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 autorisés 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/collet 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.
```

- Pour supprimer l'affichage de LinuxLogo lors de la connexion, éditer le fichier : `vi /etc/pam.d/sshd`
  - Descendez jusqu'à trouver les lignes contenant `pam_motd.so` . Il y en a généralement deux.
  - Ajoutez la ligne : `session [success=2 default=ignore] pam_succeed_if.so user = dbreader` juste au-dessus.
  - Le bloc devrait ressembler exactement à ça :

```
session [success=2 default=ignore] pam_succeed_if.so user =
dbreader
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so  noupdate
```

  - Si vous n'avez qu'une seule ligne `pam_motd.so` dans votre fichier, remplacez `success=2` par `success=1`
  - Après l'enregistrement de la modification, la bannière LinuxLogo disparaîtra pour cet utilisateur.

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" ''
- Pour attribuer les droits de lecture seule à un utilisateur, si les droits ont été révoqué sur le schéma `public` des bases de données, il suffit depuis Postgresql 14 d'attribuer l'utilisateur lecteur au rôle par défaut `pg_read_all_data` et d'autoriser sa connexion à la base. 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 ;

-- Associer l'utilisateur gnreader au rôle par défaut pg_read_all_data
GRANT pg_read_all_data TO gnreader;
```

- Pour révoquer les droits d'accès en lecture seule à une base de données :

```
REVOKE CONNECT ON DATABASE <nom-base> FROM <nom-utilisateur-lecteur>;
```

Base "geonature2db" (avant Postgresql 14)

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

```
-- Autoriser l'utilisation de tous les schémas de la base :
-- 1. Générer la requête à exécuter
SELECT 'GRANT USAGE ON SCHEMA ' || string_agg(nspname, ', ') || ' 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 sélection 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(nspname, ', ') || ' 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 gnreader 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 sequences :
-- Réutiliser la requête précédente et remplacer la dernière ligne :
ALTER DEFAULT PRIVILEGES FOR USER gnreader 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 ;
```

Base "gnatlas" (avant Postgresql 14)

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

```
-- Autoriser l'utilisation de tous les schémas de la base :
-- 1. Générer la requête à exécuter
SELECT 'GRANT USAGE ON SCHEMA ' || string_agg(nspname, ', ') || ' 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, taxonomie, synthese, ref_geo, atlas, pg_temp_22,
pg_toast_temp_22, pg_temp_24, pg_toast_temp_24, pg_temp_45,
pg_toast_temp_45, pg_temp_37, pg_toast_temp_37
    TO gnreader ;

-- Autoriser l'utilisateur à faire des sélection 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(nspname,
', ') || ' 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, taxonomie, synthese, ref_geo, atlas, pg_temp_22,
pg_toast_temp_22, pg_temp_24, pg_toast_temp_24, pg_temp_45,
pg_toast_temp_45, pg_temp_37, pg_toast_temp_37
    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, taxonomie, synthese, ref_geo, atlas, pg_temp_22,
pg_toast_temp_22, pg_temp_24, pg_toast_temp_24, pg_temp_45,
pg_toast_temp_45, pg_temp_37, pg_toast_temp_37
    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 gnreader IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, taxonomie, synthese, ref_geo, atlas, pg_temp_22,
pg_toast_temp_22, pg_temp_24, pg_toast_temp_24, pg_temp_45,
pg_toast_temp_45, pg_temp_37, pg_toast_temp_37
    GRANT SELECT ON TABLES TO gnreader ;

-- Ajouter l'accès en lecture sur les futures sequences :
-- Réutiliser la requête précédente et remplacer la dernière ligne :
ALTER DEFAULT PRIVILEGES FOR USER gnreader IN SCHEMA
pg_toast, pg_temp_1, pg_toast_temp_1, pg_catalog, public,
information_schema, taxonomie, synthese, ref_geo, atlas, pg_temp_22,
pg_toast_temp_22, pg_temp_24, pg_toast_temp_24, pg_temp_45,
pg_toast_temp_45, pg_temp_37, pg_toast_temp_37
    GRANT SELECT ON SEQUENCES TO gnreader ;

-- Ajouter l'accès aux "foreign data tables" :
GRANT USAGE ON FOREIGN SERVER geonaturedbserver TO gnreader ;

GRANT USAGE ON FOREIGN DATA WRAPPER postgres_fdw TO gnreader ;

CREATE USER MAPPING FOR gnreader SERVER geonaturedbserver OPTIONS (USER
'gnreader', password '<mot-de-passe-de-gnreader>');
```

Ajout d'un espace permetant la création de table/VM pour gnreader

gnreader est un utilisateur en lecture seule sur les schémas de GeoNature mais nous lui créons un schéma où il aura un accès en écriture. Cela permettra la création de table ou VM pour des requêtes d'extraction sur les tables de GeoNature.

```
-- Création du schéma "playground"
CREATE SCHEMA playground AUTHORIZATION geonatadmin;

-- Ajout des droits d'écriture sur schéma à l'utilisateur gnreader
GRANT USAGE, CREATE ON SCHEMA playground 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
scram-sha-256
host    gnatlas         gnreader        10.0.1.20/32
scram-sha-256
```

- Recharger la configuration *Postgresql* : '' systemctl reload postgresql ''

Configuration de l'accès avec DBeaver

Vous pouvez configurer votre accès avec DBeaver comme indiqué sur la page spécifique à cet outil.

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

Permanent link:
https://wiki-sinp.cbn-alpin.fr/serveurs/installation/db-srv/postgresql-ssh-tunnel

Last update: **2026/09/16 16:28**