

# Backups Overall

We haven't gone into backups yet. Maybe we should. In general, writing a script to dump a database, tar files, etc, is not rocket science (though it's a bit tedious to set up). The trickiest part in any of these is getting the backed-up files to a different machine.

## Host

We actually haven't done a whole heck of a lot to the host. Certainly, we've got an `rc.conf` and a `pf.conf`. We've also got a script and a `cron` entry. We'll be adding more scripts (backup scripts) and more `cron` entries. And we've potentially got data directories on the host as well, considering we'll probably `cp/tar` files into some backup directory first (which we can then ZFS snapshot!) before then `scp`'ing to remote machine.

## Bookstack

This one is semi-tricky, but actually not bad. You must use the correct credentials to dump the database to a file, and you must grab a few additional directories that are resources used by the database but not actually saved inside it.

## Website

If you're using Gitea to populate the website's document root, then perhaps backing up Gitea will suit your needs. And in that case, you also most likely have a local copy of the `git` repository working tree, so you can already repopulate the document root in a pinch.

Whether or not you're using Gitea/`git`, it's generally a simple task to `tar` up the document root directory and `scp` it to a remote machine.

## Gitea

There are several ways to look at backing this up depending on how you're using it.

# Standalone

If you're reliant solely on Gitea, then you should back things up. There are two directories and a file.

## git home

This stores the `*.git` directories containing the commit history and whatnot. It feels weird to `tar` these up when they can easily be cloned by conventional means. Choose your poison.

## Database

This category consists of the db itself as well as some accompanying files and directories. The db contains the website configuration, including users. Backing this up will depend on the type of database. I used `sqlite`, which is not authenticated. The process for `mariadb` and others would be different, though you can follow the procedure used for BookStack, since that uses `mariadb`.

## `{..}/app.ini`

These are the runtime parameters.

# Mirrors

If you're using Gitea to mirror a website from Github or Gitlab (or wherever), then there isn't much of a need for a backup because you can just recreate the Gitea repo from the repo you're mirroring.

If you're using Github (or wherever) to mirror Gitea, then... you still don't have much to worry about. The reason for this (though I don't know if it applies outside of Github) is that Github does not allow you to mirror external repos. So if you're using Github to mirror Gitea, you have accomplished this by adding a post-receive hook to your Gitea repo that pushes the changes automatically to Github. So in this case too, you can restart the Gitea repo by importing the Github repo. However, it may make sense to back up your keys, considering you had to provide authorization for Gitea to push to Github, so there may be an SSH key that you'd want to just put back into Gitea rather than creating a new key pair and loading the newly created pubkey into Github.

# Bitwarden

There's actually not much to this. You can look at their recommendations here:

[https://github.com/dani-garcia/bitwarden\\_rs/wiki/Backing-up-your-vault](https://github.com/dani-garcia/bitwarden_rs/wiki/Backing-up-your-vault) which basically come down

to running `sqlite3 db > backup` and saving your icons.

# Caddy

No backup needed, really. I mean, in its current state, it'd take two seconds to replace. The more you add to it, the more you maybe want to copy a backup of the Caddyfile somewhere, but that's about it.

---

Revision #3

Created 8 August 2020 22:18:40 by scoob

Updated 31 July 2021 04:50:03 by scoob