

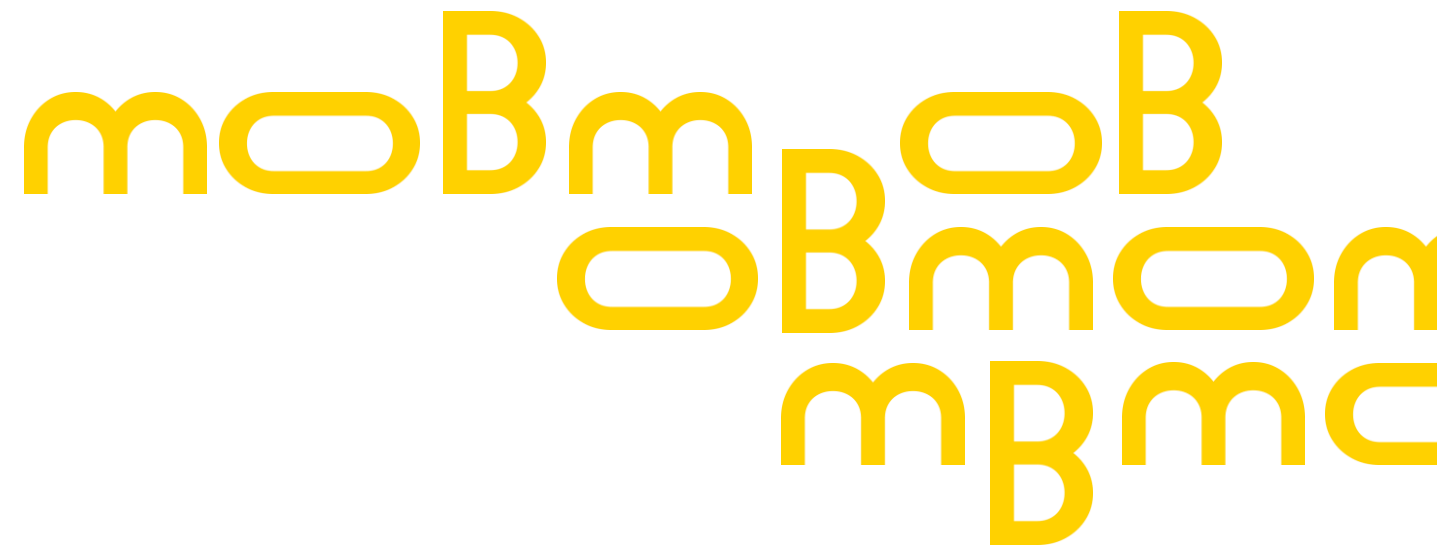
# Vault

## Présentation Technique



# Sommaire

1. Introduction sur le Vault
2. Description des fichiers utilisés
3. Utiliser le Vault en local
4. Éléments stockés dans le vault
5. Comportement sur les branches



# Introduction sur le Vault



# Introduction sur le Vault

- Génération d'un couple de clés
- Enregistrement de la clé publique dans l'API moB
- Chiffrement des justificatifs à l'aide de la clé publique
- Authentification par certificat
- Déchiffrement de la clé publique à l'aide de la clé privée côté website
- Scripts de génération automatique, de renouvellement et d'envoi de clés à moB













# Description des fichiers utilisés



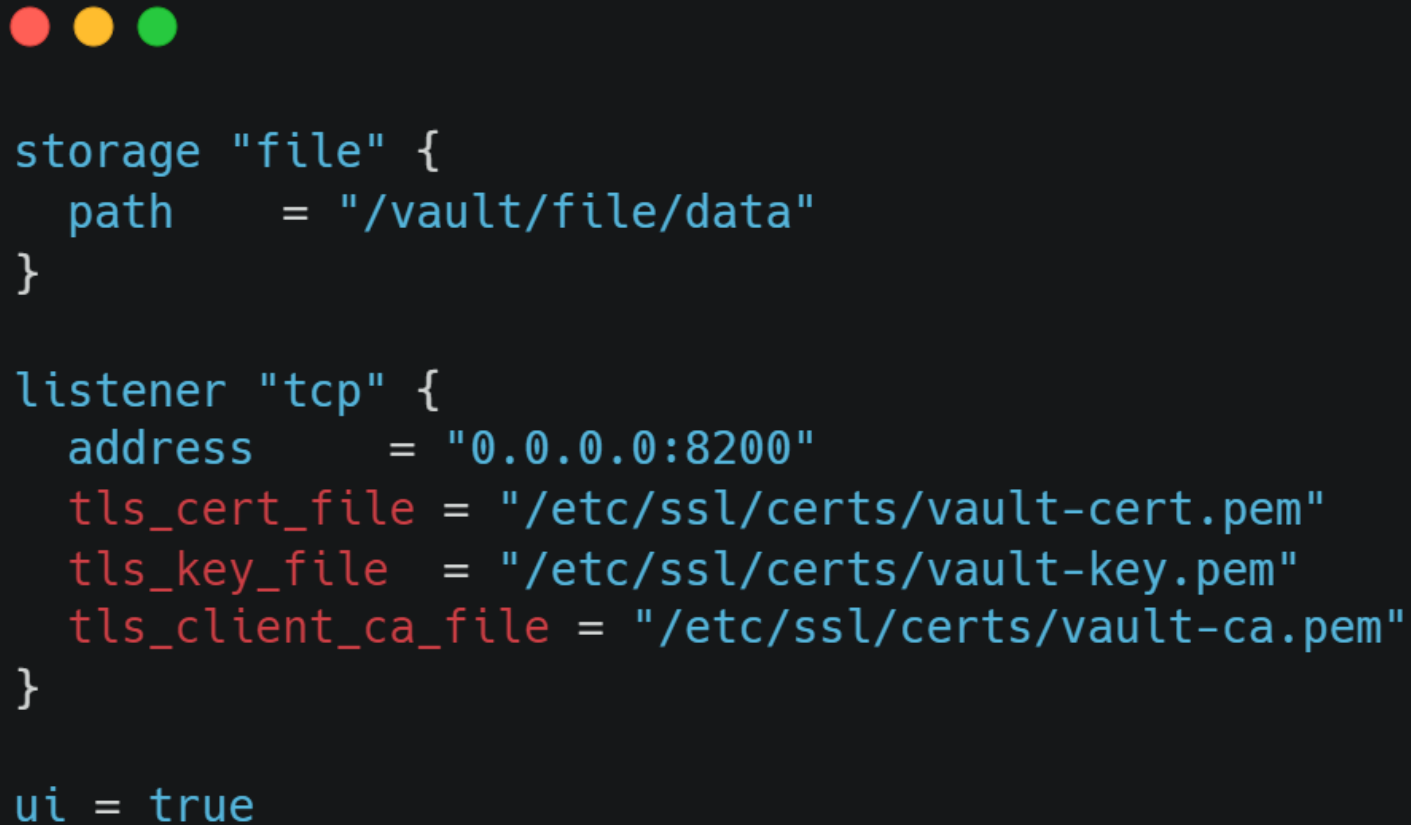
## Description des fichiers utilisés

## Contenu de la release

-  .env.sample
-  admin-policy.hcl
-  config.hcl
-  createCertificates.sh
-  Dockerfile
-  init-vault.sh
-  manager-policy.hcl
-  renew-key.sh
-  vault-crontab
-  vault-docker-compose.yml

# Config.hcl


Fichier de configuration du Vault



```
storage "file" {  
  path = "/vault/file/data"  
}  
  
listener "tcp" {  
  address = "0.0.0.0:8200"  
  tls_cert_file = "/etc/ssl/certs/vault-cert.pem"  
  tls_key_file = "/etc/ssl/certs/vault-key.pem"  
  tls_client_ca_file = "/etc/ssl/certs/vault-ca.pem"  
}  
  
ui = true
```

# .env.sample

Liste des variables d'environnement à renseigner



```
FUNDER_TOKEN=  
CLIENT_ID=  
CLIENT_SECRET=  
VAULT_ADDR=  
VAULT_API_ADDR=  
API_URL=  
IDP_URL=  
FUNDER_ID=  
VAULT_CACERT=  
VAULT_CERT=  
VAULT_KEY=  
VAULT_ROOT_CA=  
CLIENT_CA=  
ADMIN_CERT=  
ADMIN_CERT_KEY=
```



# Fichiers policy

Droits d'accès au Vault pour l'admin et le manager

- Pour l'admin : admin-policy.hcl
- Pour le manager : manager-policy.hcl

# Dockerfile

## Création de l'image Docker avec les librairies et les fichiers nécessaires

```
ARG BASE_IMAGE_VAULT
FROM ${BASE_IMAGE_VAULT}

ARG VAULT_CERT
ARG VAULT_KEY
ARG VAULT_ROOT_CA
ARG ADMIN_CERT
ARG ADMIN_CERT_KEY
ARG CLIENT_CA

RUN apk add --update -v coreutils apk-cron curl jq util-linux && rm -rf /var/cache/apk/*

COPY ./init-vault.sh /usr/local/bin/init-vault.sh
COPY ./renew-key.sh /usr/local/bin/renew-key.sh
COPY ./config.hcl /vault/config/config.hcl
COPY ./manager-policy.hcl /vault/config/manager-policy.hcl
COPY ./admin-policy.hcl /vault/config/admin-policy.hcl

COPY ${VAULT_CERT} /etc/ssl/certs/vault-cert.pem
COPY ${VAULT_KEY} /etc/ssl/certs/vault-key.pem
COPY ${VAULT_ROOT_CA} /etc/ssl/certs/vault-ca.pem

COPY ${CLIENT_CA} /etc/ssl/certs/client-ca.pem
COPY ${ADMIN_CERT} /etc/ssl/certs/admin-client-cert.pem
COPY ${ADMIN_CERT_KEY} /etc/ssl/certs/admin-client-key.pem

COPY vault-crontab /etc/cron.d/vault-crontab

ENV VAULT_CACERT=/etc/ssl/certs/vault-ca.pem

RUN chmod 777 -R /vault/config
RUN chmod 644 /etc/ssl/certs/vault-ca.pem
RUN chmod 644 /etc/ssl/certs/vault-cert.pem
RUN chmod 644 /etc/ssl/certs/vault-key.pem
RUN chmod 644 /etc/ssl/certs/client-ca.pem
RUN chmod 644 /etc/ssl/certs/admin-client-cert.pem
RUN chmod 644 /etc/ssl/certs/admin-client-key.pem
RUN chmod +x /usr/local/bin/init-vault.sh
RUN chmod +x /usr/local/bin/renew-key.sh
RUN chmod 0644 /etc/cron.d/vault-crontab && crontab /etc/cron.d/vault-crontab

EXPOSE 8200
```

# Vault-docker-compose.yml

Lancer les 3 services du Vault : principal, init et cron

- 3 service
  - Vault : instance du vault lancée, port 8200
  - Vault-init : script d'initialisation, création du couple de clés
  - Vault-cron : cronjob de renouvellement de clés

# Create-vault-release.sh

Script de création du zip de release à fournir aux financeurs

- `./create-vault-release.sh 1.0.0`

```
#!/bin/sh

if [ "$#" -ne 1 ]
then
    echo "Usage: Must supply a release version"
    echo "Example : 1.0.0"
else
    RELEASE_VERSION=$1

    zip mcm-vault-v$RELEASE_VERSION.zip admin-policy.hcl config.hcl createCertificates.sh Dockerfile init-vault.sh
    manager-policy.hcl renew-key.sh vault-crontab vault-docker-compose.yml
fi
```

# CreateCertificates.sh

## Script de création de certificats serveur et client

- `./createCertificates.sh vault.example.com`
- Création d'une autorité de certification
- Création d'un certificat serveur pour vault.example.com signé par l'autorité de certification
- Création d'un certificat client Admin signé par l'autorité de certification
- Création d'un certificat client Manager signé par l'autorité de certification

# Init-vault.sh

## Script d'initialisation du Vault

- Lancé par le service vault-init
- Initialise le vault : création des clés de scellement
- Descellement du Vault
- Login en tant que Root
- Création des policy
- Autorisation de l'authentification par certificat et création des rôles Admin et Manager
- Login à l'aide du certificat Admin
- Configuration du CORS
- Activation des paths kv et transit
- Génération d'un couple de clés et envoi de la clé publique aux financeurs côté moB



# Renew-key.sh

## Script de renouvellement du couple de clés

- Login à l'aide du certificat Admin
- Si le couple de clés a expiré (existe depuis + de 6 mois) :
  - **Rotation du couple de clés**
  - **Envoi de la nouvelle clé publique aux financeurs côté moB**

# Vault-crontab

## Cronjob pour lancé le script renouvellement de clés

- Cronjob lancé par le service vault-cron
- **Se déclenche tous les samedis à 3h du matin**
- **Lance le script renew-key.sh**

```
0 3 * * 6 /usr/local/bin/renew-key.sh
```

# Utiliser le Vault en local



## Utiliser le vault en local

## Lancer l'api dans un terminal WSL

- Récupérer l'ip WSL : `ip addr | grep eth0`
- Remplir un fichier `.env` dans le dossier `api/`
- **`docker compose -f api-docker-compose.yml up --build`**



```
API_URL=http://172.18.124.191:3000
IDP_URL=http://172.18.124.191:9000
S3_URL=http://172.18.124.191:9001
IDP_DB_HOST=172.18.124.191
MONGO_HOST=172.18.124.191
BUS_HOST=172.18.124.191:5672
CLIENT_SECRET_KEY_KEYCLOAK_API=ZJF1m5GCPggbP74n3AX84kWMJiIvthtY
CLAMAV_HOST=172.18.124.191
CLAMAV_PORT=3310
API_KEY=apikey
ADMIN_ACCES_ROLE=content_editor
PORT=3000
```

## Utiliser le vault en local

# Créer des certificats pour le vault local

- Choisir un nom de domaine, ex: ***vault.example.com***
- ***Cd vault/***
- ***./createCertificates.sh vault.example.com***
- ***Remplir les informations demandées pendant le déroulement du script (voir diapos suivantes)***

# Utiliser le vault en local

## Créer des certificats pour le vault local

### Génération du ROOT CA

- Enter pass phrase for rootCA.key : **pass**
- Enter pass phrase for rootCA.key : **pass**
- Verifying - Enter pass phrase for rootCA.key : **pass**
- Enter pass phrase for rootCA.key: **pass**
- Country Name (2 letter code) [AU]: **FR**
- State or Province Name (full name) [Some-State]: **IDF**
- Locality Name (eg, city) []: **Paris**
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: **MCM**
- Organizational Unit Name (eg, section) []: **MOB**
- Common Name (e.g. server FQDN or YOUR name) []: **MCM ROOT CA**
- Email Address []: **mail@example.com**

### Génération du Certificat Serveur vault.example.com

- Country Name (2 letter code) [AU]: **FR**
- State or Province Name (full name) [Some-State]: **IDF**
- Locality Name (eg, city) []: **Paris**
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: **MCM**
- Organizational Unit Name (eg, section) []: **MOB**
- Common Name (e.g. server FQDN or YOUR name) []: **vault.example.com**
- Email Address []: **mail@example.com**
- A challenge password []: **pass**
- An optional company name []: **MCM**
- Enter pass phrase for rootCA.key: **pass**
- Enter Export Password: **pass**
- Verifying - Enter Export Password: **pass**



# Lancement du vault en local

## Créer des certificats pour le vault local

### Génération du certificate client admin

- Country Name (2 letter code) [AU]: **FR**
- State or Province Name (full name) [Some-State]: **IDF**
- Locality Name (eg, city) []: **Paris**
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: **MCM**
- Organizational Unit Name (eg, section) []: **MOB**
- Common Name (e.g. server FQDN or YOUR name) []: **MCM ADMIN CERT**
- Email Address []: **[mail@example.com](mailto:mail@example.com)**
- A challenge password []: **pass**
- An optional company name []: **MCM**
- Enter pass phrase for rootCA.key: **pass**
- Enter Export Password: **pass**
- Verifying - Enter Export Password: **pass**

### Génération du certificate client manager

- Country Name (2 letter code) [AU]: **FR**
- State or Province Name (full name) [Some-State]: **IDF**
- Locality Name (eg, city) []: **Paris**
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: **MCM**
- Organizational Unit Name (eg, section) []: **MOB**
- Common Name (e.g. server FQDN or YOUR name) []: **MCM ADMIN CERT**
- Email Address []: **[mail@example.com](mailto:mail@example.com)**
- A challenge password []: **pass**
- An optional company name []: **MCM**
- Enter pass phrase for rootCA.key: **pass**
- Enter Export Password: **pass**
- Verifying - Enter Export Password: **pass**

## Lancement du vault en local

## Contenu du dossier vault.example.com généré

Ubuntu-20.04 > home > wsadaoui > platform > vault > vault.example.com				
	Nom	Modifié le	Type	Taille
✦	admin-client-cert.pem	11/01/2023 15:07	Fichier PEM	2 Ko
✦	admin-client-key.pem	11/01/2023 15:04	Fichier PEM	2 Ko
✦	admin-client-req.pem	11/01/2023 15:07	Fichier PEM	1 Ko
✦	manager-client-cert.pem	11/01/2023 15:19	Fichier PEM	2 Ko
✦	manager-client-cert.pfx	11/01/2023 15:19	Échange d'informa...	3 Ko
✦	manager-client-key.pem	11/01/2023 15:09	Fichier PEM	2 Ko
	manager-client-req.pem	11/01/2023 15:13	Fichier PEM	2 Ko
	rootCA.key	11/01/2023 14:54	Fichier KEY	2 Ko
	rootCA.pem	11/01/2023 15:00	Fichier PEM	2 Ko
	rootCA.srl	11/01/2023 15:19	Fichier SRL	1 Ko
	vault.example.com.crt	11/01/2023 15:04	Certificat de sécuri...	2 Ko
	vault.example.com.csr	11/01/2023 15:03	Fichier CSR	2 Ko
	vault.example.com.ext	11/01/2023 15:03	Fichier EXT	1 Ko
	vault.example.com.key	11/01/2023 15:00	Fichier KEY	2 Ko
	vault.example.com.p12	11/01/2023 15:04	Échange d'informa...	3 Ko

## Lancement du vault en local

# Importer les certificats dans le gestionnaire de certificats

- Dans les autorités de certification racines de confiance
  - rootCA.pem
- Dans les certificats personnels de l'utilisateur
  - manager-client-cert.pfx

## Lancement du vault en local

## Mise à jour des fichiers hosts

- Dans wsl, relier l'ip de WSL au nom de domaine créé dans le fichier ***/etc/hosts***, et faire de même sous Windows (en tant qu'admin) dans le fichier ***C:/Windows/System32/drivers/etc/hosts***, exemple :
  - **172.18.124.191 vault.example.com**

## Lancement du vault en local

## Lancer le vault dans un terminal WSL

- Récupérer l'ip WSL : `ip addr | grep eth0`
- Remplir un fichier `.env` dans le dossier `api/`
- ***`docker compose -f vault-docker-compose.yml up --build`***

```
FUNDER_TOKEN=hvsD6vIJ0MjxQil8u9j0vExmL25
AVAILABLE_KEYS=2
CLIENT_ID=simulation-maas-backend
CLIENT_SECRET=uLMt10kqqDKmEgr8DourBFY20nlSwo88
VAULT_ADDR=https://vault.example.com:8200
VAULT_API_ADDR=https://vault.example.com:8200
API_URL=http://172.18.124.20:3000
IDP_URL=http://172.18.124.20:9000
FUNDER_IDS=74ff920f-8d96-42e0-842d-1d5a9bbf1dfe,f3ce6e1e-b5b3-4935-a1ba-0fbd11fe88a1
VAULT_CERT=./vault.example.com/vault.example.com.crt
VAULT_KEY=./vault.example.com/vault.example.com.key
VAULT_ROOT_CA=./vault.example.com/rootCA.pem
CLIENT_CA=./vault.example.com/rootCA.pem
ADMIN_CERT=./vault.example.com/admin-client-cert.pem
ADMIN_CERT_KEY=./vault.example.com/admin-client-key.pem
```

# Éléments stockés dans le Vault





# Path kv

[← secrets](#) [← kv](#)

kv

Version 2

Secrets

Configuration

Filter secrets

Create secret +

<div><div></div><div><a href="#">74ff920f-8d96-42e0-842d-1d5a9bbf1dfe</a></div></div>	...
<div><div></div><div><a href="#">90f9cec4-b015-4b88-8e62-5b9c29b71b82</a></div></div>	...
<div><div></div><div><a href="#">f3ce6e1e-b5b3-4935-a1ba-0fbd11fe88a1</a></div></div>	...
<div><div></div><div><a href="#">key-version</a></div></div>	...
<div><div></div><div><a href="#">simulation-maas-backend</a></div></div>	...

# Path kv/\${CLEINT\_ID}

< kv < simulation-maas-backend

## simulation-maas-backend

Secret Metadata

<div><div></div>JSON</div>		Delete	Copy ▾	Version 1 ▾	Create new version +
Key	Value	Version created Oct 12, 2022 10:01 AM			
funderIdList	<div><div></div><div></div>74ff920f-8d96-42e0-842d-1d5a9bbf1dfe, f3ce6e1e-b5b3-4935-a1ba-0fbd11fe88a1, 90f9cec4-b015-4b88-8e62-5b9c29b71b82</div>				

# Path kv/key-version

[< kv](#) [< key-version](#)

## key-version

[Secret](#) [Metadata](#)

☒ JSON

Delete

Copy ▾

Version 1 ▾

Create new version +

Version Data

```
{
  "keyPairId": "c297fa53-e4c1-4680-985f-bcc4679668a1",
  "version": 1
}
```

# Path kv/\${funderId}

< kv < key-version

## key-version

Secret Metadata

JSON

Delete

Copy

Version 1

Create new version

### Version Data

```
{
  "keyPairId": "c297fa53-e4c1-4680-985f-bcc4679668a1",
  "version": 1
}
```

# Comportement du Vault sur les branches



# Comportement du Vault sur les branches

## Actions manuelles à réaliser en preprod

- Avant le lancement du Vault si pas déjà fait :
  - Créer les rôles vault et service\_vault pour simulation-maas-backend
  - Créer les Protocol Mappers group membership, vault\_rôle et vault\_name
  - Ajouter les financeurs pour lesquels on souhaite utiliser le vault au group membership du service account user de simulation-maas-backend
- Après le lancement du vault:
  - Se connecter au Vault et créer dans le path kv le secret « simulation-maas-backend » et ajouter à l'intérieur la clé « funderIdList » avec pour valeur la liste des ids des financeurs



# Comportement du Vault sur les branches

## Actions automatisées

- Initialisation du Vault : génération des clés de descellement, descellement du Vault, stockage des clés
- Création du couple clé publique/privée pour le client simulation-maas-backend
- Sauvegarde dans le Vault des éléments nécessaires au renouvellement du couple de clés
- Envoi de la première clé publique pour la liste des financeurs référencés (\$FUNDER\_IDS) liés à simulation-maas-backend en préprod
- Lancement du cron job pour renouveler les clés en préprod

# Comportement du Vault sur les branches

**Utiliser le Vault sur les branches ou sur la préprod avec un autre client que simulation-maas-backend**

- Voir la procédure dans le ppt suivant :

[Guide de configuration du Vault en preview et preprod](#)