# Installation d'un espace de Pré-Production pour GeoNature

# Dépendances nécessaires

• Se connecter à l'utilisateur geonat de bkp-srv :

```
ssh geonat@bkp-<region>-sinp
```

• Installer les dépendances (NVM, Python...) de GeoNature en suivant la documentation.

## Procédure rapatriement de la prod

• Se connecter à l'utilisateur geonat de bkp-srv :

```
ssh geonat@bkp-<region>-sinp
```

 Récupérer l'installation des outils GeoNature de web-srv (supprimer l'option --dry-run si tout est OK) :

```
rsync -e "ssh -p <port-ssh-web-srv>" -av --exclude .cache/ --exclude
venv/ --exclude venv.save/ geonat@web-<region>-sinp:/home/geonat/www/
/home/geonat/www/ --dry-run
```

• Récupérer les données de db-srv (supprimer l'option - - dry - run si tout est OK) :

```
rsync -av -e "ssh -p <port-ssh-db-srv>" --exclude "raw/*" geonat@db-
<region>-sinp:/home/geonat/data//home/geonat/data --dry-run
```

Recréer les liens symboliques de la racine du home de geonat :

```
ln -s www/atlas atlas
ln -s www/geonature geonature
ln -s www/taxhub taxhub
ln -s www/usershub usershub
```

## Procédure: installation via Git

- Cloner le dépôt avec l'URL en https: git clone https://github.com/PnX-SI/GeoNature.git geonature
- Se placer dans le dossier du dépôt cloné : cd geonature
- Récupérer le fichier config/settings.ini avec : scp geonat@web-<region>sinp:~/geonature/config/settings.ini ./config/
  - ∘ Adapter son contenu : vi config/settings.ini
- Récupérer le fichier config/geonature\_config.toml avec : scp geonat@web-

```
\frac{\text{appace.}}{2024/06/20} \text{ serveurs:installation:bkp-srv:pre-prod-geonature https://sinp-wiki.cbn-alpin.fr/serveurs/installation/bkp-srv/pre-prod-geonature?rev=1718873932
      <region>-sinp:~/geonature/config/geonature config.toml ./config/
          Adapter son contenu : vi config/geonature_config.toml
    • Réinstaller le venv du backend :
          ∘ cd install/
          ∘ ./01 install backend.sh

    Arrêter le script lors de la demande de mot de passe pour l'installation des scripts

                  SvstemD: CTRL+C
          Installer le module Sentry :
                source ../backend/venv/bin/activate
                pip install sentry sdk
                deactivate
```

- Copier les composants "custom" présents dans frontend/src/custom/components/
  - ∘ scp -r geonat@web-<region>sinp:~/geonature/frontend/src/custom/components/ frontend/src/custom/
  - Si besoin restaurer les fichiers .sample : git restore \*.sample
- Copier les images "custom" présentes dans frontend/src/custom/images/
  - o scp -r geonat@web-<region>sinp:~/geonature/frontend/src/custom/images frontend/src/custom/
- Copier le fichier favicon.ico :
  - ∘ scp -r geonat@web-<region>
    - sinp:~/geonature/frontend/src/favicon.ico frontend/src/
- Penser à reporter les éventuelles modifications du fichier

frontend/src/assets/i18n/fr.json:

- ∘ scp geonat@web-<region>sinp:~/geonature/frontend/src/assets/i18n/fr.json ./frontend/src/assets/i18n/
- Récupérer le fichier *environ* :
  - o scp geonat@web-<region>-sinp:~/geonature/environ ./
  - Adapter le fichier environ aux caractéristiques du serveur de la pré-prod : vi ./environ
- Recréer les liens symboliques présents dans frontend/src/external assets/ et dans external\_modules/
- Réinstaller les modules pre-existant :

```
source backend/venv/bin/activate
pip list | grep modules
pip install -e /home/geonat/www/modules/<module>
pip list | grep modules
```

- Essayer de réinstaller le frontend à l'aide du script :
  - ∘ cd install/
  - ∘ ./04 install fontend.sh
- Sinon, une alternative et la réinstallation manuelle du frontend :
  - Réinstaller Npm et Node à l'aide de Nvm :
    - cd frontend/
    - nvm use
  - Réinstaller les paquets Npm :
    - cd frontend/
    - npm install
  - Reconstruire le frontend à l'aide d'Angular :

- cd frontend/
- npm run build
- Corriger/Créer un fichier Logrotate :
  - ∘ vi /etc/logrotate.d/geonature
- Corriger/Créer le service SystemD

## **Reconfigurer GeoNature**

- Installer/Mettre à jour les dépendances de GeoNature.
- **NOTES**: il est nécessaire que Gunicorn écoute sur l'IP 0.0.0.0 car Nginx étant dans un container il ne peut accéder à l'IP 127.0.0.1. Le parefeu se charge de rendre inaccessibles le port 8000 de GeoNature ainsi que ceux d'UsersHub et TaxHub.
- Reconfigurer GeoNature :
  - Changer les paramètres suivants dans geonature config.toml :

```
vi ~/geonature/config/geonature config.toml
```

```
SQLALCHEMY DATABASE URI = "postgresql://geonatadmin:<mot-de-passe-
geonatadmin-sur-bkp-srv>@127.0.0.1:5432/geonature2db"
URL APPLICATION = "https://gnpp.silene.eu"
API ENDPOINT = "https://gnpp.silene.eu/api"
API TAXHUB = "https://thpp.silene.eu/api"
# Remplacer par une clé alétoire complexe
SECRET KEY = "<utiliser-uuid-pour-générer-une-clé>"
appName = "<nom-du-site> - Pré-Prod"
# Set Sentry DSN
SENTRY DSN = "<nouveau-sentry-DSN-généré-dans-app-monitor>"
[USERSHUB]
    URL USERSHUB = "https://uhpp.<sinp-domaine>"
[ACCOUNT_MANAGEMENT]
    VALIDATOR EMAIL = [
            "adminsys@<sinp-domaine>",
[FRONTEND]
    # Durée de vie du cache des stats de la page d'accueil
    # 86 400 = 1 jour
    STAT BLOC TTL = 86400
[PERMISSION MANAGEMENT]
    VALIDATOR_EMAIL = [
            "adminsys@<sinp-domaine>",
```

Changer les paramètres suivants dans settings.ini :

```
vi ~/geonature/config/settings.ini
```

```
#MODE=dev #Voir s'il faut utiliser git pour l'installation et donc
le mode "dev"...
my url=https://gnpp.silene.eu/
db host=127.0.0.1
user pg pass=<mot-de-passe-geonatadmin-sur-bkp-srv>
```

• Réinstaller le venv et Node pour GeoNature :

```
cd ~/geonature/frontend
nvm install
cd ~/geonature/backend
rm -fR venv
cd ~/geonature/install
./01 install backend.sh
```

• Réinstaller le frontend :

```
cd ~/geonature/install
./04 install frontend.sh
```

• Si erreur An unhandled exception occurred: Cannot find module '@angular-builders/custom-webpack/package.json:

```
cd ~/geonature/frontend/
nvm use
npm install
npm run build
```

• Créer une dossier pour les logs :

```
mkdir /var/log/geonature/
mv /var/log/usershub.log /var/log/geonature/
chown -R geonat: /var/log/geonature/
```

• Créer un fichier logrotate :

```
vi /etc/logrotate.d/geonature
```

```
/var/log/geonature/geonature.log {
    su geonat geonat
    daily
    rotate 8
    size 100M
    create
    compress
    postrotate
    systemctl reload geonature || true
    endscript
```

}

• Surcharger le service SystemD :

```
systemctl edit geonature
```

```
[Unit]
Description=GeoNature - PreProd
After=docker.service
StartLimitIntervalSec=60min
StartLimitBurst=25

[Service]
Restart=on-failure
RestartSec=2min
StandardOutput=append:/var/log/geonature/geonature.log
```

• Créer un fichier environ :

GUNICORN TIMEOUT=300

```
vi ~/geonature/environ

GUNICORN_HOST=0.0.0.0
GUNICORN NUM WORKERS=9
```

## **Reconfigurer l'Atlas**

Changer les paramètres suivants dans config.py :

```
vi ~/atlas/atlas/configuration/config.py

database_connection = "postgresql://geonatatlas:<mot-de-passe-
utilisateur-geonatatlas>@127.0.0.1:5432/gnatlas"
```

• Changer les paramètres suivants dans settings.ini :

```
vi ~/www/atlas/atlas/configuration/settings.ini
```

SECRET KEY = '<utiliser-uuid-pour-générer-une-clé>'

NOM APPLICATION = "Silene Nature - Pré-Prod"

```
db_host=0.0.0.0
user_pg_pass=<mot-de-passe-utilisateur-geonatatlas>
owner_atlas_pass=<mot-de-passe-utilisateur-geonatadmin>
db_source_host=127.0.0.1
atlas_source_pass=<mot-de-passe-utilisateur-geonatadmin>
```

• Réinstaller le venv et Node pour GeoNature-Atlas :

```
cd ~/atlas/static/
```

```
nvm install
cd ~/atlas/
rm -fR venv
./install_app.sh
```

Créer une dossier pour les logs :

```
mkdir /var/log/geonature-atlas/
mv /var/log/usershub.log /var/log/geonature-atlas/
chown -R geonat: /var/log/geonature-atlas/
```

• Créer un fichier logrotate :

```
vi /etc/logrotate.d/geonature-atlas
```

```
/var/log/geonature-atlas/geonature-atlas.log {
    su geonat geonat
    daily
    rotate 8
    size 100M
    create
    compress
    postrotate
    systemctl reload geonature-atlas || true
    endscript
}
```

• Surcharger le service SystemD :

```
systemctl edit geonature-atlas
```

```
[Unit]
Description=Atlas - PreProd
After=docker.service
StartLimitIntervalSec=60min
StartLimitBurst=25
[Service]
Restart=on-failure
RestartSec=2min
StandardOutput=append:/var/log/geonature-atlas/geonature-atlas.log
```

• Créer un fichier environ :

```
vi ~/atlas/environ
```

```
GUNICORN HOST=0.0.0.0
GUNICORN_NUM_WORKERS=4
GUNICORN TIMEOUT=90
```

#### **Reconfigurer UsersHub**

• Changer les paramètres suivants dans config.py :

```
vi ~/usershub/config/config.py

SQLALCHEMY_DATABASE_URI = "postgresql://geonatadmin:<mot-de-passe-
geonatadmin-sur-bkp-srv>@127.0.0.1:5432/geonature2db"

URL_APPLICATION ='https://uhpp.silene.eu'
SECRET_KEY = '<utiliser-uuid-pour-générer-une-clé>'
```

• Changer les paramètres suivants dans settings.ini :

```
vi ~/usershub/config/settings.ini
```

```
db_host=0.0.0.0
user_pg_pass=<mot-de-passe-utilisateur-geonatadmin>
url_application=https://uhpp.silene.eu
```

Réinstaller le venv et Node pour UsersHub :

```
cd ~/usershub/app/static/
nvm install
cd ~/usershub/
rm -fR venv
./install_app.sh
```

• Créer une dossier pour les logs :

```
mkdir /var/log/usershub/
mv /var/log/usershub.log /var/log/usershub/
chown -R geonat: /var/log/usershub/
```

• Créer un fichier logrotate :

```
vi /etc/logrotate.d/usershub
```

```
/var/log/usershub/usershub.log {
    su geonat geonat
    daily
    rotate 8
    size 100M
    create
    compress
    postrotate
    systemctl reload usershub || true
    endscript
}
```

• Surcharger le service SystemD :

```
systemctl edit usershub.service
```

```
[Unit]
Description=UsersHub - PreProd
After=docker.service
StartLimitIntervalSec=60min
StartLimitBurst=25
[Service]
Restart=on-failure
RestartSec=2min
StandardOutput=append:/var/log/usershub/usershub.log
```

• Créer un fichier environ :

```
vi ~/usershub/environ
GUNICORN HOST=0.0.0.0
```

## **Reconfigurer TaxHub**

• Changer les paramètres suivants dans config.py :

```
vi ~/taxhub/apptax/config.py
DEBUG=True
SQLALCHEMY_DATABASE_URI = "postgresql://geonatadmin:<mot-de-passe-
geonatadmin-sur-bkp-srv>@127.0.0.1:5432/geonature2db"
SECRET KEY = '<utiliser-uuid-pour-générer-une-clé>'
```

• Changer les paramètres suivants dans settings.ini :

user pg pass=<mot-de-passe-utilisateur-geonatadmin>

```
vi ~/taxhub/settings.ini
db host=0.0.0.0
```

• Réinstaller le venv et Node pour TaxHub :

```
cd ~/taxhub/static/
nvm install
cd ~/taxhub/
rm -fR venv
./install app.sh
```

• Créer une dossier pour les logs :

```
mkdir /var/log/taxhub/
mv /var/log/usershub.log /var/log/taxhub/
```

```
chown -R geonat: /var/log/taxhub/
```

• Créer un fichier logrotate :

```
vi /etc/logrotate.d/taxhub
```

```
/var/log/taxhub/taxhub.log {
    su geonat geonat
    daily
    rotate 8
    size 100M
    create
    compress
    postrotate
    systemctl reload taxhub || true
    endscript
}
```

Surcharger le service SystemD :

```
systemctl edit taxhub
```

```
[Unit]
Description=TaxHub - PreProd
After=docker.service
StartLimitIntervalSec=60min
StartLimitBurst=25

[Service]
Restart=on-failure
RestartSec=2min
StandardOutput=append:/var/log/taxhub/taxhub.log
```

• Créer un fichier environ :

```
vi ~/taxhub/environ
```

```
GUNICORN_HOST=0.0.0.0
```

## Création du container Docker hébergeant Postgresql & Nginx

#### Activation de l'IPv6 sur Docker et la stack proxy

• Afin d'avoir un support complet d'IPv6 sur la pré-prod, il est nécessaire d'activer le support d'IPv6 par Docker.

#### Création de la stack preprod

Ajouter l'utilisateur geonat au groupe docker :

```
sudo usermod -aG docker geonat
```

• Se déconnecter et se reconnecter au compte geonat et vérifier la prise en compte du groupe avec:

id

• Tester Docker en tant qu'utilisateur geonat avec :

```
docker run hello-world
```

- Si le message d'erreur docker: Got permission denied while trying to connect to the Docker daemon socket apparait, redémarrer la machine pour activer la prise en compte de Docker.
- Créer le dossier qui hébergera la stack GeoNature Docker Compose :

```
mkdir -p ~/docker/preprod
```

- Se placer dedans et récupérer les fichiers disponible sur le dépôt Github sinp-paca-srv ou sur le dépôt Github sinp-aura-srv.
- Copier le fichier .env.sample en .env :

```
cp .env.sample .env
```

- Configurer le fichier .env
- Lancer la stack avec :

```
docker compose up -d
```

• Vérifier le bon fonctionnement avec Portainer.

#### Accéder au shell Psql depuis le container Postgres

- Assurer vous que le container preprod-postgres est bien dans l'état running (via Portainer ou en ligne de commande).
- Accéder à l'intérieur du container :

```
docker exec -it preprod-postgres bash
```

Accéder à Psql avec :

```
psql -U ${POSTGRES USER} ${POSTGRES DB}
```

• Dans le container, l'utilisateur \${POSTGRES USER} et la base \${POSTGRES DB} remplace l'utilisateur par défaut postgres. Il est donc nécessaire de se connecter avec ces arguments !

#### Installer le client Postgresql sur l'hôte

- Vérifier la présence du dépôt de paquets deb pour Postgresql dans : /etc/apt/sources.list.d/
- Vérifier la version de Postgresql utilisé par le contenair Docker Postgresql de la pré-prod.
  - Utiliser la même version pour l'installation du paquet postgresql-client. Ex. : apt install postgresql-client-15

# Remplacer la base par la dernière version sauvegardée

Se connecter en root sur bkp-srv :

```
ssh admin@bkp-<region>-sinp ; sudo -i
```

Monter le dépôt Borg de db-srv :

```
borg mount /home/backups/db-srv /tmp/repo
```

- Saisir la *passphrase* de *db-srv*.
- Extraire le dump de la base de GeoNature et le copier dans l'espace de restauration du container preprod-postgres-restore avec :
  - Pour geonature2db :

```
cp -r /tmp/repo/db-srv-<date-et-
heure>/root/.borgmatic/postgresql_databases/172.18.5.1/geonatu
re2db
/home/geonat/docker/preprod/postgres/restore/<date>_geonature2
db.custom
```

• Pour *gnatlas* :

```
cp -r /tmp/repo/db-srv-<date-et-
heure>/root/.borgmatic/postgresql_databases/172.18.5.1/gnatlas
/home/geonat/docker/preprod/postgres/restore/<date>_gnatlas.cu
stom
```

- **ATTENTION** : le nom du fichier du dump récupéré doit contenir le nom de la base de donnée *geonature2db* ou *gnatlas*. Cette information est récupéré par le script de restauration pour effectuer certaines tâches spécifiques.
- Donner les droits à geonat sur les dumps :

```
chown geonat:
/home/geonat/docker/preprod/postgres/restore/*.custom
```

Démonter le dépôt :

```
borg umount /tmp/repo
```

- Se connecter en geonat sur bkp-srv :
  - Se placer dans le dossier de la stack *preprod* :

```
cd ~/docker/preprod/
```

• Restauration de GeoNature :

Lancer la restauration de la base de donnée (ATTENTION. Ex. avec geonature2db : cela supprimera la base en place) :

```
docker compose run --rm preprod-postgres-restore
/restore/restore.sh -d "2023-06-14 geonature2db.custom"
```

- Restauration de **GeoNature Atlas** :
  - Lancer la restauration de la base de donnée (ATTENTION. Ex. avec gnatlas : cela supprimera la base en place):

```
docker compose run --rm preprod-postgres-restore
/restore/restore.sh -d "2023-06-23 gnatlas.custom"
```

- ATTENTION : La restauration précédente va bloquer sur le rafraîchissement de la VM "observations" pour une raison encore inconnue. Stopper la restauration avec CTRL+C. Les tables et les VMs sont malgré tout généré dans la base. Il suffit donc de rafraichir l'ensemble des VMs pour finaliser la restauration.
- Rafraîchir les VMs grâce au script gnatlas refresh all.sql:

```
docker compose run --rm preprod-postgres-restore psql -U
geonatadmin -e -d gnatlas -f /restore/gnatlas refresh all.sql
```

## Docker Compose, Services SystemD et Nginx Proxy

- Pour fonctionner la pré-prod nécessite d'avoir démarré dans l'ordre suivant :
  - 1. le serveur Postgresql preprod-postgres de la stack Docker Compose "preprod"
  - 2. le fonctionnement sur l'hôte des services SystemD de GeoNature :

```
systemctl status geonature
systemctl status usershub
systemctl status taxhub
systemctl status geonature-atlas
```

3. le serveur Nginx *preprod-nginx* de la stack Docker Compose "preprod"

# Installer le dépôt sinp-<region>-data

• Se connecter en geonat sur bkp-srv :

```
ssh geonat@bkp-<region>-sinp
```

- Deux solutions pour créer le dossier ~/data et son contenu :
  - Copier directement le dossier ~/data de production présent sur le serveur db-srv puis modifier les fichiers de config si nécessaire.
  - Cloner le dépôt dans un dossier ~/data/ ce qui implique ensuite de copier tous les fichiers de config. Ex.:

git clone --recursive https://github.com/cbn-alpin/sinp-auradata.git data

• Se placer dans le dossier du dépôt :

cd ~/data/

• Installer Pipx:

#### sudo apt install pipx

o Configurer les chemins d'accès :

pipx ensurepath

• Relancer le terminal pour prendre en compte les changement ou essayer :

source ~/.bashrc

• Vérifier que Pipx est fonctionnel :

pipx --version

• Installer Pipenv:

pipx install pipenv

• Vérifier que Pipenv est fonctionnel :

pipenv --version

- Installer les dépendances de import-parser avec : cd ~/data/import-parser/; pipenv install
- Installer les dépendances de gn2pg avec : cd ~/data/gn2pg/; pipenv install
  - Si nécessaire, supprimer le dossier ~/.gn2pg si ce n'est pas un lien ls -al ~/.gn2pg puis si c'est un dossier rm -fr ~/.gn2pg
  - Recréer un lien symbolique : ln -s ~/data/gn2pg/config ~/.gn2pg
- Copier TOUS les fichiers de config depuis le serveur db-srv

From

https://sinp-wiki.cbn-alpin.fr/ - CBNA SINP

Permanent link:

https://sinp-wiki.cbn-alpin.fr/serveurs/installation/bkp-srv/pre-prod-geonature?rev=1718873932

Last update: 2024/06/20 08:58

