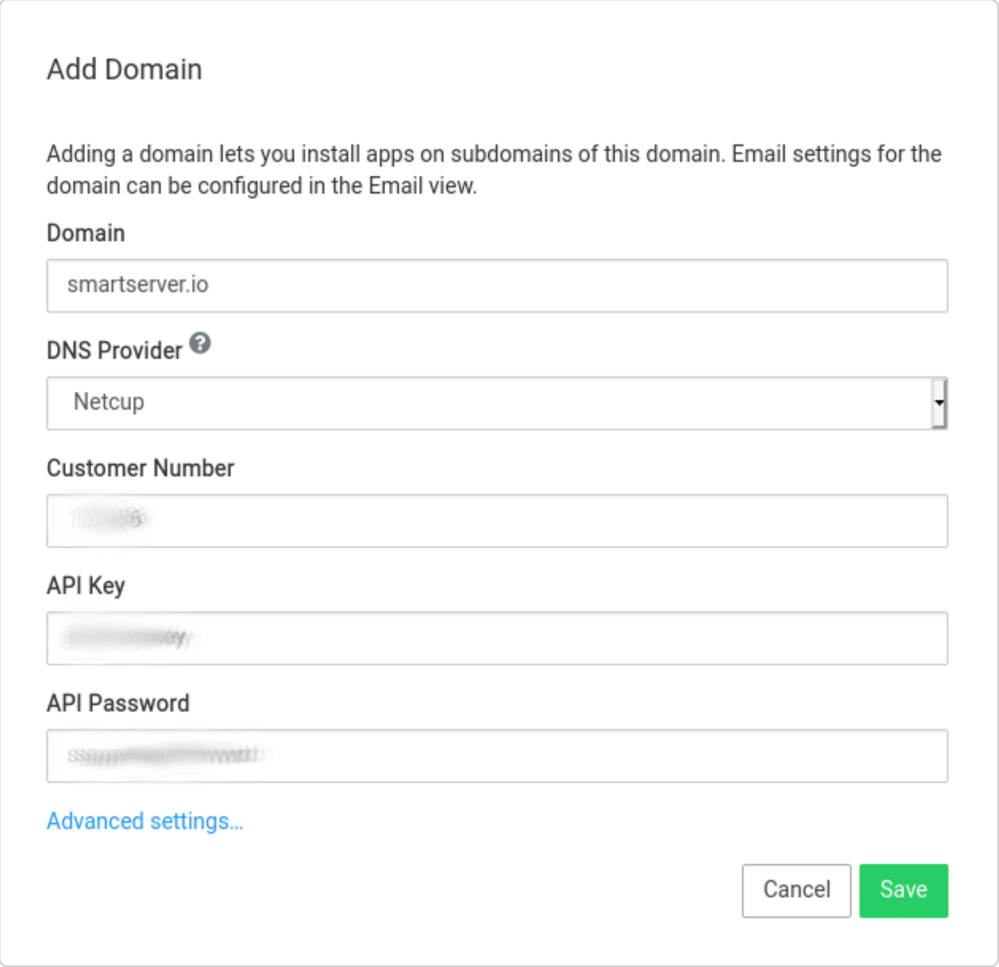
If your domain is registered with Netcup, you can use Cloudron's Netcup DNS backend to manage the DNS.

DNS updates are slow

On Netcup DNS updates take longer than on other providers. This means app installation or location may take a few minutes.

To get started:

- Create an API Key and API Password for the Netcup account. To do so go to the Customer Control Panel and agree to the ToS. Then create both an API Key and the API Password.
- In the Cloudron dashboard, choose Netcup from the drop down and enter your customer number, seen in the top of the customer control panel as well as the API key and password.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: smartserver.io

DNS Provider: Netcup

Customer Number: [redacted]

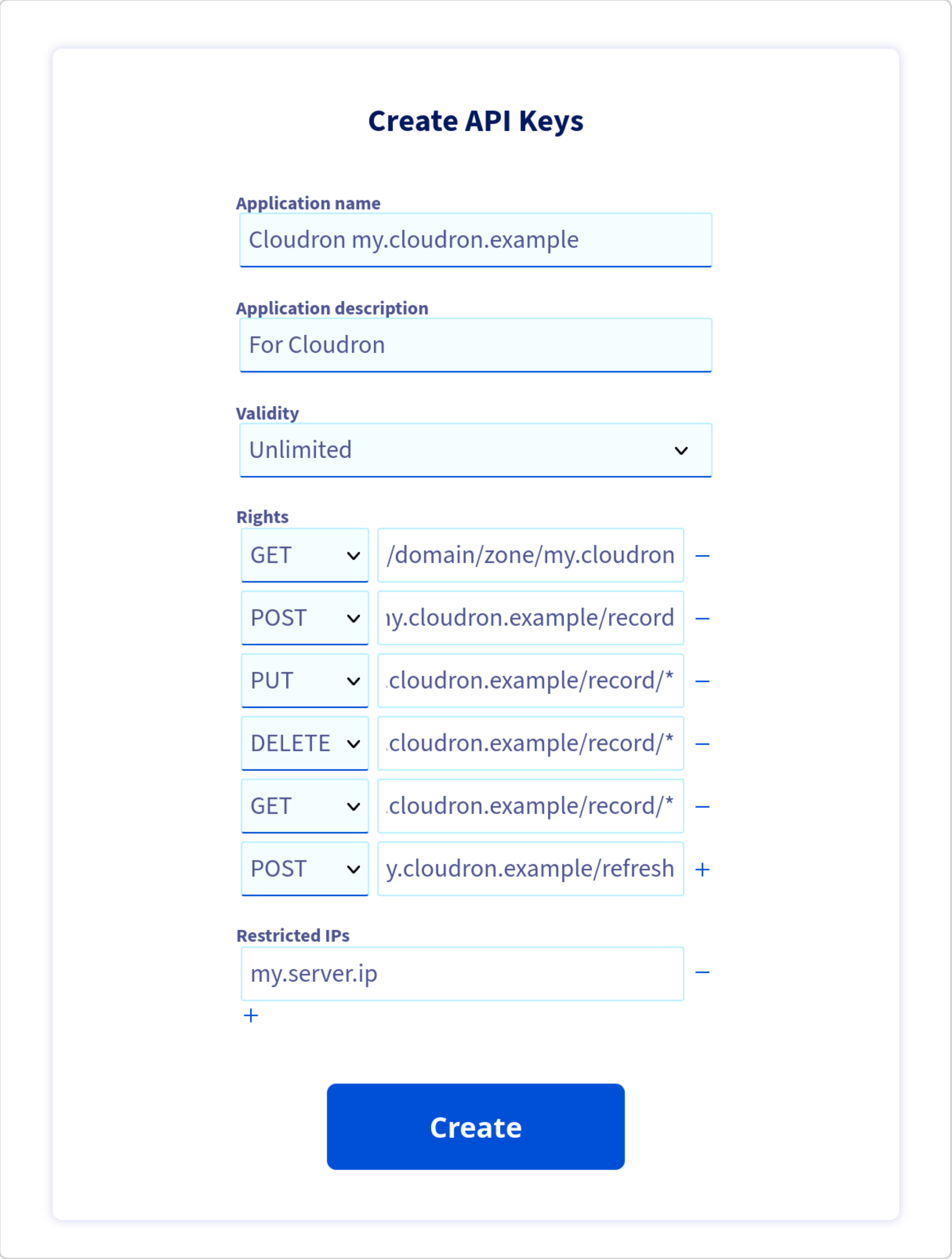
API Key: [redacted]

API Password: [redacted]

OVH DNS

To get started:

- Create an Application key, Application secret and Consumer Key by following the docs at OVHCloud Help.
- API Keys can be created with a minimal scope:



Create API Keys

Application name
Cloudron my.cloudron.example

Application description
For Cloudron

Validity
Unlimited

Rights

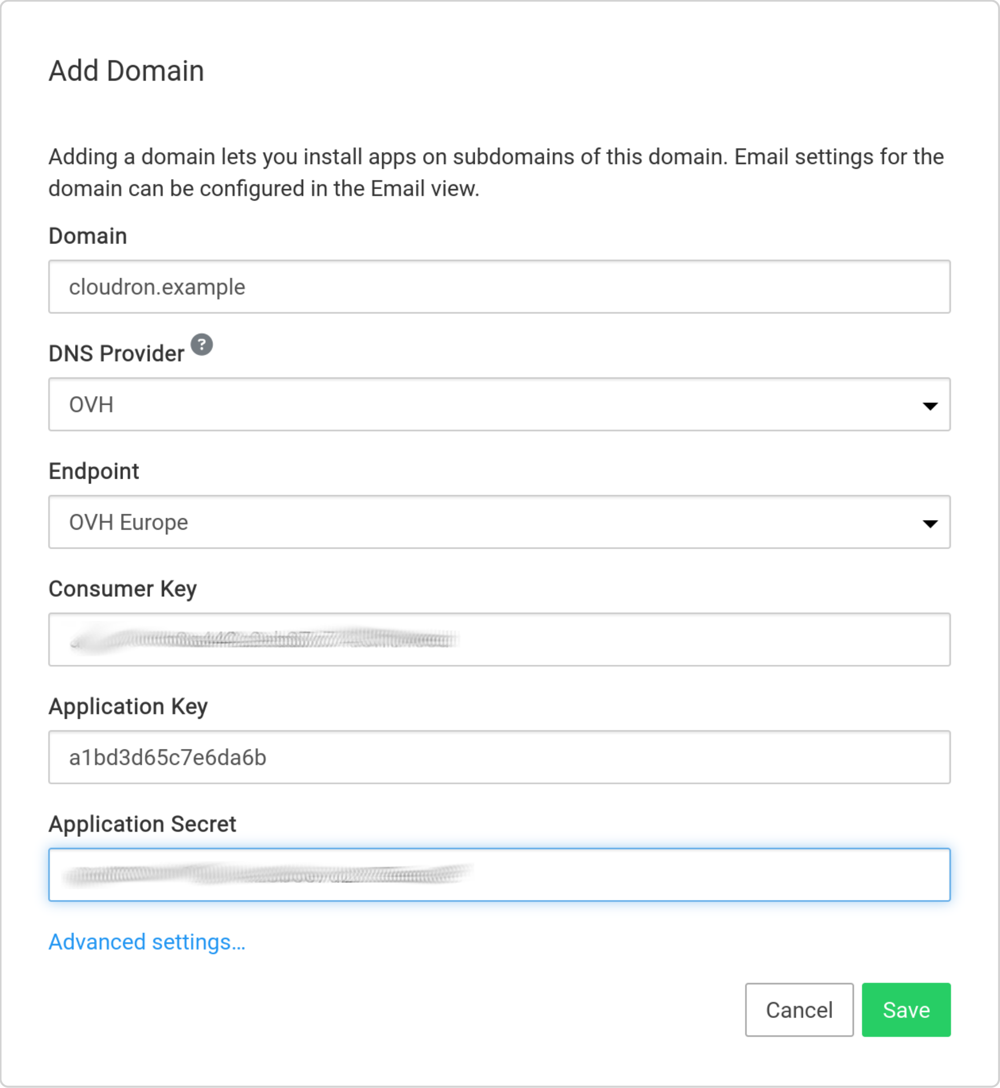
GET	/domain/zone/my.cloudron	–
POST	ıy.cloudron.example/record	–
PUT	.cloudron.example/record/*	–
DELETE	.cloudron.example/record/*	–
GET	.cloudron.example/record/*	–
POST	y.cloudron.example/refresh	+

Restricted IPs
my.server.ip –
+

```
GET /domain/zone/{zone Name}/record
POST /domain/zone/{zone Name}/record
PUT /domain/zone/{zone Name}/record/*
DELETE /domain/zone/{zone Name}/record/*
```

```
GET /domain/zone/{zone Name}/record/*
POST /domain/zone/{zone Name}/refresh
```

- In the Cloudron dashboard, choose OVH from the drop down and enter the Application key, Application secret & Consumer Key.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron.example

DNS Provider: OVH

Endpoint: OVH Europe

Consumer Key: [redacted]

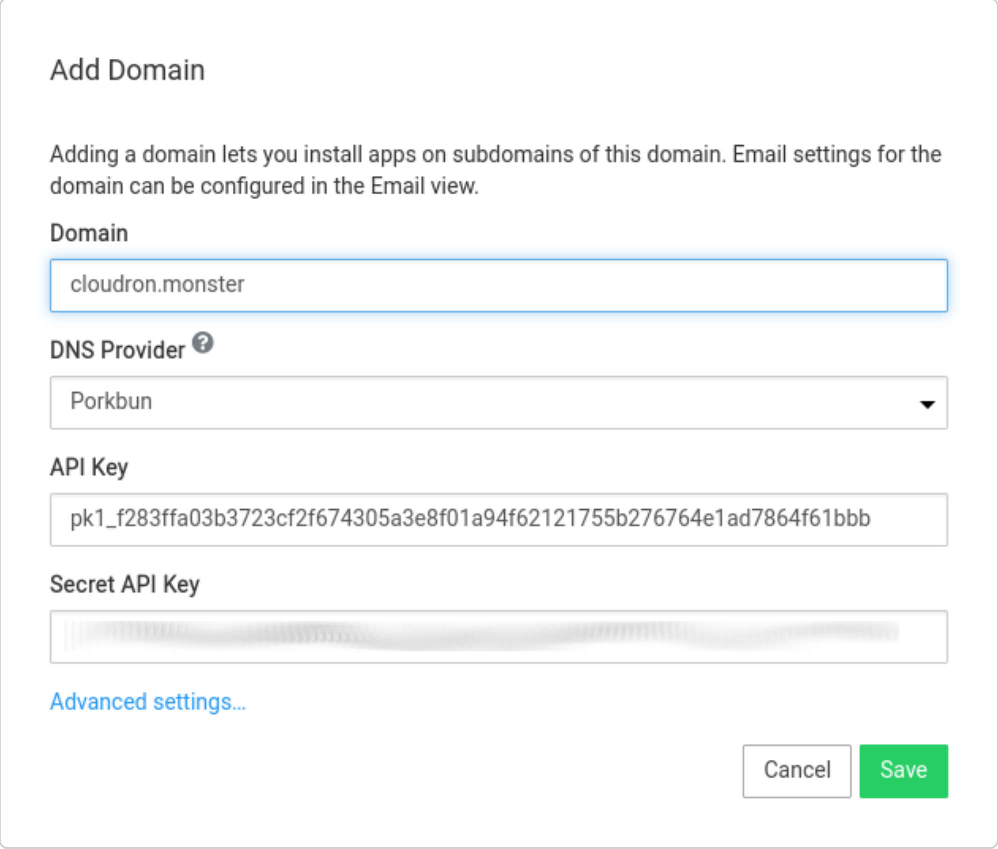
Application Key: a1bd3d65c7e6da6b

Application Secret: [redacted]

Porkbun DNS

To get started:

- Create an API Key and API Secret in the Porkbun dashboard.
- Ensure `API ACCESS` is enabled for the domain. Note that API access to a domain is disabled by default, so this is a necessary step.
- In the Cloudron dashboard, choose Porkbun from the drop down and enter the API key and API secret.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain: cloudron.monster

DNS Provider: Porkbun

API Key: pk1_f283ffa03b3723cf2f674305a3e8f01a94f62121755b276764e1ad7864f61bbb

Secret API Key: [redacted]

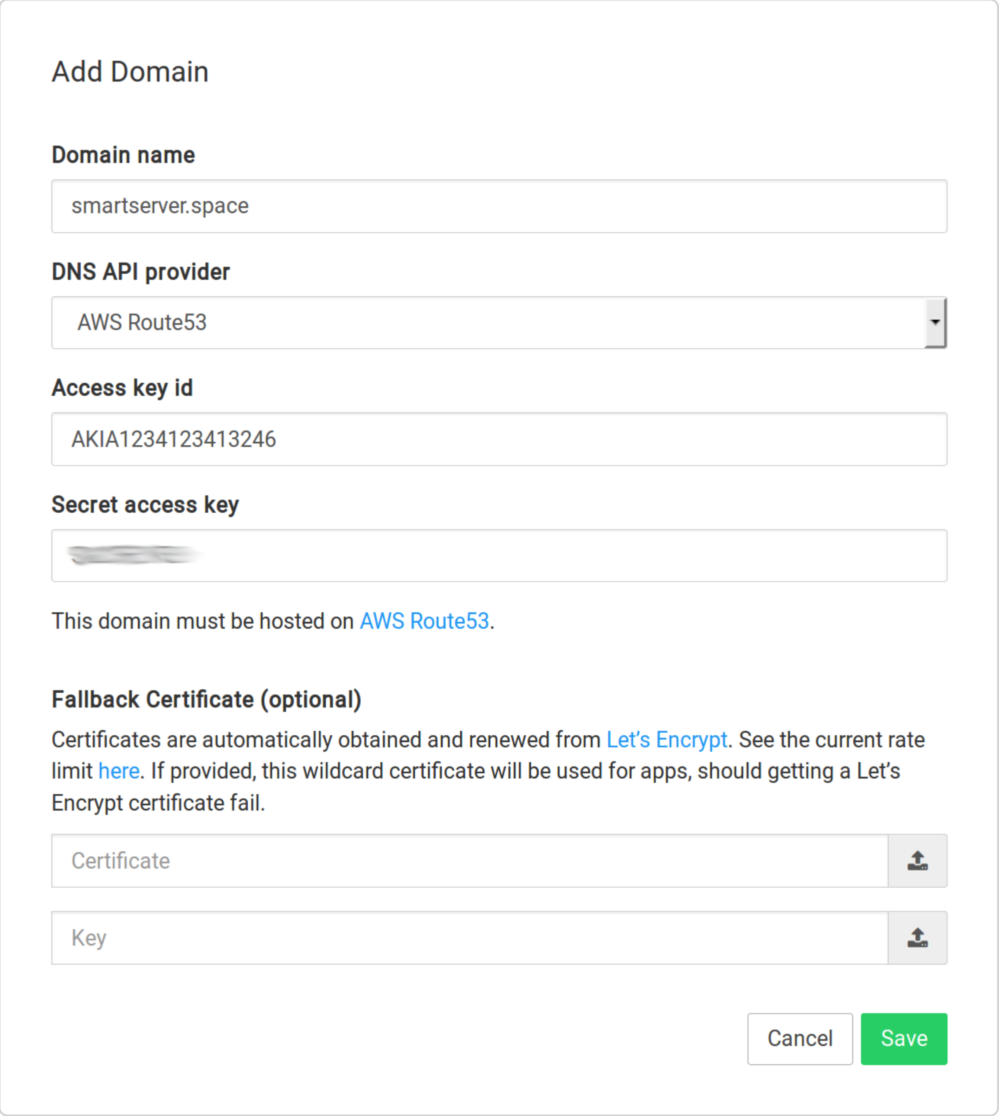
Route53 DNS

To get started:

- Ensure the domain is hosted using AWS Route53. If not, you can follow this guide to migrate an existing domain to use the service.
- AWS has two forms of security credentials - root and IAM. When using root credentials on AWS, follow the instructions here to create access keys. When using IAM, follow the instructions here to create a user and use the following policy to give the user access to the domain. The `<hosted zone id>` below must be replaced with the zone's id which is available from the Route53 console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "route53:*",
            "Resource": [
                "arn:aws:route53:::hostedzone/<hosted zone id>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53:ListHostedZones",
                "route53:listHostedZonesByName",
                "route53:GetHostedZone",
                "route53:GetChange",
                "route53:ChangeResourceRecordSets"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

- In the Cloudron dashboard, choose AWS Route53 from the drop down.



Add Domain

Domain name: smartserver.space

DNS API provider: AWS Route53

Access key id: AKIA1234123413246

Secret access key: [redacted]

This domain must be hosted on AWS Route53.

Fallback Certificate (optional)

Certificates are automatically obtained and renewed from Let's Encrypt. See the current rate limit here. If provided, this wildcard certificate will be used for apps, should getting a Let's Encrypt certificate fail.

Certificate

Key

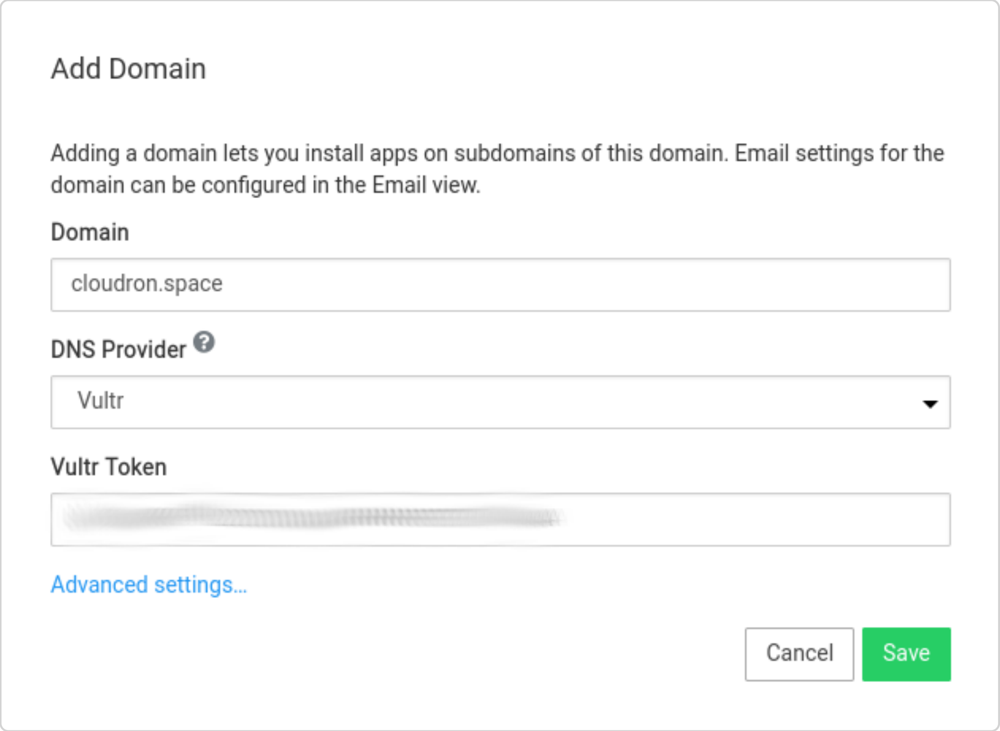
⚠ **Not available in AMI**

This feature is disabled in AWS Marketplace AMI. AWS Marketplace Policy disallows AMIs from requesting IAM credentials from users to access Route53 hosted domains. Please use the Wildcard or Manual provider instead.

Vultr DNS

To get started:

- Ensure the domain is hosted using Vultr DNS. If not, you can follow this guide to migrate an existing domain to use the service.
- Create a Vultr API token. Add the server's IP in the Access Control section.
- In the Cloudron dashboard, choose Vultr from the dropdown.



Add Domain

Adding a domain lets you install apps on subdomains of this domain. Email settings for the domain can be configured in the Email view.

Domain

cloudron.space

DNS Provider

Vultr

Vultr Token

[redacted]

Wildcard DNS

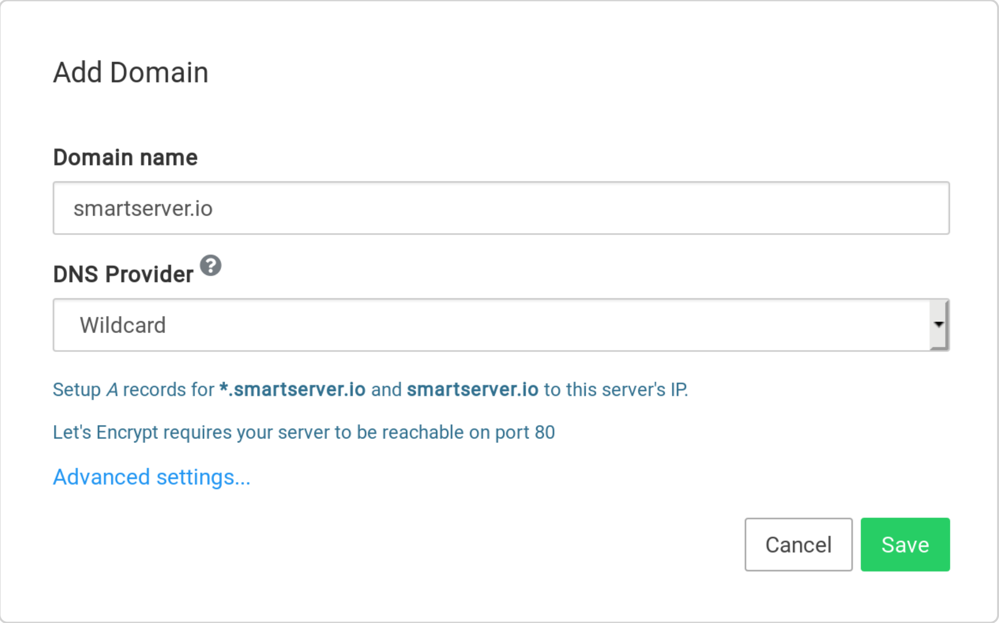
If your domain is not hosted on any of the DNS providers supported by Cloudron, you can use the Wildcard DNS backend.

To get started:

- Add a DNS A record with name `*.example.com` to point to your server's IP.

> **Wildcard entry has lower precedence**
>
> In DNS, a wildcard entry has lower precedence to subdomains that are explicitly defined. This means that if you already have a `blog.example.com` pointing to a different IP address, then it will be unaffected by the addition of this wildcard entry.

- (Optional) Add a DNS A record with name `example.com` to point to your server's IP. This is required only if you intend to host an app on the naked/bare domain (`example.com`) on the Cloudron.
- In the Cloudron dashboard, choose Wildcard from the dropdown.



Add Domain

Domain name: smartserver.io

DNS Provider: Wildcard

Setup *A* records for ***.smartserver.io** and **smartserver.io** to this server's IP.

Let's Encrypt requires your server to be reachable on port 80

- For sending email, Cloudron requires DKIM and SPF records to be setup as well. These records will be displayed in the UI after installation and have to be setup manually.

> **Let's Encrypt integration**
>
> Cloudron will use Let's Encrypt HTTP validation to procure certificates for apps. For this reason, you must open port 80 of your server when using the Wildcard provider.

Manual DNS

If your domain is not hosted on any of the DNS providers supported by Cloudron, and you cannot use the Wildcard DNS provider, then you can use the Manual DNS provider.

With the manual DNS provider, you have to setup DNS records prior to installing Cloudron and also prior to installing each app. App installation will not succeed until DNS records are setup correctly.

If you are attempting to finish Cloudron setup:

- Set the `my` subdomain to the server's public IP
- Choose Manual from the DNS provider drop down
- For sending email, Cloudron requires DKIM and SPF records to be setup as well. These records will be displayed in the UI after installation and have to be setup manually.
- Remember to setup A records for subdomains to the server's public IP and then install apps.

Let's Encrypt integration

Cloudron will use Let's Encrypt HTTP validation to procure certificates for apps. For this reason, you must open port 80 of your server when using the Wildcard provider.

No-op DNS

The No-op DNS backend disables Cloudron's DNS functionality and is intended to be used for testing and development.

When using other DNS backends, Cloudron will setup the DNS automatically and also check if the DNS changes have propagated. This prevents the user from hitting name resolution (NXDOMAIN) errors. When using the No-op backend, the setup and checks are disabled and you are on your own to ensure that names are getting resolved correctly.

4.7.3 Zone Name

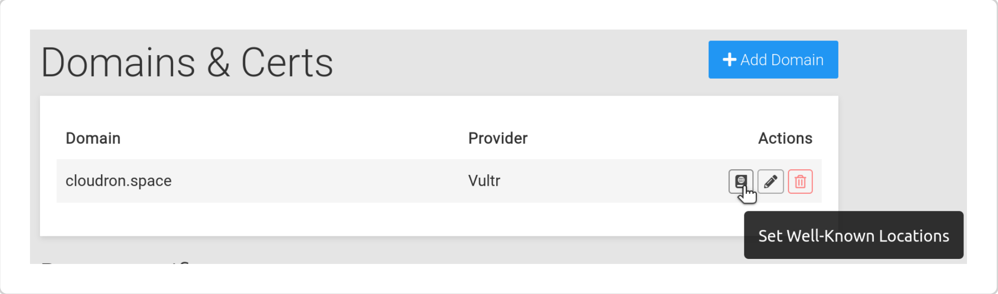
The DNS Zone Name is the domain name that is being managed by the DNS provider. By default, this value is the top level domain like `example.com`. If the domain and subdomain are managed by different DNS providers, then provide the zone name here.

For example, `example.com` might have been purchased at GoDaddy. You can delegate a subdomain like `internal.example.com` to DigitalOcean by adding it in DigitalOcean and setting the NS records of `internal.example.com` to DigitalOcean DNS. In such a situation, if you use `cloudron.internal.example.com` as the primary domain for Cloudron, the the zone name must be set to `internal.example.com`.

4.7.4 Well Known Locations

A well-known URI is a Uniform Resource Identifier for a URL path prefixes that start with `/.well-known/`. They are implemented in webservers so that requests to the servers for well-known services or information are available at URLs consistent well-known locations across servers. See RFC 8615 for more information.

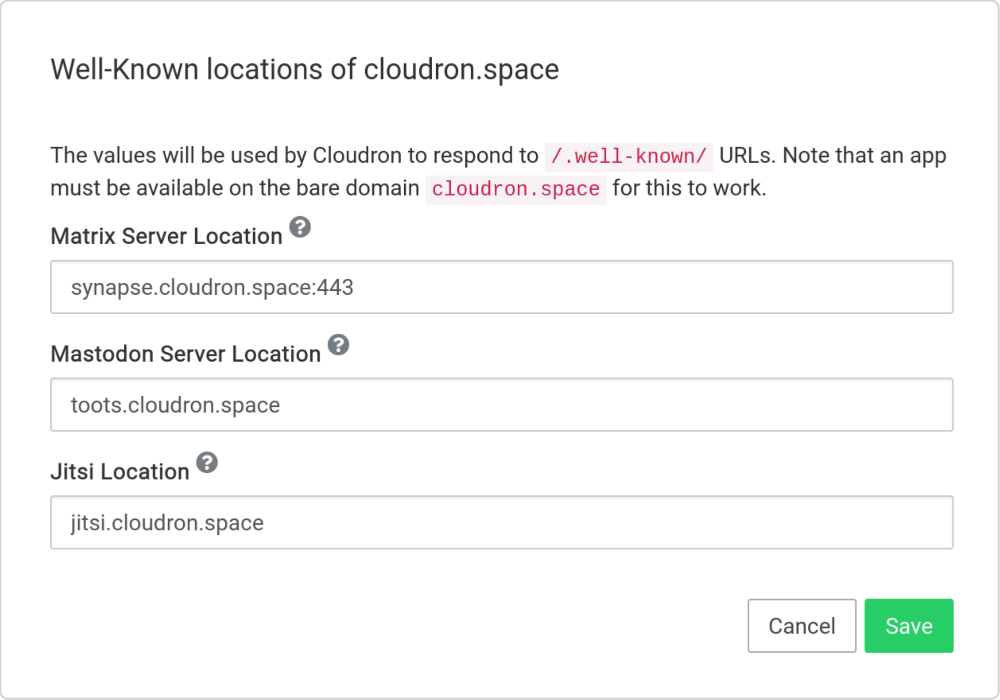
You can edit well known locations in the `Domains` view:



Domains & Certs

Domain	Provider	Actions
cloudron.space	Vultr	

Clicking the button will open up a dialog where you can fill up well known locations:



Well-Known locations of cloudron.space

The values will be used by Cloudron to respond to `/.well-known/` URLs. Note that an app must be available on the bare domain `cloudron.space` for this to work.

Matrix Server Location: synapse.cloudron.space:443

Mastodon Server Location: toots.cloudron.space

Jitsi Location: jitsi.cloudron.space

Matrix server location

The matrix hostname is the domain name and port on which the Matrix server is running. When set, Cloudron will respond to two well-known URIs:

- The `https://{domain}/.well-known/matrix/server` end point. This is required for federation to work. Note that Cloudron can respond to this end point only when an app is installed on the bare domain.
- The `https://{domain}/.well-known/matrix/client` end point. This is required by clients to discover the matrix servers.

> **Specify port 443 explicitly**
>
> The default matrix server port is 8448. However, the Synapse app on Cloudron uses port 443. For this reason, you must specify the port explicity, like `matrix.domain.com:443`.

> **Requires app on bare domain**
>
> In the above example, an app must be installed on the bare domain `https://cloudron.club` for Cloudron to be able to respond to well known queries.

Mastodon server location

The mastodon hostname is the domain name on which the mastodon server is running. When set, Cloudron will respond to requests to the `https://{domain}/.well-known/host-meta` end point. This is required for federation to work. Cloudron can respond to this end point only when an app is installed on the bare domain.

> **Requires app on bare domain**
>
> In the above example, an app must be installed on the bare domain `https://cloudron.club` for Cloudron to be able to respond to well known queries.

Jitsi location

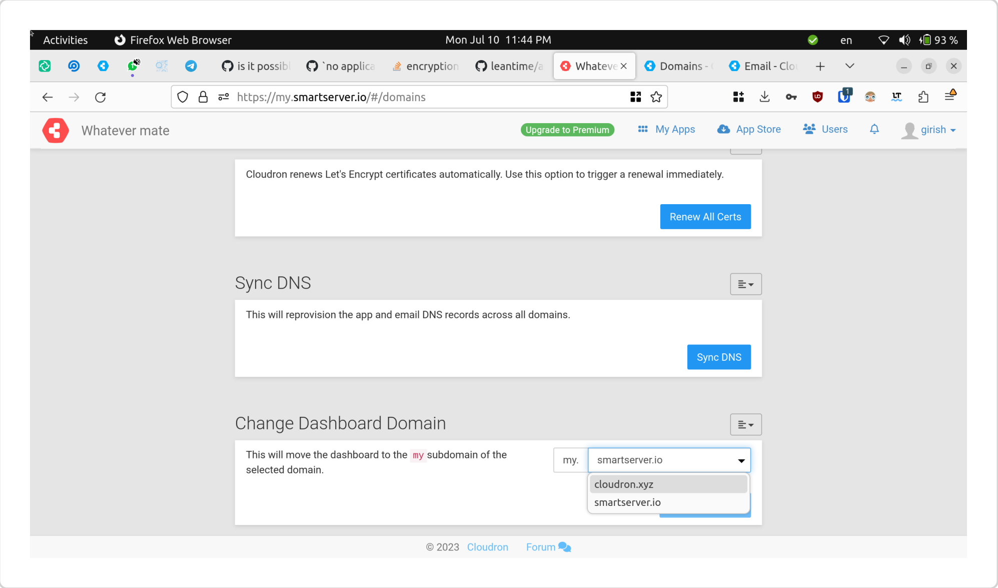
The Jitsi hostname is the domain name on which the jitsi server is running. When set, Cloudron will respond to requests to the `https://{domain}/.well-known/matrix/client` end point as suggested here.

> **Requires app on bare domain**
>
> In the above example, an app must be installed on the bare domain `https://cloudron.space` for Cloudron to be able to respond to well known queries.

4.7.5 Dashboard domain

The Cloudron dashboard/admin UI is located at the `my` subdomain of the Cloudron's primary domain. The `my` subdomain is hardcoded and cannot be changed. However, the primary domain can be changed from the `Domains` view.

Select the domain to move the dashboard into and click `Change Domain`. In a few minutes, you should be redirected to the new location.



Cloudron renews Let's Encrypt certificates automatically. Use this option to trigger a renewal immediately.

Sync DNS

This will reprovision the app and email DNS records across all domains.

Change Dashboard Domain

This will move the dashboard to the `my` subdomain of the selected domain.

my. smartserver.io

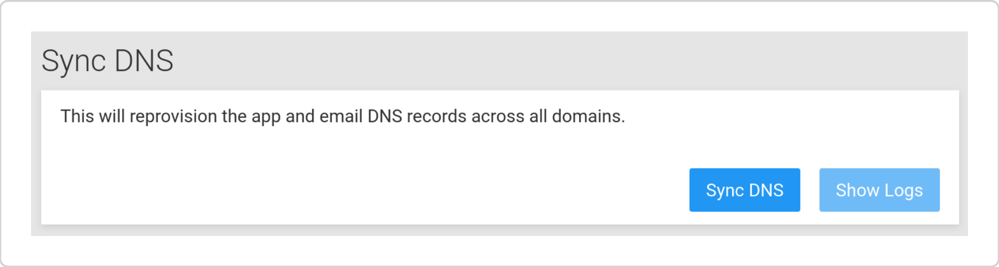
Changing the primary domain has the following implications:

- The dashboard is moved to `my.newprimarydomain.com`. Please update any browser bookmarks accordingly.
- Scripts using the Cloudron API must be updated to use the new dashboard domain.
- Email notifications from the server will now be sent as `no-reply@newprimarydomain.com`.
- The mail server location is unchanged when you change the dashboard domain.
- The old domain's `my` DNS record is not removed by Cloudron. This is because the mail server location defaults to the dashboard domain at installation time. If you end up changing the mail server location as well, the `my` subdomain of the old domain can be safely removed from the DNS.

> **Unreachable dashboard**
>
> If your dashboard is unreachable, see the troubleshooting section

4.7.6 Sync DNS

If you accidentally deleted DNS records or are switching DNS providers, you can restore them using the `Sync DNS` button in the `Domains` view. Note that Cloudron can only restore DNS records that are in use by Cloudron and not records that were externally created.



Sync DNS

This will reprovision the app and email DNS records across all domains.

4.7.7 Autodiscover

DAV

RFC 6764 provides a way for locating services like CalDAV and CardDAV of a domain. For the moment, you can current add DNS entries manually to make these services discoverable.

See this tutorial for the required entries.

Email

See Email autodiscovery for more information.

4.8 Email

4.8.1 Overview

Cloudron has a built-in mail server that can send and receive email on behalf of users and applications. By default, most of it's functionality is disabled and it only sends out mails on behalf of apps (for example, password reset and notification emails).

When `Cloudron Email` is enabled, it becomes a full-fledged mail server solution. You can create mailboxes and assign a mailbox to one or more users or groups. Users can login using Webmail/IMAP/POP3 to send and receive mail.

Features of this mail solution include:

- Multi-domain support
- Enable mailboxes for users and groups on a domain level
- Per-user and group mail aliases
- Mailbox sharing amongst users
- Group email addresses that forward email to it's members
- Email account sub-addressing by adding `+` tag qualifier
- Setup mail filters and vacation email using ManageSieve
- Catch all mailbox to receive mail sent to a non-existent mailbox
- Relay all outbound mails via SendGrid, Postmark, Mailgun, AWS SES or a Smart host
- Anti-spam. Users can train the spam filter by marking mails as spam. Built-in rDNS and zen spamhaus lookup. Admins can add custom spam rules for the entire server.
- Webmail. The SnappyMail and Roundcube apps are already pre-configured to use Cloudron Email
- Completely automated DNS setup. MX, SPF, DKIM, DMARC are setup automatically
- Let's Encrypt integration for mail endpoints
- Domains and IP addresses blacklisting
- Server-side mail signatures (can be set per domain)
- REST API to add users and groups
- Secure out of the box
- Full text search (Email Body and Email Attachments)
- Event Log
- Mail Queue management
- "All Mails" folder

Email settings are located under the `Email` menu item.

4.8.2 Setup

Incoming Email

By default, Cloudron's mail server only sends email on behalf of apps. To enable users to **receive** email, turn on the incoming email option under `Email` -> Select domain -> `Incoming Email`.

Clicking the `Enable` button will give you an option to automatically setup DNS records.

When the `Setup Mail DNS records now` option is checked, Cloudron will automatically update the `MX`, `SPF`, `DKIM`, `DMARC` DNS records of the domain. See the DNS section for more information.

⚠ **Cloudflare hosted domains**

Cloudflare does not proxy email (only http traffic). For this reason, please verify that Cloudflare proxying is disabled for the mail server location.

Server Location

By default, the location of the email (IMAP & SMTP) server defaults to the Cloudron dashboard location i.e `my.domain.com`. This can be changed in the `Mail` settings view.

Cloudron will automatically setup the required DNS records for all the domains when you change the mail server location. Any installed webmail clients will be automatically re-configured as well. Be sure to adjust the settings of any mobile and desktop mail clients accordingly.

✎ **PTR record**

The PTR record of the server's IP must be updated manually to the server location.

Send Test Email

To send a test email, click the `Send Test Email` button.

A dialog will popup where you can enter the email address to send the test email to:

If you are not receiving emails, check the server status and the event log.

Firewall

Sending and receiving email requires TCP ports to be opened up on the server. Cloudron will automatically manage opening up and closing the ports below in the server's firewall.

If you have a Cloud firewall in front of the server like EC2 security group, DO/Linode/Vultr Cloud Firewall, the ports below have to opened up in them.

INBOUND PORTS

The following ports are required for receiving mail.

Port	Notes
25 (SMTP/TCP)	Required for receiving email. When not using Cloudron Email, this can be blocked.
465 (SMTP/TCP)	Used for submitting email via TLS from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
587 (SMTP/TCP)	Used for submitting email via STARTTLS from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
993 (IMAP/TCP)	Used for accessing email from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
4190 (Sieve/TCP)	Used for accessing email filters from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.

OUTBOUND PORTS

The following ports are required for sending mail.

Port	Notes
25 (SMTP/TCP)	Required for sending out emails. If outbound port 25 is blocked by your server provider, setup an email relay. You can check if outbound port 25 is blocked by sending yourself a test email from the Cloudron.

4.8.3 Mailbox

Add

Mailboxes can be created for Users and Groups on a per-domain level. To do so, simply create them in the `Email` view.

The `Mailbox Owner` dropdown can be used to select an existing user or group. The user can then access their email using the new email and the Cloudron password using SMTP and IMAP.

When a group is selected as the owner, any member of the group can access the mailbox with their password.

Mailboxes have the following naming restrictions:

- Only alphanumerals, dot and '-' are allowed
- Maximum length of 200 characters
- Names ending with `.app` are reserved by the platform for applications
- Names with `+` are not allowed since this conflicts with the Subaddresses and tags feature.

Remove

Use the delete button to delete a mailbox.

After deletion, emails to this mailbox will bounce. If you have a catch-all address set, then emails will get delivered to that mailbox.

Deleting old emails

Deleting the mailbox does not remove old emails. You can remove the emails by removing the directory `/home/yellowtent/boxdata/mail/vmail/<mailbox@domain.com>`.

Disable

A mailbox can be temporarily disabled by unchecking the `Mailbox is active` check box.

Once disabled, emails to this mailbox will bounce. If you have a catch-all address set, then emails will get delivered to that mailbox.

Export Mailboxes

This feature has been removed in favor of using the more powerful REST API

Import Mailboxes

This feature has been removed in favor of using the more powerful REST API

4.8.4 Mail aliases

Aliases are alternate addresses for the mailbox. Emails sent to an alias will be delivered to the mailbox.

Aliases can be on the same domain or another domain. Aliases can also contain the `*` wildcard to match zero or more characters.

To send email with the alias as the "From" address, add them as an identity in your mail client. Use the mailbox credentials for authentication when sending email using an identity.

Authenticating with alias is not supported

It is not possible to login using the alias address.

4.8.5 Mailing List

A Mailing list forwards emails to one or more email addresses. A list can be created in the `Email` view.

Use the 'Restrict posting to members only' option to allow only members to post to the list. When enabled, non-members will get a bounce.

To support forwarding mails to external address, Cloudron implements SRS. SRS translation is performed using the mailing list domain before forwarding the mail.

When delivering mails to a list, the mail server checks the `To` and `Cc` fields of the message and suppresses duplicate mail delivery.

No subscribe/unsubscribe feature

Cloudron does not support creating a mailing list (i.e) a list that allows members to subscribe/unsubscribe.

4.8.6 Catch-all address

A Catch-all or wildcard mailbox is one that will "catch all" of the emails addressed to non-existent addresses. You can forward such emails to one or more user mailboxes in the Email section. Note that if you do not select any mailbox (the default), Cloudron will send a bounce.

4.8.7 Email Client Configuration

IMAP

Use the following settings to receive email via IMAP:

- Server Name - Use the mail server location of your Cloudron (default: `my` subdomain of the primary domain)
- Port - 993
- Connection Security - TLS
- Username/password - Use the email id as the username and the Cloudron account password

Multi-domain setup credentials

Use the email id as the username to access different mailboxes. For example, if email is enabled on two domains `example1.com` and `example2.com`, then use `user@example1.com` to access the `example1.com` mailbox and use `user@example2.com` to access the `example2.com` mailbox. In both cases, use the Cloudron account password.

SMTP

Use the following settings to send email via SMTP:

- Server Name - Use the mail server location of your Cloudron (default: `my` subdomain of the primary domain)
- Port - 587
- Connection Security - STARTTLS
- Username/password - Use the email id as the username and the Cloudron account password

> **Multi-domain setup credentials**
>
> Use the email id as the username to send email. For example, if email is enabled on two domains `example1.com` and `example2.com`, then use `user@example1.com` to send email as `example1.com` and use `user@example2.com` to send email as `example2.com`. In both cases, use the Cloudron account password.

Sieve

Use the following settings to setup email filtering users via ManageSieve.

- Server Name - Use the mail server location of your Cloudron (default: `my` subdomain of the primary domain)
- Port - 4190
- Connection Security - STARTTLS
- Username/password - Use the email id as the username and the Cloudron account password

> **Multi-domain setup credentials**
>
> Use the email id as the username to access different mailboxes. For example, if email is enabled on two domains `example1.com` and `example2.com`, then use `user@example1.com` to access the `example1.com` mailbox and use `user@example2.com` to access the `example2.com` mailbox. In both cases, use the Cloudron account password.

POP3

Use the following settings to receive email via POP3:

- Server Name - Use the mail server location of your Cloudron (default: `my` subdomain of the primary domain)
- Port - 995
- Connection Security - TLS
- Username/password - Use the email id as the username and the Cloudron account password

POP3 access is disabled by default and must be enabled per-mailbox.

4.8.8 Subaddresses

Emails addressed to `<username>+tag@<domain>` i.e mail addresses with a plus symbol in the username will be delivered to the `username` mailbox. You can use this feature to give out emails of the form `username+kayak@<domain>`, `username+aws@<domain>` and so on and have them all delivered to your mailbox.

To send email with the subaddress as the "From" address, add it as an identity in your mail client. Use the mailbox credentials for authentication when sending email using an identity.

4.8.9 Relay outbound mails

By default, Cloudron's built-in mail server sends out email directly to recipients. You can instead configure the Cloudron to hand all outgoing emails to a 'mail relay' or a 'smart host' and have the relay deliver it to recipients. Such a setup is useful when the Cloudron server does not have a good IP reputation for mail delivery or if server service provider does not allow sending email via port 25 (which is the case with Google Cloud and Amazon EC2).

> **Send from any email address on a domain**
>
> Cloudron uses the relay to send all outbound emails for the domain. For this reason, the relay must allow Cloudron to send emails as `<any-from-email>@mydomain.com`. This is usually called "Domain verified" identity. Using a relay that is able to send just a single address ("email address entity") will not work.

Cloudron can be configured to send outbound email via:

- Amazon SES
- Elastic Email
- Google
- Mailgun
- Mailjet
- Postmark
- SendGrid
- Office 365
- Sparkpost
- External SMTP server

To setup a relay, enter the relay credentials in the Email section. Cloudron only supports relaying via the STARTTLS mechanism (usually port 587).

Community Guides:

- SMTP Relay Configuration

Amazon SES

To setup Cloudron to relay via Amazon SES:

- Add Domain Identity - Go to Amazon SES dashboard and add a new Domain Identity. Note that Email address Identity will not work because apps on Cloudron send emails with different email addresses. When verifying the domain, leave the `Provide DKIM authentication token` unchecked.
- To complete domain verification, add the DNS keys (CNAME records) in the domain's DNS.
- Once domain is verified, click on `Account dashboard` in the left pane and scroll down to `Create SMTP credentials`.

Follow through the wizard to create a new IAM user that has the following policy

```
        "Statement": [{  "Effect":"Allow",  "Action":"ses:SendRawEmail",  "Resource":"*"}]
```

- Setup the relay on the Cloudron under the Email section:
- Use the Send Test Email button to verify emails are sent.
- If you do not receive the email, please verify that your AWS SES is not in sandbox mode. In this mode, new AWS accounts are only able to send mails to verified domains or the simulator. You can check this in the `Sending Statistics` page and looking for a note that looks like below:

To remove sandbox, log a request to increase the sending limit to say 500 emails a day. Note that, a custom MAIL FROM domain must be set for the DMARC alignment to succeed.

Google

When using Google to relay mail, if you encounter an error message of the form `Invalid login` or `Please log in via your web browser and then try again`, you must configure your Google account to either use App passwords or enable less secure apps. See Google Support for more information.

Office 365

To setup Office 365 as relay, add a connector under mail flow following the instructions under Option 3. Note that relaying via Office 365 requires port 25 to be open and requires a static IP.

SendGrid

When using SendGrid, authenticate the sender identity using Domain Authentication.

> ⚠ **Single Sender Verification will not work**
>
> Cloudron Email relay feature sends emails using multiple email addresses on the same domain, as is the case when multiple apps are installed and each app uses it's own email address.

SMTP Server

Cloudron can relay via an external SMTP server with or without authentication. Use `External SMTP server` option for relaying via a server with a username/password. For IP based authentication relays, use the `External SMTP sever (No authentication)`.

4.8.10 Forward all emails

To forward some or all emails to an external address, create a Sieve filter. Sieve filters can be created using SnappyMail, Roundcube or any other client that supports Manage Sieve.

To support forwarding mails to external address, Cloudron implements SRS.

4.8.11 Marking Spam

The spam detection agent on the Cloudron requires training to identify spam. To do this, simply move your junk mails to the pre-created folder named `Spam`. Most mail clients have a Junk or Spam button which does this automatically.

If you marked a mail as Spam incorrectly, just move it out to the Inbox and the server will unlearn accordingly.

The mail server is configured to act upon training only after seeing at least 50 spam and 50 ham messages.

4.8.12 DNSBL

The email server looks up the connecting IP address against real-time IP blocklists services known as DNSBL or RBL. By default, Cloudron uses the `zen.spamhaus.org` service from the Spamhaus project.

The DNSBL servers can be configured in the Email view.

Some of the popular DNSBL services are listed below. Please be sure to check up the reliability and trustworthiness of the services below before enabling them. This discussion is worth a read.

- `bl.mailspike.net`
- `noptr.spamrats.com`
- `bl.0spam.org`
- `dnsbl.sorbs.net`
- `black.junkemailfilter.com`
- `all.spamrats.com`
- `zen.spamhaus.org` (default)

4.8.13 Address blocklist

Use the spam filter configuration UI in the `Email` view to blacklist addresses and domains. Matched addresses will end up in the user's Spam folder. `*` and `?` glob patterns are supported. This is a global setting and applies for incoming mail to all domains.

When an email matches the above addresses, it will have `USER_IN_BLACKLIST` field set in the `X-Spam-Status` email header.

Blocking IP Addresses

To block an IP address or an entire network, use the Firewall configuration.

4.8.14 Custom Spam Filtering Rules

Custom spam filter rules can be set in the spam filter configuration UI in the `Email` view. Spam filtering rules are global and applies for incoming mail to all domains.

A simple Spam Assassin configuration rule to mark all emails containing the word 'discount' in the subject looks like this:

```
header SUBJECT_HAS_DISCOUNT  Subject =~ /\bdiscount\b/i
score SUBJECT_HAS_DISCOUNT   100
describe SUBJECT_HAS_DISCOUNT    I hate email discounts
```

See this guide for writing custom rules.

Spam Threshold

Emails above the spam score of 5.0 are considered spam and will have the `X-Spam-Flag` set to `YES` in the email header.

4.8.15 Change FROM address of an app

By default, Cloudron allocates the `location.app@domain` mailbox for each installed app. When an app sends an email, the FROM address is set to `location.app@domain.com`. The mailbox name can be changed in the configure dialog of the app.

4.8.16 Disable FROM address validation

By default, the Cloudron does not allow masquerading - one user cannot send email pretending to be another user. To disable this, enable masquerading in the Email settings.

4.8.17 Max mail size

The maximum size of emails that can be sent can be set using the Maximum Mail Size setting.

4.8.18 Storage quota

Storage limit can be set per mailbox by enabling Storage Quota.

When storage limit is hit, the email sender will get a bounce message indicating that the receiving mail box is full.

An email notification is placed in the mailbox when it nears 80% and 95% of it's storage quota. Entries are created in the event log as well.

4.8.19 Vacation mail

An out of office / vacation mail message can be setup using Sieve filters. When using SnappyMail, a vacation message can be set in `Settings` -> `Filters` -> `Add filter` -> `Vacation message` action.

4.8.20 Signature

A disclaimer, confidentiality information or legalese can be appended to every outbound email via the Email Signature setting. This setting is per-domain.

4.8.21 Event Log

Mail server activity can be monitored using the Eventlog UI.

Clicking the `Logs` button will open up the Eventlog.

Clicking the `Logs` button again in the Eventlog view will open up the raw logs. Raw logs are located in the disk at `/home/yellowtent/platformdata/logs/mail`. Up to 10MB of current logs are retained along side 1 rotated log. Logs older than 14 days are removed.

Only available for superadmin

For security & privacy reasons, email event log is only viewable by superadmins (as they most likely have SSH access anyway).

Type	Notes
Bounce	mail resulted in a bounce, and a bounce notification was sent
Deferred	temporary delivery failure of an outbound mail
Denied	connection was denied by the mail server. for example, the remote IP is in DNSBL or the mailbox is invalid
Queued	mail was queued for inbound or outbound delivery
Quota	mailbox hit storage quota limits
Saved	mail was saved in a mailbox
Sent	mail was transferred to another server
Spam	incoming mail was identified as spam
Spam Training	spam training cron job

4.8.22 Mail Queue

Mail queue be monitored and managed using the Mail Queue UI.

Clicking the `Queue` button will open up the `Mail Queue`.

4.8.23 Full Text Search

By default, every text search involves a sequential search through the body of all emails. With a small number of emails (< 5GB), the performance of search is usually acceptable. If there are a large number of emails or if you want to search attachments as well, the emails can be indexed (using Solr and Tika).

To enable the search index, enable Full Text Search for the mail body and mail attachments from the Email settings:

Note that because the indexer consumes a lot of memory, Cloudron might decide to not run it, if the mail server has not been allocated enough memory. The status of the indexer is seen in the Email view:

Email contents and attachments are automatically indexed as they come in and no manual intervention is required. If you enable indexing on a server with existing mails, the first search triggers indexing of that mailbox. If that mailbox has a lot of mails, then the first search can take a long time - maybe even up to 10 mins for 1GB of mail, it all depends on how fast your server is.

Both, Roundcube and SOGo can take advantage of indexed search.

High Resource use

Solr (the indexer) consumes a lot of memory and disk space. To use it, you must allocate at least 3GB for the mail service.

4.8.24 Recursive Search

Recursive Search in IMAP is implemented using `MULTISEARCH` command. However, support for this is non-existent in mail clients.

Roundcube, Thunderbird and SOGo implement recursive search natively.

For other clients like SnappyMail, there is a Virtual `All Mail` folder. The `All Mail` folder contains all the emails across all folders in a single virtual folder. By providing a single folder, SnappyMail can search through all emails easily.

The `All Mail` folder can be enabled in the Email Settings:

See this blog post for more information.

4.8.25 Mailbox Sharing

Mailbox Sharing feature allows users to share their mailboxes with each other using IMAP ACLs.

Mailbox Sharing can be enabled in the `Email` view:

Once enabled, users can share their mailboxes using webmail like SOGo and Roundcube.

For example, in Roundcube, a user can share a folder of their mail account with another user like below:

The other user can see the shared folder in their account:

Manual migration

Mailbox sharing does not work out of the box for Cloudrons that were installed before 6.0. Follow this guide to migrate.

4.8.26 Alternate MX

A failover/alternate/backup MX can be setup for a domain on Cloudron. To set this up, do the following:

- Enable incoming mail for the Cloudron domain
- Cloudron has a anti-spoof feature to ensure that only it can generate emails for the incoming domains. This feature will prevent the external MX from forwarding emails to it. However, Cloudron skips this spoof check for servers listed in the domain's SPF record. So, white list the MX's IP address block in the domain's SPF record. Note that it is necessary to specifically whitelist the server(s). Just a permissive SPF with `~all` and `?all` is not enough.

4.8.27 Archived Emails

When deleteing a mailbox, you can choose whether to delete the content (emails) inside the mailbox. If you choose to keep the emails, then you can find the archived emails at `/home/yellowtent/boxdata/mail/vmail/`.

4.8.28 Autodiscover

Auto discover is a feature where an email client is able to automatically detect the SMTP and IMAP configuration from the email address. Setting up auto discover is very client specific.

Autoconfig.xml

The autoconfig.xml config format is supported by many email clients like Thunderbird, K-9, KMail, Evolution, Kontact & others. Cloudron automatically serves up the autoconfig.xml for all configured mail domains. Note that this feature requires an app to be installed on each of the email bare domains i.e `domain1.com`, `domain2.com`, `domain3.com` and so on. These bare domains can also be just redirects to an existing app. If the bare domain is hosted outside Cloudron, see this forum post for instructions.

> **Requires app on email domain**
>
> Cloudron can serve `autoconfig.xml` only if there is an app installed on the email domain address. For example, if the email address `girish@cloudron.example`, you must have an app (or redirect/alias) at `cloudron.example` and `https://cloudron.example` should work.

To test if the file is being served up, run the following command and validate the output against the spec:

```
curl https://cloudron.example/.well-known/autoconfig/mail/config-v1.1.xml
```

Autodiscover.xml

The autodiscover.xml config format is a process by which a Microsoft Exchange Server client can determine the URL of a particular Microsoft Exchange ActiveSync endpoint. Cloudron supports ActiveSync via SOGo. Note that only Outlook is known to implement this protocol.

If your email address is `girish@cloudron.example`, you must configure the app hosted at `cloudron.example`, to serve up a file named `autodiscover/autodiscover.xml` i.e `https://cloudron.example/autodiscover/autodiscover.xml` must have the file below:

```
<Autodiscover>
  <Response>
    <Culture>en:us</Culture>
    <Action>
      <Settings>
        <Server>
          <Type>MobileSync</Type>
          <Url>https://sogo.cloudron.example</Url>
          <Name>https://sogo.cloudron.example</Name>
        </Server>
      </Settings>
    </Action>
  </Response>
</Autodiscover>
```

Outlook

> ⚠ **Doesn't work**
>
> This method does not seem to work with Microsoft Outlook 2016, Microsoft Outlook 2019 or Microsoft Outlook for Office 365. Your mileage may vary.

Outlook performs an HTTP POST request to the following in order:

- `https://example.com/autodiscover/autodiscover.xml`
- `https://autodiscover.example.com/autodiscover/autodiscover.xml`
- DNS SRV lookup of `_autodiscover._tcp.example.com` and use the response to look up `https://srvresponse/autodiscover/autodiscover.xml.` ` The SRV record is of the format` 0 0 443 ssl.mailprovider.com`.

Because, it is a POST request, you cannot use some static hosting setup. For example, you can use a PHP file like below (there are also more sophisticated attempts):

```
<?php
$raw = file_get_contents('php://input');
$matches = array();
preg_match('/<EMailAddress>(.*)<\/EMailAddress>/', $raw, $matches);
header('Content-Type: application/xml');
?>
<Autodiscover xmlns="http://schemas.microsoft.com/exchange/autodiscover/responseschema/2006">
  <Response xmlns="http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/2006a">
    <User>
      <DisplayName>Cloudron</DisplayName>
      <EMailAddress><?php echo $matches[1]; ?></EMailAddress>
    </User>
    <Account>
      <AccountType>email</AccountType>
      <Action>settings</Action>
      <Protocol>
        <Type>IMAP</Type>
        <Server>my.example.com</Server>
        <Port>993</Port>
        <DomainRequired>off</DomainRequired>
        <SPA>off</SPA>
        <SSL>on</SSL>
        <AuthRequired>on</AuthRequired>
        <LoginName><?php echo $matches[1]; ?></LoginName>
      </Protocol>
      <Protocol>
        <Type>SMTP</Type>
        <Server>my.example.com</Server>
        <Port>587</Port>
        <DomainRequired>off</DomainRequired>
        <SPA>off</SPA>
        <Encryption>TLS</Encryption>
        <AuthRequired>on</AuthRequired>
        <LoginName><?php echo $matches[1]; ?></LoginName>
      </Protocol>
    </Account>
  </Response>
</Autodiscover>
```

4.8.29 SRS

Sender Rewriting Scheme is a scheme for rewriting the envelope sender address of an email message. This method was devised to forward email without breaking SPF.

Cloudron mail server implements SRS for forwarded emails. Cloudron's SRS implementation is in-line with email providers like Office 365 and namecheap.

4.8.30 TLS version

There exist email servers in the wild that use old and obsolete TLS protocols like SSLv3, TLSv1 and TLSv1.1. By default, all these protocols are disabled on the mail server since they are insecure. You can double check if the mail server is using these old protocols using `nmap --script ssl-enum-ciphers -p 25 <mailserverip>`.

If you want to enable support for these insecure protocols, you can do the following:

- `docker exec -ti mail /bin/bash`
- Edit `/run/haraka/config/tls.ini` and the line `minVersion=TLSv1`. Add this line to the beginning of the file.
- `supervisorctl restart haraka`

Note that the setting is not persistent across mail container and server restarts. So, you have to add this line by hand, if those events happen.

4.8.31 Server status

The status of the DNS and Email delivery is available in the `Status` tab of the mail domain (`Email` -> `Select Domain` -> `Status`).

It's a good idea to check your spam score periodically at mail-tester.com. This gives a view of how rest of the world views your email server.

DNS Records

SPF

SPF records specify which servers are allowed to send emails on behalf of a domain name. When you setup Cloudron, it will create a SPF record and set itself as the sender. If a SPF record already exists, it will add itself to the existing record.

If you use a relay, you must modify the SPF record to allow the relay to send emails on behalf of the domain. Please refer to your relay provider's documentation on how to setup SPF.

The SPF project has detailed information on the syntax of this record.

> **SPF Record Type**
>
> Some DNS providers list a DNS record type of `SPF`. This DNS record is obsolete. Use a TXT record instead.

DKIM

DKIM records specify a public key that can be used to authenticate mails from a domain. The rough idea is to generate a public/private key and use the private key to sign all outbound mails. The public key is listed in the DNS and can be used to verify the email.

Cloudron automatically generates a DKIM key pair for each domain and sets up the DNS with the selector `cloudron-<uniqueid>`. The unique id suffix is necessary for the domain to be used across multiple Cloudrons.

DMARC

DMARC is a protocol that uses SPF and DKIM to detect email spoofing. For DMARC validation to succeed, along with SPF or DKIM check, DMARC alignment needs to succeed as well.

By default, Cloudron sets up DMARC records to reject all mails that fail SPF/DKIM validation. This way the Cloudron administrator can feel fairly safe that nobody else is sending mails with their domain.

```
$ host -t TXT _dmarc.girish.in
_dmarc.girish.in descriptive text "v=DMARC1; p=reject; pct=100"
```

When using a relay, please check your service provider documentation to see how to make sure DMARC alignment can succeed.

MX RECORD

The MX record is required to receive mail. If you unable to receive mail despite this being set, check if the outbound port 25 is open.

DANE

DANE provides a mechanism to authenticate TLS certificates without a certificate authority (CA).

DNSSEC is a prerequisite for DANE.

DANE record has to be set up manually. Use this site or the `tlsa` tool to generate records. On Cloudron, the private key for a domain does not change. For this reason, it is safe to choose `SPKI: Use subject public key` as the selector and `DANE-EE: Domain Issued Certificate` for usage.

```
$ tlsa --usage 3 --selector 1 --mtype 1 --port 25 my.cloudron.example
Got a certificate for 185.232.70.47 with Subject: /CN=*.cloudron.example
_25._tcp.my.cloudron.example. IN TLSA 3 1 1 c93c8d09af08595d90c2b59319277eb12eaeb2f7f0fada65a5f23a5b18746110
Got a certificate for 2a03:4000:4e:9d:b4a5:5aff:fec0:3347 with Subject: /CN=*.cloudron.example
_25._tcp.my.cloudron.example. IN TLSA 3 1 1 c93c8d09af08595d90c2b59319277eb12eaeb2f7f0fada65a5f23a5b18746110
```

PTR RECORD

PTR records or rDNS or reverse DNS are DNS entries that can be used to resolve an IP address to a fully-qualified domain name. For example, the PTR record of the IP 1.2.3.4 can be looked up as `host -t PTR 4.3.2.1.in-addr.arpa`.

In the context of email, the PTR record must match the server location. A PTR record is required only when sending email directly from the server. If you are using a mail relay, you do not need to set the PTR record.

The PTR record is set by your VPS provider and not by your DNS provider.. For example, if your server was created in Digital Ocean, you must go to Digital Ocean to set the PTR record.

We have collected a few links to help you set the PTR record for different VPS:

- **AWS EC2 & Lightsail** - Fill the PTR request form.
- **Digital Ocean** - Digital Ocean sets up a PTR record based on the droplet's name. So, simply rename your droplet to `my.<domain>`.
- **Hetzner** - Follow this guide.
- **Linode** - Follow this guide.
- **Netcup** - You can enter a reverse lookup in the customer area CCP for your vServer - wiki doc
- **Scaleway** - You can also set a PTR record on the interface in their control panel.
- **Time4VPS** - Follow this guide.
- **UpCloud** - The PTR can be set in the Network section of the VPS instance configuration.
- **Vultr** - The PTR record can be set for each public IP in the Vultr server settings and is called `Reverse DNS`. It takes about 24h to propagate, so make sure to do this well in advance of enabling email.
- **Home servers** - The PTR record has to be set by your ISP. Customers on Business plans (like Comcast Business) can send a support request to their support to have the PTR set.

Once setup, you can verify the PTR record here.

SMTP Status

OUTBOUND SMTP

To send email, outbound port 25 needs to be unblocked. Most VPS providers block outbound port 25 as a spam control measure.

For some providers like Digital Ocean, Vultr and Linode, you can contact their support to get port 25 unblocked. Providers like EC2 and Lightsail allow oubound port but with a rate limit. For other providers like Google Cloud and home servers, you are left with no choice but to set up a relay.

BLACKLISTS

The server's IP plays a big role in how emails get handled. Cloudron automatically checks the following services to see if your IP is blacklisted. If it is, please follow the links below and contact the services below to get your IP removed (usually this involves filling up some form).

- Abuse.ch
- Barracuda
- Composite Blocking List
- Multi SURBL
- Passive Spam Block List
- Sorbs Aggregate Zone
- Sorbs spam.dnsbl Zone
- SpamCop
- SpamHaus Zen
- The Unsubscribe Blacklist(UBL)
- UCEPROTECT Network

You can also check valli.org for an exhaustive IP blacklist check.

4.8.32 Security

- Cloudron checks against the Zen Spamhaus DNSBL before accepting mail.
- Email can only be accessed with IMAP over TLS (IMAPS).
- Email can only be relayed (including same-domain emails) by authenticated users using SMTP/STARTTLS.
- Cloudron ensures that `MAIL FROM` is the same as the authenticated user. Users cannot spoof each other.
- All outbound mails from Cloudron are `DKIM` signed.
- Cloudron automatically sets up SPF, DMARC policies in the DNS for best email delivery.
- All incoming mail is scanned via `Spamassasin`.

4.8.33 Troubleshooting

Login

If you cannot login, double check that the username is the name of the mailbox i.e mailbox@domain.com . Note that using the cloudron username works in some apps. This is only because some code exists to help migration from legacy setups.

Solr Index Corruption

If search is not working, check if the solr index is corrupt. You can check if this is the case by running `docker exec -ti mail /bin/bash` via SSH and checking the contents of `/run/solr/solr.log`. It might contain a big Java backtrace/exception.

In such cases, it's best to simply recreate the index by running the following commands via SSH:

```
docker stop mail
rm -rf /home/yellowtent/boxdata/mail/solr/dovecot
docker start mail
docker exec -it mail /app/code/dovecot-config/rebuild-index.sh
```

Haraka Queue Corruption

When the Haraka mail queue becomes corrupt, Haraka will keep crashing. To fix this, you can simply empty out the queue.

Run these commands via SSH:

```
docker stop mail
rm /home/yellowtent/boxdata/mail/haraka-queue/*
docker start mail
```

4.9 Networking

4.9.1 IPv4

Using the IPv4 configuration settings, you can configure the IPv4 address, Cloudron uses to configure to the DNS `A` records.

Public IP

When using the `Public IP` provider, Cloudron will automatically detect the server's public IP address by querying this url.

Network Interface

If the server has multiple IP addresses, you can configure the preferred IP address by specifying the network interface. The interfaces can be listed using `ip -f inet -br addr`.

Static IPv4

Use this option to provide a static IPv4 address. This IP address can be public or private. Some use cases for using this provider are:

- Digital Ocean Floating Address
- AWS VPC IP address
- OVH Failover IP

4.9.2 IPv6

Using the IPv6 configuration settings, you can configure the IPv6 address, Cloudron uses to configure to the DNS `AAAA` records.

Public IPv6

When using the `Public IP` provider, Cloudron will automatically detect the server's public IPv6 address by querying this url.

Network Interface

If the server has multiple IPv6 interfaces, you can configure the preferred IPv6 address by specifying the network interface. The interfaces can be listed using `ip -f inet6 -br addr`.

Static IPv6

Use this option to provide a static IPv6 address. It is common for servers to be allocated a `/64` IPv6 block. In such situations, you can use this setting to assign a specific address from that block.

Disabled

To disable IPv6 support, choose `Disabled` in the provider drop down.

> **Existing AAAA records are not removed**
>
> Any existing AAAA records are not automatically removed from the DNS. Please remove them manually.

4.9.3 DNS

All apps and services use the default Ubuntu setup for name resolution. On most VPS providers, this is `systemd-resolved`.

For recursive DNS lookups and DNSBL lookups (Email Server), Cloudron runs the `unbound` DNS resolver.

systemd-resolved

The `systemd-resolved` service runs at 127.0.0.53 . You can use `resolvectl` to check the nameservers being used by `systemd-resolved` and check it's status using `systemctl status systemd-resolvectl`.

Despite the service running, it may be unused due to misconfiguration. `/etc/resolv.conf` must contain `nameserver 127.0.0.53` for the system to use `systemd-resolved`.

If `host www.cloudron.io` resolves, then it's working.

unbound

The `unbound` DNS resolver run internally at `127.0.0.150`. This service does not interfere with DNS resolution of apps and services. Use `systemctl status unbound` to check it's status.

`unbound` may have trouble resolving if your network disallows DNS requests or if a domain has misconfigured DNSSEC. To check if it's working, `host www.cloudron.io 127.0.0.150` should resolve.

If unbound does not resolve, you have to investigate why using `journalctl -u unbound`. You cou can also add a custom config file `/etc/unbound/unbound.conf.d/custom.conf` to disable DNSSEC:

```
# this disables DNSSEC for all domains. alternately, use 'domain-insecure: "example.com"' to disable DNSSEC for specific domain
server:
  val-permissive-mode: yes
```

You can also add another section to forward all queries (change `name` to `example.com.` to restrict to single doamain) to another DNS server, use the configuration below.

```
# forward all queries to the network's internal DNS 10.0.0.2. You can also use 1.1.1.1 (cloudflare) or 8.8.8.8 (google)
forward-zone:
  name: "."
  forward-addr: 10.0.0.2
```

Restart unbound using `sudo systemctl restart unbound` and check it's status using `sudo systemctl status unbound` and check the resolution again using `host www.cloudron.io 127.0.0.150` .

4.9.4 Dynamic DNS

Enable this option to keep all your DNS records in sync with a changing IP address. This is useful when Cloudron runs in a network with a frequently changing public IP address like a home connection.

4.9.5 Internal network

Cloudron runs all apps and services in an internal network (not reachable from outside the server). This network address is hardcoded to `172.18.0.0/16`. Some services like databases have static IPs to aid in connectivity from outside via a SSH tunnel. App addresses are dynamic.

Service	IP
MongoDB	`172.18.30.3`
MySQL	`172.18.30.1`
PostgreSQL	`172.18.30.2`

4.9.6 Firewall

Blocklist

Using the blocklist configuration, one or more IP addresses and/or networks can be blocked from connecting to Cloudron. You can download various country based blocklists from `www.ipdeny.com`: IPv4 and IPv6.

You can also add in comments to the line items as needed, but comments must remain on their own line, something similar to:

```
# spammy IP
111.111.111.111
```

⚠ **Do not lock yourself out**

Be careful about what IP addresses you block. If you lock yourself out, you must get Console access to the server, remove the file `/home/yellowtent/platformdata/firewall/blocklist.txt` and reboot the server.

Whitelist ports

Cloudron does not support installing additional packages or running other services on the server. With that warning out of the way, you can configure the firewall to permit additional (incoming) TCP and UDP ports. For this, edit the the file `/home/yellowtent/platformdata/firewall/ports.json` (create this file if it does not exist and change the owner to the user `yellowtent`).

```
{
    "allowed_tcp_ports": [ 2140, 3540 ],
    "allowed_udp_ports": [ ]
}
```

Restart the firewall to apply the configuration:

```
systemctl restart cloudron-firewall
```

Trusted IPs

When Cloudron is behind a HTTP(S) proxy, you can set the IP address(es) of the proxy as trusted. Doing so will make sure Cloudron trusts the values of various HTTP headers in the request. For example, it can pick up the original client IP address from `X-Forwarded-For` header and use it in logs and email notifications.

✎ **Cloudflare**

When Cloudron is behind Cloudflare, you can use the IP list from here.

4.10 Notifications

4.10.1 Overview

Cloudron will display notifications in the dashboard about various events like:

- If app goes down
- If app run out of memory
- Low disk space
- Updates available
- App updated

The notifications can be read by clicking on the icon in the navigation bar.

4.10.2 Email Notifications

Email notifications can be configured per user in the `Notifications` view.

4.10.3 Server Health Check

For reliability, it is best to configure an external service to monitor the healthcheck of Cloudron itsef. Use `https://my.<domain>/api/v1/cloudron/status` as the health check URL.

4.11 Profile

4.11.1 Account settings

Users can view and edit their personal information in the `Profile view:

Username

Username is used to login to the Dashboard and the apps. The username cannot be changed. To change the username, the admin has to delete the old username and create a new one.

Display name

The Display Name is the first name and last name of the user.

Display Name cannot be changed if the admin has locked user profiles or if the user is from an external directory.

Primary email

The Primary email is the email that is exposed to apps. Apps may send notifications to the user to this email address. This can be set to an email address hosted on Cloudron.

Primary email cannot be changed if the admin has locked user profiles or if the user is from an external directory.

Password recovery email

The Password recovery email is the email to which Cloudron password resets are sent. This should be set to an email address that is not hosted on Cloudron. If not set, it defaults to the Primary email.

Password recovery email cannot be changed if the admin has locked user profiles or if the user is from an external directory.

4.11.2 Icon

The profile icon or gravatar of a user can be changed by clicking on the profile icon.

4.11.3 Background image

A background image for the dashboard can be set using the `Set Background Image` button:

When set, the dashboard will have a background:

4.11.4 Enabling 2FA

2FA can be enabled using the `Enable 2FA button` from the `profile` view in the dashboard.

Clicking on the button will display a QR Code which can be scanned using a TOTP app such as Google Authenticator (Android, iOS), FreeOTP authenticator (Android, iOS)

2FA cannot be enabled if the user is from an external directory that supports 2FA e.g. when authenticating against another Cloudron Directory Server.

4.11.5 Disabling 2FA

Users can disable 2FA by clicking on the `Disable 2FA button`. In the event, the user loses their 2FA device, a Cloudron administrator can reset it.

4.11.6 App Passwords

App passwords can be used as a security measure in desktop, email & mobile clients. For example, if you are trying out a new mobile app from an untrusted vendor, you can generate a temporary password that provides access to a specific app.This way your main password does not get compromised (and thus providing access to other apps as well).

Click the 'New Password' button to create a new app password:

You can delete the password from the password list:

4.11.7 API Tokens

Cloudron API tokens can be created from the Profile view by clicking `New API Token`.

API Tokens are created with a readonly or read write scope:

Tokens can be viewed and revoked from the token listing:

4.11.8 Language

Users can specify the Language setting for the Cloudron dashboard using the language selector:

4.12 Security

4.12.1 Turnkey security

Security is a core feature of Cloudron and we continue to push out updates to tighten the Cloudron firewall's security policy. Our goal is that Cloudron users should be able to rely on Cloudron being secure out of the box without having to do manual configuration.

4.12.2 Privacy & Control

Cloudron is designed to provide complete data ownership and control. Cloudron's design is analogous to how smartphones (like Android, iOS) work. The Cloudron platform is installed on your server similar to how iOS is installed on an iPhone. Apps are installed on the server by contacting the Cloudron App Store (at cloudron.io) similar to how phones install apps by contacting their App Stores. The platform periodically checks for updates by polling cloudron.io (meaning, updates are pulled by your server and not pushed from our servers).

What all this means is:

- Cloudron UG has no mechanism to access your server. There's no hooks for us to administer your server remotely unless you explicitly give us permission to access your server.
- Your server and apps continue running if Cloudron App Store goes down. At worst, you won't be able to install new apps since the Cloudron App Store is down.
- The Cloudron backend code and frontend code are entirely hosted on your server. No CDNs, analytics and other cloud services are used.
- All notifications you get about your apps are sent from your server and not from cloudron.io.
- Cloudron does not collect any user or app information and this is not our business model. As such, there are no incoming requests made to your server from cloudron.io. Cloudron only makes requests to api.cloudron.io for packaging related information and that code can be reviewed here.

4.12.3 HTTPS

- All apps on the Cloudron can only be reached by `https`. `http` automatically redirects to `https`.
- Cloudron admin has a CSP policy that prevents XSS attacks.
- Cloudron set various security related HTTP headers like `X-XSS-Protection`, `X-Download-Options`, `X-Content-Type-Options`, `X-Permitted-Cross-Domain-Policies`, `X-Frame-Options` across all apps.
- Cloudron apps have a default `Referrer-Policy` of `same-origin`.

4.12.4 TURN

Cloudron has a built-in TURN server that is enabled by default with a secret key. This service is carefully configured to not allow relaying to internal services.

4.12.5 SSL/TLS

- Cloudron enforces HTTPS across all apps. HTTP requests are automatically redirected to HTTPS. All certs are EC certs.
- Cloudron supports Let's Encrypt wildcard certificates that help in hiding the subdomain of apps.
- The Cloudron automatically installs and renews certificates for your apps as needed. Should installation of certificate fail for reasons beyond it's control, Cloudron admins will get a notification about it.
- Cloudron sets the `Strict-Transport-Security` header (HSTS) to protect apps against downgrade attacks and cookie hijacking.
- Cloudron has A+ rating for SSL from SSL Labs.

- Let's Encrypt submits all certificates to Certificate Transparency Logs. This is an important consideration when using non-wildcard Let's Encrypt certificates because apps that you install and use are going to be listed in the logs. For example, crt.sh, Google transparency report can display all your subdomains and you can visit those subdomains and guess the app. Please note that this is not a problem when using wildcard certs.

4.12.6 App isolation and sandboxing

- Apps are isolated completely from one another. One app cannot tamper with another apps' database or local files. We achieve this using Linux Containers.
- Apps run with a read-only rootfs preventing attacks where the application code can be tampered with.
- Apps can only connect to addons like databases, LDAP, email relay using authentication.
- Apps are run with an AppArmor profile that disables many system calls and restricts access to `proc` and `sys` filesystems.
- Most apps are run as non-root user. In the future, we intend to implement user namespaces.
- Each app is run in it's own subdomain as opposed to sub-paths. This ensures that XSS vulnerabilities in one app doesn't compromise other apps.
- Process caps like `NET_RAW` are dropped.

4.12.7 TCP Security

- Cloudron blocks all incoming ports except 22 (ssh), 80 (http), 443 (https)
- When email is enabled, Cloudron allows 25 (SMTP), 587 (MSA/STARTTLS), 465 (MSA/TLS), 993 (IMAPS) and 4190 (WebSieve)

4.12.8 Updates

Signed releases

Cloudron platform updates are signed and verified using GPG. Cloudron releases are signed using a GPG key by a Cloudron team member. Cloudron installations download the release packages and verify the signature before applying the updates.

The GPG keys used for signing release packages are maintained offline. This security mechanism means that Cloudron installations will not be compromised even if the Cloudron's cloud infrastructure is compromised.

OS Updates

Ubuntu automatic security updates is enabled. Recent CVEs are listed here.

The system is automatic and does not need any manual intervention. If in doubt, check these:

- `systemctl status unattended-upgrades` provides the status of the upgrade service.
- `sudo apt update && sudo apt list --upgradable | grep security` provides the security updates pending.
- `/var/log/unattended-upgrades/unattended-upgrades.log` has the automatic upgdate logs.
- `sudo unattended-upgrades` to apply any pending security updates at this instant

4.12.9 Rate limits

The goal of rate limits is to prevent password brute force attacks.

- Cloudron password verification routes - 10 requests per second per IP.
- HTTP and HTTPS requests - 5000 requests per second per IP.
- SSH and SFTP access - 5 connections per 10 seconds per IP.
- Email access (Port 25, 465, 587, 993, 4190) - 50 connections per second per IP/App.
- Database addons access - 5000 connections per second per app (addons use 128 byte passwords).
- Email relay access - 500 connections per second per app.
- Email receive access - 50 connections per second per app.
- Auth addon access - 500 connections per second per app.

4.12.10 Password restrictions

- Cloudron requires user passwords to have 1 uppercase, 1 number and 1 symbol.
- User passwords must be at least 8 and at most 256 characters long
- Passwords are individually salted and hashed using PKBDF2 (Section 5.1 of https://www.ietf.org/rfc/rfc2898.txt)

4.12.11 Backups

- Backups are optionally encrypted with AES-256-CBC.

4.12.12 Activity log

The `Activity` view shows the activity on your Cloudron. It includes information about who is using the apps on your Cloudron and also tracks configuration changes.

4.12.13 Cloud Firewall

Cloudron automatically configures the server's OS firewall (iptables) as required. We highly recommend **not** modifying iptables with `iptables`, `ufw` and other tools. Instead, please use the firewall provided by the VPS provider for further hardening. For example, on AWS this would be security group settings, on Digital Ocean the Firewall settings etc.

Inbound ports

Port	Notes
22 or 202 (SSH/ TCP)	Used for SSH access to the server. We recommend disabling password based access and moving this to a different port. See this guide.
222 (SFTP)	Used for SFTP access.
80 (HTTP/TCP)	When using manual and wildcard DNS, Let's Encrypt certificates require this port to be open. This port can be blocked when using programmable DNS.
443 (HTTPS/TCP)	Used for accessing the web page of all apps.
25 (SMTP/TCP)	Used for receiving email. When not using Cloudron Email, this can be blocked.
465 (SMTP/TCP)	Used for submitting email via TLS from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
587 (SMTP/TCP)	Used for submitting email via STARTTLS from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
993 (IMAP/TCP)	Used for accessing email from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
3478 (TCP and UDP)	STUN server
4190 (Sieve/TCP)	Used for accessing email filters from mobile phone or desktop apps. When using only webmail or not using Cloudron Email, this port can be blocked.
5349 (TCP and UDP)	TURN server
50000 - 51000 (UDP)	TURN server communication ports
Other ports	Other ports have to be opened up depending on the apps installed. For example, the git+ssh port has to be opened when using GitLab. Port 53 (UDP) is required if you install an app like AdGuard Home.

Outbound ports

We recommend leaving all outbound ports open. Some providers like AWS EC2, Google Cloud, Digital Ocean forcefully block outbound port 25 for reducing email spam. The only way around this is to either request your server provider to unblock this port or better to setup an Email relay.

4.12.14 Securing SSH access

It is highly recommended to disable password based access to your server since many online attackers brute force passwords. Configuring SSH access to be based on a SSH key secures the server with the equivalent of a 634 length password with random letters and numbers. It is not possible to brute force SSH keys even with modern technology. We recommend using EdDSA instead of RSA keys.

It is a good idea to disable root login via SSH and instead use a sudo user. For this, make sure you have a sudo user first. Usually, there is a user named `ubuntu` that has sudo access.

```
# If you set this to 'no', store the ssh keys in sudo user (e.g /home/ubuntu/.ssh/authorized_keys)
# If you set this to 'prohibit-password', store the ssh keys in /root/.ssh/authorized_keys
PermitRootLogin no             # Can also be 'prohibit-password' to only allow root login with keys.
```

It's best to disable password authentication for all accounts altogether and only use ssh keys. Check for the following line in `/etc/ssh/sshd_config`:

```
PasswordAuthentication no      # Make sure ssh keys are in place for the root or the sudo user!
```

By default, the SSH server runs on port 22. We recommend moving this to port 202 to prevent brute force attacks. Be careful to not lock yourself out when following the instructions below.

To change the SSH port, change the following line in `/etc/ssh/sshd_config`:

```
Port 202   # Do not use any other port. Only this port is not blocked by the Cloudron firewall
```

Port 202

Note that we carefully chose port 202 because this port is specifically unblocked in `iptables` during the Cloudron installation. If you choose some other port, you have to whitelist it. Be careful not to lock yourself out!

The SSH service can be restarted using `systemctl restart ssh`. On Ubuntu 24, you might have to restart the server for the changes to take effect. Use `ssh -p 202 root@ip` to connect to the server.

Fail2Ban

Fail2Ban reads app log files and automatically block IPs. Be aware that Fail2Ban only works partially on Cloudron because most apps do not log failed authenticated attempts in a manner that Fail2Ban can pars. That said, Fail2Ban can be used to block brute force SSH logins by simply installing it:

```
    apt install fail2ban
```

2FA

You can setup Multi-Factor Authentication for SSH by following this guide.

4.12.15 Email

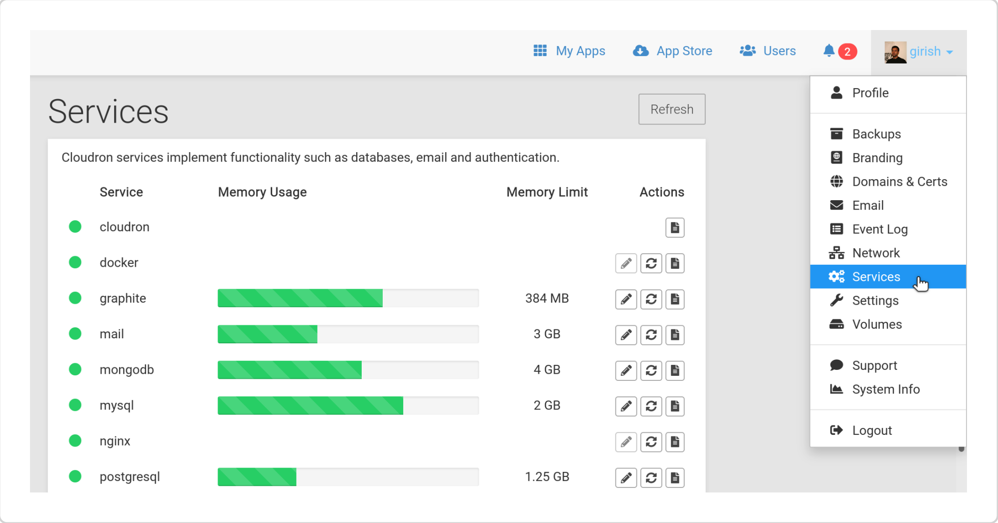
See Email security

4.13 Services

4.13.1 Overview

The `Services` view can be used to monitor the status and manage the configuration of services used by apps. Services are managed entirely by Cloudron. When apps are deployed, Cloudron provisions and configures services internally.

You don't have to worry about keeping the services updated - it's part of the Cloudron platform.



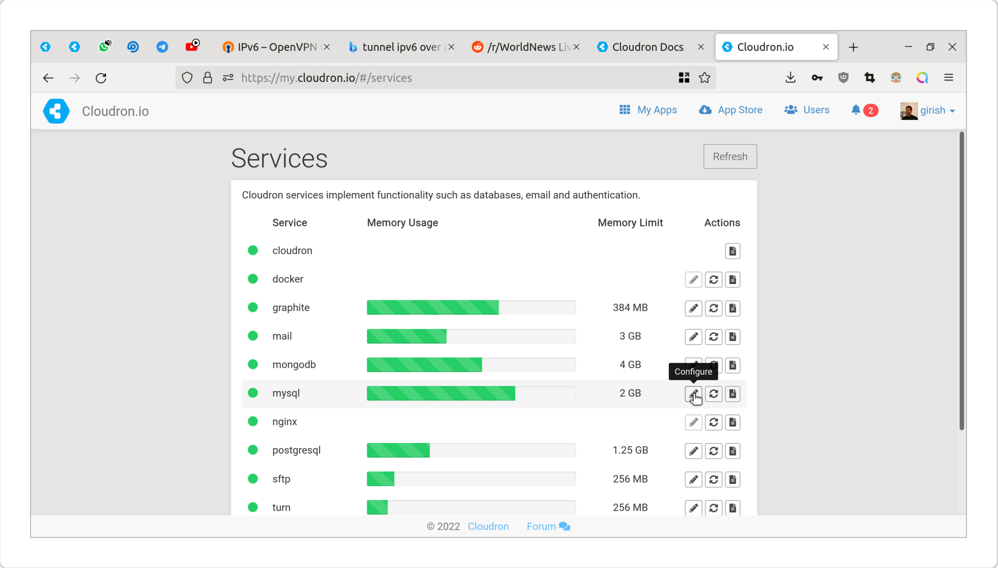
Services

Cloudron services implement functionality such as databases, email and authentication.

Service	Memory Usage	Memory Limit	Actions
cloudron			
docker			
graphite		384 MB	
mail		3 GB	
mongodb		4 GB	
mysql		2 GB	
nginx			
postgresql		1.25 GB	

4.13.2 Configure

To configure a service, use the `Configure` button:

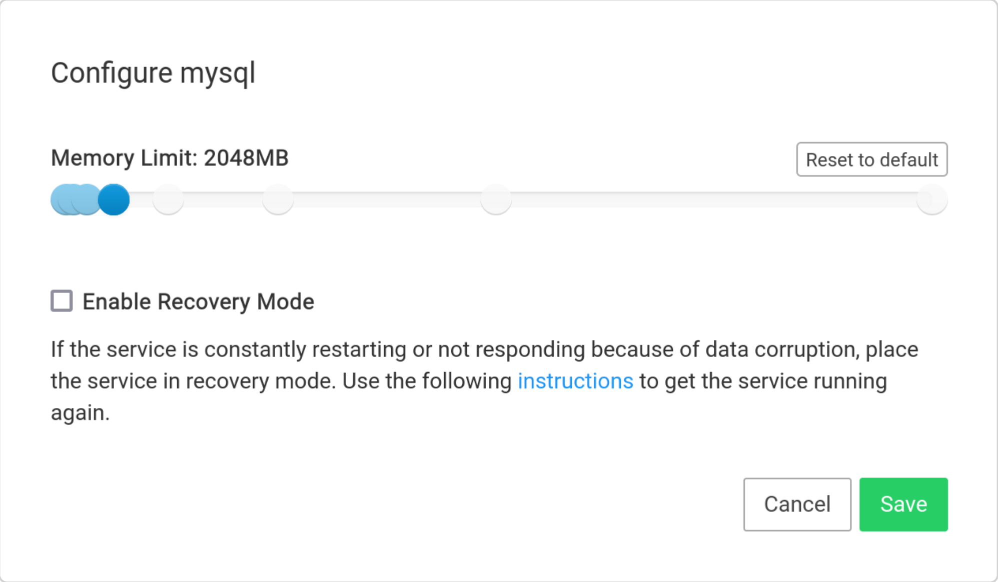


Services

Cloudron services implement functionality such as databases, email and authentication.

Service	Memory Usage	Memory Limit	Actions
cloudron			
docker			
graphite		384 MB	
mail		3 GB	
mongodb		4 GB	
mysql		2 GB	
nginx			
postgresql		1.25 GB	
sftp		256 MB	
turn		256 MB	

This will open up a dialog with various settings depending on the service:



Configure mysql

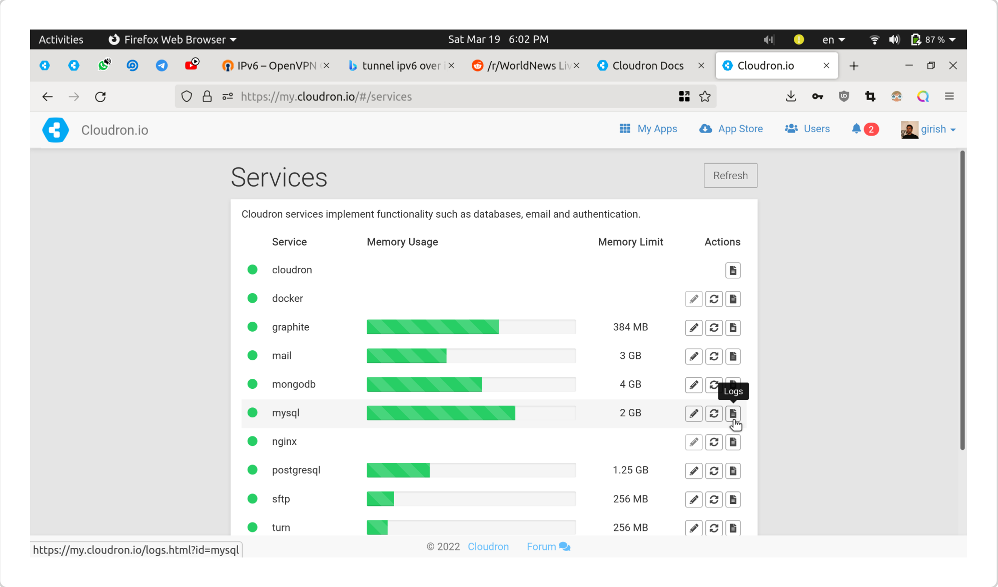
Memory Limit: 2048MB

□ Enable Recovery Mode

If the service is constantly restarting or not responding because of data corruption, place the service in recovery mode. Use the following instructions to get the service running again.

4.13.3 Logs

To view the logs of a services, click the `Logs` button:



Services

Cloudron services implement functionality such as databases, email and authentication.

Service	Memory Usage	Memory Limit	Actions
cloudron			
docker			
graphite		384 MB	
mail		3 GB	
mongodb		4 GB	
mysql		2 GB	
nginx			
postgresql		1.25 GB	
sftp		256 MB	
turn		256 MB	

https://my.cloudron.io/logs.html?id=mysql

© 2022 Cloudron Forum

Up to 10MB of logs is retained per service along side 1 rotated log. Logs older than 14 days are removed. The raw logs are located under `/home/yellowtent/platformdata/logs/<servicename>`.

4.13.4 Troubleshooting

If a service is not running, the first step is to restart it and/or to try increasing the memory limit.

Check the service logs for hints of why it's not starting up.

Corrupt Addon

The instructions here apply to the following database addons:

Database	Container name	Logs	Data Directory
MySQL	mysql	/home/yellowtent/platformdata/logs/mysql/app.log	/home/yellowtent/platformdata/mysql
PostgreSQL	postgresql	/home/yellowtent/platformdata/logs/postgresql/app.log	/home/yellowtent/platformdata/postgresql
MongoDB	mongodb	/home/yellowtent/platformdata/logs/mongodb/app.log	/home/yellowtent/platformdata/mongodb
Redis	redis-	/home/yellowtent/platformdata/logs/redis-/app.log	/home/yellowtent/platformdata/redis/

In case of disk corruption, it's best to just start afresh and restore the data from backups.

To restore a specific database:

```
docker stop <containername>
mv /home/yellowtent/platformdata/<database> /home/yellowtent/platformdata/<database>-copy # stash the database data directory. see table above
mkdir /home/yellowtent/platformdata/<database> # create a fresh database directory
docker restart <containername> # start the addon. this will re-initialize the addon with no data
```

Now, restore each app that uses the addon in the Cloudron dashboard.

4.14 Settings

4.14.1 System Time Zone

The System Time Zone setting controls various cron jobs like backup, updates, date display in emails etc. This time zone can be changed from the settings view.

⚠ **Server is UTC**

The System time zone setting does not change the Server's timezone. The server should always be in UTC. This ensures all Database timestamps and logs are in UTC. This is intentional and should not be changed.

Browser timestamps

For convenience, when viewing logs using the log viewer, timestamps are converted into the browser timezone.

4.14.2 Language

The default language of Cloudron is English. This can be changed using the Language settings. When set, users will be sent out invitation and reset emails using the selected language. Note that users can always set a different language in their profile.

4.14.3 Private Docker Registry

A private docker registry can be setup to pull the docker images of custom apps.

Use docker.io and not docker.com

If you are using the private DockerHub registry, use `docker.io` and not `docker.com`

4.15 Storage

4.15.1 Disk layout

Cloudron data and temporary/runtime data is sandboxed in `/home/yellowtent` . For the curious, `yellowtent` was the code name for Cloudron. The subdirectories are:

- `box` - This contains the Cloudron code. The Cloudron code does not run as root.
- `boxdata` - This contains data that is generated by the Cloudron code including certs. This also contains all user emails in the `mail/vmail` directory in the maildir format.
- `appsdata` - This contains the data generated by each app. Each directory here corresponds to the application id.
- `platformdata` - This contains 'runtime' data of the platform for mysql, postgres, mongodb, nginx.

`boxdata`, `appsdata` and `platformdata` are relocatable.

The other important locations on the server are:

- `/var/lib/docker` - This contains docker images. This is relocatable.
- `/etc/nginx` - These contains the reverse proxy configuration. It is not meant to be edited manually.
- `/apps.swap` - This is a swap file that is dynamically resized by Cloudron. It will be no more than 4GB. You can resize this to any size after the initial setup. If you want to disable the swap altogether, just truncate the file to 0 size.

4.15.2 Docker images

⚠ **Do not use a Cloudron Volume**

Do not use a Cloudron Volume as storage location for Docker images. Set up fstab or systemd mount manually.

Cloudron uses Docker for containerizing applications and docker images tend to consume a lot of space. The docker images are stored by default at `/var/lib/docker`. They can be moved to an external storage as follows:

- Stop docker and the box code:

```
systemctl stop docker box
systemctl mask docker
```

- To change docker image location to say `/mnt/docker` by creating a systemd drop-in `/etc/systemd/system/docker.service.d/custom.conf` (sic) with the following contents:

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// --log-driver=journald --exec-opt native.cgroupdriver=cgroupfs --storage-driver=overlay2 --data-root=/mnt/docker --
experimental --ip6tables
```

- Start docker

```
systemctl daemon-reload
systemctl unmask docker
systemctl start docker
systemctl status docker
docker network create --subnet=172.18.0.0/16 --ip-range=172.18.0.0/20 --gateway 172.18.0.1 cloudron --ipv6 --subnet=fd00:c107:d509::/64
```

- Now edit the file `/home/yellowtent/platformdata/INFRA_VERSION`. Bump the `version` field carefully by only adjusting the last digit from `48.18.0` to `48.18.1`. Do not change the other digits. This is just a way to inform the cloudron code that the infrastructure has changed and it has to re-create containers.
- Reboot the server. This is required for the `docker network` to start working in many VPSes (for unknown reasons).
- On reboot, addons and app images will be recreated preserving existing data. This can take a while. You can use `tail -f /home/yellowtent/platformdata/logs/box.log` to see the progress.
- Finally, after everything is up, you can remove the old docker images

```
rm -rf /var/lib/docker
```

Docker image size

The `du -hcs /var/lib/docker` command will often return incorrect output with the overlay2 driver because of how du works. Use `docker system df` instead to determine the size

Docker storage driver

Cloudron apps are only tested using the overlay2 graph driver. Other storage drivers are not tested. In the past, we have seen some apps to not work correctly with storage drivers like devicemapper.

4.15.3 Default Data Directory

Do not use a Cloudron Volume

Do not use a Cloudron Volume as storage location for the directories below. Set up fstab or systemd mount manually.

Unsupported

While symlinking various directories will work, this is not supported. Instead, move the Data Directory of high storage consuming apps to an external volume.

Please make sure you have a complete backup before following the procedure below.

Apps store their data under `/home/yellowtent/appsdata`. Cloudron itself stores it's data (users, groups, certs, mails etc) under `/home/yellowtent/boxdata`.

If the server is running out of disk space, one or more of these directories can be moved to another ext4 disk/location as follows:

```
    systemctl stop box
    systemctl stop cloudron-syslog
    systemctl stop docker
    systemctl mask docker
    DATA_DIR="/var/data"  # this is the external disk location
    mkdir -p "${DATA_DIR}"

    # (optional) move out apps data to the external disk
    mv /home/yellowtent/appsdata "${DATA_DIR}"
    ln -s "${DATA_DIR}/appsdata" /home/yellowtent/appsdata

    # (optional) move out box data to the external disk
    mv /home/yellowtent/boxdata "${DATA_DIR}"
    ln -s "${DATA_DIR}/boxdata" /home/yellowtent/boxdata

    # (optional) move out app database storage to the external disk
    mv /home/yellowtent/platformdata "${DATA_DIR}"
    ln -s "${DATA_DIR}/platformdata" /home/yellowtent/platformdata
    systemctl unmask docker
    systemctl start docker
    systemctl start cloudron-syslog
    systemctl start box
```

If the disk graph does not display properly, do a `systemctl restart collectd`.

Note: data directory must be an `ext4` filesystem.

4.15.4 Server Resize

For VPS providers that support it, the server (cpu/disk/memory) can be resized and Cloudron will automatically adapt to the available resources after a server restart.

Some providers do not automatically resize the disk after enlargement. You can check if your disk has the new bigger size with the following command:

```
df -h
```

As an example, your disk `/dev/sda` was enlarged and your partition `/dev/sda1` is your data partition, you need to run the following command:

```
resize2fs /dev/sda1
```

Please always check your provider's documentation for more in detail steps.

> **IP changes**
>
> Some VPS providers change the IP address of the virtual machine after a resize. You will have to change the IP address in your DNS, if this is the case.

AWS EBS

On AWS, the partition and the filesystem must be resized after resizing the EBS volume. This can be done by following the AWS Guide.

4.15.5 Swap

To give the server more swap space, run the following commands:

```
dd if=/dev/zero of=/extra.swap bs=1024 count=2097152
chmod 600 /extra.swap
mkswap /extra.swap
swapon /extra.swap
```

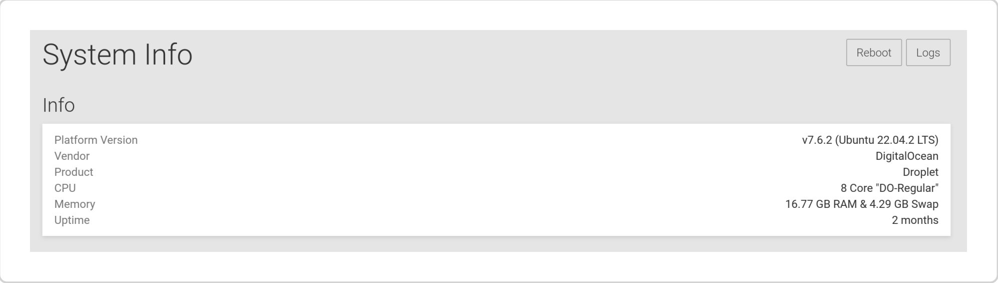
To make sure the swap persists reboots, add a line in `/etc/fstab`:

```
/extra.swap  none  swap  sw  0 0
```

4.16 System

The System view gives an overview of the Server, CPU, Memory & Disk usage.

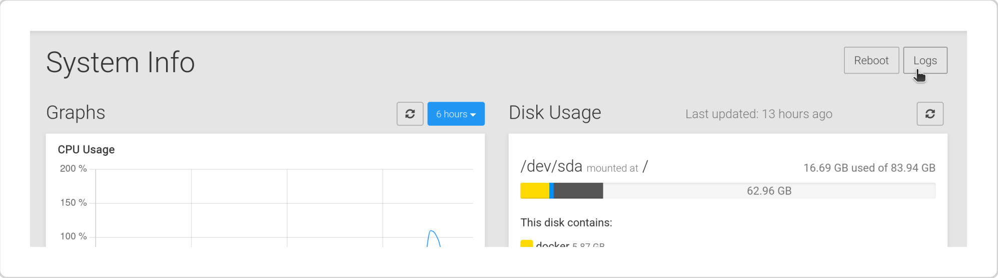
4.16.1 Info



Platform Version	v7.6.2 (Ubuntu 22.04.2 LTS)
Vendor	DigitalOcean
Product	Droplet
CPU	8 Core "DO-Regular"
Memory	16.77 GB RAM & 4.29 GB Swap
Uptime	2 months

4.16.2 Logs

The `Logs` button can be used to view the System logs. In the situation that the dashboard is unreachable/down, the raw logs are located at `/home/yellowtent/platformdata/logs/box.log`. Up to 10MB of active logs is kept along side 5 rotated logs. Logs older than 14 days are removed.



Last updated: 13 hours ago

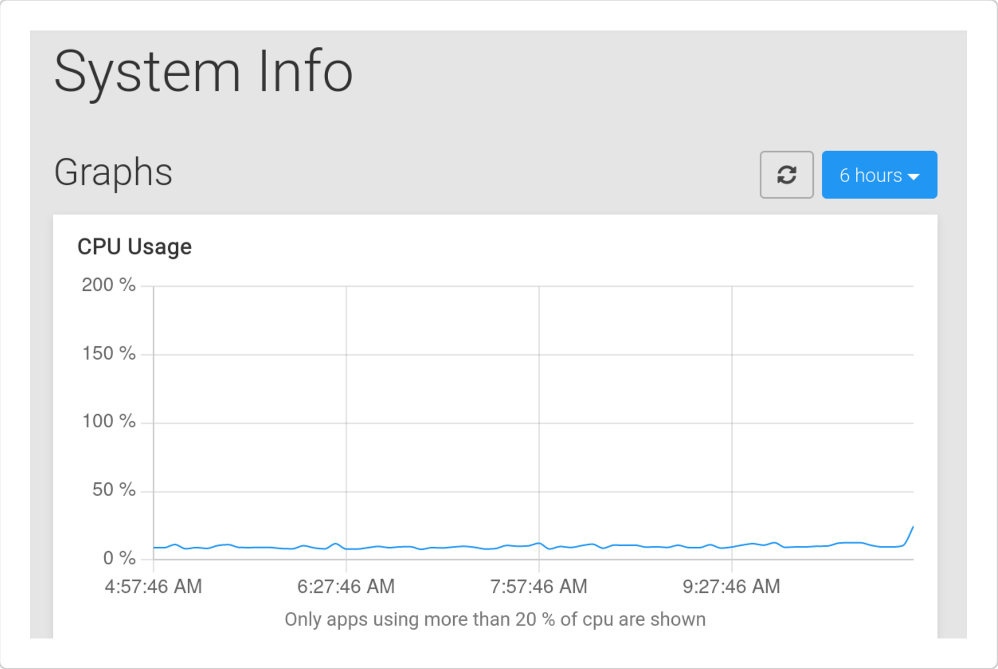
/dev/sda mounted at / 16.69 GB used of 83.94 GB

62.96 GB

This disk contains:

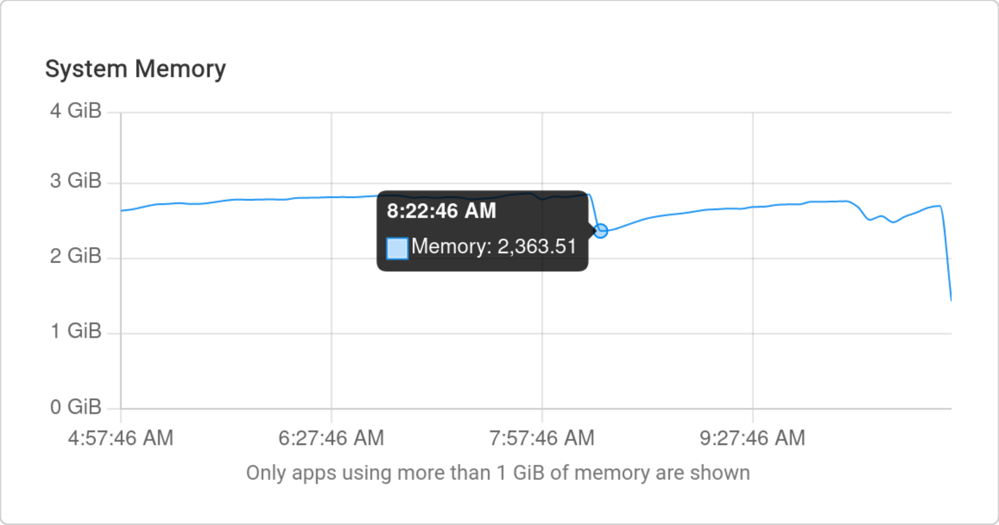
docker 5.87 GB

4.16.3 CPU Usage



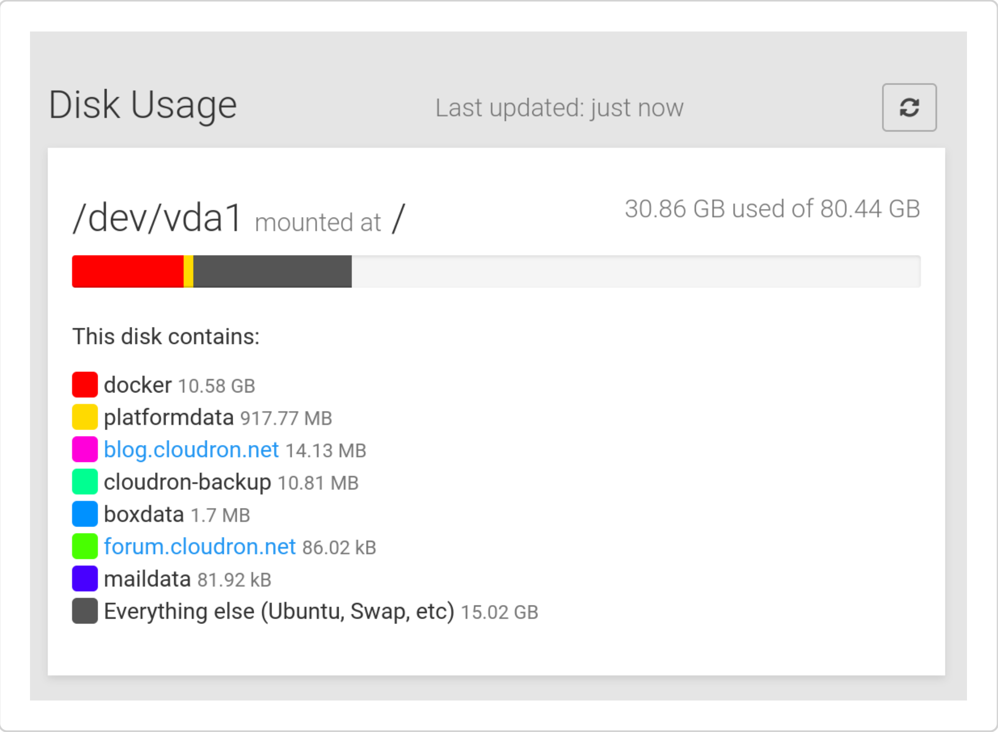
Only apps using more than 20 % of cpu are shown

4.16.4 Memory Usage



4.16.5 Disk Usage

Disk usage is only computed on demand. Clicking on the `Refresh` button in the top right will compute disk usage in that instant.



Disk Usage Last updated: just now

/dev/vda1 mounted at / 30.86 GB used of 80.44 GB

This disk contains:

- 🟥 docker 10.58 GB
- 🟨 platformdata 917.77 MB
- 🟪 blog.cloudron.net 14.13 MB
- 🟩 cloudron-backup 10.81 MB
- 🟦 boxdata 1.7 MB
- 🟩 forum.cloudron.net 86.02 kB
- 🟪 maildata 81.92 kB
- ⬛ Everything else (Ubuntu, Swap, etc) 15.02 GB

Directory	Notes
`docker`	Size of docker images. Cloudron's docker images take around 10GB. In addition, app images vary greatly in size and it's normal for each app to be around 2GB in size.
`docker-volumes`	Temporary data of apps. If this is large, find the offending app with `docker ps -q | xargs -I {} sh -c 'echo "Container: {}"; docker exec {} du -sh /run /tmp'` .
`/apps.swap`	System swap. Usually same amount as system RAM, but limited to 4GB max.
`boxdata`	Box (Cloudron code) data.
`maildata`	Email data.
`Everything else (Ubuntu, etc)`	System packages. Use system tools like `du` to investigate if this is more than say 20GB
`platformdata` - This contains logs, database directories (mysql, postgres, mongo, redis), logs and performance metrics.	
App	Persistent data of app excluding database size

Excluding disks

If your disk is too slow, it can be excluded from periodic disk usage collection. Add the filesystem path (much match `df` output) per line to `/home/yellowtent/platformdata/diskusage/exclude` .

4.17 Updates

4.17.1 Cloudron Updates

Cloudron checks cloudron.io periodically and applies any platform updates automatically.

Cloudron always takes a complete backup of the platform data and all apps before applying an update. Should the backup creation fail, the update will not be performed.

Updates are GPG signed for security.

It is always recommended to review the official Cloudron Changelog for a list of changes made in each version.

4.17.2 App updates

Cloudron Apps are installed from the Cloudron.io App Store. Cloudron checks the App Store for updates periodically and updates them based on the update settings.

Cloudron always takes a backup of an app before performing an update. Should the backup creation fail, the update will not be applied. If an update fails or the updated app misbehaves, the Cloudron administrator can rollback from a backup.

Updates can be disabled on a per app level from the `Updates` section:

When app updates are disabled or an update is pending, an update indicator is shown:

4.17.3 Update schedule

The update schedule can be set in the `Updates` section in the `Settings` menu:

To disable updates entirely, select `Update manually`. When automatic updates are disabled, the Cloudron administrators will get an email notification as and when updates are available. Updates can then be applied by clicking the update button.

4.17.4 Rollback

To rollback an app update, simply restore from a backup.

To rollback a Cloudron update, it can be restored from the backup.

4.17.5 Canceled Subscription

When you cancel the subscription, the Cloudron and installed apps stop receiving updates. The server and the apps will continue to run forever. The subscription may be renewed at any time (we may contact you if we notice that you regularly cancel and renew your subscription).

4.18 Users & Groups

4.18.1 Overview

Cloudron provides a central user directory that apps can use for authentication. This feature allows users to use the same username & password for logging in to apps.

When installing an app, you can choose if the app is to be configured to use the Cloudron user directory. Disabling the integration can be beneficial when the app is meant to be used primarily by external users (for example, a community chat or a public forum).

When Cloudron user directory integration is available for an app, the user management options will look like below:

When `Leave user management to the app` is selected, the app's Cloudron SSO integration will be disabled and all user management has to be carried from within the app. This is useful when the app primarily caters to external users (like say a community chat).

When Cloudron user directory integration is unavailable for an app, the user management options look like below:

4.18.2 Users

New users can be added with their email address from the `Users` menu.

Click on `New User` to add a new user:

They will receive an invite to sign up. Once signed up, they can access the apps they have been given access to.

To remove a user, simply remove them from the list. Note that the removed user cannot access any app anymore.

Valid usernames

The following characters are allowed in usernames:

- Alphanumeric characters
- '.' (dot)
- '-' (hyphen)

Usernames must be chosen with care to accomodate the wide variety of apps that run on Cloudron. For example, very generic words like `error`, `pull`, `404` might be reserved by apps.

4.18.3 Groups

Groups provide a convenient way to group users. You can assign one or more groups to apps to restrict who can access for an app.

You can create a group by using the `Groups` menu item.

Click on `New Group` to add a new group:

To set the access restriction use the app's configure dialog.

Valid group names

The following characters are allowed in group names:

- Alphanumeric characters
- '.' (dot)
- '-' (hyphen)

4.18.4 Roles

Roles provide a way to restrict the permissions of a user. You can assign a role from the `Users` page.

User

A Cloudron user can login to the Cloudron dashboard and use the apps that they have access to. They can edit their profile (name, password, avatar) on the dashboard.

To allow a user to configure and manage specific apps, see the App Operator feature.

User Manager

A User Manager can add, edit and remove users & groups. Newly added users always get the `User` role. User Manager cannot modify the role of an existing user.

Mail Manager

A Mail Manager can add, edit and remove mailboxes and mailing lists, in addition to being able to manage users.

> **No access to mail server logs**
>
> For security reasons, a Mail Manager does not have access to the email server logs.

Administrator

A Cloudron administrator can manage apps and users. An admin can:

- Login to any app even if they have not explicitly granted access to in the `Access Control` section
- Impersonate any user
- Access data of other users via File Manager or Web Terminal
- Configure various aspects of Cloudron like branding, networking, domains, services etc
- Access mail server logs

If you only want to make a user configure and manage specific apps, use the App Operator feature.

Superadmin

A Cloudron superadmin has all the capabilities of administrator. In addition, a superadmin can:

- Manage the Cloudron subscription
- Manage backup storage and policy
- Open support tickets

A good way to think about the superadmin role is a person who is in charge of server administration and billing.

> **Automatic login**
>
> When clicking the `Manage Subscription` button in the `Settings` view, they are automatically logged in to the cloudron.io account.

4.18.5 Impersonate user

A common situation is to be able to pre-setup applications on behalf of a new user. In some cases, the user has to login to an app first before they can be added to a channel/group/team or given specific permissions.

For such situations, the Cloudron admin can use the `Impersonate` button to generate a password that lets a Cloudron login as another user. The password is time limited and can be used to login to the Cloudron dashboard and the apps.

Clicking the button will show a temporary password:

Does not reset existing password

The temporary password does not overwrite the user's existing password.

Impersonate via ghost file

One can create a file named `/home/yellowtent/platformdata/cloudron_ghost.json` which contains an username and a fake password like:

```
{"girish":"secret123"}
```

With such a file in place, you can login to the Webadmin UI using the above credentials (the user has to already exist). This helps you debug and also look into how the UI might look from that user's point of view.

4.18.6 Password reset (link)

Users & Admins can reset their own passwords from the login screen. `https://my.example.com/login.html?passwordReset` is a direct link.

Admins can also email a password reset link for other users by clicking on the `Reset password` button.

This will open up a dialog showing the password reset link. If email delivery is not working for some reason, the link can be sent to the user by some other means.

4.18.7 Password reset (ssh)

A temporary one-time use password for the superadmin account can be generated by SSHing into the server:

```
# sudo cloudron-support --owner-login
Login as superadminname / mW5x5do99TM2 . Remove /home/yellowtent/platformdata/cloudron_ghost.json when done.
```

The above password also bypasses any 2FA and can thus be used in situations where the superadmin has lost the 2FA device.

4.18.8 Disable 2FA

If a user loses their 2FA device, the Cloudron administrator can disable 2FA in the user's edit dialog.

Once disabled, user can login with just their password. After login, they can re-setup 2FA.

If the superadmin of Cloudron has lost their 2FA device, see the password reset section to generate a one-time use password that will bypass 2FA.

4.18.9 Disable user

To disable a user, uncheck the `User is active` option. Doing so, will invalidate all existing Cloudron dashboard session of the user and will log them out. The user may still be able to use any apps that they were logged into prior to the de-activation. To log them out from the apps, you can check if the app provides a way to log them out (support for this depends on the app).

Disabling does not delete user data

Disable a user only blocks the login access for the user. Any data generated by the user inside apps is not deleted.

4.19 User Directory

4.19.1 Overview

Cloudron provides a central user directory that apps can use for authentication. This feature allows users to use the same username & password for logging in to apps.

Cloudron is an OIDC provider as well as a LDAP server. Cloudron App Policy is to use OIDC integration whenever possible since this is more secure and support 2FA.

4.19.2 Lock profile

Admins can disallow users from changing their email and full name by locking user profiles. To lock the profile, simple uncheck the setting in the `User Directory` view.

4.19.3 Mandatory 2FA

Admins can require all users to set up two factor authentication by enabling the Mandatory 2FA setting. To enable, use the setting in the `User Directory` view.

When enabled, all existing users without a 2FA setup are logged out immediately.

When users without 2FA attempt to login, they will be forced to setup 2FA:

When the user clicks `Setup Two-Factor`, they go through the 2fa setup flow:

4.19.4 External Directory Connector

The External Directory connector allows users from an existing LDAP or Active Directory to authenticate with Cloudron.

When enabled, Cloudron will use profile information like Username, Display Name and Email from LDAP.

2FA behavior depends on the provider. When using the Cloudron provider, 2FA of the external directory is used. When using other providers, users can setup 2FA locally.

The user's role and active state are local and not synced from LDAP.

Providers

CLOUDRON

To use another Cloudron as the external LDAP directory, do the following:

- Enable `Directory Server` in the `User Directory` view of the other Cloudron. Be sure to whitelist this Cloudron and specify a secure secret.
- On this Cloudron, select `Cloudron` as the provider.

2FA Support

The Cloudron connector is the only one that supports 2FA. If the user has 2FA setup in the Cloudron LDAP Server, then 2FA is required to login. The Mandatory 2FA check is skipped for external users. The Mandatory 2FA flag can be enabled on the other Cloudron to enforce 2FA setup.

JUMPCLOUD

The following screenshot shows the available configure options using a jumpcloud external LDAP directory:

- `Server URL` : `ldaps://ldap.jumpcloud.com:636`
- `Base DN` : `ou=users, o=3214565, dc=jumpcloud, dc=com`
- `Filter` : `(objectClass=inetorgperson)`
- `Bind DN` : `uid=ldap_admin,ou=Users,o=3214565,dc=jumpcloud,dc=com`
- `Bind password` : `admin password`
- `Username field` : `uid`

OKTA

To use the Okta integration, do the following:

- In Okta, enable the LDAP interface. You can do this from the `Directory Integrations` page.
- By default, Okta uses email as the default uid. Cloudron requires usernames for LDAP integration to work. If you already have a field in Okta that can provide usernames, provide that as the `username field`. If not, you can create a new field in the profile editor and set that.
- Cloudron configuration (replace 'org' below):
    - `Server URL` : `ldaps://<org>.ldap.okta.com`
    - `Base DN` : `ou=users, dc=<org>, dc=okta, dc=com`
    - `Filter` : `(objectClass=inetorgperson)`
    - `Bind DN` : `uid=<admin>, dc=<org>, dc=okta, dc=com`
    - `Bind password` : `admin password`
    - `Username field` : see above

DISABLE

This disables External LDAP authentication. When disabled, Cloudron will switch all existing users to local.

Sync

The local directory is synced with the external directory every 4 hours.

To trigger a manual sync, click the `Sync` button. Be sure to check the logs to see any conflicts.

External users and groups have an icon in the User view:

> ⚠ **Users are not deleted**
>
> Currently, users removed from the external directory are not deleted from Cloudron during a sync. This is not a security issue because the user cannot authenticate anymore with the external directory.

AUTO-CREATE USERS

Use the `Automatically create users when they login to Cloudron` option to automatically create users locally on first login.

When not set, users are only Automatically created during Sync.

SYNC GROUPS

When `Sync Groups` is enabled, external groups will be created locally and users will be associated.

External Groups are readonly and cannot be edited. Therefore, local users cannot be added to external groups.

Local groups can still be created and they can have both local and external users as group members.

⚠ **Groups are not deleted**

Currently, groups removed from the external directory are not deleted from Cloudron during a sync.

SELF SIGNED CERTIFICATE

Use the `Accept Self-signed certificate` option to accept any self-signed certificate from the LDAP server.

4.19.5 LDAP Directory Server

Cloudron can act as a (readonly) LDAP server for apps hosted externally to Cloudron. External apps can then be configured to list Cloudron users and allow users to authenticate with their Cloudron password.

You can enable the Directory Server from the `User Directory` view:

For security reasons, the LDAP server will only accept connections from specific white listed IPs and ranges.

Configuring Clients

External apps can be configured to use the Directory Server as follows:

- Use the dashboard domain as the LDAP server hostname. Port 636 (TLS). LDAP server uses the same certificate as the dashboard domain.
- Set `cn=admin,ou=system,dc=cloudron` as the Bind DN. Use the secret listed in the above screenshot as the Bind Password.

Users:

- Users DN is `ou=users,dc=cloudron`.
- Use `user` as objectclass
- Username attribute is `uid`
- RDN attribute is `cn`
- UUID attribute is `uid`
- User LDAP filter is `(&(objectclass=user)(|(username=%uid)(mail=%uid)))` . Note that you will have to adjust the syntax based on the app. The objectclass might be redundant. `%uid` is a template variable for the username the user enters in the app's login form.

Groups:

- Groups DN is `ou=groups,dc=cloudron`.
- Use `group` as objectclass for groups
- Groupname attribute is `cn`
- Group LDAP filter is `(&(objectclass=group)(cn=%gname))` . Note that you will have to adjust the syntax based on the app . `%gname` is the group name you are searching for.

Troubleshooting

The setup can be tested as follow:

```
$ ldapsearch  -x -b "ou=users,dc=cloudron" -D "cn=admin,ou=system,dc=cloudron" -W -H ldaps://my.example.com:636
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <ou=users,dc=cloudron> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# uid-0cfbd3d8-6547-4332-9415-dadfe8b78ac4, users, cloudron
dn: cn=uid-0cfbd3d8-6547-4332-9415-dadfe8b78ac4,ou=users,dc=cloudron
objectclass: user
```

```
objectclass: inetorgperson
objectclass: person
objectcategory: person
...
```

4.19.6 OpenID Connect

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol.

Cloudron is an OIDC provider. The main advantages the OpenID integration offers over LDAP are:

- True single sign-on across apps. Once logged into the main dashboard, users can automatically login to apps.
- Manage app sessions from the Dashboard
- 2FA support across apps
- More secure since apps never see the user's password

Apps integrate automatically with the OIDC server using the oidc addon.

OIDC Provider Name

The OIDC Provider Name is the same as the Cloudron Name. Supporting apps will pick up this value and show this in the Login Button text.

Endpoints

Name	URL
Discovery URL	`https://my.cloudron.example/.well-known/openid-configuration` and `https://my.cloudron.example/openid/.well-known/openid-configuration`
Issuer URL	`https://my.cloudron.example/openid`
Auth URL	`https://my.cloudron.example/openid/auth`
Token URL	`https://my.cloudron.example/openid/token`
Keys URL (Certificate URL)	`https://my.cloudron.example/openid/jwks` and `https://my.cloudron.example/openid/jwks_rsaonly`
Profile URL	`https://my.cloudron.example/openid/me`

Cloudflare Access

Cloudflare Access has a bug that it gets confused by `OKP` key type. To circumvent, use the special RSA only keys URL.

Scopes and Claims

For most clients, it is recommend to add `openid` (mandatory), `profile` and `email` scope. Scopes are space separated. This will ensure the app has access to the user profile claims.

Cloudron currently provides the following scopes and corresponding claims:

Scope	Claim
`profile`	`family_name` , `given_name` , `locale` (always `en-US`), `name` , `preferred_username`
`email`	`email` , `email_verified`

On Cloudron, the `sub` property (the unique user identifier) is the **username**.

OIDC Clients

OIDC clients can be managed in the `User Directory` view. To create a new client, provide the `clientid`, `clientsecret` and the callback URL.

Then, in the client app, configure the provider as follows:

4.20 Volumes

4.20.1 Overview

Volumes are local or remote file systems. They can be used as an app's main data storage or as a shared storage location between apps.

4.20.2 Add

Volumes can be added in the `Volumes` view. Click the `Add Volume` button to add a volume and select the mount type.

Once added, the volumes can either be used as an app's data directory or be mounted into one or more apps.

Data Directory Limitation

Only volumes with Mount Type `EXT4` and `NFS` can be used as the data directory, as other Mount Types do not properly support file permissions.

4.20.3 Mount Type

Cloudron supports a variety of mount types. When using a mount type other than `No-op`, Cloudron will setup systemd mount config files to automatically mount on server start up. These mount points are created under `/mnt/volumes`.

Do not create fstab entry

When using the NFS/EXT4/CIFS/SSHFS/XFS providers, do not add an entry in `/etc/fstab` because Cloudron will already set up the mount via systemd. Use the `No-op` provider if you want to add an `/etc/fstab` entry.

CIFS

The CIFS mount type is used to mount CIFS shares. Note that, unlike EXT4 and NFS mount types, CIFS does not have a concept of users and groups. This makes it unsuitable for use as an app's data directory but will work fine for volumes and backups.

Hetzner Storage Box

When using Hetzner Storage Box, the Remote Directory is `/backup` for the main account. For sub accounts, the Remote Directory is `/subaccount`.

EXT4

The EXT4 mount type is used to mount external hard disks or block storage. To add an external EXT4 disk, first make sure the disk is formatted as EXT4 using `mkfs.ext4 /dev/<device>`. Then, run `blkid` or `lsblk` to get the UUID of the disk.

Filesystem

The Filesystem type is used for giving apps access to directories on the server. These are just directories on the local filesystem and do not require any mounting configuration. We recommend giving this directory `chmod 777` permissions for maximum compatibility across apps.

As a security measure, only host paths under `/mnt`, `/media`, `/srv` and `/opt` are allowed.

Filesystem (mountpoint)

When using the `mountpoint` mount type, Cloudron will not configure the server to mount the mount point. You have to set up `/etc/fstab` or systemd mount config files on your own. Use this if you want to set up an unsupported mount type or want to add specialized mount flags.

As a security measure, only mount points under `/mnt`, `/media`, `/srv` and `/opt` are allowed.

NFS

The NFS mount type is used to mount NFSv4 shares. If you need help setting up an NFS server, see this article. By default, NFS shares will change the root user to be owned by the `nobody` user. This is done for security purposes since it prevents creating files with setuid bit set. You can add `no_root_squash` to the options in the NFS server's exports file to circumvent this.

⚠ **Insecure traffic**

Please note that NFS traffic is unencrypted and can be tampered. For this reason, you must use NFS mounts only on secure private networks.

Noop

When using the `No-op` type, only paths under `/mnt`, `/media`, `/srv`, `/opt` can be added for security reasons.

SSHFS

The SSHFS mount type is used to mount a file system over SSH (using the SFTP protocol).

Cloudron does not support setting SSHFS volume as an app's data directory but will work fine for volumes and backups.

✎ **Hetzner Storage Box**

When using Hetzner Storage Box, the Remote Directory is `/home` for the main account. You can also leave this field empty. We have found sub accounts to be quite unreliable with SSFS. We recommend using CIFS instead if you want to use subaccounts.

XFS

The XFS mount type is used to mount external hard disks or block storage. To add an external XFS disk, first make sure the disk is formatted as XFS using `mkfs.xfs /dev/<device>`. Then, run `blkid` or `lsblk` to get the UUID of the disk.

4.20.4 Remount

Volumes can be remounted using the `Remount Volume` button. This is useful in situations where a networked volume got disconnected.

4.20.5 File manager

The File Manager can be used to access the volume's file system from the browser. Use the File Manager button to open the File Manager:

Clicking the icon will pop up a new window. Note that there are action like Rename, Delete, Change Ownership in the context menu.

4.20.6 Sharing

Sharing a volume across apps can be tricky because each app is packaged differently and the run-as user of each app varies.

The permissions and ACL of the mount directory have to be carefully set to make it work across apps. The general idea is to make file access work across multiple app containers by using the `media` group. The `media` group is currently hardcoded in the app containers to have the users www-data (uid 33) and cloudron (uid 1000). Most of the apps use one of these two users.

Do the following to prepare the volume for sharing:

- Identify the mount directory. For non-filesystem volumes, the mount directory is under `/mnt/volumes/<volume-id>`. For file system volumes, this is the host path. Set the `MOUNT_DIR` below accordingly.
- Run the following commands:

```
root@my:/# export MOUNT_DIR=/path/to/the/mount/directory
root@my:/# chmod 777 $MOUNT_DIR
root@my:/# chgrp media $MOUNT_DIR
root@my:/# chmod g+s $MOUNT_DIR
root@my:/# setfacl -d -m g::rwx $MOUNT_DIR
root@my:/# setfacl -d -m o::rx $MOUNT_DIR
```

The idea of the above commands is that newly created files are owned by the `media` group automatically. To double check that the changes were correctly applied:

```
root@my:/# touch $MOUNT_DIR/test-file
root@my:/# ls -l $MOUNT_DIR/test-file
-rw-rw-r-- 1 root media 0 Nov  3 22:41 /../test-file
```

As seen above, a newly created file has the correct permissions for access by `media` group automatically.

5. Guides

5.1 Connect to MongoDB Addon

5.1.1 Overview

Apps on Cloudron use an internal MongoDB database (addon). For security reasons, the database is only accessible from inside the server and cannot be accessed directly from outside. Instead, the database is accessed using an SSH tunnel to the server.

In this guide, we will see how to connect to the MongoDB addon database from outside (say, your laptop/PC).

5.1.2 Database credentials

To get the database credentials, open the Web terminal of an app and run `env | grep CLOUDRON_MONGODB`. Make a note of the credentials. Note that each app has it's own separate database and thus it's own database credentials.

5.1.3 Internal IP Address

The internal IP address of the MongoDB server is `172.18.30.3`.

5.1.4 DB Clients

CLI

To connect via MongoDB Shell , open a Web Terminal and click the `MongoDB` button at the top. This will paste the CLI connection command into the Web Terminal. Press enter to execute the command and use the CLI.

MongoDB Compass

MongoDB Compass is intuitive and flexible, provides detailed schema visualizations, real-time performance metrics, sophisticated querying abilities, and much more. It can be downloaded here.

Create a new connection and click on `Advanced Connection Options`. Fill in the server and authentication options from the step above:

> **Direct Connection**
>
> It is important to check the `Direct Connection` checkbox. Without this, the connection won't work.

Configure SSH tunnel by clicking on the Proxy/SSH tab:

You should now be able to view the database:

Studio 3T

Studio 3T is the Ultimate GUI for MongoDB. You can download it here.

Create a new MongoDB connection in Studio 3T via `File` -> `Connect`.

Choose manually enter values and fill up the values gathered in the steps above:

Configure SSH tunnel by clicking on the SSH tab:

You should now be able to view the database:

5.2 Connect to MySQL Addon

5.2.1 Overview

Apps on Cloudron use an internal MySQL database (addon). For security reasons, the database is only accessible from inside the server and cannot be accessed directly from outside. Instead, the database is accessed using an SSH tunnel to the server.

In this guide, we will see how to connect to the MySQL addon database from outside (say, your laptop/PC).

5.2.2 Database credentials

To get the database credentials, open the Web terminal of an app and run `env | grep CLOUDRON_MYSQL`. Make a note of the credentials. Note that each app has it's own separate database and thus it's own database credentials.

5.2.3 Internal IP Address

The internal IP address of the MySQL server is `172.18.30.1`.

5.2.4 DB Clients

CLI

To connect via MySQL CLI, open a Web Terminal and click the `MySQL` button at the top. This will paste the CLI connection command into the Web Terminal. Press enter to execute the command and use the CLI.

DBeaver

DBeaver is a free multi-platform database tool for developers, database administrators. You can download it here.

Create a new MySQL connection in DBeaver.

Fill up the values gathered in the steps above:

Configure SSH tunnel by clicking on the SSH tab:

You should now be able to view the database:

MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs.

Create a new connection in MySQL workbench.

Select `Standard TCP/IP over SSH` as the connection method. Fill up the values gathered in the steps above:

You should now be able to view the database:

5.3 Connect to PostgreSQL Addon

5.3.1 Overview

Apps on Cloudron use an internal PostgreSQL database (addon). For security reasons, the database is only accessible from inside the server and cannot be accessed directly from outside. Instead, the database is accessed using an SSH tunnel to the server.

In this guide, we will see how to connect to the PostgreSQL addon database from outside (say, your laptop/PC).

5.3.2 Database credentials

To get the database credentials, open the Web terminal of an app and run `env | grep CLOUDRON_POSTGRESQL`. Make a note of the credentials. Note that each app has it's own separate database and thus it's own database credentials.

5.3.3 Internal IP Address

The internal IP address of the PostgreSQL server is `172.18.30.2`.

5.3.4 DB Clients

CLI

To connect via PostgreSQL CLI, open a Web Terminal and click the `PostgreSQL` button at the top. This will paste the CLI connection command into the Web Terminal. Press enter to execute the command and use the CLI.

DBeaver

DBeaver is a free multi-platform database tool for developers, database administrators. You can download it here.

Create a new PostgreSQL connection in DBeaver.

Fill up the values gathered in the steps above:

Configure SSH tunnel by clicking on the SSH tab:

You should now be able to view the database:

5.4 Decrypt Backups

5.4.1 Overview

Cloudron supports encryption of backups with a passphrase. In this guide, we will see how to decrypt Cloudron backups on your laptop/PC.

5.4.2 Prerequisites

Have your PC/laptop setup with the Cloudron CLI tool. You can do this by `sudo npm install -g cloudron`.

Then, login using `cloudron login my.example.com`.

5.4.3 Download backup

This guide assumes you have the backup downloaded locally. If not, see this guide.

5.4.4 Decrypt

tgz

If the backup format is tgz, the download will be a single file with the extension `.tar.gz.enc`.

```
$ cloudron backup decrypt --password=passphrase backupid.tar.gz.enc backupid.tar.gz
$ tar zxvf backupid.tar.gz
```

rsync

If the backup format is rsync, the download will just be a directory.

```
$ mkdir /path/to/unencrypted
$ cloudron backup decrypt-dir --password=passphrase /path/to/encrypted /path/to/unencrypted
```

5.5 Download Backups

5.5.1 Overview

In this guide, we will see how to download backups.

5.5.2 Backup file names

Backups are categorized at the top level as timestamped directories. This timestamp indicates the time at which the backup was started. In addition, there is a directory called `snapshot`. The `snapshot` directory is a working directory for the backup logic and should not be touched.

Inside each timestamped directory, Cloudron stores the backup of each app individually. These backups have the file name `app_<location>_v<package_version>`. When using the `tgz` backup format, the file will have an extension `.tar.gz`. When using the `rsync` backup format, this will be a directory.

Email backups are stored with the file name `mail_v<box_version>.tar.gz` (or without the extension for `rsync`).

Cloudron stores the backup of the platform itself with the file name `box_v<box_version>.tar.gz` (or without the extension for `rsync`).

5.5.3 File System

To download a backup when using the `Filesystem` provider, simply use a tool like `scp`.

```
$ scp root@<ip>:/var/backups/2022-04-05-184006-496/app_wekan.cloudron.space_v4.19.2.tar.gz .
```

5.5.4 S3 (UI)

If you use one of the S3 compatible providers, you can usually go to the UI of the storage provider and find the latest timestamp. For example, in DO spaces:

In the screenshot above, we can see that `2021-03-10-182730-145` is the latest. Navigate inside the directory and download the files.

5.5.5 S3 (AWS CLI)

In some cases, some S3 providers do not have a UI to the object storage or the UI may simply fail to load with many objects. We can use the S3 CLI in that case to determine the backup ID.

- First, install the AWS CLI tool. On Linux, you can simply do `sudo apt install awscli`. For other platforms, see this guide.
- List the objects using the CLI tool. Note that sometimes you have to set the `AWS_DEFAULT_REGION` to be the region name of the provider and sometimes to `us-east-1`. This setting depends on your provider. In the example below, `cloudron-backups2` is the name of the bucket. If you provided a `prefix`, then you must list `s3://cloudron-backups2/prefix` in the example below.

```
$ export AWS_ACCESS_KEY_ID=JWAJUADXZJU2LBPBQLG2
$ export AWS_SECRET_ACCESS_KEY=SomeSecretKeyThatCannotBePublic
$ export AWS_DEFAULT_REGION=us-east-1
$ aws --endpoint-url=https://sfo2.digitaloceanspaces.com s3 ls s3://cloudron-backups2/
                           PRE 2021-03-02-162423-447/
                           PRE 2021-03-02-162651-715/
                           PRE 2021-03-05-072427-396/
                           PRE 2021-03-10-182730-145/
                           PRE snapshot/
```

We can see above that the latest timestamp is `2021-03-10-182730-145`. Now, we can list the contents of that directory.

```
$ aws --endpoint-url=https://sfo2.digitaloceanspaces.com s3 ls s3://cloudron-backups2/2021-03-10-182730-145/
2021-03-10 10:27:33    7129765 app_wekan.cloudron.space_2021-03-10-182732-640_v1.2.0.tar.gz
2021-03-10 10:27:34     422273 box_2021-03-10-182734-919_v6.2.3.tar.gz
```

Use `aws s3 sync` to download the file.

5.6 GitHub Action: Push Repo to Cloudron App

5.6.1 Overview

This guide explains how to set up the GitHub Action: Cloudron Push to App to automatically push your application to a Cloudron instance app whenever you push changes to your GitHub repository.

As an example, you have a custom web application and want to deploy it to a Cloudron LAMP app whenever you push changes to your GitHub repository.

Example repository: cloudron-io/github-action-test-repo.

This repo contains a simple `index.html` , `index.css` and `composer.json` file and is configured to use the GitHub Action to push to a Cloudron app. You can copy the `.github/workflows/deploy-to-cloudron.yaml` file from this repository to your own repository to get started quickly. You will still need to set up the GitHub Secrets as described below.

- Action repository: cloudron-io/cloudron-push-to-app
- GitHub Marketplace: Cloudron Push to App

5.6.2 Setup Instructions

1. Create a Cloudron API Token with read-write permissions and save it, you will need it in the next step
2. Add the API Token as a GitHub Secret
    a. Go to your GitHub repository
    b. Navigate to `Settings` > `Environments`
    c. Create a new environment with a name of your choice (e.g., `cloudron-deploy`) - each environment is for one Cloudron instance
    d. In the newly created environment, click `Add environment secret` and add the following secrets:
        i. `CLOUDRON_FQDN` : The FQDN of your Cloudron instance (e.g., `my.demo.cloudron.io`)
        ii. `CLOUDRON_TOKEN` : The API token you created in step 1
        iii. `CLOUDRON_APP_ID` : The App ID of the Cloudron app where you want to deploy your application. You can find the App ID in the Cloudron dashboard when configuring the app in the URL. Alternatively, you can also use the location e.g., `lamp` or fully qualified domain name e.g., `lamp.demo.cloudron.io`

3. Create a GitHub Actions Workflow

   a. In your GitHub repository, create the folder `.github/workflows` if it doesn't already exist

   ```
   mkdir -p .github/workflows
   ```

   b. Create a new file named `deploy-to-cloudron.yaml.yml` in the `.github/workflows` folder with the following content:

   ```yaml
   # this is the rule when the action will be triggered - see https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions#on
   # this rule triggers the action on every push to the main branch
   on:
     push:
       branches:
         - main
   jobs:
     # this is the job name definition - you can name it as you like
     deploy-to-cloudron-app:
       # this is the runner definition, you can think of this as the operating system where the job will run - here we use the ubuntu-latest
       runs-on: ubuntu-latest
       # the environment that we created in step 2
       environment: cloudron-deploy
       # the steps to execute in this job
       steps:
         # Name of the step and which action to use - here we checkout the repository code with https://github.com/actions/checkout
         - name: Checkout Repository
           uses: actions/checkout@v6
         # Name of the step and which action to use - here we use the Cloudron Push to App action
         - name: Cloudron Push to App
           uses: cloudron-io/cloudron-push-to-app@latest
           # the parameters/variables to pass to the action
           with:
             # secret CLOUDRON_FQDN that we created in step 2 for the environment cloudron-deploy
             CLOUDRON_FQDN: "${{ secrets.CLOUDRON_FQDN }}"
             # secret CLOUDRON_TOKEN that we created in step 2 for the environment cloudron-deploy
             CLOUDRON_TOKEN: "${{ secrets.CLOUDRON_TOKEN }}"
             # secret CLOUDRON_APP_ID that we created in step 2 for the environment cloudron-deploy
             CLOUDRON_APP_ID: "${{ secrets.CLOUDRON_APP_ID }}"
             # optional: specify the destination path in the Cloudron app where the files should be pushed - default is /app/data/public
             # the action will copy all files from the repository root to this destination path
             CLOUDRON_PUSH_DESTINATION: "/app/data/public"
             # optional: create a backup of the app before deploying - default is "true"
             CLOUDRON_CREATE_APP_BACKUP: "true"
   ```

4. Commit and push the changes to your GitHub repository

5. Monitor the GitHub Actions tab in your repository to see the deployment progress

Multiple Environments - Staging and Live

You can also create multiple environments for e.g., `live` and `staging` and have a different `CLOUDRON_FQDN`, `CLOUDRON_TOKEN` and `CLOUDRON_APP_ID` and use the environments in another workflow that only triggers on a specific branch e.g., `develop`.

This way you can have a staging and live deployment setup

5.6.3 Customizing GitHub Action

You can add more steps to the GitHub Actions workflow before the `Cloudron Push to App` step to prepare your application for deployment.

For example, if your application requires Composer dependencies to be installed, you can add a step to install the dependencies before pushing the app to Cloudron:

```yaml
on:
  push:
    branches:
      - main
jobs:
  deploy-to-cloudron-app:
    runs-on: ubuntu-latest
    environment: my.cloudron.dev
    steps:
      - name: Checkout Repository
        uses: actions/checkout@v6
      # Extra step to install composer dependencies
      - name: Install composer dependencies
        uses: php-actions/composer@v6
      - name: Cloudron Push to App
        uses: cloudron-io/cloudron-push-to-app@latest
        with:
          CLOUDRON_FQDN: "${{ secrets.CLOUDRON_FQDN }}"
          CLOUDRON_TOKEN: "${{ secrets.CLOUDRON_TOKEN }}"
          CLOUDRON_APP_ID: "${{ secrets.CLOUDRON_APP_ID }}"
          CLOUDRON_PUSH_DESTINATION: "/app/data/public"
          CLOUDRON_CREATE_APP_BACKUP: "true"
```

5.7 Import email

5.7.1 Overview

In this guide, we will see how to import existing email from an external email provider to Cloudron Mail. Importing email works by fetching email via IMAP from the external provider and pushing it to Cloudron Mail server via IMAP.

Various tools like imapsync, OfflineIMAP and isync can be used for this purpose. In this guide, we will use imapsync.

5.7.2 Add domain

The first step is to add the domain in Cloudron and enable IMAP access for the domain on the Cloudron mail server. To do so, add the domain in `Domain` view. Then in the `Email` view, select the domain and click `Enable` in the `Incoming` section.

At this point, we only want to import email and do not want Cloudron to start receiving emails for the domain. For this reason, uncheck the `Setup Mail DNS records now` option when enabling email for the domain. This will allow you to continue using email with your current provider.

5.7.3 Create mailbox

Next, add the mailbox(es) that you want to import in Cloudron in the `Email` view.

5.7.4 Imapsync

imapsync can be download to your PC from here. imapsync will do an incremental and recursive transfer from one mailbox to another.

To import say `founders@cloudron.club`, use something like the following command:

```
imapsync \
    --host1 imap.external.server --user1 founders@cloudron.club --password1 externalmailboxpassword \
    --host2 my.cloudron.club --user2 founders@cloudron.club --password2 cloudronmailboxpassword
```

The above command has to be repeated for each mailbox you have on the external server.

There are various caveats when importing mail from GMail, you can read more here. An example usage for GMail would be:

```
imapsync --gmail1 --user1 founders@cloudron.club --password1 MASKED \
    --host2 my.cloudron.club --user2 founders@cloudron.club --password2 MASKED \
    --maxbytespersecond 20000 --useheader=X-Gmail-Received --useheader Message-Id --automap --regextrans2 s,\[Gmail\].,, --skipcrossduplicates   --folderlast
[Gmail]/All Mail --exclude "\[Gmail\]/Spam"
```

5.7.5 Verify

Once imported, you can verify if the mails look correct on Cloudron using apps like SnappyMail or Roundcube.

5.7.6 Finish

The final step is to switch over the domain's DNS to use Cloudron mail. This can be done by clicking the `Re-setup DNS` button in the `Email` -> Select domain -> `Status` page.

5.8 Import MongoDB

5.8.1 Overview

In this guide, we will see how to export a MongoDB database from your current setup and import it into the MongoDB database of a Cloudron app.

5.8.2 Export

The first step is to create a dump in your existing MongoDB database setup:

```
$ mongodump -h host:port -u mongouser -p mongopassword --db databasename --out /tmp
$ tar -C /tmp -zcvf /tmp/mongodump.tar.gz databasename
```

If the MongoDB server is on Cloudron, you can export it using this command line on the Web Terminal:

```
# mongodump -u "${CLOUDRON_MONGODB_USERNAME}" -p "${CLOUDRON_MONGODB_PASSWORD}" -h ${CLOUDRON_MONGODB_HOST}:${CLOUDRON_MONGODB_PORT} --db $
{CLOUDRON_MONGODB_DATABASE} --out /tmp

# tar -C /tmp -zcvf /tmp/mongodump.tar.gz ${CLOUDRON_MONGODB_DATABASE}
```

5.8.3 Import

- After install, enable 'Recovery Mode'in the `Repair` section. This will ensure that the app is paused and not actively using the database while you are importing.
- Open a Web Terminal by using the Terminal button in the `Console` section.
- In the terminal, upload the dump file using the `Upload to /tmp` button.
- Extract the dump:

```
root@e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72:/tmp# tar -zxvf mongodump.tar.gz
e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72/
e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72/rocketchat_custom_sounds.metadata.json
e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72/users.metadata.json
e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72/rocketchat_custom_sounds.bson
...
```

- Clear the existing database:

```
# mongosh -u "${CLOUDRON_MONGODB_USERNAME}" -p "${CLOUDRON_MONGODB_PASSWORD}" ${CLOUDRON_MONGODB_HOST}:${CLOUDRON_MONGODB_PORT}/${CLOUDRON_MONGODB_DATABASE}
--eval 'db.getCollectionNames().forEach(function (col) { db.getCollection(col).deleteMany({}) })'
```

- In the terminal, import the dump using the `mongorestore` command.

```
# mongorestore -u "${CLOUDRON_MONGODB_USERNAME}" -p "${CLOUDRON_MONGODB_PASSWORD}" -h ${CLOUDRON_MONGODB_HOST}:${CLOUDRON_MONGODB_PORT} --db $
{CLOUDRON_MONGODB_DATABASE} --dir=/tmp/e6bb147e-9f9c-4d17-a9af-65d9fbb0dd72/
```

- To start the app again, click the `Disable Recovery Mode` in `Repair` section:

5.8.4 Verify

Click the `MongoDB` button on top of the terminal to paste the command line required to access the MongoDB database. You can now press enter/return to get the MongoDB shell.

5.9 Import MySQL

5.9.1 Overview

In this guide, we will see how to export a MySQL database from your current setup and import it into the MySQL database of a Cloudron app.

5.9.2 Export

The first step is to create a dump in your existing MySQL database setup:

```
mysqldump -hmyservername -umyusername -pmypassword --single-transaction --routines --triggers databasename > mysqldump.sql
```

If the MySQL server is on Cloudron, you can export it using this command line on the Web Terminal:

```
mysqldump -h${CLOUDRON_MYSQL_HOST} -u${CLOUDRON_MYSQL_USERNAME} -p${CLOUDRON_MYSQL_PASSWORD} --single-transaction --routines --triggers --no-tablespaces ${CLOUDRON_MYSQL_DATABASE} > /tmp/mysqldump.sql
```

5.9.3 Import

- After install, enable 'Recovery Mode'in the `Repair` section. This will ensure that the app is paused and not actively using the database while you are importing.
- Open a Web Terminal by using the Terminal button in the `Console` section.
- In the terminal, upload the dump file using the `Upload to /tmp` button.
- In the terminal, clear the existing database:

```
mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST} ${CLOUDRON_MYSQL_DATABASE} -Nse 'show tables' | while read table; do mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST} ${CLOUDRON_MYSQL_DATABASE} -e "SET FOREIGN_KEY_CHECKS = 0; drop table \`$table\`"; done
```

- In the terminal, import the dump using the `mysql` command. Click the `MySQL` button on top of the terminal to paste the command line required to access the MySQL database and redirect the dump file to the command:

```
mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST} ${CLOUDRON_MYSQL_DATABASE} < /tmp/mysqldump.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
root@838249e2-d2ae-4a40-80bf-4f1632e0d376:/app/code#
```

An empty output like above means the command succeeded.

- To start the app again, click the `Disable Recovery Mode` in `Repair` section:

5.9.4 Verify

Click the `MySQL` button on top of the terminal to paste the command line required to access the MySQL database. You can now press enter/return to get the MySQL shell.

5.10 Import PostgresSQL

5.10.1 Overview

In this guide, we will see how to export a PostgreSQL database from your current setup and import it into the PostgreSQL database of a Cloudron app.

5.10.2 Dump

The first step is to create a dump of your existing PostgreSQL database. This can be done using `pgdump`:

```
$ PGPASSWORD=password pg_dump --no-owner --no-privileges --username=username --host=myserver databasename > pgdump.sql
```

If this database is on a Cloudron, you can use the following command:

```
# PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} pg_dump --no-owner --no-privileges --username=${CLOUDRON_POSTGRESQL_USERNAME} --host=${CLOUDRON_POSTGRESQL_HOST}
${CLOUDRON_POSTGRESQL_DATABASE} > /tmp/pgdump.sql
```

5.10.3 Import

- After install, enable 'Recovery Mode'in the `Repair` section. This will ensure that the app is paused and not actively using the database when you are importing.
- Open a Web Terminal by using the Terminal button in the `Console` section.
- Upload the dump file using the `Upload` button.
- The dump file might contain extension information that needs to be first commented out.

```
# sed -e 's/CREATE EXTENSION/-- CREATE EXTENSION/g' -e 's/COMMENT ON EXTENSION/-- COMMENT ON EXTENSION/g' /tmp/pgdump.sql > /tmp/pgdump_mod.sql
```

- Clear the existing database

```
# PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d $
{CLOUDRON_POSTGRESQL_DATABASE} -c "DROP SCHEMA public CASCADE; CREATE SCHEMA public"
```

- Finally, import the dump using the `psql` command:

```
# PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d $
{CLOUDRON_POSTGRESQL_DATABASE} --set ON_ERROR_STOP=on --file=/tmp/pgdump_mod.sql
```

5.10.4 Verify

Click the `PostgreSQL` button on top of the terminal to paste the command line required to access the PostgreSQL database. You can now press enter/return to get the PostgreSQL shell.

5.11 Import Redis

5.11.1 Overview

In this guide, we will how to import redis database from your current setup into a Cloudron app.

Cloudron uses Redis' RDB persistence format for point-in-time snapshots at specified intervals. To import the RDB, you must first create a dump file in your existing redis installation. You can do this by sending redis the `SAVE` command.

5.11.2 Import

- Stop the app using the stop button in `Console` section. This will stop any dependent redis container as well.
- Identify, the app's id from the `Updates` section.
- Copy over the redis dump file via SCP. Be sure to replace `server-ip` with your server's IP address and the app id `85a26cdf-3858-4784-b2c1-6ddb9c37e5b9` with your app's id below.

```
$ scp dump.rdb root@server-ip:/home/yellowtent/platformdata/redis/85a26cdf-3858-4784-b2c1-6ddb9c37e5b9/dump.rdb
dump.rdb                                                                                          100%   692     4.8KB/s   00:00
```

- Start the app using the start button in `Console`.

5.11.3 Verify

- Open a Web Terminal by using the Terminal button in the `Console` section.
- Click the 'Redis' button on top of the terminal window. This will paste the Redis CLI command required to access redis. Simply, press enter to start using the Redis shell.

5.12 Running Laravel Apps

5.12.1 Overview

In this guide, we will see how to run Laravel apps on Cloudron using the LAMP stack.

5.12.2 Install LAMP

First, install the LAMP app on Cloudron.

5.12.3 Increase memory limit

As the next step, bump the memory limit of the LAMP app to 1GB. This is required for composer to run reliably.

5.12.4 Create Laravel Project

Open a Web Terminal and create a Laravel app using composer. We switch to `www-data` user because the web server runs as that user. Then, we switch the directory to `/app/data` and use composer to create a empty Laravel project.

```
root@7c29d3b7-b93d-4c75-932e-c771a7383e39:/app/code# su - www-data
www-data@7c29d3b7-b93d-4c75-932e-c771a7383e39:~$ cd /app/data
www-data@7c29d3b7-b93d-4c75-932e-c771a7383e39:~$ composer create-project laravel/laravel my-project
Creating a "laravel/laravel" project at "./my-project"
Installing laravel/laravel (v8.5.18)
  - Installing laravel/laravel (v8.5.18): Downloading (100%)
Created project in /app/data/my-project
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Warning from https://repo.packagist.org: You are using an outdated version of Composer. Composer 2 is now available and you should upgrade. See https://
getcomposer.org/2
Updating dependencies (including require-dev)
Package operations: 104 installs, 0 updates, 0 removals
  - Installing voku/portable-ascii (1.5.6): Downloading (100%)
  - Installing symfony/polyfill-php80 (v1.22.1): Downloading (100%)
  - Installing symfony/polyfill-mbstring (v1.22.1): Downloading (100%)
  - Installing symfony/polyfill-ctype (v1.22.1): Downloading (100%)
  - Installing phpoption/phpoption (1.7.5): Downloading (100%)
....
> @php artisan key:generate --ansi
Application key set successfully.
```

5.12.5 Configure Apache

By default, Apache is configured to serve the `/app/data/public` directory. The Laravel public directory is however located under `/app/data/my-project/public`. To configure Apache, use the File Manager and edit `/app/data/apache/app.conf`. After saving the config, restart the app for the changes to take effect.

```
ServerName %{HTTP_HOST}

<VirtualHost *:80>
    DocumentRoot /app/data/my-project/public

    LogFormat "%{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" proxy
    CustomLog "|/bin/cat" proxy
    ErrorLog "|/bin/cat"

    <Directory /app/data/my-project/public>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # Do not remove this include. It's required for your app to see the Real IP
    Include "/app/code/apache/rpaf.conf"
    # This line can be commented out, if you do no require PHPMyAdmin Access
    Include "/app/code/apache/phpmyadmin.conf"

</VirtualHost>
```

After app restart, you should see the default Laravel page:

5.12.6 Configure Queue Worker

To configure Laravel Queue worker, edit `/app/data/run.sh` with the following line and restart the app:

```
sudo -u www-data /usr/bin/php /app/data/my-project/artisan queue:work --queue=high,standard,low --sleep=3 --tries=3 &
```

5.12.7 Cron Jobs

Laravel also depends on a scheduler for tasks scheduled for later. Those can be run via cron. Add the following job into the crontab as mentioned in the cron docs:

```
* * * * * cd /app/data/my-project/ && sudo -u www-data /usr/bin/php /app/data/my-project/artisan schedule:run
```

5.13 Load Large Data into MySQL

5.13.1 Overview

In this guide, we will see how to load large data into a MySQL database using the `LOAD DATA INFILE` mechanism. If your data is not very large, it's best to just import data using a MySQL client connect - either via a program or `mysql` client from the app. This guide is only worth the trouble if you want to load several GB of data.

> ⚠ **SSH access required**
>
> To follow this guide, SSH access to the server and basic docker knowledge is required.

5.13.2 Copy data

The `LOAD DATA INFILE` command works by importing a file which is located on the same file system as the MySQL server. On Cloudron, this means that the data file must be located under `/run/mysql-files/` of the `mysql` container.

- First, copy the data file, says `data.csv`, into some location on the server.
- Next, copy the data file into the `mysql` container.

```
root@my:~# docker cp data.csv mysql:/run/mysql-files/data.csv
```

5.13.3 Load data

To load the data, you have to execute `LOAD DATA INFILE` from inside the `mysql` container.

```
root@my:~# docker exec -ti mysql /bin/bash
root@mysql:/# mysql -uroot -p${CLOUDRON_MYSQL_ROOT_PASSWORD}
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)

mysql>
```

With the MySQL prompt, you can load the data into the databases. To identify the database of your app, check the `CLOUDRON_MYSQL_DATABASE` environment variable in your app's Web Terminal.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| 405ae49cca94474c   |

...

mysql> use 405ae49cca94474c;
mysql> LOAD DATA INFILE '/run/mysql-files/data.csv' INTO TABLE employees FIELDS TERMINATED BY ',';
Query OK, 5 rows affected (0.00 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 0
```

5.14 Migrate existing WordPress installation to Cloudron

5.14.1 Overview

In this guide, we will see two approaches to migrate a WordPress installation from another server or service to Cloudron. There are no specific limitations of either approach and the method to choose depends on what tools are avaiable in your existing hosting.

5.14.2 Migration without a plugin

This approach requires access to a File Manager for downloading site content and PHPMyAdmin for creating a database dump.

Export WordPress

The first step is to export the contents of your existing WordPress installation.

- Export the `wp-content/` folder on the origin server to a wp-content.zip file
- Export the database on the source server to a .sql file. In PHPMyAdmin or equivalent, select all the tables, and export to `wordpress.sql`

Install WordPress on Cloudron

Install the WordPress Developer app on Cloudron.

Import WordPress Content

Go to the file manager of the WordPress installation and into the `public/` folder.

- Upload the wp-content.zip file
- Delete existing `wp-content/` folder
- Unzip the file wp-content.zip
- Change ownership to `www-data` from `wp-content/` folder which has been unzipped

Import WordPress Database

Upload the `wordpress.sql` database dump from the initial step to the path `/app/data`. Go the Web Terminal of the app and type the following:

```
# cd /app/data
# wp db reset
# wp db import wordpress.sql
```

Verify

At this point, the migration is complete and your site should be live. For good measure, restart the app to make sure that the site works across restarts.

Replacing URLs

Some WordPress core files, themes and plugins store the entire URL in the database (as opposed to just a relative URL). The WP CLI tool has a handy `search-replace` subcommand to replace strings in the database.

Open a Web Terminal and run the command below:

```
# sudo -E -u www-data /app/pkg/wp --path=/app/data/public/ search-replace 'https://old.cloudron.space' 'https://new.cloudron.space'
# sudo -E -u www-data /app/pkg/wp --path=/app/data/public/ cache flush
```

5.14.3 Migration with All-in-One WP Migration plugin

This approach requires installing the All-in-One WP Migration plugin on your existing site.

Export WordPress

First install the `All-in-One WP Migration` plugin and activate it. Post activation, click on `All-in-One WP Migration` in the side pane and click Export to file. Download the migration file (has a .press extension).

Install WordPress on Cloudron

Install the WordPress Developer app on Cloudron.

Import WordPress Content

Install the `All-in-One WP Migration` plugin on the Cloudron installation and activate it. Post activation, click on `All-in-One WP Migration` in the side pane and click Import.

Verify

At this point, the migration is complete and your site should be live. For good measure, restart the app to make sure that the site works across restarts.

5.14.4 Migration with UpdraftPlus plugin

This approach requires installing the UpdraftPlus plugin on your existing site.

Export WordPress

First install the `UpdraftPlus` plugin and activate it. Post activation, create a backup.

Download all the backup files once the backup is complete. As of this writing, there were 5 types of files - Database, themes, plugins, uploads & others. You have to download each individually.

Install WordPress on Cloudron

Install the WordPress Developer app on Cloudron.

Import WordPress Content

Install the `UpdraftPlus` plugin on the Cloudron installation and activate it. Post activation, upload the backup files and click `Restore`.

You can ignore the warning about the migration.

Verify

At this point, the migration is complete and your site should be live. For good measure, restart the app to make sure that the site works across restarts.

5.15 Mailbox Sharing

5.15.1 Overview

Cloudron 6 introduced IMAP mailbox sharing. This feature will work by default on new Cloudron 6 installations. For Cloudrons that were installed before 6.0, some manual steps are required to make mailbox sharing work.

5.15.2 Background

Cloudron uses dovecot as the IMAP server. The default namespace separator in dovecot is `.`. However, this separator conflicts with the mailbox sharing feature This is because Cloudron Email server users the user's email as the username and email addresses have a `.` in the domain portion.

The solution is to migrate to using `/` as the namespace separator. New installations use this separator by default. Existing installations must be migrated manually to use this separator.

⚠ **Automated migration**

The reason this migration is not automated is because there is a small chance that email clients might have cached this separator. For such clients, one will have to re-add the email account. At the time of writing, we are collecting information on what possible mail clients will be affected. Roundcube, SnappyMail, SOGo, Thunderbird do not have a problem, but you might have to logout and login again. On Android K-9, one has to 'Refresh Folder List' on the account.

5.15.3 Pre-flight

Before proceeding, please check what separator is used by the mail server. You can do this by inspecting the file `/home/yellowtent/boxdata/mail/dovecot/config.ini`. If `namespace_separator` has `/`, then you can skip this guide entirely. If it has `.`, then read on.

5.15.4 Migrate

SSH into the server and run the following commands to migrate to `/` separator.

```
cd /home/yellowtent/boxdata/mail/vmail
find . -type f -name '*.sieve' -exec sed -e '/fileinto/{s,\.,/,g}' -i.bak '{}' \;
sed -i -e 's/namespace_separator=.*/namespace_separator=\//' /home/yellowtent/boxdata/mail/dovecot/config.ini
docker restart mail
```

The above script creates a backup of all the sieve scripts. You can safely delete them once all your users have reported back that there are no issues.

```
find . -type f -name '*.sieve.bak' -exec rm '{}' \;
```

5.16 NFS Share

5.16.1 Overview

It can be sometimes useful to expose application data as a NFS share. For example, you might want to copy a large number of media files into (or from) Emby/Jellyfin or copy images into Surfer etc.

In this guide, we will see how to expose directories on Cloudron via NFS.

⚠ **Insecure traffic**

Please note that NFS traffic is unencrypted and can be tampered. For this reason, you must use NFS mounts only on secure private networks.

5.16.2 Install NFS Server

To install, run the following command on Cloudron server:

```
sudo apt install nfs-kernel-server
```

5.16.3 Disable NFSv3

By default, the NFS server support v3 and v4. v3 has various security implications and we recommend disabling it.

- Edit `/etc/nfs.conf` for Ubuntu 22.04 and newer, older systems should edit the `/etc/default/nfs-kernel-server` file and add:

```
RPCNFSDOPTS="-N 2 -N 3"
```

- Restart the server using `systemctl restart nfs-kernel-server`
- Verify it got disabled by checking `cat /proc/fs/nfsd/versions`

```
-2 -3 +4 +4.1 +4.2
```

- Disable the `rpcbind` service. This is required only for NFSv3.

```
systemctl disable rpcbind.socket rpcbind.service
systemctl stop rpcbind.socket rpcbind.service
```

5.16.4 Exposing a directory

Edit `/etc/exports` and add a line like so:

```
# this exposes data of the app with id appid to the Client IP address client_ip
/home/yellowtent/appsdata/app_id/data client_ip(rw,sync,no_subtree_check,no_root_squash)
```

Meaning of the options:

- `rw` : can read and write to volume
- `sync` : server replies to requests only after the changes have been committed to stable storage
- `no_subtree_check` : when exporting subdirectories, the server skips checking if every file access is still in the originally exported filesystem
- `no_root_squash` : the root user of client is mapped to root user on server as well.

5.16.5 Export the directory

Export the NFS directory that we configured above:

```
exportfs -a
systemctl restart nfs-kernel-server
```

5.16.6 Expose NFS port

Port 2049 (TCP/UDP) is used for NFS traffic. Add this to Cloudron Firewall by editing `/home/yellowtent/platformdata/firewall/ports.json`:

```
{
    "allowed_tcp_ports": [ 2049 ],
    "allowed_udp_ports": [ 2049 ]
}
```

Restart the firewall to apply the configuration:

```
systemctl restart cloudron-firewall
```

5.16.7 Mounting on client

Add the following entry to your laptop/PC's `/etc/fstab`:

```
cloudron_ip:/home/yellowtent/appsdata/app_id/data              /mounts/app      nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

5.17 Multiple databases with LAMP app

5.17.1 Overview

Custom PHP apps can be hosted on Cloudron using the LAMP app. This app provides the PHP code access to a single MySQL database. The LAMP app cannot create another database because of Cloudron's isolation and sandboxing practices. In this guide, we will see how to create a custom LAMP app that can use multiple MySQL databases.

5.17.2 Prerequisites

If you are on Windows or have a slow internet connection, we recommend just using a Ubuntu 18.04 VPS. A small 1GB droplet from Digital Ocean will do.

Have your PC/laptop setup with the following tools:

- Cloudron CLI tool - You can do this by `sudo npm install -g cloudron`. Then login to Cloudron using `cloudron login my.example.com`.
- Docker
- A free account at Docker Hub or just about any other Docker registry. Use `docker login` to login into the docker hub account.

5.17.3 Building custom app

Start out by cloning the LAMP 7.4 multidb app package

```
~$ git clone https://git.cloudron.io/cloudron/lamp7.4-multidb-app.git
Cloning into 'lamp7.4-multidb-app'...
Receiving objects: 100% (841/841), 225.28 KiB | 1.06 MiB/s, done.
Resolving deltas: 100% (552/552), done.
```

For the curious, the main difference between this app and LAMP 7.4 app package in the App Store is that this uses the `multipleDatabases` option for `mysql` addon in `CloudronManifest.json`. When using this option, an environment variable named `CLOUDRON_MYSQL_DATABASE_PREFIX` will be set instead of `CLOUDRON_MYSQL_DATABASE`.

Next, build the app using `cloudron build`:

```
~/lamp7.4-multidb-app$ cloudron build --local
Enter repository (e.g registry/username/lamp.cloudronapp.php73): girish/lamp-multidb

Building locally as girish/lamp-multidb:20200518-010956-446902496
...
6597da2e2e52: Layer already exists
977183d4e999: Layer already exists
c8be1b8f4d60: Layer already exists
20200518-012539-407f7a758: digest: sha256:2ed41f543da49d1504fc3994efd107d8a4034dee753731f3073f885cfcf02bed size: 7819
```

In the example above, `girish` is my docker hub username and `lamp-multidb` is the repository name. The build command will build the Dockerfile locally and push the resulting image to Docker Hub with a tagged timestamp (as you can see from the output above).

5.17.4 Install app

The built app can now be installed using `cloudron install`:

```
~/lamp-multidb-app$ cloudron install
Location: lamp-multidb
App is being installed.

 => Queued
 => Registering subdomains
 => Downloading image ....
 => Setting up addons .............
 => Creating container
 => Wait for health check ........

App is installed.
```

The app is now available at `lamp-multidb.example.com`! You can use `cloudron open` to open it in the browser.

5.17.5 Creating databases

When using the `multipleDatabases` option, Cloudron does not create any databases. Instead, it creates a MySQL user that has the permission to create databases with a prefix.

To create a database, simply use `cloudron exec`. This gives you a shell which has the same context as the app. By using the `env` command in the shall, we can see that `CLOUDRON_MYSQL_DATABASE_PREFIX` is set.

```
~/lamp-multidb-app$ cloudron exec
root@9bbac34d-452d-49ab-beac-a90f90549085:/app/data# env | grep CLOUDRON_MYSQL_
CLOUDRON_MYSQL_PORT=3306
CLOUDRON_MYSQL_PASSWORD=cd544c6690c6b6f903b037e24b0f9293d6dfc56fc6c02e05
CLOUDRON_MYSQL_USERNAME=e26ceb5eff7b17b8
CLOUDRON_MYSQL_DATABASE_PREFIX=e26ceb5eff7b17b8_
CLOUDRON_MYSQL_HOST=mysql
```

We can now use the MySQL CLI tool to create any number of databases with this prefix.

```
root@9bbac34d-452d-49ab-beac-a90f90549085:/app/data# mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=$
{CLOUDRON_MYSQL_HOST}
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2701
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

mysql> CREATE DATABASE `e26ceb5eff7b17b8_db1`;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE DATABASE `e26ceb5eff7b17b8_db2`;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| e26ceb5eff7b17b8_db1 |
| e26ceb5eff7b17b8_db2 |
+----------------------+
3 rows in set (0.00 sec)
```

5.17.6 Upload App

The app can be uploaded via SFTP. SFTP credentials are available in the `Access Control` section of the Cloudron dashboard.

In PHP code, it is recommended to use the database names as `getenv('CLOUDRON_MYSQL_DATABASE_PREFIX') . db1` instead of the hardcoded prefix. This makes it possible to use Cloudron's clone feature to clone the app and not make any changes.

5.17.7 Update App

All the databases are part of the app's backup! The app package we cloned also has `redis` and `sendmail` addons. You can remove them from the manifest, if you don't intend to use them and adjust `start.sh` accordingly.

As this app was custom built, you have the responsibility of keeping it up-to-date. This is fairly easy. All you have to do is `cloudron build` and `cloudron update`:

```
~/lamp-multidb-app$ cloudron build
Building locally as girish/lamp-multidb:20200518-014229-318890494

Sending build context to Docker daemon  53.76kB
...

20200518-014229-318890494: digest: sha256:48215586dbcd980abb6e467b353e5bff42285271368d7c7daede24291497af77 size: 7819


~/tmp/lamp-multidb-app$ cloudron update

 => Waiting for app to be updated
 => Queued
 => Backup - Snapshotting app lamp-multidb.cloudron.fun .
 => Backup - Copying XJivOprCjge3aQ9woWIDYeyNrZ0m-CuDFbkXW95brWE (lamp-multidb.cloudron.fun)
 => Downloading image
 => Updating addons
 => Creating container
 => Wait for health check ....

App is updated.
```

5.18 Per-App storage limit

This guide explains how one can set-up disk storage limits for singular or multiple apps

Working with disks can lead to data loss - always have a backup ready

Example

The app `lamp1.cloudron.dev` should have a 10GB disk storage limitation.

For simplicity the external disk will always be named `/dev/sdc`.

5.18.1 External EXT4 block storage

Disk that has 10 GB storage

CREATING THE FILESYSTEM

Create the EXT4 file system on the disk:

```
mkfs.ext4 /dev/sdc
```

MOUNTING THE DISK

Mount the disk to `/mnt/lamp1`:

```
# Create the directory for mounting first
mkdir -p /mnt/lamp1
# Mount the device to the directory
mount /dev/sdc /mnt/lamp1
```

Backup your `fstab` file

Editing the `fstab` can be dangerous, always create a backup first!

```
cp /etc/fstab /etc/fstab_backup_$(date +%d.%m.%Y-%H:%M:%S)
```

For more details for `fstab` please read the Arch Wiki - Fstab

Add a `fstab` entry for `/dev/sdc` so the disk gets mounted when the server boots:

```
/dev/sdc /mnt/lamp1 ext4 defaults,noatime,nofail 0 2
```

CONFIGURING CLOUDRON AND THE APP

Now we add this disk as a Cloudron Volume - Filesystem Mountpoint.

Filling the fields:

- Name: `lamp1`
- Mount Type: `Filesystem Mountpoint`
- Local Directory: `/mnt/lamp1`

Press "Save".

Now configure the `lamp1` apps Storage Data Directory to use the created volume `lamp1` and set "Subdirectory" to `data` and press `Move Data`.

Cloudron will now move the app data from `/home/yellowtent/appsdata/$APPID` to `/mnt/lamp1/data`. After the process is done you can confirm the data is present in `/mnt/lamp1/data` with `ls -lah /mnt/lamp1/data/`

```
ls -lah /mnt/lamp1/data/
total 40K
drwxr-xr-x 4 www-data www-data 4.0K Nov  7 10:34 .
drwxr-xr-x 4 root     root     4.0K Nov  7 10:34 ..
drwxr-xr-x 2 www-data www-data 4.0K Nov  7 10:04 apache
-rw-r--r-- 1 www-data www-data 2.3K Nov  7 10:34 credentials.txt
-rw-r--r-- 1 www-data www-data  157 Nov  7 10:04 php.ini
-rw-r--r-- 1 www-data www-data   44 Nov  7 10:04 .phpmyadminauth
-rw-r--r-- 1 www-data www-data  343 Nov  7 10:04 phpmyadmin_login.txt
-rw-r--r-- 1 www-data www-data  100 Nov  7 10:04 PHP_VERSION
drwxr-xr-x 2 www-data www-data 4.0K Nov  7 10:04 public
-rw-r--r-- 1 www-data www-data   50 Nov  7 10:04 run.sh
```

Now the `lamp1` app uses the `lamp1` Volume and is limited to 10GB disk storage.

Disk that has 50 GB storage

If you have a disk with 50 GB storage or more and you want a 10 GB storage limit for `lamp1` you will need to create partitions.

FORMATTING THE DISK AN CREATING THE FIRST PARTITION

Explanation:

- `-s` silent (non-interactive) mode
- `mklabel gpt` create a new GPT partition table
- `mkpart primary ext4 0% 10GB` make one primary partition from the start to the 10 GB mark

This will erase all data from `/dev/sdc` => Click this box if you are sure you want to do this

```
wipefs -a /dev/sdc
# this will create /dev/sdc1
parted -s /dev/sdc mklabel gpt mkpart primary ext4 0% 10GB
```

We can add more 10GB partitions with the following command:

Explanation:

`mkpart primary ext4 10GB 20GB`:

- `primary` partition type
- `ext4` intended filesystem
- `10GB` start of the partition (immediately after the first 10 GB)
- `20GB` end of the partition (10 GB size)

```
# partiton two - will be /dev/sdc2
parted -s /dev/sdc mkpart primary ext4 10GB 20GB
# partiton three - will be /dev/sdc3
parted -s /dev/sdc mkpart primary ext4 20GB 30GB
# partiton four - will be /dev/sdc4
parted -s /dev/sdc mkpart primary ext4 30GB 40GB
# partition five - will be /dev/sdc5
parted -s /dev/sdc mkpart primary ext4 40GB 50GB
```

We can inspect the result with `fdisk -l /dev/sdc`:

```
fdisk -l /dev/sdc
Disk /dev/sdc: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: Volume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: CAD6AF11-8089-4609-8246-9041B047D4D4

Device        Start      End  Sectors  Size Type
/dev/sdc1      2048 19531775 19529728  9.3G Linux filesystem
/dev/sdc2  19531776 39061503 19529728  9.3G Linux filesystem
```

```
/dev/sdc3  39061504 58593279 19531776  9.3G Linux filesystem
/dev/sdc4  58593280 78125055 19531776  9.3G Linux filesystem
/dev/sdc5  78125056 97656831 19531776  9.3G Linux filesystem
```

Now we do the same steps as above for mounting and configuring Cloudron and the app.

Creating directories for the mounts:

```
mkdir -p /mnt/lamp{1,2,3,4,5}
```

Creating the `ext4` filesystem for the partitions:

```
for i in {1..5}; do
    mkfs.ext4 /dev/sdc$i
done
```

Mounting the partitions:

```
for i in {1..5}; do
    mount /dev/sdc$i /mnt/lamp$i
done
```

Backup your `fstab` file

Editing the `fstab` can be dangerous, always create a backup first!

```
cp /etc/fstab /etc/fstab_backup_$(date +%d.%m.%Y-%H:%M:%S)
```

For more details for `fstab` please read the Arch Wiki - Fstab

Create `fstab` records for the mounts:

```
for i in {1..5}; do
    echo "/dev/sdc$i /mnt/lamp$i ext4 defaults,noatime,nofail 0 2" >> /etc/fstab
done
```

5.18.2 External XFS block storage with project quotas

Creating the filesystem

This will delete all data on `/dev/sdc` => Click this box if you are sure you want to do this

```
wipefs -a /dev/sdc
mkfs.xfs -f /dev/sdc
```

Configure Cloudron and the app

- 1 Configure a Cloudron Volume - XFS, select the device `/dev/sdc` and give the Volume a name.
- 2 Configure the app that you would like to have a quota to use the created XFS Volume and use an identifying name e.g. the APP ID from the URL `fa43ff2b-c511-4bd8-8c94-79f6910f2aee` as the `Subdirectory` name.

The data will be moved to `/mnt/volumes/$VOLUMEID/fa43ff2b-c511-4bd8-8c94-79f6910f2aee/`:

```
ls -lah /mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/fa43ff2b-c511-4bd8-8c94-79f6910f2aee/
total 24K
drwxr-xr-x 5 www-data www-data  167 Nov  7 23:23 .
drwxrwxrwx 3 root     root       50 Nov  7 23:23 ..
drwxr-xr-x 2 www-data www-data   46 Nov  7 23:12 apache
-rw-r--r-- 1 www-data www-data 2.3K Nov  7 23:23 credentials.txt
drwxr-xr-x 3 www-data www-data   17 Nov  7 12:06 mnt
-rw-r--r-- 1 www-data www-data  157 Nov  7 23:12 php.ini
-rw-r--r-- 1 www-data www-data   44 Nov  7 23:12 .phpmyadminauth
-rw-r--r-- 1 www-data www-data  343 Nov  7 23:12 phpmyadmin_login.txt
-rw-r--r-- 1 www-data www-data  100 Nov  7 23:12 PHP_VERSION
```

```
drwxr-xr-x 2 www-data www-data   40 Nov  7 23:12 public
-rw-r--r-- 1 www-data www-data   50 Nov  7 23:12 run.sh
```

- 3 Create the `/etc/projects` and `/etc/projid` files:

```
touch /etc/projects /etc/projid
```

- 4 Add the project-controlled directories to `/etc/projects`:

Info

In the following code block the number `1` is the ID, the character `:` separates and after that comes the path to the folder.

You can chose the ID freely, but be sure to also edit the following code accordingly.

```
1:/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/fa43ff2b-c511-4bd8-8c94-79f6910f2aee
```

- 5 Add project names to `/etc/projid` to map project IDs to project names:

Info

In the following code block the string `lamp1` is a project name, the character `:` separates and after that comes the project ID from the `/etc/projects` file.

The `lamp1` string can be changed freely, but be sure to also edit the following code accordingly.

```
lamp1:1
```

- 6 Initialize the project directory:

```
xfs_quota -x -c "project -s lamp1" "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

- 7 Configure quotas for projects with initialized directories:

```
xfs_quota -x -c "limit -p bsoft=10G bhard=10G lamp1" "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

- 8 Verify quotas:

```
xfs_quota -x -c 'report -h' "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

Output should look something like this:

```
Project quota on /mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7 (/dev/sdc)
                        Blocks
Project ID   Used   Soft   Hard Warn/Grace
---------- --------------------------------
#0              0      0      0  00 [------]
lamp1        108K    10G    10G  00 [------]
```

Changing the project quota

You can change projects quotas like you did configure them above.

Reduce the project quota of `lamp1` to `5M`:

```
xfs_quota -x -c "limit -p bsoft=5M bhard=5M lamp1" "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

Confirm with:

```
xfs_quota -x -c 'report -h' "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

The output should look something like this:

```
Project quota on /mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7 (/dev/sdc)
                        Blocks
Project ID   Used   Soft   Hard Warn/Grace
---------- ---------------------------------
#0              0      0      0  00 [------]
lamp1        108K     5M     5M  00 [------]
```

Confirming the quota works

Assuming we have set the quota to `5M`.

Open the Web Terminal of your app.

Create a `4M` big file in `/app/data`:

```
fallocate -l 4M /app/data/4M
```

Check the quota report with:

```
xfs_quota -x -c 'report -h' "/mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7/"
```

We can see that not much space is left:

```
Project quota on /mnt/volumes/af0b5df5b3254d6db4c3018885cb45d7 (/dev/sdc)
                        Blocks
Project ID   Used   Soft   Hard Warn/Grace
---------- ---------------------------------
#0              0      0      0  00 [------]
lamp1        4.1M     5M     5M  00 [------]
```

Try to create another `1M` file with:

```
fallocate -l 1M /app/data/1M
```

We will get an error:

```
fallocate: fallocate failed: No space left on device
```

Success

The quota works.

5.19 SSH Tunnel

5.19.1 Overview

Secure Shell Protocol or SSH is a protocol to login to servers securely and execute commands.

SSH Tunnel is a secure tunnel created between the client and the server to transfer any kind of traffic. The mechanism allows a way to connect to applications and services inside your internal network.

There are two types of tunneling:

- Local port forwarding
- Remote port forwarding

The distinction is the location of the port being forwarded - the local port or the remote port.

5.19.2 Prerequisites

It's essential to have a server which accepts SSH connections. The SSH client connects to the SSH server to establish the tunnel.

5.19.3 Local Port Forwarding

With local port forwarding, a tunnel is set up to forward traffic from the local port to the remote port.

The general syntax to set up this forwarding is:

```
ssh -L [bind_ip:]local_port:destination_server_ip:remote_port user@server_hostname
```

For example, to forward local port 8000 of local server to remote port 4000:

```
ssh -L 8000:127.0.0.1:4000 user@remote_server_ip
```

More explanation:

- In the above example, anyone can connect on port 8000. The authentication depends on the destination service at port 4000.
- You can restrict the listening of port 8000 by providing a specific bind address. For example, `127.0.0.1:8000:127.0.0.1:4000` makes SSH listen on port 8000 only on the localhost. This way only local apps can access the remote port 4000.
- `127.0.0.1` above is in the context of the remote server. This can be any IP address that the remote server can connect to. In such set ups, the remote server is just an intermediary server.
- Multiple local ports can be forwarded by chaining multiple `-L`.

Picture credit - user erik from StackOverflow

5.19.4 Remote Port Forwarding

With remote port forwarding, a tunnel is set up to forward traffic from the remote port to the local port.

The general syntax to set up this forwarding is:

```
ssh -R [bind_ip:]remote_port:local_server_ip:local_port user@server_hostname
```

For example, to forward remote port 4000 of remote server to local port 8000:

```
ssh -R 4000:127.0.0.1:8000 user@remote_server_ip
```

More explanation:

- `127.0.0.1` above is in the context of the local server. Connection is forwarded to local host and port 8000.
- Connection can also be forwarded to any server that the local server can connect to. For example, `4000:5.6.7.8:8000` will forward to port 8000 of server with IP `5.6.7.8`.
- Multiple remote ports can be forwarded by chaining multiple `-R`.

Picture credit - user erik from StackOverflow

5.19.5 Persistent tunnel

To create a persistent tunnel, create a systemd service (on Cloudron) with a file named `/etc/systemd/system/ssh-tunnel.service`.

In the example below, a MySQL server running on remote server `5.75.134.144` is exposed to the Cloudron network (172.18.0.1) on port 6612. Apps can connect to 172.18.0.1:6612 to connect to the remote MySQL server.

```
[Unit]
Description=Make database available to Cloudron apps
After=network.target

[Service]
Restart=on-failure
RestartSec=5
ExecStart=/usr/bin/ssh -NTC -o ServerAliveInterval=60 -o ExitOnForwardFailure=yes -L 172.18.0.1:6612:127.0.0.1:3306 root@5.75.134.144

[Install]
WantedBy=multi-user.target
```

To enable and restart the server:

```
systemctl daemon-reload
systemctl enable ssh-tunnel.service
systemctl start ssh-tunnel.service
```

5.19.6 Cloudron Network

IP

Cloudron runs all apps and services in an internal network (not reachable from outside the server). This network address is hardcoded to `172.18.0.0/16`.

To expose a port to apps, use the internal docker bridge IP `172.18.0.1`. Apps can connect to this internal IP address.

Databases

To set up connections, see the internal IP of the databases.

5.20 Upgrade PHP in LAMP App

5.20.1 Overview

In this guide, we will see how to upgrade the PHP version of the LAMP app.

Cloudron runs apps as read-only containers for immutability and security reasons. For this reason, it is not possible to "apt upgrade" on an installed LAMP app to update PHP. On Cloudron, you must instead install a newer version of the LAMP app and import your existing app into it.

This guide can be used to upgrade from the LAMP 7.2 to LAMP 7.3 or from LAMP 7.2/7.3 to LAMP 7.4.

5.20.2 Backup existing app

The first step is to take a backup of the existing app. Then, click on `Download Backup configuration` to download the JSON file with the latest backup information.

5.20.3 Import backup

Install the latest version of the LAMP app from the App Store. You can install this in a new subdomain like `next.domain.com`. Once installed, go to `Backups` section of the app and click on `Import`.

This will open a popup dialog. Upload the backup configuration that we downloaded in the previous step.

5.20.4 Configure database credentials

On Cloudron, each app is sandboxed and has it's own database. This means that the new LAMP app we installed will have a different database than our existing LAMP app. You will have to adjust the database credentials that your app uses by using the File Manager or the Web Terminal. The new credentials are available in `/app/data/credentials.txt`.

5.20.5 Switch the app location

Once the app is satisfactorily running, it is time to relocate the new app to production. For this, first move the old app to a new location like `old.domain.com` from the `Location` section. Then, move the new app to the desired location.

We recommend keeping the old app for a couple of days before uninstalling it. For this, you can simply keep the old app stopped (`Console` -> `Stop app`). This way, if your app is mis-behaving with a newer PHP version, you can easily switch back.

5.21 Upgrading to Ubuntu 18.04 (Bionic Beaver)

Ubuntu 18 LTS has reached its end of life

Ubuntu 18 LTS has reached its end of life in May 2023.

We encourage you to upgrade to Ubuntu 20 as soon as possible after upgrading to Ubuntu 18

5.21.1 Overview

In this guide, we will see how to upgrade an existing Ubuntu 16.04 based Cloudron to Ubuntu 18.04.

Warning

Cloudron 6.3 is the last version of Cloudron that supports Ubuntu 16.04

5.21.2 Checklist

Before upgrading, please note the following:

- ✓ Cloudron has to be on at least version 3.3. This can be verified by checking the version in the Settings view. Cloudron releases prior to 3.3 do not support Ubuntu 18.04.
- ✓ Ubuntu has to be on version 16.04. Check the output of `lsb_release -a` to confirm this.
- ✓ The upgrade takes around 1-3 hours based on various factors like network/cpu/disk etc

5.21.3 Pre-flight

Before starting the upgrade process, it's a good idea to create a server snapshot to rollback quickly. If your VPS does not have snapshotting feature, it's best to create a full Cloudron backup before attempting the upgrade (Backups -> Create Backup now).

5.21.4 Upgrading

Start the upgrade:

```
dpkg --configure -a
do-release-upgrade
```

Upgrade notes:

- Accept running an additional ssh deamon at port 1022
- For all packages (nginx, timesyncd, journald etc), select N or O : keep your currently-installed version. This is the 'default'.
- Accept removal of obsolete packages.
- Restart the server in the end

5.21.5 Post Upgrade

Finishing the upgrade:

```
systemctl stop systemd-resolved
systemctl disable systemd-resolved
systemctl restart unbound
systemctl status unbound # this should show 'active (running)'
```

If unbound is still not running, check the output of `lsof -i :53`. If it shows `named`, then:

```
systemctl stop bind9
systemctl disable bind9
systemctl restart unbound
systemctl status unbound # this should show 'active (running)'
```

5.21.6 Post Upgrade Checks

- ✓ Verify that the output includes `Ubuntu 18.04`

  Bash

  ```
  lsb_release -a
  ```

Upgrade Successful

If the above steps are all confirmed, you have successfully upgraded your Ubuntu 16 to Ubuntu 18

5.22 Upgrading to Ubuntu 20.04 (Focal Fossa)

Ubuntu 20 LTS has reached its end of life

Ubuntu 20 LTS has reached its end of life in April 2025.

We encourage you to upgrade to Ubuntu 22 as soon as possible after upgrading to Ubuntu 20

5.22.1 Overview

In this guide, we will see how to upgrade an existing Ubuntu 18.04 based Cloudron to Ubuntu 20.04. If you are still on Ubuntu 16, you must first upgrade to Ubuntu 18 before upgrading to Ubuntu 20. Follow this guide to upgrade to Ubuntu 18.

Warning

Cloudron 7.5 is the last version of Cloudron that supports Ubuntu 18.04

Cloudron 9 dropped the support for Ubuntu 20

5.22.2 Checklist

Before upgrading, please note the following:

- ✓ Cloudron has to be on at least version 6.0. This can be verified by checking the version in the Settings view. Cloudron releases prior to 6.0 do not support Ubuntu 20.04.
- ✓ Ubuntu has to be on version 18.04. Check the output of `lsb_release -a` to confirm this.
- ✓ The upgrade takes around 1-3 hours based on various factors like network/cpu/disk etc

5.22.3 Pre-flight

Before starting the upgrade process, it's a good idea to create a server snapshot to rollback quickly. If your VPS does not have snapshotting feature, it's best to create a full Cloudron backup before attempting the upgrade (Backups -> Create Backup now).

Highly recommend taking server snapshot

Taking a server snapshot will save a lot of trouble if your server is unable to start up after a failed Ubuntu upgrade. If your VPS provider does not have snapshot feature, it is critical to have backups outside of your server. This is because in corner cases Ubuntu fails to boot entirely and your data and backups will be locked up inside the server disk.

Digital Ocean

We have received numerous complaints about Digital Ocean droplet upgrades from Ubuntu 18 to Ubuntu 20 . It results in a completely hosed droplet where you cannot even SSH after upgrade. We recommend simply starting with a fresh ubuntu 22.04 instance and restoring.

5.22.4 Upgrading

Start the upgrade:

```
dpkg --configure -a
apt update
apt upgrade
do-release-upgrade
```

Upgrade notes:

- Accept running an additional ssh deamon at port 1022
- For all packages (nginx, timesyncd, journald etc), select N or O : keep your currently-installed version. This is the 'default'.
- You can pick the default for all questions asked in the upgrade process. Like LXD version can be 4.0 etc.
- Accept removal of obsolete packages.
- **IMPORTANT:** On some VPS, the upgrade process will uninstall MySQL 5.7. It is OK to uninstall it, but be sure to select the option to **not** remove MySQL data.
- On some VPS, the upgrade fails to start with an error like "After updating your package information, the essential package'ubuntu-minimal' could not be located.". This is because the mirror in `/etc/apt/sources.list` has some issue. To rectify, replace that file with the original source.list.

Once upgrade is complete, restart the server.

5.22.5 Post Upgrade

- Fixup collectd. `/etc/default/collectd` must have the following line (Add the line to end, if it doesn't exist):

```
LD_PRELOAD=/usr/lib/python3.8/config-3.8-x86_64-linux-gnu/libpython3.8.so
```

- Install MySQL 8.

```
systemctl stop box
apt remove mysql-server-5.7
# this is only needed, if it wasn't removed in the upgrade process above. if it asks, then be sure to preserve MySQL data. if it doesn't, it preserves it by default.
apt install mysql-server-8.0
/home/yellowtent/box/setup/start.sh
systemctl start box
reboot
```

5.22.6 Post Upgrade Checks

✓ Verify that the output includes `Ubuntu 20.04`

Bash

```
lsb_release -a
```

✓ Verify that the output includes `active (running)`

Bash

```
systemctl status box` will say
```

✓ Verify that the output includes `active (running)`

Bash

```
systemctl status collectd
```

✓ Verify that all the services in the Services view of Cloudron dashboard are running.

✓ **Upgrade Successful**

If the above steps are all confirmed, you have successfully upgraded your Ubuntu 18 to Ubuntu 20

5.23 Upgrading to Ubuntu 22.04 (Jammy Jellyfish)

5.23.1 Overview

In this guide, we will see how to upgrade an existing Ubuntu 20.04 based Cloudron to Ubuntu 22.04. If you are still on Ubuntu 18, you must first upgrade to Ubuntu 20 before upgrading to Ubuntu 22. Follow this guide to upgrade to Ubuntu 20.04.

⚠ **Cloudron 9 dropped the support for Ubuntu 20**

5.23.2 Checklist

Before upgrading, please note the following:

- ✓ Cloudron has to be on at least version 7.2. This can be verified by checking the version in the Settings view. Cloudron releases prior to 7.2.0 do not support Ubuntu 22.04.
- ✓ Ubuntu has to be on version 20.04. Check the output of `lsb_release -a` to confirm this.
- ✓ The upgrade takes around 1-3 hours based on various factors like network/cpu/disk etc

⚠ **Upgrade fails with Full text search feature**

There is a bug in Cloudron 7.2 where in the mail server does not start in Ubuntu 22 when the Full text search feature is turned on. Please wait till Cloudron 7.3 to upgrade.

5.23.3 Pre-flight

Before starting the upgrade process, it's a good idea to create a server snapshot to rollback quickly. If your VPS does not have snapshotting feature, it's best to create a full Cloudron backup before attempting the upgrade (Backups -> Create Backup now).

⚠ **Highly recommend taking server snapshot**

Taking a server snapshot will save a lot of trouble if your server is unable to start up after a failed Ubuntu upgrade. If your VPS provider does not have snapshot feature, it is critical to have backups outside of your server. This is because in corner cases Ubuntu fails to boot entirely and your data and backups will be locked up inside the server disk.

5.23.4 Upgrading

Start the upgrade:

```
dpkg --configure -a
apt update
apt upgrade
do-release-upgrade
```

Upgrade notes:

- Accept running an additional ssh deamon at port 1022
- Choose 'yes' to Restart services without asking.
- For all packages (mime, nginx, timesyncd, logrotate, journald etc), select N or O : keep your currently-installed version. This is the 'default'.
- `nginx` upgrade might fail towards the end. You can ignore this.

Once upgrade is complete, restart the server.

5.23.5 Post Upgrade

- Verify that ubuntu upgraded using `lsb_release -a`.
- Run `/home/yellowtent/box/scripts/init-ubuntu.sh` to install missing packages.
- Run `/home/yellowtent/box/setup/start.sh` to fix up package configuration.
- Ubuntu 22 uses cgroups v2 by default. All the docker containers have to be recreated to recognize this change in host configuration. To complete the upgrade, run `/home/yellowtent/box/scripts/recreate-containers` script. This script is only part of Cloudron 7.3. For earlier versions, it can be downloaded here.

5.23.6 Post Upgrade checks

✓ Verify that the output includes `Ubuntu 22.04`

Bash

```
lsb_release -a
```

✓ Verify that the output includes `active (running)`

Bash

```
systemctl status box
```

✓ Verify that all the services in the Services view of Cloudron dashboard are running.

Cloudron Version 8.X

If you are running a Cloudron version before version 9, also run the following command:

Bash

```
systemctl status collectd
```

The output should include `active (running)`.

Upgrade Successful

If the above steps are all confirmed, you have successfully upgraded your Ubuntu 20 to Ubuntu 22

5.24 Upgrading to Ubuntu 24.04 (Noble Numbat)

5.24.1 Overview

In this guide, we will see how to upgrade an existing Ubuntu 22.04 based Cloudron to Ubuntu 24.04. If you are still on Ubuntu 20, you must first upgrade to Ubuntu 22 before upgrading to Ubuntu 24. Follow this guide to upgrade to Ubuntu 22.04.

Please note that Ubuntu 22.04 will be supported by Canonical till 2027. Cloudron will support Ubuntu 22.04 and 24.04. It has the same feature set across both the versions.

5.24.2 Checklist

Before upgrading, please note the following:

- ✓ Cloudron has to be on at least version 8.0. This can be verified by checking the version in the Settings view. Cloudron releases prior to 8.0 do not support Ubuntu 24.04.
- ✓ Ubuntu has to be on version 22.04. Check the output of `lsb_release -a` to confirm this.
- ✓ The upgrade takes around 1-3 hours based on various factors like network/cpu/disk etc

5.24.3 Pre-flight

Before starting the upgrade process, it's a good idea to create a server snapshot to rollback quickly. If your VPS does not have snapshotting feature, it's best to create a full Cloudron backup before attempting the upgrade (Backups -> Create Backup now).

> ⚠ **Highly recommend taking server snapshot**
>
> Taking a server snapshot will save a lot of trouble if your server is unable to start up after a failed Ubuntu upgrade. If your VPS provider does not have snapshot feature, it is critical to have backups outside of your server. This is because in corner cases Ubuntu fails to boot entirely and your data and backups will be locked up inside the server disk.

5.24.4 Upgrading

Start the upgrade:

```
dpkg --configure -a
apt update
pt upgrade
do-release-upgrade
```

Upgrade notes:

- Accept running an additional ssh deamon at port 1022
- Choose 'yes' to Restart services without asking.
- For all packages (mime, nginx, timesyncd, logrotate, journald etc), select N or O : keep your currently-installed version. This is the 'default'.

Once upgrade is complete, restart the server.

5.24.5 Post Upgrade

⚠ **Version older than Cloudron 9**

With Cloudron 9, collectd was removed and this step can be skipped.

- Fixup collectd. `/etc/default/collectd` might have a `LD_PRELOAD` line from previous releases. If you make any changes, restart collectd using `systemctl restart collectd` .

```
# DELETE THIS LINE OR EQUIVALENT
LD_PRELOAD=/usr/lib/python3.10/config-3.10-x86_64-linux-gnu/libpython3.10.so
```

- Install unbound-anchor using:

Bash

```
apt install unbound-anchor
```

In some VPS, this is not installed automatically.

5.24.6 Post Upgrade checks

✓ Verify that the output includes Ubuntu 24.04 from running the following command:

Bash

```
lsb_release -a
```

✓ Verify that the output states `active (running)` from running the following command:

bash

```
systemctl status box
```

✓ Verify that the output states `active (running)` from running the following command:

Bash

```
systemctl status unbound
```

✓ Verify that all the services in the Services view of Cloudron dashboard are running.

✓ **Upgrade Successful**

If the above steps are all confirmed, you have successfully upgraded your Ubuntu 22 to Ubuntu 24

6. Community Guides

6.1 Cloudron Community Guides

6.1.1 Overview

The purpose of the Cloudron Community Guides section is to provide Cloudron Community created documentation in an effort to provide additional information on the Cloudron platform, apps, and tutorials.

⚠ **Community Guide**

This guide was contributed by the Cloudron Community, and does not imply support, warranty. You should utilize them at your own risk. You should be familiar with technical tasks, ensure you have backups, and throughly test.

6.1.2 Contributing

Please send a merge request to contribute to the docs here.

6.2 Firewall: Automatic update of blocklists (that block IP ranges from certain countries)

Background

Cloudron has a feature to block connections:

> Using the blocklist configuration, one or more IP addresses and/or networks can be blocked from connecting to Cloudron.

Based on the discussion here, @imc67 came up with a great script to update such blocklists, whereas I only changed the script a little bit to meet my needs. Below you'll find the script with all comments to guide you through.

The basic idea:

- Save code as as script
- Modify blocklist urls to your liking
- Modify `cloudron_api_endpoint` and `api_key` with your values
- run the script daily via `cron` (for information on cron and crontab, see here).
- The script pulls updated lists from https://iplists.firehol.org/ and https://www.ipdeny.com/ipblocks/ and
- uploads the blocklists to Cloudron's firewall via the API

CAVEATS

Note: Blocking certain countries is only one small part of securing Cloudron but it can have a significant impact on the amount of spam and mail server abuse.

Note: If you block certain country IP ranges this way, it also means, you might not be able to receive mails from servers / vendors located in these countries! So blocking the US is probably a bad idea; if you do regular business e.g. with China, the same applies.

Improvements are welcome as there are certainly people out there with better bash skills!

SCRIPT

do
# Splitting the URL information
IFS=',' read -r url description <<< "$url_info"
# Add comment with the URL and description
echo "# URL: $url" >> "$output_file"
echo "# Description: $description" >> "$output_file"
echo "Download IP list from $url"
# Download the IP list and add it to the file
curl -sS "$url" >> "$output_file"
done
echo "Merge completed! The merged list is stored in $output_file"
# Formatting the file for the Cloudron Blocklist API
formatted_file="formatted_$output_file"
# Add "\n" to the end of each line
awk '{printf "%s\\n",$0}' "$output_file" > "$formatted_file"
# Cloudron Blocklist API endpoint
cloudron_api_endpoint="https://my.yourdomain.com/api/v1/network/blocklist"
# API Key for authentication (replace 'your-api-key' with your API key)
api_key="#######"
Upload to Cloudron Blocklist API with wget
echo "# Upload to Cloudron Blocklist API with wget..."
# The data in the required format for the API
data="{\"blocklist\":\"$(cat "$formatted_file" | tr '\n' '\\n')\"}"
# Send the file with a POST request via wget
echo "$data"> temp_data.txt
curl -X POST "$cloudron_api_endpoint" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $api_key" \
--data-binary "@temp_data.txt" \

6.2.1

```
#!/bin/bash
# Save script in cd ~/urlfilter/, make executable, and run via cron e.g. daily, some time at night
#
# Variable for Current date and time
current_datetime=$(date +"%Y%m%d_%H%M%S")
#
# Location of script and the blocklist archive
cd ~/urlfilter/
#
# Array containing the URLs of the IP lists and their descriptions
# The first two from firehol.org are standard blocklists
# The next four are an example how to block connections from Afghanistan and Azerbaijan - first two are for IPv4 addresses, the next two for IPv6 addresses.
# In order to add countries, go to https://www.ipdeny.com/ipblocks/ to find out the relevant country names and abbreviations and add URLs accordingly on this scheme:
# IPv4:  "https://www.ipdeny.com/ipblocks/data/aggregated/[ABBREVIATIONS, small caps]-aggregated.zone,[ABBREVIATIONS, all caps] - [COUNTRY NAME]”
# IPv6:  "https://www.ipdeny.com/ipv6/ipaddresses/aggregated/[ABBREVIATIONS, small caps]-aggregated.zone,[ABBREVIATIONS, all caps] - [COUNTRY NAME]”
# So for Brunei it would be:
# Abbreviation: bn / BN
# Country Name: BRUNEI DARUSSALAM
# i.e.  "https://www.ipdeny.com/ipblocks/data/aggregated/bn-aggregated.zone,BN - Brunei Darussalam"
#
declare -a urls=(
  "https://iplists.firehol.org/files/spamhaus_drop.netset,Spamhaus - Drop"
  "https://iplists.firehol.org/files/spamhaus_edrop.netset,Spamhaus - eDrop"
  "https://www.ipdeny.com/ipblocks/data/aggregated/af-aggregated.zone,AF - Afganistan"
  "https://www.ipdeny.com/ipblocks/data/aggregated/az-aggregated.zone,AZ - Azerbaijan"
  "https://www.ipdeny.com/ipv6/ipaddresses/aggregated/af-aggregated.zone,AF - Afganistan"
  "https://www.ipdeny.com/ipv6/ipaddresses/aggregated/az-aggregated.zone,AZ - Azerbaijan"
 )

# File name with the current date and time
output_file="merged_list_${current_datetime}.txt"

# Download and merge the IP lists
for url_info in "${urls[@]}"
do
    # Splitting the URL information
    IFS=',' read -r url description <<< "$url_info"
    # Add comment with the URL and description
    echo "# URL: $url" >> "$output_file"
    echo "# Description: $description" >> "$output_file"
    echo "Download IP list from $url"
    # Download the IP list and add it to the file
    curl -sS "$url" >> "$output_file"
done
echo "Merge completed! The merged list is stored in $output_file"
# Formatting the file for the Cloudron Blocklist API
formatted_file="formatted_$output_file"

# Add "\n" to the end of each line
awk '{printf "%s\\n",$0}' "$output_file" > "$formatted_file"
# Cloudron Blocklist API endpoint
cloudron_api_endpoint="https://my.yourdomain.com/api/v1/network/blocklist"

# API Key for authentication (replace 'your-api-key' with your API key)
api_key="#######"

# Upload to Cloudron Blocklist API with wget
echo "# Upload to Cloudron Blocklist API with wget..."

# The data in the required format for the API
data="{\"blocklist\":\"$(cat "$formatted_file" | tr '\n' '\\n')\"}"

# Send the file with a POST request via wget
echo "$data"> temp_data.txt
curl -X POST "$cloudron_api_endpoint" \
     -H "Content-Type: application/json" \
     -H "Authorization: Bearer $api_key" \
     --data-binary "@temp_data.txt" \
```

```
--verbose \
-o output.txt
# Delete temporary files
rm temp_data.txt
# Compress lists and create an archive of the last 10 updates
7z a -mx9 "${current_datetime}.7z" "formatted_$output_file"
rm "formatted_$output_file"
rm "$output_file"
ls -td *.7z | grep -v '/$' | tail -n +10 | while IFS= read -r f; do rm -f "$f"; done
```

```
    --verbose \
    -o output.txt
# Delete temporary files
rm temp_data.txt
# Compress lists and create an archive of the last 10 updates
7z a -mx9 "${current_datetime}.7z" "formatted_$output_file"
rm "formatted_$output_file"
rm "$output_file"
ls -td *.7z | grep -v '/$' | tail -n +10 | while IFS= read -r f; do rm -f "$f"; done
```

Authors:

- @necrevistonnezr
- @imc67

```
do
# Splitting the URL information
IFS=',' read -r url description <<< "$url_info"
# Add comment with the URL and description
echo "# URL: $url" >> "$output_file"
echo "# Description: $description" >> "$output_file"
echo "Download IP list from $url"
# Download the IP list and add it to the file
curl -sS "$url" >> "$output_file"
done
echo "Merge completed! The merged list is stored in $output_file"
# Formatting the file for the Cloudron Blocklist API
formatted_file="formatted_$output_file"

# Add "\n" to the end of each line
awk '{printf "%s\\n",$0}' "$output_file" > "$formatted_file"
# Cloudron Blocklist API endpoint
cloudron_api_endpoint="https://my.yourdomain.com/api/v1/network/blocklist"
# API Key for authentication (replace 'your-api-key' with your API key)
api_key="#######"

# Upload to Cloudron Blocklist API with wget
echo "# Upload to Cloudron Blocklist API with wget..."

# The data in the required format for the API
data="{\"blocklist\":\"$(cat "$formatted_file" | tr '\n' '\\n')\"}"

# Send the file with a POST request via wget
echo "$data"> temp_data.txt
curl -X POST "$cloudron_api_endpoint" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $api_key" \
--data-binary "@temp_data.txt" \
--verbose \
-o output.txt
# Delete temporary files
rm temp_data.txt
# Compress lists and create an archive of the last 10 updates
7z a -mx9 "${current_datetime}.7z" "formatted_$output_file"
rm "formatted_$output_file"
rm "$output_file"
ls -td *.7z | grep -v '/$' | tail -n +10 | while IFS= read -r f; do rm -f "$f"; done
```

6.3 Beszel Agent on Cloudron

6.3.1 Background

Beszel is a lightweight server monitoring platform that includes Docker statistics, historical data, and alert functions. Bezel is available on the Cloudron App Store.

Beszel agent can be installed on Cloudron to monitor the various containers (addons and apps).

6.3.2 Risk warning

Please be aware of the following risks when installing the Beszel agent:

- Cloudron does not support installing additional software on the base system.
- The agent will have access to all the containers and data. A compromised agent (or the app) will compromise your entire system.

6.3.3 Configure Beszel

After installing Beszel from the App Store:

- `Add System`
- Use the public IP or dashboard domain name `my.example.com` of Cloudron
- Make a note of the public key. This ireferenced as BESZEL_PUBLIC_KEY in the script below.

6.3.4 Installation

SSH into the server.

Install the agent. The command creates a user for beszel and add it to the docker group as well. `/etc/systemd/system/beszel-agent.service` is the systemd service file.

```
curl -sL https://get.beszel.dev -o /tmp/install-agent.sh
chmod +x /tmp/install-agent.sh
/tmp/install-agent.sh -p 45876 -k "$BESZEL_PUBLIC_KEY"
```

6.3.5 Open Firewall

You must open port 45876 in the Cloudron Firewall and any other firewall in front of the server.

On the server, edit `/home/yellowtent/platformdata/firewall/ports.json`:

```
{
    "allowed_tcp_ports": [ 45876 ],
    "allowed_udp_ports": [ ]
}
```

Restart the firewall to apply the configuration:

```
systemctl restart cloudron-firewall
```

6.3.6 Uninstalling

`/tmp/install-agent.sh -u` will uninstall the agent

6.4 Guide: Monitoring and Alerting for Cloudron Servers

6.4.1 Introduction

As a Cloudron user, one of the challenges I've consistently faced is the lack of a built-in monitoring and alerting system. This guide aims to provide a straightforward solution for setting up monitoring on your Cloudron servers. Our goals are to make the setup process simple, enable notifications for alerts, handle multiple servers, and offer a freemium solution that starts free but scales with your needs.

⚠ **Community Guide**

This guide was contributed by the Cloudron Community, and does not imply support, warranty. You should utilize them at your own risk. You should be familiar with technical tasks, ensure you have backups, and throughly test.

- **Contributor**: AmbroiseUnly
- Discuss this guide

6.4.2 Solution: Using Netdata

Netdata is a robust monitoring tool that meets all these requirements. It is simple to install and use, supports multiple servers, and provides immediate data visualization and alerting.

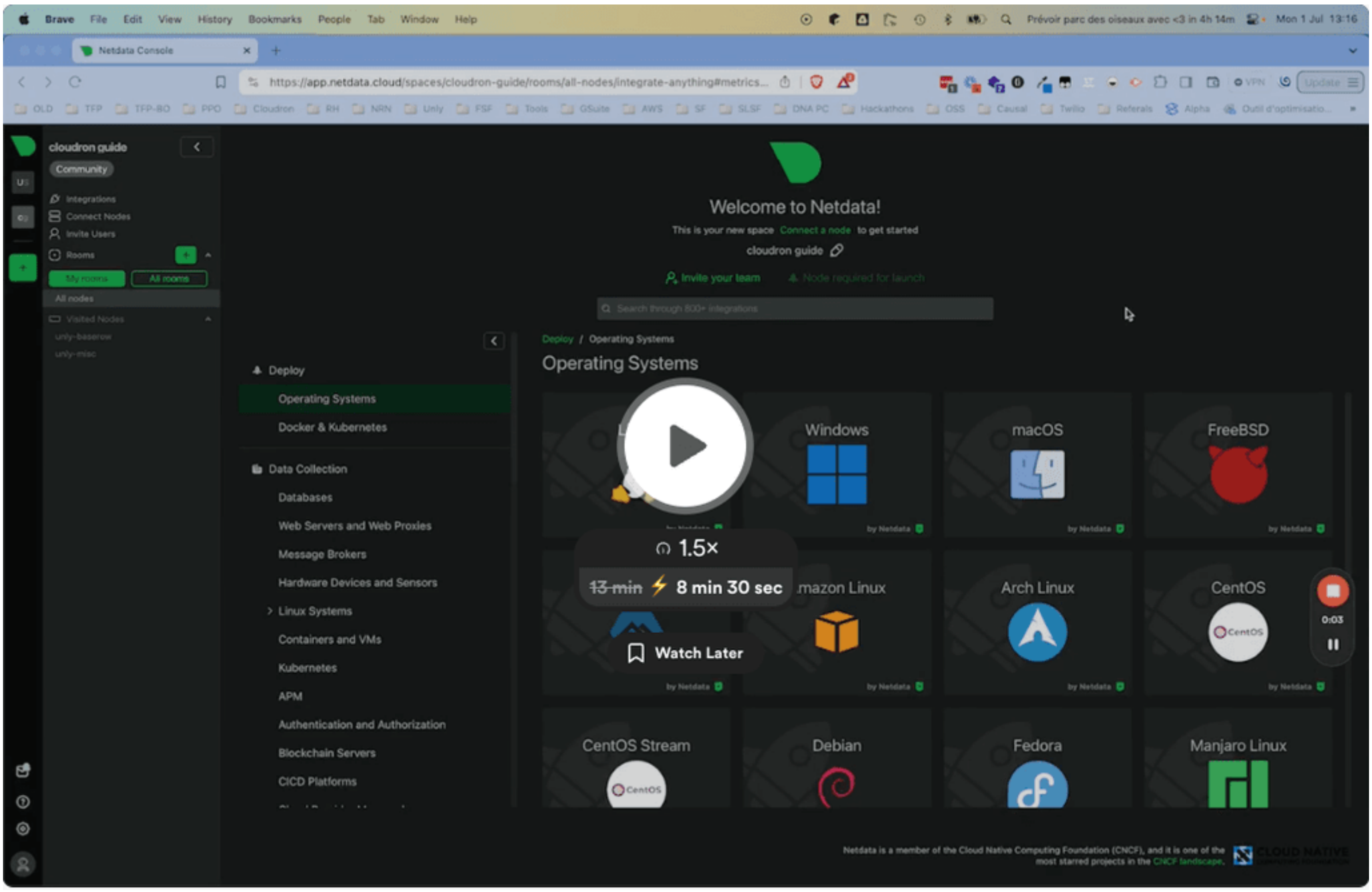
Installation Steps

Follow these steps to install Netdata on your Cloudron server. The setup takes approximately 5 minutes:

1. **Create a Netdata Account**: Sign up for a free account on Netdata.
2. **Add Your First Node**: In your Netdata dashboard, navigate to `Deploy/Operating Systems/Linux` to add your first node.
3. **Copy the Command Line**: Netdata will provide a command line script (`wget...`) for installation.
4. **Execute the Command on Your Server**: Connect to your Cloudron server via SSH and run the provided command.
5. **Follow Installation Prompts**: Netdata will prompt you for confirmation to install as a background service.
6. **Verify Installation**: Check your Netdata dashboard to ensure data is being collected and displayed.

Video Guide

For a detailed walkthrough, including a demonstration of the notification system, watch the following video guide:



6.4.3 Alternative Solutions

Grafana

Grafana is another powerful open-source monitoring tool. However, it has a more complex installation and configuration process, which might be challenging for users unfamiliar with such setups. Many users, including myself, have found it difficult to configure the notification system in Grafana.

Custom Scripts

Some community members have suggested using custom bash scripts for monitoring. While this is a viable option, it requires significant maintenance and expertise. As someone who is not a sysadmin expert, I prefer relying on specialized tools like Netdata that offer ease of use and comprehensive features out of the box.

6.4.4 Conclusion

Netdata provides an efficient and user-friendly solution for monitoring Cloudron servers. Its simplicity, combined with robust features, makes it an ideal choice for users looking to set up monitoring and alerting quickly and effectively.

6.5 Recovery form MySQL corruption

6.5.1 Background

There are two instances of MySQL on Cloudron. One instance runs on the host and is used by the platform. Another instance is the MySQL addon which runs in a container named `mysql` and is shared by apps. You can use the steps below to recover the host MySQL.

Danger

Always make sure to take a backup/snapshot of the server before proceeding

6.5.2 Soft recovery

If MySQL is running and you can connect to it using `mysql -uroot -ppassword` (sic), then maybe the data is already salvagable.

Try to create a dump of existing data:

```
mysqldump -uroot -ppassword --single-transaction --routines --triggers box > box.mysql
mysql -uroot -ppassword -e "DROP DATABASE box"
mysql -uroot -ppassword -e "CREATE DATABASE IF NOT EXISTS box"
mysql -uroot -ppassword box < box.mysql
```

If one or more steps above get "stuck", it means MySQL database is corrupt. Follow the hard recovery steps below.

6.5.3 Hard Recovery

1. Stop box code:

   bash

   ```
   systemctl stop box
   ```

2. Create a backup of the MySQL data with:

   Bash

   ```
   rsync -avrlP /var/lib/mysql /root/mysql_backup_$(date +%Y%m%d-%H%M%S)
   ```

3. Edit `/etc/mysql/my.cnf` to have the below. See the recovery docs for details.

   /etc/mysql/my.cnf

   ```
   [mysqld]
   innodb_force_recovery = 1
   ```

4. Keep increasing the above value till MySQL starts with:

   Bash

   ```
   systemctl start mysql
   ```

5. Once it starts, we have to take a dump of the database:

   Bash

   ```
   mysqldump -uroot -ppassword --skip-lock-tables -A > /root/alldb.sql
   ```

6. Now that we created the dump, stop MySQL:

   Bash

```
systemctl stop mysql
```

7. Remove the `innodb_force_recovery` in `/etc/mysql/my.cnf`
8. Recreate the existing MySQL installation:

```
mv /var/lib/mysql /var/lib/mysql.old
mkdir /var/lib/mysql
chown -R mysql:mysql /var/lib/mysql
mysqld --initialize   # This will dump the MySQL root password in /var/log/mysql/error.log
```

9. Start MySQL:

Bash

```
systemctl start mysql
```

10. Change the root password to `password` (sic) -

Bash

```
mysql -uroot -p<password from /var/log/mysql/error.log>  # there is no space between p and the password. e.g -pAS23kdI
```

SQL

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'password';
```

11. Import the database:

Bash

```
mysql -uroot -ppassword < /root/alldb.sql
```

12. Start the platform code again:

Bash

```
systemctl restart box
```

6.6 Remote backup of local Cloudron backup snapshots with restic & rclone

This is what I use for remote backups of my local Cloudron backup snapshots (done by rsync) via restic / rclone to Onedrive.

restic is a robust backup solution for incremental, encrypted, mountable(!) backups to local and remote storage. rclone, an equally robust sync software, is just a "transporter tool" that expands the available remote storages by a lot.

Maybe it can be a starting point and some inspiration for your personal needs.

- **Contributor**: necrevistonnezr

TOOLS

- rclone: https://rclone.org/docs/
- restic: https://restic.readthedocs.io/en/stable/030_preparing_a_new_repo.html#other-services-via-rclone
- ssmtp: https://wiki.archlinux.org/title/SSMTP

INSTALLATION

- Install tools above via apt
- afterwards update to latest version (repo versions are old): `sudo restic self-update && sudo rclone selfupdate`

SETUP RCLONE

- Enter an interactive setup process via `rclone config`
- in my case I use Onedrive as it has 1TB of space coming with my Office 365 subscription
- for the rest of this summary, we assume you gave it the repository name "REPOSITORY"
- details at https://rclone.org/commands/rclone_config/

SETUP RESTIC

- set up a backup repository `restic -r rclone:REPOSITORY init` (for compression support add `--repository-version 2` - recommended!)
- for a subfolder on onedrive just use `restic -r rclone:REPOSITORY:subfolder init` (for compression support add `--repository-version 2` - recommended!)
- save password that you gave the repository in file `/home/USER/resticpw`
- details at https://restic.readthedocs.io/en/latest/030_preparing_a_new_repo.html#other-services-via-rclone

SETUP SSMTP

- for receiving backup results, otherwise not needed
- See https://wiki.archlinux.org/title/SSMTP

CLOUDRON BACKUP SETTINGS

- Provider: mountpoint
- Location: `/media/CloudronBackup` (<-- obviously adjust to your settings)
- this creates a snapshot at `/media/CloudronBackup/snapshot` for the current backup
- Storage Format: rsync
- Adjust schedule and retention to your liking

BACKUP, PRUNE AND CHECK SCRIPTS

`restic-cron-backup.sh`: The actual backup

```
#!/bin/bash
d=$(date +%Y-%m-%d)
if pidof -o %PPID -x “$0”; then
echo “$(date “+%d.%m.%Y %T”) Exit, already running.”
exit 1
```

```
fi
restic -r rclone:REPOSITORY:subfolder backup /media/CloudronBackup/snapshot -p=/home/USER/resticpw
restic -r rclone:REPOSITORY:subfolder forget --keep-monthly 12 --keep-weekly 5 --keep-daily 14 -p=/home/USER/resticpw
restic -r rclone:REPOSITORY:subfolder check --read-data-subset=2% -p=/home/USER/resticpw
exit
```

First line does the backup (incremental, encrypted), second line is the backup retention, third line checks a random 2 % of all data for errors. Note that I only backup the `/snapshot` folder as all versioning is done by restic. For compression, add `--compression auto` (or `max`) to the `backup` command.

`restic-cron-prune.sh`: Pruning unused files in the backup

```
#!/bin/bash
d=$(date +%Y-%m-%d)
if pidof -o %PPID -x “$0”; then
echo “$(date “+%d.%m.%Y %T”) Exit, already running.”
exit 1
fi
restic -r rclone:REPOSITORY:subfolder prune -p=/home/USER/resticpw
exit
```

removes unused data from the repository, I run this once a week

`restic-cron-check.sh`: thorough health check of the backups

```
#!/bin/bash
d=$(date +%Y-%m-%d)
if pidof -o %PPID -x “$0”; then
echo “$(date “+%d.%m.%Y %T”) Exit, already running.”
exit 1
fi
restic -r rclone:REPOSITORY:subfolder check --read-data -p=/home/USER/resticpw
exit
```

checks all data for errors, I run this once a week

CRONTAB

```
30 2 * * * sh /home/USER/restic-cron-backup.sh | mailx -s "Restic Backup Results" server@mydomain.com
1 5 1 * * sh /home/USER/restic-cron-prune.sh | mailx -s "Restic Prune Results" server@mydomain.com
1 8 1 * * sh /home/USER/restic-cron-check.sh | mailx -s "Restic Full Check Results" server@mydomain.com
```

Backup daily at 2:30, prune and check once a week. Receive results to specified mail

MOUNT BACKUPS

Just to be complete: You can mount restic backups locally like `restic -r rclone:REPOSITORY:subfolder mount /media/resticmount/ -p=/home/USER/resticpw && cd /media/resticmount` obviously adjust `/media/resticmount/` to your settings; allows you to browse and copy from full snapshots for each backup

LIST BACKUPS

For listing all available snapshots use `restic -r rclone:REPOSITORY:subfolder snapshots -p=/home/USER/resticpw`

MIGRATE EXISTING BACKUPS TO COMPRESSED BACKUPS

For migrating existing repos to compressed repos use these two steps (will take long!)

- `restic -r rclone:REPOSITORY:subfolder migrate upgrade_repo_v2 -p=/home/USER/resticpw`
- `restic -r rclone:REPOSITORY:subfolder prune --repack-uncompressed -p=/home/USER/resticpw`

See https://restic.readthedocs.io/en/latest/045_working_with_repos.html#upgrading-the-repository-format-version for details.

6.7 Running Firefly-III Importer

6.7.1 Overview

This guide explains how to run the Firefly III Importer on a Cloudron LAMP App.

> ⚠ **Community Guide**
>
> This guide was contributed by the Cloudron Community, and does not imply support, warranty. You should utilize them at your own risk. You should be familiar with technical tasks, ensure you have backups, and throughly test.

- **Contributor**: JLX89
- Discuss this guide

6.7.2 Getting Started

- Sign up with GoCardless API or Spectre. These services are free and can import most countries (except the United States).
- The importer does't seem to have an authentication gateway to prevent "public" access
- Your Personal Access Token is embedded in the `.env` file.

Installing the Data Importer

- Install a fresh LAMP App from the Cloudron Store. It's a good idea to disable indexing via App > Security.
- Open the file manager and open the `PHP_VERSION file`. Change the value to PHP 8.2.

```
; Set the desired PHP version in this file
; Restart app for changes to take effect
PHP_VERSION=8.2
```

- Save your file and restart the LAMP App.
- Launch the terminal and complete the following commands:

```
cd /app/data/
composer create-project firefly-iii/data-importer
```

- Stay in the Terminal and next you'll want to make sure the permissions are correct by running the following commands:

```
sudo chown -R www-data:www-data /app/data/data-importer
sudo chmod -R 775 /app/data/data-importer/storage
```

- Next, you're going to need to update your Apache configuration to point from the default `/app/data/public/` folder to `/app/data/data-importer/public/`. Open `/app/data/apache/app.conf` and update the following:
- Find the lines that have the previous public route of `/app/data/public` and change them to `/app/data/data-importer/public`.
- For good measure, I would disable phpMyAdmin also in the apache.conf file: `# Include "/app/code/apache/phpmyadmin.conf"`
- Restart the LAMP App
- **Personal Access Token**: You're going to need to generate a Personal Access Token from Firefly at: https://firefly.domain.tld/profile > OAUTH > Personal Token
- Launch the Importer's file manager and go to `/app/data/data-importer` . Open the `.env` file and update the file as required:

```
FIREFLY_III_URL={https://firefly.domain.tld}
VANITY_URL={https://firefly.domain.tld}
FIREFLY_III_ACCESS_TOKEN={ACCESS-TOKEN}
AUTO_IMPORT_SECRET={GENERATED_SECRET}
TZ={America/New_York}
ENABLE_MAIL_REPORT=false
MAIL_DESTINATION={EMAIL-ADDRESS}
MAIL_FROM_ADDRESS=getenv('CLOUDRON_MAIL_FROM')
```

```
MAIL_HOST=getenv('CLOUDRON_MAIL_SMTP_SERVER')
MAIL_PORT=getenv('CLOUDRON_MAIL_SMTP_PORT')
MAIL_USERNAME=getenv('CLOUDRON_MAIL_SMTP_USERNAME')
MAIL_PASSWORD=getenv('CLOUDRON_MAIL_SMTP_PASSWORD')
MAIL_ENCRYPTION=null
REDIS_HOST=getenv('CLOUDRON_REDIS_HOST')
REDIS_PASSWORD=etenv('CLOUDRON_REDIS_PASSWORD')
REDIS_PORT=getenv('CLOUDRON_REDIS_PORT')
```

- Restart the LAMP App and you can access the interface at: https://firefly-importer.domain.tld

6.8 SMTP Relay Configuration

6.8.1 Overview

The following are examples for setting up a SMTP Relay to relay outbound mails.

⚠ **Community Guide**

This guide was contributed by the Cloudron Community, and does not imply support, warranty. You should utilize them at your own risk. You should be familiar with technical tasks, ensure you have backups, and throughly test.

- **Contributor**: JLX89
- Discuss this guide

6.8.2 ZeptoMail

Zeptomail is a Transaction email service provided by Zoho, which offers a reliable and secure service that takes care of delivering these all-important transactional emails instantly, while you can focus on building your business.

1. Navigate to your email settings in your Cloudron instance. You can do this by clicking on your username in the top left corner and clicking "email", or by going directly to: `https://my.{server-url}/#/email`
2. Next, click on the domain you'd like configure, and the `Outbound` tab, or going directly to: `https://my.{server-url}/#/email/{domain-name}/outbound`

Create API Key

1. Go to ZeptoMail Console
2. Mail Agents > {mail-agent-name}
3. Setup Info > SMTP > `Send Mail API Token #`

Cloudron Configuration

- Select `External SMTP Server`
- SMTP Host: `smtp.zeptomail.com`
- SMTP Port: 587
- Username: emailapikey
- Password: zeptomail-smtp-api-key

6.9 Find and Delete Ghost Files in Nextcloud

Info

This guide was written by @jdaviescoates in post forum.cloudron.io/post/110118

6.9.1 Motivation

If you're seeing way more disk space used than expected in your Nextcloud app (`33 GB used` but `177 GB occupied` etc.), here are some terminal commands you can run inside the app to identify and clean up ghost files, previews, app caches, and more.

You will need to open the Web Terminal of your Nextcloud app.

Or use the `cloudron exec --tty --app $APPID` command to get an interactive shell in your local terminal.

6.9.2 Check Overall Disk Usage

```
du -sh /app/data
```

See how much disk space is used by Nextcloud's data directory.

List Folder Sizes in `/app/data`

```
du -sh /app/data/* | sort -h
```

This shows usage per user/app folder so you can spot what's bloating.

6.9.3 Clean Common Space Hogs

1. Delete Preview Cache (safe to delete)

```
rm -rf /app/data/appdata_*/preview/*
```

You can regenerate them later as needed.

2. Delete Old File Versions & Deleted Files

```
occ versions:cleanup
occ trashbin:expire
occ trashbin:cleanup
```

If that doesn’t clear enough, nuke them manually:

```
rm -rf /app/data/*/files_versions/*
rm -rf /app/data/*/files_trashbin/*
```

3. Clear Unused App Data (e.g. SnappyMail)

⚠ **Warning**

Make sure your email is stored elsewhere (e.g. Cloudron mail server) before deleting.

If you’re not using the built-in SnappyMail app and use external email instead:

```
rm -rf /app/data/appdata_snappymail
```

6.9.4 Reindex Files After Cleanup

```
occ files:scan --all
```

This updates Nextcloud’s file index to reflect the actual files on disk.

6.9.5 Disable or Remove Unused Apps

If not using an app, stop it from consuming space:

```
occ app:disable snappymail
occ app:remove snappymail
```

6.9.6 Bonus: See Per-User File Usage

```
du -sh /app/data/*/files | sort -h
```

These steps helped me recover over **50 GB** of space (from 142 GB → 87 GB) just by clearing previews and unused mail cache. Hopefully they help you too!

Let me know if you spot anything else that tends to grow quietly.

6.10 Nextcloud with OpenID / Social Login - Calendar CalDav synchronization

6.10.1 TL;DR

> Create an app password in Nextcloud and use that.

6.10.2 Requirements

- Nextcloud with the social login app using Cloudron OpenID
- within that Nextcloud the calendar app
- an urge to use the calendar in your favorite Client e.g. Thunderbird, mobile phone google calendar etc

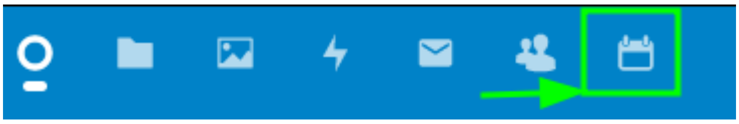
The predicament explained.

Since CalDav needs authentication (username and password) you would use that. Easy. But now, since we login with Cloudron via. OpenID the User in Nextcloud has no "password" and can also not be set.

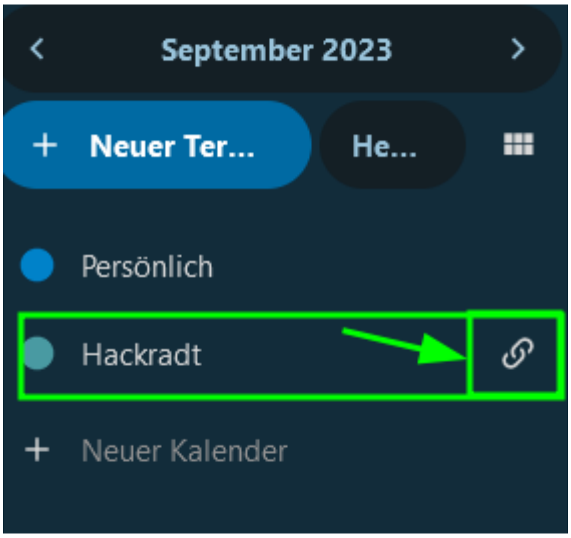
6.10.3 What to do

Get into your Nextcloud and copy your internal calendar url:

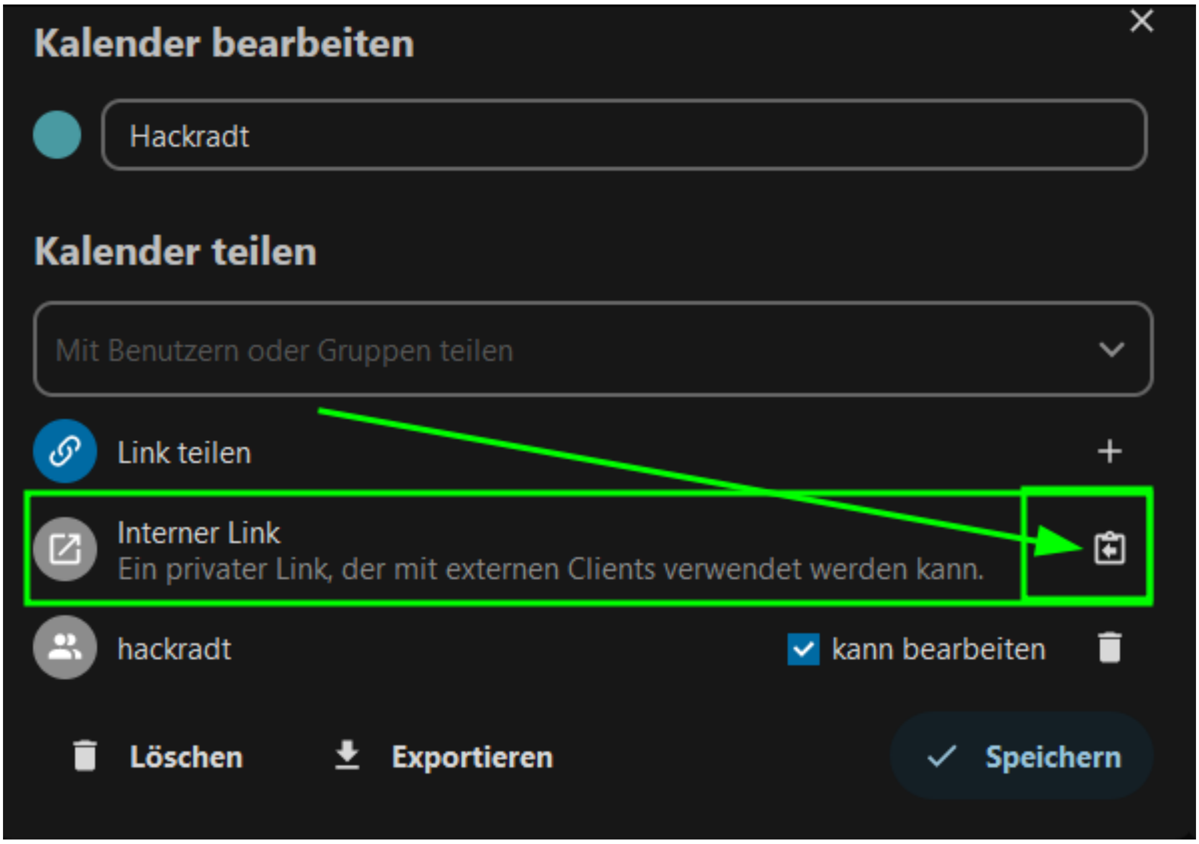
Step 1 - click your calendar



Step 2 - share your calendar / view the calendar sharing information:



Step 3 - Copy the internal link



Kalender bearbeiten

Hackradt

Kalender teilen

Mit Benutzern oder Gruppen teilen

Link teilen

Interner Link
Ein privater Link, der mit externen Clients verwendet werden kann.

hackradt ☑ kann bearbeiten

Löschen Exportieren ✓ Speichern

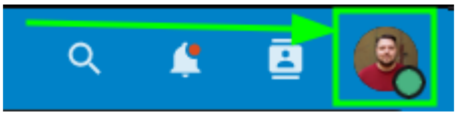
Link should look something like this:

```
https://YOUR.DOMAIN.TLD/remote.php/dav/calendars/my.DOMAIN.tld-USER.NAME/hackradt/
```

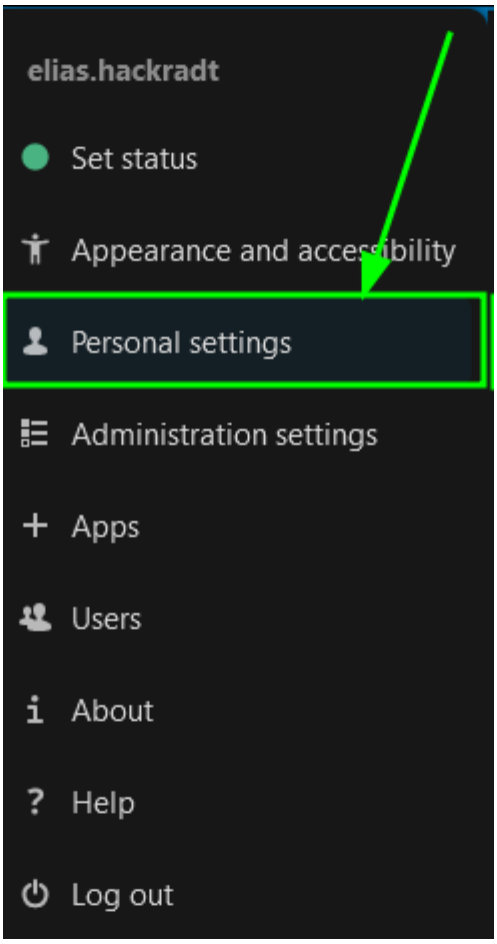
please note that `hackradt` is the name of the calendar.

Save this into a notepad, you will need it later.

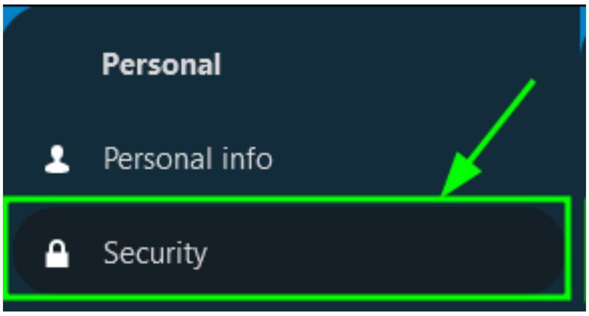
Get credentials for your User. Like above explained our User has no password, but we can set an App password! Step 1 - click your user profile icon (top right)



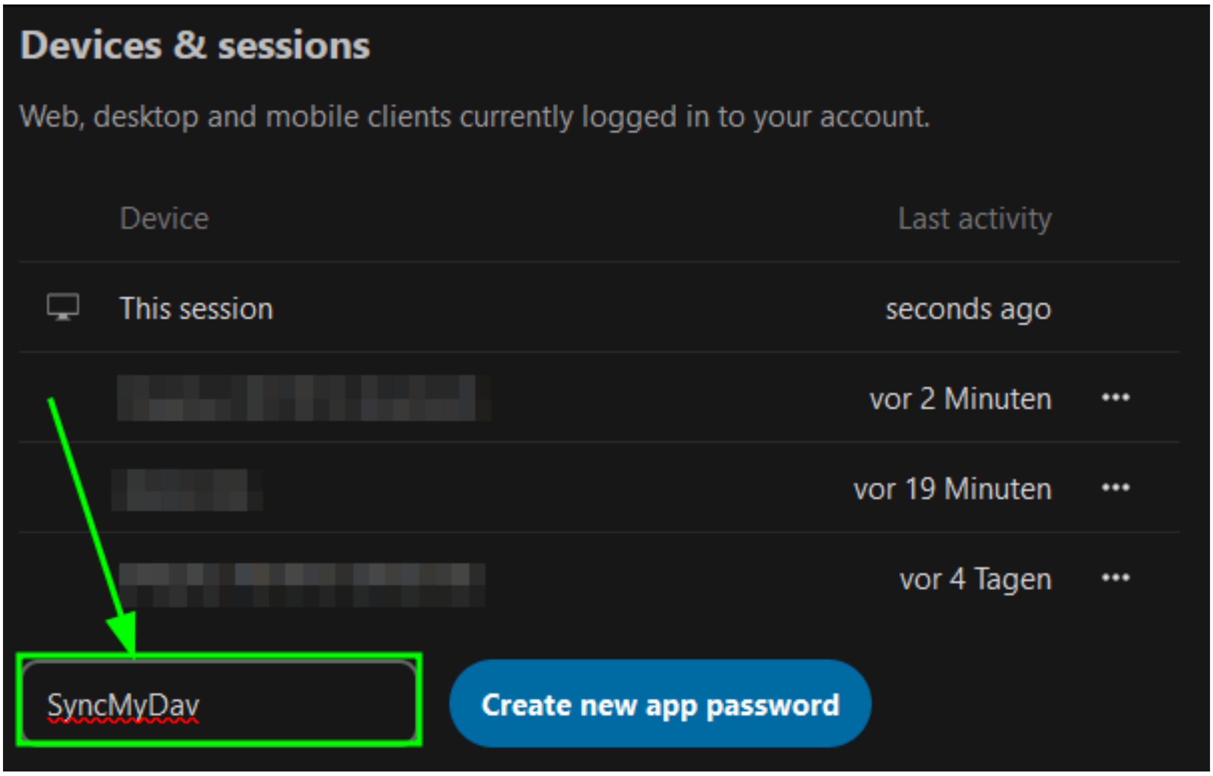
Step 2 - click "personal settings" (in the dropdown menu)



Step 3- click "security" (left side)



In the bottom of that page you got "Devices & sessions". Enter a new app name, I choose "SyncMyDav"



Devices & sessions

Web, desktop and mobile clients currently logged in to your account.

Device	Last activity
This session	seconds ago
	vor 2 Minuten
	vor 19 Minuten
	vor 4 Tagen

Click "Create new app password" Note down the Username and Password and be 100% sure to click Done - if you forget the Done part it will not work!

Now you can use these credentials and the URL to configure your Thunderbird or DAVx⁵ for your Android phone.

Finally you can use your external Nextcloud with your Cloudron login, with en extra app password for your calendar.

7. Packages

7.1 App Packages

7.1.1 Overview

Apps have their own config files that can be edited to further configure or customize the app.

7.1.2 Customization Strategy

As an example, the SFTPGo app has a `/app/data/sftpgo.json` file where users can add custom configuration.

At the same time, Cloudron has to ensure that some configurations are made so the basic functionally like database, email, authentication is always given. Cloudron sets these configurations on every app restart.

What configurations are overwritten by default can be viewed in the app startup file in the package repository.

For the SFTPGo app, the start.sh has lines like this:

```
# postgresql
yq -i -o json '.data_provider.driver="postgresql"' $config_file
yq -i -o json '.data_provider.name=strenv(CLOUDRON_POSTGRESQL_DATABASE)' $config_file
yq -i -o json '.data_provider.host=strenv(CLOUDRON_POSTGRESQL_HOST)' $config_file
yq -i -o json '.data_provider.port=strenv(CLOUDRON_POSTGRESQL_PORT)' $config_file
yq -i -o json '.data_provider.username=strenv(CLOUDRON_POSTGRESQL_USERNAME)' $config_file
yq -i -o json '.data_provider.password=strenv(CLOUDRON_POSTGRESQL_PASSWORD)' $config_file
```

This will always be executed on every app restart and PostgreSQL settings will always be overwritten.

If you ever encounter a situation where the config file resets on its own, this might be the culprit. You can always confirm that with a bit of code review of the app in question - Packages Repos.

7.2 2FAuth App

7.2.1 About

2FAuth is a web based self-hosted alternative to One Time Passcode (OTP) generators like Google Authenticator, designed for both mobile and desktop.

- Questions? Ask in the Cloudron Forum - 2FAuth
- 2FAuth Docs
- 2FAuth Discussions
- 2FAuth issue tracker

7.2.2 Custom Config

Custom config can be be placed in `/app/data/env`. See https://github.com/Bubka/2FAuth/blob/master/.env.example for various settings.

Be sure to restart the app after making any changes.

7.3 Ackee App

7.3.1 About

Ackee is a Self-hosted, Node.js based analytics tool for those who care about privacy.

- Questions? Ask in the Cloudron Forum - Ackee
- Ackee Website
- Ackee docs
- Ackee issue tracker

7.3.2 Admin Password

The admin password can be changed by editing `/app/data/env` using the File manager. Be sure to restart the app after making the change.

7.3.3 Adding a domain

- First, add the domain to be tracked, inside Ackee's Settings page. For example, if the website `www.cloudron.space` is to be tracked, add it like so:
- Next, you must configure Ackee with the correct CORS configuration for the domain you added. To do so, edit `/app/data/env` using the File manager and add the tracked website in `ACKEE_ALLOW_ORIGIN`.

```
ACKEE_ALLOW_ORIGIN="https://www.cloudron.space"
```

- Restart Ackee after making the above change.
- Embed Ackee's tracker.js script in the tracked website. Be sure to replace the ackee URL and the domain ID. You can get the snippet below by clicking on the domain entry in Ackee's setting view as well.

```
<script async src="https://ackee.cloudron.space/tracker.js" data-ackee-server="https://ackee.cloudron.space" data-ackee-domain-id="217118b9-1843-4462-82ba-2e0acd189b91"></script>
```

7.3.4 Data collection

By default, Ackee won't track personal information like device and browser info. To enable detailed tracking, pass the `data-ackee-opts` to the script tag:

```
<script async src="https://ackee.cloudron.space/tracker.js" data-ackee-server="https://ackee.cloudron.space" data-ackee-domain-id="217118b9-1843-4462-82ba-2e0acd189b91" data-ackee-opts='{ "ignoreLocalhost": true, "detailed": true }'></script>
```

You should now be able to see detailed stats:

See Ackee's docs for more information.

7.4 Actual App

7.4.1 About

Actual is a local-first personal finance tool.

- Questions? Ask in the Cloudron Forum - Actual
- Actual Website
- Actual docs
- Actual issue tracker

7.4.2 Resetting a password

If you lost your password and wish to reset it use the following command in the Web Terminal

```
node /app/code/build/src/scripts/reset-password.js
It looks like you already have a password set. Let’s reset it!
Enter a password, then press enter:
```

After entering a new password and confirming it, you can login with your new password. An app restart is not required.

7.5 AdGuard Home App

7.5.1 About

AdGuard Home is a network-wide software for blocking ads & tracking.

- Questions? Ask in the Cloudron Forum - AdGuard Home
- AdGuard Home Website
- AdGuard Home forum
- AdGuard Home issue tracker

7.5.2 Change Password

To change the AdGuard Home password, one must use the `htpasswd` tool. First, open the Web terminal and run the command below. Note that the single quote around the password below is not part of the password. It is needed for the shell to execute the command correctly when your password has special characters.

```
$ htpasswd -nbB admin 'MyNewPassword'
admin:$2y$05$zsr9LdcnDQ3TCBLuyljJHer6XS03ute6GiuA8H7ZjvKuJikud/wk2
```

Copy the password part (after the 'admin:') and put it in `/app/data/AdGuardHome.yaml` (use the File Manager. It's a good idea to quote the password field. So, it will look like this:

```
users:
- name: admin
  password: "$2y$05$zsr9LdcnDQ3TCBLuyljJHer6XS03ute6GiuA8H7ZjvKuJikud/wk2"
```

The app must be restarted for the password change to take effect.

7.5.3 Securing Installation

While the admin page is password protected, the DNS server is not. This is because DNS has no notion of authentication. Leaving your DNS server open will lead to it getting abused for conducting DDoS reflection and amplification attacks. Many VPS providers will likely send you a warning/caution email, if you run a open DNS resolver.

We strongly recommend securing your installation in the following ways:

- When available, use your VPS providers firewall functionality to restrict access to Port 53 (TCP & UDP).
- In the AdGuard Home dashboard, go to `Settings` -> `DNS settings`. Scroll to the bottom for `Access settings` and set a list of clients that can access the DNS server. You can also use ipdeny lists (IPv4 and IPv6) to set access and block lists.

7.5.4 DoH

DNS over HTTPS is enabled by default. Note that there is a Settings page that lets you enable DoH but you won't able to save that page since this is not implemented yet.

This is fine because DoH is enabled in the underlying configs and it's just an UI issue.

To use Client ID identifiers, you can add aliases to the app.

In the screenshot below, a wildcard alias is set up to make it possible to identify `somedevice` by configuring it to make DoH requests to `somedevice.adguard.smartserver.io`.

7.5.5 DoT

DNS over TLS (DoT) is supported and uses port 853 by default. DoT is required for Android's "Private DNS mode" (available since Android 9.0 Pie).

To use Client ID identifiers, you must add a wildcard subdomain alias of the form `*.adguard.domain.com`.

In the screenshot below, a wildcard alias is set up:

The phone can be configured in the `Private DNS` settings as below:

7.6 Alltube App

7.6.1 About

Alltube provides a Web GUI for youtube-dl. Despite it's name, Alltube can be used to download videos from all kinds of sources including Dailymotion, Vimeo, SoundCloud, Facebook and Instagram.

- Questions? Ask in the Cloudron Forum - AllTube
- AllTube Website
- AllTube issue tracker

7.6.2 Customization

Use the File Manager to edit `/app/data/config.yml` to add custom configuration. See the upstream file for reference.

7.7 Ampache App

7.7.1 About

Ampache is a web based audio/video streaming application and file manager. Allowing you to access your music & videos from anywhere, using almost any internet enabled device.

- Questions? Ask in the Cloudron Forum - Ampache
- Ampache Website
- Ampache issue tracker

7.7.2 Import Audio Files

Ampache works with audio catalogs. To import songs into Ampache:

- Create a directory for it through the File Manager. For example create `/app/data/catalogs/music`. You can upload new audio files to this folder now or later.
- Create a new catalog. You can do this by clicking the `Admin` icon (next to the gear icon).
- Set the catalog type to `local`
- Set the directory path `/app/data/catalogs/music` in the path text input
- Trigger a catalog update via the ampache UI to update the catalog.

7.7.3 Customization

Use the Web terminal or the File Manager to edit custom configuration under `/app/data/config/ampache.cfg.php`. Full documentation is available here.

7.7.4 Playback issues

For larger audio files, ampache often needs more memory to play them. Unlike other php based apps, increasing the memory limit in php.ini will get overwritten. The correct file for ampache is `/app/data/config/ampache.cfg.php`:

```
memory_limit = 512M
```

7.7.5 CLI

Ampache CLI can be used to run various common management tasks. You can use the CLI using the Web terminal as follows:

```
    sudo -u www-data /usr/bin/php /app/code/bin/install/catalog_update.inc
```

7.7.6 Features

Ampache has a lot of exciting features that are worth exploring further:

- Subscribe to Podcasts
- Icecast compatible Channels
- Democratic play
- Subsonic API

7.8 Apache Answer App

7.8.1 About

Apache Answer is a Q&A platform software for teams at any scales.

- Questions? Ask in the Cloudron Forum - Apache Answer
- Apache Answer Website
- Apache Issue Tracker

7.9 ArchiveBox App

7.9.1 About

ArchiveBox is a powerful, self-hosted internet archiving solution to collect, save, and view websites offline.

- Questions? Ask in the Cloudron Forum - ArchiveBox
- ArchiveBox repo
- ArchiveBox issue tracker
- ArchiveBox wiki

7.9.2 Troubleshooting

If it fails to create screenshots, pdf, html dom, you can run it manually in the Web terminal:

```
$ cd /app/data/archivebox/
$ /app/data/archivebox# gosu cloudron:cloudron archivebox update -t timestamp YOUR_TIMESTAMP
```

Where `YOUR_TIMESTAMP` is a value taken from the snapshot page.

Or you can update all snapshots by running:

```
$ cd /app/data/archivebox/
$ gosu cloudron:cloudron archivebox update
```

7.10 Astral App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.10.1 About

Astral is an open source application that allows you to organize your GitHub Stars with ease.

- Questions? Ask in the Cloudron Forum - Astral
- Astral Website
- Astral issue tracker

7.10.2 Setup

Astral requires a Github OAuth Application. You are able to create one from you Github account and then adjust the `env` file to use this application:

- Go to Github -> Settings -> OAuth Apps and create a new OAuth app.
- Enter `Astral`, `astral.yourcloudron.com`, and `https://astral.yourcloudron.com/auth/github/callback` for Application Name, Homepage URL, and Authorization callback URL, respectively.
- Generate a new Client Secret, make note of this, you'll only see it once.
- Use the app's File Manager to edit `/app/data/env` file. Update the `GITHUB_CLIENT_ID` and `GITHUB_CLIENT_SECRET` values with your Github OAuth App's Client ID and Client Secret.
- Restart the app.
- Login with Github!

7.11 Audiobookshelf App

7.11.1 About

Audiobookshelf is a self-hosted audiobook and podcast server.

- Questions? Ask in the Cloudron Forum - Audiobookshelf
- Audiobookshelf repo
- Audiobookshelf issue tracker

7.11.2 Library

The package auto creates two directories to store files:

- `/app/data/podcasts` - for podcasts
- `/app/data/audiobooks` - for audiobooks

You can upload content into above folders using the File manager.

> ✎ **Enter path manually**
>
> Library names do not appear properly when clicking `Browse for Folder` button. Instead, add paths in the input box manually like above.

7.12 AzuraCast App

7.12.1 About

AzuraCast is a free and open-source, self-hosted web radio station in a box.

- Questions? Ask in the Cloudron Forum - AzuraCast
- AzuraCast Website
- AzuraCast issue tracker

7.13 Baserow App

7.13.1 About

Baserow is an open source no-code database tool and Airtable alternative.

- Questions? Ask in the Cloudron Forum - Baserow
- Baserow website
- Baserow community
- Baserow issue tracker

7.13.2 Custom configuration

Custom env variables can be set in the file `/app/data/env.sh` using the File manager.

Be sure to restart the app after making any changes.

7.13.3 Registration

To disable public registration, `Admin` -> `Settings` -> uncheck `Allow creating new accounts` .

7.13.4 Log Retention

Baserow stores audit logs in the database and this can consume a lot of space. This can be configured using two variables in `/app/data/env.sh`:

```
export BASEROW_ROW_HISTORY_RETENTION_DAYS=365
export BASEROW_ENTERPRISE_AUDIT_LOG_RETENTION_DAYS=30 # set this even if not enterprise customer
```

Be sure to restart the app after making any changes.

7.14 Beszel App

7.14.1 About

A lightweight server monitoring platform that includes Docker statistics, historical data, and alert functions.

- Questions? Ask in the Cloudron Forum - Beszel
- Beszel Website
- Beszel issue tracker

7.14.2 Agent Installation

Installing the agent on Cloudron server is risky:

- Cloudron does not support installing additional software on the base system. Changes to base system might interfere with future updates
- The agent will have access to all the containers and data. A compromised agent (or the app) will compromise your entire system.

It is strongly discouraged to install the agent daemon on Cloudron itself.

But if you are willing to take risks, you can follow the agent installation guide.

7.15 BookStack App

7.15.1 About

BookStack is a simple, self-hosted, easy-to-use platform for organising and storing information.

- Questions? Ask in the Cloudron Forum - BookStack
- BookStack Website
- BookStack discord
- BookStack issue tracker

7.15.2 Admin

When using Cloudron user management, BookStack's built-in admin user is disabled. See the BookStack docs for more information. In addition, the app is pre-setup to give admin status to all users. You can change this by going to `Settings` -> `Registration` and adjusting the value of `Default user role after registration`. This way, the first user to login will be an admin and the roles of rest of the users can be managed inside BookStack.

7.15.3 Customization

BookStack env vars file has a list of various customizable options including default book view, default light/dark mode, timezone etc.

To change a setting, use the File Manager to edit custom configuration in `/app/data/env`. Be sure to restart the app after making any changes.

7.15.4 External registration

Bookstack does not allow external users to register when Cloudron user management (LDAP) is enabled. If you require external registration, install Bookstack with Cloudron user management disabled.

See the Bookstack docs to enable 3rd party auth like Google, GitHub, Twitter, Facebook & others.

7.15.5 Themes

Custom theme can be placed in `/app/data/themes` . You can edit the files here using the File manager.

7.16 Cal.com App

7.16.1 About

Cal.com is the event-juggling scheduler for everyone. Focus on meeting, not making meetings. Free for individuals.

- Questions? Ask in the Cloudron Forum - Cal.com
- Cal.com Website
- Cal.com Docs
- Cal.com Issue Tracker

7.16.2 Manage users

Managing users is an enterprise feature. As such, there is no way to list or delete users via the UI.

7.16.3 Disable Registration

Registration can be enabled in the admin features page.

7.16.4 Custom config

Various custom configuration can be set by changing `/app/data/env` using the File manager.

Be sure to restart the app after making any changes.

7.17 Calibre Web App

7.17.1 About

Calibre Web is a web app for browsing, reading and downloading eBooks stored in a Calibre database.

- Questions? Ask in the Cloudron Forum - Calibre Web
- Calibre Web Website
- Calibre Web issue tracker

7.17.2 Importing existing calibre database

To import an existing Calibre database, do the following:

- Stop the app
- Copy the database into `/app/data/library` using the File Manager
- Start the app

7.17.3 Library

By default, the library is located at `/app/data/library`. This can be moved to another location by copying over the existing library to the new cloudron. Note that the library files must have the owner `cloudron`.

To start with a fresh library is tricky since calibredb offers no straightforward command to create an empty library. As a workaround, run the following commands using the Web Terminal:

```
# cd /path/to/library
# chown cloudron:cloudron /path/to/library
# gosu cloudron:cloudron calibredb restore_database --really-do-it --with-library /path/to/library
Starting restoring preferences and column metadata ... 0%
Cannot restore preferences. Backup file not found. ... 100%
Restoring database succeeded
old database saved as /path/to/library/metadata_pre_restore.db
```

Switch the library in `Admin` -> `Edit Calibre Database Configuration` :

7.18 Castopod App

7.18.1 About

Castopod is a free and open-source podcast hosting solution made for podcasters who want engage and interact with their audience.

- Questions? Ask in the Cloudron Forum - Castopod
- Castopod Website
- Castopod issue tracker

7.18.2 Custom config

Custom configuration can be added in `/app/data/env` by using the File Manager. The various configurations options are documented here.

Be sure to restart the app after making any changes.

7.19 Change Detection App

7.19.1 About

Change detection is the best and simplest self-hosted free open source website change detection monitoring and notification service.

- Questions? Ask in the Cloudron Forum - Change Detection
- Change Detection Website
- Change Detection issue tracker

7.19.2 Notifications

Change Detection uses AppRise to send notifications.

An example email notification using SMTP:

```
mailtos://mail.example.com?to=<to-email>&user=<username>&pass=<password>&from=<from-email>
```

7.19.3 Custom Config

Custom environment variables can be set in `/app/data/env.sh` using the File Manager. Restart the app for any changes to take effect.

7.20 Chatwoot

Chatwoot App

7.20.1 About

Chatwoot is an open-source customer engagement suite, an alternative to Intercom, Zendesk, Salesforce Service Cloud etc.

- Questions? Ask in the Cloudron Forum - Chatwoot
- Chatwoot Website
- Chatwoot docs
- Chatwoot discussions
- Chatwoot issue tracker
- Chatwoot Discord

7.20.2 Email Inbox Configuration

To configure Chatwoot to use Cloudron mail, add an inbox channel of type email with the following IMAP and SMTP configuration:

Assuming `my.cloudron.io` is your mail server and `support@cloudron.io` is your desired support mailbox:

- Create a mailbox in your Cloudron dashboard for `support@cloudron.io`
- Set `export MAILER_INBOUND_EMAIL_DOMAIN=<mailbox-domain>` in `/app/data/env.sh` and restart the app. This is the email domain of 'forwarding inbox' inside Chatwoot. Note that you must set this value before you proceed to create an inbox inside Chatwoot.
- Add inbox channel of type email in Chatwoot
- Configure IMAP
- Configure SMTP

Forwarding Address

The forwarding address displayed by Chatwoot can be safely ignored.

7.20.3 Custom config

Custom environment variables can be set in `/app/data/env.sh` using the File manager. Be sure to reboot the app after making any changes.

7.20.4 Rails Console

To access the Chatwoot admin console, use the Web terminal and source the environment first before running rails commands.

```
# source /app/data/env.sh
# source /run/chatwoot/env.sh
# RAILS_ENV=production bundle exec rails c
```

7.20.5 Custom Helpdesk Domain

To set a custom domain for Helpdesk, add an alias in the Cloudron dashboard:

Then, set the custom domain inside Chatwoot:

7.21 Collabora Online (CODE)

7.21.1 About

Collabora is a collaborative online office suite based on LibreOffice technology.

- Questions? Ask in the Cloudron Forum - Collabora Office
- Collabora Office Website
- Collabora Office forum
- Collabora Office issue tracker

7.21.2 Setup

The Collabora app can be used to provide rich document editing functionality for files hosting inside NextCloud.

- Install NextCloud from the App Store. For this example, we assume NextCloud was installed at `nextcloud.example.com`.
- Install Collabora from the App Store
- In the Collabora setup UI, provide the domain of the NextCloud installation. If the main domain is the same for both apps, no changes have to be made.
- Enable the Nextcloud Office app in NextCloud. This app is under the `Office & text` category in the NextCloud app store. Once installed, go to NextCloud `Settings` and select the `Office` item on the left pane. Enter the domain of the collabora installation.
- Further secure the instance by setting `Allow list for WOPI requests` to `172.18.0.0/16` below.
- You should now be able to view and edit rich text documents right inside NextCloud.

7.21.3 Spell check

The empty document templates that are provided by default in Nextcloud are German documents. For this reason, it might appear that the spell-checker is flagging a lot of spelling errors.

The language of a document can be changed in the footer of the editor.

7.21.4 Adding fonts

To add custom TTF fonts, place them into `/app/data/fonts` and restart the app.

7.21.5 Custom config

Configuration can be customized by editing '/app/data/coolwsd.xml` using the File manager. Use the source code as reference for the values to customize.

Be sure to restart the app after making any changes.

7.22 Comentario App

7.22.1 About

Comentario is an open-source web comment engine, which adds discussion functionality to plain, boring web pages.

- Questions? Ask in the Cloudron Forum - Comentario
- Comentario Website
- Comentario Docs
- Comentario Source

7.22.2 Identity Providers

Comentario supports login for users and commenters via many identity providers like Facebook, Google, GitHub. These can be configured by editing `/app/data/secrets.yaml` using the File Manager.

See upstream docs for available providers and how to configure them.

Be sure to restart the app after making any changes.

7.22.3 Custom Config

Custom configuration can be added by editing `/app/data/env.sh` using the File Manager. See upstream docs for available options.

Be sure to restart the app after making any changes.

7.23 Commento++ App

7.23.1 About

Commento++ is a fast, privacy-focused commenting platform. Comment++ is a fork of Commento.

- Questions? Ask in the Cloudron Forum - Commento++
- Original Commento Website
- Commento++ Website

7.23.2 Registration

Commento++ does not integrate with Cloudron Directory. It has two types of users:

1. **Owners** can manage domains and act as moderators.
2. **Commentors** can write comments.

Owner signup is enabled by default. To disable open owner registration, edit `/app/data/commento.conf` using the File manager:

```
COMMENTO_FORBID_NEW_OWNERS=true
```

Be sure to restart the app after making any changes.

7.23.3 Sign-in with Google

Commento++ also supports OAuth sign-in with Google for users wanting to comment. To enable this feature, create a Google API secret and key pair and put those into `/app/data/commento.conf`:

```
COMMENTO_GOOGLE_KEY=<your key>
COMMENTO_GOOGLE_SECRET=<your secret>
```

7.24 Confluence App

7.24.1 About

Confluence is purpose-built for teams that need a secure and reliable way to collaborate on mission-critical projects.

- Questions? Ask in the Cloudron Forum - Confluence
- Confluence Website
- Confluence forum

7.24.2 Completing the installation

To finish the installation, do the following:

- Select Production Installation
- Add your license key.
- Select `PostgreSQL` as External Database.
- Choose `By connection string`.
- Use the File Manager and open `/app/data/credentials.txt` to get database settings.

7.24.3 Adminstration check list

- Disable `Backup Confluence` Schedule Jobs. The Cloudron backups up confluence already.
- To enable LDAP, go to `Confluence Administration`:
  - `Users & Security` -> `User directories`
  - `Add Directory` -> `Internal with LDAP authentication`
  - Use the File Manager and open `/app/data/credentials.txt` to get LDAP settings.
  - You can make Cloudron users admin once they log in to Confluence and appear in user listing via LDAP
- To configure mail:
  - `Mail Servers` -> `Add SMTP mail`
  - Use the File Manager and open `/app/data/credentials.txt` to get LDAP settings.

7.24.4 Oracle Java

OpenJDK is not supported by Confluence. For this reason, the Cloudron app uses Oracle Java.

7.25 CouchPotato App

⚠ **Discontinued**

Please note this app is not available anymore since upstream does not make releases anymore.

7.25.1 About

CouchPotato automatically find movies you want to watch.

- Questions? Ask in the Cloudron Forum - CouchPotato
- CouchPotato Website
- CouchPotato Forum

7.26 CryptPad App

7.26.1 About

CryptPad is an end-to-end encrypted and open-source collaboration suite.

- Questions? Ask in the Cloudron Forum - CryptPad
- CryptPad Website
- CryptPad docs
- CryptPad issue tracker

7.26.2 Admin

To make a user an admin:

- Go to `/settings/#account` and copy the Public Signing Key.
- Edit `/app/data/config.js` using File manager and put the key in the `adminKeys` field.
- Restart the app

7.26.3 Disable registration

Registration can be disabled in the `Administration` section.

7.26.4 Checkup

Visit `/checkup` of your installation to run through the CryptPad checklist.

7.27 Cubby App

7.27.1 About

Cubby is a filesharing app with built-in viewers. It further supports an external collabora office installation.

- Questions? Ask in the Cloudron Forum - Cubby
- Cubby Website

7.27.2 Admin

The first user is made Admin. This admin user can make other users admins.

Admins can:

- Manage users
- Configure Groups & Group Folders
- Configure Office Integration

7.27.3 Sharing

Cubby supports three types of sharing:

- Sharing with internal users
- Sharing with external users
- Group Folders

Internal Share

Files and Folders can be shared with other users using the 'Internal Share' mechanism. An Internal Share may be revoked at any time.

Note that Internal Shares become unavailable when the owner of the file/folder is deleted from Cubby.

External Share

Files and Folders can be shared with external users (users who do not have an account) via the 'External Share' mechanism. An External Share can be revoked at any time.

Note that External Shares become unavailable when the owner of the file/folder is deleted from Cubby.

Group Folders

Group Folders are directories shared by a group of users. Group Folders do not have an owner and all users have equal permissions. Contents of Group Folders are tracked separately and not part of any specific user's data. This means that contents of Group Folders are still available if a user is removed entirely.

7.28 Dawarich App

7.28.1 About

Do you remember where you've been last year? With Dawarich, you don't have to worry about forgetting. We help you remember the places you've been and the life you've lived — day by day, on a beautiful private timeline only you control.

- Questions? Ask in the Cloudron Forum - Dawarich
- Dawarich Website
- Dawarich issue tracker

7.29 Directus App

7.29.1 About

Directus is an Instant App & API for your SQL Database.

- Questions? Ask in the Cloudron Forum - Directus
- Directus Website
- Directus issue tracker

7.29.2 Login

Cloudron Directory

By default, local users (including the auto-created admin user) are allowed to login along with Cloudron Directory users. To disable this, set the following in `/app/data/env.sh` using the File manager and restart the app.

```
export AUTH_DISABLE_DEFAULT="true" # no local user login
```

By default, Cloudron users are given the built-in `Adminstrator` role. You can create another role and assign this as the default instead. To do so, set the following in `/app/data/env.sh` using the File manager and restart the app.

```
export AUTH_CLOUDRON_DEFAULT_ROLE_ID=<id of role>
```

7.29.3 Environment variables

Custom environment variables can be set in `/app/data/env.sh` using the File manager.

Be sure to restart the app after making any changes.

7.29.4 Extensions

Directus has a number of extensions.

On Cloudron, extensions are installed in the extensions folder. This is set to `/app/data/extensions`.

We have a helper script at `/app/pkg/install-extension.sh` to install extensions:

```
# /app/pkg/install-extension.sh directus-extension-wpslug-interface

added 1 package in 1s

Restart Directus to enable the extension
```

Killed

If the script above ends with `Killed`, it means that it got killed by the out of memory killer. To fix this, increase the memory limit of the app to say 2GB and re-run the script.

7.30 Discourse App

7.30.1 About

Discourse is a platform for community discussion. Free, open, simple.

- Questions? Ask in the Cloudron Forum - Discourse
- Discourse Website
- Discourse forum

7.30.2 Installing plugins

To install a plugin, open a Web terminal and run the following commands:

```
cd /app/code/plugins
git clone <plugin-repo>
chown -R cloudron:cloudron <plugin-dir>
cd /app/code
gosu cloudron:cloudron bundle exec rake plugin:install_all_gems
gosu cloudron:cloudron bundle exec rake db:migrate
gosu cloudron:cloudron bundle exec rake assets:precompile
```

Restart the application to double check everything works.

If the plugin modifies the posts in some way, you might want to rebake posts.

> ⚠ **Install with care**
>
> Use caution when installing plugins since they might break your installation. The list of official plugins can be found here.

7.30.3 Changing root account email

The email of the root account is `root@cloudron.local`. Discourse sends an activation email to the old email address to switch email. Since we don't have access to the default email account, we have to use the rails console to switch the email.

Open a Web terminal and run the following commands:

```
cd /app/code
gosu cloudron:cloudron bundle exec script/rails console
# The following lines start with "discourse(prod)>" that indicates the rails console is active
u = User.find_by_username("root")
u.email = "YOUR_NEW_EMAIL_ADDRESS"
u.save!
exit
```

7.30.4 Changing root account password

To change the password of the root account, open a Web terminal and run the following commands:

```
cd /app/code
gosu cloudron:cloudron bundle exec script/rails console
# The following lines start with "discourse(prod)>"" that indicates the rails console is active
u = User.find_by_username("root")
u.password = "YOUR_NEW_PASSWORD"
u.save!
exit
```

7.30.5 Changing domain

When changing the domain of an existing discourse installation, Cloudron automatically rebuilds the assets. However, the posts in the forum are not re-written. To rebake the posts, open a Web terminal and run the following command:

```
cd /app/code
gosu cloudron:cloudron bundle exec ruby script/discourse remap old.domain.com new.domain.com
```

7.30.6 Rebaking posts

To rebuild all posts (for example, to apply formatting provided by a newly installed plugin to old posts), open a Web terminal and run the following command:

```
cd /app/code
gosu cloudron:cloudron bundle exec rake posts:rebake
```

7.30.7 Importing Settings

Discourse allows importing settings from a file. This is useful for a faster configuration of plugins or discourse itself.

As an example, you install the plugin translator and wish to import a working configuration:

Example config file content for importing:

```
translator_enabled: true
translator_provider: LibreTranslate
translator_libretranslate_endpoint: https://com.libretranslate.cloudronapp.cloudron.dev
translator_libretranslate_api_key: e356cb19-3f06-4120-af2f-fa86df56f278
max_translations_per_minute: 30
```

This file has to be uploaded to the Discourse app with the Web Terminal.

Now you can import this file with:

```
gosu cloudron:cloudron bundle exec rake site_settings:import </tmp/settings.yml
```

This will import the given settings directly into the discourse database.

Restart the app.

After the restart, you should see the configuration applied in the admin view for all site settings.

7.30.8 Incoming email setup

- set up a mail inbox for your discourse app E.g. `forum@cloudron.dev`
- enable pop3 for that mailbox
- set the app as the owner of this mailbox
- Set both the `Mail FROM Address` and `Incoming mail` to the same mailbox.

⚠ **Use same mailbox**

Discourse does not support having different mail from and incoming mail address

In Discourse, you'll need to enable `email in` in the administration settings.

Topic creation with unique category inbox name

To enable creation of new topics via email, go to the settings page of a category and set a custom incoming email address.

If you want unique addresses for categories you need to setup mail aliases for `forum@YourMailDomain.TLD`. Say there is a category named `offtopic` then your mail alias should be something like `forum@YourMailDomain.TLD`. Or create a wildcard mail alias `*` so that you can chose freely any category name.

If you want unique addresses for categories you need to setup aliases for `forum@YourMailDomain.TLD`. Say there is a category named `offtopic` then your alias should be something like `forum@YourMailDomain.TLD`.

7.31 Docker Builder App

7.31.1 Purpose

Cloudron can be used to build and install custom apps using docker images. Building docker images locally might require many CPU resources depending on the app. Pushing docker images can also be network intensive (for e.g, if you are working from a coffee shop).

This app tries to solve the above situation by simply building and pushing docker images on the Cloudron where it is installed. This app merely acts a proxy for authenticated users to build and push docker images to a configured registry.

7.31.2 Configuring CLI

Cloudron CLI can be configured to use the build service using `cloudron build --set-build-service`. The CLI will then ask for the Cloudron credentials on which the build service is installed.

```
$ cloudron build --set-build-service
Enter build service URL: https://buildbot.example.com
Using build service https://buildbot.cloudron.ml
Building girish/nodejs-app:20191113-015207-340e7f520
Uploading source tarball...
Build Service login (https://buildbot.example.com):
Username: username
Password: *********
Login successful.
Step 1/8 : FROM cloudron/base:2.0.0@sha256:96cb00e968d7f78ff6c7f6a373ce184e0f94ad4a5298d849031201bf4a9e3bf6
 ---> 534bd0efda10
Step 2/8 : RUN mkdir -p /app/code
 ---> Running in 75e1b25ffd14
...
```

7.31.3 Private registry auth

The build service requires authentication information to be able to push images to private dockerhub repositories or a private registry. Credentials can be set by opening the Web terminal and editing `/app/data/docker.json`. Be sure to restart the app after setting the credentials.

```
{
  "docker.io": {
    "username": "username",
    "password": "mypassword"
  },
  ...
  "custom.registry.org": {
    "username": "username",
    "password": "mypassword"
  }
}
```

7.32 Docker Registry App

7.32.1 About

Docker Registry is used for storing and distributing Docker and OCI images using the OCI Distribution Specification.

- Questions? Ask in the Cloudron Forum - Docker Registry
- Docker Registry docs
- Docker Registry issue tracker

7.32.2 User Management

Cloudron Directory

When Cloudron user management is enabled, simply use the Cloudron username and password to login.

```
$ docker login registry.cloudron.space
Username: girish
Password:
Login Succeeded
```

Then, you can push images like so:

```
$ docker push registry.cloudron.space/hello-world
Using default tag: latest
The push refers to repository [registry.cloudron.space/hello-world]
e07ee1baac5f: Pushed
latest: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```

Without Cloudron Directory

When Cloudron user management is disabled, the registry is setup with no authentication. The main use case for this is to have the registry authenticate with an external provider such as GitLab instead of Cloudron. See the GitLab section below on how to set this up.

GITLAB INTEGRATION

The following steps can be used to setup GitLab container registry.

- Create a volume named `registry-shared`.
- Attach volume name `registry-shared` to both GitLab and Docker Registry apps. Be sure to uncheck the `Read Only` checkbox.
- Create folders `containers` and `certs` on the host filesystem inside the path that is assigned to the `registry-shared` volume.
- Run the following commands inside the certs folder:

```
openssl req -nodes -newkey rsa:2048 -keyout registry-auth.key -out registry-auth.csr -subj "/CN=gitlab-issuer"
openssl x509 -in registry-auth.csr -out registry-auth.crt -req -signkey registry-auth.key -days 365000
chmod 777 registry-auth.key registry-auth.crt registry-auth.csr
```

- Modify the permissions from `root` to `cloudron` inside the Docker Registry app for the created folders and files.

```
chown -R cloudron:cloudron /media/registry-shared/
```

- Modify `/app/data/config.yml` of the Docker Registry app using the File manager by altering or adding the auth part to resemble the following:

```
auth:
  token:
    realm: https://<GITLAB_HOST>/jwt/auth
    service: container_registry
    issuer: gitlab-issuer
    rootcertbundle: /media/registry-shared/certs/registry-auth.crt
```

Change the 'rootdirectory' value inside the same config file to:

```
/media/registry-shared/containers
```

Save the file and restart the app.

- Modify `/app/data/gitlab.yml` of the GitLab app by adding the following lines (some of them might already be there, so skip them):

```
production:
  <<: *base

  registry:
    enabled: true
    host: <DOCKER_REGISTRY_HOST>
    port: 443
    api_url: https://<DOCKER_REGISTRY_HOST>
    key: /media/registry-shared/certs/registry-auth.key
    path: /media/registry-shared/containers
    issuer: gitlab-issuer
```

Save the file and restart the app.

For Gitlab pipelines use the `$CI_REGISTRY_PASSWORD` and `$CI_REGISTRY_USER` for authentification with the registry.

7.32.3 Delete Tag

To delete a tag, use the delete button in the UI.

While the tag gets deleted immediately, the image blobs are not. The app is configured to run the Garbage collector every day to remove dangling blobs. To remove them immediately, open the Web Terminal and run:

```
/usr/local/bin/gosu cloudron:cloudron /app/code/registry garbage-collect /app/data/config.yml
```

7.32.4 Delete Repository

All repositories are stored in the `/app/data/storage/docker/registry/v2/repositories/` folder.

To delete a repository, delete the corresponding repository folder using the Web Terminal or the File Manager.

While the repository gets deleted immediately, the image blobs are not. The app is configured to run the Garbage collector every day to remove dangling blobs. To remove them immediately, open the Web Terminal and run:

```
/usr/local/bin/gosu cloudron:cloudron /app/code/registry garbage-collect /app/data/config.yml
```

7.32.5 Delete Images

To enable deletion of images via the UI, enable the `storage.delete.enabled` setting in `/app/data/config.yml` using the File manager. After enabling it, restart the app and you should see a button in the UI to delete images.

7.32.6 Custom UI configuration

Docker Registry UI has many customizable settings. They can be set in `/app/data/registry-ui.sh` using the File manager.

Be sure to restart the app after making any changes.

7.33 Documenso App

7.33.1 About

The Open Source DocuSign Alternative.

- Questions? Ask in the Cloudron Forum - Documenso
- Documenso Website
- Documenso issue tracker

7.33.2 Signing Certificate

For the digital signature of your documents, you need a signing certificate in .p12 format (public and private key).

You can either upload an existing one or generate a new one.

Existing Certificate

If you already have a signing certificate, upload it as `/app/data/resources/cert.p12` using the File Manager.

If the certificate has a passphrase, set it in `/app/data/env` file:

```
NEXT_PRIVATE_SIGNING_PASSPHRASE={YourStrongPassHere}
```

Restart the app for the changes to take effect.

Generate Certificate

To create a self-signed certificate, open a web terminal and run the following commands:

```
$ cd /run

# generate private key
$ openssl genrsa -out private.key 2048

# generate self-signed certificate valid for 10 years
$ openssl req -new -x509 -key private.key -out certificate.crt -days 3650

# create p12 certificate. password is optional
$ openssl pkcs12 -export -out cert.p12 -inkey private.key -in certificate.crt -legacy

# move cert into documenso
$ mv /run/cert.p12 /app/data/resources/cert.p12
$ chown cloudron:cloudron /app/data/resources/cert.p12
```

If you set an optional password, open File Manager and put passphrase into `/app/data/env` file:

```
NEXT_PRIVATE_SIGNING_PASSPHRASE={YourStrongPassHere}
```

Restart the app for changes to take effect.

7.34 Documize App

7.34.1 About

Documize Community is an open source, modern, self-hosted, enterprise-grade knowledge management solution.

- Questions? Ask in the Cloudron Forum - Documize
- Documize Website
- Contact

7.35 DocuSeal App

7.35.1 About

DocuSeal allows to create, fill, and sign digital documents.

- Questions? Ask in the Cloudron Forum - DocuSeal
- DocuSeal repo
- DocuSeal issue tracker

7.36 Dokuwiki App

7.36.1 About

DokuWiki is a simple to use and highly versatile Open Source wiki software that doesn't require a database

- Questions? Ask in the Cloudron Forum - DokuWiki
- DokuWiki Website
- DokuWiki forum
- DokuWiki issue tracker

7.36.2 User management

Cloudron Directory

When Cloudron user management is enabled, only Cloudron users can login to the wiki and edit pages.

To make a Cloudron user an admin, use the File Manager and edit `/app/data/conf/local.php` :

```
<?php

$conf['superuser'] = "userid1,userid2";
```

To make users of the Cloudron `wikiadmins` as wiki admins edit `/app/data/conf/local.php` and add:

```
<?php

$conf['superuser'] = '@wikiadmins';

// configure the oauth plugin to read in cloudron groups
$conf['plugin']['oauthgeneric'] = [
    'json-grps' => 'groups'
];
```

By default, the pages are readable by all. The wiki can be made readable only for logged in users by changing the ACL Rules for the `@ALL` group in dokuwiki's `Access Control List Management` admin page.

See upstream docs for more configuration options.

Without Cloudron SSO

When not using Cloudron authentication, first register a new user.

To make the new user an admin, use the File Manager and edit `/app/data/conf/local.php` :

```
<?php

$conf['superuser'] = "userid1,userid2";
```

DISABLING REGISTRATION

To disable registration, `Admin` -> `Configuration Manager` -> `Disable DokuWiki actions` and check `Register` .

7.36.3 Configuring

At a high level, Dokuwiki applies configuration as follows:

- `conf/foo.conf` – default value that comes with Dokuwiki. **Do not make changes to these files, they will be lost across updates.**
- `conf/foo.protected.conf` – settings applied by the Cloudron package. **Do not make changes to these files, they will be lost across updates.**
- `conf/foo.local.conf` – changed by plugin manager or Cloudron user. Changes can be freely made to these files and they will be retained across updates.

7.36.4 Plugins

Plugins can be installed using the `Extension Manager`.

Some suggested plugins:

- Blockquote – Easily add blockquoted text
- Blog – displays your posts in a familiar blog format
- Todo – add todo’s to wiki pages and assign to users if desired
- Dokubookmark – archive web pages with a simple bookmarklet to your wiki
- DW2PDF – Export wiki pages as PDFs
- EditTable – Visually edit and add tables
- Gallery – Embed image galleries in pages
- Move – move pages and namespaces while preserving all links
- Note – Insert notes that stand out from the rest of your text. Useful for documentation.
- NSPages – Automatically generate a custom list of pages in your wiki or namespace
- TemplatePageName – Changes the default template names so they can be edited from within the wiki.
- Struct – Index, display, and query structured data in your wiki pages (requires SQLite)
- Tag – add tagging functions
- VShare – Embed videos
- Wrap – probably the most useful formatting plugin – easily add columns, notes, divs, etc.
- Yearbox – Auto generate a table with links for a journal or diary, very customizable.

Some plugins require a higher memory limit to successfully install. For this, increase the memory limit via the Cloudron dashboard and then add the following line to `php.ini` using the File manager and restart the app:

```
memory_limit = 512M
```

7.36.5 Themes

Themes (templates) can be installed using the `Extension Manager`. Note that the theme has to be activated after installation. This can be done using the `Configuration Manager`'s `template` drop down setting.

7.36.6 Debugging

To debug LDAP, set the following

```
$conf['plugin']['authldap']['debug']      = 0;
```

7.37 Dolibarr App

7.37.1 About

Dolibarr is an open source ERP & CRM for business.

- Questions? Ask in the Cloudron Forum - Dolibarr
- Dolibarr Website
- Dolibarr forum
- Dolibarr issue tracker

7.37.2 Sync Users

Users are synced from Cloudron to Dolibarr every hour . You can also sync manually by running `/app/pkg/sync-users.sh` manually using the Web Terminal.

7.38 Easy!Appointments App

7.38.1 About

Easy!Appointments is a web appointment scheduler.

- Questions? Ask in the Cloudron Forum - EasyAppointments
- EasyAppointments Website
- EasyAppointments issue tracker

7.39 Element App

7.39.1 Setup

Element is a front end that lets you connect to Matrix home servers. See the Synapse package for installing a home server.

This app is pre-configured to use the matrix installation at `matrix.yourdomain.com`. For example, if you installed Element at `element.example.com`, the application is pre-configured to use `matrix.example.com`.

You can change the homeserver location, by using a Web Terminal and editing `/app/data/config.json`.

7.39.2 Custom configuration

You can add custom Element configuration using the Web terminal:

- Add any custom configuration in `/app/data/config.json`.
- Restart the app

See config.json for reference.

7.39.3 Custom files

Custom files can be located under `/app/data/custom`. They can then be used in some of the configurations (like background) as follows:

```
    "branding": {
        "welcomeBackgroundUrl": "/custom/background.png",
        "authHeaderLogoUrl": "/custom/header.png"
    }
```

7.40 Emby App

7.40.1 About

Bringing all of your home videos, music, and photos together into one place has never been easier. Your personal Emby Server automatically converts and streams your media on-the-fly to play on any device.

- Questions? Ask in the Cloudron Forum - Emby
- Emby Website
- Emby support
- Emby forum

7.40.2 Mobile Apps

Emby Apps for various devices can be downloaded here.

For iOS and Android, you should be able to connect simply using the `https://emby.mydomain.com`. in the custom server url field. No further ports or custom api urls need to be specified.

> **Use port 443**
>
> When connecting with the mobile apps use port 443 and not port 8920.

7.40.3 Hardware Transcoding

> **Cloudron 5.6 required**
>
> Cloudron 5.6 is the first release that supports hardware transcoding.

Emby supports 3 types of hardware acceleration on Linux - Nvidia NVDEC, VA API and Intel QuickSync. Cloudron does not support Nvidia at the time of this writing.

There are various steps to check if your hardware supports transcoding and if Emby is able to take advantage of it.

- Check the output of `vainfo` on your server. You might have to run `apt-get install vainfo libva2 i965-va-driver` if that command is not available on your server. The output should look like below. `VAEntrypointVLD` means that your card is capable to decode this format, `VAEntrypointEncSlice` means that you can encode to this format.

```
$ vainfo
error: can't connect to X server!
libva info: VA-API version 1.1.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_1_1
libva info: va_openDriver() returns 0
vainfo: VA-API version: 1.1 (libva 2.1.0)
vainfo: Driver version: Intel i965 driver for Intel(R) CherryView - 2.1.0
vainfo: Supported profile and entrypoints
      VAProfileMPEG2Simple            : VAEntrypointVLD
      VAProfileMPEG2Simple            : VAEntrypointEncSlice
      VAProfileMPEG2Main              : VAEntrypointVLD
      VAProfileMPEG2Main              : VAEntrypointEncSlice
      VAProfileH264ConstrainedBaseline: VAEntrypointVLD
      VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice
      VAProfileH264Main               : VAEntrypointVLD
      VAProfileH264Main               : VAEntrypointEncSlice
      VAProfileH264High               : VAEntrypointVLD
      VAProfileH264High               : VAEntrypointEncSlice
      VAProfileH264MultiviewHigh      : VAEntrypointVLD
      VAProfileH264MultiviewHigh      : VAEntrypointEncSlice
      VAProfileH264StereoHigh         : VAEntrypointVLD
      VAProfileH264StereoHigh         : VAEntrypointEncSlice
      VAProfileVC1Simple              : VAEntrypointVLD
      VAProfileVC1Main                : VAEntrypointVLD
      VAProfileVC1Advanced            : VAEntrypointVLD
      VAProfileNone                   : VAEntrypointVideoProc
      VAProfileJPEGBaseline           : VAEntrypointVLD
      VAProfileJPEGBaseline           : VAEntrypointEncPicture
      VAProfileVP8Version0_3          : VAEntrypointVLD
      VAProfileVP8Version0_3          : VAEntrypointEncSlice
      VAProfileHEVCMain               : VAEntrypointVLD
```

- Next step is to check Emby logs. In `Manage Emby Server` -> `Advanced` -> `Logs`, look for a file named `hardware_detection-<something>.txt`. If you cannot find this file, simply restart Emby and it will appear on startup. The log file output will indicate that it detected the DRI device and can access it and what it can transcode.
- Now that Emby can access the DRI devices, play a video. For every video played, Emby will generate a log file of the `ffmpeg-transcode-<something>.txt`. The log file outputs the Porcessing plan. Finally, when the video is playing, open a new browser tab and see the `Active Devices` in the Emby dashboard. This will show that the video is indeed transcoding.

7.41 EspoCRM App

7.41.1 About

EspoCRM is a web application that allows users to see, enter and evaluate all your company relationships regardless of the type. People, companies, projects or opportunities — all in an easy and intuitive interface.

- Questions? Ask in the Cloudron Forum - Espo CRM
- Espo CRM Website
- Espo CRM forum
- Espo CRM issue tracker

7.41.2 Admin access

EspoCRM is automatically setup with an admin account.

```
    username: admin
    password: changeme
```

Be sure to change the admin credentials immediately after installation.

To make an existing Cloudron user an admin, set the admin flag under `Teams and Access Control`. Currently, this requires that the Cloudron user log into EspoCRM first.

7.41.3 Portals

EspoCRM Portal is a functionality that provides the access specific CRM data and functions for your customers and partners. Portals can either be sub paths or standalone domains.

Cloudron supports standalone domains for EspoCRM portals using App Location Aliases. To setup a portal domain:

- Create the EspoCRM portal
- Specify a custom URL and custom ID for the portal. For example, `https://acme.cloudron.club` in the screenshot below:
- Add the domain in the Cloudron dashboard.
- Add portal configuration rewrite rules in `/app/data/apache/portals.conf` using the File Manager:

```
    RewriteCond %{HTTP_HOST} ^acme\.cloudron\.club$
    RewriteRule ^client - [L]

    RewriteCond %{HTTP_HOST} ^acme\.cloudron.club$
    RewriteCond %{REQUEST_URI} !^/portal/acme/.*$
    RewriteRule ^(.*)$ /portal/acme/$1 [L]
```

- Restart the app for the apache configuration changes to take effect.
- You can now visit `https://acme.cloudron.club`.

7.41.4 Resetting passwords

Especially for the `admin` user if the email was not changed to a real mailbox, a new password can be set using the following command in the Web terminal:

```
/usr/local/bin/gosu www-data:www-data php command.php set-password admin
```

7.41.5 Cron Jobs

EspoCRM cron is already pre-configured on Cloudron and runs every minute. However, the scheduled jobs to run are not pre-configured and this must be configured by the administrator depending on what feature of EspoCRM used.

Below is a list of jobs available out-of-the-box. You have to add them in `Administration` > `Scheduled Jobs`.

- CheckEmailAccounts – checks personal email accounts
- CheckInboundEmails – checks group email accounts
- Cleanup
- ProcessWebhookQueue
- SendEmailNotifications
- ControlKnowledgeBaseArticleStatus
- ProcessMassEmail
- SendEmailReminders
- SubmitPopupReminders

7.41.6 Uninstalling extension

To uninstall an extension:

- Enable Recovery Mode.
- Open the Web Terminal.
- `php command.php extension -l` to list all extensions
- To uninstall a specific extension: `php command.php extension -u --name="Google Integration"`

See EspoCRM documentation for more information.

7.42 Etherpad App

7.42.1 Installing plugins

To install plugins or change the configuration, visit the admin interface at `/admin`.

A complete list of available plugins is available here.

7.42.2 Admin user

Any user can be made admin.

- Adjust the `users` section in `/app/data/settings.json` via the filemanager, remove any `password` field if exist and make sure the usernames match the username on Cloudron:

```
{
  "users": {
    "username1": {
      "is_admin": true
    },
    "username2": {
      "is_admin": true
    }
  }
}
```

- Restart the app
- Relogin with that user in Etherpad. Only login via open or createing a new pad works. Login via `admin/login` does not work, as that view has no OpenID integration!
- Access `/admin`

7.42.3 Custom settings

Use a Web terminal and add any custom settings to `/app/data/settings.json`.

> The app has to be restarted after editing `/app/data/settings.json`

Make Documents Public

By default the app will always require login with a valid user. To allow any visitor to create and edit documents, add the following to `/app/data/settings.json`:

```
    "requireAuthentication": false,
```

7.42.4 Customizing CSS

This feature was removed, but there are a few skin variants available, like the dark mode mentioned below. See the etherpad docs for more information.

Dark mode

The app ships with the **colibris** theme/skin. This skin supports a dark mode through the skinVariants. To enable that, edit `/app/data/settings.json`:

```
    "skinVariants": "super-dark-toolbar super-dark-editor dark-background",
```

7.42.5 API Access

The Etherpad API can be accessed by obtaining the APIKEY. For this, open a Web terminal and view the contents of the file `/app/data/APIKEY.txt`.

Example usage:

```
curl https://etherpad.domain/api/1.2.7/listAllPads?apikey=c5513793f24a6fbba161e4497b26c734ff5b2701fad0f1211097ccb405ea65c7
```

7.42.6 Troubleshooting

If the app does not start, especially after an update, usually this is related to incompatible plugins. To fix this situation, put the app in recovery mode and open a Web terminal. Get a list of installed plugins:

```
npm ls 2> /dev/null | grep ep_
```

The two plugins `ep_cloudron` and `ep_etherpad-lite` are required, any other plugin might cause the issue. Uninstall other plugins one by one with:

```
npm rm <pluginname>
```

Then see if the app can start up again by running `/app/pkg/start.sh`. If the app starts up and is accessible normally, disable recovery mode again, otherwise try the next one.

Plugins which did not cause the problem can be reinstalled again with:

```
npm i <pluginname>
```

7.43 evcc App

7.43.1 About

evcc is an energy management system with a focus on electromobility. The software controls your EV charger or smart plug. It communicates with your vehicle, inverter or home storage to make intelligent charging decisions.

- Questions? Ask in the Cloudron Forum - evcc
- evcc repo
- evcc issue tracker

7.44 Fider App

7.44.1 About

Fider is a feedback portal for feature requests and suggestions.

- Questions? Ask in the Cloudron Forum - Fider
- Fider repo
- Fider issue tracker

7.44.2 Customization

Custom configuration can be added in `/app/data/env` using the File manager.

Restart the app after making any changes.

7.45 File Pizza App

7.45.1 About

File Pizza imlements peer-to-peer file transfers in your browser.

- Questions? Ask in the Cloudron Forum - File Pizza
- File Pizza Website
- Upstream File Pizza issue tracker

7.46 FindMyDevice App

7.46.1 About

FindMyDevice (FMD) Server finds your device and control it remotely. It aims to be a secure open source alternative to Google's Find My Device.

The FMD app for android can be found here (source].

- Questions? Ask in the Cloudron Forum - FindMyDevice
- FindMyDevice repo
- FindMyDevice issue tracker

7.46.2 Custom Config

Custom config can be set in `/app/data/config.yml` using the File manager.

Be sure to restart the app after making any changes.

7.46.3 UnifiedPush

Communication between server and app relies on UnifiedPush. The push can be either be selfhosted using ntfy.sh or use the service at https://ntfy.sh . If you decide to selfhost ntfy server, you must enable support for UnifiedPush. See https://docs.ntfy.sh/config/#example-unifiedpush

7.47 Firefly III App

7.47.1 About

Firefly III is a free and open source personal finance manager.

- Questions? Ask in the Cloudron Forum - Firefly III
- Firefly III Website
- Firefly III issue tracker

7.47.2 Admin

Cloudron user can login to Firefly III using their username and password. This app has no separate 'admin' account. The first user to login is made an admin (site owner). Site owners can edit and remove other users.

The Firefly III UI does not have a way to grant site owner permissions to another user.

Firefly III uses the `SITE_OWNER` environment variable in some error messages. For this reason, it is recommended to change this value in `/app/data/env` using the File manager.

7.47.3 Sharing accounts

Currently, sharing account between users is not implemented. See the upstream bugtracker for more information - #372 and #2531.

7.47.4 Community Guides

- Running Firefly-III Importer

7.48 Formbricks App

7.48.1 About

A privacy-first Experience Management suite built on the largest open source survey platform worldwide. Gather feedback on websites, apps and everywhere else to understand what your customers need.

- Questions? Ask in the Cloudron Forum - Formbricks
- Formbricks docs
- Formbricks issue tracker

7.48.2 Default user role

You can update a default user role set for users on first login by setting `DEFAULT_ORGANIZATION_ROLE` in `/app/data/env`.

7.49 FreeScout App

7.49.1 About

FreeScout is the super lightweight free open source help desk and shared inbox.

- Questions? Ask in the Cloudron Forum - FreeScout
- FreeScout Website
- FreeScout issue tracker

7.49.2 Mailbox Setup

Mailboxes do not need to be hosted on Cloudron itself. The app acts as a regular email client and thus can be setup for any IMAP mailbox. For sending emails of a specific mailbox, the STMP method has to be selected as `php mail()` or `sendmail` wont work on Cloudron.

Cloudron mailbox

To configure Freescout with a Cloudron mailbox, use the following connection settings.

For Sending Emails:

Set encryption to None if FreeScout and Cloudron Mail are on same server

For technical reasons, the mail server does not offer encryption for apps hosted on the same server. For this reason, set the Encryption in the screenhot below to None instead of TLS. This is safe since all communication is internal to the server.

For Fetching Emails, use the configuration below. Unlike sending emails, the encryption setting is always `SSL`.

7.49.3 Modules

FreeScout supports a wide variety of modules to extend the app. Most of them can simply be installed through the FreeScout user interface, however some require additional steps to migrate the database before being used.

To ensure all necessary steps are run after a module installation, simply restart the app through the Cloudron dashboard after activating the module.

7.49.4 Reset admin password

There is no CLI command to reset the admin password. Instead, just create a new temporary admin and then reset the password of the original admin.

Using a Web terminal run the following command:

```
sudo -E -u www-data php artisan freescout:create-user --role admin --firstName Second --lastName Admin --email admin2@cloudron.local --password changeme123
--no-interaction
```

7.50 FreshRSS

7.50.1 About

FreshRSS is a free, self-hostable aggregator.

- Questions? Ask in the Cloudron Forum - FreshRSS
- FreshRSS Website
- FreshRSS issue tracker

7.50.2 Installing Extensions

To install extensions, simply extract them to `/app/data/extensions` and restart the app.

The File manager can be used to upload and extract extensions.

7.50.3 Apps

To enable mobile access,

- In `Administration` -> `Authentication`, enable the option “Allow API access (required for mobile apps)”.
- Under the section `Profile`, fill-in the field `API password (e.g., for mobile apps)`.
- Note that every user must define an API password.

FeedMe

The FeedMe Android app has support for FreshRSS. When using this app, remember to use the hostname as `https://freshrss.example.com/api/greader.php`

7.51 GeoIP App

7.51.1 About

A simple Geolocation service based on maxmind.

- Questions? Ask in the Cloudron Forum - GeoIP Service
- GeoIP Website

7.51.2 Usage

When making any request to either `/json` or `/jsonp?callback=functionName`, the service will resolve the IP to a location and return as much information as possible about the country and city.

Example:

```
$ curl https://geoip.example.com/json
{"city":{"geoname_id":5344157,"names":{"en":"Dublin","ru":"Дублин","zh-CN":"   "}},"continent":{"code":"NA","geoname_id":6255149,"names":
{"de":"Nordamerika","en":"North America","es":"Norteamérica","fr":"Amérique du Nord","ja":"     ","pt-BR":"América do Norte","ru":"Северная Америка","zh-
CN":"   "}},"country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United States","es":"Estados Unidos","fr":"États-Unis","ja":"
 ","pt-BR":"Estados Unidos","ru":"США","zh-CN":"  "}},"location":{"accuracy_radius":50,"latitude":37.7201,"longitude":-121.919,"metro_code":
807,"time_zone":"America/Los_Angeles"},"postal":{"code":"94568"},"registered_country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United
States","es":"Estados Unidos","fr":"États-Unis","ja":"       ","pt-BR":"Estados Unidos","ru":"США","zh-CN":"  "}},"subdivisions":[{"geoname_id":
5332921,"iso_code":"CA","names":{"de":"Kalifornien","en":"California","es":"California","fr":"Californie","ja":"        ","pt-
BR":"Califórnia","ru":"Калифорния","zh-CN":"      "}}]}
```

Additionally both routes can take a query param `ip` which will override the source ip.

```
$ curl -4 https://geoip.example.com/json?ip=8.8.8.8
{"continent":{"code":"NA","geoname_id":6255149,"names":{"de":"Nordamerika","en":"North America","es":"Norteamérica","fr":"Amérique du Nord","ja":"     ","pt-
BR":"América do Norte","ru":"Северная Америка","zh-CN":"   "}},"country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United
States","es":"Estados Unidos","fr":"États Unis","ja":"    ","pt-BR":"EUA","ru":"США","zh-CN":"  "}},"location":{"accuracy_radius":1000,"latitude":
37.751,"longitude":-97.822,"time_zone":"America/Chicago"},"registered_country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United
States","es":"Estados Unidos","fr":"États Unis","ja":"    ","pt-BR":"EUA","ru":"США","zh-CN":"  "}}}
```

7.51.3 Access Token

An access token can be set in `/app/data/apitoken.txt`. If set, you have to pass the `accessToken` query parameter.

```
$ curl https://geoip.example.com/json?accessToken=thetoken
{"city":{"geoname_id":5344157,"names":{"en":"Dublin","ru":"Дублин","zh-CN":"   "}},"continent":{"code":"NA","geoname_id":6255149,"names":
{"de":"Nordamerika","en":"North America","es":"Norteamérica","fr":"Amérique du Nord","ja":"     ","pt-BR":"América do Norte","ru":"Северная Америка","zh-
CN":"   "}},"country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United States","es":"Estados Unidos","fr":"États-Unis","ja":"
 ","pt-BR":"Estados Unidos","ru":"США","zh-CN":"  "}},"location":{"accuracy_radius":50,"latitude":37.7201,"longitude":-121.919,"metro_code":
807,"time_zone":"America/Los_Angeles"},"postal":{"code":"94568"},"registered_country":{"geoname_id":6252001,"iso_code":"US","names":{"de":"USA","en":"United
States","es":"Estados Unidos","fr":"États-Unis","ja":"       ","pt-BR":"Estados Unidos","ru":"США","zh-CN":"  "}},"subdivisions":[{"geoname_id":
5332921,"iso_code":"CA","names":{"de":"Kalifornien","en":"California","es":"California","fr":"Californie","ja":"        ","pt-
BR":"Califórnia","ru":"Калифорния","zh-CN":"      "}}]}
```

7.52 Ghost App

7.52.1 About

Ghost makes it simple to publish content online, grow an audience with email newsletters, and make money from premium memberships.

- Questions? Ask in the Cloudron Forum - Ghost
- Ghost Website
- Ghost forum
- Ghost issue tracker

7.52.2 Structured data

Ghost outputs basic meta tags to allow rich snippets of your content to be recognised by popular social networks. Currently there are 3 supported rich data protocols which are output in `{{ghost_head}}`:

- Schema.org - http://schema.org/docs/documents.html
- Open Graph - http://ogp.me/
- Twitter cards - https://dev.twitter.com/cards/overview

The Cloudron app enables output of structured data by default.

7.52.3 Gravatar

For privacy reasons, Gravatar functionality is disabled by default. You can re-enable this by editing the `useGravatar` field in `/app/data/config.production.json` using the File Manager. Be sure to restart the app after editing the config file.

7.52.4 Importing

You can import content from another Ghost installation from Settings -> Labs -> Import content.

If the JSON file is large, the import might fail. To fix this:

- Give the app more memory (say 2GB).
- Increase the SQL service memory limit to at least 2GB each. This can be reverted back to default after import.
- Next, use the File Manager to edit `/app/data/env.sh` to adjust the NodeJS `max-old-space-size` limit. Set it to 2GB using a line like this `export NODE_OPTIONS="--max-old-space-size=2048"`
- Restart Ghost and try the import again.

7.52.5 Subscribe Button

The `Subscribe` button lets uses subscribe to your blog. If you don't use this feature, you can disable the functionality as follows:

- `Settings` -> `Membership` -> change `Subscription access` to `Nobody`. This will remove the floating Subscribe button at the bottom of the page. You can also remove the button from the Portal using `Settings` -> `Membership` -> `Customize Portal`.
- To remove the Subscribe button on the nav bar, you have to adjust the theme or inject custom CSS as suggested here.

7.52.6 Email

Ghost sends two types of emails:

- Transactional emails - used for password reset, member sign up confirmation, invites etc.
- Bulk emails - for newsletters. This requires Mailgun.

Transactional emails

Like all apps, Ghost sends out emails via Cloudron's mail server. You can setup a relay in Cloudron's Email view to configure how these mails are ultimately sent out.

There are three email addresses in Ghost:

- Password Reset & Invitation - This address can be changed using the Mail FROM address set in the Email section of the app.
- Support email address - defaults to `noreply@app.example.com`.
- Newsletter email address - defaults to `noreply@app.example.com`.

When trying to change the Support and Newsletter email addresses, to say `newaddress@example.com`, Ghost will send a confirmation mail from `noreply@example.com` to `newaddress@example.com`. The `noreply@` address is hardcoded and cannot be changed.

Out of the box, Cloudron will block this confirmation mail because it does not allow apps to send emails with arbitrary FROM addresses. To remedy this:

- In Cloudron dashboard, go to Email -> Select Domain -> Settings -> Enable Masquerading. This will allow Ghost to send emails with `noreply@example.com`. Do not turn off this setting later because Ghost will need to send email as the Support email address for subscription confirmation.
- Create the `support@` and `newsletter@` mailboxes in `example.com`. This is required to receive the confirmation mails.
- Change the addresses inside Ghost:
  - The Support email address can be changed in Membership -> Portal Settings -> Customize Portal -> Account page settings . Click on the received confirmation link.
  - The Newsletter email address can be changed in Email newsletter -> Newsletter -> Customize -> Email addresses -> Sender email address. Click on the received confirmation link.
- Click on the received confirmation links.

See this forum post for a step by step guide.

Bulk emails

Newsletters are sent out to members using Mailgun. Mailgun is required. You can sign up with Mailgun and setup the API Key at `Settings` -> `Email newsletter`.

7.53 Gitea App

7.53.1 About

Gitea is a community managed lightweight code hosting solution written in Go.

- Questions? Ask in the Cloudron Forum - Gitea
- Gitea Website
- Gitea forum
- Gitea issue tracker

7.53.2 Customizing Gitea

Customizing Gitea is typically done using the custom folder. This is the central place to override configuration settings, templates, etc.

On Cloudron, the custom data folder is located at `/app/data/custom`.

Gitea also supports various configuration customizations. To add customizations, use the File Manager and edit the file named `/app/data/app.ini`.

After editing, restart the app for the changes to take effect.

7.53.3 LFS

Gitea supports any s3-like storage as a backend for LFS and attachments (see Gitea Config Cheat Sheet). Add the following configuration and restarted gitea from the dashboard:

```
[server]
LFS_START_SERVER = true

[storage.my-storage]
STORAGE_TYPE = minio
SERVE_DIRECT = true
MINIO_ENDPOINT = s3.us-west-001.backblazeb2.com
MINIO_ACCESS_KEY_ID = {secret-id}
MINIO_SECRET_ACCESS_KEY = {secret-key}
MINIO_BUCKET = my-bucket
MINIO_LOCATION = us-west-001
MINIO_USE_SSL = true

[lfs]
STORAGE_TYPE = my-storage

[attachment]
STORAGE_TYPE = my-storage
MAX_SIZE = 50
```

7.53.4 CLI

The gitea CLI can be used as follows:

```
sudo -u git /home/git/gitea/gitea -c /run/gitea/app.ini  --help
```

For example, to change the root password:

```
sudo -u git /home/git/gitea/gitea -c /run/gitea/app.ini admin user change-password -u root -p changeme123
```

7.53.5 GPG

The package sets `GNUPGHOME` environment variable to `/app/data/gnupg` . The `gpg` CLI tool uses this path and so does gitea to locate keys.

7.54 GitHub Pages App

7.54.1 About

GitHub pages compatible repos can be published using this app.

- Questions? Ask in the Cloudron Forum - GitHub Pages
- GitHub Pages Website
- Upstream GitHub Pages forum
- Upstream GitHub Pages issue tracker

7.54.2 Publishing pages

Pages can be published by pushing the GitHub pages repo via SSH or HTTP.

SSH

To publish your page via SSH:

1. You must save your SSH public key to `/app/data/ssh/authorized_keys` using the File manager.
2. Add SSH remote and push

```
git remote add page ssh://git@<app.example.com>:<sshport>/app/data/repo.git
git push page master
```

The SSH Port can be configured in the Location section of the app.

HTTP

⚠ **Use SSH key instead**

This method is not recommended since it involves storing your credentials in plain text. We recommand using SSH keys instead.

To publish your page via HTTP:

```
git remote add page https://<app.example.com>/_git/page
git push page master
```

When pushing, `git` will prompt for Cloudron username and credentials. Any Cloudron user with access to the app can push.

It can be convenient to store the HTTP username and password in the `~/.netrc` file:

```
machine app.example.com login fred password bluebonnet
```

Adjust `app.example.com`, `fred` and `bluebonnet` above to your setup.

7.54.3 Build Branch

By default, the app renders the `master` branch. To change this, edit `/app/data/env.sh` using the File manager:

```
export BUILD_BRANCH=main
```

Be sure to restart the app, after making any changes.

7.54.4 Jekyll

The app uses the pages gem to statically build the website.

Limitations:

- Only Jekyll 3 is supported. For Jekyll 4, you must build the site manually.
- Only the plugins listed in the pages plugins page are supported.

7.54.5 Using mkdocs

mkdocs has a command called `gh-deploy` that can automatically build docs and publish the site to a specific remote and branch.

To force skip of Jekyll build, create a file name `.nojekyll` at the root of the repo.

```
git remote add page https://site.cloudron.xyz/_git/page # add the github-pages app remote
mkdocs gh-deploy --remote-name page --remote-branch master --force
```

7.54.6 Jekyll 4

This app only supports automatic build of Jekyll 3 websites. To publish a website with Jekyll 4:

- Install Jekyll locally with `sudo gem install jekyll`.
- Build the site locally on your laptop with `jekyll build`.
- Add the `_site` to your git repo.
- Add a `.nojekyll` at the root of the repo.
- Push to the pages app.

7.55 GitLab App

7.55.1 About

GitLab is the complete DevOps platform.

- Questions? Ask in the Cloudron Forum - GitLab
- GitLab Website
- GitLab forum
- GitLab issue tracker

7.55.2 Custom gitlab.yml

GitLab is customized using GitLab's admin interface. Some options can ony be changed in `gitlab.yml`. For such situations, use the File Manager and modify `/app/data/gitlab.yml` and restart the app.

7.55.3 Disabling registration

By default, GitLab allows external people to sign up. This can be disabled to restrict use only to Cloudron users as follows:

GitLab > Admin area > Settings > Features > remove the check mark "Sign-up enabled"

7.55.4 GitLab Runner for CI

GitLab CI involves installing one or more GitLab Runners. These runners carry out tasks as instructed by the main GitLab installation. When installing a runner, you have to select the project tags to which the runner will respond and the type of tasks ("executor") it can carry out. For example, there is a Shell executor, Docker executor etc.

Once GitLab runner is installed, you have to add the runner in GitLab. When adding the runner in GitLab, you can decide how GitLab schedules tasks in the runner ie. if the runner is exclusive to a project ('Specific Runner') or shared between projects ('Shared Runner) or specific to a group ('Group Runner').

Cloudron's GitLab package can be used with GitLab Runner as follows.

- First create a **new** server and install GitLab Runner on it following the instructions at GitLab docs. In short:

```
# For ubuntu
curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash
sudo apt-get install gitlab-runner
```

- Get the token listed in GitLab under `https://<gitlab.example.com>/admin/runners` (under shared runners section).
- Register the runner with the token from the above step

```
root@localhost:~# sudo gitlab-runner register
Running in system-mode.

Please enter the gitlab-ci coordinator URL (e.g. https://gitlab.com/):
https://gitlab.cloudron.xyz
Please enter the gitlab-ci token for this runner:
xzdZgdsXq5uSFCyAK7pP
Please enter the gitlab-ci description for this runner:
[localhost]: Shell Jobs Runner
Please enter the gitlab-ci tags for this runner (comma separated):

Whether to lock the Runner to current project [true/false]:
[true]: false
Registering runner... succeeded                     runner=xzdZgdsX
Please enter the executor: docker, docker-ssh, shell, ssh, virtualbox, docker-ssh+machine, parallels, docker+machine, kubernetes:
```

```
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
```

- The Runner should now be listed under `https://<gitlab.example.com>/admin/runners` .
- Now push a .gitlab-ci.yml to your project to start using the runner.

7.55.5 Rails Console

To open a rail console, first open the Web terminal and then run:

```
sudo -u git bundle exec rails c -e production
```

7.55.6 Reset Admin Password

To reset the admin password, run the following commands using the Web terminal:

```
# sudo -u git bundle exec rails c -e production
=> user = User.where(id: 1).first
=> user.password = 'NEW_PASS'
=> user.password_confirmation = 'NEW_PASS'
=> user.save
=> exit
```

7.55.7 App Settings

There are various other settings GitLab supports via `gitlab.rb`. On Cloudron those are specified in `/app/data/gitlab.yml`. The upstream docs are referring to the key style of `gitlab.rb` and the mapping of keys for the yml file are described here.

7.55.8 Incoming Email

GitLab has several features based on receiving incoming email messages:

- Reply by Email
- New issue by Email
- New merge request by email
- Service Desk.

See the upstream docs for more information.

Set up

If you are not using Cloudron Email server, configure `production.incoming_email` in `/app/data/gitlab.yaml` with the IMAP information. The `mailroom` service is automatically started withen `production.incoming_email.enabled` is set to true (verify using `supervisorctl status` in the Web Terminal).

If you are using Cloudron Email on the same server as the Gitlab installation:

- Create a mailbox say `git@cloudron.space`
- Set the inbox to use in the `Email` section of the app. When configured, Cloudron will configure GitLab to use Email sub-addressing and the `mailroom` service is enabled.

⚠ **Careful about the incoming email domain**

GitLab recommends using a subdomain for incoming email in their security note. On Cloudron, you can add the subdomain in the `Domains` view, enable email server for the subdomain and then select a subdomain email address.

Testing

- Reply by Email - Each issues has a per user email address. Click on the ribbon in the issue page to find this address.
- New issue by Email - You will find the unique email address per user in the issues page of the project.
- Service Desk - Enable this for a project in Settings -> General -> Service Desk. The support desk email address is displayed here.
- Service Desk custom address - This step is only required if you want to a set a different support email address than the incoming email address. To set a custom Service Desk address which is shared for the full instance, add a section `service_desk_email` in `/app/data/gitlab.yml` as mentioned here:

```
production: &base
  ... other configs ...

  service_desk_email:
    enabled: true
    address: "project_contact+%{key}@example.com"
    user: "project_contact@example.com"
    password: "[REDACTED]"
    host: "my.cloudrondomain.com"
    port: 993
    ssl: true
    start_tls: false
    log_path: "log/mailroom.log"
    mailbox: "inbox"
    idle_timeout: 60
    expunge_deleted: true
```

Be sure to restart GitLab after making the above changes.

7.55.9 Migration to Cloudron

This guide aims to assist in migrating an already running Gitlab into the Gitlab Cloudron app.

If you have any problems please do not delay and seek help in the Forum.

This guide got written for a migration from a Gitlab installation via Omnibus.

⚠ **Make sure both Gitlab instances are running the same version! Before continuing to the next step!**

The backup Rake task GitLab provides does not store your configuration files. The primary reason for this is that your database contains items including encrypted information for two-factor authentication and the CI/CD secure variables. Storing encrypted information in the same location as its key defeats the purpose of using encryption in the first place.

At the very minimum, you must backup:

For Omnibus:

- `/etc/gitlab/gitlab-secrets.json`
- `/etc/gitlab/gitlab.rb`

For installation from source:

- `/home/git/gitlab/config/secrets.yml`
- `/home/git/gitlab/config/gitlab.yml`

NOTE! Since we are switching from Omnibus TO a source installation (The Cloudron Gitlab app is a source installation) we will need to convert the `gitlab-secrets.json` to `secrets.yml`.

Also in the converted `secrets.yml` you will need to change `gitlab_rails:` to `production:` - otherwise gitlab will generate new secrets for rails.

Create a backup of your running Gitlab

Gitlab Documentation - Backup and Restore

> GitLab 12.2 or later:
>
> ```
> sudo gitlab-backup create
> ```
>
> GitLab 12.1 and earlier:
>
> ```
> gitlab-rake gitlab:backup:create
> ```
>
> If you installed GitLab from source, use the following command: (This is how to do it in Cloudron)
>
> ```
> sudo -u git -H bundle exec rake gitlab:backup:create RAILS_ENV=production
> ```
>
> For more examples please visit the official Gitlab Documentation - Backup and Restore

Save the generated file I.e. `1632462433_2021_09_24_14.2.4_gitlab_backup.tar` on your local computer.

Change the database owner in the created backup

In the Omnibus version the default user for the database was `gitlab` in my case.

You need to change this into the PostgreSQL user provided by Cloudron.

Go into the Web terminal of your Cloudron Gitlab app to get the username:

```
# echoing the single variable
echo $CLOUDRON_POSTGRESQL_USERNAME
userd5499e3cf81b43d093724d69fa223688

# getting all Postgresql variables
printenv | grep -i POSTGRES
CLOUDRON_POSTGRESQL_URL=postgres://
userd5499e3cf81b43d093724d69fa223688:ab1569471419f341ed83f18538b275c09c1389fdb248398640d48fdc8847275858aeca488021da55edb460051a2a0595f226602afc7828becd1c17d9
1f55eee2@postgresql/dbd5499e3cf81b43d093724d69fa223688
CLOUDRON_POSTGRESQL_DATABASE=dbd5499e3cf81b43d093724d69fa223688
CLOUDRON_POSTGRESQL_PASSWORD=ab1569471419f341ed83f18538b275c09c1389fdb248398640d48fdc8847275858aeca488021da55edb460051a2a0595f226602afc7828becd1c17d91f55eee2
CLOUDRON_POSTGRESQL_USERNAME=userd5499e3cf81b43d093724d69fa223688
CLOUDRON_POSTGRESQL_HOST=postgresql
CLOUDRON_POSTGRESQL_PORT=5432
```

The content of the backup looks like this:

```
1632462433_2021_09_24_14.2.4_gitlab_backup/
├── artifacts.tar.gz
├── backup_information.yml
├── builds.tar.gz
├── db
│   └── database.sql.gz
├── lfs.tar.gz
├── pages.tar.gz
├── repositories
│   └── @hashed
└── uploads.tar.gz
```

Extract the `database.sql.gz` to edit the `database.sql`.

We will need to replace all `OWNER TO gitlab;` strings to `OWNER TO userd5499e3cf81b43d093724d69fa223688;`.

Save the `database.sql` and `gzip` the file back into the `database.sql.gz`.

Put it back together into the `1632462433_2021_09_24_14.2.4_gitlab_backup.tar`.

Example all done in a terminal:

```
# Move the created backup into a seperate folder for extraction
# Extract the created tar
tar -xf 1632745419_2021_09_27_14.2.4_gitlab_backup.tar

# decompress the gziped database.sql.gz
```

```
gzip -d db/database.sql.gz

# replace all `OWNER TO gitlab;` with `OWNER TO OWNER TO userd5499e3cf81b43d093724d69fa223688;`
sed -e 's/OWNER TO gitlab;/OWNER TO userd5499e3cf81b43d093724d69fa223688;/' -i db/database.sql

# compress the `database.sql`
gzip db/database.sql

# Create the new `1632745419_2021_09_27_14.2.4_gitlab_backup.tar`
# You will get a warning since it wont tar it self
tar -cf 1632745419_2021_09_27_14.2.4_gitlab_backup.tar ./
tar: ./1632745419_2021_09_27_14.2.4_gitlab_backup.tar is the archive; not backed up.
```

Restoring the Backup

- Upload the new tar to the Gitlab Cloudron app to `/app/data/backups/1632745419_2021_09_27_14.2.4_gitlab_backup.tar`.
- Upload the converted `secrets.yml` to `/app/data/secrets.yml`
- Create a snapshot of the running Cloudron Gitlab app (this way we can jump back if something does not work, and you won't need to re-upload the backup)
- Open the Web terminal of the Cloudron Gitlab app and run the restore

```
sudo -u git -H GITLAB_ASSUME_YES=1 bundle exec rake --trace gitlab:backup:restore RAILS_ENV=production
```

Now the restore should start looking something like this:

```
`/home/git` is not writable.
Bundler will use `/tmp/bundler20210927-435-134v1je435' as your home directory temporarily.
. . .
2021-09-24 06:22:36 +0000 -- done
2021-09-24 06:22:36 +0000 -- Restoring uploads ...
2021-09-24 06:22:36 +0000 -- done
2021-09-24 06:22:36 +0000 -- Restoring builds ...
2021-09-24 06:22:36 +0000 -- done
2021-09-24 06:22:36 +0000 -- Restoring artifacts ...
2021-09-24 06:22:36 +0000 -- done
2021-09-24 06:22:36 +0000 -- Restoring pages ...
2021-09-24 06:22:36 +0000 -- done
2021-09-24 06:22:36 +0000 -- Restoring lfs objects ...
2021-09-24 06:22:36 +0000 -- done
```

Now migrate your custom settings from the `gitlab.rb` settings into `/app/data/gitlab.yml` file.

Restart the Cloudron Gitlab app.

Now everything should work as intended.

Migration of Gitlab Users to Cloudron LDAP Users

You might want to link old Gitlab users to the new Cloudron LDAP Users. This is rather simple, just make sure the username of the user is the same as in Cloudron.

Example: Gitlab username is `tina.testing` so the Cloudron username should also be `tina.testing`

7.56 Gogs App

7.56.1 About

Gogs is a painless self-hosted Git service.

- Questions? Ask in the Cloudron Forum - Gogs
- Gogs Website
- Gogs issue tracker

7.56.2 Customizing Gogs

Gogs supports various customizations. To add customizations, use the File Manager) and edit the file named `/app/data/app.ini`.

After editing, restart the app for the changes to take effect.

7.56.3 Custom Templates

You can override HTML templates (including templates for emails) by creating a customized version under `/app/data/custom/templates/` directory. See Custom Templates for more information.

You can edit the files using the File Manager. After editing, restart the app for changes to take effect.

7.56.4 CLI

The gogs CLI can be used as follows:

```
sudo -u git /home/git/gogs/gogs -c /run/gogs/app.ini
```

For example, to delete inactive accounts:

```
sudo -u git /home/git/gogs/gogs admin delete-inactive-users -c /run/gogs/app.ini
```

To create a backup:

```
sudo -u git GOGS_CUSTOM=/app/data/custom /home/git/gogs/gogs backup -c /run/gogs/app.ini --target /app/data/backups
```

7.57 Grafana App

7.57.1 About

Grafana is the open and composable observability and data visualization platform.

- Questions? Ask in the Cloudron Forum - Grafana
- Grafana Website
- Grafana forum
- Grafana issue tracker

7.57.2 Customizations

Custom configuration can be added in the file `/app/data/custom.ini` using the File Manager. See the Grafana docs on the various configuration options. Be sure to restart the app after making any changes.

7.57.3 Installing plugins

To install plugins, you run the grafana CLI tool using the Web Terminal.

For example,

```
# /app/code/bin/grafana cli -homepath /app/code -config /run/grafana/custom.ini plugins install grafana-worldmap-panel
```

7.57.4 Reset admin password

```
# /app/code/bin/grafana cli --homepath /app/code  --config /run/grafana/custom.ini admin reset-admin-password secret123
```

7.58 Grav App

7.58.1 About

Grav is a modern open source flat-file CMS.

- Questions? Ask in the Cloudron Forum - Grav
- Grav Website
- Grav forum
- Grav issue tracker

7.58.2 Admin Plugin

This package pre-installs the Admin plugin. This admin plugin for Grav is an HTML user interface that provides a convenient way to configure Grav and easily create and modify pages.

Do not uninstall admin plugin

While the Admin Plugin is totally optional in the upstream project, this package is designed to work with the Admin Plugin. It should not be uninstalled.

7.58.3 CLI

GPM and Grav commands can be executed by opening a Web terminal:

```
# cd /app/code
# sudo -u www-data -- /app/code/bin/gpm install bootstrap4-open-matter
# sudo -u www-data -- /app/code/bin/grav
```

7.58.4 Skeletons

Grav Skeletons are completely packaged sample sites. They include plugins, themes, pages in one bundle. They can be downloaded from here.

Skeletons don't work well with this Cloudron package:

- Skeletons provide the core and plugin files and they are often out-of-date. On Cloudron, core files are read only for update and security reasons.
- Skeletons may or may not contain the admin plugin. This package is designed for use with the admin plugin.

For this reason, it's best to use Grav Skeletons with the LAMP App.

- Install LAMP app
- Upload the Skeleton zip file and extract it to the `/app/data/public` folder using the File manager:
- Change the ownership of the `/app/data/public` directory to `www-data`.
- Your skeleton should be live!

7.58.5 Plugins

Git Sync

If you want to use the Git Sync plugin, you have to configure the git config.

```
su www-data -c 'git config --global user.name YourUsername'
u www-data -c 'git config --global user.email your@mail.address'
```

7.59 Greenlight App

7.59.1 About

Greenlight is a really simple end-user interface for your BigBlueButton server.

- Questions? Ask in the Cloudron Forum - Greenlight
- Greenlight Website
- Greenlight issue tracker

7.59.2 Installing BigBlueButton

This app is intended to work alongside a BigBlueButton installation. Greenlight is the frontend which has Cloudron user authentication and the BigBlueButton install is the backend. BBB must be installed on a separate VM following the instructions in BigBlueButton documentation and bbb-install script documentation

To summarize the installation procedure:

- Create a new 64-bit Ubuntu 20.04 server with a public IP and 4G of memory and 4 CPU. Note installation script does not work without these minimum requirements.
- Configure the DNS to point `bbb.example.com` to the public IP below.
- Execute

```
wget -qO- https://raw.githubusercontent.com/bigbluebutton/bbb-install/v2.7.x-release/bbb-install.sh | bash -s -- -w -v focal-270 -s bbb.example.com -e
info@example.com
```

- Install Greenlight on Cloudron.
- Take the output of `bbb-conf --secret` on the BigBlueButton server and put these values to the following variables in `/app/data/env` of Greenlight using the File manager.

```
BIGBLUEBUTTON_ENDPOINT=
BIGBLUEBUTTON_SECRET=
```

- Restart the Greenlight app.

7.59.3 User management

Cloudron Directory

When Cloudron user management is enabled, only Cloudron users can login to greenlight, create and manage meetings. By default, Cloudron users have `Administrator` role permissions. This can be changed in `Administrator Panel > Site Settings > Registration`.

Without Cloudron SSO

When not using Cloudron authentication, there is an Administrator account created with the following credentials at the first run:

```
username: admin@cloudron.local
password: ChangeMe!321
```

7.59.4 Creating a new user with a role

To create a new User, open a Web terminal and run the following commands:

```
# Creating an admin
bundle exec rake admin:create["admin","admin@server.local","changeme"]

# Information
# bundle exec rake user:create["name","email","password","role"]
# Creating a user
bundle exec rake user:create["user","user@server.local","changeme","User"]
```

7.60 Guacamole App

7.60.1 About

Apache Guacamole is a clientless remote desktop gateway. It supports standard protocols like VNC, RDP, and SSH.

- Questions? Ask in the Cloudron Forum - Guacamole
- Guacamole Website
- Guacamole support
- Guacamole issue tracker

7.60.2 User Management

Apache Guacamole is configured to use Cloudron's OIDC Provider. To grant a user access to a resource, you have to manually create the user in Guacamole (Settings -> Users). Only a username is required and will be mapped to Cloudron's username.

Guacamole also maps a user's group. To assign permissions to a group, create the group using its Cloudron group name (Settings -> Groups).

7.60.3 RDP

Most Windows/RDP servers do not have a valid certificate installed. For this reason, be sure to check the ignore server certificate checkbox in the Parameters section.

7.60.4 Guacamole menu

The Guacamole menu is a sidebar which is hidden until explicitly shown. On a desktop or other device which has a hardware keyboard, you can show this menu by pressing **Ctrl+Alt+Shift**. If you are using a mobile or touchscreen device that lacks a keyboard, you can also show the menu by swiping right from the left edge of the screen.

See the docs for more information.

7.60.5 Extensions

Extensions can be downloaded from the Apache Guacamole releases page. They are packaged as tar.gz and when extracted have a jar file inside them.

To add an extension:

- Upload the jar file (and not .tar.gz) to `/app/data/extensions` using the File manager
- Restart Guacamole

Recording

The plugin `guacamole-history-recording-storage` enables the recordings of sessions.

Download the version that fits your Guacamole installation from https://guacamole.apache.org/releases/.

Upload the `.jar` file from inside the archive to your Guacamole app `/app/data/extensions` with the Web File Manager or the Web Terminal.

With the Web Terminal create a folder for the recordings:

```
mkdir -p /app/data/recordings
chown -R cloudron:tomcat /app/data/recordings
chmod 2750 /app/data/recordings/
```

Edit the configuration file `/app/data/guacamole.properties` with the Web File Manager or the Web Terminal and add the following line:

```
recording-search-path: /app/data/recordings
```

Restart the app.

With an Admin user you can now edit your connections and under `Screen Recording` set the following configurations:

- `Recording path:` to `${HISTORY_PATH}/${HISTORY_UUID}`
- `Recording name:` to `${GUAC_DATE}-${GUAC_TIME}`
- `Automatically create recording path:` checked

Save the configuration of the connection.

If you now start a session recording files are stored under `/app/data/recordings` and viewable with the history viewer.

7.60.6 Custom properties

Some extensions require adding custom properties. You can add these using the File manager by editing `/app/data/guacamole.properties`. Be sure to restart the app after you make changes.

7.60.7 Branding

To change the logo and style, you have to use an extension. You can start with the zipping the contents in this repo.

- `zip -r /tmp/guacamole-branding-example.jar ./` inside the directory to create the jar file.
- Upload it using the File manager to `/app/data/extensions`
- Restart Guacaomole

The login screen should already be different now. You can now modify the `guacamole-branding-example.jar` as needed to make changes. Note that a `jar` file is just a zip archive. Just extract it, change files and compress it again. When re-creating the jar file, make sure that the files are at the top level and not inside a sub directory.

7.61 Hastebin App

7.61.1 About

Hastebin is an open source pastebin written in node.js.

- Questions? Ask in the Cloudron Forum - Hastebin
- Hastebin Website
- Hastebin issue tracker

7.61.2 Deleting pastes

Pastes never expire and have to be cleaned up manually. For this, use the File Manager and simply remove the pastes stored under `/app/data` as needed.

7.61.3 Custom configuration

To add custom configuration, edit `/app/data/config.json` using the File manager and restart the app.

7.62 HedgeDoc App

7.62.1 About

HedgeDoc is the best platform to write and share markdown.

- Questions? Ask in the Cloudron Forum - HedgeDoc
- HedgeDoc Website
- HedgeDoc issue tracker

Impending rename

CodiMD is being renamed to HedgeDoc. This app package will be renamed accordingly.

7.62.2 Custom configuration

Use the File manager to place custom configuration under `/app/data/config.json`.

See HedgeDoc docs for configuration options reference.

7.62.3 Image uploads

By default, images are uploaded to the data directory of the app itself. To switch to another provider like MinIO, first configure minio to image uploads. Then, use the following configuration in `/app/data/config.json`:

```
    "imageUploadType": "minio",
    "s3bucket": "codimd-images",
    "minio": {
        "accessKey": "MINIO_ACCESS_KEY",
        "secretKey": "MINIO_SECRET_KEY",
        "endPoint": "minio.cloudrondomain.com",
        "secure": true,
        "port": 443
    }
```

7.62.4 CLI Tool

The CLI tool `bin/manage_users` can be used to manage users. Open a Web Terminal:

```
root@8b1237e8-6ae8-4dd8-8bce-22b404c03f9d:/app/code# export CMD_DB_URL="${CLOUDRON_POSTGRESQL_URL}"
root@8b1237e8-6ae8-4dd8-8bce-22b404c03f9d:/app/code# bin/manage_users
You did not specify either --add or --del or --reset!

Command-line utility to create users for email-signin.

Usage: bin/manage_users [--pass password] (--add | --del) user-email
    Options:
        --add   Add user with the specified user-email
        --del   Delete user with specified user-email
        --reset Reset user password with specified user-email
        --pass  Use password from cmdline rather than prompting
```

7.63 Home Assistant App

7.63.1 About

Open source home automation that puts local control and privacy first. Powered by a worldwide community of tinkerers and DIY enthusiasts.

- Questions? Ask in the Cloudron Forum - Home Assistant
- Home Assistant Help
- Home Assistant Website

7.64 HumHub App

7.64.1 About

HumHub is a free and open-source social network software written on top of the Yii PHP framework that provides an easy to use toolkit for creating and launching your own social network.

- Questions? Ask in the Cloudron Forum - HumHub
- HumHub Website
- HumHub issue tracker

7.64.2 Themes

Themes can be installed from the HumHub marketplace. In such a case, it is installed into the module directory.

The HumHub theme structure also supports installation into the `themes` subdirectory. For this, use the File manager to create a subdirectory in `/app/data/themes` and upload the theme.

7.65 Immich App

7.65.1 About

Immich is a high performance self-hosted photo and video backup solution.

- Questions? Ask in the Cloudron Forum - Immich
- Immich Website
- Immich issue tracker

7.65.2 Settings

The Cloudron app works based on a config file at `/app/data/immich.json`. The settings UI in Immich is **disabled**.

All config options can be seen here

After changing the file, the app needs to be restarted.

7.65.3 immich-admin CLI Tool

Immich comes with the `immich-admin` tool.

For available commands and details see: Immich Admin CLI Documentation

Reset Admin Password

If you have lost access to the Immich admin account, you can reset the password using the `immich-admin` CLI tool.

Run the following command and follow the prompts to reset the password:

```
immich-admin reset-admin-password
```

7.66 Invidious App

7.66.1 About

Invidious is an open source alternative front-end to YouTube.

- Questions? Ask in the Cloudron Forum - Invidious
- Invidious Website
- Invidious Documentation
- Invidious Contact
- Invidious Issue Tracker

7.66.2 Periodic Restart

The upstream docs recommends restarting Invidious often, at least once a day, ideally every hour because of various issues.

If you hit issues, you can set up a cron job to restart it periodically:

```
# restart every hour
0 * * * * echo "==> Restarting invidious periodically" && supervisorctl restart invidious
```

7.66.3 Password Reset

Note

From the official documentation Reset user password abridged to fit Cloudron needs.

The following script can be copy and pasted into your Web Terminal of the Invidious app and will create a new admin password.

You will need the user id of which user your want to reset the password for.

To list all users you can run:

```
PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE} -c "SELECT email FROM users;"
```

Copy and paste this into your Web Terminal

```
#!/bin/sh
set -e
clear

printf 'User ID: '
read -r ID
if [ "$(PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE} -c "SELECT email FROM users WHERE email = '$ID';" | tail -n 2 | head -n 1)" != '(1 row)' ]; then
    echo 'Error: User ID does not exist'
    exit 1
fi
PASSWORD=$(openssl rand -hex 16)
printf "New password is: $PASSWORD\n"
HASH=$(python3 -c "import bcrypt; print(bcrypt.hashpw(b\"$PASSWORD\", bcrypt.gensalt(rounds=10)).decode(\"ascii\"))")
PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE} -c "UPDATE users SET password = '$HASH' WHERE email = '$ID';" &> /dev/null
```

7.67 Invoice Ninja 5 App

7.67.1 About

InvoiceNinja 5 is the leading self-host platform to create invoices, accept payments, track expenses & time tasks. Support WePay, Stripe, Braintree, PayPal, Zapier, and more!

- Questions? Ask in the Cloudron Forum - Invoice Ninja
- Invoice Ninja Website
- Invoice Ninja forum

7.67.2 Changing Domain

Invoice Ninja stores the domain for each company dataset. This is saved when the company is first setup and cannot be changed later from the UI. If the domain for the app instance is adjusted in Cloudron, those database records need to be updated manually.

For instances which only use **one domain** for all company datasets run the following mysql command in a web terminal:

```
mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST} ${CLOUDRON_MYSQL_DATABASE} -e "UPDATE companies
SET portal_domain='${CLOUDRON_APP_ORIGIN}'"
```

Otherwise adjust the SQL command above to only update the domains which need updating.

The client portal URL may also need to be updated. This can be done from the Invoice Ninja internal settings page under the `Client Portal` section.

7.67.3 Customizations

InvoiceNinja customizations can be made by opening a File Manager and editing `/app/data/env` .

7.68 IT Tools App

7.68.1 About

IT-Tools is a collection of handy online tools for developers, with great UX.

- Questions? Ask in the Cloudron Forum - IT-Tools
- IT-Tools issue tracker
- IT-Tools forum

7.69 Jellyfin App

7.69.1 About

Jellyfin is the volunteer-built media solution that puts you in control of your media. Stream to any device from your own server, with no strings attached. Your media, your server, your way.

- Questions? Ask in the Cloudron Forum - Jellyfin
- Jellyfin Website
- Jellyfin issue tracker

7.69.2 Hardware Transcoding

Cloudron 5.6 required

Cloudron 5.6 is the first release that supports hardware transcoding.

Jellyfin supports hardware acceleration on Linux - Nvidia NVDEC, VA API and Intel QuickSync. Cloudron does not support Nvidia at the time of this writing.

There are various steps to check if your hardware supports transcoding and if Jellyfin is able to take advantage of it.

- Check the output of `vainfo` on your server. You might have to run `apt-get install vainfo libva2 i965-va-driver` if that command is not available on your server. The output should look like below. `VAEntrypointVLD` means that your card is capable to decode this format, `VAEntrypointEncSlice` means that you can encode to this format.

```
$ vainfo
error: can't connect to X server!
libva info: VA-API version 1.1.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_1_1
libva info: va_openDriver() returns 0
vainfo: VA-API version: 1.1 (libva 2.1.0)
vainfo: Driver version: Intel i965 driver for Intel(R) CherryView - 2.1.0
vainfo: Supported profile and entrypoints
      VAProfileMPEG2Simple            : VAEntrypointVLD
      VAProfileMPEG2Simple            : VAEntrypointEncSlice
      VAProfileMPEG2Main              : VAEntrypointVLD
      VAProfileMPEG2Main              : VAEntrypointEncSlice
      VAProfileH264ConstrainedBaseline: VAEntrypointVLD
      VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice
      VAProfileH264Main               : VAEntrypointVLD
      VAProfileH264Main               : VAEntrypointEncSlice
      VAProfileH264High               : VAEntrypointVLD
      VAProfileH264High               : VAEntrypointEncSlice
      VAProfileH264MultiviewHigh      : VAEntrypointVLD
      VAProfileH264MultiviewHigh      : VAEntrypointEncSlice
      VAProfileH264StereoHigh         : VAEntrypointVLD
      VAProfileH264StereoHigh         : VAEntrypointEncSlice
      VAProfileVC1Simple              : VAEntrypointVLD
      VAProfileVC1Main                : VAEntrypointVLD
      VAProfileVC1Advanced            : VAEntrypointVLD
      VAProfileNone                   : VAEntrypointVideoProc
      VAProfileJPEGBaseline           : VAEntrypointVLD
      VAProfileJPEGBaseline           : VAEntrypointEncPicture
      VAProfileVP8Version0_3          : VAEntrypointVLD
      VAProfileVP8Version0_3          : VAEntrypointEncSlice
      VAProfileHEVCMain               : VAEntrypointVLD
```

- Enable Hardware transcoding in Jellyfin. Admin Dashboard > Playback > Transcoding.
- Next step is to check the Jellyfin logs of the name `FFmpeg.Transcode-*`. This can be found at Admin Dashboard > Logs. They are also at `/app/data/jellyfin/log`. Please note that if this file does not exist, it's probably not transcoding.
- Finally, when the video is playing, open a new browser tab and see the `Active Devices` in the Emby dashboard. Click on 'i' for transcoding information.

7.70 Jingo App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.70.1 About

The aim of Jingo Wiki is to provide an easy way to create a centralized documentation area for people used to work with git and markdown.

- Questions? Ask in the Cloudron Forum - Jingo
- Jingo Website
- Jingo issue tracker

7.70.2 Custom configuration

Use the File manager to place custom configuration under `/app/data/config.yml`.

See Jingo docs for configuration options reference.

7.70.3 Look and feel

You can add a sidebar, footer, custom CSS and custom Javascript by following the instructions in Jingo docs.

7.71 Jirafeau App

7.71.1 About

Jirafeau is a "one-click-filesharing": Select your file, upload, share a link. That's it.

- Questions? Ask in the Cloudron Forum - Jirafeau
- Jirafeau Website
- Jirafeau issue tracker

7.71.2 Change admin password

1. Choose a new password
2. Create a sha256 hash of the new password. To do this, use the web terminal (or any terminal) and do:

```
echo -n "newpassword" | sha256sum
```

1. Then, use the File Manager to edit `custom.php` and set the sha256 hash of your new password generated in step 2 above.

```
$cfg['admin_password'] = '<new sha256 hash of your new password goes here>';
```

Disable admin interface

To disable the admin interface, leave the password blank. Like `$cfg['admin_password'] = ''`.

7.71.3 Customization

The app allows for a multitude of customization by overwriting config values in `/app/data/custom.php` using the File Manager.

All options can be seen here.

For example settings the page title can be done by adding the following:

```
$cfg['organisation'] = 'My filedrop';
```

7.71.4 Restricting Uploads

By default, anyone can upload to the instance. You can add customizations in `/app/data/custom.php` to protect your instance.

To protect uploads with a password:

```
$cfg['upload_password'] = array('psw1');        // One password
```

Alternately, uploads can be restricted by IP address(es):

```
$cfg['upload_ip'] = array('123.45.0.0/16');
```

If you set a password, you can set a list of IPs that can upload without password:

```
$cfg['upload_ip_nopassword'] = array();
```

7.71.5 Restrict file availabilities

If you wish to restrict how long files are available you need to add the following configuration in `/app/data/custom.php`.

For this example, we only want to allow `week` as possible selection.

```
$cfg['availabilities'] = array(
    'minute' => false,
    'hour' => false,
    'day' => false,
    'week' => true,
    'fortnight' => false,
    'month' => false,
    'quarter' => false,
    'year' => false,
    'none' => false
);
```

7.71.6 Themes

To change the theme, edit `/app/data/custom.php`. List of theme names is available here.

```
$cfg['style'] = 'dark-courgette';
```

If you want to make a custom theme, then use the Web Terminal to first copy over from an existing theme (themes are located under `/app/code/media.original`):

```
# cd /app/data/media
# cp -r /app/code/media.original/dark-courgette/ mydarkhorse
```

Edit the new theme `mydarkhorse` to your content. Then, change `/app/data/custom.php`:

```
$cfg['style'] = 'mydarkhorse';
```

7.71.7 Custom ToS

A custom ToS can be placed in `/app/data/tos.local.txt` using the File Manager.

If you like, you can start with a copy of `/app/code/lib/tos.original.txt` and edit as needed.

7.72 Jitsi App

7.72.1 About

Jitsi Meet is an open source JavaScript WebRTC application used primarily for video conferencing.

- Questions? Ask in the Cloudron Forum - Jitsi
- Jitsi Website
- Jitsi docs
- Jitsi issue tracker

7.72.2 Custom configuration

Custom configuration can be set in `/app/data/jitsi-meet-config.js` using the File manager. See the upstream config file for configuration options.

Be sure to restart the app after making any changes.

7.72.3 Ports

Jitsi uses P2P for conferences with only 2 people.

With more than 2 people, it uses the Jitsi Videobridge for mixing audio/video. Jitsi requires the port `UDP/10000` to be open for conferences more than 2 people.

7.73 Joplin Server App

7.73.1 About

Joplin is a free, open source note taking and to-do application, which can handle a large number of notes organised into notebooks.

- Questions? Ask in the Cloudron Forum - Joplin Server
- Website
- Issue tracker
- Upstream Forum

7.73.2 Clients

Download the clients here.

7.73.3 Custom configuration

Custom configuration can be set by editing `/app/data/config` using the File manager. See the list of configurable options here.

Be sure to restart the app after making changes.

7.74 JupyterHub App

7.74.1 About

JupyterHub brings the power of notebooks to groups of users. It gives users access to computational environments and resources without burdening the users with installation and maintenance tasks.

- Questions? Ask in the Cloudron Forum - Jupyter Hub
- Jupyter Hub Website
- Jupyter Hub Community
- Jupyter Hub issue tracker

7.74.2 How it works

The JupyterHub app is run as a container (like any other Cloudron app). The hub app manages user login and creates a separate container for each user's notebooks. The notebook container is created from the `c.DockerSpawner.image` setting (see below on how to customize this). Each notebook container is run with a configurable memory limit based on `c.Spawner.mem_limit`. The advantage of this approach is that you can control how much compute/memory is allocated to each user and a notebook cannot bring down the whole server.

If you change the notebook image or any configuration, the notebook containers have to be "recreated". To help with this, the `/app/code/remove_notebook_containers.py` script can be run. Note that this only removes the containers but not the data of user's notebooks itself.

7.74.3 Selecting a notebook image

By default, the app uses the `jupyter/datascience-notebook`. The upstream Jupyterhub project maintains many other notebook images.

To use a different notebook image, use the File Manager to place custom configuration under `/app/data/customconfig.py`. For example, add a line like below:

```
c.DockerSpawner.image = 'quay.io/jupyter/all-spark-notebook:lab-4.1.5'
```

It is also possible to use any arbitrary docker image built from `jupyter/base-notebook` or any of the specialized notebooks that are base on this. For example:

```
FROM quay.io/jupyter/all-spark-notebook

RUN conda install --yes -c conda-forge git-lfs
```

Build and push the above image to Dockerhub. Then, update `c.DockerSpawner.image` as shown above.

To apply the configuration, restart the app using the Restart button.

Remove existing notebook containers

For the container image to take effect, you have to remove any existing docker notebook containers using the `/app/code/remove_notebook_containers.py` script. Notebook data will be intact despite deleting the container.

7.74.4 Notebook Memory limit

By default, notebooks are given 500M (including swap). This can be changed by editing `/app/data/customconfig.py`.

```
c.Spawner.mem_limit = '1G'
```

To apply the configuration, restart the app using the Restart button.

Remove existing notebook containers

For the memory limit to take effect, you have to remove any existing docker notebook containers using the `/app/code/remove_notebook_containers.py` script. Notebook data will be intact despite deleting the container.

7.74.5 Notebook persistence

All notebooks are part of the application backup and persisted across updates.

Libraries installed using `conda` are not part of the backup and are part of the notebook container. Idle notebooks are shutdown over time but they are not destroyed. This means that if any libraries installed in notebook container will generally persist.

If the notebook container is deleted, any libraries that were previously installed have to be re-installed.

7.74.6 Multiple user environments

By default, the app allows a user to create multiple environments. You can disable this by setting `allow_named_servers` to `False` in `/app/data/customconfig.py`.

```
c.JupyterHub.allow_named_servers = False
```

7.74.7 Sharing

To share notebooks between users with a shared directory (`/shared`):

- Create `/app/data/shared` directory and make it accessible by all users. Open a JupyterHub Web Terminal.

```
mkdir /app/data/shared
chmod 0777 /app/data/shared
```

- Then, edit `/app/data/customconfig.py` , add the configuration:

```
    c.DockerSpawner.volumes['/app/data/shared'] = '/home/jovyan/shared'
```

- Remove existing user notebook workspaces by running `/app/code/remove_notebook_containers.py` in JupyterHub's Web Terminal.
- Restart the app

The above approach can be extended to use Cloudron Volumes . Please note that when using Volumes the data in the shared directory is not backed up because Volume storage is not part of backups.

7.74.8 Extensions

It's possible to enable and install extensions. However, as note in Notebook persistence, the extensions installed using pip or conda are not part of the backup and thus they need to re-installed when the notebook image is changed.

7.74.9 Other custom configuration

Use the File Manager to place custom configuration under `/app/data/customconfig.py`.

See the docs for more information.

7.75 Kanboard App

7.75.1 About

Kanboard is a free and open source Kanban project management software.

- Questions? Ask in the Cloudron Forum - Kanboard
- Kanboard Website
- Kanboard issue tracker

7.75.2 Installing plugins

Kanboard Plugins are used to extend Kanboard. Kanboard has a UI to install and uninstall plugins at `/app/data/plugins` directly.

Plugins can also be installed manually at `/app/data/plugins`. Use the file manager to upload and extract plugins under that directory.

7.75.3 Custom configuration

Custom plugin configuration can be stored in `/app/data/customconfig.php`. To edit the file, use the file manager for the app.

7.76 Kavita App

7.76.1 About

Kavita is a fast, feature rich, cross platform reading server. Built with the goal of being a full solution for all your reading needs.

- Questions? Ask in the Cloudron Forum - Kavita
- Kavita Website
- Kavita issue tracker
- Kavita Discussions
- Kavita Wiki

7.76.2 Passwords

The maximum password length for Kavita is 32 characters.

7.77 Keila App

7.77.1 About

Keila is an Open-source email newsletters for creators and businesses.

- Questions? Ask in the Cloudron Forum - Keila
- Keila Website
- Keila Community
- Keila Docs
- Keila issue tracker

7.78 Keycloak App

7.78.1 About

Keycloak is an Open Source Identity and Access Management.

- Questions? Ask in the Cloudron Forum - Keycloak
- Keycloak Website
- Keycloak issue tracker
- Keycloak community

7.78.2 Features

Keycloak has packed some functionality in features.

The following features are enabled in the Cloudron package:

- authorization
- account:v3
- account-api
- impersonation
- client-policies
- passkeys (preview, may change of be removed in a future release)

To add new features, edit `/app/data/env.sh` using the File Manager and change `KEYCLOAK_BUILD_ARGS` as required . Restart the app for the changes to take effect.

7.79 Kimai App

7.79.1 About

Kimai is a free & open source timetracker.

- Questions? Ask in the Cloudron Forum - Kimai
- Kimai Website
- Kimai issue tracker

7.79.2 Plugins

Kimai2 supports both free and paid plugins or sometimes called bundles. The marketplace offers a selection of mostly paid plugins, while there are many free ones on github. Plugins can be obtained either through git checkout or downloading and extracting a zip bundle into `/app/data/plugins`.

The following example installes the demo plugin using the Web terminal into the running app instance:

```
cd /app/data/plugins
git clone https://github.com/Keleo/DemoBundle.git
chown -R www-data.www-data .
cd /app/code
sudo -u www-data bin/console kimai:reload
```

7.79.3 Customization

Use the File Manager to edit custom configuration under `/app/data/local.yaml`.

See Kimai customization docs for more information.

Once the `local.yaml` is updated, restart the app from the Cloudron dashboard.

7.80 Koel App

7.80.1 About

Koel is a personal music streaming server.

- Questions? Ask in the Cloudron Forum - Koel
- Koel Website
- Koel issue tracker

7.80.2 Increasing the upload limit

In order to increase the upload limit of the Koel App you need to edit two files with either the Web Terminal or the Web File manager.

Edit the `/app/data/php.ini` and add the following lines lines - in this example we set the limit to 1GB.

```
; upload_max_filesize and post_max_size are set to 50M by default
upload_max_filesize = 1G
post_max_size = 1G
```

Save and close the file.

Now edit the `/app/data/public/htaccess` file. In this file you should find and edit the following two lines to also represent the new limit of 1GB:

```
php_value upload_max_filesize 1G
php_value post_max_size 1G
```

7.81 Komga App

7.81.1 About

Komga is a media server for your comics, mangas, BDs, magazines and eBooks.

- Questions? Ask in the Cloudron Forum - Umami
- Komga Website
- Komga docs
- Komga issue tracker

7.81.2 Reset password

To reset password of a user:

- Place the app in recovery mode
- In the Web Terminal, run the following:

```
/usr/local/bin/gosu cloudron:cloudron java -jar komga.jar --komga.config-dir=/app/data/komga --reset=test@cloudron.io --newpassword=YourNewPassword
```

- You should already be able to login to Komga with the new password
- Disable Recovery Mode

7.82 Kopano Meet App

7.82.1 TURN Server

The Kopano Meet app is pre-configured to use Cloudron's built-in TURN server. No additional configuration is required.

7.83 Kutt App

7.83.1 About

Kutt is a free Modern URL Shortener.

- Questions? Ask in the Cloudron Forum - Kutt
- Kutt Website

7.83.2 Custom config

Custom configuration can be put in `/app/data/env` using the Web Terminal or the File Manager. The `env` file contains various customization options with documentation on them. After changing the file, make sure to restart the app.

7.83.3 Themes

Themes can be placed in `/app/data/custom` directory using the File Manager. See upstream docs on how to create styles and change images.

7.83.4 Registration

Registration is disabled by default. It can be enabled by editing `/app/data/env` using the File manager:

```
DISALLOW_REGISTRATION=false
```

Be sure to restart the app after making changes.

7.83.5 3rd party packages

Kutt integrates with a variety of languages and frameworks. See upstream docs for more information

7.83.6 Custom domains

Kutt supports having more than one domain. You can add domains in the custom domain section. Note `Set domain` below is a bit misleading because it's really `Add domain`.

> **Ignore instruction to add A record**
>
> You can ignore the instruction above to add an A record to the server's IP address. Cloudron will add this record when you create the aliases below.

Then, in the Cloudron Dashboard add the custom domains as domain aliases in the `Location` view:

7.84 Lamp App

7.84.1 About

Running LAMP apps on the Cloudron is no different than what is available on many hosting providers. You can upload your PHP code using SFTP or the File Manager and then modify the `.htaccess` and `php.ini` files as required. Most commonly used PHP extensions are pre-installed and you don't have to worry about keeping them up-to-date.

The main advantage of using the Cloudron to host LAMP apps are:

- DNS configuration, Let's Encrypt (SSL) certificate installation and renewal are automated.
- Can use MySQL, redis and send email out of the box.
- Don't have to worry about app and server backups, restore and updates since the Cloudron takes care of it.
- Run multiple LAMP apps, isolated from one another, on same server easily.
- Questions? Ask in the Cloudron Forum - LAMP

7.84.2 Supported PHP Versions

The LAMP app supports the following PHP versions:

- 8.1
- 8.2
- 8.3 (default)
- 8.4
- 8.5

To switch the PHP version, edit `/app/data/PHP_VERSION` using the file manager and restart the app.

See supported versions for PHP support status.

⚠ **PHP CLI**

The `php` binary is hardcoded to use PHP 8.3. In scripts, use `php8.2`, `phar8.2` and so on explicitly.

7.84.3 Uploading Files

The LAMP app can be uploaded using the File Manager or SFTP.

SFTP

The app can be uploaded using an SFTP client like FileZilla.

You can find the SFTP login details by clicking the Documentation drop down.

✎ **SFTP Access**

SFTP access for non-admin users can be granted using the operator role.

7.84.4 PHP settings

Custom PHP settings can be added in two ways:

- App's apache configuration - `/app/data/apache/app.conf`
- Via htaccess - `/app/data/public/.htaccess`

The files above can be edited using the File Manager. Note that settings with a mode of `PHP_INI_SYSTEM` cannot be set in htaccess files.

Example htaccess configuration:

```
#example
php_value post_max_size 600M
php_value upload_max_filesize 600M
php_value memory_limit 128M
php_value max_execution_time 300
php_value max_input_time 300
php_value session.gc_maxlifetime 1200
```

7.84.5 Apache settings

Custom Apache settings can be added in two ways:

- App's apache configuration - `/app/data/apache/app.conf`
- Via htaccess - `/app/data/public/.htaccess`

The files above can be edited using the File Manager. Be sure to restart the app after making changes.

Example htaccess configuration:

```
ServerSignature Off
```

7.84.6 Custom HTTP headers

Custom HTTP headers can be set in `/app/data/public/.htaccess`. apache `mod_headers` is already enabled. See this article for more information.

7.84.7 PHP extensions

The LAMP app already includes most of the popular PHP extensions including the following:

- php-apcu
- php-cli
- php-curl
- php-fpm
- php-gd
- php-gmp
- php-imap
- php-intl
- php-json
- php-mbstring
- php-mcrypt
- php-mysql
- php-mysqlnd
- php-pgsql
- php-redis
- php-sqlite
- php-xml
- php-xmlrpc
- php-zip

You can check the complete list of pre-installed extensions by visiting the default index.php of the app that prints out `phpInfo()`.

7.84.8 Installing custom PHP extensions

The LAMP app supports installing custom PHP extensions. As an example, we will install ionCube Loader, which is often required to install commercial PHP apps.

ionCube is already installed

The LAMP app has built-in support for ionCube. The installation steps for ionCube here are just an example.

Step 1: Download extension

Download and extract the `tar.gz` or `zip` Linux 64-bit ionCube packages to your PC/Mac from the ionCube website or use the direct link.

Step 2: Upload using SFTP

Upload the extracted directory to the SFTP root directory (`/app/data`) of the Cloudron app (i.e one level above `public/`).

Step 3: Enable extension

In the top level directory of the Cloudron app (in `/app/data`), you will find a `php.ini`.

Add the following line to enable the extension (just add it before the many `;extension` lines):

```
zend_extension=/app/data/ioncube/ioncube_loader_lin_7.2.so
```

The LAMP app has thread safety disabled, so we choose the extension without the `ts` extension.

Step 4: Restart app

Lastly, restart the app for the extension to be enabled.

Step 5: Verifying installation

Visit the LAMP app's default page to verify that the extension is enabled.

7.84.9 Configuring MySQL

Database credentials can be found in `/app/data/credentials.txt` using the File manager.

On a technical note, MySQL credentials are exposed as environment variables to the app. These variables can change over time. This approach makes it possible for Cloudron to transparently rotate the MySQL password periodically as a security measure and also makes app easily migratable across Cloudrons.

The exposed environment variables are:

```
CLOUDRON_MYSQL_URL=            # the mysql url (only set when using a single database, see below)
CLOUDRON_MYSQL_USERNAME=       # username
CLOUDRON_MYSQL_PASSWORD=       # password
CLOUDRON_MYSQL_HOST=           # server IP/hostname
CLOUDRON_MYSQL_PORT=           # server port
CLOUDRON_MYSQL_DATABASE=       # database name (only set when using a single database, see below)
```

If the PHP app has a `config.php` that requires the MySQL credentials to be set, they can set as below:

```
'db' => array (
    'hostname' => getenv("CLOUDRON_MYSQL_HOST"),
    'username' => getenv("CLOUDRON_MYSQL_USERNAME"),
    'password' => getenv("CLOUDRON_MYSQL_PASSWORD"),
    'database' => getenv("CLOUDRON_MYSQL_DATABASE")
  ), // Database configuration
```

Some apps show a setup screen and will require the raw MySQL credentials. For such apps, the MySQL credentials can be obtained using the File Manager inside the file `/app/data/credentials.txt`.

IMPORTANT: Once the installation is completed, be sure to switch the config file of the app to use the environment variables using `getenv()` instead of the raw credentials. Otherwise, future updates might break the app.

Customizing MySQL

On Cloudron, the MySQL server is shared across all apps. Each app gets non-root credentials to the database that helps isolate them from one another. This means one cannot configure mysql for one app specifically.

However, many MySQL variables like `sql_mode` can be set per session by modifying your code as follows:

```
// connect to mysql and call the first query
mysql_query("SET SESSION SQL_MODE = 'TRADITIONAL'");
mysql_query("SET SESSION UNIQUE_CHECKS = false");
mysql_query("SET SESSION FOREIGN_KEY_CHECKS=0");
```

7.84.10 phpMyAdmin

phpMyAdmin can be accessed at the `/phpmyadmin` path of the app. It uses basic auth through a htpasswd file and is pre-setup with an admin account and a generated password. The password can be found in the `phpmyadmin_login.txt` file, alongside with details how to managed more users.

If access does not work anymore, simply remove the file `.phpmyadminauth` and restart the app. This will generate new phpMyAdmin credentials.

Disabling phpMyAdmin

It is good security practice to disable phpMyAdmin once you have finished using it. To disable it, edit `/app/data/apache/app.conf` using the File Manager and comment out the following line:

```
    # This line can be commented out, if you do no require PHPMyAdmin Access
    # Include "/app/code/apache/phpmyadmin.conf"
```

Be sure to restart the app after making the above change.

7.84.11 Email

On Cloudron, Email credentials are exposed as environment variables to the app.

The exposed environment variables are:

```
CLOUDRON_MAIL_SMTP_SERVER   # SMTP server
CLOUDRON_MAIL_SMTP_PORT         # SMTP server port
CLOUDRON_MAIL_SMTPS_PORT    # SMTPS server port. This is mostly for legacy apps
CLOUDRON_MAIL_SMTP_USERNAME     # Username
CLOUDRON_MAIL_SMTP_PASSWORD     # Password
CLOUDRON_MAIL_FROM          # The MAIL FROM. If you want to change this, see [this](../apps.md#mail-from-address)
CLOUDRON_MAIL_DOMAIN        # The mail domain
```

You can use `getenv()` to get the values of the above environment variables in code. The raw values can be obtained using the File Manager inside the file `/app/data/credentials.txt`.

> ⚠ **The built-in PHP mail() function does not work**
>
> It uses the local sendmail binary, which is **not configured** on Cloudron.

You can use PHPMailer to send emails (installed using `composer require phpmailer/phpmailer`):

```
<?php
//Import PHPMailer classes into the global namespace
//These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader
require 'vendor/autoload.php';

//Create an instance; passing `true` enables exceptions
$mail = new PHPMailer(true);

try {
    //Server settings
    $mail->SMTPDebug = SMTP::DEBUG_SERVER;                      //Enable verbose debug output
    $mail->isSMTP();                                            //Send using SMTP
    $mail->Host       = getenv('CLOUDRON_MAIL_SMTP_SERVER');    //Set the SMTP server to send through
    $mail->SMTPAuth   = true;                                   //Enable SMTP authentication
    $mail->Username   = getenv('CLOUDRON_MAIL_SMTP_USERNAME');  //SMTP username
    $mail->Password   = getenv('CLOUDRON_MAIL_SMTP_PASSWORD');  //SMTP password
    $mail->SMTPSecure = '';
    $mail->Port       = getenv('CLOUDRON_MAIL_SMTP_PORT');

    //Recipients
    $mail->setFrom(getenv('CLOUDRON_MAIL_FROM'), 'Mailer');
    $mail->addAddress('test@cloudron.io', 'Cloudron Test');     //Add a recipient

    //Content
    $mail->isHTML(true);                                  //Set email format to HTML
    $mail->Subject = 'Here is the subject';
    $mail->Body    = 'This is the HTML message body <b>in bold!</b>';
    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

    $mail->send();
    echo 'Message has been sent';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

7.84.12 Redis

On Cloudron, Redis credentials are exposed as environment variables to the app.

The exposed environment variables are:

```
CLOUDRON_REDIS_URL          # redis URL of the form redis://username:password@host:port
CLOUDRON_REDIS_HOST             # redis hostname
CLOUDRON_REDIS_PORT     6379    # redis port
CLOUDRON_REDIS_PASSWORD      # redis password
```

You can use `getenv()` to get the values of the above environment variables in code. The raw values can be obtained using the File Manager inside the file `/app/data/credentials.txt`

7.84.13 LDAP

On Cloudron, LDAP credentials are exposed as environment variables to the app.

The exposed environment variables are:

```
CLOUDRON_LDAP_SERVER=                                # ldap server IP
CLOUDRON_LDAP_HOST=                                  # ldap server IP (same as above)
CLOUDRON_LDAP_PORT=                                  # ldap server port
CLOUDRON_LDAP_URL=                                   # ldap url of the form ldap://ip:port
CLOUDRON_LDAP_USERS_BASE_DN=                         # ldap users base dn of the form ou=users,dc=cloudron
CLOUDRON_LDAP_GROUPS_BASE_DN=                        # ldap groups base dn of the form ou=groups,dc=cloudron
CLOUDRON_LDAP_BIND_DN=                               # DN to perform LDAP requests
CLOUDRON_LDAP_BIND_PASSWORD=                         # Password to perform LDAP requests
```

To protect the site with basic authentication via LDAP, use the following for apache config:

```
    <Directory /app/data/public>
        Options +FollowSymLinks

        AllowOverride None
        Require valid-user
        AuthName "Cloudron LDAP Authentication"
        AuthBasicProvider ldap
        AuthType Basic
        AuthLDAPURL ${CLOUDRON_LDAP_URL}/${CLOUDRON_LDAP_USERS_BASE_DN}?username?sub?(username=*)
        AuthLDAPBindDN ${CLOUDRON_LDAP_BIND_DN}
        AuthLDAPBindPassword ${CLOUDRON_LDAP_BIND_PASSWORD}
    </Directory>
```

7.84.14 Custom Startup Script

A custom startup script can be placed at `/app/data/run.sh`. For example,

```
#!/bin/bash

echo "This script is called before the app starts"

# create symlinks
rm -rf /app/data/var/cache
mkdir -p /run/cache
ln -sf /run/cache /app/data/var/cache
```

7.84.15 Composer

`composer`, `npm` and other common tools are installed in from the Cloudron base app image. To run these tools, first switch to the `www-data` user (most of them should not be run as root).

```
su - www-data
cd /app/data/public           # this is where PHP code resides
composer require drush/drush

npm install
```

Memory limit

The LAMP app runs with 256MB ram by default which is not enough for Composer and possibly others. If you see a `Killed` error message after a run, increase the memory limit of the app to 1GB.

7.84.16 Laravel

To run Laravel apps, see this article.

7.84.17 Reverse proxy setup

If you want to run for example a custom WordPress within this app, please note that the code will run behind a nginx proxy. Apps like WordPress require some code in `wp-config.php` to handle such a setup:

```
/*
 http://cmanios.wordpress.com/2014/04/12/nginx-https-reverse-proxy-to-wordpress-with-apache-http-and-different-port/
 http://wordpress.org/support/topic/compatibility-with-wordpress-behind-a-reverse-proxy
 https://wordpress.org/support/topic/wp_home-and-wp_siteurl
 */
// If WordPress is behind reverse proxy which proxies https to http
if (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) {
    $_SERVER['HTTP_HOST'] = $_SERVER['HTTP_X_FORWARDED_HOST'];

    if ($_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https')
        $_SERVER['HTTPS']='on';
}
```

7.84.18 Health check

The LAMP app expects a 2xx response from the '/' path. If your app is completely protected, then the healthcheck logic will mark your app as `not responding` instead of `running`.

You can work around this by adding the following in `/app/data/public/.htaccess`:

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} CloudronHealth
RewriteRule ^ - [R=200]
```

Alternately, add something like below in the app's config.php or index.php:

```
if ($_SERVER["REMOTE_ADDR"] == '172.18.0.1') {
    echo "Cloudron healthcheck reponse";
    exit;
}
```

7.85 Language Tool App

7.85.1 About

LanguageTool’s multilingual grammar, style, and spell checker is used by millions of people around the world

- Questions? Ask in the Cloudron Forum - Language Tool
- Language Tool Website
- Language Tool issue tracker

7.85.2 Security

By default, the app's API is public. To avoid misuse, it's best to change the URL to some secret. To do so:

- Edit `/app/data/env` using the File manager
- Change `API_PATH_PREFIX` to some secret. You can generate a secret using `pwgen -1s 6` .
- Restart the app.

In the apps, use `https://languagetool.domain.com/<secret>/v2` as the URL instead of the default `https://languagetool.domain.com/v2` .

To test, if `hushhush` is the secret:

```
curl -d "text=Hello world" -d "language=de-DE" https://languagetool.domain.com/hushhush/v2/check
```

Use alphanumerals for the secret

The secret is part of the URL. For this reason, keep it simple and use only alphanumerals for the secret.

7.85.3 n-grams

LanguageTool can optionally make use of large n-gram data sets to detect errors with words that are often confused, like their and there.

The n-gram data set is huge and thus not part of the LT download. The install path is the first argument for the download command below. The default is `/app/data/ngrams` but you can use e.g. a volume path if you don't want ngrams to be part of your backup.

To download the n-gram data set of a language:

- Open a Web Terminal and run the command below. Replace the argument `/app/data/ngrams` with your desired path (e.g. a volume) if you don't want ngrams to be part of your backup .

```
# /app/pkg/install-ngrams.sh /app/data/ngrams en es
==> Installing en ngram dataset from https://languagetool.org/download/ngram-data/ngrams-en-20150817.zip
/tmp/en.zip 100%[================================================================>]   8.35G   112MB/s    in 77s
==> Unpacking en ngram dataset
Archive:  /tmp/en.zip
   creating: en/
   creating: en/3grams/
  inflating: en/3grams/_1e4.fdt
  inflating: en/3grams/_1e4.si
 extracting: en/3grams/_1e4.nvd

...

  inflating: en/1grams/_1p.si
==> en ngram dataset has been installed.
==> Done
```

```
Set NGRAM_DATASET_PATH=/app/data/ngrams in /app/data/env and restart the app.
```

- Open `/app/data/env` using the file manager and check that `NGRAM_DATASET_PATH` is set to the correct path.
- Restart the app.

> ⚠ **Large data set**
>
> n-gram data sets are large. For example, the `en` is around 14GB and `de` is 3GB.

7.86 Leantime App

7.86.1 About

Leantime is a strategic open source project management system for innovative companies and teams looking to go from start to finish.

- Questions? Ask in the Cloudron Forum - Leantime
- Leantime Website
- Leantime issue tracker

7.87 Translate App

7.87.1 About

Free and Open Source Machine Translation API. Self-hosted, offline capable and easy to setup.

- Questions? Ask in the Cloudron Forum - Translate

This package is proudly built from LibreTranslate's open source software.

No affiliation or endorsement

Please note that Cloudron has no affiliation with LibreTranslate and this package is not endorsed by them.

Trademark requirements

Please conform to Trademark requirements, when using this app.

7.87.2 Languages

By default the app comes with EN and DE installed. If more languages are wanted, update `LT_LANGUAGE_MODELS` variable in `/app/data/env.sh`, and restart the app. e.g.

```
export LT_LANGUAGE_MODELS=en,de,fr
```

To remove languages, update `LT_LANGUAGE_MODELS` variable appropriately, remove `/app/data/argos-translate` folder and restart the app.

7.87.3 User Management

The app has no user management of it's own. By default, anyone (without a token) can make 30 requests per minute. You can change this in `/app/data/env.sh`.

7.87.4 Custom config

Custom environment variables can be set in `/app/data/env.sh` using the File manager.

7.87.5 API

By default, the API can be used without a key to make 30 requests per minute:

```
curl -d "q=cloudron%20ist%20wunderbar&source=de&target=en" -X POST https://translate.domain.com/translate
```

To lock the app and enforce use of a key, set the below in `/app/data/env.sh`:

```
export LT_REQUIRE_API_KEY_ORIGIN=true
export LT_REQ_LIMIT=0
```

Create Key

Use the ltmanage command to create an API key. On Cloudron, LibreTranslate runs as the `cloudron` user. Run the following command using a Web Terminal to create an API with a limit of 120 requests per minute:

```
source /app/code/venv/bin/activate
ltmanage keys --api-keys-db-path /app/data/db/api_keys.db add 120
```

The key can be used as follows:

```
curl -d "q=cloudron%20ist%20wunderbar&api_key=f67d5afe-aa14-4d37-90f7-f7d9636b721e&source=de&target=en" -X POST https://translate.domain.com/translate
```

Existing API keys can also be listed with

```
source /app/code/venv/bin/activate
ltmanage keys --api-keys-db-path /app/data/db/api_keys.db
```

and then also revoked with

```
source /app/code/venv/bin/activate
ltmanage keys --api-keys-db-path /app/data/db/api_keys.db remove <key>
```

7.88 LibreChat App

7.88.1 About

Enhanced ChatGPT Clone: Features Agents, DeepSeek, Anthropic, AWS, OpenAI, Assistants API, Azure, Groq, o1, GPT-4o, Mistral, OpenRouter, Vertex AI, Gemini, Artifacts, AI model switching, message search, Code Interpreter, langchain, DALL-E-3, OpenAPI Actions, Functions, Secure Multi-User Auth, Presets, open-source for self-hosting.

- Questions? Ask in the Cloudron Forum - LibreChat
- LibreChat Website
- LibreChat issue tracker

7.88.2 Custom Config

Custom configuration can be added in `/app/data/env` and `/app/data/librechat.yaml` files using the File Manager. Be sure to restart the app after making any changes.

Refer to LibreChat documentation

7.88.3 Disable Registration

To disable user registration you should set `ALLOW_REGISTRATION=false` in `/app/data/env` and restart the app.

7.89 LimeSurvey App

7.89.1 About

LimeSurvey is the most versatile online survey tool for newcomers and professionals.

- Questions? Ask in the Cloudron Forum - LimeSurvey
- LimeSurvey Website
- LimeSurvey forum

7.89.2 Command Line Tool

The CLI tool can run using the Web Terminal as follows:

```
sudo -E -u www-data php /app/code/application/commands/console.php
```

7.89.3 Templates

Questionnaire templates can be downloaded from the LimeSurvey site here.

7.90 Linkding App

7.90.1 About

Linkding is a bookmark manager that you can host yourself. It's designed be to be minimal, fast, and easy to set up.

- Questions? Ask in the Cloudron Forum - Linkding
- Linkding GitHub
- Linkding issue tracker

7.90.2 Customization

Linkding has various options.

To customize, edit `/app/data/env.sh` using the File manager. Be sure to restart the app after making any changes.

7.91 Linkwarden App

7.91.1 About

Linkwarden is a self-hosted, open-source collaborative bookmark manager to collect, organize and archive webpages.

- Questions? Ask in the Cloudron Forum - Linkwarden
- Linkwarden Website
- Linkwarden GitHub
- Linkwarden issue tracker

7.92 Listmonk App

7.92.1 About

Listmonk is a standalone, self-hosted, newsletter and mailing list manager. It is fast, feature-rich, and packed into a single binary.

- Questions? Ask in the Cloudron Forum - Listmonk
- Listmonk Website
- Listmonk issue tracker

7.92.2 Timezone

To set the timezone, edit the `TZ` variable inside `/app/data/env.sh` using the File manager.

Be sure to restart the app, after making any changes.

7.92.3 Static templates

System templates are used for rendering public user facing pages such as the subscription management page, and in automatically generated system e-mails such as the opt-in confirmation e-mail.

The original template files, matching the upstream version, are located at `/app/pkg/static.template`.

You can make a copy of those into `/app/data/static` using the Web terminal to run the following command:

```
cp -rf /app/pkg/static.template/* /app/data/static/
```

Then restart the app to make it pickup the changes. A restart is required everytime files in that folder are changed.

> ⚠ **Version incompatibilities**
>
> Since the template files might depend on specific listmonk versions, any app update might break due to incompatibilites. If custom templates are used, it may be good to disable automatic app updates and check functionality after each manually applied update.

7.92.4 i18n

Additional languages can be uploaded to `/app/data/i18n` using the File manager.

7.93 Loomio App

7.93.1 About

Loomio is a collaborative decision making tool.

- Questions? Ask in the Cloudron Forum - Loomio
- Loomio Website
- Loomio issue tracker

7.93.2 Custom config

Custom configuration can be added by editing `/app/data/env.sh` using the File manager.

See this file for a list of supported options.

Be sure to restart the app after making any changes.

7.93.3 Registration

By default, registration is enabled and new users can create groups. You can adjust this using the following variables in `/app/data/env.sh`:

```
export FEATURES_DISABLE_CREATE_USER=1     # users must be invited
export FEATURES_DISABLE_CREATE_GROUP=1    # users cannot create groups
export FEATURES_DISABLE_PUBLIC_GROUPS=1   # disable /explore
export FEATURES_DISABLE_HELP_LINK=1       # disable the help link
export FEATURES_DISABLE_EMAIL_LOGIN=1     # Disable login via email (usually when you have enabled SSO of some kind)
```

7.93.4 Rails Console

The rails console can be access using the Web Terminal. The redis and database environment variables must be set as below:

```
# export DATABASE_URL=postgresql://${CLOUDRON_POSTGRESQL_USERNAME}:${CLOUDRON_POSTGRESQL_PASSWORD}@postgresql/${CLOUDRON_POSTGRESQL_DATABASE}
# export REDIS_URL=redis://${CLOUDRON_REDIS_HOST}
# source /app/data/env.sh
# rails console
Loading production environment (Rails 7.0.7.2)
irb(main):001:0>
```

7.94 Lychee App

7.94.1 About

Lychee is a free photo-management tool, which runs on your server or web-space

- Questions? Ask in the Cloudron Forum - Lychee
- Lychee Website

7.94.2 User management

While lychee has a UI to change the password, there is no UI to reset the password. If you forget the password, reset the password of the admin user, use the Web terminal.

```
# sudo -E -u www-data php artisan lychee:reset_admin
```

After running the above command, just logout from Lychee and it will ask for creating the admin user again. Note that existing data is not lost when resetting admin credentials.

7.94.3 Using an existing folder structure

Lychee has it's own folder-structure and database. You have to re-upload or re-import all your photos to use them.

7.95 Mailtrain App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.95.1 About

Mailtrain is a self hosted newsletter app.

- Questions? Ask in the Cloudron Forum - Mailtrain
- Mailtrain Website
- Mailtrain issue tracker

7.95.2 VERP handling

VERP is a feature where bounces are sent back to special addresses. Mailtrain supports VERP and can be configured to process these bounces and automatically unsubscribe addresses that are bouncing.

7.95.3 Changing Email "from" Address

Cloudron's email server does not allow apps to send emails with arbitrary FROM addresses. This is a simple security measure that prevents apps from unintentionally sending out emails that they are not supposed to send.

Cloudron assigns `location.app@domain` address by default to apps. To change this email address, follow the instructions here.

7.95.4 External mail server

To revert from an external email server setup to the Cloudron setup, set the SMTP Hostname to `localhost` and then restart the app. This will inject the Cloudrons SMTP credentials automatically.

7.95.5 Customizations

Additional settings can be added to `/app/data/production.toml` using the File manager.

7.96 Mastodon App

7.96.1 About

Mastodon is an open source decentralized social network - by the people for the people. Join the federation and take back control of your social media!

- Questions? Ask in the Cloudron Forum - Mastodon
- Mastodon Website
- Mastodon issue tracker

7.96.2 Admin

Mastodon 4 introduced customizable roles.

To make a user an administrator, use the Web Terminal and run the following command:

```
/app/code/bin/tootctl accounts modify <username> --role Owner
```

Note that there is also a default `Admin` role. See this PR for more information.

Once made an owner/admin, you will `Administration` under `Preferences` (at `/admin/dashboard`).

7.96.3 Adding users

When used with Cloudron authentication, simply add new users to the Cloudron dashboard. As of this writing, mixing Cloudron users and external users does not work well (See this issue).

Without Cloudron authentication, new users can be added and approved using the CLI:

```
/app/code/bin/tootctl accounts create testusername --email=test@cloudron.io --approve
```

7.96.4 Username restriction

Usernames can contain only letters, numbers and underscores. On Cloudron, usernames can be created with hyphen and dot. Such usernames cannot login to Mastodon and will get a "HTTP ERROR 422" when attempting to login. Please see the upstream bugreport for more information.

7.96.5 Registration

Registration is closed by default. To enable, login to Mastodon as an admin and change the "Registration mode" under `Administration` -> `Site Settings`.

> ⚠ **Does not work well with Cloudron user management**
>
> Various features like password reset, re-invitation etc do not work well because of upstream bug.

7.96.6 Federation

Cloudron will setup Mastodon accounts to be of the form `username@social.example.org` when you install Mastodon at `social.example.org`. This domain is called the `LOCAL_DOMAIN` in Mastodon terminology.

Changing the `LOCAL_DOMAIN` will let you have handles as `username@example.org` even when installed at `social.example.org` Changing the `LOCAL_DOMAIN` is not recommended since it is complicated and in most cases unnecessary. This is because Mastodon account names are not intended to be remembered like usernames (it's not like email where you can start following another account). Instead, users usually visit a website and click the 'Follow' button.

If you decide to not change the `LOCAL_DOMAIN`, no further configuration is required and your Mastodon instance is already set up for federation.

Changing LOCAL_DOMAIN

You can change the account domain name by using the File Manager and changing `LOCAL_DOMAIN` in `/app/data/env.production`. After that, you have to configure `LOCAL_DOMAIN`'s web server to serve up `.well-known/host-meta` query.

If `LOCAL_DOMAIN` is an app on Cloudron, you can use Cloudron's Well Known URI support. Go to the `Domains` view and set the Mastodon domain in the `Advanced` settings:

If the `LOCAL_DOMAIN` is **NOT** hosted on Cloudron, you must figure out a suitable way to serve up the well-known documents. Here are some hints:

- For WordPress, you can setup a redirect using Redirection plugin
- For Ghost,you can add a redirects.json
- For Surfer, simply upload the XML above into `.well-known/host-meta`.
- For anything else, setup nginx config as follows:

```
location = /.well-known/host-meta {
       return 301 https://social.example.org$request_uri;
}
```

7.96.7 Following users

To follow external users, visit their mastodon account and click on 'Follow'. This will popup a window asking your mastodon identity (which will be `username@LOCAL_DOMAIN`).

If you have an existing account on another server, you can bring those connections with you to your own server. For this, go to Settings -> Data Export and download your following list as a CSV file, and finally on your own server, you go to Settings -> Import and upload that file.

7.96.8 Scaling

Various parameters in Puma, Streaming and Sidekiq can be fine-tuned for concurrency. To change the parameters, edit `/app/data/config.sh` using the File manager and restart the app. See scaling docs for more information.

Sample configuration for `/app/data/config.sh` (from here):

```
# Puma
export WEB_CONCURRENCY=4 # number of worker processes
export MAX_THREADS=5 # the number of threads per process

# Streaming API
export STREAMING_CLUSTER_NUM=2 # number of worker processes

# Sidekiq
export SIDEKIQ_THREADS=4
export DB_POOL=25 # must be at least the same as the number of threads
```

7.96.9 S3 for media storage

Media storage can consume quite a bit of space. To move it to S3 or S3 compatible provider, edit `/app/data/env.production` using the File manager and add something like this:

```
S3_ENABLED=true
S3_BUCKET=bucket-name
AWS_ACCESS_KEY_ID=<key_id>
AWS_SECRET_ACCESS_KEY=<secret_key>
S3_REGION=<region>
```

```
S3_PROTOCOL=https
S3_HOSTNAME=<hostname>
```

See this thread for more information.

7.96.10 Cache retention days

The app will clean various federation caches regularily. Depending on the usage of an instance, the days cached resources should be keept can be configured via `CACHE_RETENTION_DAYS` in the `cache-env.sh` file.

By default all cached assets are kept for 2 days.

7.97 Matomo App

7.97.1 About

Matomo is Google Analytics alternative that protects your data and your customers' privacy

- Questions? Ask in the Cloudron Forum - Matomo
- Matomo Website
- Matomo forum
- Matomo issue tracker

7.97.2 Installing plugins

Matomo plugins can be installed from the Marketplace view. Only the Matomo super-user can install plugins (Matomo admins cannot install plugins).

Plugins can also be installed by manually extracting them into `/app/data/plugins` using the File manager.

7.97.3 Debugging the tracker

To debug the tracker, add the following to `config.ini.php` using the file manager:

```
[Tracker]
debug = 1
debug_on_demand = 1
```

You can then debug the request using the following curl request (be sure to fix up the idsite):

```
curl -X POST 'https://matomo.example.com/matomo.php?idsite=1&rec=1&bots=1&debug=1
```

7.97.4 System check warning

LOAD DATA INFILE

The LOAD DATA INFILE warning can be ignored. More information is available in matomo forum - here and here. There is no loss of functionality by ignoring the warning.

From Cloudron point of view, the LOAD DATA INFILE feature requires MySQL and Matomo to share the same file system. This is a security risk and not desirable. On Cloudron, apps and database containers are isolated and cannot access each other's file system.

7.98 Mattermost App

7.98.1 About

Mattermost is an open source, self-hosted Slack-alternative.

- Questions? Ask in the Cloudron Forum - Mattermost
- Mattermost Website
- Mattermost forum
- Mattermost docs
- Mattermost issue tracker

7.98.2 Config

The config file is located at `/app/data/config.json` and can be edited using the File manager. Be sure to restart the app after making changes to the config file.

7.98.3 Command Line Tool

The Mattermost CLI tool can be used to administer user and team management tasks.

Starting Mattermost 6.0, the CLI tool has to installed locally on your Mac/PC. See the CLI docs for downloading the binaries.

Once installed, you can run mmctl like so:

```
$ /app/code/mmctl auth login https://mattermost.cloudron.space --name myserver --username admin
Password:

  credentials for "myserver": "admin@https://mattermost.cloudron.space" stored
```

You can then run commands like this `mmctl team list` etc.

7.98.4 Migrate External Instance

In you want to migrate your existing non-Cloudron mattermost installation to Cloudron, use the bulk export/import functionality of mmctl.

Caveats:

- plugins, integrations, webhook, custom message formats, system settings have to be setup again
- User passwords are not migrated. All users have to reset their password in the new instance.

Export External Instance

Export data from the old installation using `mmctl`. `mmctl` can be installed on your PC/laptop.

```
$ mmctl auth login https://mmold.instance
$ mmctl export create
Export process job successfully created, ID: 8qxfhwmrejnkij3rbqzpo1q7wy
$ mmctl export list
8qxfhwmrejnkij3rbqzpo1q7wy_export.zip
$ mmctl export download 8qxfhwmrejnkij3rbqzpo1q7wy_export.zip
```

Import to Cloudron Instance

Install mattermost on Cloudron. Because we don't have any users yet to login and authenticate, we must use the local socket to authenticate with mattermost.

- Change `EnableLocalMode` to true in `/app/data/config.json` and `LocalModeSocketLocation` to `/tmp/mattermost_local.socket` using the File manager and restart the app.
- Open a Web Terminal .
- Upload the export file to `/tmp'.
- Import using the mmctl binary.

```
$ cd /tmp
$ curl -L https://releases.mattermost.com/mmctl/v9.8.0/linux_amd64.tar | tar xvf -
$ export MMCTL_LOCAL_SOCKET_PATH=/tmp/mattermost_local.socket
$ ./mmctl --local import process --bypass-upload /tmp/8qxfhwmrejnkij3rbqzpo1q7wy_export.zip
Import process job successfully created, ID: ksy4gtmxqpbgumcw7hd84dcr6r
$ ./mmctl --local import job list
  ID: ksy4gtmxqpbgumcw7hd84dcr6r
  Status: success
  Created: 2024-05-31 11:33:09 +0000 UTC
  Started: 2024-05-31 11:33:19 +0000 UTC
  Data: map[extract_content:true import_file:/tmp/8qxfhwmrejnkij3rbqzpo1q7wy_export.zip local_mode:true]
```

- Change `EnableLocalMode` to false in `/app/data/config.json` and restart the app.

7.98.5 Enterprise

To switch to the Enterprise version of Mattermost, edit `/app/data/edition.ini` using the File manager and change `edition=enterprise`.

Be sure to restart the app after making any changes.

7.98.6 Migrate to Postgres

Mattermost has decided to deprecate MySQL support and focus on PostgreSQL support. For this reason, we have made a new package that uses PostgreSQL with appstore id `org.mattermost.cloudronapp2` and deprecated the old MySQL package (appstore id `org.mattermost.cloudronapp`).

There are two ways to migrate: Database migration and Export/Import.

Database Migration

For this, use the upstream guide as the reference.

> **Use forum for help**
>
> Please note that we are not experts on Mattermost or it's schema. Database migration is a complex topic and it is unfortunate that upstream has decided to deprecate support for MySQL. For this reason, please use the Cloudron Forum or the upstream Mattermost forum to ask for migration help.

`pgloader` tool is used to migrate data from MySQL to PostgreSQL. The migration config for pgloader is located at `/app/pkg/migration.load` . This file is copied from the upstream docs. Double check the content of this file before and the upstream doc before continuing. The main difference is that the upstream docs uses a dot in variables names and this breaks pgloader's mustache templating.

- Install a fresh instance of Mattermost. New instances use PostgreSQL by default. You don't need to stop or delete your existing MySQL installation.
- In the new instance, open a Web Terminal.
- Create a folder `/tmp/migration`

```
mkdir -p /tmp/migration
```

- Create a file named `/tmp/migration/context.ini` and fill it up like below. You can get the MySQL connection information of your old installation by running `printenv | grep CLOUDRON_MYSQL` in the old instance. You can get the PostgreSQL connection information of your new installation by running `printenv | grep CLOUDRON_POSTGRESQL` in your new instance.

```
touch /tmp/migration/context.ini
```

```
[pgloader]
mysql_user=f1b2b6873b4dad41
mysql_password=f1c879c56c68abb03f77ef26e20b5898fc9bfeb2f87ae358
mysql_address=mysql:3306
source_schema=f1b2b6873b4dad41

pg_user=userbec36139a3c04f7a97394a5348d4ea87
pg_password=d307edc09a68a3a8cd6b7d4a614a4c5bcb2dbe98bef0b7c86ceab55a97691f15c9c246e82e01b2e9f1db44d8d9de0c58cad3de13c169fafcaace2c7b289c6033
postgres_address=postgresql:5432
target_schema=dbbec36139a3c04f7a97394a5348d4ea87
```

- Download the latest version of the migration-assist in `/tmp/migration`

```
# as of writing this doc version v0.2 is latest, please check for a newer version
cd /tmp/migration/
wget https://github.com/mattermost/migration-assist/releases/download/v0.2/migration-assist-Linux-x86_64.tar.gz
tar xzf migration-assist-Linux-x86_64.tar.gz
rm migration-assist-Linux-x86_64.tar.gz
```

- Run the migration:

```
$ pgloader --context=/tmp/migration/context.ini /app/pkg/migration.load > /app/data/migration.log
```

- If the above works (no error output) check the logfile in `/app/data/migration.log` for anything suspicious, include this log in the Forum when reporting errors
- migrate other directories like `files`, `plugins` from the old instance to the new instance. You can do this using the File Manager's download and upload functionality. Plugin specific migration is covered in upstream docs.
- Restore full-text indexes with the `migration-assist` by getting the PostgreSQL connection URL from `printenv CLOUDRON_POSTGRESQL_URL` and running the `migration-assist` as following

```
# migration-assist postgres post-migrate "<POSTGRES_DSN>"
# The <POSTGRES_DSN> is the output of printenv CLOUDRON_POSTGRESQL_URL
# WATCHOUT! You will have to add ?sslmode=disable at the end of the POSTGRES_DSN
migration-assist postgres post-migrate "postgres://userPLACEHOLDER:PLACEHOLDERPASSWORD@postgresql/dbPLACEHOLDER?sslmode=disable"
```

- Restart mattermost and test the whole installation before moving your new installation to the desired location.

Bulk Export/Import

This approach is more portable and safer.

Known limitations:

- plugins, integrations, webhook, custom message formats, system settings have to be setup again
- User passwords are not migrated. All users have to reset their password in the new instance.

Steps to migrate are very similar to Migrating an External Instance:

- Export the existing MySQL installation using the same procedure as Export External Instance.
- Install a fresh instance of Mattermost. New instances use PostgreSQL by default. You don't need to stop or delete your existing MySQL installation.
- In the new instance, import using the same procedure as Import to Cloudron Instance.
- Note that you have to reset the password to login in the new instance. Once you are satisfied that data is intact, you can move over the app to your existing location. We recommend stopping the MySQL instance and moving it to another location and keeping it around for some time.

7.99 Mautic App

7.99.1 About

Mautic a free and opensource marketing automation tool.

- Questions? Ask in the Cloudron Forum - Mautic
- Mautic Website
- Mautic forum
- Mautic issue tracker

7.99.2 System cron jobs

Mautic often requires cron jobs to be triggered manually. The package comes with a a default cronjob template. This is saved in `/app/data/crontab.system`.

You can change the schedule and arguments of the cronjobs as desired using the File manager. Be sure to restart the app after making any changes.

To run a cron job manuall, open a Web terminal and run the job manually like so:

```
sudo -E -u www-data php /app/code/bin/console mautic:segments:update
```

7.99.3 Custom cron jobs

For cron support, app custom commands in the app's cron section.

The crontab contains a line like:

```
0,15,30,45 * * * *  sudo -E -u www-data php /app/code/app/console mautic:integration:fetchleads --integration=Hubspot
```

7.99.4 Mailer

Mautic 5 supports sending emails via Symfony mailer. The following mailer packages are pre-installed:

- symfony/amazon-mailer
- symfony/mailgun-mailer
- symfony/postmark-mailer
- symfony/sendgrid-mailer
- symfony/mailjet-mailer

⚠ **Disable email auto-configuration**

On Cloudron, mail settings are managed by the platform. By default, mautic will send messages via the built-in Cloudron Email Server. If you want to change the mail configuration to another mailer, you can either set it at the server level as an outbound relay or at the app level by disabling email configuration.

7.99.5 Clearing cache

To clear the cache, you can either delete `/run/mautic/var/cache` or you can run the following command using the Web terminal:

```
sudo -E -u www-data php /app/code/bin/console cache:clear
```

7.99.6 Admin

Admin password reset

To reset the admin user password, run the following command using the Web terminal:

```
set-admin-password <newstrongpassword>
```

Admin email reset

To set the admin user email, run the following command using the Web terminal:

```
set-admin-email <email@example.com>
```

7.100 Mealie App

7.100.1 About

Mealie is a self hosted recipe manager and meal planner with a RestAPI backend and a reactive frontend application built in Vue for a pleasant user experience for the whole family. Easily add recipes into your database by providing the url and mealie will automatically import the relevant data or add a family recipe with the UI editor

- Questions? Ask in the Cloudron Forum - Mealie
- Mealie Website
- Mealie issue tracker

7.101 MediaWiki App

7.101.1 About

MediaWiki is a collaboration and documentation platform brought to you by a vibrant community.

- Questions? Ask in the Cloudron Forum - MediaWiki
- MediaWiki Website

7.101.2 Admin

Admin operations can be performed by visiting `Tools` menu on the left > `SpecialPages`.

To change or reset the admin password, open a Web Terminal and run:

```
php maintenance/run.php changePassword --user=administrator --password=supersecretpassword
```

7.101.3 Access Control

Default setup

When using Cloudron SSO, the wiki is setup to be editable only by Cloudron users. Anonymous users can read all pages.

When not using Cloudron SSO, the wiki is setup to be editable by users with a registered wiki account.

Changing permissions

Use the File manager to edit the values in `/app/data/LocalSettings.php`.

Here are some commonly requested settings:

- To disable read access for anonymous users, add the following line:
  $wgGroupPermissions['*']['read'] = false;
- To allow write access to anonymous users, add the following line:
  $wgGroupPermissions['*']['edit'] = true;
- To disable email confirmation before new users are allowed to edit files:
  $wgEmailConfirmToEdit = false;
- To disallow account creation and remove the 'Create account' link:
  $wgGroupPermissions['*']['createaccount'] = false;

7.101.4 Custom icon

To set a custom icon, use the File manager and upload a file named `/app/data/images/wiki.png`.

7.101.5 Extensions

Installing

MediaWiki extensions can be installed as follows:

- Use the File manager to upload the tarball and extract the package under `/app/data/extensions`.
- Change the ownership of the newly uploaded directory to `www-data`
- Load the skin in `/app/data/LocalSettings.php` by adding this line:

```
        wfLoadExtension( '<extension-name>' );
```

* Additional extension settings may be set in `/app/data/LocalSettings.php`

Suppressing skins

To suppress one or more skins add the following line to `/app/data/LocalSettings.php`:

```
    $wgSkipSkins = array( "cologneblue", "monobook" );
```

7.101.6 Skins

Installing

MediaWiki skins can be installed as follows:

- Use the File manager to upload the tarball and extract the package under `/app/data/skins`.
- Change the ownership of the newly uploaded directory to `www-data`
- Load the skin in `/app/data/LocalSettings.php` by adding this line:

```
        wfLoadSkin( '<skin-name>' );
```

- The default skin for new users can be changed by adding this line to `/app/data/LocalSettings.php`:

```
        $wgDefaultSkin = '<skin-name>';
```

Suppressing skins

To suppress one or more skins add the following line to `/app/data/LocalSettings.php`:

```
    $wgSkipSkins = array( "cologneblue", "monobook" );
```

7.101.7 Exporting a Wiki

To export in XML format, use the dumpBackup script as part of MediaWiki installation. Open a Web terminal and run the following commands:

```
# cd /app/code/maintenance
# php dumpBackup.php --full > /tmp/dump.xml
```

You can download the dump using the download button at the top of the terminal and entering `/tmp/dump.xml`.

7.101.8 Importing a Wiki

To import in XML format, use the importDump script as part of MediaWiki installation.

Open a Web terminal:

- Upload the XML using the Upload button
- Run the following commands

```
# cd /app/code/maintenance
# php importDump.php < /tmp/dump.xml
You might want to run rebuildrecentchanges.php to regenerate RecentChanges,
and initSiteStats.php to update page and revision counts
# php rebuildrecentchanges.php
# php initSiteStats.php
```

When importing a wiki, the Main Page might still appear without the correct content. You can fix this by going to the Main Page's History and undoing the latest change. Please note that the administrator account needs a valid email for this to work (preferences -> confirm email address).

7.102 Meemo App

7.102.1 About

Meemo simplifies your life by capturing what's on your mind. From short lists, ideas, links, inspiration, to lengthy research, pictures and memories.

- Questions? Ask in the Cloudron Forum - Meemo
- Meemo Website

7.102.2 Chrome extension

The Meemo chrome extension allows to add the current page with notes into Meemo.

7.103 Memos App

7.103.1 About

Open Source, Self-hosted, Your Notes, Your Way. Effortlessly craft your impactful content.

- Questions? Ask in the Cloudron Forum - Memos
- Memos Website
- Memos issue tracker

7.104 Metabase App

7.104.1 About

Metabase is the simplest, fastest way to get business intelligence and analytics to everyone in your company.

- Questions? Ask in the Cloudron Forum - Metabase
- Metabase Website
- Metabase forum
- Metabase issue tracker

7.104.2 Custom Config

Custom environment variables can be set in `/app/data/env.sh` using the File manager.

Be sure to restart the app after making any changes

7.104.3 Reset password

To reset the password of an admin with the email `admin@cloudron.local`, open a Web terminal:

```
export MB_DB_TYPE="postgres"
export MB_DB_HOST=${CLOUDRON_POSTGRESQL_HOST}
export MB_DB_PORT=${CLOUDRON_POSTGRESQL_PORT}
export MB_DB_USER=${CLOUDRON_POSTGRESQL_USERNAME}
export MB_DB_PASS=${CLOUDRON_POSTGRESQL_PASSWORD}
export MB_DB_DBNAME=${CLOUDRON_POSTGRESQL_DATABASE}
source /app/data/env.sh
java -jar metabase.jar reset-password admin@cloudron.local
```

This will produce an output that contains the token:

```
...
OK [[[1_63d77453-846f-44ca-939e-7aa0fa09b870]]]
```

Navigate to `https://metabase.example.com/auth/reset_password/1_63d77453-846f-44ca-939e-7aa0fa09b870` to reset the password. Be sure to replace the metabase domain name and the token with your own.

7.105 Minecraft App

7.105.1 About

This app sets up a multiplayer minecraft server.

- Questions? Ask in the Cloudron Forum - Minecraft Server
- Minecraft Server Website

7.105.2 Supported editions

There are 3 different app packages:

- Minecraft Java Edition Server
- Minecraft Java Edition Forge server
- Bedrock/Pocket Edition

7.105.3 Java Edition

Common commands

Please note that you have to run these commands when the user is logged into the app instance from the Cloudron dashboard. The username and password are your Cloudron credentials.

- Whitelist a client - `/whitelist minecraft_username`
- Blacklist a client - `/blacklist minecraft_username`
- Become the server operator - `/op your_minecraft_username`
- Reload server after chaning config files like **server.properties** - `/reload`

RCON

The RCON port for remote configuration tools is active, if the RCON port is enabled in the app instance configuration.

The default password is `changeme1234` so if the instance has RCON port enabled, make sure to set a custom strong password via the File manager in the `server.properties` file:

```
...
rcon.password=yourstrongpassword
...
```

7.105.4 Forge Edition

Mods

Mods can be added by downloading the appropriate .jar files and uploading them to `/app/data/mods` using the File manager. Be sure to restart the app after uploading a mod. More information on adding mods is available here

The list of compatible mods is available here.

7.105.5 Bedrock Edition

Allowlist

The allowlist can be configured in `/app/data/bedrock/allowlist.json` . You can edit this file using the File manager.

Be sure to restart the app after making any changes.

Example:

```
[
{"ignoresPlayerLimit":false,"name":"Player 1","xuid":"################"},
{"ignoresPlayerLimit":false,"name":"Player 2","xuid":"################"},
{"ignoresPlayerLimit":false,"name":"Player 3","xuid":"################"}
]
```

Permissions

The allow list can be configured in `/app/data/bedrock/permissions.json` . You can edit this file using the File manager.

Be sure to restart the app after making any changes.

Example:

```
[
  {
    "permission": "operator",
    "xuid": "XUID1"
  },
  {
    "permission": "operator",
    "xuid": "XUID2"
  },
```

7.106 Miniflux App

7.106.1 About

Miniflux is a minimalist and opinionated feed reader.

- Questions? Ask in the Cloudron Forum - Miniflux
- Miniflux Website
- Miniflux docs
- Miniflux issue tracker

7.106.2 Reset password

To reset the password, open a Web terminal:

```
# export DATABASE_URL="${CLOUDRON_POSTGRESQL_URL}?sslmode=disable"
# /app/code/miniflux -reset-password
```

7.106.3 Custom variables

Custom environment variables can be set in `/app/data/env.sh` using the File manager. Be sure to restart the app after making any changes.

```
export FETCH_YOUTUBE_WATCH_TIME=1
```

See upstream docs for the various configuration options.

7.107 MinIO App

7.107.1 About

Minio is a high performance S3 compatible Object Storage.

- Questions? Ask in the Cloudron Forum - Minio
- Minio Website
- Minio issue tracker

7.107.2 Admin credentials

To change admin credentials, use the File manager to edit the variables `MINIO_ROOT_USER` and `MINIO_ROOT_PASSWORD` in `/app/data/env.sh` .

```
export MINIO_ROOT_USER=superadmin
export MINIO_ROOT_PASSWORD='Secr$et#pass'
```

Be sure to restart the app after making changes.

7.107.3 Domains

MinIO uses two domains:

- Console domain - this domain is for accessing the MinIO console.
- API Domain - this endpoint responds to S3 API requests. This is the domain that you need to put into various configs and tools like s3cmd. Only the domain name is needed, no port should be added.

In the screenshot below, you would use `minio.cloudron.site` as the console domain. You can login and view your files. The `minio-api.cloudron.site` is the API domain which responds to API requests.

7.107.4 Access Keys

Recent version of MinIO removed the ability to manage and generate keys via the UI. The `mc` CLI tool must be used instead. See the CLI section for more information.

To create access keys, you have to set up a user, attach a policy to the user and generate access keys for the user.

- Connect the CLI to your MinIO server

```
    mc alias set myminio https://minio-api.smartserver.io minioadmin minioadmin
```

- Create a user

```
    mc admin user add myminio alice sTr0ngPassword!
```

- Create the bucket policy. Create a file named `policy.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::mybucket",
        "arn:aws:s3:::mybucket/*"
      ]
    }
  ]
}
```

Then, create the policy on the server:

```
mc admin policy create myminio mybucket-fullaccess ./policy.json
```

- Attach the bucket with the user:

```
mc admin policy attach myminio mybucket-fullaccess --user alice
```

- Finally, generate access keys for the user:

```
mc admin accesskey create myminio/ alice
```

7.107.5 Cloudron Backup

Cloudron supports backing up to minio. Backing up a Cloudron to a minio installed in another Cloudron will work fine. However, backing up a Cloudron to a minio installed in the very same Cloudron is not supported.

7.107.6 Custom configuration

Custom config variables can be exported in `/app/data/env.sh`. This file is sourced automatically on startup.

7.107.7 CLI

MinIO supports multiple long term users in addition to default user created during server startup. New users have to added using the CLI tool. You can read the full docs here.

- Install Minio CLI tool . Or use the Binaries
- Configure CLI tool to point to your minio installation

  ```
  mc alias set myminio https://minio.cloudron.club minioadmin minioadmin --api s3v4
  ```

- Create a policy file

  ```
  cat > getonly.json << EOF
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": [
          "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::my-bucketname/*"
        ],
        "Sid": ""
      }
    ]
  }
  EOF
  ```

- Add the policy

```
mc admin policy add myminio getonly getonly.json
```

- Add new user

```
mc admin user add myminio newuser password123
```

- Set policy on user

```
mc admin policy set myminio getonly user=newuser
```

7.108 MiroTalk App

7.108.1 About

Free browser based Real-time video calls. Simple, Secure, Fast. Start your next video call with a single click. No download, plug-in, or login is required. Just get straight to talking, messaging, and sharing your screen.

- Questions? Ask in the Cloudron Forum - MiroTalk
- MiroTalk Website
- MiroTalk issue tracker

The comes in two flavors, one which relies on peer-to-peer (p2p) connections and another with a selective forwarding unit (SFU). Depending on your use-case and networking setup one might work better than the other.

7.108.2 P2P Flavor

Peer-to-peer MiroTalk

Email Alerts

The app can optionally send email alerts to a specified email address. For this set `EMAIL_SEND_TO` in `/app/data/env` and restart the app.

```
...
EMAIL_SEND_TO=alerts@example.com
...
```

7.108.3 SFU Flavor

Selective forwarding unit MiroTalk.

Configuration

Custom config can be added in `/app/data/env` using the File Manager.

Config options are documented here.

Restart the app after making any configuration changes.

7.108.4 BRO Flavor

Broadcasting version of MiroTalk to stream videos, audio and screen from one to many.

7.109 Monica App

7.109.1 About

Monica helps you organize the social interactions with your loved ones.

- Questions? Ask in the Cloudron Forum - MonicaHQ
- MonicaHQ Website
- MonicaHQ issue tracker

7.109.2 Custom config

Custom config can be added to `/app/data/env` using the File manager.

Be sure to restart the app after making any changes.

7.109.3 Multiuser

By default, Monica is setup to be single user.

To enable user registration, open the File manager and add the following variable in `/app/data/env` :

```
APP_DISABLE_SIGNUP=false
```

Restart the app and the login page will show a sign-up link in the login page.

7.109.4 Mobile App

⚠ **Mobile App Deprecated**

The mobile app (Chandler) was deprecated by the upstream project.

To enable the mobile app, open the Web terminal and run the following command:

```
$ php artisan passport:client --password --name="MobileApp"
Password grant client created successfully.
Client ID: 8
Client Secret: Q1gM1DXaMUt8rdvU3MhC4dnxGrV2EdjnBfyj9Sjm
```

Now edit the file `/app/data/env` and search/edit the values:

```
PASSPORT_PERSONAL_ACCESS_CLIENT_ID=
PASSPORT_PERSONAL_ACCESS_CLIENT_SECRET=
```

7.110 Moodle App

7.110.1 About

Moodle is the world's most popular learning management system. Start creating your online learning site in minutes!

- Questions? Ask in the Cloudron Forum - Moodle
- Moodle Website
- Moodle forum

7.110.2 Updates

Moodle is a complex application with a complicated upgrade procedure. It supports over 25 different types of plugins each located in a different location in the source code. While we have automated the upgrade, do not use any more plugins than necessary to reduce update issues. On the same note, do not edit the source code of core moodle since it will be overwritten on an update.

File structure

For security reasons , Cloudron deploys apps in a readonly filesystem. This means that the application code is unmodifiable. All application data is stored in a special directory `/app/data`.

Unfortunately, Moodle is not deployable into a readonly file system since it has way too many plugins and directories that it requires to be writable. As a workaround, we deploy Moodle into a writable directory named `/app/data/moodle` . The actual Moodle data is stored at `/app/data/moodledata`. On Cloudron, we rely on app source code being readonly to perform reliable updates. As mentioned earlier, this is not possible with Moodle. Therefore, it is important to note that Moodle source code under `/app/data/moodle` while editable should not be changed and treated as readonly. All changes will be lost across updates. The only files that can be changed are the various plugin and theme directories.

When performing an update, Cloudron will move the old version to `/app/data/moodle-prev-do-not-touch` and the latest version will be at `/app/data/moodle` . As is clear with the directory name, you should not modify anything inside `/app/data/moodle-prev-do-not-touch`.

7.110.3 Themes

To install a theme, extract the new theme under `/app/data/moodle/theme` using the File manager. Then, complete the installation by going to `Site Administration` in Moodle.

7.111 Navidrome App

7.111.1 About

Navidrome is an open source web-based music collection server and streamer.

- Questions? Ask in the Cloudron Forum - Navidrome
- Navidrome Website
- Navidrome community
- Navidrome issue tracker

7.111.2 Music folder

To change the music folder, edit `config.toml` using the File Manager

```
MusicFolder = '/media/MyMusic`
```

Be sure to restart the app after changing the music folder location.

7.111.3 Custom Configuration

Custom configuration can be placed in `/app/data/config.toml`. See the Navidrome docs for all the options.

7.111.4 CLI

To trigger a scan, open a Web terminal and run the following command:

```
gosu cloudron:cloudron /app/code/navidrome -c /app/data/config.toml scan -f
```

7.112 The nextcloud logo Nextcloud App

7.112.1 About

Nextcloud is the self-hosted productivity platform that keeps you in control.

- Questions? Ask in the Cloudron Forum - Nextcloud
- Nextcloud Website
- Upstream Nextcloud forum
- Upstream Nextcloud issue tracker

7.112.2 Installing Nextcloud client on Ubuntu

Nextcloud provides its own desktop client for Linux in form of AppImage which can be downloaded by issuing the following command:

```
sudo wget -nv https://download.nextcloud.com/desktop/releases/Linux/latest -O Nextcloud.AppImage
```

Alternatively, for Ubuntu, the latest version of the client can be installed from PPA following the instructions here:

```
sudo add-apt-repository ppa:nextcloud-devs/client
sudo apt-get update
sudo apt-get install nextcloud-client
```

For other platforms, please follow client installation instructions located at the Nextcloud website.

7.112.3 Custom config

To add custom configuration, use the File Manager to edit `/app/data/config/config.php`.

7.112.4 Plugin warning

We do not recommend installing apps in Nextcloud unless absolutely required. Maintaining such systems is a security hassle since you need to keep them up-to-date. Apps often break when Nextcloud is updated and you have to know how to fix them. Finally, Nextcloud apps are not run sandboxed. This means that a faulty plugin might compromise the whole app and also not make the app work at all. Nextcloud apps also write into the same database as the main application which might result in unintended data corruption.

For the above reason, extensive use of Nextcloud plugins is highly discouraged since it will eventually break your install.

7.112.5 Running occ tool

The `occ` tool can be used for Nextcloud administrative tasks.

The occ command can be run using the Web terminal. For example, to list the users:

```
sudo -u www-data php -f /app/code/occ user:list
```

7.112.6 Resetting admin password

To reset the admin password, run the following occ command using the Web terminal:

```
sudo -u www-data php -f /app/code/occ user:resetpassword admin
```

If you had deleted the admin user previously by mistake, you can create it again:

```
sudo -u www-data php -f /app/code/occ user:add --display-name="Admin" -g admin admin
```

You can also make an existing user an admin:

```
sudo -u www-data php -f /app/code/occ group:adduser admin <username> -n
```

7.112.7 LDAP Sync

Nextcloud will periodically sync users from LDAP. However, we have noticed that this fails at times. To trigger a manual sync, use the Web terminal and run the following command:

```
sudo -u www-data php -f /app/code/occ ldap:check-user --update <username>
```

7.112.8 Managing deleted files

When you delete a file in Nextcloud, it is not immediately deleted permanently. Instead, it is moved into the trash bin. It is not permanently deleted until you manually delete it, or when the Deleted Files app deletes it to make room for new files.

To configure, how items are permanently deleted, configure the trashbin_retention_obligation parameter.

The parameter can be edited using the File Manager and editing the file `config/config.php`.

> **Default is `7,30`**
>
> In recent versions of the nextcloud package, cloudron sets the default retention to a min of 7 days and a max of 30 days.

7.112.9 Attaching external storage

Many VPS providers like Digital Ocean, Linode allow attaching external block storage to the server. Nextcloud has a feature that allows mounting additional directories on the server as external storage.

Mounting an existing server directory as 'external storage' on Nextcloud is currently not supported.

If the intent is to simply increase the amount of storage available to Nextcloud (since you have run out of disk space in the default data partition), there are two options:

- Configure Nextcloud to use an external object storage like Digital Ocean Spaces, AWS S3 etc
- DigitalOcean Spaces Guide
- Configure Cloudron to store all of Nextcloud's data in the external block storage. To achieve this, follow the guide for moving a single app's data directory to another location.

Moving Nextcloud's directory entirely has the advantage that the iOS/Android app's Instant Upload feature uses this new disk.

7.112.10 Rescan files

Nextcloud will not pick up files if they are added directly in the data directory of the user on the server. To make it rescan, open a Web terminal and run the following command:

```
sudo -u www-data php -f /app/code/occ files:scan <username>
```

To rescan external storage, use the `--path` parameter.

```
sudo -u www-data php -f /app/code/occ files:scan <username> --path=/<username>/files/externaltest
```

7.112.11 Fixing a broken install

The Nextcloud App Store has a wide variety of apps that can be installed on top of Nextcloud. Nextcloud has no native sandboxing mechanism for plugins - if a plugin fails, it will bring down the whole installation. Plugins might also break an installation after a Nextcloud upgrade. For this reason, we encourage carefully reviewing apps before using them.

If it's not immediately obvious which plugin may be causing an issue, edit `/app/data/config/config.php` and set `debug` to `true`. Save the file and refresh the browser (do not restart the app because restarting it will reset the `debug` value back to `false`). It will then contain a stack trace of the error which should help identify the plugin.

Once identified, open a Web terminal. Then run the following commands:

```
sudo -u www-data php -f /app/code/occ app:list            # this lists the apps
sudo -u www-data php -f /app/code/occ app:disable <app>   # use this to disable the faulty app
sudo -u www-data php /app/code/occ maintenance:mode --off
```

After running the commands, end the repair for the app to come up.

7.112.12 Collabora Online Document Editor

Collabora Online is a powerful online office suite that supports all major document, spreadsheet and presentation file formats, which you can integrate in your own infrastructure. Key features are collaborative editing and excellent office file format support.

See the Collabora App docs on how to setup Nextcloud with Collabora Office.

a screenshot of the collabora editor showing the default Nextcloud document

7.112.13 Previews

By default, Nextcloud generates previews for text and images. Previews for other document types is disabled for privacy reasons. Note that generating previews also require more memory and CPU.

To enable previews for PDF and OpenOffice documents, open a File Manager and edit `config/config.php` and add the following setting:

```
'enable_previews' => true,
'enabledPreviewProviders' =>
   array (
      0 => 'OC\\Preview\\TXT',
      1 => 'OC\\Preview\\MarkDown',
      2 => 'OC\\Preview\\OpenDocument',
      3 => 'OC\\Preview\\PDF',
      4 => 'OC\\Preview\\MSOffice2003',
      5 => 'OC\\Preview\\MSOfficeDoc',
      6 => 'OC\\Preview\\PDF',
      7 => 'OC\\Preview\\Image',
      8 => 'OC\\Preview\\Photoshop',
      9 => 'OC\\Preview\\TIFF',
     10 => 'OC\\Preview\\SVG',
     11 => 'OC\\Preview\\Font',
     12 => 'OC\\Preview\\MP3',
     13 => 'OC\\Preview\\Movie',
     14 => 'OC\\Preview\\MKV',
     15 => 'OC\\Preview\\MP4',
     16 => 'OC\\Preview\\AVI',
   ),
```

7.112.14 Removing NextCloud users

To delete obsolete LDAP users and their data, see the nextcloud docs.

7.112.15 Skeleton directory

A skeleton directory provides the initial list of files for Nextcloud when a new user is created. By default, the skeleton directory is `/app/code/core/skeleton`. This directory is read only but can be changed to a custom directory.

1. Open the File Manager.
   a. Create a directory named `skeleton`.
   b. Set the owner of the directory to `www-data`. You can do this by clicking the icon to the right of the directory.
   c. Add files and directories to this new skeleton directory. Be sure to fix ownership of the files to `www-data`
2. Edit `config/config.php` to contain the following line:

```
    'skeletondirectory' => '/app/data/skeleton',
```

New users will receive the contents of `skeleton` directory on first log-in. The `skeletondirectory` property above can be set to empty string (`''`) to have no files added on first login.

7.112.16 Email

NextCloud has apps like Mail, SnappyMail to access email.

> **Not recommended**
>
> We do not recommend installing apps in Nextcloud unless absolutely required. Maintaining such systems is a security hassle since you need to keep them up-to-date. Apps often break when Nextcloud is updated and you have to know how to fix them. Finally, Apps are not run sandboxed. This means that a faulty plugin might compromise the whole app and also not make the app work at all.

Rainloop App

> **Discontinued**
>
> Please do not use the Rainloop app. Upstream development has ceased.

That warning aside, it is possible to configure Rainloop to access mail as follows.

1. Login to Rainloop admin. You can find the admin link by going to Nextcloud `Settings` -> `Administration` -> `Additional Settings`.
2. Login as `admin` / `12345`
3. Add your domain for email. In the example below, we use `my.cloudron.cf`.

IMAP and SMTP configuration:

a screenshot showing the configuration of IMAP and SMTP settings

Note that the SMTP encryption is turned off intentionally. This is safe because the communication is within the same server. (STARTTLS is disabled by Cloudron Email intentionally in the internal network for app compatibility when used with various languages and frameworks).

Sieve configuration:

a screenshot showing the configuration of Sieve settings

Mail App

The Mail icon will appear in the top bar of Nextcloud. You can configure your mailbox as below:

a screenshot showing the mailbox configuration of the mail app of nextcloud

Note that the SMTP encryption is turned off intentionally. This is safe because the communication is within the same server. (STARTTLS is disabled by Cloudron Email intentionally in the internal network for app compatibility when used with various languages and frameworks).

7.112.17 Password reset link

The password reset link can be customized by editing `/app/data/config/config.php`'s `lost_password_link` parameter using the File Manager.

If you are using Cloudron exclusively for user authentication, you can set it like this:

```
    'lost_password_link' => getenv('CLOUDRON_WEBADMIN_ORIGIN') . '?passwordReset',
```

Please note that if you added users inside Nextcloud that are not managed by Cloudron, the link might be misleading.

7.112.18 OIDC and App-Passwords for external applications

Creating an App-Password for Nextcloud Files Desktop App and other non Nextcloud Web-Browser based applications is required since Nextcloud has moved to OpenID connect login since package version v5.0.4.

If you are using the Nextcloud Files Desktop App, you will need to generate a App-Password within Nextcloud. To do so, you must go to your users `Personal Settings` > `Security` scroll all the way down and within the section of `Devices & sessions` you can create an App-Password. This will give you a username and a password for e.g. the Nextcloud Files Desktop app.

7.113 n8n App

7.113.1 About

n8n Free and open fair-code licensed node based Workflow Automation Tool.

- Questions? Ask in the Cloudron Forum - n8n
- n8n Website
- n8n forum
- n8n issue tracker

7.113.2 Custom env

Custom environment variables can be set in `/app/data/env.sh` using the File manager. Be sure to restart the app after making changes.

7.113.3 Timezone

To set the timezone, set the `GENERIC_TIMEZONE` environment variable in `/app/data/env.sh`. Be sure to restart the app after setting the timezone.

7.113.4 Built-in node modules

n8n allows using built-in node modules. To use an built-in node module, edit `/app/data/env.sh`, add a line like below and restart the app:

```
export NODE_FUNCTION_ALLOW_BUILTIN=crypto
```

See upstream docs for more information.

7.113.5 Custom node modules

To install custom node modules, edit `/app/data/env.sh` using the File manager:

```
# note: this is a space separated list
export EXTRA_NODE_MODULES="handlebars@4.7.7 jsonata@2.0.2 marked@4.3.0"
```

The modules have to be whitelisted for use:

```
# note: this is a comma separated list
export NODE_FUNCTION_ALLOW_EXTERNAL=handlebars,jsonata,marked
```

Restart the app and use the module in Function nodes. Restarting the app will install the modules specified in `EXTRA_NODE_MODULES` automatically. See upstream docs for more information.

7.113.6 CLI

To use the n8n CLI, open a Web terminal:

```
source /run/n8n/env.sh
gosu cloudron /app/code/node_modules/.bin/n8n export:workflow --backup --output=/tmp/n8n/
```

7.114 NocoDB App

7.114.1 About

NocoDB is an open source #NoCode platform that turns any database into a smart spreadsheet.

- Questions? Ask in the Cloudron Forum - NocoDB
- NocoDB website
- NocoDB docs
- NocoDB issue tracker

7.114.2 Custom config

Custom configuration can be set in `/app/data/env.sh` using the File manager.

See upstream docs for more information.

7.114.3 Telemetry

As per upstream default, telemetry is enabled by default. You can disable this by setting the following in `/app/data/env.sh` using the File manager:

```
export NC_DISABLE_TELE=true
```

7.114.4 Email

- Go to Account Settings as the Admin, then to the App Store page
- Install and configure the `SMTP` app. If you are using Cloudron Email, create a mailbox and configure it here. Use settings similar to below:

7.115 NodeBB App

7.115.1 About

NodeBB is next generation forum software. It's powerful, mobile-ready and easy to use.

- Questions? Ask in the Cloudron Forum - NodeBB
- NodeBB Website
- NodeBB forum
- NodeBB issue tracker

7.115.2 Installing plugins

NodeBB admin dashboard offers a UI to install plugins and themes in their dashboard. However, some plugins/themes may need to be installed by hand. To do so, use the Web terminal:

```
cd /app/code
/usr/local/bin/gosu cloudron:cloudron npm install nodebb-theme-timuu
```

After installation, restart the app and activate the plugin in the NodeBB dashboard.

7.115.3 Disabling plugins

The list of plugins can be viewed as follows:

```
cd /app/code
./nodebb plugins
```

Plugins can sometimes make NodeBB not start up. To fix this, first put the app in recovery mode. Then, open a Web Terminal:

```
cd /app/code
/app/pkg/start.sh # this create and configures nodebb
./nodebb reset -p nodebb-plugin-pluginname
```

7.116 ntfy App

7.116.1 About

ntfy lets you send push notifications to your phone or desktop via scripts from any computer, using simple HTTP PUT or POST requests. I use it to notify myself when scripts fail, or long-running commands complete.

- Questions? Ask in the Cloudron Forum - ntfy
- ntfy Website
- ntfy Docs
- ntfy issue tracker

7.116.2 Custom config

Custom configuration can be set in `/app/data/config/server.yml` using the File manager.

7.116.3 CLI

To access the ntfy CLI, simply open a Web terminal and run `ntfy`:

```
root@89c61cad-e48d-4fce-bcc2-b82b3b1b8a69:/app/code# ntfy
NAME:
   ntfy - Simple pub-sub notification service

USAGE:
   ntfy [OPTION..]

COMMANDS:
...
```

CLI configuration can be adjusted in `/app/data/config/client.yml`.

7.116.4 Access control

ntfy's access control mechanism can be used to configure authentication and authorization.

By default, Cloudron configures ntfy as a private instance. `auth-default-access` is set to `deny-all`.

Web interface

The web interface is open to all and can be used to send/receive notifications from one or more ntfy servers. Keeping the UI unprotected is harmless as it's purely a frontend application and one cannot send and receive notifications without authentication.

7.116.5 Change password

To change the password of a user, use the Web terminal:

```
# ntfy user change-pass admin
changed password for user admin
```

As note here, the web interface can send and receive notifications from one or more ntfy servers. Changing the password only changes the password in the backend. To change the password in the frontend, you have to enter the above password in `Settings` -> `Manage Users`.

7.117 Ollama App

7.117.1 About

Ollama is the easiest way to get up and running with large language models such as gpt-oss, Gemma 3, DeepSeek-R1, Qwen3 and more.

- Questions? Ask in the Cloudron Forum - Ollama
- Website
- Issue tracker

7.117.2 API Key

The app generates an api key, which can be used as a bearer token for other apps like librechat or OpenWebUI. You find that key in `/app/data/.api_key` or also in the startup logs of the app. Any key can be placed in that file to change it, but the app has to be restarted afterwards.

7.117.3 Downloading Models

Models can be downloaded via the Web Terminal using the command:

```
ollama pull $MODELL_NAME
```

A list of available models can be found on the Ollama Models Page.

7.118 OmekaS App

7.118.1 About

Omeka provides open-source web publishing platforms for sharing digital collections and creating media-rich online exhibits.

- Questions? Ask in the Cloudron Forum - OmekaS
- OmekaS Website
- OmekaS issue tracker

7.118.2 User Management

Cloudron user-management is supported by the pre-installed Ldap module where user email and name attributes settings with `mail` and `username`. The default user role is "researcher", the lower role. Please note that users who connect this way must login with their username and not their email address.

7.118.3 Modules

Themes and modules can be installed via the Web terminal or via the file manager UI. Take care to verify compatibility withe Omeka S version and preinstalled components with Cloudron Omeka S app.

7.118.4 CORS

```
<IfModule mod_headers.c>
    <FilesMatch "\.json$">
        Header add Access-Control-Allow-Origin "*"
        Header add Access-Control-Allow-Headers "origin, x-requested-with, content-type"
        Header add Access-Control-Allow-Methods "GET, POST, OPTIONS"
    </FilesMatch>
</IfModule>
```

7.119 ONLYOFFICE App

7.119.1 About

ONLYOFFICE has to be integrated with some the document store. On Cloudron there is currently Nextcloud available as a document store application, other 3rdparty solutions are also supported.

- Questions? Ask in the Cloudron Forum - ONLYOFFICE Docs
- ONLYOFFICE Website
- ONLYOFFICE forum
- ONLYOFFICE Docs issue tracker

7.119.2 Changing default app secret

The default secret for the ONLYOFFICE app package in Cloudron is `changeme`. Please change that to some unique secret:

- Open a File Manager into the app
- Edit the file `/app/data/config/production-linux.json`
- Locate the section called `secret`.

```
"secret": {
  "inbox": {
    "string": "changeme"
  },
  "outbox": {
    "string": "changeme"
  }
```

- Be sure to change the **two secrets** above to the same unique password.
- Restart the app

7.119.3 Setup Nextcloud connector

Do not install the Document Server

There are two ONLYOFFICE apps - Community Document Server and ONLYOFFICE. Be sure to install the latter.

To integrate ONLYOFFICE into Nextcloud for office document editing and collaboration, install ONLYOFFICE from the Nextcloud app library and configure the plugin as follows, adjusting the domain and secret:

If your Nextcloud has a self-signed certificate, you should also add the following stanza under `services` -> `CoAuthoring` in `/app/data/config/production-linux.json` using the File manager. Be sure to restart the app after adding the section below:

```
  "services": {
    "CoAuthoring": {
      "requestDefaults": {
        "rejectUnauthorized": false
      },
      ...
    }
  }
```

7.119.4 Custom Fonts

ONLYOFFICE supports adding custom fonts. Place the TTF files in `/app/data/fonts/` via the File Manager, then restart the app and the browser cache has to be cleared for the frontend to pickup the new fonts.

7.119.5 Enterprise License

OnlyOffice on Cloudron comes in two editions. Install the enterprise edition if an OnlyOffice enterprise license should be used. A license can be purchased here.

The license key has to be put in the app at `/app/data/license.lic` and the app needs to be restarted then to pick it up.

7.120 OpenHAB App

7.120.1 About

The open Home Automation Bus (openHAB, pronounced ˈəʊpənˈhæb) is an open source, technology agnostic home automation platform which runs as the center of your smart home!

- Questions? Ask in the Cloudron Forum - OpenHAB
- OpenHAB Website

7.121 OpenProject App

7.121.1 About

OpenProject is an efficient classic, agile or hybrid project management in a secure environment.

- Questions? Ask in the Cloudron Forum - OpenProject
- OpenProject Website
- OpenProject forum
- OpenProject issue tracker

7.121.2 User management

Cloudron Directory

Cloudron users can login to the OpenProject. The Cloudron admin status however is not carried over to the app, thus it comes with a pre-setup admin account, which requires a password change upon first login. Check the post-install notes when installing the app for admin account username and password.

OpenProject supports various authentication methods in parallel, this means even when Cloudron Directory is used, non Cloudron users can still be invited to the app.

Without Cloudron Directory

The app has a pre-setup admin account, which requires a password change upon first login. Check the post-install notes when installing the app for admin account username and password. Other users can be invited or added by this admin.

7.121.3 Custom Configuration

Use the File Manager to edit custom configuration in `/app/data/env.sh`. See the OpenProject docs for all available options.

Once the file was changed, the app has to be restarted to apply the changes.

7.122 OpenWebUI App

7.122.1 About

Open WebUI is an extensible, feature-rich, and user-friendly self-hosted WebUI for various LLM runners, supported LLM runners include Ollama and OpenAI-compatible APIs.

- Questions? Ask in the Cloudron Forum - OpenWebUI
- OpenWebUI Website
- OpenWebUI Docs
- OpenWebUI Community
- OpenWebUI issue tracker

7.122.2 User management

Registration is enabled by default. However, new users have to be approved by administrator before they can start using the app. You can disable new registration altogether in the admin settings.

7.122.3 Ollama

To use Ollama you need to provide an Ollama API key and the Ollama host URL in the app settings.

- navigate to `admin/settings/connections`
- under `Connections` press the cogwheel next to `Ollama API`
- press `Connection Type` text `Local` to change it into `External`
- enter the `URL` and `API Key` from e.g. your Ollama Cloudron app
- press `Test Connection` to verify the connection
- press `Save`

O■ma app

If you want to use Cloudron's Ollama app use the domain of your Ollama app as host URL, e.g. `https://ollama.yourdomain.com` and the API key found in `/app/data/.api_key` of the Ollama app.

To install modells in the Ollama app please refer to the Ollama app documentation.

7.123 Open Web Calendar App

7.123.1 About

The Open Web Calendar uses ICS/ICal calendars online and displays them in one calendar. You can use it with Nextcloud, Outlook, Google Calendar, Meetup and other calendar systems using the ICS standard.

- Questions? Ask in the Cloudron Forum - Open Web Calendar
- Open Web Calendar repo
- Open Web Calendar issue tracker

7.124 oPodSync App

7.124.1 About

oPodSync is a minimalist podcast synchronization server

- Questions? Ask in the Cloudron Forum - oPodSync
- oPodSync repo
- oPodSync issue tracker

7.125 osTicket App

7.125.1 About

osTicket is the world's most popular customer support software.

- Questions? Ask in the Cloudron Forum - osTicket
- osTicket Website
- osTicket forum
- osTicket issue tracker

7.125.2 Admin Checklist

There are a number of email addresses that need to be fixed up before you can start using osTicket.

- Change the administrator email (default: `admin@server.local`) and password. This can be changed under `Agents` tab.
- osTicket adds 3 email addresses by default (listed under `Emails` -> `Email Addresses`. To use the app, you must change the email addresses below and setup their `Outbound (SMTP)` configuration. Without SMTP configuration, osTicket uses phpmailer, which does not work on Cloudron. If mailboxes are hosted on Cloudron, see the SMTP Configuration section below.
  - `osTicket Alerts <alerts@server.local>` - Email address from which Alerts & Notices are sent to Agents. This can be changed in `Emails` -> `Email Settings and Options` -> `Default Alert Email`.
  - `noreply@server.local` - This is added by default and not used anywhere. Can be removed if you have no use for this.
  - `Support <support@server.local>` - Email address from which outgoing emails are sent. This can be changed in `Emails` -> `Email Settings and Options` -> `Default System Email`.
- Change the Admin email address. This is the email address to which System Errors and New Ticket Alerts (if enabled) are sent. This can be changed under `Emails` -> `Email Settings and Options` -> `Admin's Email Address`. If you miss this, osTicket will send alerts to this address and bounces get attached to tickets.

7.125.3 User Management

osTicket is integrated with Cloudron user management. However, osTicket does not support auto creation of user accounts - see this and this for more information.

To workaround agents must be manually added into osTicket before they can login. When adding an agent, choose LDAP as the authentication backend.

7.125.4 Cloudron SMTP Configuration

osTicket can be configured to process emails from mailboxes hosted with or without Cloudron.

When the mailbox is hosted in Cloudron, you can use the IMAP+SSL at port 993 for receiving Email:

To send email, use port 587:

7.125.5 CLI

osTicket comes with a CLI tool for various administrative tasks like managing users. Use the Web Terminal to run the following command:

```
sudo -E -u www-data php /app/code/upload/manage.php
```

7.126 Outline App

7.126.1 About

Outline is the fastest knowledge base for growing teams.

- Questions? Ask in the Cloudron Forum - Outline
- Outline Website
- Outline Forum

7.126.2 Login

Without Cloudron Directory

Outline requires at least one SSO provider configured. This means that when authentication against Cloudron Directory is disabled, there is no way to login to the app and you will simply see a login screen with no input fields post installation.

To proceed:

- Configure one of the SSO methods in `/app/data/env.sh` using the File manager.
- Restart the app

7.126.3 License

Please be aware of the BSL license restrictions before installing Outline – selling, reselling, or hosting Outline as a service is a breach of the terms and automatically terminates your rights under the license.

See https://docs.getoutline.com/s/hosting/ for more information.

7.126.4 Custom config

Custom environment variables can be set in `/app/data/env.sh` using the File manager. For example:

```
export FILE_STORAGE_UPLOAD_MAX_SIZE=26214400
export MAXIMUM_IMPORT_SIZE=5120000
```

Be sure to restart the app after making any changes.

7.127 Owncast App

7.127.1 About

Owncast is an open source, self-hosted, decentralized, single user live video streaming and chat server for running your own live streams similar in style to the large mainstream options. It offers complete ownership over your content, interface, moderation and audience.

- Questions? Ask in the Cloudron Forum - Owncast
- Owncast Website
- Owncast docs
- Owncast issue tracker
- Owncast Rocket.chat

7.127.2 Broadcasting

The way it works is that you point your Broadcasting software (like OBS) to Owncast and Owncast shows a web page to the world with your stream. Broadcasting software talks to Owncast using RTMP. Use `rtmp://yourserver/live` as the RTMP destination. By default the stream key is `abc123`. For broadcasting software without a separate stream key section, one can use the RTMP URL as `rtmp://yourserver/live/abc123` .

7.127.3 Embedding

The stream can be embedded like so:

```
<iframe
  src="https://your.host/embed/video"
  title="Owncast"
  height="350px" width="550px"
  referrerpolicy="origin"
  scrolling="no"
  allowfullscreen>
</iframe>
```

7.127.4 HLS Stream

The HTL Stream is located at `http://your.host/hls/stream.m3u8`

7.128 ownCloud App

7.128.1 About

ownCloud is a suite of client–server software for creating and using file hosting services.

- Questions? Ask in the Cloudron Forum - ownCloud
- ownCloud Website

7.128.2 Installing ownCloud client on Ubuntu

The ownCloud client on Ubuntu is outdated. The client will display an error:

```
Error downloading https://SERVERNAME/owncloud/remote.php/webdav/ - server replied: Forbidden (Unsupported client version.)".
```

To resolve the problem, install the ownCloud client by following the instructions here:

```
sudo wget -nv https://download.owncloud.com/repositories/desktop/Ubuntu_16.04/Release.key -O Release.key
sudo apt-key add - < Release.key
sudo sh -c "echo 'deb http://download.owncloud.com/repositories/desktop/Ubuntu_16.04/ /' > /etc/apt/sources.list.d/owncloud.list"
sudo apt-get update
sudo apt-get install owncloud-client
```

> ⚠ **Warning**
>
> As of this writing, the ownCloud website links to the opensuse website for installing the owncloud client. The packages in the opensuse website do not work. See the forum and GitHub issue for more information.

7.128.3 Running occ tool

The `occ` tool can be used for ownCloud administrative tasks.

The occ command can be run using the Web terminal. For example, to list the users:

```
    sudo -u www-data php -f /app/code/occ app:list
```

7.128.4 Managing deleted files

When you delete a file in ownCloud, it is not immediately deleted permanently. Instead, it is moved into the trash bin. It is not permanently deleted until you manually delete it, or when the Deleted Files app deletes it to make room for new files.

To change permanent deletion policy configure the trashbin_retention_obligation parameter.

The parameter can be edited using the Web terminal and editing the file `/app/data/config/config.php`.

7.128.5 Max upload size

The app is configured to allow maximum uploads of up to 5GB.

7.128.6 Attaching external storage

Many VPS providers like Digital Ocean, Linode allow attaching external block storage to the server. ownCloud has a feature that allows mounting additional directories on the server as external storage.

Mounting an existing server directory as 'external storage' on ownCloud is currently not supported.

If the intent is to simply increase the amount of storage available to ownCLoud (since you have run out of disk space in the default data partition), there are two options:

- Configure ownCloud to use an external object storage like Digital Ocean Spaces, AWS S3 etc.
- Configure Cloudron to store all of ownCloud's data in the external block storage. To achieve this, follow the guide for moving a single app's data directory to another location.

Moving ownCloud's directory entirely has the advantage that the iOS/Android app's Instant Upload feature uses this new disk.

7.128.7 Rescan files

ownCloud will not pick up files if they are added directly in the data directory of the user on the server. To make it rescan, open a Web terminal and run the following command:

```
sudo -u www-data php -f /app/code/occ files:scan <username>
```

To rescan external storage, use the `--path` parameter.

```
sudo -u www-data php -f /app/code/occ files:scan <username> --path=/<username>/files/externaltest
```

7.128.8 Fixing a broken install

The ownCloud Marketplace has a wide variety of apps that can be installed on top of ownCloud. ownCloud has no native sandboxing mechanism for plugins - if a plugin fails, it will bring down the whole installation. Plugins might also break an installation after a ownCloud upgrade. For this reason, we encourage carefully reviewing apps before using them.

To fix a broken installation, open a Web terminal and repair the app. Then run the following commands:

```
sudo -u www-data php -f /app/code/occ app:list            # this lists the apps
sudo -u www-data php -f /app/code/occ app:disable <app>   # use this to disable the faulty app
sudo -u www-data php /app/code/occ maintenance:mode --off
```

After running the commands, end the repair for the app to come up.

7.129 PairDrop App

7.129.1 About

Local file sharing in your browser. Inspired by Apple's AirDrop. Fork of Snapdrop.

- Questions? Ask in the Cloudron Forum - PairDrop
- Pairdrop Website
- PairDrop Issue Tracker
- PairDrop Discussions

7.129.2 Custom RTC Config

To use this app with a custom TURN server:

- Create a file named `/app/data/rtc_config.json` using the File Manager. See upstream example.
- Add the line below to `/app/data/env.sh` using the File Manager.

```
export RTC_CONFIG=/app/data/rtc_config.json
```

- Disable Cloudron's TURN Server auto-configuration for the app in the TURN section. Note that if you do not provide a custom config file above, the TURN server defaults to Google's server (stun.l.google.com).

7.130 Paperless-ngx App

7.130.1 About

Paperless-ngx is an application that manages your personal documents. With the help of a document scanner (see Scanner recommendations), paperless transforms your wieldy physical document binders into a searchable archive and provides many utilities for finding and managing your documents.

- Questions? Ask in the Cloudron Forum - Paperless-ngx
- Paperless-ngx website
- Paperless-ngx community
- Paperless-ngx issue tracker

7.130.2 Custom config

Custom configuration can be set in `/app/data/paperless.conf` using the File manager. See upstream docs for various options.

7.130.3 Uploading

Files should be uploaded to `/app/data/consume` using the File manager. Once uploaded, a background task scans the documents automatically.

7.130.4 Document retagger

To run the document retagger, open a Web Terminal and execute:

```
# cd /app/code/src
# python3 manage.py document_retagger -T
```

7.130.5 Exporting

To export existing documents, open a Web Terminal and execute:

```
# cd /app/code/src
# mkdir -p /app/data/out
# python3 manage.py document_exporter /app/data/out
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 545.92it/s]
```

7.130.6 Importing

To import an existing export at `/app/data/in`, open a Web Terminal and execute:

```
# cd /app/code/src
# python3 manage.py document_importer /app/data/in
Installed 4 object(s) from 1 fixture(s)
Copy files into paperless...
100%|█████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 34.11it/s]
Updating search index...
100%|████████████████████████████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 104.00it/s]
```

7.131 PeerTube App

7.131.1 About

PeerTube is an activityPub-federated video streaming platform using P2P directly in your web browser.

- Questions? Ask in the Cloudron Forum - PeerTube
- PeerTube Website
- PeerTube issue tracker

7.131.2 Customization

Use the File manager to edit custom configuration under `/app/data/production.yaml`.

7.131.3 CLI

The CLI can be accessed using the `peertube-cli` command. You can view the various commands in the PeerTube docs.

Uploading video

```
# peertube-cli up --file /tmp/video.wmv --url https://peertube.cloudron.club --username root --password changeme --video-name "Sample video"
Uploading Sample video video...
Video Sample video uploaded.
```

7.131.4 Disable P2P

If playback is slow, it might be a good idea to disable the P2P functionality. For this, use the file File manager to edit `/app/data/production.yaml` and set tracker to false.

```
tracker:
  # If you disable the tracker, you disable the P2P aspect of PeerTube
  enabled: false
```

7.132 Penpot App

7.132.1 About

Penpot is the first Open Source design and prototyping platform meant for cross-domain teams. Non dependent on operating systems, Penpot is web based and works with open standards (SVG). Penpot invites designers all over the world to fall in love with open source while getting developers excited about the design process in return.

- Questions? Ask in the Cloudron Forum - Penpot
- Penpot Website
- Penpot Community

7.133 Phabricator App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.133.1 About

Phabricator is a set of tools that help companies build better software, faster.

- Questions? Ask in the Cloudron Forum - Phabricator
- Phabricator Website

7.133.2 Empower

A registered user can be made an administrator by running the following command in the Web Terminal:

```
# /app/code/phabricator/bin/user empower --user <username>
```

See the phabricator docs for more information.

7.133.3 Admin recovery

When not using Cloudron authentication, If you accidentally log yourself out before adding an Auth provider, you must use the CLI tool to recover it (or simply re-install phabricator). See T8282 for more information.

```
# /app/code/phabricator/bin/auth recover <admin-username>
```

7.133.4 Uploading large files

This app is configured to accept files upto 512MB. Note that large files need to be dragged and dropped (instead of the file upload button).

See Q216

7.134 PHP Server Monitor App

7.134.1 About

PHP Server Monitor is a script that checks whether your websites and servers are up and running.

- Questions? Ask in the Cloudron Forum - PHP Server Monitor
- Website
- Issue tracker

7.134.2 Check interval

The app is configured to check the status of online servers every 5 minutes and check the status of offline servers every minute.

7.134.3 Custom config

Custom configuration can be added in `/app/data/config.php` using the File manager.

7.134.4 Public page

To setup a public page (accessed at `/public.php`):

- Set `PSM_PUBLIC` to true in `/app/data/config.php`.
- Create a user named `__PUBLIC__`. Set the Level to `Anonymous`.
- Add servers to user '**PUBLIC**'.
- Go to `/public.php`.

7.135 Piwigo App

7.135.1 About

Piwigo is open source photo gallery software for the web. Designed for organisations, teams and individuals.

- Questions? Ask in the Cloudron Forum - Piwigo
- Piwigo Website
- Piwigo issue tracker

7.136 Pixelfed App

7.136.1 About

A free and ethical photo sharing platform, powered by ActivityPub federation.

- Questions? Ask in the Cloudron Forum - Pixelfed
- Pixelfed Website
- Pixelfed docs
- Pixelfed issue tracker

7.136.2 Admin

To make a register user an admin, use the Web terminal and run the following command:

```
# sudo -E -u www-data php artisan user:admin username_here

Found username: girish

 Add admin privileges to this user? (yes/no) [no]:
 > yes

Successfully changed permissions!
```

Further administration commands like removing a user, removing unused media can be found in Pixelfed docs.

The admin page is located at `https://pixelfed.example.com/i/admin/dashboard`.

7.136.3 Federation

To test if federation works, search for a handle like `@girish@pixelfed.social` or `@social@pixelfed.cloudron.io`. You should then be able to follow that handle. Note that you can only see posts that are made after you followed the handle. Existing posts will **not** appear in you stream.

If following doesn't work:

- Login as an admin user on your Pixelfed instance
- Check `https://pixelfed.domain.com/horizon/failed` for job failures

7.136.4 Customizations

Customizations can be made by editing in one of the following ways:

- In the Admin page. In Pixelfed, `Go back to previous design` and then select `Admin` from the navigation bar. Note that when making changes in the Admin Panel, the app has to be restarted and caches still have to be cleared.
- Editing `/app/data/env.production` using the File manager. See the Pixelfed docs on the various config options.

Note that any value in the settings UI seems to supersede the value in the env file. The exact ordering and priority of configuration settings is unclear.

Don`t forget to restart the app after making any changes.

Disable Registration

Registration can be disabled in the Admin page (see above).

Clear Cache

When making changes via the admin panel, often the cache has to be cleared in addition to an app restart. To clear the cache, open a Web Terminal and run the following commands:

```
# sudo -E -u www-data php artisan cache:clear
# sudo -E -u www-data php artisan optimize:clear
# sudo -E -u www-data php artisan optimize
```

7.137 PocketBase App

7.137.1 About

PocketBase is an Open Source backend for your next SaaS and Mobile app in 1 file.

- Questions? Ask in the Cloudron Forum - PocketBase
- PocketBase Site
- PocketBase Docs
- PocketBase Discussions
- PocketBase issue tracker

7.137.2 Default Site

PocketBase serves the static content in `/app/data/pb_public` (html, css, images) as the website. By default, the Cloudron package provides an `index.html` that redirects to the admin page. You can customize this as needed using the File manager.

7.137.3 Routes

There are 3 routes:

- `/` - the default site served from `/app/data/pb_public`
- `/_/` - Admin dashboard UI
- `/api/` - REST API

7.138 Postiz App

7.138.1 About

Postiz is the ultimate social media scheduling tool, with a bunch of AI.

- Questions? Ask in the Cloudron Forum - Postiz
- Postiz Website
- Postiz issue tracker

7.138.2 Customization

Postiz has various options.

To customize, edit `/app/data/env` using the File manager. Be sure to restart the app after making any changes.

⚠ **Cloudron specific variables that can't be overwritten**

Certain variables can't be overwritten with the `/app/data/env` file since they are defined in the start.sh. Please check the start.sh to see what variables are defined.

7.139 Pretix App

7.139.1 About

Ticket shop application for conferences, festivals, concerts, tech events, shows, exhibitions, workshops, barcamps, etc.

- Questions? Ask in the Cloudron Forum - Pretix
- Pretix Website
- Pretix issue tracker

7.139.2 Plugins

Pre-installed plugins are in `/app/code/venv/lib/python3.12/site-packages`.

Marketplace/Custom/User plugins are going to be installed to the default upstream location `/var/pretix/venv/lib/python3.12/site-packages`.

Plugins can be found in the Pretix Marketplace and then installed as follows:

```
source /var/pretix/venv/bin/activate
(venv)$ pip install <plugin name>
```

After that, the app can be restarted to rebuild plugins and apply db migrations.

7.140 Prometheus App

7.140.1 About

Prometheus is an open-source systems monitoring and alerting toolkit.

- Questions? Ask in the Cloudron Forum - Ackee
- Prometheus website
- Prometheus docs
- Prometheus support
- Prometheus issue tracker

7.140.2 CLI Args

Various operations aspects can be set in the command line. To adjust Prometheus command line options, edit `/app/data/env.sh` using the File manager and adjust `cli_options` variable as needed. For example:

`` ` ``

```
    export cli_options="--storage.tsdb.retention.time=25d --storage.tsdb.path=/app/data/storage"
```

Be sure to restart the app after making any changes.

7.140.3 Custom Config

Custom config can be added in `/app/data/config/prometheus.yml` using the File manager.

Be sure to restart the app after making any changes.

7.140.4 Accessing the HTTP API Endpoint

From the official prometheus.io/docs most examples use the URL `http://localhost:9090/api/v1/`. For the Cloudron app `https` is used and the port `:9090` needs to be left out.

As an example our Prometheus is running at `pm.demo.cloudron.io` and we want to use Prometheus in Grafana. Then we need to use the full `https://pm.demo.cloudron.io` URL and basic auth.

7.141 PrivateBin App

7.141.1 About

PrivateBin is a minimalist, open source online pastebin where the server has zero knowledge of pasted data. Data is encrypted/ decrypted in the browser using 256 bits AES.

- Questions? Ask in the Cloudron Forum - PrivateBin
- PrivateBin Website
- PrivateBin issue tracker

7.141.2 Customizations

Various PrivateBin settings can be configured by editing `/app/data/conf.php` using the File manager.

7.141.3 Custom template

You can set a custom template as follows:

- Create the template in `/app/data/custom_template/custom.php`. The name `custom.php` is hardcoded in the package.
- Change the template name in `/app/data/conf/conf.php` to be `custom`.
- You can save additional js/css/img in `/app/data/custom_template/` and access them from the php script as `js/custom/..`, `css/custom/...`, `img/custom/...`.

7.142 qBittorrent App

7.142.1 About

qBittorrent is an advanced and multi-platform BitTorrent client.

- Questions? Ask in the Cloudron Forum - qBittorent
- qBittorrent Website
- Vuetorrent Website

7.143 Radicale App

7.143.1 About

Radicale is a small but powerful CalDAV (calendars, to-do lists) and CardDAV (contacts) server.

- Questions? Ask in the Cloudron Forum - Radicale
- Radicale Website
- Radicale issue tracker

7.143.2 Custom permissions (e.g. shared calendar)

Per default, each user can only access their own calendar and contacts. If you want something more complicated you can change the permissions.

You can change the permissions editing `/app/data/rights` using the File Manager in the app instance. The default content of that file is:

```
[owner-write]
user = .+
collection = %(login)s(/.*)?
permission = rw

[read]
user = .*
collection =
permission = r
```

You can extend the file using the syntax described in the radicale documentation.

7.144 Rainloop App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.144.1 About

Rainloop is a simple, modern & fast web-based email client.

- Questions? Ask in the Cloudron Forum - RainLoop
- RainLoop Website
- RainLoop docs
- RainLoop issue tracker

7.144.2 Default Setup

Rainloop is pre-configured for use with Cloudron Email. The app automatically generates domain configuration for all the apps that have email enabled at installation time. If you enable or disable email on one or more domains, simply reconfigure the app and it will re-generate the necessary configuration.

7.144.3 Multi-domain Setup

There are two ways to use Rainloop when using Cloudron Email with multiple domains.

- Users can login with their email and password to access their mailbox. If the Cloudron has two domains, `example1.com` and `example2.com`, the user can login using `user@example1.com` and `user@example2.com`. Aliases can be added as identities under Rainloop settings.
- Users can login using one of the email domains and add the other domains using the `Add Account` dialog. For example, user can login as `user@example1.com` and add `user@example2.com` in the `Add Account` dialog.

Multiple accounts

Rainloop tracks accounts based on the login email. This means that in the example above, if the user logs in `user@example2.com`, the `user@example1.com` id will not show up.

7.144.4 External domains

To add one or more external domains, add them in Rainloop's admin panel.

7.144.5 Vacation Email

An out of office / vacation mail message can be setup using Sieve filters.

A vacation message can be set in `Settings` -> `Filters` -> `Add a filter`. Choose `Vacation message` action.

At most once a day

Vacation messages are sent at most once a day to the same sender. This setting cannot be changed.

7.144.6 Forwarding all emails

To forward all emails to an external mail, setup a Sieve filter in `Settings` -> `Filters` -> `Add a filter` -> `Forward to`

7.144.7 Admin panel

The admin panel is located at `/?admin` and is disabled by default.

To enable it, open a File manager and edit the file `/app/data/_data_/_default_/configs/application.ini`. Set the value of `allow_admin_panel` to `On`. The default admin credentials are:

```
Username: admin
Password: 12345
```

Restart the app for the changes to take effect.

⚠ **Disable admin panel after use**

We highly recommend disabling the admin panel after use.

7.144.8 Attachment size

To adjust the attachment size, change the PHP configuration and the Rainloop configuration.

- For PHP configuration, edit `/app/data/php.ini` using the File manager and add

  ```
  [PHP]
  upload_max_filesize = 40M
  post_max_size = 40M
  ```

- For Rainloop configuration, edit `/app/data/_data_/_default_/configs/application.ini` using the File Manager and adjust `attachment_size_limit`.

Be sure to restart the app after making the above changes.

In addition to above, you also have to configure the mail server to allow large email.

7.145 Redash App

7.145.1 About

Make Your Company Data Driven. Connect to any data source, easily visualize, dashboard and share your data.

- Questions? Ask in the Cloudron Forum - Redash
- Redash Website
- Redash forum
- Redash issue tracker

7.145.2 SSH tunnel

If your data source cannot be reached over the internet, you can setup a SSH tunnel. stunnel is another way to setup a TCP tunnel.

7.145.3 Custom config

Custom configuration can be added by editing `/app/data/env.sh` using the File Manager.

Be sure to restart the app after making any changes.

7.145.4 Admin access

The app is pre-setup with a admin account. To give admin access to other users, login as admin and add users to the `admin` group.

Alternately, use the Web terminal and run the following command:

```
/app/code/redash/manage.py users grant_admin <email>
```

7.145.5 Login form

By default, the app allows user's to login via `Cloudron LDAP` and via email. Once you have made one or more Cloudron user's as Redash admins, the email login can be disabled. For this, use the File manager and add/edit a file `/app/data/env.sh` and add the following line:

```
export REDASH_PASSWORD_LOGIN_ENABLED=false
```

Restart the app for changes to take effect.

7.146 Rallly App

7.146.1 About

Self-hostable doodle poll alternative. Find the best date for a meeting with your colleagues or friends without the back and forth emails.

- Questions? Ask in the Cloudron Forum - Rallly
- Rallly Website
- Rallly issue tracker

7.146.2 Restrict sign-up

To restrict sign-ups, edit `ALLOWED_EMAILS` in `/app/data/env` using the File manager.

For example, to restrict only to company email ids:

```
ALLOWED_EMAIL=*@cloudron.io
```

Be sure to restart the app after making any changes.

7.147 Redmine App

7.147.1 About

Redmine is a flexible project management web application

- Questions? Ask in the Cloudron Forum - Redmine
- Redmine Website
- Redmine forum

7.147.2 Installing plugins

To install plugins in redmine, simply extract them to `/app/data/plugins` and run the db migration.

- Open a File manager for the app.
- Upload and extract the plugin to `/app/data/plugins`
- Some plugins require additional gems. If so, install them using the Web terminal:

```
# cd /app/code
# bundle install
```

- Initialize the database of the plugin

```
# cd /app/code
# bundle exec rake redmine:plugins NAME=redmine_checklists RAILS_ENV=production
```

- Restart redmine using the `restart` button

7.147.3 Installing themes

To install plugins in redmine, simply extract them to `/app/data/themes`, install dependancies and run the build script

```
cd /app/data/themes/
git clone https://github.com/hardpixel/minelab.git
cd minelab
bundle install
./bundle.sh
```

7.147.4 Code repositories

Redmine can integrate various source code management tools like git, cvs, subversion. The repositories have to be created manually in `/app/data/repos/` and then configured with that path in the project settings.

For further more detailed information for repository integration can be found here.

7.147.5 SSH Keys

Redmine is run as the `cloudron` user. To generate SSH keys for this user, open a Web Terminal and run the following commands:

```
su - cloudron
ssh-keygen          # generates keys under ~/.ssh. keys are part of the backup
```

7.147.6 Custom Cron

Custom cron jobs can be placed in the file `/app/data/cron`.

7.147.7 Custom Settings

The ruby app settings can be overwritten in the file `/app/data/additional_environment.rb`. For example to put the app in debug mode, add the following to this file and restart the app:

```
config.log_level = :debug
```

7.148 Release Bell App

7.148.1 About

Release Bell is a self-hosted release notification service.

- Questions? Ask in the Cloudron Forum - Release Bell
- Release Bell Website

7.148.2 Supported Providers

ReleaseBell supports release notification for repos hosted on the following providers:

- GitHub
- GitLab

7.149 Rocket.Chat App

7.149.1 About

Rocket.Chat is a solution for communities and companies wanting to privately host their own chat service or for developers looking forward to build and evolve their own chat platforms.

- Questions? Ask in the Cloudron Forum - Rocket.Chat
- Rocket.Chat Website
- Rocket.Chat issue tracker

7.149.2 Mobile Clients

Rocket.Chat mobile clients are available for most mobile platforms:

- Play Store
- Apple App Store

7.149.3 Uploads

By default, Rocket.chat is configured to store files using `GridFS`. This stores attachments in the MongoDB database. You can configure the File Upload storage in the Rocket.Chat administration -> `Settings` -> `Upload`.

To change it to the File system:

- First, create a directory using the Web terminal:

```
# mkdir -p /app/data/uploads
# chown cloudron:cloudron /app/data/uploads
```

- Then, set the above directory as the target for uploads:

7.149.4 Webhook Integrations

Webhook Integrations can be added in the Administration panel under `Integrations`. Rocket.Chat supports notifications from and to other apps or services through its webhook integrations. Incoming notifications require a message body parsing code snippet, which transforms the incoming webhook to a readable message, which will be posted into a selected chat channel.

GitLab

Create a new `Incoming WebHook Integration` and configure the destination channel and user, then enalbe a custom sript. GitLab supports multiple types of webhook notifications and thus requires a more sophisticated transform script. The below works for issue, comment, merge request, push and tag events:

```
/*jshint  esnext:true*/
// see https://gitlab.com/help/web_hooks/web_hooks for full json posted by GitLab
const NOTIF_COLOR = '#6498CC';

class Script {
    process_incoming_request({
        request
    }) {
        try {
            switch (request.headers['x-gitlab-event']) {
                case 'Push Hook':
                    return this.pushEvent(request.content);
                case 'Merge Request Hook':
                    return this.mergeRequestEvents(request.content);
                case 'Note Hook':
                    return this.commentEvent(request.content);
                case 'Issue Hook':
                    return this.issueEvent(request.content);
                case 'Tag Push Hook':
                    return this.tagEvent(request.content);
            }
        } catch (e) {
            console.log('gitlabevent error', e);
            return {
                error: {
                    success: false,
                    message: e.message || e
                }
            };
        }
    }

    issueEvent(data) {
        return {
            content: {
                username: data.user.name,
                text: `${data.user.username} ${data.object_attributes.state} an issue _${data.object_attributes.title}_ on ${data.project.name}. *Description:* ${data.object_attributes.description}. See: ${data.object_attributes.url}`,
                icon_url: data.user.avatar_url
            }
        };
    }

    commentEvent(data) {
        const comment = data.object_attributes;
        const user = data.user;
        let text;
        if (data.merge_request) {
            let mr = data.merge_request;
            text = `${user.name} commented on Merge Request #${mr.id} [${mr.title}](${comment.url})`;

        } else if (data.commit) {
            let commit = data.commit;
            let message = commit.message.replace(/\n[^\s\S]+/, '...').replace(/\n$/, '');
            text = `${user.name} commented on commit [${commit.id.slice(0, 8)} ${message}](${comment.url})`;
        } else if (data.issue) {
            let issue = data.issue;
            text = `${user.name} commented on issue [#${issue.id} ${issue.title}](${comment.url})`;
        } else if (data.snippet) {
            let snippet = data.snippet;
            text = `${user.name} commented on code snippet [#${snippet.id} ${snippet.title}](${comment.url})`;
        }
        return {
            content: {
                username: 'gitlab/' + data.project.name,
                icon_url: data.project.avatar_url || user.avatar_url || '',
                text,
                attachments: [{
                    text: comment.note,
                    color: NOTIF_COLOR
                }]
            }
        };
    }

    mergeRequestEvent(data) {
        const user = data.user;
        const mr = data.object_attributes;
        return {
            content: {
                username: `gitlab/${mr.target.name}`,
                icon_url: mr.target.avatar_url || mr.source.avatar_url || user.avatar_url || '',
                attachments: [{
                    title: `${user.name} ${mr.action} Merge Request #${mr.id} ${mr.title}`,
                    title_link: mr.url,
                    text: `_${mr.source_branch} into ${mr.target_branch}_`,
                    color: NOTIF_COLOR
                }]
            }
        };
    }

    pushEvent(data) {
        const project = data.project;
        return {
            content: {
                username: `gitlab/${project.name}`,
                text: `![${data.user_name}](${data.user_avatar}) ${data.user_name} pushed ${data.total_commits_count} commits to ${project.name}. See: ${project.web_url}`,
                icon_url: project.avatar_url || data.user_avatar || '',
                attachments: [{
                    title: data.total_commits_count + ' Commits',
                    title_link: project.web_url,
                    text: data.commits.map((commit) => `  - ${new Date(commit.timestamp).toUTCString()} [${commit.id.slice(0, 8)}](${commit.url}) by ${commit.author.name}: ${commit.message.replace(/\s*$/, '')}`).join('\n'),
                    color: NOTIF_COLOR
                }]
            }
        };
    }

    tagEvent(data) {
        let tag = data.ref.replace(/^.*?([^\/]+)$/, '$1');
        return {
            content: {
                username: `gitlab/${data.project.name}`,
                icon_url: data.project.avatar_url || data.user_avatar || '',
                text: `${data.user_name} push tag [${tag} ${data.checkout_sha.slice(0, 8)}](${data.project.web_url}/tags/${tag})`
            }
        };
    }

}
```

After the integration is setup, the Webhook URL and Secret Token is generated and can now be setup with the project in GitLab at `Settings` -> `Integrations`.

⚠ **Webhook test**

Triggering a test webhook from GitLab will likely result in an error, since the above transform script expects a body, which is not provided from GitLab while running a test call. A test has to be made with a real action on the repo.

7.149.5 Live Chat

Rocket.Chat has a live chat feature that allows you to embed a chat widget into your website.

Head over to Rocket.Chat app's Administration view, select the Livechat section and enable it. After this, you will find a `Livechat` entry in the side bar.

You will also find options to customize the live chat widget appearance to match your look and feel there.

Then, copy the Javascript code snippet shown in the Livechat Installation view and paste it to the bottom of your website's html code as the last thing before the `</body>` tag. WordPress users can use the Rocket.Chat LiveChat WordPress plugin instead.

You will now see the live chat widget on the bottom right of your page.

7.149.6 Reset Admin Password

If you lost the admin password, you can make an existing user an admin. Use the Web Terminal to open a MongoDB shell and run the following command:

```
db.users.update({username:'myusername'}, {$set: {'roles' : [ "admin" ]}});
```

7.149.7 Environment Variables

Custom environment variables can be set in `/app/data/env` using the File Manager. Be sure to restart the app after exporting variables.

Setting timezone

Timezone can be set adding `export TZ=America/Brasilia` in `/app/data/env` and restarting the app.

7.150 Roundcube App

7.150.1 About

Roundcube webmail is a browser-based multilingual IMAP client with an application-like user interface.

- Questions? Ask in the Cloudron Forum - Roundcube
- Roundcube Website
- Upstream Roundcube issue tracker

7.150.2 Default Setup

Roundcube is pre-configured for use with Cloudron Email.

7.150.3 Multi-domain Setup

Users can login with their email and password to access their mailbox. If the Cloudron has two domains, `example1.com` and `example2.com`, the user can login using `user@example1.com` and `user@example2.com`. Aliases can be added as identities under Roundcube settings.

7.150.4 External domains

The roundcube app does not support adding domains that are not managed in Cloudron. Consider using Snappymail app (Rainloop fork) as an alternative.

7.150.5 Custom config

Custom config can be placed in `/app/data/customconfig.php` using the File Manager.

7.150.6 Vacation Email

An out of office / vacation mail message can be setup using Sieve filters.

A vacation message can be set in `Settings` -> `Filters` -> `Add filter` -> `Vacation message` action.

7.150.7 Forwarding all emails

To forward all emails to an external mail, setup a Sieve filter in `Settings` -> `Filters` -> `Add a filter` -> `Forward to`

7.150.8 Plugins

Plugins can be installed from plugin release tarballs. Please check plugin compatibility with your Roundcube version as several of the listed plugins are out-of-date and may break Roundcube.

- Upload the release tarball of the plugin using the File Manager into `/app/data/plugins` and extract it. After extract, you might need to prettify the directory name. For example, rename `roundcube-contextmenu-3.3` to just `contextmenu`. Make a note of this directory name since it required below.
- Change the ownership of the extracted plugin to `www-data` using the File Manager.
- Add the plugin to `$config['plugins']` in `/app/data/customconfig.php`. The name below must match the name of the directory in the first step.

```
array_push($config['plugins'], 'myplugin');
```

- Some plugin release tarballs do not contain dependancies and have to be installed via `composer`. `composer` requires a lot of RAM for it's resolution mechanism. For this reason, first bump the memory limit of the app to 2GB (you can reset back the memory limit after plugin installation). Open a Web Terminal and run the following:

```
# cd /app/data/plugins/<plugin>
# composer install --no-dev
# chown -R www-data:www-data .
```

Cannot edit root composer.json

The official plugin repository suggests installing plugins by editing the local `/app/code/composer.json`. This method of installation is not supported since code on Cloudron is read only.

Enabling PGP support

The Enigma plugin can be used to enable PGP support. The Enigma plugin is part of the roundcube code and no installation is required. To enable the plugin:

- Add the following lines to `/app/data/customconfig.php`:

```
array_push($config['plugins'], 'enigma');
$config['enigma_pgp_homedir'] = '/app/data/enigma';
```

- Create the directory where enigma will save the PGP keys on the server:

```
mkdir /app/data/enigma
chown www-data:www-data /app/data/enigma
```

* New PGP keys can be created or existing ones can be imported in `Settings` -> `PGP Keys`

- When composing new mail, you will see an Encryption icon in the tool bar.

7.150.9 Changing the title

Add the following lines to `/app/data/customconfig.php` using the File Manager:

```
$config['product_name'] = 'My Hosting Company';
```

7.150.10 Skins

Skins can be installed as follows:

- Extract the skin using the File Manager into `/app/data/skins`.
- Change the ownership of the extracted skin to `www-data`.
- Set the new skin as the default skin by adding this line in `/app/data/customconfig.php`:

```
$config['skin'] = 'newskin_directory_name';
```

Customizing CSS and logo

To customize CSS and logo, it's best to create a copy of an existing skin and make changes as needed.

- Open a Web terminal:

```
# cd /app/data/skins
# cp -Lr larry customskin
```

```
... make changes in customskin ...
# chown -R www-data:www-data customskin
```

- Set the new skin as the default skin by adding this line in `/app/data/customconfig.php`:

```
    $config['skin'] = 'customskin';
```

7.150.11 Search

By default, the search field only searches the current folder. To search across all folders, change the `Scope`.

7.150.12 Upload size

- To increase the upload size, edit `/app/data/customconfig.php`. Note that Roundcube only uses 1/3rd of the size below for the attachment.

```
    $config['max_message_size'] = '100M';
```

- Edit `/app/data/php.ini` accordingly:

```
    post_max_size = 75M
    upload_max_filesize = 75M
```

- Adjust the mail server's max mail size accordingly.

Be sure to restart the app after making the above changes.

7.151 RSS-Bridge App

7.151.1 About

RSS-Bridge is a PHP project capable of generating RSS and Atom feeds for websites that don't have one.

- Questions? Ask in the Cloudron Forum - RSS-Bridge
- RSS-Bridge Wiki
- RSS-Bridge issue tracker

7.151.2 Whitelist

RSS-Bridge supports whitelists in order to limit the available bridges on your web server. The Cloudron package whitelists all the available bridges, by default. To change this, edit `/app/data/whitelist.txt` using the File Manager.

7.152 Scrumblr App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.152.1 About

Scrumblr is a collaborative online scrum tool.

- Questions? Ask in the Cloudron Forum - AllTube
- AllTube Website
- Scrumblr issue tracker

7.153 SearXNG App

7.153.1 About

SearXNG is a privacy-respecting metasearch engine.

- Questions? Ask in the Cloudron Forum - SearXNG
- SearXNG Website
- SearXNG issue tracker

7.153.2 Custom configuration

Use the File Manager to edit `/app/data/settings.yml`. See this file for the possible options.

7.154 SerpBear App

7.154.1 About

SerpBear is an Open Source Search Engine Position Tracking App. It allows you to track your website's keyword positions in Google and get notified of their positions.

- Questions? Ask in the Cloudron Forum - SerpBear
- SerpBear Website
- SerpBear issue tracker

7.155 SFTPGo App

7.155.1 About

SFTPGo is a full-featured and highly configurable event-driven file transfer solution.

- Questions? Ask in the Cloudron Forum - SFTPGo
- Paid support is offered by the creator SFTPGo under plans

7.155.2 Custom config

Default Overwrites

Be aware, we are overwriting specific options with every startup of the app. This can be viewed in the start.sh.

Custom configuration can be put in `/app/data/sftpgo.json` using the Web Terminal or the File Manager.

More details for each option can be found in the official documentation: docs.sftpgo.com/2.6/config-file.

After changing the file, make sure to restart the app.

7.155.3 Reset the admin password and 2FA

Info

The following command will always reset the 2FA and the password for the admin.

In the Web Terminal run:

```
./sftpgo resetpwd --admin admin
```

You will be asked to provide a new password.

7.156 Shaarli App

7.156.1 About

The personal, minimalist, super-fast, database free, bookmarking service.

- Questions? Ask in the Cloudron Forum - Shaarli
- Shaarli Website
- Shaarli Docs
- Shaarli issue tracker

7.156.2 Browser Extensions

Shaarli's browser extensions currently do not work on Cloudron because Cloudron sets the `X-Frame-Options` as a security measure. A future version of Cloudron might support disabling this security measure.

7.156.3 Using the bookmarklet

- Open your Shaarli and Login
- Click the Tools button in the top bar
- Drag the `✚Shaare link` button, and drop it to your browser's bookmarks bar.

You can read more about setting up the bookmarklet in the Shaarli docs

7.157 Shiori App

7.157.1 About

Shiori is a simple bookmarks manager written in the Go language. Intended as a simple clone of Pocket.

- Questions? Ask in the Cloudron Forum - Shiori
- Shiori Website
- Shiori Issues
- Shiori Forum

7.158 SickChill App

7.158.1 About

SickChill is an automatic Video Library Manager for TV Shows.

- Questions? Ask in the Cloudron Forum - SickChill
- SickChill Website
- SickChill issue tracker

Configuration

You will have to enable NZB/Torrent providers. You may also want to enable post-processing.

Do not enable "Reverse proxy headers" : SickChill will block you from connecting, because it will believe that you are connecting from a remote address with no passwords, as it does not see Cloudron's authentication wall.

7.159 Simple Torrent App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.159.1 About

Simple Torrent is a self-hosted remote torrent client (rebranded from Cloud Torrent).

- Questions? Ask in the Cloudron Forum - Simple Torrent
- Simple Torrent Website
- Simple Torrent issue tracker

7.159.2 Customization

Use the File Manager and edit `cloud-torrent.yaml` for customization. See the docs on what can be customized. Some important fields are:

- `downloaddirectory` - where the download files are placed.
- `donecmd` - a script to run once file has downloaded. See DoneCmdUsage for more information.

7.160 SnappyMail App

7.160.1 About

SnappyMail is a simple, modern & fast web-based email client.

- Questions? Ask in the Cloudron Forum - SnappyMail
- SnappyMail Website
- SnappyMail issue tracker

7.160.2 Default Setup

SnappyMail is pre-configured for use with Cloudron Email. The app automatically generates domain configuration for all the apps that have email enabled at installation time. If you enable or disable email on one or more domains, simply reconfigure the app and it will re-generate the necessary configuration.

7.160.3 2FA

SnappyMail has 2FA support built-in, it requires activating the "Two Factor Authentication" plugin as admin.

Activate the admin panel as explained below, log in as admin, activate the plugin. Afterwards, users can activate and configure 2FA under Settings.

7.160.4 Plugins

Snappymail has several plugins that can be activated in the admin panel.

Rule of thumb (as for all software with plugins): Activate as little as possible for maximum stability.

7.160.5 Multi-domain Setup

There are two ways to use SnappyMail when using Cloudron Email with multiple domains.

- Users can login with their email and password to access their mailbox. If the Cloudron has two domains, `example1.com` and `example2.com`, the user can login using `user@example1.com` and `user@example2.com`. Aliases can be added as identities under SnappyMail settings.
- Users can login using one of the email domains and add the other domains using the `Add Account` dialog. For example, user can login as `user@example1.com` and add `user@example2.com` in the `Add Account` dialog.

Multiple accounts

SnappyMail tracks accounts based on the login email. This means that in the example above, if the user logs in `user@example2.com`, the `user@example1.com` id will not show up.

7.160.6 External domains

To add one or more external domains, add them in SnappyMail's admin panel.

7.160.7 Filters

SnappyMail has simple filters built-in (see Settings - Filters).

More complex filters are possible but require knowledge of the Sieve language to as explained e.g. in these tutorials:

- https://p5r.uk/blog/2011/sieve-tutorial.html
- https://docs.gandi.net/en/gandimail/sieve/sieve_tutorial.html
- https://www.fastmail.help/hc/en-us/articles/360060591373-How-to-use-Sieve

Vacation Email

An out of office / vacation mail message can be setup using Sieve filters.

```
require ["vacation"];
if true
{
    vacation :subject "Out of office" "Off to the alps!";
}
```

At most once a day

Vacation messages are sent at most once a day to the same sender. This setting cannot be changed.

Forwarding all emails

To forward all emails to an external mail, setup a Sieve filter as follows:

```
require ["copy","fileinto","vacation"];

if true
{
    redirect :copy "test@cloudron.io";
}
```

7.160.8 Admin panel

The admin panel is located at `/?admin` and is disabled by default.

To enable it:

- Open a File manager and edit the file `/app/data/_data_/_default_/configs/application.ini`. Set the value of `allow_admin_panel` to `On`.
- Visit `https://snappymail.domain.com/?admin` on your browser. When you visit this URL, the admin password is generated at `/app/data/_data_/_default_/admin_password.txt`.
- You can now login with the username `admin` and the password located at `/app/data/_data_/_default_/admin_password.txt`.
- The file `/app/data/_data_/_default_/admin_password.txt` may be deleted after you made a note of the password.
- Please change the password immediately.

Disable admin panel after use

We highly recommend disabling the admin panel after use.

7.160.9 Attachment size

To adjust the attachment size, change the PHP configuration and the SnappyMail configuration.

- For PHP configuration, edit `/app/data/php.ini` using the File manager and add

```
[PHP]
upload_max_filesize = 40M
post_max_size = 40M
```

- For SnappyMail configuration, edit `/app/data/_data_/_default_/configs/application.ini` using the File Manager and adjust `attachment_size_limit`.

Be sure to restart the app after making the above changes.

In addition to above, you also have to configure the mail server to allow large email.

7.161 Snipe-IT App

7.161.1 About

Snipe-IT is a free open source IT asset/license management system.

- Questions? Ask in the Cloudron Forum - Snipe-IT
- Snipe-IT Website
- Snipe-IT docs
- Snipe-IT issue tracker
- Snipe-IT Discord

7.161.2 Customization

Use the Web terminal to edit custom configuration under `/app/data/env`.

7.161.3 Full Company Support

Full Multiple Companies Support lets you set up Snipe-IT as a multi-tenant application. This features allows super-admins to restrict the assets non-super-admins can see. This feature is disabled by default but can be enabled from the General Settings page.

7.162 SOGo App

7.162.1 About

SOGo is a fully supported and trusted groupware server with a focus on scalability and open standards.

- Questions? Ask in the Cloudron Forum - SOGo
- SOGo Website
- SOGo support
- SOGo docs
- SOGo issue tracker

7.162.2 Login

SOGo only works with mailbox accounts on the Cloudron. Login using the full email address including the domain as the username.

7.162.3 Identities

SOGo is setup out of the box as an email client for Cloudron for all mailboxes on the Cloudron.

The full name is set from the Cloudron user profile. To change the name, you must change the name in Cloudron dashboard and restart SOGo. Without restarting SOGo, the name change does not get picked up by SOGo since it appears to cache the value in memory.

To add an identity, go to `Preferences` -> `Mail` -> `IMAP Accounts` -> `New identity`:

By default, user's cannot change their full name inside SOGo. To allow users to change their name, change `SOGoMailCustomFromEnabled` to `YES` in `/app/data/sogo.conf` using the File manager and restart the app. Please note that this variable already exists in the config file and creating duplicate entries will cause SOGo to not start up.

7.162.4 External domains

The SOGo app does not support adding domains that are not managed in Cloudron. Consider using the rainloop app as an alternative.

Sieve Scripts

SOGo UI only supports setting up a limited set of filtering rules. You can setup more advanced rules using the SnappyMail or Roundcube app.

7.162.5 CalDAV

SOGo supports syncing using CalDAV:

Clicking on the 'ribbon' next to the calendar shows a popup menu.

Clicking on `Links to this Calendar` will show the calendar settings for various clients.

Calendar URLs

CalDAV URL - https://sogo.example.com/SOGo/dav//Calendar/personal/
Wedav ICS URL - https://sogo.example.com/SOGo/dav//Calendar/personal.ics
WebDAV XML URL - https://sogo.example.com/SOGo/dav//Calendar/personal.xml

7.162.6 CardDAV

Clicking on the 'ribbon' next to the address book shows a popup menu.

Clicking on `Links to this Address book` will show the address book settings for various clients.

> **Address book URLs**
>
> CardDAV URL - https://sogo.example.com/SOGo/dav//Contacts/personal/

7.162.7 ActiveSync

Exchange ActiveSync is a protocol used by Microsoft Exchange to sync mobile devices. ActiveSync clients can fully synchronize contacts, emails, events and tasks with SOGo. Freebusy and GAL lookups are also supported, as well as "Smart reply" and "Smart forward" operations.

Cloudron SOGo package supports ActiveSync out of the box. It can be added in Window Mail using `Add an account` -> `Advanced Setup` and choosing `Exchange ActiveSync`. When adding the account:

- Use the mailbox as the email address. For example, `girish@cloudron.example`.
- Use the mailbox owner's password as the password.
- Use the mailbox as the username. For example, `girish@cloudron.example`.
- Use the mail domain as the Domain. For example, `cloudron.example`.
- Use the SOGo app domain as the Server. For example, `sogo.cloudron.example`.
- Make sure SSL is checked
- Use the mailbox as the Account name. For example, `girish@cloudron.example`.

See the forum thread for some screenshots.

> **Autodiscover**
>
> The account form can be filled up automatically if autodiscover.xml is set up.

Disabling ActiveSync

To disable ActiveSync , set `WOWorkersCount` to 0 in `/app/data/sogo.conf` using the File manager and restart the app.

Tuning ActiveSync

EAS support can be performance heavy. See SOGo configuration guide section 8 for fine tuning options. The suggestion configurion in that document for 100 users and 10 EAS devices is:

```
WOWorkersCount = 15;
SOGoMaximumPingInterval = 3540;
SOGoMaximumSyncInterval = 3540;
SOGoInternalSyncInterval = 30;
```

Make the necessary adjustments in `/app/data/sogo.conf` using the File manager and restart the app.

7.162.8 UI Issues

SOGo behaves differently depending on how you access the app. If you navigate to SOGo by clicking on the icon on your Cloudron dashboard, parts of the SOGo UI do not work.

This issue manifests itself as:

- Email delete button not working
- Compose email popup not closing. Sometimes, it ends up closing the tab itself.
- The browser's web inspector console displays a DOMException with the message `"Permission denied to access property \"$mailboxController\" on cross-origin object"`.

To workaround this, always use SOGo by opening a new browser tab and entering the SOGo domain name directly.

7.162.9 CalDAV and CardDAV Migration

Follow this guide to migrate CardDAV and CalDAV resources to and from existing installations.

7.162.10 Mark as spam

To mark emails as spam (or ham), click on the gravatar icon in the email header. Then, there is a thumbs down icon for marking as spam.

7.162.11 Cloudron user directory

The Cloudron user directory can be used in SOGo as a shared read-only address book.

To enable or disable this behavior, set `isAddressBook` to `YES` or `NO` in the `SOGoUserSources` section in the end of `/app/data/sogo.conf`:

```
...
    SOGoUserSources = ({
        type = ldap;
        ...
        isAddressBook = YES;
        displayName = Cloudron;
    });
...
```

The `displayName` can also be modified and is used in the UI to represent this addressbook.

After saving the file, the app needs to be restarted.

7.162.12 Reset 2FA

2FA for users can be enabled inside SOGo options. To reset 2FA, run the following command in the Web Terminal:

```
sogo-tool user-preferences set defaults your@email.com SOGoTOTPEnabled '{"SOGoTOTPEnabled":0}'
```

7.163 Statping App

⚠ **Discontinued**

Please note this app is not available anymore since upstream development has stopped.

7.163.1 About

Statping is a status Page for monitoring your websites and applications with beautiful graphs, analytics, and plugins.

- Questions? Ask in the Cloudron Forum - Statping
- Statping Website
- Statping issue tracker

7.163.2 Config changes

Sometimes after making config or setting changes, the graphs do not appear. To fix this, go to admin settings and clear the cache after doing any config or setting changes.

7.164 Stirling-PDF App

7.164.1 About

Stirling-PDF is a powerful based PDF manipulation tool using docker that allows you to perform various operations on PDF files, such as splitting merging, converting, reorganizing, adding images, rotating, compressing, and more.

- Questions? Ask in the Cloudron Forum - Stirling-PDF
- Stirling-PDF GitHub
- Stirling-PDF issue tracker
- Stirling Discord

7.164.2 Custom configuration

Stirling PDF allows easy customization using settings.yml.

Custom configuration can be changed in `/app/data/configs/settings.yml` using the File manager. Be sure to restart the app after making any changes.

7.164.3 Large PDF

Large PDF manipulation requires more memory. To fix, simply increase the memory limit of the app.

7.165 Superset App

7.165.1 About

Apache Superset is a modern data exploration and visualization platform.

- Questions? Ask in the Cloudron Forum - Superset
- Superset Website
- Superset issue tracker

7.165.2 Demo data

Demo data is a great way to get to know how superset works. To import open the Web terminal into the app and run the following commands:

```
source /app/pkg/env.sh
superset fab create-admin --username admin --firstname Superset --lastname Admin --email admin@cloudron.local --password changeme
superset load_examples
```

Time and memory consuming

Demo data takes many minutes and a lot of memory to get fetched and imported. For this set the memory limit of the app at least to 2Gb

7.165.3 Custom Config

Custom configuration can be placed in `/app/data/config/config_user.py`. For example:

```
FEATURE_FLAGS = {
  'SSH_TUNNELING': True
}
```

Be sure to restart the app after making any changes.

7.165.4 Sqlite

To enable Sqlite connections, add the following to `/app/data/config/config_user.py` using the File manager. Be sure to restart the app after making any changes.

```
PREVENT_UNSAFE_DB_CONNECTIONS = False
```

Be sure to restart the app after making any changes.

Sqlite SQLAlchemy path

SQLAlchemy's Sqlite path has 4 slashes for absolute paths. For example, `file:////app/data/sample/northwind.db`

7.165.5 RBAC

To configure which roles can access a dashboard, enable `DASHBOARD_RBAC` feature flag in `/app/data/config/config_user.py`.

```
FEATURE_FLAGS = {
    'DASHBOARD_RBAC': True
}
```

7.165.6 Public Dashboards

For publicly accessible dashboards, try a config like below. Please read the Superset docs to understand the security implications.

```
AUTH_ROLE_PUBLIC = 'Public'
PUBLIC_ROLE_LIKE = 'Gamma'

FEATURE_FLAGS = {
    'DASHBOARD_RBAC': True
}
```

7.165.7 User Role

The default role for a new user is administrator.

To change this, edit `/app/data/config/config_user.py` using the File manager and change `AUTH_USER_REGISTRATION_ROLE`. For example:

```
AUTH_USER_REGISTRATION_ROLE = "Public"
```

Be sure to restart the app after making any changes.

7.166 Surfer App

7.166.1 About

Surfer is a simple static file server.

- Questions? Ask in the Cloudron Forum - Surfer
- Surfer repo
- Surfer issue tracker

7.166.2 Admin page

The web interface is available under the `https://[appdomain]/_admin/` location.

7.166.3 Managing files

There are 4 ways to manage files in surfer:

- Web interface
- CLI tool
- WebDAV endpoint to manage files in your local file manager.
- SFTP

Web interface

Files and folders can be uploaded via the web interface located at `/_admin`.

CLI Tool

The Surfer cli tool can be installed using npm.

```
npm -g install cloudron-surfer
```

Login using an API access token created in the surfer admin interface:

```
surfer config --server <this app's domain> --token
```

Put some files:

```
surfer put index.html favicon.ico /
```

Put a directory (the `/*` below means that the contents of `build` dir get copied into the root of surfer. Without it, a `build` directory will get created in the root of surfer).

```
surfer put build/* /
```

To get help:

```
$ surfer
Usage: surfer [options] [command]

Options:
  -V, --version                output the version number
  -h, --help                   display help for command

Commands:
  login                        Set default server
  logout                       Unset default server
  config|configure [options]   Configure default server
  put [options] <file|dir...>  Uploads a list of files or dirs to the destination. The last argument is destination dir
  get [options] [file|dir]     Get a file or directory listing
```

```
    del [options] <file>          Delete a file or directory
    help [command]                display help for command
```

WebDAV

WebDAV is a well supported extension of the Hypertext Transfer Protocol that allows clients to perform remote Web content authoring operations. WebDAV shares can be mounted usually with your local file manager.

The URI schemes differ on the common platforms:

Platform	URI
Windows	`https://[appdomain]/_webdav/`
Mac	`https://[appdomain]/_webdav/`
Gnome	`davs://[appdomain]/_webdav/`
KDE	`webdavs://[appdomain]/_webdav/`

WebDAV Access

For WebDAV authentication, create an access token for a user, using the surfer admin UI, and use the access token as the password.

On Linux the Davfs2 library can also be used to locally mount a share:

```
mount -t davfs https://[appdomain]/_webdav/ /mount/point
```

SFTP

Files can be uploaded using an SFTP client like FileZilla. See SFTP access for login details.

SFTP Access

SFTP access for non-admin users can be granted using the access control UI.

7.166.4 Folder structure

By default, any `index.html` or `index.htm` file is served up. This file name can be changed in the `Settings` page.

If Public Folder Listing is enabled, the directory contents is listed, provided the directory has no index page.

If a `404.html` is present in the root directory, it will be served up for missing pages.

7.166.5 Access Control

Access to the site can be controlled from the `Settings` page. There are 3 options:

- Public (everyone) - anyone can view the site
- Password restricted - anyone with a password can view the site
- User restricted - only users with a Cloudron login can view the site

7.166.6 CI/CD integration

You can setup your CI/CD to automatically push static files to surfer using the CLI as follows:

- First, create an `Access Token` in surfer from the `Settings` menu.
- Install the surfer cli tool as part of the CI/CD pipeline.
- Push the artifacts (`dist/` in the example below):

```
surfer put --token api-7e6d90ff-5825-4ebe-a85b-a68795055955 --server surfer.cloudron.ml dist/* /
```

7.167 Synapse App

7.167.1 About

Matrix is an open network for secure, decentralized communication.

- Questions? Ask in the Cloudron Forum - Matrix Synapse
- Matrix Synapse Website
- Matrix Synapse issue tracker

7.167.2 Post installation

Step 1: Select Matrix IDs

Users on matrix has a unique universal id (just like email). These ids are of the form `@username:domain.com`.

To give users a "memorable id", this app (also known as home server) is pre-setup to use the second level domain for the domain part of the id (also known as the `server_name`). For example, if you installed the app at `synapse.example.com`, this app package will set the `server_name` to `example.com`. This will generate user ids of the form `@username:example.com`.

If you require a different server name, use a File Manager to edit `/app/data/configs/homeserver.yaml` and restart the app.

Step 2: Delegation

Matrix clients and servers discover Matrix servers using Well-Known URIs. The Well-Known URI is a document served up from the `server_name` domain (i.e `example.com`). It delegates the handling to the server installation (i.e `synapse.example.com`).

If `server_name` is an app hosted on Cloudron, you can use Cloudron's Well Known URI support to serve up well-known documents.

You can edit well known locations in the `Domains` view:

Clicking the button will open up a dialog where you can fill up well known locations:

> **Specify port 443 explicitly**
>
> The default matrix server port is 8448. However, the Synapse app on Cloudron uses port 443. For this reason, you must specify the port explicity, like `matrix.domain.com:443`.

To verify the delegation setup, try this command from your laptop/PC:

```
$ curl https://example.com/.well-known/matrix/server
{ "m.server": "synapse.example.com:443" }
```

> **Requires app on bare domain**
>
> In the above example, an app must be installed on the bare domain `https://cloudron.space` for Cloudron to be able to respond to well known queries.

Step 3. Federation

Federation setup is automatic. Use the Federation Tester to verify that everything is setup properly. Note you must enter the `server_name` (like `example.com`) in the form field in the website and NOT the location of your home server (despite what the form says).

7.167.3 Admin

To make an existing user an admin, open a Web terminal and run the following command:

```
PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE} -c "UPDATE users SET admin=1 WHERE name='@user:example.com'"
```

No Admin UI

Element has no admin UI (see this and this. The Admin user can use the admin API. For an Admin UI, see the section below.

7.167.4 Admin UI

Synapse does not have a built in admin UI. The Synapse Admin project providers a UI for Synapse configuration.

To use Synapse Admin:

- Install Surfer
- Upload the latest tarball from https://github.com/Awesome-Technologies/synapse-admin/releases into surfer
- Extract tarball
- Login with Synapse credentials. Note that the user you are trying to login with has to be Synapse admin. See [#admin] on how to make a user an admin.
- You can restrict the login to a specific matrix instance by editing `config.json` :

```
{
  "restrictBaseUrl": "https://homeserver.url"
}
```

7.167.5 Public room listing

By default an instance cannot be added for room discovery in other instances. To enable this feature to make public rooms discoverable, add the following to the config file at `/app/data/configs/homeserver.yaml` :

```
allow_public_rooms_over_federation: true
```

Afterwards the matrix synapse app has to be restarted. More details on that option can be found here.

7.167.6 Customizations

Synapse offers a variety of customizations. To make changes, use a File Manager to edit `/app/data/configs/homeserver.yaml` and restart the app.

7.167.7 Home page

The `index.html` can be customized by editing `/app/data/index.html` . Note that any assets have to be embedded inline.

7.167.8 Spaces

MSC1772: Matrix Spaces support can be enabled by editing `/app/data/configs/homeserver.yaml` using the File Manager and adding the following line:

```
experimental_features: { spaces_enabled: true }
```

Be sure to restart the app after making the change.

7.168 Syncthing App

7.168.1 About

Syncthing is a continuous file synchronization program. It synchronizes files between two or more computers in real time, safely protected from prying eyes.

- Questions? Ask in the Cloudron Forum - Syncthing
- Syncthing Website
- Syncthing forum
- Syncthing docs
- Syncthing issue tracker

7.168.2 Apps

Complete list of native GUIs and integrations is available here.

- Android app

7.168.3 Changing password

The Web UI password uses HTTP Basic Auth and can be changed from `Actions` dropdown -> `Advanced` -> `GUI`. Just enter the desired password in the `Password` input box. Syncthing will automatically hash the password and restart itself.

Note that there is no logout functionality. To test this new password, clear your browser cache.

7.168.4 CLI

The CLI can be used using the Web Terminal by running `/app/data/syncthing`.

Example:

```
/app/code/syncthing decrypt --to /tmp/unencrypted --password <secret> /app/data/folders/DirToDecrypt
```

7.169 Taiga App

7.169.1 About

Taiga is the project management tool for multi-functional agile teams.

- Questions? Ask in the Cloudron Forum - Taiga
- Taiga Website
- Taiga issue tracker

7.169.2 Custom configuration

Taiga customizations are placed in two files:

- `conf.json` - This contains UI settings. On the Cloudron app, this file is located at `/app/data/conf.json`.
- `local.py` - This contains backend settings. On the Cloudron app, this file is located at `/app/data/customlocal.py`. The default settings are listed in the common.py file.

These customizations will persist across updates and restarts.

To edit these files use the File manager.

7.169.3 Disabling external registration

When using Cloudron auth, external registration is already disabled. When now using Cloudron auth, edit the following files using File manager.

- `PUBLIC_REGISTER_ENABLED = False` in `/app/data/customlocal.py`
- `"publicRegisterEnabled": false` in `/app/data/conf.json`

7.169.4 Importing a project

An existing project's json can be imported into Taiga as follows:

- Connect to taiga using the Web terminal
- Upload the project.json using the 'Upload to /tmp' button
- `su cloudron`
- `source /app/code/taiga/bin/activate`
- `export DJANGO_SETTINGS_MODULE=settings.config`
- `cd /app/code/taiga-back`
- `python manage.py load_dump /tmp/project.json email@domain.tld`

Asana

To import projects from Asana:

- Create an Asana Connect client and secret. The redirect URL must be set to `https://taiga.example.com/project/new/import/asana`
- Edit the backend config at `/app/data/customlocal.py` to contain the key:

```
IMPORTERS["asana"] = {
    "active": True, # Enable or disable the importer
    "callback_url": "{}://{}/project/new/import/asana".format(SITES["front"]["scheme"],
                                                              SITES["front"]["domain"]),
    "app_id": "client id from above",
    "app_secret": "client secret from above"
}
```

- Edit `/app/data/conf.json`:

```
  "importers": [
    "asana"
  ]
```

- Restart the app

7.169.5 Plugins

Slack

The `slack` plugin is installed but not enabled. To enable it, add the following line to `/app/data/customlocal.py`:

```
INSTALLED_APPS += ["taiga_contrib_slack"]
```

Enable the frontend of Slack plugin by editing `/app/data/conf.json`:

```
    "contribPlugins": [ "/plugins/slack/slack.json" ]
```

Then, initialize the database by running the following command in the web terminal:

```
source /app/code/taiga/bin/activate
cd /app/code/taiga-back
python manage.py migrate taiga_contrib_slack
```

Be sure to restart the app for the changes to take effect.

7.170 Tandoor App

7.170.1 About

Manage your ever growing recipe collection online. Drop your collection of links and notes.

- Questions? Ask in the Cloudron Forum - Mealie
- Tandoor Website
- Tandoor issue tracker

7.170.2 Custom config

Custom configuration can be added in `/app/data/env` using the File manager. See upstream config file for the available options.

Be sure to restart the app after making any changes.

7.171 TeamSpeak App

7.171.1 About

Use crystal clear sound to communicate with your team mates cross-platform with military-grade security, lag-free performance & unparalleled reliability and uptime.

- Questions? Ask in the Cloudron Forum - Teamspeak Server
- Teamspeak Server Website
- Teamspeak Server forum

7.171.2 Initial Setup

After installation, check the app logs (in the `Console` section) to get the admin server token. You can now connect using one of the Teamspeak client.

On first time connect, the client with ask for a privilege key. This is the same as the admin server token.

7.171.3 License

To configure a Teamspeak license in an app instance, upload the license `.dat` file to `/app/data/licensekey.dat` using the web terminal or Cloudron cli tool, then restart the app from the dashboard.

7.172 Teddit App

7.172.1 About

Teddit is a free and open source alternative Reddit front-end focused on privacy.

- Questions? Ask in the Cloudron Forum - Teddit
- Teddit Website
- Teddit issue tracker

7.172.2 Customization

A variety of settings are customizable via config.js.

To customize, edit `/app/data/config.js` using the File manager and restart the app.

7.173 The Lounge App

7.173.1 About

The Lounge is a self-hosted web IRC client.

- Questions? Ask in the Cloudron Forum - The Lounge
- The Lounge Website
- The Lounge community
- The Lounge issue tracker
- The Lounge docs

7.173.2 User management

When installed with Cloudron SSO enabled, add and remove users in the Cloudron admin page.

When installed without Cloudron SSO enabled, new users must be added using the Lounge CLI tools.

- Open a Web terminal for the app.
- Use the lounge CLI command to add a user:

```
root@3543a0255d97:/home/cloudron# thelounge add girish
2017-10-28 05:21:59 [PROMPT] Enter password:
2017-10-28 05:22:02 [PROMPT] Save logs to disk? (yes)
2017-10-28 05:22:04 [INFO] User girish created.
2017-10-28 05:22:04 [INFO] User file located at /app/data/users/girish.json.
```

- To remove a user:

```
root@3543a0255d97:/home/cloudron# thelounge remove girish
2017-10-28 05:22:21 [INFO] User girish removed.
```

⚠ **Default admin user**

With SSO disabled, the Cloudron app creates a default user named 'admin' for convenience. Be sure to change the password in the Lounge setting's page. If you do not intend to use this user, you can delete this user.

7.173.3 Customization

Lounge supports various customizations. You can edit `/app/data/config.js` using the File manager to add customizations.

Be sure to restart the app after making any changes.

7.173.4 Installing themes

Lounge themes can be installed using the lounge CLI tool.

- First, look for a theme at npm
- Open a Web terminal for the app.
    - Run command `gosu cloudron:cloudorn thelounge install thelounge-theme-custom`
- Restart the app
- Select theme in options

7.174 TLDraw App

7.174.1 About

A tiny little drawing app.

- Questions? Ask in the Cloudron Forum - TLDraw
- TLDraw Website
- TLDraw issue tracker

7.175 Tiny Tiny RSS

7.175.1 About

Tiny Tiny RSS is a free and open source web-based news feed (RSS/Atom) reader and aggregator

- Questions? Ask in the Cloudron Forum - Tiny Tiny RSS
- Tiny Tiny RSS Website
- Tiny Tiny RSS forum

7.175.2 Customizing configuration

Use the File Manager to place custom configuration under `/app/data/env.sh`.

Please be careful when changing configuration since the Cloudron packaging might depend on it.

7.175.3 Installing Plugins

To install plugins, simply extract them using the File Manager to `/app/data/plugins.local` and restart the app.

To enable the plugin globally, you must edit the `TTRSS_PLUGINS` variable in `/app/data/env.sh`. Alternately, each user can enable the plugin individually under Preferences.

You can see list of plugins here

7.175.4 Installing themes

To install themes, simply extract them using the File Manager to `/app/data/themes.local` and restart the app.

Some suggested themes:

- Reeder
- Clean GReader
- Feedly

7.175.5 Fever support

TinyTinyRSS supports Fever API using a plugin. There are many version of fever API floating around but this version is known to work.

7.175.6 External registration

To enable external registration, make two changes to `/app/data/env.sh` using the File Manager:

- Set `TTRSS_ENABLE_REGISTRATION` to `true`
- Edit the `TTRSS_PLUGINS` variable to include `auth_internal`. Note that `TTRSS_PLUGINS` is comma separated plugin names. This enables auth via the internal database authentication in addition to LDAP that Cloudron setup.

7.175.7 Migrating from mysql to postgresql

The current latest package has moved to postgresql since upstream has deprecated mysql version of TinyTinyRSS. We will maintain the mysql flavor for the time being but users should migrate to the new package with postgresql at some point.

For this follow the below steps:

- Export RSS feeds and settings from the old instance at Settings -> Feeds -> OPML -> export
- Install a new instance of TinyTinyRSS
- Log into the new instance if Cloudron user management is enabled, or create a new account otherwise
- Import the exported OPML file
- Wait for at least 15 min to have all feeds synced. They may report as all unread unfortunately.

7.176 Traccar App

7.176.1 About

Traccar is an open source GPS tracking system.

- Questions? Ask in the Cloudron Forum - Traccar
- Traccar Website
- Traccar docs
- Traccar forums
- Traccar issue tracker

7.176.2 Admin Password

If you forgot the admin password, it cannot be reset by email.

From the comment here, you can reset the password to `password` by updating the database directly.

Open a Web terminal and click the PostgreSQL button on the top bar and click enter. In the prompt, execute

```
dbb969ebe64f58473ebed981bb1b69d64c=> UPDATE tc_users SET hashedpassword='ef38a22ac8e75f7f3a6212dbfe05273365333ef53e34c14c',
salt='000000000000000000000000000000000000000000000000' WHERE name='admin';
UPDATE 1
```

7.176.3 Custom config

Traccar provides a wide variety of configuration options like:

- Notifications
- SMS Service
- Reverse Geocoding

They can be placed in `/app/data/traccar.xml` using the File manager.

7.176.4 Registration

Registration is disabled by default. This can be enabled in `Settings` -> `Server` -> `Permissions` -> `Registration`.

7.176.5 Ports for Devices

Traccar can work with a large variety of devices and protocols but often the ports required for this are hardcoded. A full list of devices and ports can be found here. If further ports are required, please let us know in the forum and we will add those.

7.177 Transmission App

7.177.1 About

Transmission is a fast, easy and free BitTorrent client.

- Questions? Ask in the Cloudron Forum - Transmission
- Transmission Website
- Transmission forum
- Transmission issue tracker

7.177.2 Download Path

By default, this app is configured to download files into `/app/data/downloads`. You can change this path to a volume by editing `/app/data/config/settings.json` and changing `download-dir`.

7.177.3 Post Processing Script

There is a sample post processing script in `/app/code/transmission/torrentDone.js` . This script, at the end of a download, automatically hard-link the downloaded files from `/app/data/files/Downloading/` to `/app/data/files/Downloaded/`. This allows potential post-processing by other apps of the file in `Downloaded`, without interfering with seeding from the file in `Downloading`. Also, it seeds torrents up to a ratio of 2, then automatically remove them, and removes the corresponding files from `Downloading`.

The script can be enabled by changing `script-torrent-done-enabled` in in custom config.

You can also write your own post processing scripts and set `script-torrent-done-filename` accordingly. See https://github.com/transmission/transmission/blob/main/docs/Scripts.md for more information.

7.177.4 Custom config

Custom configuration can be edited in `/app/data/config/settings.json` using the File Manager. Be sure to restart the app after making changes.

See transmission docs for a full list of configurable options.

7.178 TriliumNext App

7.178.1 About

TriliumNext Notes is a hierarchical note taking application with focus on building large personal knowledge bases.

- Questions? Ask in the Cloudron Forum - TriliumNext Notes
- TriliumNext Matrix
- TriliumNext Github Discussions
- TriliumNext Wiki

7.178.2 Multiple users

TriliumNext does not support multiple users. It's a single user application intended for personal notes. See the FAQ for more information.

7.178.3 Usage

It can be a bit confusing but the app is both a sync server and an online web taking app.

7.179 Typebot App

7.179.1 About

Typebot is a visual chatbot builder that helps you create chatbots for your website without coding.

- Questions? Ask in the Cloudron Forum - Typebot
- Typebot Website
- Typebot Issues

7.179.2 Custom Configuration

Typebot supports various configuration parameters that can be set as environment variables. You can edit `/app/data/env.sh` using the File Manager to set custom configuration.

For example:

```
export NEXT_PUBLIC_UNSPLASH_APP_NAME = Whatever_Name_I_Gave_This_App_On_Unsplash
export NEXT_PUBLIC_UNSPLASH_ACCESS_KEY = theRandomStringThatUnsplashGaveAsAnAPIkey
```

Be sure to restart the app after making any changes.

7.179.3 Media Uploads

Typebot requires S3 compatible storage access to store media uploads.

An example configuration using minio is described below.

- Install minio and create a bucket named `typebot`.
- Configure the bucket as outlined in typebot docs:
    - In the bucket's policy, add an `Anonymous` access rule for the prefix `public/`.
    - minio allows CORS requests from all endpoints by default. Thus, `CORS policy` setting can be skipped
- Create access keys for bucket access in minio
- Set the access keys for typebot by editing `/app/data/env.sh` using the File manager:

```
export S3_ACCESS_KEY=rEbfwwqVZvMfBko2GuGx
export S3_SECRET_KEY=o69OUZsJD2bv1SBh0eTRJVSrtOu2VqrlxUau99Ye
export S3_BUCKET=typebot
export S3_ENDPOINT=minio-api.cloudron.example
export S3_SSL=true
export S3_REGION=us-east-1
```

- Restart Typebot

7.180 Umami App

7.180.1 About

Umami is a simple, fast, privacy-focused alternative to Google Analytics.

- Questions? Ask in the Cloudron Forum - Umami
- Umami Website
- Umami docs
- Umami issue tracker

7.180.2 Admin password

To change the password of a user, open a Web Terminal and click the PostgreSQL button at the top. This will paste the PostgreSQL connection command line, just press enter.

> ⚠ **Replace the username**
>
> The command below resets the password of `admin` to `umami`.

```
root@0d0dbccd-43ee-43c0-b1a4-80613567bd6a:/app/code# PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p $
{CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE}
psql (16.6 (Ubuntu 16.6-0ubuntu0.24.04.1), server 16.8 (Ubuntu 16.8-0ubuntu0.24.04.1))
Type "help" for help.

db0d0dbccd43ee43c0b1a480613567bd6a=> UPDATE "user" SET password='$2b$10$BUli0c.muyCW1ErNJc3jL.vFRFtFJWrT8/GcR4A.sUdCznaXiqFXa' WHERE username='admin';
UPDATE 1
```

7.180.3 Custom config

Custom configuration can be added to `/app/data/env.sh` using the File manager.

Be sure to restart the app after adding any custom configuration.

7.180.4 Custom endpoint

To set a custom script name edit `/app/data/env.sh` using the File manager:

```
export TRACKER_SCRIPT_NAME=clickcounter
```

To set a custom API endpoint:

```
export COLLECT_API_ENDPOINT=/api/counter
```

Be sure to restart the app after making any changes.

7.181 Uptime Kuma App

7.181.1 About

Uptime Kuma is a self-hosted monitoring tool like "Uptime Robot".

- Website
- Issue tracker

7.181.2 Custom config

Custom environment variables can be set by editing `/app/data/env` using the File manager. Be sure to restart the app after making any changes.

Example:

```
export UPTIME_KUMA_DISABLE_FRAME_SAMEORIGIN=true
```

7.181.3 Reset Password

To reset the password, open a Web Terminal and run:

```
# gosu cloudron:cloudron npm run reset-password

> uptime-kuma@1.23.16 reset-password
> node extra/reset-password.js

== Uptime Kuma Reset Password Tool ==
Connecting the database
2025-01-01T13:55:10+00:00 [DB] INFO: Data Dir: ./data/
Found user: girish
New Password:
```

7.181.4 Remove 2FA

To remove the 2FA, open a Web Terminal and run:

```
# gosu cloudron:cloudron npm run remove-2fa
```

7.182 Valheim App

7.182.1 About

This app packages the Valheim Dedicated Server

- Questions? Ask in the Cloudron Forum - Valheim
- Valheim Game Website

7.182.2 Mod Support

If `MODSUPPORT` is enabled, all files and folders from `/app/data/mods/` will by synced to `/run/vhserver-steam/`.

Place your Mod `*.dll` files in `/app/data/mods/BepInEx/plugins/` and the Mod Config files in `/app/data/mods/BepInEx/config/`

Mod Example ValheimPlus

The UnixServer.zip of ValheimPlus has the same folder structure which cold be extracted into `/app/data/mods/` but this would also overwrite the cloudron auto installed version of BepInEx. If you make a mistake here, don't worry. Simply clear the `/app/data/mods/` folder and it will be recreated after a app restart.

7.183 Vault App

7.183.1 About

Vault is a tool for secrets management, encryption as a service, and privileged access management.

- Questions? Ask in the Cloudron Forum - Vault
- Vault Website
- Vault community
- Vault issue tracker

7.183.2 Setup

On first visit, you can generate the keys for the vault by specifying the number of shares and the threshold. The idea is that a master key is generated and split into the number of shares. To unlock the vault, you must provide at least threshold count of the parts. See this blog for a full explanation on how Vault uses Shamir's secret sharing algorithm.

7.183.3 Unsealing

Vault stores all the secrets in memory. It uses the mlock capability of the kernel to prevent swapping. When the app restarts or the server restarts, it must be unsealed using the keys that were generated during setup. This is by Vault design.

7.183.4 OIDC

After the Vault is unsealed, you can run the following command to enable Cloudron authentication.

```
/app/pkg/enable-oidc.sh <root-token>
```

The root token is part of the keys generated during setup.

7.184 Vaultwarden App

7.184.1 About

Bitwarden is an Open Source Password Management solution for individuals, teams, and business organizations. Vaultwarden is an unofficial Bitwarden compatible server written in Rust, fully compatible with the client apps.

- Questions? Ask in the Cloudron Forum - Vaultwarden
- Vaultwarden Website
- Vaultwarden issue tracker

7.184.2 Users

Bitwarden does not support Single Sign On. This is by design for security reasons. You must create a new password for your Bitwarden account.

By default, open registration is enabled. This can be changed via the config variables by editing `/app/data/config.json` using the File Manager. For example, to disable signup but allow invitations set the variables as below:

```
  "signups_allowed": false,
  "invitations_allowed": true,
```

7.184.3 Admin

The admin UI is located `/admin`. To login, look for the `admin_token` field inside `/app/data/config.json` using the File manager.

Starting with version 1.28, it is safer to generate an admin token using the built-in hash feature. Otherwise, warnings might be shown in the logs - `Please generate a secure Argon2 PHC string by using vaultwarden hash or argon2`.

To fix, open a web terminal and run:

```
# /app/code/vaultwarden hash
Generate an Argon2id PHC string using the 'bitwarden' preset:

Password:
Confirm Password:

ADMIN_TOKEN='$argon2id$v=19$m=65540,t=3,p=4$RCpl3a+FItyn4KBJVAtZ+EyP9+fK0hoRqqo9jEdyRJE$d7UfKfZYsZJad6OIKpzPtO2o2ccLkrHjEi5jXdWWkO0'
```

Take the above `ADMIN_TOKEN` and put it in `/app/data/config.json` in the field `admin_token`.

Important:

- Remove the single quotes around the argon2id string above.
- The token to login to the admin page is the password you entered above to generate the `ADMIN_TOKEN`.
- `config.json` should be edited like below (please be careful about the quoting):

```
  "admin_token": "$argon2id$v=19$m=65540,t=3,p=4$RCpl3a+FItyn4KBJVAtZ+EyP9+fK0hoRqqo9jEdyRJE$d7UfKfZYsZJad6OIKpzPtO2o2ccLkrHjEi5jXdWWkO0"
```

Restart the app and verify if token actually changed.

7.184.4 Custom config

Custom environment variables can be set in `/app/data/env.sh` using the File Manager.

Note that Vaultwarden's admin page generates `config.json` which overrides the above env vars. See config docs for more information on which values are readonly and can only be set using environment variables.

7.185 Verdaccio App

7.185.1 About

Verdaccio is a lightweight open source private npm proxy registry.

- Questions? Ask in the Cloudron Forum - Verdaccio
- Verdaccio Website
- Verdaccio issue tracker

7.185.2 Custom configuration

You can add custom verdaccio configuration using the File Manager:

- Add any custom configuration in `/app/data/config.yaml`.
- Restart the app

See config.yaml for reference.

7.185.3 Access control

See the Verdaccio docs on how to allow or restrict access to packages.

7.185.4 CLI Authentication

The package use OIDC authentication and providing the raw username and password to the npm/yarn/pnpm CLI tools will not work.

There are two ways to set up CLI authentication:

- `npx verdaccio-openid@latest --registry http://your-registry.com`

You can then use the above credentials in `npm adduser --registry https://your-registry.com/`

- Login to the Web UI . Settings -> Configuration

7.185.5 Publishing packages

To publish a package:

```
npx verdaccio-openid@latest --registry http://your-registry.com
npm publish --registry https://your-registry.com
```

EPUBLISHCONFLICT

Verdaccio is a proxying package manager. Packages are checked against npmjs registry before being published. For this reason, publishing packages that already exist on npmjs will result in a `EPUBLISHCONFLICT` error. See GitHub issue 1203 for a workaround.

7.185.6 Installing packages

To set the custom registry globally:

```
npx verdaccio-openid@latest --registry http://your-registry.com
npm set registry https://your-registry.com
```

To set it on just the project:

```
npx verdaccio-openid@latest --registry http://your-registry.com
npm install my-package --registry https://your-registry.com
```

7.186 Vikunja App

7.186.1 About

Vikunja is the to-do app to organize your life.

- Questions? Ask in the Cloudron Forum - Vikunja
- Vikunja Website
- Vikunja community

7.186.2 Config

Custom configuration can be set using the File manager by editing `/app/data/config.yml`. The various options are documented here.

7.187 VPN App

7.187.1 About

VPN app allows you to create a trusted and secure private network. It supports both OpenVPN and WireGuard.

- Questions? Ask in the Cloudron Forum - OpenVPN
- OpenVPN Website
- WireGuard Website
- Upstream OpenVPN forum

7.187.2 Desktop and Mobile Clients

The OpenVPN app has been tested with the following clients:

- NetworkManager on Ubuntu
- Tunnelblick on Mac OS X
- OpenVPN for Android
- WireGuard for Android
- OpenVPN for Mac

7.187.3 Configs

The OpenVPN config can be downloaded using the download button. There are a couple of formats:

- `OpenVPN` - For linux and OpenVPN app on Android
- `Tunnelblick` - For Tunnelblick
- `WireGuard` - For WireGuard

7.187.4 How to connect on Ubuntu

Download the .ovpn embedded certs config file from the OpenVPN app. Then, in the Network settings in Ubuntu, click on 'Add VPN' and then `Import from file...`.

7.187.5 Admin Settings

The first user to login is made the admin.

The admin panel can be used to customize some of the popular settings like the network address, DNS server and client-to-client connectivity.

To make another user an admin, edit the file `/app/data/vpn.db` and edit the `admins` field. This is a JSON database, so be careful with quotations.

7.187.6 Built-in DNS Server

This app has a built-in Dnsmasq DNS server which is pushed to clients. This DNS server allows resolution of connected clients using `devicename.username`.

You can configure this DNS server by editing `/app/data/dnsmasq.conf` using the File manager. For example, to make Dnsmasq forward DNS requests to an internal DNS server, use the following:

```
server=internal-server-ip
```

See Dnsmasq docs for all the available options.

7.187.7 Custom DNS Server

You can configure the VPN to use a custom DNS server - for example, an external AdGuard or a Pi-Hole installation. Simply, put in the IP of the DNS server in the Admin Settings. In the screenshow below, `104.207.150.252` is the AdGuard DNS Server IP.

7.187.8 Privacy

The VPN app provides a tunnel to channel all the traffic from your devices via the Cloudron. Websites and services that you visit will not see the IP address of your devices but they will see the IP address and possibly the RDNS hostname of your Cloudron.

You can check what sort of information can be gathered from your Cloudron's IP address using ipleak.net.

7.187.9 IPv6

Transport

If you have enabled IPv6 support on Cloudron, Clients can connect (the "transport" protocol) to the OpenVPN server via IPv6. Whether the connection uses IPv4 or IPv6 is a client OS preference.

Inside Tunnel

IPv6 is supported inside the tunnel (independent of the transport protocol). The server needs to have an IPv6 address assigned for IPv6 tunneling to work.

7.187.10 OpenVPN

TCP/UDP

OpenVPN can be configured to use TCP or UDP. TCP mode uses more bandwidth and slower than UDP.

The UDP port or TCP port can be configure the app's Location section. If UDP is enabled, TCP is automatically disabled.

OpenVPN Customization

You can customize various settings by editing `/app/data/openvpn.conf` using the File Manager. Some popular options are discussed below:

Custom OpenVPN routes

By default, clients are configured to route all traffic via the VPN. If you disable this, you would want to push custom routes for the network and hosts behind the VPN. For example, edit the file as below and restart the app.

```
# push "redirect-gateway def1 bypass-dhcp"
push "route 178.128.183.220 255.255.255.255"
push "route 178.128.74.0 255.255.255.0"
```

Custom Client Configuration

Custom Client Configuration allows the OpenVPN admin to assign a specific IP address to a client or push specific options such as compression and DNS server to a client.

To add custom settings:

- Edit `/app/data/openvpn.conf` and add the following line:

```
client-config-dir /app/data/ccd
```

- Create the directory `/app/data/ccd`
- You can create custom client configs in this directory by creating files with the name `[username]_[devicename]`. You can also create a file named `DEFAULT` which will be used if no device specific file exists.
- For example, to assign a static IP to a client, you can add the line `ifconfig-push 10.8.0.50 10.8.0.51` (requires IP pair)
- Restart the app for changes to take effect.

Concurrent Connections

By default, every connection must have one unique cert. With `duplicate-cn`, one cert can be used by more than one connection.

Enable this if multiple clients with the same VPN config can concurrently connect:

```
duplicate-cn
```

7.187.11 Troubleshooting

If you are unable to connect to the OpenVPN server, make sure that your VPS firewall allows the OpenVPN/WireGuard ports. For example, you might have to add this incoming port as part of EC2 security group.

7.188 Wallabag App

7.188.1 About

Wallabag is a self hostable application for saving web pages: Save and classify articles. Read them later. Freely.

- Questions? Ask in the Cloudron Forum - Wallabag
- Wallabag Website
- Wallabag docs
- Wallabag issue tracker

7.188.2 Using the Browser Extensions

- Firefox
- Chrome

Follow the instructions in the wallabag website to configure the extension to use the Cloudron app.

7.188.3 Admin

To make an existing user an admin, first make note of the user's id (100 in the example below). Then, run the following PostgreSQL command using the Web Terminal:

```
UPDATE wallabag_user SET roles = 'a:2:{i:0;s:9:"ROLE_USER";i:1;s:16:"ROLE_SUPER_ADMIN";}' where id = 100;
```

7.188.4 Custom CSS

Following https://doc.wallabag.org/en/admin/custom_css.html the stylesheet can be adjusted by editing `/app/data/custom.css`.

7.188.5 Import

Wallabag supports importing from various sources. To run an importer, open a Web Terminal and run:

```
# /usr/local/bin/gosu www-data:www-data php bin/console wallabag:import:redis-worker --env prod <source>
```

`source` is one of: * pocket * readability * pinboard * instapaper * wallabag_v1 * wallabag_v2 * firefox * chrome

7.189 Wallos App

7.189.1 About

Wallos is an Open-Source Personal Subscription Tracker.

- Questions? Ask in the Cloudron Forum - Wallos
- Wallos Website
- Wallos issue tracker

7.190 WBO App

7.190.1 About

WBO is an online collaborative whiteboard that allows many users to draw simultaneously on a large virtual board.

- Questions? Ask in the Cloudron Forum - WBO
- WBO Website
- WBO issue tracker

7.190.2 Custom config

Custom config can be set in `/app/data/env` using the File manager. Be sure to reboot the app after making changes.

7.191 Weblate App

7.191.1 About

Weblate is a libre web-based translation tool with tight version control integration.

- Questions? Ask in the Cloudron Forum - Weblate
- Weblate Website
- Weblate docs
- Weblate issue tracker

7.191.2 Custom Config

Custom configuration can be set in `/app/data/custom_settings.py` using the File Manager.

Python configuration file

Please be aware that the configuration file is in Python. Python is sensitive to whitespace and indentantion. Check your configuration file syntax if Weblate is not applying your changes.

Example:

```
REGISTRATION_OPEN = False
```

Be sure to restart the app after making any changes.

7.191.3 Celery Worker Options

Weblate has a few celery worker for background processing. Depending on the allocated resources and available CPUs it may be required to adjust the worker options. Those can be set by editing the file at `/app/data/celery.env` :

The default file contains one env variable per worker:

```
export CELERY_MAIN_OPTIONS=""
export CELERY_NOTIFY_OPTIONS=""
export CELERY_TRANSLATE_OPTIONS=""
export CELERY_BACKUP_OPTIONS=""
export CELERY_BEAT_OPTIONS=""
```

To set for example the concurrency for one worker, adjust the corresponding line like:

```
export CELERY_MAIN_OPTIONS="--concurrency 16"
```

Then restart the app. If the app does not start up again, take a look at the logs to see if an option may not be accepted for a worker.

7.192 Wekan App

7.192.1 About

Wekan is an open source Kanban board.

- Questions? Ask in the Cloudron Forum - Wekan
- Wekan Website
- Wekan chat
- Wekan issue tracker

7.192.2 API

Wekan has a REST API for managing users.

7.192.3 Webbooks

When a webhook is activated, Wekan sends the related information within the POST request body. See Webhook-data for details.

7.192.4 Users

Wekan currently does not support syncing or searching users via LDAP. This limitation means that a user must login to Wekan first before they become available for sharing boards.

7.192.5 Admin

To make a Cloudron user an admin, first make sure the user has logged once into Wekan (this creates the user in Wekan). Then, run the following command after replacing `ADMIN_USERNAME` below in the Web Terminal:

```
# mongosh -u "${CLOUDRON_MONGODB_USERNAME}" -p "${CLOUDRON_MONGODB_PASSWORD}" ${CLOUDRON_MONGODB_HOST}:${CLOUDRON_MONGODB_PORT}/${CLOUDRON_MONGODB_DATABASE}
--eval "db.users.update({ username: 'ADMIN_USERNAME' }, { \$set: {isAdmin: true } })"

MongoDB shell version v3.6.3
connecting to: mongodb://mongodb:27017/b52e1fdb-5b2d-417e-a41c-53d060a97141
MongoDB server version: 3.6.3
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

7.192.6 Registration

External user registration is enabled by default. User registration can be disabled from the Wekan's Admin Panel. The Wekan Admin Panel option is only visible when the Cloudron user is a Wekan administrator.

7.192.7 Customization

Custom env vars can be added in `/app/data/env` using the File Manager. Be sure to restart the app after editing this file.

7.193 Wiki.js App

7.193.1 About

Wiki.js is a wiki engine running on Node.js

- Questions? Ask in the Cloudron Forum - Wiki.js
- Wiki.js Website
- Wiki.js issue tracker

7.193.2 Permissions

When authenticating against the Cloudron Directory, users only have the `Guest` permission (read pages) by default. Because of this, it might appear that users cannot login.

You can adjust the default permissions of Cloudron users as follows:

- Login as admin
- Create a Group with necessary permissions
- `Authentication` -> `Cloudron` -> `Registration` . Add the group created above to the auto assign groups.

7.193.3 Git Storage

Wiki.js supports storing documentation in git storage using the Git module.

Generate SSH Keys

You can skip this section entirely, if you already have existing SSH keys.

- Open a Web Terminal and generate the keys in `/app/data/ssh/id_rsa` (see below). Be sure to leave the passphrase empty as required by Wiki.js.

```
# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /app/data/ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /app/data/ssh/id_rsa
Your public key has been saved in /app/data/ssh/id_rsa.pub
...

# chown -R cloudron:cloudron /app/data/ssh
```

- Add `/app/data/ssh/id_rsa.pub` as SSH public key into GitHub/GitLab/Gitea/Gogs.

Create repository directory

Open a Web Terminal and run the following commands:

```
# mkdir -p /app/data/repo
# chown -R cloudron:cloudron /app/data/repo
```

Configure Git storage

- Inside Wikijs, go to `Modules` -> `Storage` and select `Git` .
- If you generated the SSH keys as per the instructions above, set the `SSH Private Key Mode` to `path` and set `SSH Private Key Path` to `/app/data/ssh/id_rsa` . Alternately, you can also set this to `contents` and paste in your existing SSH private key.
- Set the Local Repository Path to `/app/data/repo` (created above)

- Newer GitHub repositories use the branch name as `main` instead of `master`. Use the appropriate branch name in Wiki.js config page.

7.194 The Woodpecker App

7.194.1 About

Woodpecker is a simple CI engine with great extensibility. It runs your pipelines inside Docker containers.

- Questions? Ask in the Cloudron Forum - Woodpecker
- Woodpecker Website
- Woodpecker docs
- Woodpecker issue tracker

7.194.2 Authentication

Woodpecker authenticates against your existing version control system (Github/Gitea/Gogs/GitLab etc). See the upstream docs for the supported providers.

For example, to use Gitea for authentication, create an OAuth2 application inside Gitea and edit `/app/data/env.sh` like so:

```
export WOODPECKER_GITEA=true
export WOODPECKER_GITEA_URL=https://gitea.cloudron.example
export WOODPECKER_GITEA_CLIENT=1950a7f0-74e1-4003-8d51-6253eb673940
export WOODPECKER_GITEA_SECRET=gto_kgoizrbde5qv242q2waw7hocb5sa5snafqhwoeownflb4lml7dsa
```

Be sure to restart the app after making any changes.

⚠ **Only one auth provider**

Only one auth provider can be active at a time. For example, if you enabled Gitea, make sure all other values like `WOODPECKER_GITLAB` are removed or set to false.

7.194.3 Agent

Agents are workers that execute the pipelines. Agents must be installed in a separate VM.

Agent binaries are available here. A shared secret token is required to run the agent. This token is automatically generated and located at `/app/data/env.sh`. Please use the File manager to make a note of this value and use it in the `WOODPECKER_AGENT_SECRET` variable below.

Documentation of agent variables can be found here.

To run the agent using Docker:

```
docker run --name=woodpecker-agent --restart=always --detach \
    -e WOODPECKER_SERVER="woodpecker.cloudron.example:9000" \
    -e WOODPECKER_MAX_WORKFLOWS=4 \
    -e WOODPECKER_GRPC_SECURE=true \
    -e WOODPECKER_LOG_LEVEL=info \
    -v /var/run/docker.sock:/var/run/docker.sock \
    -e WOODPECKER_BACKEND=docker \
    -e WOODPECKER_AGENT_SECRET="see /app/data/env.sh for the value" \
    woodpeckerci/woodpecker-agent:latest
```

You can view the agent logs using `docker logs -f woodpecker-agent`.

```
# docker logs -f woodpecker-agent
{"level":"debug","time":"2023-04-27T09:06:42Z","message":"loaded docker backend engine"}
{"level":"debug","time":"2023-04-27T09:06:42Z","message":"request next execution"}
```

⚠ **Do not install agent on Cloudron VM**

Do not install the agent on Cloudron server itself. This is dangerous because the agent has full access to docker and it can (accidentally) delete or corrupt your apps.

7.194.4 Customization

To customize, edit `/app/data/env.sh` using the File manager. Be sure to restart the app after making any changes.

See upstream docs for all the server configuration options.

7.194.5 Registration

By default, users from the configured authentication provider can login to Woodpecker. If you want to prevent new users from signing in, you can disable this by setting `export WOODPECKER_OPEN=false` in `/app/data/env.sh`.

7.195 WordPress (Developer) App

7.195.1 About

This app is targeted at users who want to have complete control over their WordPress installation.

Features:

- The WordPress code can be modified. This means that you have to manage updates to WordPress yourself using WordPress' built-in updater.
- Custom Apache configuration via `.htaccess`
- Multisite support

If you prefer delegating the responsibility of applying updates to the Cloudron team, use the WordPress Managed app instead.

- Questions? Ask in the Cloudron Forum - WordPress (Developer)

7.195.2 Admin page

The WordPress admin page is located `https://<my.example.com>/wp-login.php`.

7.195.3 Using SFTP

The app can be uploaded using an SFTP client like FileZilla.

You can find the SFTP login details when clicking on the `i` icon in the app grid.

SFTP Access

SFTP access for non-admin users can be granted using the access control UI.

7.195.4 Memory limits

To adjust memory allocated for WordPress, edit `/app/data/wp-config.php` using the File manager and add the following line at the end of the file:

```
define('WP_MEMORY_LIMIT', '128M');
define('WP_MAX_MEMORY_LIMIT', '256M');
```

Note that the app also has a separate memory limit controlled by the app's memory limit. If you increase `WP_MEMORY_LIMIT`, be sure to increase the app's memory limit. A good formula is to provide the app 6 times the `WP_MEMORY_LIMIT` value at the bare minimum.

`WP_MAX_MEMORY_LIMIT` is the limit for administration tasks, which often require more.

A detailed explanation can be found in the WordPress docs.

7.195.5 Apache Config

Apache configuration can be tweaked using the `htaccess` mechanism. By default, the app does not have an `.htaccess` file. It can be added via SFTP or the File manager at `/app/data/public/.htaccess`. Like any other standard Apache setup, `.htaccess` can also be added on other WordPress subfolders in `/app/data/public/` as needed.

7.195.6 Cron tasks

The app is configured to run WordPress cron tasks every minute.

To run the cron tasks manually run the following command using the Web terminal:

```
wp cron event run --due-now
```

WordPress' built-in cron task schedule `wp-cron` is disabled since it is not effective for low traffic websites.

To add custom cron events, use Cloudron's built-in cron or use a plugin like WP Crontrol.

7.195.7 Plugins

Unlike the Managed WordPress app, you can install plugins that modify the code.

Disabling plugin

If a plugin is preventing WordPress from starting, open the File manager. Navigate to `/app/data/public/wp-content/plugins` and rename the offending plugin directory from `plugin-name` to say `plugin-name-broken`.

Note that to enable it back, you have to not only rename the folder but also activate the plugin in the WordPress dashboard.

Disabling all plugins

To disable all plugins, rename `/app/data/public/wp-content/plugins` to `/app/data/public/wp-content/plugins-broken` using the File manager.

Note that to enable back all the plugins, you have to not only rename the folder but also activate the plugins in the WordPress dashboard.

7.195.8 Performance

GTmetrix is a great site for getting performance metrics on the WordPress installation.

- To set the expires headers for all pages, the WP Fastest Cache plugin can be installed.
- For CDN caching, we recommend WP Fastest Cache or W3 Total Cache for CDN based cache. Ryan Kite has a good tutorial on how to setup AWS Cloudfront with WP Fastest Cache.

7.195.9 Database access

Cloudron does not support PHPMyAdmin. It is, however, possible to access the database using other methods:

- Open a Web terminal and press the 'MySQL' button to get console access. You can execute SQL commands directly.
- Use a plugin like WP phpMyAdmin

7.195.10 WP CLI

WP CLI is a command line interface to WordPress. To run commands using the CLI tool, open a Web terminal and execute commands WP CLI using simply `wp`. It is pre-setup to run as the correct user already. For example:

```
wp user list
```

If one or more plugins/themes are erroring, you can make WP CLI skip loading plugins/themes as follows:

```
wp --skip-plugins --skip-themes
```

Additional php settings can be configured, when running the cli manually with `php -d key=value`:

```
sudo -E -u www-data php -d max_execution_time=100 /app/pkg/wp --path=/app/data/public/
```

In this case setting the maximum execution timeout to 100 seconds.

7.195.11 PHP settings

You can add custom PHP settings in `/app/data/php.ini`

File upload size

Change the following values in `/app/data/php.ini`:

```
post_max_size = 256M
upload_max_filesize = 256M
memory_limit = 256M
```

7.195.12 Migrating existing site

See our blog on how to migrate an existing WordPress site to Cloudron.

7.195.13 File editing

WordPress' built-in file editing functionality is enabled by default. For security reasons, we recommend that you turn this off by editing `/app/data/wp-config.php` and setting `DISALLOW_FILE_EDIT` to true.

```
define('DISALLOW_FILE_EDIT', true);
```

7.195.14 Email

By default, the app is configured to use smtp-mailer plugin.

A custom mailer plugin can be used as follows:

- Disable Email Configuration in App -> Email -> `Do not configure app's mail delivery settings`. When disabled, Cloudron will not try to configure `smtp-mailer` on every restart.
- Install your preferred mailer plugin in WordPress.
- The email credentials depend on your setup. With an external mail relay like Mailgun/SES/Postmark, you can use those credentials directly in WordPress Alternately, you can create relay or mailbox credentials in your Email provider.
- When using Cloudron as your email server, simply create a mailbox and use an email app password. Use the server's SMTP configuration as the sending server. For further security, you can consider a separate Cloudron user that is the owner of the created mailbox (this way a bad plugin cannot access your personal mailboxes). Note the SMTP username is same as the mailbox address (and not the Cloudron username).

Fluent SMTP plugin configuration:

7.195.15 Unfiltered HTML

Non-admins are allowed to post unfiltered HTML content. You can disable this by editing `/app/data/wp-config.php` and setting `DISALLOW_UNFILTERED_HTML` to true.

```
define('DISALLOW_UNFILTERED_HTML', true);
```

7.195.16 Multisite

To multisite or not to multisite

WordPress multisite is a complex system with many compatibility gotchas. Unless you have a strong reason, we recommend installing a separate WordPress app for each site.

To enable WordPress multisite, start with a fresh installation and use the WordPress Network Setup Tool.

- Enable Multisite in `/app/data/public/wp-config.php` by adding the following line using the File manager. Add this line above the line that says "That’s all, stop editing! Happy blogging.":

```
/* Multisite */
define( 'WP_ALLOW_MULTISITE', true );
```

- Go to `Tools` -> `Network Setup` in the WordPress dashboard. As instructed in that page, deactivate all the plugins before proceeding. Cloudron supports both sub-domain and sub-directory installation.
- Once you click install, you will see a message `Warning! Wildcard DNS may not be configured correctly!`. To fix this, go to the `Location` view of the Cloudron dashboard and configure a Wildcard alias. Once the alias has been added, the warning will disappear (you have to refresh the WordPress dashboard).
- To complete the network installation, add the following to `/app/data/public/wp-config.php` as instructed.

```
define('MULTISITE', true);
define('SUBDOMAIN_INSTALL', true);
define('DOMAIN_CURRENT_SITE', 'msite.cloudron.club');
define('PATH_CURRENT_SITE', '/');
define('SITE_ID_CURRENT_SITE', 1);
define('BLOG_ID_CURRENT_SITE', 1);
```

Also, completely replace the contents of `/app/data/public/.htaccess` as instructed. Note that the Rewrite rules are slightly different for sub-domain and sub-directory setups. The config below is for sub-domain setup:

```
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]

# add a trailing slash to /wp-admin
RewriteRule ^wp-admin$ wp-admin/ [R=301,L]

RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^(wp-(content|admin|includes).*) $1 [L]
RewriteRule ^(.*\.php)$ $1 [L]
RewriteRule . index.php [L]
```

- You can add new sites from the `Network Admin` menu. You can add a site as a subdomain or subdirectory. The Site Address can be changed after addition by editing the site.
- If you set the Site Address to a different domain, you simply have to add it to the domain aliases in the `Location` section in Cloudron Dashboard.

Email Setup (Multisite)

In multisite mode, the SMTP mailer plugin can be configured per-site. After adding a site, restart the app for the package code to automatically configure the plugin.

7.196 WordPress (Managed) App

7.196.1 About

This app is targeted at users who want a managed WordPress installation. The Cloudron team tracks upstream WordPress releases and publishes updates. The WordPress code is read-only and you have to extend WordPress using plugins. If you require full control over the installation including editing the WordPress code, use the WordPress (unmanaged) app instead.

- Questions? Ask in the Cloudron Forum - WordPress

7.196.2 Admin page

The WordPress admin page is located `https://<my.example.com>/wp-login.php`.

7.196.3 Admin user

When using WordPress with Cloudron user management, an admin user is created by default with a random password and the email as `admin@cloudron.local`. You can delete the admin user post-installation by logging in as a Cloudron admin. It is not deleted automatically because the default posts auto-generated by WordPress installer are assigned to admin.

7.196.4 Administration Email Address

WordPress uses the Administration Email Address to send out important administrative emails. To be able to receive those, make sure to change that address in the `Settings` section.

7.196.5 Using SFTP

The Managed WordPress app does not support editing files via SFTP. If you require SFTP access to edit WordPress files, use the WordPress (Developer) app instead.

7.196.6 Memory limits

To adjust memory allocated for WordPress, edit `/app/data/wp-config.php` using the File manager and add the following line at the end of the file:

```
define('WP_MEMORY_LIMIT', '128M');
define('WP_MAX_MEMORY_LIMIT', '256M');
```

Note that the app also has a separate memory limit controlled by the app's memory limit. If you increase `WP_MEMORY_LIMIT`, be sure to increase the app's memory limit. A good formula is to provide the app 6 times the `WP_MEMORY_LIMIT` value at the bare minimum.

`WP_MAX_MEMORY_LIMIT` is the limit for administration tasks, which often require more.

A detailed explanation can be found in the WordPress docs.

7.196.7 Cron tasks

The app is configured to run WordPress cron tasks every 5 minutes.

To run the cron tasks manually run the following command using the Web terminal:

```
wp cron event run --due-now
```

WordPress' built-in cron task schedule `wp-cron` is disabled since it is not effective for low traffic websites.

To add custom cron events, use Cloudron's built-in cron or use a plugin like WP Crontrol.

7.196.8 Plugins

Cloudron does not support plugins that modify the code. Code is read-only and immutable. This property is essential for Cloudron to update apps correctly.

Use the WordPress (unmanaged) app as an alternative to install plugins that modify the code.

Disabling plugin

If a plugin is preventing WordPress from starting, open the File manager. Navigate to `/app/data/wp-content/plugins` and rename the offending plugin directory from `plugin-name` to say `plugin-name-broken`.

Note that to enable it back, you have to not only rename the folder but also activate the plugin in the WordPress dashboard.

Disabling all plugins

To disable all plugins, rename `/app/data/wp-content/plugins` to `/app/data/wp-content/plugins-broken` using the File manager.

Note that to enable back all the plugins, you have to not only rename the folder but also activate the plugins in the WordPress dashboard.

7.196.9 Performance

GTmetrix is a great site for getting performance metrics on the WordPress installation.

- To set the expires headers for all pages, the WP Fastest Cache plugin can be installed.
- For CDN caching, we recommend WP Fastest Cache or W3 Total Cache for CDN based cache. Ryan Kite has a good tutorial on how to setup AWS Cloudfront with WP Fastest Cache.

7.196.10 Database access

Cloudron does not support PHPMyAdmin. It is, however, possible to access the database using other methods:

- Open a Web terminal and press the 'MySQL' button to get console access. You can execute SQL commands directly.
- Use a plugin like WP phpMyAdmin

7.196.11 WP CLI

WP CLI is a command line interface to WordPress. To run commands using the CLI tool, open a Web terminal and execute commands WP CLI. For example,

```
wp user list
```

If one or more plugins/themes are erroring, you can make WP CLI skip loading plugins/themes as follows:

```
wp --skip-plugins --skip-themes
```

Additional php settings can be configured, when running the cli with `php -d key=value`:

```
sudo -E -u www-data php -d max_execution_time=100 /app/pkg/wp
```

In this case setting the maximum execution timeout to 100 seconds.

Disable themes

To list the themes:

```
sudo -E -u www-data /app/pkg/wp --skip-plugins --skip-themes theme list
```

To disable a theme, just activate another theme:

```
sudo -E -u www-data /app/pkg/wp --skip-plugins --skip-themes theme activate sometheme
```

7.196.12 PHP settings

You can add custom PHP settings in `/app/data/htaccess` using the File manager. Note that settings with a mode of `PHP_INI_SYSTEM` cannot be set in htaccess files.

For example:

```
#example
php_value post_max_size 600M
php_value upload_max_filesize 600M
php_value memory_limit 128M
php_value max_execution_time 300
php_value max_input_time 300
php_value session.gc_maxlifetime 1200
```

7.196.13 Migrating existing site

See our blog on how to migrate an existing WordPress site to Cloudron.

7.196.14 File editing

For security reasons, WordPress' built-in file editing functionality is disabled by default. You can enable this back by editing `/app/data/wp-config.php` and setting `DISALLOW_FILE_EDIT` to false.

```
define('DISALLOW_FILE_EDIT', false);
```

7.196.15 Unfiltered HTML

For security reasons, non-admins are not allowed to post unfiltered HTML content. You can enable this back by editing `/app/data/wp-config.php` and setting `DISALLOW_UNFILTERED_HTML` to false.

```
define('DISALLOW_UNFILTERED_HTML', false);
```

7.197 XBackBone App

7.197.1 About

XBackBone is a simple and lightweight PHP file manager that support the instant sharing tool ShareX and UNIX systems.

- Questions? Ask in the Cloudron Forum - XBackBone
- XBackBone Website
- XBackBone forum
- XBackBone issue tracker

7.197.2 Registration

Registration is disabled by default. This can be enabled in `System` -> `System Settings`.

7.197.3 ShareX setup

ShareX can be set up to upload images to a XBackBone instance as follows:

- Download Client configuration from `Profile` -> `Client Configuration` -> `ShareX`.
- Upload the client configuration in ShareX. `Destinations` -> `Custom uploader` . Then, `Import` -> `From file`
- Change the default Image uploader location to your custom uploader.

7.197.4 File upload size

Change the following values in `/app/data/php.ini` using the File manager:

```
post_max_size = 256M
upload_max_filesize = 256M
memory_limit = 256M
```

Be sure to restart the app after making any changes.

7.198 YOURLS App

7.198.1 About

YOURLS is a URL Shortener.

- Questions? Ask in the Cloudron Forum - Yourls
- Yourls Website
- Yourls issue tracker

7.198.2 Admin password

If the app is installed without Cloudron user directory integration, the admin password can be changed using the File Manager and editing the file `/app/data/user/config.php` and set a new plain text password.

YOURLS will automatically encrypt this field if you refresh your browser as explained here.

```
$yourls_user_passwords = array(
    'admin' => 'supersecret123' /* Password encrypted by YOURLS */
);
```

7.198.3 Custom index page

You can edit `/app/data/index.html` using the File Manager to your liking to customize the home page. You can also add a `/app/data/index.php` to place a PHP script.

7.198.4 Public Shortner

You can make the URL shortner available to all users. Be aware that a public interface will attract spammers. YOURLS project comes with a public page that be used for this purpose.

1. Use the Web terminal and set the sample public page as the default page:

```
    cp /app/code/sample-public-front-page.txt /app/data/index.php
```

1. fix the require_once path in `/app/data/index.php` from `dirname(__FILE__).'/includes/load-yourls.php'` to `/app/code/includes/load-yourls.php`. Like so:

```
    require_once( '/app/code/includes/load-yourls.php' );
```

7.198.5 Plugins

YOURLS supports a wide variety of plugins. To install a plugin, use the Web terminal to unpack a plugin into `/app/data/user/plugins/`. For example,

```
# cd /app/data/user/plugins
# git clone https://github.com/apelly/YourlsBlacklistDomains.git
# chown -R www-data:www-data YourlsBlacklistDomains
```

Then, activate the plugin using the `Manage Plugins` link in the admin dashboard.

8. Packaging Apps

8.1 Cloudron CLI

8.1.1 Overview

Cloudron CLI is a command line tool used for building and installing custom apps for Cloudron.

All CLI commands operate on 'apps' and not on the server. For example, `cloudron restart`, `cloudron uninstall` etc are operating on an app and not the server.

8.1.2 Installing

Cloudron CLI is distributed via `npm`. The Cloudron CLI can be installed on Linux/Mac using the following command:

```
sudo npm install -g cloudron
```

The Cloudron CLI is not actively tested on Windows but is known to work with varying success. If you use Windows, we recommend using a Linux VM instead.

> ⚠ **Do not install on Cloudron**
>
> The Cloudron CLI is intended to be installed on your PC/Mac and should **NOT** be installed on the Cloudron.

8.1.3 Updating

Cloudron CLI can be updated using the following command:

```
npm install -g cloudron@<version>
```

8.1.4 Login

Use the `login` command to authenticate with your Cloudron:

```
cloudron login my.example.com
```

A successful login stores the authentication token in `~/.cloudron.json`.

> ✎ **Self-signed certificates**
>
> When using Cloudron with self-signed certificates, use the `--allow-selfsigned` option.

8.1.5 Listing apps

Use the `list` command to display the installed apps:

```
cloudron list
```

The `Id` is the unique application instance id. `Location` is the domain in which the app is installed. You can use either of these fields as the argument to `--app`.

8.1.6 Viewing logs

To view the logs of an app, use the `logs` command:

```
cloudron logs --app blog.example.com
cloudron logs --app 52aae895-5b7d-4625-8d4c-52980248ac21
```

Pass the `-f` to follow the logs. Note that not all apps log to stdout/stderr. For this reason, you may need to look further in the file system for logs:

```
cloudron exec --app blog.example.com       # shell into the app's file system
# tail -f /run/wordpress/wp-debug.log      # note that log file path and name is specific to the app
```

8.1.7 Pushing a file

To push a local file (i.e on the PC/Mac) to the app's file system, use the `push` command:

```
cloudron push --app blog.example.com dump.sql /tmp/dump.sql
cloudron push --app blog.example.com dump.sql /tmp/               # same as above. trailing slash is required
```

To push a directory recursively to the app's file system, use the following command:

```
cloudron push --app blog.example.com files /tmp
```

8.1.8 Pulling a file

To pull a file from apps's file system to the PC/Mac, use the `pull` command:

```
cloudron pull --app blog.example.com /app/code/wp-includes/load.php .  # pulls file to current dir
```

To pull a directory from the app's file system, use the following command:

```
cloudron pull --app blog.example.com /app/code/ .            # pulls content of code to current dir
cloudron pull --app blog.example.com /app/code/ code_backup  # pulls content of code to ./code_backup
```

8.1.9 Environment variables

To set an environment variable(s):

```
cloudron env set --app blog.example.com RETRY_INTERVAL=4000 RETRY_TIMEOUT=12min
```

To unset an environment variable:

```
cloudron env unset --app blog.example.com RETRY_INTERVAL
```

To list environment variables:

```
cloudron env list --app blog.example.com
```

To list a single environment variable:

```
cloudron env get --app blog.example.com RETRY_INTERVAL
```

8.1.10 Application Shell

On the Cloudron, apps are containerized and run with a virtual file system. To navigate the file system, use the `exec` command:

```
cloudron exec --app blog.example.com
```

Apart from 3 special directories - `/app/data`, `/run` and `/tmp`, the file system of an app is read-only. Changes made to `/run` and `/tmp` will be lost across restarts (they are also cleaned up periodically).

8.1.11 Execute a command

The Cloudron CLI tool can be used to execute arbitrary commands in the context of app.

```
cloudron exec --app blog.example.com
# ls                             # list files in the app's current dir
# mysql --user=${MYSQL_USERNAME} --password=${MYSQL_PASSWORD} --host=${MYSQL_HOST} ${MYSQL_DATABASE} # connect to app's mysql
```

It's possible to pass a command with options by using the `--` to indicate end of arguments list:

```
cloudron exec --app blog.example.com -- ls -l
```

If the command has environment variables, then execute it using a shell:

```
cloudron exec --app blog.example.com -- bash -c 'mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST}
${CLOUDRON_MYSQL_DATABASE} -e "SHOW TABLES"';
```

8.1.12 CI/CD

To integrate the CLI tool as part of a CI/CD pipeline, you can use `--server` and `--token` arguments. You can get tokens by navigating to `https://my.example.com/#/profile`.

```
cloudron update --server my.example.com --token 001e7174c4cbad2272 --app blog.example.com --image username/image:tag
```

8.2 Packaging Tutorial

8.2.1 Overview

This tutorial outlines how to package a web application for the Cloudron.

Creating an application for Cloudron can be summarized as follows:

- Create a Dockerfile for your application.
- Create a CloudronManifest.json. This file specifies the addons (like database) required to run your app. When the app runs on the Cloudron, it will have environment variables set for connecting to the addon.
- Build the app using `docker build` and push the image to any public or private docker registry using `docker push`. To help out with the build & push workflow, you can use `cloudron build`.
- Install the app on the cloudron using `cloudron install --image <image>`.
- Update the app on the cloudron using `cloudron update --image <newimage>`.

8.2.2 Prerequisites

Cloudron CLI

Cloudron CLI is a command line tool used for building and installing custom apps for Cloudron. You can install the CLI tool **on your PC/Mac** as follows:

```
$ sudo npm install -g cloudron
```

You can login to your Cloudron now:

```
$ cloudron login my.example.com
Enter credentials for my.example.com:
Username: girish
Password:
Login successful.
```

`cloudron --help` provides a list of all the available commands. See CLI docs for a quick overview.

Docker

Docker is used for building application images. You can install it from here.

8.2.3 Sample app

We will package a simple app to understand how the packaging flow works. You can clone any of the following repositories to get started (you can also use `cloudron init` to create a bare bone app):

- Nodejs App

  ```
  $ git clone https://git.cloudron.io/docs/tutorial-nodejs-app
  ```

- Typescript App

  ```
  $ git clone https://git.cloudron.io/docs/tutorial-typescript-app
  ```

- PHP App

  ```
  $ git clone https://git.cloudron.io/docs/tutorial-php-app
  ```

- Multi-process App

  ```
  $ git clone https://git.cloudron.io/docs/tutorial-supervisor-app
  ```

All our published apps are Open Source and available in our git. You can use any of those as a starting point.

8.2.4 Build

The next step is to build the docker image and push the image to a repository.

```
# enter app directory
$ cd nodejs-app

# build the app
$ docker build -t username/nodejs-app:1.0.0 .

# push the image. if the push fails, you have to 'docker login' with your username
$ docker push username/nodejs-app:1.0.0
```

8.2.5 Install

If you use the public docker registry, Cloudron can pull the app image that you built with no authentication. If you use a private registry, Cloudron has to be configured with the private registry credentials. You can do this in the `Settings` view of Cloudron.

We are now ready to install the app on Cloudron.

```
# be sure to be in the app directory
$ cd tutorial-nodejs-app

$ cloudron install --image username/nodejs-app:1.0.0
Location: app.example.com
App is being installed.

 => Starting ...
 => Registering subdomains
 => Downloading image ....
 => Setting up collectd profile

App is installed.
```

Private registry

If you are using a private registry for your image, first configure Cloudron with the private registry credentials. Then, prefix the registry to `--image`. E.g `cloudron install --image docker.io/username/nodejs-app:1.0.0`.

Open the app in your default browser:

```
$ cloudron open
```

You should see `Hello World` on your browser.

8.2.6 Logs

You can view the logs using `cloudron logs`. When the app is running you can follow the logs using `cloudron logs -f`.

For example, you can see the console.log output in our server.js with the command below:

```
$ cloudron logs
Using cloudron craft.selfhost.io
16:44:11 [main] Server running at port 8000
```

8.2.7 Update

To update the application, simply build a new docker image and apply the update:

```
$ docker build -t username/nodejs-app:2.0.0 .
$ docker push username/nodejs-app:2.0.0
$ cloudron update --image username/nodejs-app:2.0.0
```

Note that you must provide a tag different from the existing installation for the docker image when calling `cloudron update`. This is because, if the tag stays the same, the Docker client does not check the registry to see if the local image and remote image differ.

To workaround this, we recommend that you tag docker images using a timestamp:

```
$ NOW=$(date +%s)
$ docker build -t username/nodejs-app:$NOW
$ docker push username/nodejs-app:$NOW
$ cloudron install --image username/nodejs-app:$NOW
```

Alternately, the `cloudron build` command automates the above workflow. The `build` command remembers the registry and repository name as well (in `~/.cloudron.json`).

You can do this instead:

```
# this command will ask the repository name on first run
$ cloudron build
Enter repository (e.g registry/username/com.example.cloudronapp): girish/nodejs-app

Building com.example.cloudronapp locally as girish/nodejs-app:20191113-014051-30452a2c5
...
Pushing girish/nodejs-app:20191113-014051-30452a2c5
...

# the tool remembers the last docker image built and installs that
$ cloudron update
Location: app.cloudron.ml
App is being installed.

 => Starting ...
 => Registering subdomains
 => Creating container

App is updated.
```

This way you can just use `cloudron build` and `cloudron update` repeatedly for app development.

Build service

Building docker images locally might require many CPU resources depending on your app. Pushing docker images can also be network intensive. If you hit these constraints, we recommend using the Docker Builder App. The builder app is installed on a separate Cloudron (not production Cloudron) and acts as a proxy for building docker images and also pushes them to your registry.

8.2.8 Next steps

This concludes our simple tutorial on building a custom app for Cloudron.

There are various Cloudron specific considerations when writing the Dockerfile. You can read about them in the development guide.

8.3 Cheat Sheet

This cheat sheet covers various Cloudron specific considerations, caveats and best practices when deploying apps on Cloudron.

8.3.1 Dockerfile.cloudron

If you already have an existing Dockerfile in your project, you can name the Cloudron specific Dockerfile as `Dockerfile.cloudron` or `cloudron/Dockerfile`.

8.3.2 Examples

We have tagged many of our existing app packages by framework/language. You can also ask for help in our forum.

- https://git.cloudron.io/explore/projects?tag=php
- https://git.cloudron.io/explore/projects?tag=java
- https://git.cloudron.io/explore/projects?tag=rails
- https://git.cloudron.io/explore/projects?tag=ruby
- https://git.cloudron.io/explore/projects?tag=node
- https://git.cloudron.io/explore/projects?tag=meteor
- https://git.cloudron.io/explore/projects?tag=python
- https://git.cloudron.io/explore/projects?tag=rust
- https://git.cloudron.io/explore/projects?tag=nginx
- https://git.cloudron.io/explore/projects?tag=go

8.3.3 Filesystem

Read-only

The application container created on the Cloudron has a `readonly` file system. Writing to any location **at runtime** other than the below will result in an error:

Dir	Description
`/tmp`	Use this location for temporary files. The Cloudron will cleanup any files in this directory periodically.
`/run`	Use this location for runtime configuration and dynamic data. These files should not be expected to persist across application restarts (for example, after an update or a crash).
`/app/data`	Use this location to store application data that is to be backed up. To use this location, you must use the localstorage addon.
Other paths	Not writable

Suggestions for creating the Dockerfile:

- Install any required packages in the Dockerfile.
- Create static configuration files in the Dockerfile.
- Create symlinks to dynamic configuration files (for e.g a generated config.php) under `/run` in the Dockerfile.

One-time init

A common requirement is to perform some initialization the very first time the app is installed. You can either use the database or the filesystem to track the app's initialization state. For example, create a `/app/data/.initialized` file to track the status. We can save this file in `/app/data` because this is the only location that is persisted across restarts and updates.

```
if [[ ! -f /app/data/.initialized ]]; then
  echo "Fresh installation, setting up data directory..."
  # Setup commands here
  touch /app/data/.initialized
  echo "Done."
fi
```

File ownership

When storing files under `/app/data`, be sure to change the ownership of the files to match the app's user id before the app starts. This is required because ownership information can be "lost" across backup/update/restore. For example, if the app runs as the `cloudron` user, do this:

```
# change ownership of files
chown -R cloudron:cloudron /app/data

# start the app
gosu cloudron:cloudron npm start
```

For Apache+PHP apps you might need to change permissions to `www-data:www-data` instead.

8.3.4 Start script

Many apps do not launch the server directly. Instead, they execute a `start.sh` script (named so by convention, you can name it whatever you like) which is used as the app entry point.

At the end of the Dockerfile you should add your start script (`start.sh`) and set it as the default command. Ensure that the `start.sh` is executable in the app package repo. This can be done with `chmod +x start.sh`.

```
ADD start.sh /app/code/start.sh
CMD [ "/app/code/start.sh" ]
```

8.3.5 Non-root user

Cloudron runs the `start.sh` as root user. This is required for various commands like `chown` to work as expected. However, to keep the app and cloudron secure, always run the app with the least required permissions.

The `gosu` tool lets you run a binary with a specific user/group as follows:

```
/usr/local/bin/gosu cloudron:cloudron node /app/code/.build/bundle/main.js
```

8.3.6 Environment variables

The following environment variables are set as part of the application runtime.

Name	Description
`CLOUDRON`	Set to '1'. This is useful for writing Cloudron specific code
`CLOUDRON_ALIAS_DOMAINS`	Set to the domain aliases. Only set when `multiDomain` flag is enabled
`CLOUDRON_API_ORIGIN`	Set to the HTTP(S) origin of this Cloudron's API. For example, `https://my.example.com`
`CLOUDRON_APP_DOMAIN`	The domain name of the application. For example, `app.example.com`
`CLOUDRON_APP_ORIGIN`	The HTTP(s) origin of the application. For example, `https://app.example.com`
`CLOUDRON_PROXY_IP`	The IP address of the Cloudron reverse proxy. Apps can trust the HTTP headers (like `X-Forwarded-For`) for requests originating from this IP address.
`CLOUDRON_WEBADMIN_ORIGIN`	The HTTP(S) origin of the Cloudron's dashboard. For example, `https://my.example.com`

You can set custom environment variables using `cloudron env`.

8.3.7 Logging

Cloudron applications stream their logs to stdout and stderr. Logging to stdout has many advantages:

- App does not need to rotate logs and the Cloudron takes care of managing logs.
- App does not need special mechanism to release log file handles (on a log rotate).
- Integrates better with tooling like cloudron cli.

In practice, this ideal is hard to achieve. Some programs like apache simply don't log to stdout. In such cases, simply log to a subdirectory in `/run` (two levels deep) into files with `.log` extension and Cloudron will autorotate the logs.

8.3.8 Multiple processes

Docker supports restarting processes natively. Should your application crash, it will be restarted automatically. If your application is a single process, you do not require any process manager.

Use supervisor, pm2 or any of the other process managers if your application has more then one component. This **excludes** web servers like apache, nginx which can already manage their children by themselves. Be sure to pick a process manager that forwards signals to child processes.

supervisor

Supervisor can be configured to send the app's output to stdout as follows:

```
[program:app]
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
stderr_logfile=/dev/stderr
stderr_logfile_maxbytes=0
```

8.3.9 Memory Limit

By default, applications get 256MB RAM (including swap). This can be changed using the `memoryLimit` field in the manifest.

Design your application runtime for concurrent use by 100s of users. Cloudron is not designed for concurrent access by 1000s of users.

An app can determine it's memory limit by reading `/sys/fs/cgroup/memory/memory.limit_in_bytes` if the system uses groups v1 or `/sys/fs/cgroup/memory.max` for cgroups v2. For example, to spin one worker for every 150M RAM available to the app:

```
if [[ -f /sys/fs/cgroup/cgroup.controllers ]]; then # cgroup v2
    memory_limit=$(cat /sys/fs/cgroup/memory.max)
    [[ "${memory_limit}" == "max" ]] && memory_limit=$(( 2 * 1024 * 1024 * 1024 )) # "max" really means unlimited
else
    memory_limit=$(cat /sys/fs/cgroup/memory/memory.limit_in_bytes) # this is the RAM. we have equal amount of swap
fi
worker_count=$((memory_limit/1024/1024/150)) # 1 worker for 150M
worker_count=$((worker_count > 8 ? 8 : worker_count )) # max of 8
worker_count=$((worker_count < 1 ? 1 : worker_count )) # min of 1
```

8.3.10 SIGTERM handling

bash, by default, does not automatically forward signals to child processes. This would mean that a SIGTERM sent to the parent processes does not reach the children. For this reason, be sure to `exec` as the last line of the start.sh script. Programs like gosu, nginx, apache do proper SIGTERM handling.

For example, start apache using `exec` as below:

```
echo "Starting apache"
APACHE_CONFDIR="" source /etc/apache2/envvars
rm -f "${APACHE_PID_FILE}"
exec /usr/sbin/apache2 -DFOREGROUND
```

8.3.11 Debugging

To inspect the filesystem of a running app, use `cloudron exec`.

If an application keeps restarting (because of some bug), then `cloudron exec` will not work or will keep getting disconnected. In such situations, you can use `cloudron debug`. In debug mode, the container's file system is read-write. In addition, the app just pauses and does not run the `RUN` command specified in the Dockerfile.

You can turn off debugging mode using `cloudron debug --disable`.

8.3.12 Popular stacks

Apache

Apache requires some configuration changes to work properly with Cloudron. The following commands configure Apache in the following way:

- Disable all default sites
- Print errors into the app's log and disable other logs
- Limit server processes to `5` (good default value)
- Change the port number to Cloudron's default `8000`

```
RUN rm /etc/apache2/sites-enabled/* \
    && sed -e 's,^ErrorLog.*,ErrorLog "/dev/stderr",' -i /etc/apache2/apache2.conf \
    && sed -e "s,MaxSpareServers[^:].*,MaxSpareServers 5," -i /etc/apache2/mods-available/mpm_prefork.conf \
    && a2disconf other-vhosts-access-log \
    && echo "Listen 8000" > /etc/apache2/ports.conf
```

Afterwards, add your site config to Apache:

```
ADD apache2.conf /etc/apache2/sites-available/app.conf
RUN a2ensite app
```

In `start.sh` Apache can be started using these commands:

```
echo "Starting apache..."
APACHE_CONFDIR="" source /etc/apache2/envvars
rm -f "${APACHE_PID_FILE}"
exec /usr/sbin/apache2 -DFOREGROUND
```

Nginx

`nginx` is often used as a reverse proxy in front of the application, to dispatch to different backend programs based on the request route or other characteristics. In such a case it is recommended to run nginx and the application through a process manager like `supervisor`.

Example nginx supervisor configuration file:

```
[program:nginx]
directory=/tmp
command=/usr/sbin/nginx -g "daemon off;"
user=root
autostart=true
autorestart=true
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
stderr_logfile=/dev/stderr
stderr_logfile_maxbytes=0
```

The nginx configuration, provided with the base image, can be used by adding an application specific config file under `/etc/nginx/sites-enabled/` when building the docker image.

```
ADD <app config file> /etc/nginx/sites-enabled/<app config file>
```

Since the base image nginx configuration is unpatched from the ubuntu package, the application configuration has to ensure nginx is using `/run/` instead of `/var/lib/nginx/` to support the read-only filesystem nature of a Cloudron application.

Example nginx app config file:

```
client_body_temp_path /run/client_body;
proxy_temp_path /run/proxy_temp;
fastcgi_temp_path /run/fastcgi_temp;
scgi_temp_path /run/scgi_temp;
uwsgi_temp_path /run/uwsgi_temp;

server {
  listen 8000;

  root /app/code/dist;

  location /api/v1/ {
    proxy_pass http://127.0.0.1:8001;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 86400;
  }
}
```

PHP

PHP wants to store session data at `/var/lib/php/sessions` which is read-only in Cloudron. To fix this problem you can move this data to `/run/php/sessions` with these commands:

```
RUN rm -rf /var/lib/php/sessions && ln -s /run/php/sessions /var/lib/php/sessions
```

Don't forget to create this directory and it's ownership in the `start.sh`:

```
mkdir -p /run/php/sessions
chown www-data:www-data /run/php/sessions
```

Java

Java scales its memory usage dynamically according to the available system memory. Due to how Docker works, Java sees the hosts total memory instead of the memory limit of the app. To restrict Java to the apps memory limit it is necessary to add a special parameter to Java calls.

```
if [[ -f /sys/fs/cgroup/cgroup.controllers ]]; then # cgroup v2
    ram=$(cat /sys/fs/cgroup/memory.max)
    [[ "${ram}" == "max" ]] && ram=$(( 2 * 1024 * 1024 * 1024 )) # "max" means unlimited
else
    ram=$(cat /sys/fs/cgroup/memory/memory.limit_in_bytes) # this is the RAM. we have equal amount of swap
fi
```

```
ram_mb=$(numfmt --to-unit=1048576 --format "%fm" $ram)
export JAVA_OPTS="-XX:MaxRAM=${ram_mb}M"
java ${JAVA_OPTS} -jar ...
```

8.4 Addons

8.4.1 Overview

Addons are services like database, authentication, email, caching that are part of the Cloudron runtime. Setup, provisioning, scaling and maintenance of addons is taken care of by the platform.

The fundamental idea behind addons is to allow sharing of services across applications. For example, a single MySQL server instance can be used across multiple apps. The Cloudron platform sets up addons in such a way that apps are isolated from each other.

8.4.2 Using Addons

Addons are opt-in and must be specified in the Cloudron Manifest. When the app runs, environment variables contain the necessary information to access the addon. For example, the mysql addon sets the `CLOUDRON_MYSQL_URL` environment variable which is the connection string that can be used to connect to the database.

When working with addons, developers need to remember the following:

- Environment variables are subject to change every time the app restarts. This can happen if the Cloudron is rebooted or restored or the app crashes or an addon is re-provisioned. For this reason, applications must not cache the value of environment variables across restarts. Instead, they must use the environments directly. For example, use `process.env.CLOUDRON_MYSQL_URL` (nodejs) or `getenv("CLOUDRON_MYSQL_URL")` (PHP).
- Addons must be setup or updated on each application start up. Most applications use DB migration frameworks for this purpose to setup and update the DB schema.
- Addons are configured in the addons section of the manifest as below:

```
    {
      ...
      "addons": {
        "ldap": { },
        "redis" : { }
      }
    }
```

8.4.3 Addons

docker

This addon allows an app to create containers on behalf of the user. Note that this addons does not provide full fledged access to docker for security purposes. Only a limited set of operations are permitted.

Exported environment variables:

```
CLOUDRON_DOCKER_HOST=        # tcp://<IP>:<port>
```

Some important restrictions:

- Only the app can access the docker API. Containers created by the app cannot use the docker API.
- Any created containers is automatically moved to the `cloudron` internal network
- Any bind mounts have to be under `/app/data`.
- Containers created by an application are tracked by Cloudron internally and will get removed when the app is uninstalled.
- Finally, only a Cloudron superadmin can install/update/exec apps with the docker addon for security reasons.

email

This addon allows an app to send and recieve emails on behalf of the user. The intended use case is webmail applications.

If an app wants to send mail (e.g notifications), it must use the sendmail addon. If the app wants to receive email (e.g user replying to notification), it must use the recvmail addon instead.

Apps using the IMAP and ManageSieve services below must be prepared to accept self-signed certificates (this is not a problem because these are addresses internal to the Cloudron).

Exported environment variables:

```
CLOUDRON_EMAIL_SMTP_SERVER=       # SMTP server IP or hostname. This is the internal name of the mail server.
CLOUDRON_EMAIL_SMTP_PORT=         # SMTP server port. STARTL TLS is disabled on this port.
CLOUDRON_EMAIL_SMTPS_PORT=        # SMTPS server port
CLOUDRON_EMAIL_STARTTLS_PORT=     # SMTP STARTTLS port
CLOUDRON_EMAIL_IMAP_SERVER=       # IMAP server IP or hostname.
CLOUDRON_EMAIL_IMAP_PORT=         # IMAP server port
CLOUDRON_EMAIL_IMAPS_PORT=        # IMAPS server port. TLS required.
CLOUDRON_EMAIL_SIEVE_SERVER=      # ManageSieve server IP or hostname.
CLOUDRON_EMAIL_SIEVE_PORT=        # ManageSieve server port. TLS required.
CLOUDRON_EMAIL_DOMAIN=            # Primary mail domain of the app
CLOUDRON_EMAIL_DOMAINS=           # Comma separate list of domains handled by the server
CLOUDRON_EMAIL_SERVER_HOST=       # The FQDN of the mail server. Only use this, if the app cannot connect using the internal name.
```

ldap

This addon provides LDAP based authentication via LDAP version 3.

Exported environment variables:

```
CLOUDRON_LDAP_SERVER=                                # ldap server IP
CLOUDRON_LDAP_HOST=                                  # ldap server IP (same as above)
CLOUDRON_LDAP_PORT=                                  # ldap server port
CLOUDRON_LDAP_URL=                                   # ldap url of the form ldap://ip:port
CLOUDRON_LDAP_USERS_BASE_DN=                         # ldap users base dn of the form ou=users,dc=cloudron
CLOUDRON_LDAP_GROUPS_BASE_DN=                        # ldap groups base dn of the form ou=groups,dc=cloudron
CLOUDRON_LDAP_BIND_DN=                               # DN to perform LDAP requests
CLOUDRON_LDAP_BIND_PASSWORD=                         # Password to perform LDAP requests
```

The suggested LDAP filter is `(&(objectclass=user)(|(username=%uid)(mail=%uid)))` . This allows the user to login via username or email.

For debugging, cloudron exec can be used to run the `ldapsearch` client within the context of the app:

```
cloudron exec

# list users
> ldapsearch -x -H "${CLOUDRON_LDAP_URL}" -D "${CLOUDRON_LDAP_BIND_DN}" -w "${CLOUDRON_LDAP_BIND_PASSWORD}" -b "${CLOUDRON_LDAP_USERS_BASE_DN}"

# list users with authentication (Substitute username and password below)
> ldapsearch -x -H "${CLOUDRON_LDAP_URL}" -D cn=<username>,${CLOUDRON_LDAP_USERS_BASE_DN} -w <password> -b  "${CLOUDRON_LDAP_USERS_BASE_DN}"

# list groups
> ldapsearch -x -H "${CLOUDRON_LDAP_URL}" -D "${CLOUDRON_LDAP_BIND_DN}" -w "${CLOUDRON_LDAP_BIND_PASSWORD}" -b "${CLOUDRON_LDAP_GROUPS_BASE_DN}"
```

The user listing has the following LDAP attributes:

- `objectclass` - array that contains `user`
- `objectcategory` - set to 'person',
- `uid` , `entryuuid` - Unique identifier
- `cn` - Unique identifier (same as `uid`)
- `mail` - User's primary email
- `displayName` - Full name of the user
- `mailAlternateAddress` - Alternate/Fallback email address of the user (for password reset)
- `givenName` - First name of the user
- `sn` - Last name of the user
- `username` - Username set during account creation
- `samaccountname` - Same as username
- `memberof` - List of Cloudron groups the user is a memer of

The groups listing has the following LDAP attributes:

- `objectclass` - array that contains `group`
- `cn` : name of the group
- `gidnumber` : Unique identifier
- `memberuid` : array of members. Each entry here maps to `uid` in the user listing.

Unlike other addons, the LDAP addon get some special treatment and cannot be enabled on already installed apps. This means that you cannot push an update that enables LDAP addon and expect already installed apps to gain LDAP functionality. The user has to install the app afresh for LDAP integration.

The reason for this is that Cloudron keeps track of whether an app was installed with or without Cloudron user management using a "sso" flag. This flag cannot be changed after installation for simplicity. If it were dynamically changeable, it is unclear what's supposed to happen if an app was installed with sso and then later the user removed ldap addon i.e what happens to existing users? In some apps, an admin user might need to be created explicitly because they don't support LDAP and local database authentication simultaneously.

localstorage

Since all Cloudron apps run within a read-only filesystem, this addon provides a writeable folder under `/app/data/`. All contents in that folder are included in the backup. On first run, this folder will be empty. File added in this path as part of the app's image (Dockerfile) won't be present. A common pattern is to create the directory structure required the app as part of the app's startup script.

The permissions and ownership of data within that directory are not guranteed to be preserved. For this reason, each app has to restore permissions as required by the app as part of the app's startup script.

If the app is running under the recommeneded `cloudron` user, this can be achieved with:

```
chown -R cloudron:cloudron /app/data
```

FTP

FTP access can be enabled using the `ftp` option. The uid and uname refer to the user under which the ftp files will be stored in the app's local storage. FTP access should be enabled wisely since many apps don't like data being changed behind their back.

```
    "localstorage": {
      "ftp": {
        "uid": 33,
        "uname": "www-data"
      }
    }
```

SQLITE

Sqlite database files can be specified using the `sqlite` option.

Sqlite files that are actively in use cannot be backed up using a simple `cp`. Cloudron will take a consistent portable backups of Sqlite files specified in this option.

```
    "localstorage": {
      "sqlite": {
        "paths": ["/app/data/db/users.db"]
      }
    }
```

Database files must exist. If they are missing, backup and restore operations will error.

mongodb

By default, this addon provide MongoDB 4.4.

Exported environment variables:

```
CLOUDRON_MONGODB_URL=          # mongodb url
CLOUDRON_MONGODB_USERNAME=     # username
CLOUDRON_MONGODB_PASSWORD=     # password
CLOUDRON_MONGODB_HOST=         # server IP/hostname
CLOUDRON_MONGODB_PORT=         # server port
CLOUDRON_MONGODB_DATABASE=     # database name
CLOUDRON_MONGODB_OPLOG_URL=    # oplog access URL (see below)
```

App can request oplog access by setting the `oplog` option to be true.

```
"mongodb": { "oplog": true }
```

For debugging, cloudron exec can be used to run the `mongo` shell within the context of the app:

```
cloudron exec

> mongo -u "${CLOUDRON_MONGODB_USERNAME}" -p "${CLOUDRON_MONGODB_PASSWORD}" ${CLOUDRON_MONGODB_HOST}:${CLOUDRON_MONGODB_PORT}/${CLOUDRON_MONGODB_DATABASE}
```

mysql

By default, this addon provides a single database on MySQL 8.0.31. The database is already created and the application only needs to create the tables.

Exported environment variables:

```
CLOUDRON_MYSQL_URL=            # the mysql url (only set when using a single database, see below)
CLOUDRON_MYSQL_USERNAME=       # username
CLOUDRON_MYSQL_PASSWORD=       # password
CLOUDRON_MYSQL_HOST=           # server IP/hostname
CLOUDRON_MYSQL_PORT=           # server port
CLOUDRON_MYSQL_DATABASE=       # database name (only set when using a single database, see below)
```

For debugging, cloudron exec can be used to run the `mysql` client within the context of the app:

```
cloudron exec

> mysql --user=${CLOUDRON_MYSQL_USERNAME} --password=${CLOUDRON_MYSQL_PASSWORD} --host=${CLOUDRON_MYSQL_HOST} ${CLOUDRON_MYSQL_DATABASE}
```

The `multipleDatabases` option can be set to `true` if the app requires more than one database. When enabled, the following environment variables are injected and the `MYSQL_DATABASE` is removed:

```
CLOUDRON_MYSQL_DATABASE_PREFIX=      # prefix to use to create databases
```

All the databases use `utf8mb4` encoding by default.

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_unicode_ci   |
+--------------------------+----------------------+
```

To see the charset of a table: `SHOW CREATE TABLE <tablename>`. Columns can have a collation order of their own which can seen using `SHOW TABLE STATUS LIKE <tablename>`.

oidc

This addon provides OpenID connect based authentication.

Options:

```
"oidc": {
    "loginRedirectUri": "/auth/openid/callback",
    "logoutRedirectUri": "/home",
    "tokenSignatureAlgorithm": "RS256"
}
```

- `loginRedirectUri` where the user should be redirected to after successful authorization (only URL path, will be prefixed with app domain). Multiple ones can be provided, separated with comma (eg. `"/auth/login, app.immich:/"`).
- `logoutRedirectUri` where the user should be redirected to after successful logout (only URL path, will be prefixed with app domain)
- `tokenSignatureAlgorithm` can be either "RS256" or "EdDSA"

Exported environment variables:

```
CLOUDRON_OIDC_PROVIDER_NAME=     # The name of the provider. To be used for "Login with {{providerName}}" button in the login screen.
CLOUDRON_OIDC_DISCOVERY_URL=     # .well-known URL for auto-provisioning
CLOUDRON_OIDC_ISSUER=            # main OpenID provider URI
CLOUDRON_OIDC_AUTH_ENDPOINT=     # auth endpoint - mostly optional
CLOUDRON_OIDC_TOKEN_ENDPOINT=    # token endpoint - mostly optional
CLOUDRON_OIDC_KEYS_ENDPOINT=     # keys endpoint - mostly optional
CLOUDRON_OIDC_PROFILE_ENDPOINT=  # profile endpoint - mostly referred to as /me or /profile
CLOUDRON_OIDC_CLIENT_ID=         # client id
CLOUDRON_OIDC_CLIENT_SECRET=     # client secret
```

postgresql

By default, this addon provides PostgreSQL 14.9

Exported environment variables:

```
CLOUDRON_POSTGRESQL_URL=       # the postgresql url
CLOUDRON_POSTGRESQL_USERNAME=  # username
CLOUDRON_POSTGRESQL_PASSWORD=  # password
CLOUDRON_POSTGRESQL_HOST=      # server name
CLOUDRON_POSTGRESQL_PORT=      # server port
CLOUDRON_POSTGRESQL_DATABASE=  # database name
```

The postgresql addon whitelists the following extensions:

- address_standardizer;
- address_standardizer_data_us
- btree_gist
- btree_gin
- citext
- cube
- earthdistance
- fuzzystrmatch
- hstore
- ogr_fdw
- pgcrypto
- pg_stat_statements
- pg_trgm
- pgrouting
- plpgsql
- postgis
- postgis_tiger_geocoder
- postgis_sfcgal
- postgis_topology
- postgres_fdw
- uuid-ossp
- unaccent
- vector
- vectors

For debugging, cloudron exec can be used to run the `psql` client within the context of the app:

```
cloudron exec

> PGPASSWORD=${CLOUDRON_POSTGRESQL_PASSWORD} psql -h ${CLOUDRON_POSTGRESQL_HOST} -p ${CLOUDRON_POSTGRESQL_PORT} -U ${CLOUDRON_POSTGRESQL_USERNAME} -d ${CLOUDRON_POSTGRESQL_DATABASE}
```

The `locale` option can be set to a valid PostgreSQL locale. When set, `LC_LOCALE` and `LC_CTYPE` of the database are set upon creation accordingly.

proxyAuth

The `proxyAuth` addon can be used to setup an authentication wall in front of the app.

With the authentication wall, users will be faced with a login screen when visiting the app and have to login before being able to use it. The login screen uses a session (cookie) based authentication. It is also possible to login using HTTP Basic auth using the `Authorization` header.

The `path` property can be set if you want to restrict the wall to a subset of pages. For example:

```
"proxyAuth": { "path": "/admin" }
```

The `path` can also start with '!' to restrict all paths except those starting with that. For example:

```
"proxyAuth": { "path": "!/webhooks" }
```

The `basicAuth` property can be set to enable HTTP basic authentication. Enabling this property allows a user to bypass 2FA. For this reason, it is disabled by default.

The `supportsBearerAuth` can be set to indicate that an app supports bearer token authentication using the `Authorization` header. When set, all requests with `Bearer` in the `Authorization` header are forwarded to the app.

This flag utilizes two special routes - `/login` and `/logout`. These routes are unavailable to the app itself.

⚠ **Cannot add to existing app**

Due to a limitation of the platform, authentication cannot be added dynamically to an existing app. The app must be reinstalled.

recvmail

The recvmail addon can be used to receive email for the application.

Exported environment variables:

```
CLOUDRON_MAIL_IMAP_SERVER=     # the IMAP server. this can be an IP or DNS name
CLOUDRON_MAIL_IMAP_PORT=       # the IMAP server port
CLOUDRON_MAIL_IMAPS_PORT=      # the IMAP TLS server port
CLOUDRON_MAIL_POP3_PORT=       # the POP3 server port
CLOUDRON_MAIL_POP3S_PORT=      # the POP3 TLS server port
CLOUDRON_MAIL_IMAP_USERNAME=   # the username to use for authentication
CLOUDRON_MAIL_IMAP_PASSWORD=   # the password to use for authentication
CLOUDRON_MAIL_TO=              # the "To" address to use
CLOUDRON_MAIL_TO_DOMAIN=       # the mail for which email will be received
```

recvmail addon can be disabled for the cases where Cloudron is not receiving email for the domain. For this reason, apps must be prepared for the environment variables above to be missing.

For debugging, cloudron exec can be used to run the `openssl` tool within the context of the app:

```
cloudron exec

> openssl s_client -connect "${CLOUDRON_MAIL_IMAP_SERVER}:${CLOUDRON_MAIL_IMAP_PORT}" -crlf
```

The IMAP command `? LOGIN username password` can then be used to test the authentication.

redis

By default, this addon provides redis 6.0. The redis is configured to be persistent and data is preserved across updates and restarts.

Exported environment variables:

```
CLOUDRON_REDIS_URL=            # the redis url
CLOUDRON_REDIS_HOST=           # server name
CLOUDRON_REDIS_PORT=           # server port
CLOUDRON_REDIS_PASSWORD=       # password
```

App can choose to not use a password access by setting the `noPassword` option to be true. Since redis is only available reachable in the server's internal docker network, this is not a security issue.

```
"redis": { "noPassword": true }
```

For debugging, cloudron exec can be used to run the `redis-cli` client within the context of the app:

```
cloudron exec

> redis-cli -h "${CLOUDRON_REDIS_HOST}" -p "${CLOUDRON_REDIS_PORT}" -a "${CLOUDRON_REDIS_PASSWORD}"
```

scheduler

The scheduler addon can be used to run tasks at periodic intervals (cron).

Scheduler can be configured as below:

```
    "scheduler": {
        "update_feeds": {
            "schedule": "*/5 * * * *",
            "command": "/app/code/update_feed.sh"
        }
    }
```

In the above example, `update_feeds` is the name of the task and is an arbitrary string.

`schedule` values must fall within the following ranges:

- Minutes: 0-59
- Hours: 0-23
- Day of Month: 1-31
- Months: 0-11
- Day of Week: 0-6

NOTE: scheduler does not support seconds

`schedule` supports ranges (like standard cron):

- Asterisk. E.g. *
- Ranges. E.g. 1-3,5
- Steps. E.g. */2

`command` is executed through a shell (sh -c). The command runs in the same launch environment as the application. Environment variables, volumes (`/tmp` and `/run`) are all shared with the main application.

Tasks are given a grace period of 30 minutes to complete. If a task is still running after 30 minutes and a new instance of the task is scheduled to be started, the previous task instance is killed.

sendmail

The sendmail addon can be used to send email from the application.

Exported environment variables:

```
CLOUDRON_MAIL_SMTP_SERVER=       # the mail server (relay) that apps can use. this can be an IP or DNS name
CLOUDRON_MAIL_SMTP_PORT=         # the mail server port. Currently, this port disables TLS and STARTTLS.
CLOUDRON_MAIL_SMTPS_PORT=        # SMTPS server port.
CLOUDRON_MAIL_SMTP_USERNAME=     # the username to use for authentication
CLOUDRON_MAIL_SMTP_PASSWORD=     # the password to use for authentication
CLOUDRON_MAIL_FROM=              # the "From" address to use (i.e username@domain)
CLOUDRON_MAIL_FROM_DISPLAY_NAME= # the email Display name to use for the "From" address
CLOUDRON_MAIL_DOMAIN=            # the domain name to use for email sending (i.e only the domain part of username@domain)
```

The SMTP server does not require STARTTLS. If STARTTLS is used, the app must be prepared to accept self-signed certs.

For debugging, cloudron exec can be used to run the `swaks` tool within the context of the app:

```
cloudron exec

> swaks --server "${CLOUDRON_MAIL_SMTP_SERVER}" -p "${CLOUDRON_MAIL_SMTP_PORT}" --from "${CLOUDRON_MAIL_FROM}" --body "Test mail from cloudron app at $(hostname -f)" --auth-user "${CLOUDRON_MAIL_SMTP_USERNAME}" --auth-password "${CLOUDRON_MAIL_SMTP_PASSWORD}"


> swaks --server "${CLOUDRON_MAIL_SMTP_SERVER}" -p "${CLOUDRON_MAIL_SMTPS_PORT}" --from "${CLOUDRON_MAIL_FROM}" --body "Test mail from cloudron app at $(hostname -f)" --auth-user "${CLOUDRON_MAIL_SMTP_USERNAME}" --auth-password "${CLOUDRON_MAIL_SMTP_PASSWORD}" -tlsc
```

The `optional` flag can be set to `true` for apps that allow the user to completely take over the email configuration. When set, all the above environment variables will be absent at runtime.

The `supportsDisplayName` flag can be set to `true` for apps that allow the user to set the mail from display name. When enabled, the `CLOUDRON_MAIL_FROM_DISPLAY_NAME` environment variable is set.

`requiresValidCertificate` can be set to `true` for apps that require a valid mail server certificate to send email. When set, Cloudron will set `CLOUDRON_MAIL_SMTP_SERVER` to the FQDN of the mail server. In addition, it will reconfigure the app automatically when the domain name of the mail server changes.

tls

The tls addon can be used to access the certs of the primary domain of an app.

App sometimes require access to certs when implementing protocols like IRC or DNS-Over-TLS. Such apps can request access to certs using the `tls` addon.

The cert and key are made available (as readonly) in `/etc/certs/tls_cert.pem` and `/etc/certs/tls_key.pem` respectively. The app will be automatically restarted when the cert is renewed.

turn

The turn addon can be access the STUN/TURN service.

Exported environment variables:

```
CLOUDRON_TURN_SERVER=           # turn server name
CLOUDRON_TURN_PORT=             # turn server port
CLOUDRON_TURN_TLS_PORT          # turn server TLS port
CLOUDRON_TURN_SECRET            # turn server secret
```

8.5 Manifest

8.5.1 Overview

Every Cloudron Application contains a `CloudronManifest.json` that contains two broad categories of information:

- Information for installing the app on the Cloudron. This includes fields like httpPort, tcpPorts, udpPorts.
- Information about displaying the app on the Cloudron App Store. For example, the title, author information, description etc. When developing a custom app (i.e not part of the App Store), these fields are not required.

Here is an example manifest:

```
{
  "id": "com.example.test",
  "title": "Example Application",
  "author": "Girish Ramakrishnan <girish@cloudron.io>",
  "description": "This is an example app",
  "tagline": "A great beginning",
  "version": "0.0.1",
  "healthCheckPath": "/",
  "httpPort": 8000,
  "addons": {
    "localstorage": {}
  },
  "manifestVersion": 2,
  "website": "https://www.example.com",
  "contactEmail": "support@cloudron.io",
  "icon": "file://icon.png",
  "tags": [ "test", "collaboration" ],
  "mediaLinks": [ "https://images.rapgenius.com/fd0175ef780e2feefb30055be9f2e022.520x343x1.jpg" ]
}
```

8.5.2 Fields

addons

Type: object

Required: no

Allowed keys

- email
- ldap
- localstorage
- mongodb
- mysql
- oidc
- postgresql
- proxyauth
- recvmail
- redis
- sendmail
- scheduler
- tls

The `addons` object lists all the addons and the addon configuration used by the application.

Example:

```
"addons": {
  "localstorage": {},
  "mongodb": {}
}
```

author

Type: string

Required: no

The `author` field contains the name and email of the app developer (or company).

Example:

```
"author": "Cloudron UG <girish@cloudron.io>"
```

capabilities

Type: array of strings

Required: no

The `capabilities` field can be used to request extra capabilities.

By default, Cloudron apps are unprivileged and cannot perform many operations including changing network configuration, launch docker containers etc.

Currently, the permitted capabilities are:

- `net_admin` - This capability can be used to perform various network related operations like:
  - Interface configuration
  - Administration of IP firewall, masquerading, and accounting
  - Modify routing tables
- `mlock` - This prevents memory from being swapped to disk (`CAP_IPC_LOCK`).
- `ping` - This provides `NET_RAW`
- `vaapi` - This provides the container access to the VAAPI devices under `/dev/dri` . This capability was added in Cloudron 5.6.

Example:

```
"capabilities": [
  "net_admin"
]
```

changelog

Type: markdown string

Required: no

The `changelog` field contains the changes in this version of the application. This string can be a markdown style bulleted list.

Example:

```
"changelog": "* Add support for IE8 \n* New logo"
```

checklist

Type: object

Required: no

Syntax: Each key is a checklist item that contains a `message`. An optional `sso` flag may be specified.

The `checklist` is a list of items to be completed post installation. The items can be individually tracked - completed or not, by whom and when.

To illustrate, the application lists the checklists below:

```
    "checklist": {
      "todo-for-admins": { "message": "Please do this and that after installation" },
      "first-user": { "sso": true, "message": "SSO Example: First user becomes admin" },
      "change-password": { "sso": false, "message": "NoSSO Example: Change admin password on first use" }
    },
```

In the above example:

- `message` is a markdown string explaining the todo item
- `sso` flag can be used to control when the checklist item is applicable depending on the authentication setup.
    - If `sso` is `true`, the checklist item is shown only when an app is installed with Cloudron authentication.
    - If `sso` is `false`, the checklist item is shown only when an app is installed without Cloudron authentication.
    - If `sso` is missing, the checklist item is shown regardless of Cloudron authentication.

`checklist` items can be added or removed over the lifetime of a package. The platform tracks the package version when a `checklist` item was added (based on the key).

configurePath

Type: url path

Required: no

If this field is present, admins will see an additional link for an app in the dashboard. This url path will be prefixed with the app's domain and thus allows to put a direct link to an admin or settings panel in the app. This is useful for apps like WordPress or Ghost, which depending on the theme might not have admin login links visible on the page.

Example:

```
  "configurePath": "/wp-admin/"
```

contactEmail

Type: email

Required: no

The `contactEmail` field contains the email address that Cloudron users can contact for any bug reports and suggestions.

Example:

```
  "contactEmail": "support@testapp.com"
```

description

Type: markdown string

Required: no

The `description` field contains a detailed description of the app. This information is shown to the user when they install the app from the Cloudron App Store.

Example:

```
"description": "This is a detailed description of this app."
```

A large `description` can be unweildy to manage and edit inside the CloudronManifest.json. For this reason, the `description` can also contain a file reference. The Cloudron CLI tool fills up the description from this file when publishing your application.

Example:

```
"description:": "file://DESCRIPTION.md"
```

documentationUrl

Type: url

Required: no

The `documentationUrl` field is a URL where the user can read docs about the application.

Example:

```
"documentationUrl": "https://example.com/myapp/docs"
```

forumUrl

Type: url

Required: no

The `forumUrl` field is a URL where the user can get forum support for the application.

Example:

```
"forumUrl": "https://example.com/myapp/forum"
```

healthCheckPath

Type: url path

Required: yes

The `healthCheckPath` field is used by the Cloudron Runtime to determine if your app is running and responsive. The app must return a 2xx HTTP status code as a response when this path is queried. In most cases, the default "/" will suffice but there might be cases where periodically querying "/" is an expensive operation. In addition, the app might want to use a specialized route should it want to perform some specialized internal checks.

Example:

```
"healthCheckPath": "/"
```

httpPort

Type: positive integer

Required: yes

The `httpPort` field contains the TCP port on which your app is listening for HTTP requests. This is the HTTP port the Cloudron will use to access your app internally.

While not required, it is good practice to mark this port as `EXPOSE` in the Dockerfile.

Cloudron Apps are containerized and thus two applications can listen on the same port. In reality, they are in different network namespaces and do not conflict with each other.

Note that this port has to be HTTP and not HTTPS or any other non-HTTP protocol. HTTPS proxying is handled by the Cloudron platform (since it owns the certificates).

Example:

```
"httpPort": 8080
```

httpPorts

Type: object

Required: no

Syntax: Each key is the environment variable. Each value is an object containing `title`, `description`, `containerPort` and `defaultValue`.

The `httpPorts` field provides information on extra HTTP services that your application provides. During installation, the user can provide location information for these services.

To illustrate, the application lists the ports as below:

```
"httpPorts": {
  "API_SERVER_DOMAIN": {
    "title": "API Server Domain",
    "description": "The domain name for MinIO (S3) API requests",
    "containerPort": 9000,
    "defaultValue": "minio-api"
  }
},
```

In the above example:

- `API_SERVER_DOMAIN` is an app specific environment variable. It is set to the domain chosen by the user.
- `title` is a short one line information about this service.
- `description` is a multi line description about this service.
- `defaultValue` is the recommended subdomain value to be shown in the app installation UI.
- `containerPort` is the HTTP port that the app is listening on for this service.

icon

Type: local image filename

Required: no

The `icon` field is used to display the application icon/logo in the Cloudron App Store. Icons are expected to be square of size 256x256.

```
"icon": "file://icon.png"
```

id

Type: reverse domain string

Required: no

The `id` is a unique human friendly Cloudron App Store id. This is similar to reverse domain string names used as java package names. The convention is to base the `id` based on a domain that you own.

The Cloudron tooling allows you to build applications with any `id`. However, you will be unable to publish the application if the id is already in use by another application.

```
"id": "io.cloudron.testapp"
```

logPaths

Type: array of paths

Required: no

The `logPaths` field contains an array of paths that contain the logs.

Whenever possible, apps must be configured to stream logs to stdout and stderr. Only use this field when the app or service is unable to do so.

```
"logPaths": [
  "/run/app/app.log",
  "/run/app/workhorse.log"
]
```

manifestVersion

Type: integer

Required: yes

`manifestVersion` specifies the version of the manifest and is always set to 2.

```
"manifestVersion": 2
```

mediaLinks

Type: array of urls

Required: no

The `mediaLinks` field contains an array of links that the Cloudron App Store uses to display a slide show of pictures of the application.

They have to be publicly reachable via `https` and should have an aspect ratio of 3 to 1. For example `600px by 200px` (with/height).

```
"mediaLinks": [
  "https://s3.amazonaws.com/cloudron-app-screenshots/org.owncloud.cloudronapp/556f6a1d82d5e27a7c4fca427ebe6386d373304f/2.jpg",
  "https://images.rapgenius.com/fd0175ef780e2feefb30055be9f2e022.520x343x1.jpg"
]
```

memoryLimit

Type: bytes (integer)

Required: no

The `memoryLimit` field is the maximum amount of memory (including swap) in bytes an app is allowed to consume before it gets killed and restarted.

By default, all apps have a memoryLimit of 256MB. For example, to have a limit of 500MB,

```
"memoryLimit": 524288000
```

maxBoxVersion

Type: semver string

Required: no

The `maxBoxVersion` field is the maximum box version that the app can possibly run on. Attempting to install the app on a box greater than `maxBoxVersion` will fail.

This is useful when a new box release introduces features which are incompatible with the app. This situation is quite unlikely and it is recommended to leave this unset.

Cloudron updates are blocked, if the Cloudron has an app with a `maxBoxVersion` less than the upcoming Cloudron version.

minBoxVersion

Type: semver string

Required: no

The `minBoxVersion` field is the minimum box version that the app can possibly run on. Attempting to install the app on a box lesser than `minBoxVersion` will fail.

This is useful when the app relies on features that are only available from a certain version of the box. If unset, the default value is `0.0.1`.

multiDomain

Type: boolean

Required: no

When set, this app can be assigned additional domains as aliases to the primary domain of the app.

postInstallMessage

Type: markdown string

Required: no

The `postInstallMessage` field is a message that is displayed to the user after an app is installed.

The intended use of this field is to display some post installation steps that the user has to carry out to complete the installation. For example, displaying the default admin credentials and informing the user to to change it.

The message can have the following special tags:

- `<sso> ... </sso>` - Content in `sso` blocks are shown if SSO enabled.
- `<nosso> ... </nosso>`- Content in `nosso` blocks are shows when SSO is disabled.

The following variables are dynamically replaced:

Variable	Meaning
$CLOUDRON-APP-LOCATION	App subdomain
$CLOUDRON-APP-DOMAIN	App domain
$CLOUDRON-APP-FQDN	App FQDN (subdomain and domain)
$CLOUDRON-APP-ORIGIN	App origin i.e `https://FQDN`
$CLOUDRON-API-DOMAIN	Cloudron Dashboard Domain
$CLOUDRON-API-ORIGIN	Cloudron Dashboard Origin ie. `https://my.domain.com`
$CLOUDRON-USERNAME	Username of the current logged in user
$CLOUDRON-APP-ID	Unique App ID. This can be used generate the deep links into the Cloudron dashboard

optionalSso

Type: boolean

Required: no

The `optionalSso` field can be set to true for apps that can be installed optionally without using the Cloudron user management.

This only applies if any Cloudron auth related addons are used. When set, the Cloudron will not inject the auth related addon environment variables. Any app startup scripts have to be able to deal with missing env variables in this case.

runtimeDirs

Type: array of paths

Required: no

The `runtimeDirs` field contains an array of paths that are writable at run time.

On startup, the contents of these directories in the docker image are carried over to the container. Please note that these paths are not backed up. Only subdirectories of `/app/code` are allowed to be specified. These directories are also not persisted across updates.

```
"runtimeDirs": [
  "/app/code/node_modules",
  "/app/code/public"
]
```

tagline

Type: one-line string

Required: no

The `tagline` is used by the Cloudron App Store to display a single line short description of the application.

```
"tagline": "The very best note keeper"
```

tags

Type: Array of strings

Required: no

The `tags` are used by the Cloudron App Store for filtering searches by keyword.

```
"tags": [ "git", "version control", "scm" ]
```

Available tags: * blog * chat * git * email * sync * gallery * notes * project * hosting * wiki

targetBoxVersion

Type: semver string

Required: no

The `targetBoxVersion` field is the box version that the app was tested on. By definition, this version has to be greater than the `minBoxVersion`.

The box uses this value to enable compatibility behavior of APIs. For example, an app sets the targetBoxVersion to 0.0.5 and is published on the store. Later, box version 0.0.10 introduces a new feature that conflicts with how apps used to run in 0.0.5 (say SELinux was enabled for apps). When the box runs such an app, it ensures compatible behavior and will disable the SELinux feature for the app.

If unspecified, this value defaults to `minBoxVersion`.

tcpPorts

Type: object

Required: no

Syntax: Each key is the environment variable. Each value is an object containing `title`, `description` and `defaultValue`. An optional `containerPort` may be specified.

The `tcpPorts` field provides information on the non-http TCP ports/services that your application is listening on. During installation, the user can decide how these ports are exposed from their Cloudron.

For example, if the application runs an SSH server at port 29418, this information is listed here. At installation time, the user can decide any of the following: * Expose the port with the suggested `defaultValue` to the outside world. This will only work if no other app is being exposed at same port. * Provide an alternate value on which the port is to be exposed to outside world. * Disable the port/service.

To illustrate, the application lists the ports as below:

```
"tcpPorts": {
  "SSH_PORT": {
    "title": "SSH Port",
    "description": "SSH Port over which repos can be pushed & pulled",
    "defaultValue": 29418,
    "containerPort": 22,
    "portCount": 1
  }
},
```

In the above example:

- `SSH_PORT` is an app specific environment variable. Only strings, numbers and _ (underscore) are allowed. The author has to ensure that they don't clash with platform provided variable names.
- `title` is a short one line information about this port/service.
- `description` is a multi line description about this port/service.
- `defaultValue` is the recommended port value to be shown in the app installation UI.
- `containerPort` is the port that the app is listening on (recall that each app has it's own networking namespace).
- `readOnly` flag indicates the port cannot be changed.
- `portCount` number of ports to allocate in sequence starting with the set port value.
- `enabledByDefault` flag is a UI hint as to whether this port should be shown as enabled or not at installation time.

In more detail:

- If the user decides to disable the SSH service, this environment variable `SSH_PORT` is absent. Applications must detect this on start up and disable these services.
- `SSH_PORT` is set to the value of the exposed port. Should the user choose to expose the SSH server on port 6000, then the value of `SSH_PORT` is 6000.
- `defaultValue` is **only** used for display purposes in the app installation UI. This value is independent of the value that the app is listening on. For example, the app can run an SSH server at port 22 but still recommend a value of 29418 to the user.
- `containerPort` is the port that the app is listening on. The Cloudron runtime will bridge the user chosen external port with the app specific `containerPort`. Cloudron Apps are containerized and each app has it's own networking namespace. As a result, different apps can have the same `containerPort` value because these values are namespaced.
- The environment variable `SSH_PORT` may be used by the app to display external URLs. For example, the app might want to display the SSH URL. In such a case, it would be incorrect to use the `containerPort` 22 or the `defaultValue` 29418 since this is not the value chosen by the user.
- `containerPort` is optional. When omitted, the bridged port numbers are the same internally and externally. Some apps use the same variable (in their code) for listen port and user visible display strings. When packaging these apps, it might be simpler to listen on `SSH_PORT` internally. In such cases, the app can omit the `containerPort` value and should instead reconfigure itself to listen internally on `SSH_PORT` on each start up.
- `portCount` is optional. When omitted, the count defaults to 1 and starts with the `defaultValue` or what the user has configured. The maximum count is 1000 ports. For resource and performance reasons, the number should be as low as possible and cannot overlap with existing ports used by other apps on the system. The port count is exposed as a environment variable with the `_COUNT` suffix. For example, `SSH_PORT_COUNT` above.
- `enabledByDefault` flag is a UI hint as to whether this port should be shown as enabled or not at installation time.

title

Type: string

Required: no

The `title` is the primary application title displayed on the Cloudron App Store.

Example:

```
"title": "Gitlab"
```

udpPorts

Type: object

Required: no

Syntax: Each key is the environment variable. Each value is an object containing `title`, `description` and `defaultValue`. An optional `containerPort` may be specified.

The `udpPorts` field provides information on the non-http TCP ports/services that your application is listening on. During installation, the user can decide how these ports are exposed from their Cloudron.

For example, if the application runs an SSH server at port 29418, this information is listed here. At installation time, the user can decide any of the following: * Expose the port with the suggested `defaultValue` to the outside world. This will only work if no other app is being exposed at same port. * Provide an alternate value on which the port is to be exposed to outside world. * Disable the port/service.

To illustrate, the application lists the ports as below:

```
"udpPorts": {
  "VPN_PORT": {
    "title": "VPN Port",
    "description": "Port over which OpenVPN server listens",
    "defaultValue": 11194,
    "containerPort": 1194,
    "portCount": 1
  }
},
```

In the above example:

- `VPN_PORT` is an app specific environment variable. Only strings, numbers and _ (underscore) are allowed. The author has to ensure that they don't clash with platform profided variable names.
- `title` is a short one line information about this port/service.
- `description` is a multi line description about this port/service.
- `defaultValue` is the recommended port value to be shown in the app installation UI.
- `containerPort` is the port that the app is listening on (recall that each app has it's own networking namespace).
- `readOnly` flag indicates the port cannot be changed.
- `portCount` number of ports to allocate in sequence starting with the set port value. When missing, this value defaults to 1.

upstreamVersion

Type: string

Required: no

The `upstreamVersion` field specifies the version of the app. This field is only used for display and information purpose.

Example:

```
"upsteamVersion": "1.0"
```

version

Type: semver string

Required: yes

The `version` field is a semver string that specifies the packages version. The version is used by the Cloudron to compare versions and to determine if an update is available.

Example:

```
"version": "1.1.0"
```

website

Type: url

Required: no

The `website` field is a URL where the user can read more about the application.

Example:

```
  "website": "https://example.com/myapp"
```

8.6 Publishing to Cloudron App Store

8.6.1 Requirements

Publishing your app to the Cloudron App Store will help your users install it easily on their server and keep it up-to-date.

Here are the rough steps involved in getting your app published:

- Before you start packaging, please leave a note in the App Wishlist category of our forum. If a topic for your app does not exist, please create a new one. This will avoid any duplicate work since our community has already packaged apps and maybe you can use those as a starting point. You can also use this to guage interest before packaging.
- Package your app for Cloudron following the tutorial and cheat sheet. Feel free to ask any questions or help in the App Packaging & Development category of our forum. See the pinned topics in that category for answers to FAQs.
- Once packaged, please leave a note in your app's App Wishlist topic in our forum. Our community can provide you with early feedback and pre-release testing.
- At this point, Cloudron team will look into your package and get it ready for publishing. Please note that the Cloudron team will take over the packaging of the app from this point on as we have no mechanism for 3rd party authors to publish and update apps. As part of this process, we add automated tests to ensure the app installs, backs up, restores and updates properly.

8.6.2 Licensing

We require app packages to have an Open Source license. MIT, GPL, BSD are popular choices but feel free to pick whatever you are comfortable with. Please note that the license only applies to the packaging code and not to your app. Your app can be Open Source or Commercial license.

The package will be maintained in our GitLab at https://git.cloudron.io. The original package authors will be given commit permissions to the repository (and we greatly appreciate packagers who continue maintaining it!). To aid this process, we recommend that the packaging source code is in a repository of it's own and not part of the app's code repository.

9. Internationalization

Cloudron supports translation of the dashboard and transactional emails like the Welcome email and Password reset.

Cloudron supports and ships with the following languages which are above 50% translated:

Detailed Translation Status

We are more than happy to accept contributions from the community for those other languages.

9.1 Maintaining translations

Translations are maintained in a Weblate instance at https://translate.cloudron.io. All translations are public and interested users can view translated strings. If users want to contribute please send us an email to support@cloudron.io with the intended `username` and language to contribute to and we will send you an invite.

9.1.1 Language maintainer

To ensure consistency across the translated terms and have a way to make decisions on proposed translations, we are looking for one maintainer for each language. The Cloudron team itself only natively speaks English and German, so for any other language we will rely on community members. In case you want to help us maintain a language, please first reach out to us at support@cloudron.io for a Weblate account and we can grant you permissions to maintain the corresponding language.

9.2 Testing

The `cloudron-translate-update` CLI tool fetches and updates translations from Weblate. This tool is already installed on Cloudron.

The following command will download translation files for all languages and updates those in the Cloudron installation:

```
sudo cloudron-translation-update
```

Once you have run the command, visit the Cloudron dashboard and reload the page. New languages can be activated immediately from the profile or settings view.

9.3 Workflow

Adding or improving individual translation strings all happens inside Weblate. If a translation in a language is missing, one can search for the English string in Weblate and it will show all languages with the current existing or non-existing translation. Simply propose a translation in the tool and the language maintainer will accept it. Once accepted, it can be tested as explained in the previous section.

9.3.1 Deprecated Strings

In case a previously used translation string is not in use in upcoming releases anymore, it will have the `Deprecated` label set in Weblate. Such strings do not need to be furher maintained and will eventually be fully purged from Weblate.

10. Pricing & Billing

10.1 Form of payments

We accept Visa, MasterCard, American Express and Discover.

We do not store any credit card information and all payments are handled through Stripe.

10.2 Billing period

We charge on a monthly or yearly subscription basis (prepaid). Any charges will be made on the first day of the billing period. Any service can be cancelled at any time, the service will then be terminated at the end of the current billing period.

10.3 Canceling subscription

Subscription can be cancelled anytime at cloudron.io. Once cancelled, we will stop billing you from the next month.

Your apps and server will continue to work forever despite the cancellation. However, they won't receive any updates and you won't be able to install new apps.

Cloudron and app packages have a rolling release model. This means that if you fall behind releases too much, we may not be able to support your setup, if you resume the subscription later.

10.4 VAT & Taxes

For customers in the EU, VAT (value added taxes) is NOT included in the price. The VAT amount depends on the rate of the country of your billing address. Customers outside the EU won't have any taxes applied.

10.5 Credit Card Hold & Duration

When you enter a card, we verify the card through our external payment provider stripe. We will attempt a small hold on your card. This hold will get removed after 7 days.

10.6 Payment failures

Payment failures can be resolved by adding a new credit card to your account. As we only allow one credit card per account, you will need to overwrite the old credit card information in your account page. Our billing system will automatically retry using any new credit card.

10.7 Invoices

Invoices can be download from here.