Mastodon - Building the image

Our Mastodon deployment uses a custom-built image based on glitch-soc. Don't build it on the server; instead use your development computer to do so, and then push it to our container registry on Forgejo.

Local developer setup

You'll need a simple Mastodon dev setup to test our custom Solarpunk skin on a new release of glitch-soc Mastodon.

Make sure you have docker installed, with docker compose. If you're not in the docker group, you'll also have to prepend all the docker commands with `sudo`.

Pull the glitch-soc repository into a directory, like one named mastodon.

`git switch` into the latest release branch, like `stable-4.7`. Make sure you pull the latest revision. If you have an edit of the docker-compose.yml, you can do a `git stash`, `git pull`, then `git stash pop` and resolve any merge conflicts.

Follow the official instructions for setting up a development Mastodon setup in docker if you haven't already.

Then, clone our Solarpunk/mastodon repository to a parallel directory, such as solarpunk-mastodon.

Local testing of the skin (custom theme)

You can then copy the contents of the `app/` directory in our solarpunk-mastodon repo, into the developer glitch-soc repository to add the custom skin. That should be all that's necessary to

register the skin with glitch-soc.

Finally, test the developer instance of mastodon, and ensure the theme hasn't crashed and burned. It might print errors, or look wrong if so. You'll have to fix up the CSS if that happens. When Mastodon 5.0 comes out it'll be a bit of a pain - we plan to do a big rework then. If the theme is too hard to manage, you can delete the skin folder and use stock glitch-soc. In that case, you wouldn't have to build a custom image if you didn't want to, but it wouldn't hurt. We only have the custom image for the skin, currently.

I recommend creating some posts, interacting a bit, checking menus and such. Don't forget the admin interface as well.

You can also create a new account and it'll open a fake email page to let you verify (you can use Firefox Container Tabs to have multiple accounts when testing, just make sure to copy the verify link into the correct container). This lets you test what notifications look like and other interactions with users.

Building the production image

Once done testing, you can stop the dev containers. Time to build.

Open up `docker-compose.yml`, and uncomment the lines that start with `build:`. This will allow us to build the images.

There are two images: `mastodon`, and `mastodon-streaming`. If you use compose to build, it'll build both of them for you. Just run `docker compose build`. This likely will take a while. Assuming you tested and there were no CSS errors, this should go without hitches.

Once done, you need to tag the two images with the solarpunk registry tag. For convenience, you can export a variable `SLPK_VERSION` to the version of glitch you just built (replace the Xs):

```
export SLPK_VERSION=vX.X.X
```

Then run:

```
docker tag ghcr.io/glitch-soc/mastodon:$SLPK_VERSION
git.solarpunk.moe/solarpunk/$SLPK_VERSION-glitch
docker tag ghcr.io/glitch-soc/mastodon-streaming:$SLPK_VERSION
git.solarpunk.moe/solarpunk/mastodon-streaming:$SLPK_VERSION-glitch
```

Now, push to the repository. You'll need to have previously logged in with `docker login git.solarpunk.moe` to be able to push.

```
docker push git.solarpunk.moe/solarpunk/mastodon:$SLPK_VERSION-glitch
docker push git.solarpunk.moe/solarpunk/mastodon-streaming:$SLPK_VERSION-glitch
```

You're now ready to update the deployment on the server side.

Revision #7
Created 2026-10-31 05:28:51 UTC by vv
Updated 2026-11-20 15:56:56 UTC by vv