Setup

Jitsi can be ran in docker/podman containers. currently it runs on brix4, with an external STUN/TURN server at vps.shiny.space for clients with NAT issues or strict firewalls.

The basic setup is described in the docs: https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-docker/#quick-start
it boils down to: download release, unzip it, take the needed files (honestly it's only like 2 or 3... docker-compose.yml, gen-passwords.sh, env.example if needed), create the config directory where persistent data and custom config will be stored, set the desired ENV vars in .env, and run it with podman-compose/docker-compose/whatever.

At Shinyspace, we also needed some setup on the reverse proxy. find it here:
https://wiki.shiny.space/books/service-overview/page/nginx-reverse-proxy-configuration
Additionally: ports needed are 80/443 for web traffic, and 10000 for the jitsi video bridge - weirdly enough it worked in some situations without the bridge, but not always.

Here's the compose.yml, edited by lucas to actually forward the needed ENV vars in podman, it appears Docker handles env vars differently but podman-compose needs the ${ENV_VAR} part. it was manually added only where needed... fix later.

link: https://wiki.shiny.space/books/service-overview/page/jitsi-composeyml

it also has the systemd service set up for easy restarting: ~/.config/systemd/user/compose-jitsi.service

```
[Unit]
Description=Podman Compose MyService
Wants=network-online.target
After=network-online.target

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=%h/jitsi
ExecStart=/usr/bin/podman-compose up -d
ExecStop=/usr/bin/podman-compose down

[Install]
WantedBy=default.target
```

Note / Warning / TODO: currently, the shiny.space IP is specified in the jitsi config - if swisscom gives me a new one i will need to update it manually. Will add monitoring for it and write a script sometime to automate.

Note about rootless setup:
if you look at the compose file, you might see that this container doesn't use "userns-mode: keep-id" or the x-podman: in_pod: false settings we had to use for other rootles containers to fix permissions. Why this is different from other containers, i do not know - fact is we can edit the configs. BUT some data is still owned by another user, so when it comes to backup or migration stuff we need to deal with it (or just delete everything, for now the plan is that this is "ethereal" content, it can disappear any time).

Revision #5
Created 2024-10-09 23:48:58 UTC by Admin
Updated 2025-03-06 14:56:32 UTC by Admin