Altaïr □□ et ses services

- Matrix
- OTP
- Peertube
- qBittorrent
- PHP et caddy
- Nextcloud
- jellyfin
- Lamalle
- Authentification
    - Keycloak
    - Les roles
- libre office online
- freshrss
- Passbolt
- Resolver DNS
- brouter

Matrix

Il y a 3 logiciels principaux + des bridges:

- **Synapse**, le serveur
- **Element**, le client web
- **Matrix-authentication-service**, une brique d'auth pour matrix 2.0 (element X)

Les clients sont listé ici: `https://matrix.org/clients/` Tous ne supporte pas le sso ni le chiffrement. Les trois suivant fonctionnent.

Fluffy chat : Client android et ios plutot complet `https://gitlab.com/famedly/fluffychat`
Fractal : Pour Linux: client GTK moderne (il y a son cousin en QT6, mais fractal est mieux)
Element web : l'officiel, `client.matrix.fede.re`, il y a aussi l'app dans les apps store

Synapse

Synapse est installé via dnf. via le repo suivant: https://obs.infoserver.lv/project/monitor/matrix-synapse

Sa config est dans `/etc/synapse/homeserver.yaml`

Un example de config: `https://github.com/matrix-org/synapse/blob/master/docs/sample_config.yaml`

Si il y a un problème de connexion à la base de donnée:

- verifier si postgresql fonctionne
- verifier que le fichier `/var/lib/pgsql/data/pg_hba.conf` contient

```
host    synapse        synapse_user    ::1/128            md5
```

SSO (avec openid avant Matrix-authentication-service)

```
oidc_providers:
  - idp_id: keycloak
```

```
    idp_name: "PPSfleet"
    issuer: "https://auth.ppsfleet.navy/realms/Ppsfleet"
    client_id: "synapse"
    client_secret: "****"
    scopes: ["openid", "profile"]
    allow_existing_users: true # important so it will not create new account
    user_mapping_provider:
      config:
        localpart_template: "{{ user.preferred_username }}"
        display_name_template: "{{ user.name }}"
    backchannel_logout_enabled: true # Optional
```

Element

Element est installé dans `/srv/www/client.matrix.fede.re/current`

Matrix-authentication-service

Installation podman root: `matrix-auth`

```
podman run -e MAS_CONFIG=/app/config/config.yaml -v /etc/matrix-auth/config.yaml:/app/config/config.yaml:Z -p 127.0.0.1:9292:9292 -p 127.0.0.1:9291:9291 --name matrix-auth ghcr.io/element-hq/matrix-authentication-service
```

Les bridges

Les bridges sont installé via podman dans `/opt/matrix-bridges/<bridge>` avec l'user `matrix-bridge`

Facebook

Install

https://docs.mau.fi/bridges/python/setup/docker.html?bridge=facebook

Run

Commande pour run le docker facebook:

```
podman run --name matrix-facebook -d -p 127.0.0.1:29319:29319 -v /srv/matrix-bridges/facebook:/data:z
dock.mau.dev/mautrix/facebook:latest
```

Whatsapp

```
podman stop matrix-whatsapp
podman rename matrix-whatsapp matrix-whatsapp-old
podman pull dock.mau.dev/mautrix/whatsapp:latest
podman run --name matrix-whatsapp -d -p 127.0.0.1:29318:29318 -v /srv/matrix-bridges/whatsapp:/data:z
dock.mau.dev/mautrix/whatsapp:latest
```

Telegram

```
podman pull dock.mau.dev/mautrix/telegram:latest
podman run --name matrix-telegram -d -p 127.0.0.1:29317:29317 -v /opt/matrix-bridges/telegram:/data:z
dock.mau.dev/mautrix/telegram:latest
```

Discord

https://gitlab.com/mx-puppet/discord/mx-puppet-discord

Instagram

https://docs.mau.fi/bridges/general/docker-setup.html?bridge=instagram

Le `docker-compose.yml` ne sert finalement pas.

Docker network

```
podman network create matrix-instagram
```

Postgres

Pas de `sqlite` :/

```
podman run -d --name matrix-instagram-postgres -e POSTGRES_PASSWORD='*****' --network=matrix-instagram --restart always postgres:14
```

Run

```
podman run -d --name matrix-instagram --restart unless-stopped -v /srv/matrix-bridges/instagram:/data:z --network=matrix-instagram dock.mau.dev/mautrix/instagram:latest
```

OTP

Activer l'otp

```
google-authenticator -s ~/.ssh/google_authenticator
```

Config

`/etc/pam.d/sshd`

```
auth       required     pam_google_authenticator.so secret=/home/${USER}/.ssh/google_authenticator nullok
```

`/etc/ssh/sshd_config.d/10-custom.conf`

```
PasswordAuthentication no
ChallengeResponseAuthentication yes
UsePAM yes
```

Peertube

- `/data/peertube-data` : stoquage persistent de peertube
- `/etc/peertube` : dossier contenant les fichiers de configuration
- `/etc/peertube/peertube.nginx.conf` : configuration nginx

Base de données

Installation de postgres :

```
dnf install postgresql-server postgresql-contrib
postgresql-setup --initdb --unit postgresql
systemctl enable postgresql
sudo systemctl start postgresql
```

Création de la base de données :

```
# création de l'utilisateur
sudo -u postgres createuser -P --interactive
# création de la base de donnée
sudo -u postgres createdb -O peertube peertube_prod
```

Autoriser les containers à accéder à la base de données en ajoutant cette ligne à la fin du fichier `~postgres/data/pg_hba.conf` :

```
# Containers
host    all             all             10.88.0.0/24            md5
```

Pare-feu

Vérifier le firewall pour autorisé le trafic vers / depuis le réseau de podman, pour nftables ajouter ces lignes à la tables `inet filter` :

```
iifname cni-podman0 accept
oifname cni-podman0 accept
```

Peertube

Lancement de l'instance :

```
# Créer un pod pour Nginx et Peertube
podman pod create --name peertube --restart=unless-stopped -p 127.0.0.1:9000:9000 -p 0.0.0.0:1935:1935 -p
127.0.0.1:9080:9080

# Créer le conteneur Nginx
podman create --name peertube-nginx-1.25.3-1 --pod peertube \
  --restart=unless-stopped \
  -v /etc/peertube/peertube.nginx.conf:/etc/nginx/conf.d/default.conf:Z \
  -v /etc/peertube/:/config:Z \
  -v /data/peertube-data/storage:/var/www/peertube/storage:Z \
  nginx:1.25.3
# Démarrer Nginx
podman start peertube-nginx-1.25.3-1

# Créer le conteneur Peertube
podman create --name peertube-5.0.1-1 --pod peertube \
  --restart=unless-stopped \
  -v /data/peertube-data/:/var/lib/peertube:Z \
  -v /etc/peertube/:/config:Z \
  chocobozzz/peertube:v5.0.1-bullseye
# Démarrer Peertube
podman start peertube-5.0.1-1
```

Configuration Caddy

```
tube.fede.re {
    #respond "Ongoing maintenance, we'll be back soon :)" 503
    reverse_proxy {
        to localhost:9080
    }
}
```

Mise à jour de la configuration Nginx

La configuration à copier est à partir du commentaire `# Application` jusqu'à la fin du bloc. Après copie, il faut supprimer les blocs faisant référence au dossier `peertube-latest/client`.

En cas de problèmes

Au redémarrage d'Altaïr peertube risque de ne pas redémarrer correctement, si ce n'est pas up vérifier :

- que redis écoute bien sur 10.88.0.1 (`lsof -i :6379`) sinon le redémarrer via `systemctl restart redis`
- que postgres écoute bien sur 10.88.0.1 (`lsof -i :5432`) sinon le redémarrer via `systemctl restart postgres`
- Redemmarer le pod: `podman pod start peertube`

Troubleshouting 4.0.0

Python3 n'est pas installé dans le conteneur. Il faut installer python3 et symlink l'executable, depuis le conteneur:

- `apt install -y python3`
- `ln -s /usr/bin/python3 /usr/bin/local/python`

qBittorrent

La configuration de qBittorrent est un peu particulière pour éviter de sortir directement avec les IPs d'Altaïr. Un point de sortie (ici Alshain) est utilisé, et tout le traffic de qBittorrent est routé par ce point de sortie. Un tunnel Wireguard assure la liaison entre Altaïr et Alshain. L'IPv6 est routée directement alors que l'IPv4 est NATé (en deux fois, sur Altaïr et sur Alshain pour le moment).

Configuration réseaux :

Serveur	Interace	IP/Masque	Description
Alshain	int-ppsfleet	10.114.20.1/24	Réseau privée interne entre serveur PPSFleet.
Alshain	int-ppsfleet	2001:bc8:24d8:114:20::1/80	Équivalent IPv6 du réseau précédant. Le découpage du réseau est fait par bloc de 16 bits de manière à ce que chanque client aie un /96.
Altaïr	int-ppsfleet	10.114.20.10/24	IPv4 du client VPN sur Altaïr
Altaïr	int-ppsfleet	2001:bc8:24d8:114:20:10:0:1/112	IPv6 du client VPN sur Altaïr. Un sous découpage en /112 par interace dans ce /96 sur Altaïr.
Altaïr	net-external	10.89.0.1/24	Réseau IPv4 des conteneurs sortant via le VPN. Ce réseau est NATé sur Altaïr de manière à ce que le point de sortie ne voit que l'IP du client VPN.
Altaïr	net-external	2001:bc8:24d8:114:20:10:1:0/112	Réseau IPv6 des conteneurs sortant via le VPN.

Configuration du point VPN pour le point de sortie

1. Création les clés pour l'autentification et le chiffrement.

```
# Altaïr
bash -c '(umask 0077; wg genkey > altair.key)'
wg pubkey < altair.key > altair.pub
# Alshain
bash -c '(umask 0077; wg genkey > alshain.key)'
wg pubkey < alshain.key > alshain.pub

# Secret partagé
wg genpsk > altair-alshain.psk
```

2. Configurer le réseau sur Alshain en éditant le fichier `/etc/NetworkManager/system-connections/int-ppsfleet.nmconnection`

```
[connection]
id=int-ppsfleet
type=wireguard
interface-name=int-ppsfleet

[wireguard]
listen-port=51756
private-key=$ALSHAIN_PRIVATE_KEY

[wireguard-peer.$ALTAIR_PUBLIC_KEY]
preshared-key=$ALTAIR_ALSHAIN_SHARED_SECRET
preshared-key-flags=0
allowed-ips=10.114.20.10/32;2001:bc8:24d8:114:20:10::/96;

[ipv4]
address1=10.114.20.1/24
method=manual

[ipv6]
address1=2001:bc8:24d8:114:20::1/80
addr-gen-mode=stable-privacy
method=manual
```

3. Redémarrer NetworkManager sur Alshain

```
sudo systemctl restart NetworkManager
```

Configuration du réseau sur Altaïr

1. Créer la table de routage `external` avec comme ID `200` en ajoutant la ligne suivante au fichier `/etc/iproute2/rt_tables`.

```
200 external
```

2. Configurer le client VPN sur Altaïr en utilisant les clés générées précédemment. Créer le fichier `/etc/NetworkManager/system-connections/int-ppsfleet.nmconnection` avec le contenu suivant.

```
[connection]
id=int-ppsfleet
uuid=fe155cee-4941-3f9b-a441-a7fd21d2412a
type=wireguard
interface-name=int-ppsfleet

[wireguard]
peer-routes=false
private-key=$ALTAIR_PRIVATE_KEY

[wireguard-peer.$ALSHAIN_PUBLIC_KEY]
endpoint=[2001:bc8:24d8::]:51756
preshared-key=$ALTAIR_ALSHAIN_SHARED_SECRET
preshared-key-flags=0
allowed-ips=0.0.0.0/0;::/0;

[ipv4]
address1=10.114.20.10/24
method=manual
route-table=200
gateway=10.114.20.1
routing-rule1=priority 5 oif int-ppsfleet table 200

[ipv6]
addr-gen-mode=stable-privacy
address1=2001:bc8:24d8:114:20:10::1/112
method=manual
route-table=200
gateway=2001:bc8:24d8:114:20::1
```

```
routing-rule1=priority 5 oif int-ppsfleet table 200
```

3. Configurer l'interface de bridge pour les conteneurs en créant le fichier `/etc/NetworkManager/system-connections/net-external.nmconnection` avec le contenu suivant.

```
[connection]
id=net-external
uuid=d72a0e46-85bc-4b4c-bb64-fb904c463894
type=bridge
autoconnect=true
interface-name=net-external

[ethernet]

[bridge]
stp=false

[ipv4]
address1=10.89.0.1/24
method=manual
route-table=200
routing-rule1=priority 5 from 10.89.0.0/24 table 200

[ipv6]
addr-gen-mode=default
address1=2001:bc8:24d8:114:20:10:1:1/112
method=manual
route-table=200
routing-rule1=priority 5 from 2001:bc8:24d8:114:20:10:1:0/112 table 200
```

4. Redémarrer NetworkManager.

```
sudo systemctl restart NetworkManager
```

5. Configurer le firewall pour éviter d'avoir du NAT sur l'IPv6 en ajoutant les lignes suivante au fichier `/etc/sysconfig/nftables.conf`

```
table ip6 nat {
  chain POSTROUTING {
    type nat hook postrouting priority srcnat; policy accept;
    ip6 saddr 2001:bc8:24d8:114:20:10:1:0/112 return
```

```
    #iifname tun0 masquerade
  }
}
```

6. Puis redémarrer nftables.

```
sudo systemctl restart nftables
```

Créer le conteneur qBittorrent sur Altaïr

1. Créer le réseau Podman

   ```
   nmcli c down net-external
   podman network create --subnet 10.89.0.0/24 --subnet 2001:bc8:24d8:114:20:10:1:0/112 --interface-name=net-external --ignore net-external
   nmcli c up net-external
   ```

2. Démarrer le conteneur qBittorrent une première fois

   ```
   sudo podman run -d --name=qbittorrent \
     -e PUID=985 -e PGID=985 -e TZ=Europe/Paris -e WEBUI_PORT=8080 \
     -p '[::1]:8080:8080' -p 127.0.0.1:8080:8080 -p 6881:6881 -p 6881:6881/udp \
     -v /var/opt/qbittorrent/config:/config:Z -v /data/downloads:/downloads:Z \
     --restart always \
     --network net-external \
     lscr.io/linuxserver/qbittorrent:latest
   ```

3. Créer une policy SELinux custom (voir page sur le SELinux du wiki), et re-run le conteneur avec la policy

   ```
   sudo podman run --security-opt label=type:qbittorrent.process \
     -d --name=qbittorrent \
     -e PUID=985 -e PGID=985 -e TZ=Europe/Paris -e WEBUI_PORT=8080 \
     -p '[::1]:8080:8080' -p 127.0.0.1:8080:8080 -p 6881:6881 -p 6881:6881/udp \
     -v /var/opt/qbittorrent/config:/config -v /data/downloads:/data/downloads \
     --restart always \
     --network net-external \
     lscr.io/linuxserver/qbittorrent:latest
   ```

4. Créer le service associé et l'activer

```
sudo podman generate systemd --new --name qbittorrent | sudo tee
/etc/systemd/system/qbittorrent.service
systemctl daemon-reload
systemctl enable qbittorrent
```

PHP et caddy

INSTALL

On a besoin de php-fpm et caddy

Dans le fichier de conf `/etc/php-fpm.d/www.conf`, verifier que

```
listen.acl_users = apache,nginx,caddy
```

et

```
user = caddy
; RPM: Keep a group allowed to write in log dir.
group = caddy
```

La conf caddy

Ici on prend comme exemple un nom de domaine "test.ppsfleet.navy"

- creer le fichier `/etc/caddy/Caddyfile.d/test.ppsfleet.navy.conf` avec

```
test.ppsfleet.navy {

    # Set this path to your site's directory.
    root * /srv/www/test.ppsfleet.navy

    php_fastcgi unix//run/php-fpm/www.sock

    # Another common task is to set up a reverse proxy:
    # reverse_proxy localhost:8080

    # Or serve a PHP site through php-fpm:
    # php_fastcgi localhost:9000
```

```
    # Refer to the directive documentation for more options.
    # https://caddyserver.com/docs/caddyfile/directives

}
```

- Autoriser php a executer des scripts dans le dossier

```
chown -R caddy:caddy /srv/www/test.ppsfleet.navy/
semanage fcontext -a -t httpd_sys_script_exec_t '/srv/www/test.ppsfleet.navy(/.*)?'
restorecon -Rv /srv/www/test.ppsfleet.navy/
```

Nextcloud

PHP

la config php-fpm est dans /etc/php-fpm.d/nextcloud.conf

Des notes en vrac

```
php-mysqlnd php-pdo php-pecl-apcu php-xml php-process php-pecl-zip php-gd php-mbstring php-redis php-sodium



semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/nextcloud-(.*)/config(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/nextcloud-(.*)/apps(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/nextcloud-(.*)/core(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/.rnd'

semanage fcontext -a -t var_log_t '/var/log/caddy/(/*)?'

/srv/nextcloud/php-wsdlcache
/srv/nextcloud/php-opcache

semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/php-session(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/php-wsdlcache(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/nextcloud/php-opcache(/.*)?'
```

Upgrade:

- Modifier le fichier `/srv/nextcloud/nextcloud/config/config.php` et mettre maintenance à 'true'
- Telecharger la nouvelle release
- copier `/srv/nextcloud/nextcloud/config/config.php` dans la nouvelle release
- Supprimer le lien `/srv/nextcloud/nextcloud`
- Faire un lien de `/srv/nextcloud/nextcloud` vers la nouvelle release

- `chown -R nextcloud:nextcloud cloud.ppsfleet.navy`
- `find cloud.ppsfleet.navy/ -type d -exec chmod 750 {} \;`
- `find cloud.ppsfleet.navy/ -type f -exec chmod 640 {} \;`
- `sudo -u nextcloud php ./occ upgrade` (dans le dossier cloud.ppsfleet.navy)
- `sudo -u nextcloud php ./occ db:add-missing-indices`
- `restorecon -Rv /srv/nextcloud/`
- remettre maintenance à false

jellyfin

Installed via package manager (in RPM Fusion repo)

Podman way

```
podman run \
                                --detach \
                                --label "io.containers.autoupdate=registry" \
                                --name jellyfin-prod \
                                --publish 127.0.0.1:8096:8096/tcp \
                                --rm \
                                --user $(id -u):$(id -g) \
                                --userns keep-id \
                                --volume /srv/jellyfin/cache:/cache:Z \
                                --volume /srv/jellyfin/config:/config:Z \
                                --volume /data/media/video/film:/media/film \
                                --volume /data/media/video/series:/media/series \
                                --volume /data/media/video/animes:/media/animes \
                                --security-opt label=disable \
                                docker.io/jellyfin/jellyfin:latest
```

Lamalle

Répertoire de données

Création du répertoire chiffré :

```
truncate -s 3T /data/nextcloud-lamalle.luks
chown nextcloud:nextcloud nextcloud-lamalle.luks
sudo cryptsetup luksFormat nextcloud-lamalle.luks
sudo cryptsetup open nextcloud-lamalle.luks nextcloud-lamalle
sudo mkfs.ext4 -m0 -Lencrypted /dev/mapper/nextcloud-lamalle
```

Monter le répertoire :

```
sudo cryptsetup open /data/nextcloud-lamalle.luks nextcloud-lamalle
sudo mount /dev/mapper/nextcloud-lamalle /mnt/nextcloud-lamalle
```

Démonter le répertoire :

```
sudo mount /mnt/nextcloud-lamalle
sudo cryptsetup close nextcloud-lamalle
```

Source : https://askubuntu.com/questions/1338610/how-can-i-make-an-encrypted-file-directory-ubuntu-20-04-lts/1338741#1338741

Docker

```
podman pod create --name lamalle --restart=unless-stopped -p 127.0.0.1:1312:80

podman create --name lamalle-mariadb-10.6 --pod lamalle \
  --restart=unless-stopped \
  -v /mnt/nextcloud-lamalle/home/db:/var/lib/mysql:Z \
  -e MYSQL_USER=nextcloud \
  -e MYSQL_DATABASE=nextcloud \
```

```
  -e MYSQL_PASSWORD= \
  -e MYSQL_ROOT_PASSWORD= \
  mariadb:10.6

podman create --name lamalle-nextcloud-28.0.5 --pod lamalle \
  --restart=unless-stopped \
  -v /mnt/nextcloud-lamalle/home/nextcloud:/var/www/html:Z \
  -e MYSQL_USER=nextcloud \
  -e MYSQL_DATABASE=nextcloud \
  -e MYSQL_PASSWORD= \
  -e MYSQL_HOST=127.0.0.1 \
  nextcloud:28.0.5
```

Authentification

Keycloak

Les urls

- Url géneral: https://auth.ppsfleet.navy
- Pour gérer ses infos: https://auth.ppsfleet.navy/realms/Ppsfleet/account/
- Pour l'interface admin de ppsfleet: https://auth.ppsfleet.navy/admin/Ppsfleet/console/

TODO:

Configurer le login avec une clé yubikey ou autre (fido2 protocol)

https://refactorfirst.com/setup-fido2-passwordless-authentication-with-keycloak
https://www.aukfood.fr/les-differents-modes-dauthentification-sous-keycloak/

Installation (generic avec podman)

WARNING: verifier la version installé pour pas faire une descente de version.

```
podman create --name keycloak-26.0.7 \
        -v /opt/keycloak/themes:/opt/keycloak/themes \
        -p 127.0.0.1:9050:8080 \
        -e KEYCLOAK_ADMIN=master \
        -e KEYCLOAK_ADMIN_PASSWORD=... \
        -e KC_DB_URL=jdbc:postgresql://10.88.0.1:5432/keycloakdb \
        -e KC_DB_USERNAME=keycloakdb \
        -e KC_DB_PASSWORD=... \
        -e KC_DB=postgres \
        -e PROXY_ADDRESS_FORWARDING=true \
        quay.io/keycloak/keycloak:26.0.7 \
        -Djboss.bind.address.private=127.0.0.1 \
        -Djboss.bind.address=0.0.0.0 \
```

```
start --hostname auth.ppsfleet.navy --proxy-headers xforwarded --http-enabled true
```

Voir la page podman pour faire un service

Gérer le service

```
sudo -u keycloak systemctl --user start keycloak-26.2.4
```

Les services:

I - Bookstack

- https://github.com/elexis/elexis-environment/blob/master/docker/ee-util/assets/stage_ee_start_setup/keycloak/bookstack-saml.json
- https://github.com/BookStackApp/BookStack/issues/1157#issuecomment-585804153

```
AUTH_METHOD=saml2

# Set the display name to be shown on the login button.
# (Login with <name>)
SAML2_NAME=ppsfleet

# Name of the attribute which provides the user's email address

SAML2_EMAIL_ATTRIBUTE=email
SAML2_EXTERNAL_ID_ATTRIBUTE=username
SAML2_DISPLAY_NAME_ATTRIBUTES=firstName|lastName



# Enable SAML group sync.
SAML2_USER_TO_GROUPS=true

# Set the attribute from which BookStack will read groups names from.
SAML2_GROUP_ATTRIBUTE=Role

# Removed user from roles that don't match SAML groups upon login.
```

```
SAML2_REMOVE_FROM_GROUPS=true

# Name of the attribute(s) to use for the user's display name
# Can have mulitple attributes listed, separated with a '|' in which
# case those values will be joined with a space.
# Example: SAML2_DISPLAY_NAME_ATTRIBUTES=firstName|lastName
# Defaults to the ID value if not found.
#SAML2_DISPLAY_NAME_ATTRIBUTES=username

# Identity Provider entityID URL
SAML2_IDP_ENTITYID=https://auth.ppsfleet.navy/realms/Ppsfleet/protocol/saml/descriptor
```

II - Nextcloud

https://auth.ppsfleet.navy/auth/realms/Ppsfleet

Si il y a une erreur du type: "account not provisioned" c'est durement un problème de certificat.

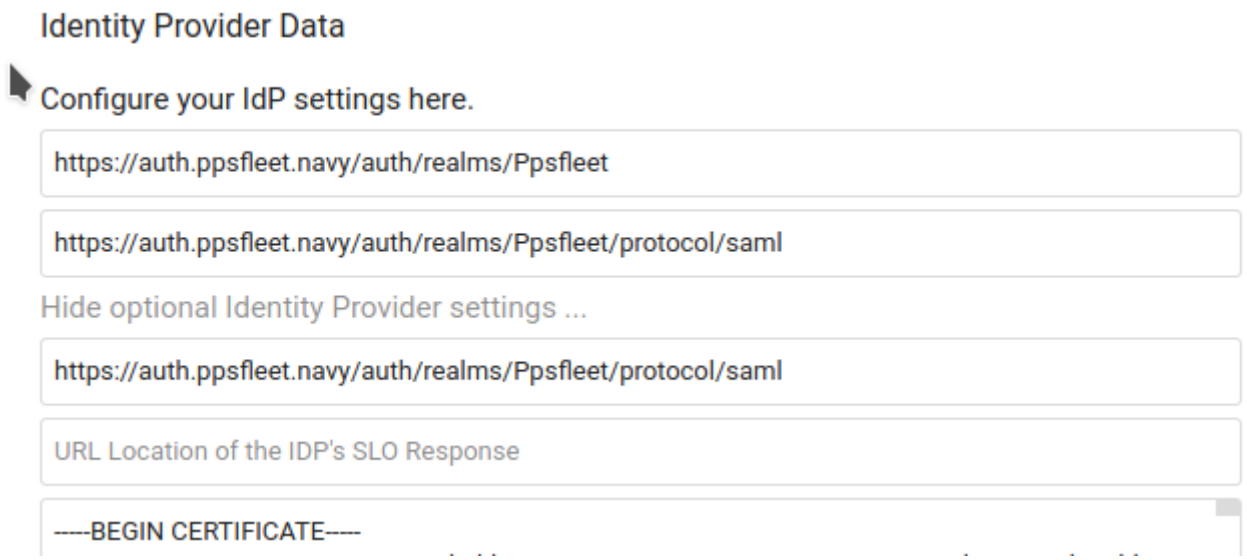
Le certificat se trouve dans keycloak: `Realm settings > keys > algorithm RS256 > Cetificate`

Il faut le mettre dans les paramètre de nextcloud: `SSO and SAML authentication > show optional Identity Provider settings > dernier champ`

Config:

Identifier: `https://auth.ppsfleet.navy/auth/realms/Ppsfleet`

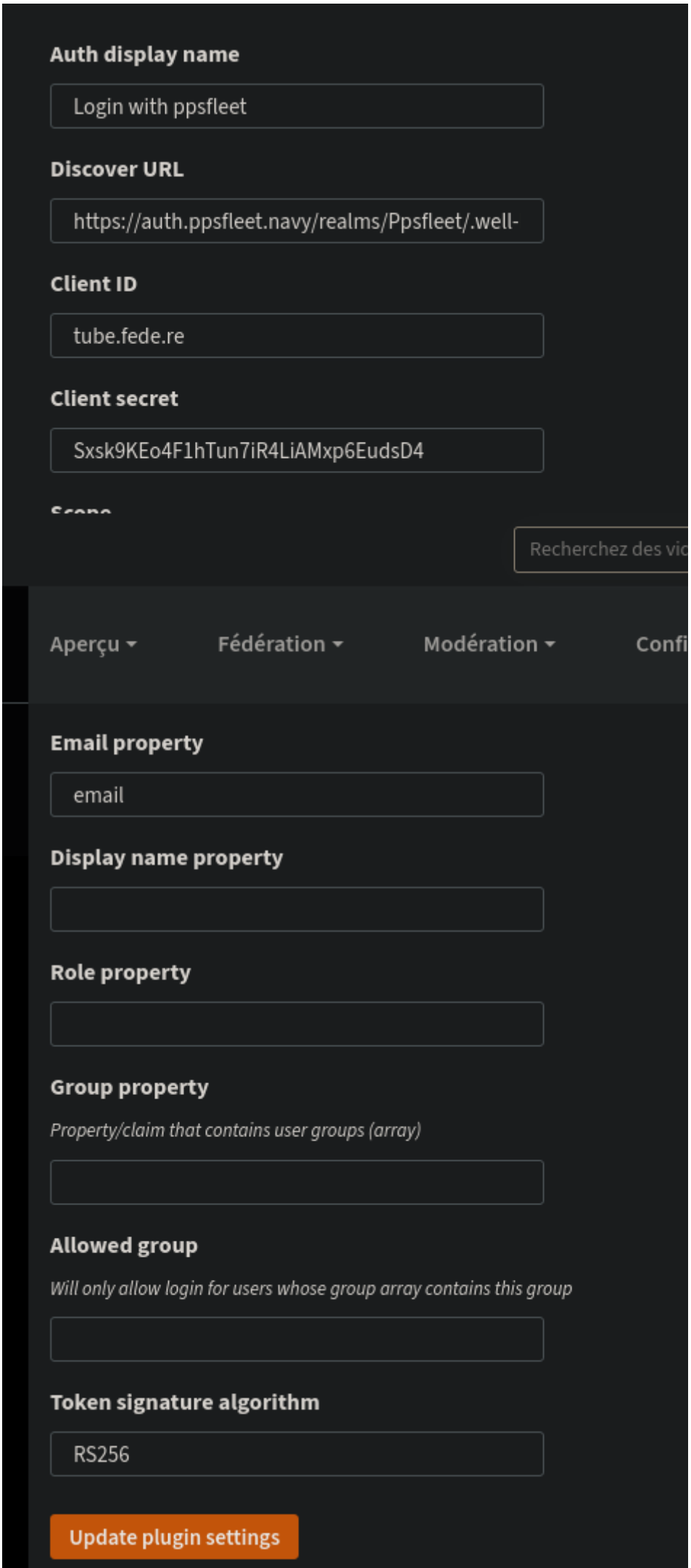
Url target: `https://auth.ppsfleet.navy/auth/realms/Ppsfleet/protocol/saml`



III - peertube

" https://auth.ppsfleet.navy/realms/Ppsfleet/.well-known/openid-configuration "

le client secret est dans l'onglet credentials de keycloak (celui du screen est plus valide)


Auth display name
Login with ppsfleet
Discover URL
https://auth.ppsfleet.navy/realms/Ppsfleet/.well-
Client ID
tube.fede.re
Client secret
Sxsk9KEo4F1hTun7iR4LiAMxp6EudsD4
Recherchez des vid
Aperçu
Fédération
Modération
Confi
Email property
email
Display name property
Role property
Group property
Property/claim that contains user groups (array)
Allowed group
Will only allow login for users whose group array contains this group
Token signature algorithm
RS256
Update plugin settings

Les roles

Groupes

Captains

Administrateur de la flotte. Ont tous les droits.

Roles :

- Nextcloud: admin, test, users
- Peertube: admin
- Wiki: Admin, Seaman

Gentlefolks

Groupe pour la famille, ou si on devient un chaton, pour nos invité·e·s premium, qui vont pas aider au bon fonctionnement du serveur mais vont utiliser le navire.

Sur nextcloud, capacité illimité. Sur le wiki, peuvent créer leur livre.

Roles :

- Nextcloud: users
- Peertube: user
- Wiki: Seaman

Travelers

Groupe pour les invité·e·s pas tant premium que ça. Capacité limité sur nextcloud. On peut les laisser editer les recettes ?

Roles :

- Nextcould: disabled

- Peertube: user
- Wiki: Seaman

Carpenters

S'occupent du travail manuel, ont accès au livre sur l'atelier. Pas de nextcloud, peuvent uploader des vidéos sur peertube, mais on les modère avant.

Roles:

- ...

libre office online

Start

```
systemctl start coolwsd.service
```

Nouvelles polices

Creer les fichiers dans /usr/share/fonts

Run:

```
fc-cache
coolconfig update-system-template
```

freshrss

Installation Podman

Doc: https://github.com/FreshRSS/FreshRSS/blob/edge/Docker/README.md

```
podman create --env-file ~freshrss/env \
  -v /srv/freshRss/podman-data/data:/var/www/FreshRSS/data:Z \
  -v /srv/freshRss/podman-data/extensions:/var/www/FreshRSS/extensions:Z \
  -p 127.0.0.1:7070:80 \
  --name freshrss \
  freshrss/freshrss
```

Installation on host

Ne permet pas d'utiliser le SSO :/

```
useradd freshrss -m -s /usr/bin/fish -d /srv/freshRss/


mkdir /srv/freshRss/php-wsdlcache /srv/freshRss/php-opcache /srv/freshRss/php-session


semanage fcontext -a -t httpd_sys_script_exec_t '/srv/freshRss(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/freshRss/php-session(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/freshRss/php-wsdlcache(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/freshRss/php-opcache(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/freshRss/FreshRSS-(.*)/data(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/srv/freshRss/logs(/.*)?'
```

cat /etc/php-fpm.d/freshrss.conf

```
listen = /run/php-fpm/freshrss.sock
```

```
php_admin_value[error_log] = /srv/freshRss/logs/php-fpm.error.log
php_admin_flag[log_errors] = on
php_admin_value[memory_limit] = 1024M



php_value[session.save_handler] = files
php_value[session.save_path]    = /srv/freshRss/php-session
php_value[soap.wsdl_cache_dir]  = /srv/freshRss/php-wsdlcache
;php_value[opcache.file_cache]  = /srv/freshRss/php-opcache
```

Extension

https://github.com/aidistan/freshrss-extensions

https://github.com/christian-putzke/freshrss-pocket-button

Passbolt

User passbolt

podman-compose (ne pas hésiter à ctrl-c un peu si ça stall)

connect to mariadb 10.88.0.1 on int-ppsfleet

Resolver DNS

Configuration Unbound

```
server
        interface: 127.0.0.1@8053
        interface: ::1@8053

        include: /etc/unbound/local.d/*.conf

        tls-service-key: ""
        tls-service-pem: ""

        https-port: 8053
        http-endpoint: "/dns-query"
        http-notls-downstream: no
```

Configuration Caddy

```
doh.fede.re {
reverse_proxy h2c://127.0.0.1:8053
}
```

Ad Block

```
curl -s https://sebsauvage.net/hosts/raw | awk '{ print "local-zone: \""$1"\" refuse"}' | sort -u >
/etc/unbound/local.d/adblock.conf
```

brouter

Installation

```
podman build -t brouter .
wget https://brouter.de/brouter/segments4/E0_N50.rd5
wget https://brouter.de/brouter/segments4/E0_N45.rd5
wget https://brouter.de/brouter/segments4/E0_N40.rd5
wget https://brouter.de/brouter/segments4/E10_N50.rd5
wget https://brouter.de/brouter/segments4/E10_N45.rd5
wget https://brouter.de/brouter/segments4/E10_N40.rd5
wget https://brouter.de/brouter/segments4/E5_N40.rd5
wget https://brouter.de/brouter/segments4/E5_N45.rd5
wget https://brouter.de/brouter/segments4/E5_N50.rd5

podman run  -d \
        -v /opt/brouter/segments:/segments4 \
        -p 17777:17777 \
        --name brouter \
        brouter
```