1. Shapps

- 1. Adding an Open Source App
- 2. Deleting Shapps

1. Adding an Open Source App

Sweet! So you want to add an open source app that does not yet exist on this documentation page and you think should be added. In this small tutorial we describe how you can add an open source app to this list. If however, you want to edit a chapter of an existing app then please read our Editing Page.

1. You most likely already did, but just to be sure first check in the **search bar** if the app does not yet exist already. Then also try different names as sometimes an apps name changes. Finally, just scroll through the categories list as it might be under a different category name then you expect.

2. If not then select the right categories i.e **Books** you believe the Shapp belongs to. Yes an application can be in multiple categories at the same time if it applies. For instance collaboration environments apps could also be in the notes category for making personal notes.

3. Give the **Chapter** the name of the Application you want to add

4. Add any Tags that are relevant to the application as well. If the tag does not yet exist than you can create the tag and also tag other applications with that tag if it applies.

5. Then in the description of the chapter you should briefly describe the goal of the app and include the following variables such that readers can quickly scan if the app is something they want to invest their time in or not..

6. The first page should contain screenshots from the application with some explanation what is seen on the screenshot

7. Optional a video instruction should be attached to the chapter as well.

8. Each Shapp must include a FAQ section as well. This allows users to ask questions and have them answered in the FAQ.

Shapp Variables

App Type

- Server side
- Client side

The App type indicates if an application is a **server side** or **client side** application. In the case of a server side application all data is stored on a server instead of on your device with a client side app. Server side has the advantage that you can view your data on multiple devices and you don't loose your data if you loose your device. However, for privacy reasons this app should be **E2EE** since otherwise the maintainer a friend or a community data center could access your data. Now client side apps have the advantage that data never leaves your device e.g your phone but loosing your device does mean loosing your data if not manually backed up.

Repository

- Forgejo
- Codeberg
- Gitlab
- Gitbucket
- Github

This should include the link to the public repository of the app. This can include one ore multiple locations.

Backups

- Automatic
- Manual

Backups is related to the **app type** since if the app is server side then your app's data is already stored on a server and thus in the case of loosing your client you don't loose your data. However, even if the app is client side backups can still happen automatically and be send to a server. For instance, an app could automatically store a backup every day on a local folder on your device. If however, that folder is synced with a folder on a server (using Syncthing for instance) then you can still receive back all your data if you loose your client device. It is just important to know if you need to do backups manually or if the app handles it for your regardless of being server or a client side app.

E2EE

- YES
- NO
- N/A

This value indicates if a server side app is end to end encrypted (E2EE) or not. That means that the administrator of an app cannot access the data of its users. However, this does not talk about metadata which has its own variable. Now, client side apps don't need to worry about this since the data never leaves their device to a server. If the app is client side then N/A can be used in that case

Admin Metadata

- Social Graph
- Online/Offline status
- What, When and where a person has been watching
- All
- URL Link to a Bookstack page if the list is to long
- ...

The Admin metadata access can be very large, which is indicated with ... Now, the goal is to write down all the metadata (data about the data) and administrator has access to. This is important for the privacy of users using someone else's server for an open source app. For instance Matrix is an E2EE chat application but it is still centrally controlled by an instance administrator that can read when you are online, with who you talk, how often you talk with a person, location data. If the list is to long then please provide a link to a page that includes all the metadata the admin has access to of such an app. If you still want to use the app but are worried about the metadata then you can always self host it yourself.

Authentication

- Password
- 2FA (Email, Authenticator)
- SSO
- Yubikey
- Biometric Verification

This section indicates which authentication methods there are possible for the application. Note down the ones that apply for the specific application.

Installation Documentation

- Docker Deployment

The value of this is Docker Deployment that includes a link to the official installation documentation website of the application. This should be by default the Quickstart guide if there is any. Also, installation instructions should follow the official documentation as well. However, you can give extra tips and tricks for anything that might improve the performance of the application, and reasoning why certain decisions are made.

Installation Videos

- Name of the creator of instruction video 1
- Name of the creator of instruction video 2
- Name of the creator of instruction video 3

The list of installation videos should include a list of videos that you think are great installation videos for a docker deployment of the application.

Community

- Matrix Channel
- Stoat
- Discord
- Simplex Channel
- Blog

This should be a list of all the community channels of the application where people can meet and mingle, enjoy each other's enthusiasm but as well to discuss problems or ways of doing things. If you solved a problem then add the solution in the FAQ section of the application.

Funding Model

- Donations
- Freemium
- Paid Cloud subscriptions
- Combination of the above

Open source projects are funded in different shapes and forms and this is where you specify how the particular app is funded. This can either be via donations, freemium, paid hosted subscriptions or a combination of them. Now, the most pure and end goal for all open source projects should be to end up with a donation based model to develop the applications and don't have any features hidden by a paywall. In the end contributions are done by multiple people and it is hard to determine who should and who shouldn't get paid. Therefore, money is only used to

At least for self hosting there should not be any money involved for obtaining all the full features of an application. On the other hand when an app is hosted/managed by a team then it makes more sense to pay for it since they need to pay for the data center space. However, it also must be said that if you are planning to host a public instance of an open source application then you can convince data centers to give you a discount at least.

Terms and Conditions

This is a link to the terms and conditions

- **E2EE =** Yes
- **Authentication:** Password, 2FA
- **Admin Metadata Access:** Access to social graph i.e who you share albums with
- **Website:** ente.com
- **Repository:** Github
- **Installation Documentation:** Docker Deployment
- **Installation Videos**: Jim's Garage
- **Community:** Discord**,** Github Discussions
- **Donations:** Ente does not work with donations to support the project. Instead it has paid hosted plans. To become truly global we hope to see a donation funding method in the future as other open source projects.
- **Terms and conditions for hosted plan:** Terms and conditions

he list is incomplete and hence we encourage others to add/edit where they seem fit. As stated in the title it is focused on the Docker VM workflow by using an optional container manager like Arcane.

It must be said that we focus on the Docker VM workflow described in the Application workflows. Other application workflows are described as well if they exist (Not all apps can be deployed with every workflow). The shelf categorizes shapps based on

1. **Category**
2. If the app is end to end encrypted (**E2EE**)
3. What **Metadata** will be available to the admin of the application for E2EE apps
4. How to **Install** them on your Docker Machine (with the help of Arcane)

Now, to use this shelf and deploy applications it is assumed that

- You have a **Docker VM** up and running
- A home network build with **OPNsense** or **OpenWrt**
- You have a **password manager** or **SSO** in place e.g Vaultwarden, Authentik to save the credentials for each application.
- Have a **reverse proxy** setup, with the focus on the Caddy reverse proxy.

Please update the books if you have time s.t others can benefit from you contributions ;)

2. Deleting Shapps