

Créer un utilisateur avec accès par tunnel SSH en lecture seule aux bases GeoNature

Principe : les bases de données de GeoNature sont accessibles uniquement en local. Il n'y a pas d'ouverture du port 5432 sur l'extérieur. Ainsi pour se connecter à la base de données Postgresql, il faut être "présent" localement sur le serveur ou sur une machine du réseau privé 10.0.1.x. Il est donc nécessaire de se connecter à Postgresql via un tunnel SSH aboutissant sur l'instance "db-srv" où là la connexion pourra se faire sur l'hôte localhost et le port 5432.

Ressources :

- [How to create a SSH user that can only connect to MySQL / PostgreSQL on Ubuntu with Username and Password](#)

Création d'un utilisateur "dbreader" sans "home"

L'utilisateur système permettant de créer le tunnel SSH sera nommé "dbreader". Il n'aura pas de dossier *home* et aucun shell actif. Pour cela suivre les étapes suivantes :

- Création de l'utilisateur *dbreader* sur l'instance "db-srv" : `useradd --no-create-home -s /usr/sbin/nologin dbreader`
 - Créer un utilisateur sans possibilité de se loguer à un shell -s /usr/sbin/nologin n'empêche pas la connexion à la base Postgresql
- Ajouter un mot de passe à l'utilisateur : `passwd dbreader`
- Modifier le fichier de config du serveur SSH pour permettre un accès par mot de passe uniquement pour cet utilisateur : `vi /etc/ssh/sshd_config`
 - Ajouter à la fin du fichier les lignes suivantes (**il est important que ces lignes soient bien complètement à la fin du fichier**) :

```
Match User dbreader
    PasswordAuthentication yes
```

- À la place de l'utilisation d'un mot de passe, il est aussi possible d'utiliser les clés SSH publiques des personnes autorisées en les plaçant dans le fichier `/etc/ssh/authorized_keys_dbreader`. Le code à ajouter à la fin du fichier sera alors :

```
Match User dbreader
    AuthorizedKeysFile /etc/ssh/authorized_keys_%u
```

- Vous pouvez copier/coller le fichier `authorized_keys` de l'utilisateur *admin* comme base de départ (et le modifier au besoin) : `cp /home/admin/.ssh/authorized_keys /etc/ssh/authorized_keys_dbreader`
- Il est nécessaire de définir les droits sur le fichier `/etc/ssh/authorized_keys_dbreader` ainsi : `chmod 600 /etc/ssh/authorized_keys_dbreader ; chown dbreader: /etc/ssh/authorized_keys_dbreader`
- Notez qu'il est aussi possible de combiner les deux possibilités (par mot de passe et clé SSH) ainsi :

```
Match User dbreader
  PasswordAuthentication yes
  AuthorizedKeysFile /etc/ssh/authorized_keys_%u
  Banner none
```

- Pour éviter l'affichage de la bannière de connexion au serveur : Banner none
- Redémarrer le serveur *Sshd* : systemctl restart sshd
- Tester la connexion au serveur : ssh dbreader@db-<region>-sinp
 - Cela devrait afficher :

```
Could not chdir to home directory /home/dbreader: No such file or directory
This account is currently not available.
Connection to db-paca-sinp closed.
```

Création d'un utilisateur en lecture seule pour Postgresql

Création de l'utilisateur et définition des droits

- Se connecter à la base avec un compte superadmin : psql -h "localhost" -U "admin" -d "geonature2db"
- Exécuter les requêtes suivantes :

```
-- Créer l'utilisateur "gnreader"
CREATE USER gnreader WITH ENCRYPTED PASSWORD '<mot-de-passe>' ;

-- Donner le droit de se connecter aux bases
GRANT CONNECT ON DATABASE geonature2db TO gnreader ;
GRANT CONNECT ON DATABASE gnatlas TO gnreader ;

-- Autoriser l'utilisation de tous les schémas de la base :
-- 1. Générer la requête à exécuter
SELECT 'GRANT USAGE ON SCHEMA ' || string_agg(nspsname, ', ') || ' TO
gnreader ;' FROM pg_namespace ;

-- 2. Exécuter la requête obtenue précédemment
GRANT USAGE ON SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, gn_commons, gn_exports, gn_imports, gn_meta,
gn_monitoring, gn_permissions, gn_sensitivity, gn_synthese, ref_geo,
ref_habitats, ref_nomenclatures, taxonomie, utilisateurs
TO gnreader ;

-- Autoriser l'utilisateur à faire des sélections sur toutes les tables
de tous les schémas (même principe que ci-dessus)
-- 1. Générer la requête à exécuter
SELECT 'GRANT SELECT ON ALL TABLES IN SCHEMA ' || string_agg(nspsname,
```

```
' , ' ) || ' TO gnreader ;' FROM pg_namespace ;

-- 2. Exécuter la requête obtenue précédemment
GRANT SELECT ON ALL TABLES IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, gn_commons, gn_exports, gn_imports, gn_meta,
gn_monitoring, gn_permissions, gn_sensitivity, gn_synthese, ref_geo,
ref_habitats, ref_nomenclatures, taxonomie, utilisateurs
TO gnreader ;

-- Autoriser l'utilisateur à faire des sélection sur toutes les
sequences (id)
-- Réutiliser la requête précédente et remplacer "TABLES" par
"SEQUENCES" :
GRANT SELECT ON ALL SEQUENCES IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, gn_commons, gn_exports, gn_imports, gn_meta,
gn_monitoring, gn_permissions, gn_sensitivity, gn_synthese, ref_geo,
ref_habitats, ref_nomenclatures, taxonomie, utilisateurs
TO gnreader ;

-- Ajouter l'accès en lecture sur les futures tables :
-- Réutiliser la requête précédente et remplacer la première et la
dernière ligne :
ALTER DEFAULT PRIVILEGES FOR USER geonatadmin IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, gn_commons, gn_exports, gn_imports, gn_meta,
gn_monitoring, gn_permissions, gn_sensitivity, gn_synthese, ref_geo,
ref_habitats, ref_nomenclatures, taxonomie, utilisateurs
GRANT SELECT ON TABLES TO gnreader ;

-- Ajouter l'accès en lecture sur les futures tables :
-- Réutiliser la requête précédente et remplacer la dernière ligne :
ALTER DEFAULT PRIVILEGES FOR USER geonatadmin IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, gn_commons, gn_exports, gn_imports, gn_meta,
gn_monitoring, gn_permissions, gn_sensitivity, gn_synthese, ref_geo,
ref_habitats, ref_nomenclatures, taxonomie, utilisateurs
GRANT SELECT ON SEQUENCES TO gnreader ;
```

Modification des autorisations d'accès au serveur Postgresql

- Modifier le fichier `pg_hba.conf`: `vi /etc/postgresql/12/main/pg_hba.conf`
 - Ajouter le contenu suivant :

```
# GeoNature : access by gnreader (read only)
host    geonature2db    gnreader      10.0.1.20/32
md5
host    gnatlas        gnreader      10.0.1.20/32
```

md5

- Recharger la configuration Postgresql : `systemctl reload postgresql`

Configuration de l'accès avec DBeaver

- Tester la connexion en lecture seule depuis Dbeaver en créant une nouvelle connexion avec ces paramètres :
 - Onglet Général :
 - Host : localhost (tester aussi 10.0.1.20 en cas de problème)
 - Port : 5432
 - Database : geonature2db
 - Authentification : Database Native
 - Nom d'utilisateur : gnreader
 - Mot de passe : <gnreader-password>
 - Cocher "Save password locally"
 - Driver name : PostgreSQL
 - Onglet Postgresql :
 - Cocher "Show all databases"
 - Laisser les autres champs avec les valeurs par défaut.
 - Onglet SSH :
 - Cocher "Utiliser le tunnel SSH"
 - Hôte/IP : <ip-db-srv>
 - Port : <port-ssh-db-srv>
 - Nom d'utilisateur : dbreader
 - Pour une connexion via le mot de passe :
 - Méthode d'authentification : "Mot de passe"
 - Mot de passe : <dbeader-unix-password>
 - Cocher "Enregistrer le mot de passe"
 - Pour une connexion via une clé SSH :
 - Méthode d'authentification : "Clef publique"
 - Clé privée : indiquer le chemin vers votre clé SSH privée (Ex. : /home/\${USER}/.ssh/id_rsa)
 - Phrase passe : indiquer votre "passe phrase" associé à votre clé SSH ou laisser vide si vous n'en avez pas associé.
 - Cocher "Enregistrer le mot de passe"
- Cliquer en bas à gauche sur "Test de la connexion..."

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

Permanent link:
<https://wiki-sinp.cbn-alpin.fr/serveurs/installation/db-srv/postgresql-ssh-tunnel?rev=1632907513>

Last update: **2021/09/29 09:25**

