

Supabase + s3 storage OVH

- [Brancher OVH Object Storage à Supabase self-hosted](#)

Brancher OVH Object Storage à Supabase self-hosted

1. Créer le bucket OVH

Dans le Public Cloud OVH :

- **Object Storage → Créer un container**
- Type : Standard
- Région : choisir la plus proche (ex: `sbg`, `gra`)
- Récupérer le **nom du bucket** + l'**endpoint** (visible dans la fiche du bucket)

Créer un utilisateur S3 :

- **Users & Roles → Créer un utilisateur S3**
- Récupérer `access_key` et `secret_key` (le secret n'est affiché qu'une fois)

2. Créer le fichier override Supabase

Supabase est en backend `file` par défaut. On override via un fichier compose dédié.

```
nano ~/project/supabase/supabase-project/docker-compose.ovh.yml
```

```
services:
  storage:
    environment:
      STORAGE_BACKEND: s3
      GLOBAL_S3_BUCKET: NOM_DU_BUCKET
      GLOBAL_S3_ENDPOINT: https://ENDPOINT_OVH
      GLOBAL_S3_FORCE_PATH_STYLE: "true"
      AWS_ACCESS_KEY_ID: ACCESS_KEY
      AWS_SECRET_ACCESS_KEY: SECRET_KEY
      AWS_DEFAULT_REGION: REGION
      REGION: REGION
      TENANT_ID: nom_projet
```

Point clé : utiliser les variables `AWS_*` (pas `GLOBAL_S3_ACCESS_KEY`) — c'est le SDK AWS qui lit ces variables.

3. Lancer Supabase avec l'override

```
cd ~/project/supabase/supabase-project
podman compose -f docker-compose.yml -f docker-compose.ovh.yml up -d
```

Vérifier les variables dans le conteneur :

```
podman exec supabase-storage env | grep -E "AWS|STORAGE_BACKEND"
```

Vérifier les logs au démarrage :

```
podman logs supabase-storage 2>&1 | tail -50
```

4. Tester

Dans Supabase Studio → Storage → New Bucket → uploader un fichier. Le fichier apparaît dans le bucket OVH sous le chemin : `BUCKET_OVH/TENANT_ID/bucket_supabase/fichier`.

Architecture finale

```
App — upload → Supabase Storage API → OVH Object Storage
                |
                ▼
            storage.objects (Postgres)
            [métadonnées + permissions RLS]
```

- **Postgres (VPS)** : métadonnées + tables métier (références aux fichiers)
- **OVH S3** : fichiers binaires uniquement
- Le lien se fait automatiquement via l'API Supabase, rien à configurer côté tables

Limitations à connaître

- Un seul bucket OVH possible avec self-hosted (`GLOBAL_S3_BUCKET` unique)
- Les buckets logiques Supabase deviennent des sous-dossiers dans le bucket OVH
- L'IHM Supabase Studio ne montre pas explicitement le lien S3 — c'est transparent