

VPS - organisation du dossier data

- [260316-montage du dossier data](#)
- [260316-Montage de la base de données alteris_spatial_db](#)
- [230319 - Observatoire d'actualités par Freshrss](#)
- [260320-Mise en place de la sauvegarde backup des bases de données du VPS](#)
- [260323- Creation d'un environnement virtuel python](#)

260316-montage du dossier data

Définition du dossier data

Le dossier "data" est le dossier d'échanges au sein du VPS entre les utilisateurs et POSTGIS - et à travers lequel l'IA exécute ses commandes de téléversements et de copie. C'est un miroir numérique avec pour références les codes INSEE des territoires.

Organisation de data pour l'étude de La Ciotat

```
/home/debian/data/  
└─ Regions/  
    └─ 93/          # Région (PACA)  
        └─ 13/      # Département (Bouches-du-Rhône)  
            └─ 200054807/ # EPCI (Métropole Aix-Marseille-Provence)  
                └─ 13028/   # Commune (La Ciotat)
```

Ligne de commande

```
# On se place dans le dossier de la commune  
cd /home/debian/data/Regions/93/13/200054807/13028  
  
# Téléchargement du contour GeoJSON simplifié (précision 100m pour la performance SIG)  
curl -o 13028_contour.geojson  
"https://geo.api.gouv.fr/communes/13028?fields=contour&format=geojson"
```

260316-Montage de la base de données alteris_spatial_db

Base de données - `alteris_spatial_db`

C'est une victoire majeure pour l'infrastructure **Altéris**. Le silence du terminal après ces lignes confirme que l'importation a réussi : vos **1,6 Go** de données régionales sont désormais structurés, indexés et prêts à l'emploi dans **PostGIS**.

Vous venez de passer de la "donnée morte" (fichiers stockés) à la "**donnée vivante**" (moteur spatial capable de calculs complexes).

État de votre base `alteris_spatial_db`

L'intégralité du squelette territorial de la région PACA est en place :

- **La morphologie** : `buildings` (bâti) et `landuse` (occupation des sols).
- **Les flux** : `roads`, `railways`, `transport` et `trafic`.
- **La vie sociale** : `pois` (points d'intérêt) et `places`.
- **L'environnement** : `water`, `waterways` et `natural`.

Pourquoi c'est le socle de la "Maîtrise du Réel" ?

Grâce aux index **GIST** que nous avons créés, vous pouvez maintenant interroger cette base pour identifier les **vulnérabilités territoriales**. Par exemple, vous pouvez mathématiquement prouver l'isolement d'un quartier ou la dangerosité d'une rupture de flux.

Prochaine étape : Isoler La Ciotat (13028)

Comme la base contient toute la région, nous allons créer une **Vue spatiale** dédiée à La Ciotat. Cela permettra à votre **QGIS Server** de n'afficher que votre zone d'étude sans charger le reste de la région, garantissant ainsi une fluidité maximale.

Voulez-vous que je vous donne la requête SQL pour créer automatiquement les vues "La Ciotat" pour les bâtiments et les routes en utilisant le contour que nous avons téléchargé ?

votre conteneur ne s'appelle pas `alteris-db`, mais `alteris_postgis`. C'est une excellente nouvelle, car il est "Up" depuis 2 jours et le port **5432** est correctement exposé.

“ On peut maintenant lancer l'importation avec les bons paramètres.

1. Préparer la base de données

On utilise le nom exact du conteneur pour créer la base `alteris_spatial_db` à l'intérieur :

Bash

```
docker exec -it alteris_postgis psql -U alteris_admin -d postgres -c "CREATE
DATABASE alteris_spatial_db;"
docker exec -it alteris_postgis psql -U alteris_admin -d alteris_spatial_db
-c "CREATE EXTENSION IF NOT EXISTS postgis;"
```

2. Lancer le téléversement massif

Maintenant que le chemin est balisé, voici la commande finale pour injecter vos **1,6 Go** de données régionales.

Bash

```
cd /home/debian/data/Regions/93/Geofabrik
```

```

for f in *.gpkg; do
    table_name=$(basename "$f" .gpkg | tr '[:upper:]' '[:lower:]' | tr ' ' '_' )
    echo "--- Importation : $f -> table $table_name ---"

    ogr2ogr -f "PostgreSQL" \
        PG:"host=127.0.0.1 port=5432 user=alteris_admin
dbname=alteris_spatial_db password=Alteris2026" \
        "$f" \
        -nln "$table_name" \
        -overwrite \
        -lco GEOMETRY_NAME=geom \
        -lco SPATIAL_INDEX=GIST \
        -nlt PROMOTE_TO_MULTI
done

```

Pourquoi c'est le socle de votre Maîtrise du Réel ?

En réussissant cette connexion, vous liez vos données brutes (le "réel" géométrique) à votre moteur d'analyse (PostGIS).

- **Sécurité territoriale** : Vous allez pouvoir requêter les **vulnérabilités** (ex: densité de bâtiments sans éclairage public répertorié).
- **Anti-fragilité** : Votre structure Docker est saine, les ports sont ouverts, et les outils (GDAL) communiquent enfin avec les services (PostGIS).

Une fois que l'importation de `buildings.gpkg` est terminée (c'est le plus gros morceau), voulez-vous que je vous donne la commande pour vérifier dans `alteris_pgadmin` que les tables sont bien peuplées ?

done

```

--- Importation : buildings.gpkg -> table buildings ---
--- Importation : landuse.gpkg -> table landuse ---
--- Importation : natural_points.gpkg -> table natural_points ---
--- Importation : natural.gpkg -> table natural ---

```

```
--- Importation : places.points.gpkg -> table places_points ---
--- Importation : places.gpkg -> table places ---
--- Importation : pofw.points.gpkg -> table pofw_points ---
--- Importation : pofw.gpkg -> table pofw ---
--- Importation : pois.points.gpkg -> table pois_points ---
--- Importation : pois.gpkg -> table pois ---
--- Importation : railways.gpkg -> table railways ---
--- Importation : roads.gpkg -> table roads ---
--- Importation : trafic.points.gpkg -> table trafic_points ---
--- Importation : trafic.gpkg -> table trafic ---
--- Importation : transport.points.gpkg -> table transport_points ---
--- Importation : transport.gpkg -> table transport ---
--- Importation : water.gpkg -> table water ---
--- Importation : waterways.gpkg -> table waterways ---
debian@vps-02c1f983-vps-ovh-net:~/data/Regions/93/Geofabrik$
```

230319 - Observatoire d'actualités par Freshrss

Nos indicateurs d'alerte

T1 - Crise de confiance - "La Ciotat" (rumeur OR complotisme OR "tensions sociales" OR "déficit démocratique")

T2 - Servitude technologique - "La Ciotat" (digital OR cyberattaque OR cybersécurité OR GAFAM OR "désinformation numérique")

T3 - Changement climatique - "La Ciotat" (inondation OR submersion OR crue OR imperméabilisation OR "risques naturels" OR "incendies de forêt" OR "îlot de chaleur" OR ICU)

T4 - Incertitudes économiques - "La Ciotat" ("entreprise fermeture" OR "plan social" OR "liquidation judiciaire")

T5 - Fragmentation sociale - "La Ciotat" (communautarisme OR "identité sociale" OR islamisme OR "mixité sociale" OR "repli communautaire" or "rupture générationnelle")

T6 - Vulnérabilités urbaines - "La Ciotat" (insécurité OR agression OR violence OR délinquance OR "délit routier" OR "refus d'obtempérer" OR "excès de vitesse" OR "délit de fuite" OR "plainte au tribunal" OR "incivilités" OR "injures" OR "brigade anti-criminalité ")

Pour aller encore plus loin - Autres outils pour muscler l'Observatoire - A FAIRE

Pour aller plus loin que Google Alerts (qui peut être parfois lent ou incomplet), voici des compléments :

- **RSS-Bridge** : C'est le compagnon indispensable de FreshRSS. Il permet de transformer en flux RSS des sites qui n'en ont pas (réseaux sociaux comme X/Twitter, pages Facebook de mairies, sites de presse locale). Idéal pour capter les **"Signaux Faibles"**.
 - **n8n ou Make (Auto-hébergé)** : Pour automatiser le flux. n8n peut lire FreshRSS, envoyer le texte à une IA pour le résumer et déterminer quel `ID_SQL` est concerné, puis l'insérer proprement dans votre base PostgreSQL.
 - **ChangeTower ou Distill.io** : Pour surveiller des changements spécifiques sur des pages web (ex: compte-rendu de conseil municipal, page "Arrêtés préfectoraux").
-

📄 Maintenance de l'Observatoire Altéris (FreshRSS)

Cette fiche regroupe les procédures techniques pour gérer la veille automatique entre FreshRSS et le référentiel d'indicateurs Altéris.

1. Accès à la base de données

Pour administrer les mots-clés et la vue de veille depuis le terminal du VPS :

Bash

```
docker exec -it freshrss-db psql -U freshrss -d freshrss
```

(Sortir avec `\q`)

2. Gestion du Dictionnaire (Table `ref_mots_cles`)

Le dictionnaire fait le lien entre un mot-clé détecté dans la presse et un **ID_SQL** de fiche Obsidian.

Action	Commande SQL
Ajouter	<code>INSERT INTO ref_mots_cles VALUES ('ID_SQL', 'mot-clé', 'Thématique');</code>
Supprimer	<code>DELETE FROM ref_mots_cles WHERE mot_cle = 'mot-clé';</code>
Lister	<code>SELECT * FROM ref_mots_cles ORDER BY thematique;</code>

Note : Pour une apostrophe (ex: refus d'obtempérer), il faut la doubler dans la commande SQL :

`'refus d''obtempérer'`.

"Ajouter une itération à l'observatoire des actualités"

Il est utile d'ajouter à votre fiche de maintenance pour le jour où vous voudrez ajouter une **nouvelle commune d'étude** (ex: Cassis ou Aubagne) :

“ 3. Ajout d'une nouvelle Commune

- Créer le dossier : `mkdir -p /home/debian/data/Cassis/`
- Dupliquer le script python et changer la variable `BASE_DIR` et le mot-clé `'La Ciotat'` dans la requête SQL.
- Ajouter une ligne dans le `crontab`.

Script Python de construction de la fiche

python3 /home/debian/scripts/archiver_veille.py

```
import subprocess
import os
from datetime import datetime

# CONFIG
BASE_DIR = "/home/debian/data/La_Ciotat/"
NOW = datetime.now()
DEST_FILE = os.path.join(BASE_DIR, f"Veille_{NOW.strftime('%Y_%m')}.md")
SQL_FILE = "/tmp/query.sql"

def run():
    os.makedirs(BASE_DIR, exist_ok=True)

    # 1. On écrit la requête dans un fichier temporaire pour éviter les problèmes de
    guillemets
    query = """
    SELECT date, title, link
    FROM "Nexte_entry"
    WHERE (title ILIKE '%La Ciotat%' OR content ILIKE '%La Ciotat%')
    AND date > (extract(epoch from now()) - 172800)
    ORDER BY date DESC;
    """
    with open(SQL_FILE, "w") as f:
        f.write(query)

    # 2. On envoie le fichier à Docker
    # On utilise cat pour injecter le SQL dans psql
    cmd = f"cat {SQL_FILE} | docker exec -i freshrss-db psql -U freshrss -d freshrss -t -A -F"
    '|'

    process = subprocess.run(cmd, shell=True, capture_output=True, text=True)
```

```

if process.returncode != 0:
    print(f"Erreur : {process.stderr}")
    return

output = process.stdout.strip()
if not output:
    print("INFO : Aucun nouvel article trouvé (Base vide ou pas de correspondance).")
    return

# 3. Traitement du Markdown
lignes = output.split('\n')

if not os.path.exists(DEST_FILE):
    with open(DEST_FILE, "w", encoding="utf-8") as f:
        f.write(f"# 📁 Archive de Veille : {NOW.strftime('%B %Y')}\n")
        f.write("| Date | Sujet | Lien |\n| :--- | :--- | :--- |\n")

with open(DEST_FILE, "r", encoding="utf-8") as r:
    contenu_existant = r.read()

with open(DEST_FILE, "a", encoding="utf-8") as f:
    count = 0
    for l in lignes:
        if '|' in l:
            p = l.split('|')
            try:
                dt = datetime.fromtimestamp(float(p[0])).strftime("%d/%m %H:%M")
                titre = p[1].replace("|", "-")
                url = p[2]
                if url not in contenu_existant:
                    f.write(f"| {dt} | {titre} | [Lien]({url}) |\n")
                    count += 1
            except:
                continue
    print(f"Succès : {count} nouveaux articles ajoutés à {os.path.basename(DEST_FILE)}")

if __name__ == "__main__":
    run()

```

Exemple fiche

▣▣ I_T3_03 - Îlots de Chaleur Urbains (ICU)

Territoire : La Ciotat (13028)

Indicateur : T3 - Risques Climatiques

Dernière analyse : 19/03/2026

1. Diagnostic de Vulnérabilité (Le "Froid")

L'analyse morphologique de La Ciotat montre trois zones critiques où l'albédo est faible et la rétention thermique élevée :

- **Cœur Historique :** Densité minérale extrême, manque de canopée.
- **Zone Athélia :** Grandes surfaces de toitures sombres et parkings bitumés.
- **Front de Mer :** Réverbération solaire importante sur les quais minéralisés.

2. Signaux Faibles & Actualités (Le "Chaud")

Cette section affiche les données collectées automatiquement par votre script Python.

▣▣ Archives de Presse : Mars 2026

“ [!INFO] Rappel

Les données ci-dessous sont extraites du fichier `[[[Veille_2026_03]]]`.

Date	Sujet	Source / Lien
22/03 09:15	Végétalisation : La mairie annonce 200 nouveaux arbres pour le quartier de l'Abeille.	Consulter l'article
20/03 14:30	Canicule : Le plan bleu activé par anticipation dans les EHPAD de La Ciotat.	Consulter l'article
18/03 11:00	Urbanisme : Inauguration de la nouvelle place des Escales : trop de béton ?	Consulter l'article

3. Mise à jour du Filtre (Vue SQL)

Si la structure change ou si vous changez d'utilisateur FreshRSS, exécutez ce bloc pour reconstruire la vue de synthèse :

SQL

```
CREATE OR REPLACE VIEW vue_veille_alteris AS
SELECT DISTINCT
    e.date AS date_parution,
    e.title AS titre,
    ref.id_sql,
    ref.thematique,
    '13028' AS code_insee,
    'La Ciotat' AS territoire,
    e.link AS url
FROM Nexte_entry e
JOIN ref_mots_cles ref ON (e.title ILIKE '%' || ref.mot_cle || '%' OR e.content ILIKE '%' ||
ref.mot_cle || '%')
```

```
WHERE e.title ILIKE '%La Ciotat%' OR e.content ILIKE '%La Ciotat%'
ORDER BY e.date DESC;
```

4. Consultation des résultats

Pour vérifier les derniers signaux faibles captés :

SQL

```
SELECT * FROM vue_veille_alteris LIMIT 10;
```

#Documentation #Maintenance #SQL #FreshRSS #Observatoire

On va ajouter un outil de surveillance des itérations dans Alteris avec Freshrss. Objectif - créer des alertes google sur de mots clés et construire des indicateurs sur les publications. Un observatoire des thématiques comme agression insécurité confiance risques....L'outil que nous choisissons esst un agregateur de flux RSS qui regroupe les flux des alertes Google. Qu'en penses tu ? Quels autres outil pourraient être mis en place ? Notamment comment on insère dans notre base de données les articles récupérés ??

Alteris Sécurité a dit

C'est une brique stratégique majeure que vous ajoutez là. Passer du diagnostic "froid" (données statistiques de l'INSEE, SIG) à un observatoire "chaud" (actualités, signaux faibles, climat social) permet de donner vie à votre **Roue de la Résilience**.

L'utilisation de **FreshRSS** pour centraliser les alertes Google est un choix excellent pour la souveraineté : vous gardez la main sur vos données, vous n'êtes pas dépendant d'un outil de veille coûteux (SaaS), et vous pouvez auto-héberger le flux.

1. Pourquoi FreshRSS + Google Alerts est un bon duo

- **Réactivité** : Vous captez l'actualité locale (ex: "incendie La Ciotat", "agression", "inondation") dès sa publication.
- **Structuration** : FreshRSS permet de créer des catégories correspondant exactement à vos thématiques (T1 à T6).

- **Filtrage** : Vous pouvez filtrer les doublons ou les sources non pertinentes avant l'insertion en base.
-

2. Comment insérer les articles dans votre base PostGIS ?

C'est ici que la magie opère pour faire le lien avec vos fiches Obsidian. Pour que vos articles alimentent `alteris_spatial_db`, vous avez besoin d'un script "pont" (souvent en Python).

La méthode recommandée : L'API FreshRSS vers PostgreSQL

1. **Récupération** : Python se connecte à l'API de FreshRSS (ou lit directement sa base de données si elle est sur le même serveur).
2. **Traitement (NLP)** : Le script analyse le texte pour identifier :
 - La **Thématique** (via vos mots-clés : agression = T6, inondation = T3).
 - Le **Lieu** (recherche du nom de la commune ou du code INSEE dans le texte).
3. **Insertion** : Une ligne est créée dans une table `journal_evenements` de votre base de données.

Exemple de structure de table SQL :

- `id_article` (Serial)
 - `date_publi` (Timestamp)
 - `id_sql` (Lien vers votre fiche Obsidian, ex: S_T6_06)
 - `code_insee` (Lien vers le territoire, ex: 13028)
 - `url_source` (Lien vers l'article)
 - `sentiment_score` (Indicateur de gravité)
-

3. Autres outils pour muscler l'Observatoire

Pour aller plus loin que Google Alerts (qui peut être parfois lent ou incomplet), voici des compléments :

- **RSS-Bridge** : C'est le compagnon indispensable de FreshRSS. Il permet de transformer en flux RSS des sites qui n'en ont pas (réseaux sociaux comme X/Twitter, pages Facebook de mairies, sites de presse locale). Idéal pour capter les **"Signaux Faibles"**.
- **n8n ou Make (Auto-hébergé)** : Pour automatiser le flux. n8n peut lire FreshRSS, envoyer le texte à une IA pour le résumer et déterminer quel `ID_SQL` est concerné, puis l'insérer proprement dans votre base PostgreSQL.
- **ChangeTower ou Distill.io** : Pour surveiller des changements spécifiques sur des pages web (ex: compte-rendu de conseil municipal, page "Arrêtés préfectoraux").

Vos premières Alertes Google (Ciblage territorial)

Allez sur [Google Alerts](#) et créez des alertes comme :

- "La Ciotat" agression -> Catégorie **T6**
- "La Ciotat" inondation -> Catégorie **T3**
- "La Ciotat" entreprise fermeture -> Catégorie **T4**

GOOGLE ALERTE

La "Recherche Exacte" vs "Recherche Large"

- **Sans guillemets (La Ciotat)** : Google cherche les pages qui contiennent "La" ET "Ciotat", mais pas forcément côte à côte. Vous pourriez tomber sur un article qui dit : "**La** ville de Marseille est plus grande que **Ciotat**-plage" ou même des résultats sur "La" (article défini).
- **Avec guillemets ("La Ciotat")** : Vous forcez Google à chercher l'expression **exacte** et **indissociable**. Les deux mots doivent être l'un après l'autre. C'est ce qu'on appelle une "chaîne de caractères".

Éviter les "Bruits" (Résultats non pertinents)

Pour une ville comme La Ciotat, le risque est limité, mais imaginez pour d'autres communes :

- Si vous cherchez **Le Mans**, sans guillemets, vous allez recevoir des milliers d'alertes contenant juste le mot "le".
- Si vous cherchez **Septèmes-les-Vallons**, les guillemets garantissent que vous ne recevez pas des articles sur "les vallons" en général.

Combiner avec vos thématiques (Opérateurs Booléens)

Dans vos alertes Google, la syntaxe idéale pour Altéris ressemble à ceci :

“ "La Ciotat" (inondation OR crue OR submersion) ”

- **"La Ciotat"** : Cible précisément le territoire.
- **(...)** : Regroupe vos mots-clés de la thématique **T3**.
- **OR** : Dit à Google "L'un ou l'autre de ces mots".

La syntaxe optimisée

Voici la version que je vous recommande de copier-coller :

“ "La Ciotat" (inondation OR crue OR submersion) ”

"La Ciotat" (inondation OR submersion OR crue OR imperméabilisation OR "risques majeurs" OR "îlot de chaleur" OR ICU)

Pourquoi ces ajustements ?

- **"îlot de chaleur"** : Avec l'accent et les guillemets.
- **ICU** : L'acronyme technique est très utilisé dans les articles d'urbanisme, c'est un mot-clé "pro" indispensable.
- **"risques majeurs"** : Le mot **risques** seul est trop générique (risques financiers, risques de grève, etc.). En ajoutant "majeurs" ou en le remplaçant par **"risques naturels"**, vous ciblez directement votre thématique **T3**.

STACK DOCKER de l'observatoire

version: '3.8'

services:

freshrss:

image: freshrss/freshrss:latest

container_name: freshrss

environment:

- PUID=1000

- PGID=1000

- TZ=Europe/Paris

- CRON_MIN=*/15

Paramètres de connexion à la DB

- DB_TYPE=postgres

- DB_HOST=freshrss-db

- DB_USER=freshrss

- DB_PASSWORD=alteris_secure_pass # <-- Changez-le si besoin

- DB_BASE=freshrss

volumes:

- /home/debian/docker/freshrss/data:/var/www/FreshRSS/data

ports:

- "8081:80"

depends_on:

- freshrss-db

restart: unless-stopped

freshrss-db:

image: postgres:15-alpine

container_name: freshrss-db

environment:

- POSTGRES_USER=freshrss
- POSTGRES_PASSWORD=alteris_secure_pass # <-- Doit être le même qu'au dessus
- POSTGRES_DB=freshrss

volumes:

- /home/debian/docker/freshrss/db:/var/lib/postgresql/data

restart: unless-stopped

260320-Mise en place de la sauvegarde backup des bases de données du VPS

C'est noté. On va droit à l'essentiel pour que tu puisses copier-coller ce "manuel de survie" dans tes notes. Voici la synthèse technique de ta configuration de sauvegarde **VPS Alteris → NAS Synology**.

▣ 1. Côté VPS (Préparation des données)

Installation du moteur :

Bash

```
sudo apt update && sudo apt install rsync -y
```

Script de backup SQL (`/usr/local/bin/docker_db_backup.sh`) :

Bash

```
#!/bin/bash

# --- CONFIGURATION ---

BACKUP_DIR="/backups/sql"
DATE=$(date +%Y-%m-%d)
mkdir -p $BACKUP_DIR
```

```

echo "--- Début du backup global : $(date) ---"

# 1. Backup Bookstack (MariaDB) - OK (Utilisateur dédié)
echo "[]Sauvegarde Bookstack..."
docker exec bookstack_db /usr/bin/mariadb-dump -u bookstack -p'260202ALTERIS!' bookstackapp --
single-transaction --quick | gzip > $BACKUP_DIR/bookstack_db_$DATE.sql.gz

# 2. Backup FreshRSS (PostgreSQL) - OK
echo "[]Sauvegarde FreshRSS..."
PG_USER_FRESH=$(docker exec freshrss-db printenv POSTGRES_USER || echo "postgres")
docker exec freshrss-db pg_dumpall -U $PG_USER_FRESH | gzip > $BACKUP_DIR/freshrss-
db_$DATE.sql.gz

# 3. Backup Alteris PostGIS (PostgreSQL) - OK
echo "[]Sauvegarde Alteris PostGIS..."
PG_USER_ALT=$(docker exec alteris_postgis printenv POSTGRES_USER || echo "postgres")
docker exec alteris_postgis pg_dumpall -U $PG_USER_ALT | gzip >
$BACKUP_DIR/alteris_postgis_$DATE.sql.gz

# --- PERMISSIONS ET NETTOYAGE ---
# On donne les droits à l'utilisateur du NAS
chown -R synobackup:synobackup $BACKUP_DIR
chmod -R 644 $BACKUP_DIR/*.gz

# Nettoyage : On ne garde que le fichier du jour sur le VPS
find $BACKUP_DIR -type f -mtime +0 -name "*.sql.gz" -not -name "*$DATE*" -delete

echo "--- Fin du backup global : $(date) ---"

```

Note : `chmod +x /usr/local/bin/docker_db_backup.sh` pour l'activer.

Automatisation (Crontab `sudo crontab -e`) :

Bash

```
0 2 * * * /usr/local/bin/docker_db_backup.sh
```

❏ 2. Côté NAS (La liaison SSH sécurisée)

Génération de la clé compatible Synology (sur le NAS) :

Bash

```
# Création de la clé RSA format PEM
openssl genrsa -out ~/.ssh/id_rsa_syno 4096
chmod 600 ~/.ssh/id_rsa_syno

# Extraction de la clé publique
ssh-keygen -y -f ~/.ssh/id_rsa_syno > ~/.ssh/id_rsa_syno.pub

# Envoi de la clé au VPS (mot de passe synobackup demandé)
cat ~/.ssh/id_rsa_syno.pub | ssh synobackup@79.137.14.202 "mkdir -p ~/.ssh && cat >>
~/.ssh/authorized_keys && chmod 700 ~/.ssh && chmod 600 ~/.ssh/authorized_keys"
```

Récupération de la clé privée pour l'interface DSM :

Bash

```
cat ~/.ssh/id_rsa_syno
```

(Copier le bloc `-----BEGIN RSA PRIVATE KEY-----` vers un fichier `backup.pem` sur ton PC).

❏ 3. Configuration Active Backup (Interface DSM)

Paramètre	Valeur
Service	Serveur de fichiers > rsync > rsync shell
Hôte / Port	<code>79.137.14.202</code> / <code>22</code>
Utilisateur	<code>synobackup</code>
Authentification	Clé privée (Charger <code>backup.pem</code>)
Dossier Source 1	<code>/backups/sql</code> (Les bases de données)
Dossier Source 2	<code>/home</code> (Les configs et Dockerfiles)
Planification	Quotidien à 04:00

Paramètre	Valeur
Rétention	Multi-version (7 jours minimum)

☐ Ce qui est maintenant "Automatique" :

1. **02h00** : Le VPS crée les fichiers SQL compressés.
2. **04h00** : Le NAS se connecte sans mot de passe, compare les fichiers et télécharge uniquement les nouveautés.
3. **Sécurité** : Si ton VPS brûle, tu as les bases ET les fichiers de config chez toi.

Est-ce que tu veux que je te génère aussi la commande pour tester la restauration d'une base sur ton VPS en cas de besoin ?

260323- Creation d'un environnement virtuel python

Nous venons de créer un **environnement virtuel Python** (souvent appelé `venv`).

C'est une pratique essentielle pour un cabinet comme **Altéris**, car elle permet d'isoler les outils de diagnostic et de traitement de données du reste du système d'exploitation.

1. Créer l'environnement virtuel

Exécutez ces commandes pour isoler nos outils géomatiques :

Bash

```
# Installation de python3-venv si nécessaire
sudo apt update && sudo apt install python3-venv python3-full -y

# Création de l'environnement dans un dossier 'scripts_alteris'
python3 -m venv ~/alteris_venv

# Activation de l'environnement
source ~/alteris_venv/bin/activate
```

Pourquoi est-ce important pour nous ?

- **Étanchéité technique** : En installant nos bibliothèques géomatiques (`geopandas`, `sqlalchemy`) à l'intérieur de `~/alteris_venv`, nous ne risquons pas de corrompre les paquets Python utilisés par Debian ou par nos conteneurs Docker.

- **Stabilité des analyses** : Cela garantit que nos scripts d'injection de données, comme celui pour l'OCS GE de La Ciotat, s'exécutent toujours avec les versions précises des dépendances que nous avons choisies.
- **Conformité Debian 12** : Nous respectons ainsi la nouvelle gestion stricte des paquets (PEP 668) que vous avez rencontrée lors de votre tentative précédente.

Prochaine étape opérationnelle

Maintenant que l'environnement est prêt et activé (votre terminal doit afficher `(alteris_venv)` au début de la ligne), vous pouvez lancer l'installation finale des outils de calcul :

Bash

```
pip install geopandas requests sqlalchemy psycopg2-binary pyogrio
```

Félicitations, votre environnement de calcul **Altéris** est désormais parfaitement opérationnel. L'installation s'est déroulée avec succès : les outils de manipulation spatiale (`geopandas`, `pyogrio`), les connecteurs de base de données (`sqlalchemy`, `psycopg2-binary`) et le moteur de requête (`requests`) sont prêts.

C'est une étape cruciale pour **maîtriser le réel** technique de votre VPS.

Prochaine étape : Intégration de l'OCS GE (La Ciotat)

Vous pouvez maintenant lancer l'injection des données d'artificialisation dans votre base `alteris_geo`.

1. **Vérifiez** que vous êtes toujours dans l'environnement actif `(alteris_venv)`.
2. **Lancez** le script que nous avons préparé :