

## POSTGRESQL

### 1. Se connecter au réseau de l'IUT

#### a. Connexion directe

Sur un ordinateur de l'IUT, il suffit de se connecter d'ouvrir une session Linux ou Windows. De votre ordinateur personnel, dans les locaux de l'IUT, connectez-vous au wifi Unice Hotspot.

#### b. Connexion par VPN

Si ce n'est pas possible, vous devez vous connecter au VPN de l'IUT.

Voici les informations de connexions :

Type : L2TP/IPSec PSK

Adresse du serveur : 134.59.20.161

Clé pré-partagée : KIGR#2456%

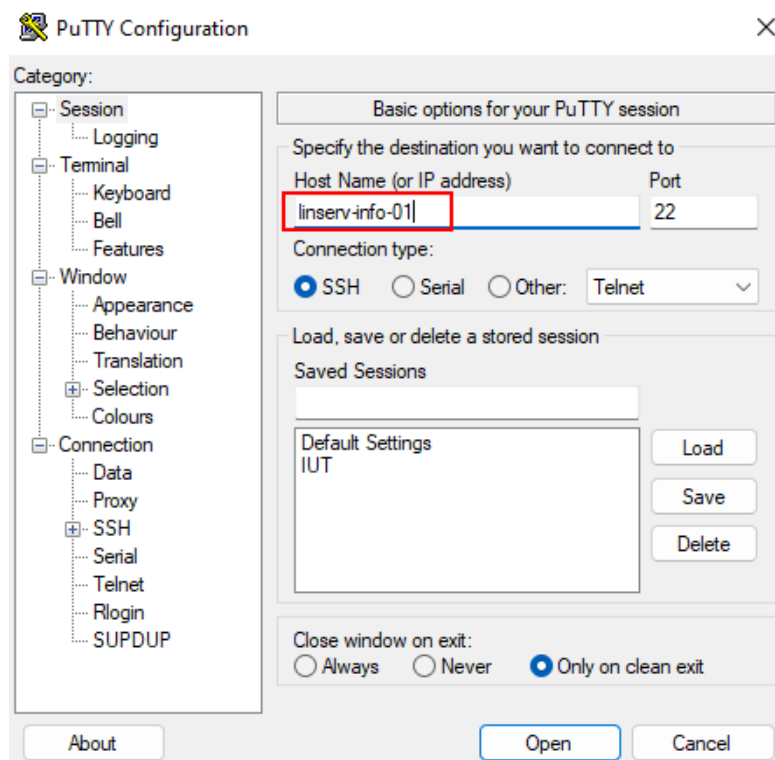
Le compte : CAMPUS\identifiant

Le mot de passe associé au compte

### 2. Se connecter à PostgreSQL

#### a. Connexion en ligne de commande

Connexion en SSH à linserv-info-01



Sous linux  
ssh linserv-info-01

Sous Windows  
Ouvrir putty puis remplir el  
champ Host name avec  
linserv-info-01 puis Open.

[En savoir plus](#)

La connexion demande ensuite votre login et pass Linux :

```
login as: tvert
tvert@LINSERV-INFO-01.campus.unice.fr's password:
Linux LINSERV-INFO-01 6.1.0-1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.4-1 (2023-01-07) x86_64

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Une mise à jour de micrologiciel est disponible pour l'appareil.
Exécutez «fwupdmgr get-upgrades» pour plus d'informations.

Last login: Mon Feb 20 21:06:54 2023 from 134.59.20.161
tvert@campus.unice.fr@LINSERV-INFO-01:~$
```

Taper `psql --dbname="ab123456" --username="ab123456"` (si votre login est ab123456) dans l'invite de commandes puis entrer votre mot de passe, ce qui permet de vous connecter au serveur PostgreSQL par défaut à votre database.

```
tvert@campus.unice.fr@LINSERV-INFO-01:~$ psql --dbname="tvert" --username="tvert"
"
Mot de passe pour l'utilisateur tvert :
psql (15.5 (Debian 15.5-0+deb12u1))
Saisissez « help » pour l'aide.

tvert=> \d
```

L'invite de commande est maintenant => car vous êtes connecté au shell postgresQL.

A la 1ere connexion, vous devrez créer la base de travail sur le serveur PostgreSQL. Cette base s'appellera ab123456\_r606 pour cela taper les commandes suivantes :

```
CREATE DATABASE ab123456_r606 ;
\c ab123456_r606
```

Vous pouvez maintenant travailler dans la base ab123456\_r606 afin de créer des tables et utiliser les requêtes CRUD SQL habituelles.

```
tvert=> CREATE DATABASE tvert_R606
tvert-> ;
CREATE DATABASE
tvert=> \c tvert_r606
Vous êtes maintenant connecté à la base de données « tvert_r606 » en tant qu'utilisateur « tvert ».
tvert_r606=> \d
Aucune relation n'a été trouvée.
tvert_r606=>
```

Lors de connexion ultérieure (autre que la 1ere), vous pourrez directement vous connecter à la base ab123456\_r606 en tapant :

```
psql --dbname="ab123456_r606" -username="ab123456"
```

## b. Connexion par un programme en Python

```
import psycopg2

conn = psycopg2.connect(host="linserv-info-01",dbname="xxxxxxxxxxx@campus.unice.fr",\
                        user="xxxxxxxxxxx@campus.unice.fr",password="xxxxxxxxxxx")
cur = conn.cursor()

requete = """CREATE TABLE backlog (id VARCHAR(32) PRIMARY KEY,name VARCHAR(32)); """
cur.execute(requete)
conn.commit()

cur.close()
conn.close()
```

La commande `psycopg2.connect` permet de se connecter au serveur PostgreSQL sur la machine `linserv-info-01`. SI la connexion est réussie, la fonction retourne un objet connexion qui permettra de créer un objet curseur afin de dialoguer avec le serveur PostgreSQL.

Plus de renseignement sur le wikipedia de l'IUT :

## PostgreSQL au département

Créé par Benoît Benedetti, dernière modification le 16 janv. 2024

Un serveur PostgreSQL est installé au département sur [LINSERV-INFO-01](#).

Vous pouvez interagir avec le serveur PostgreSQL avec l'outil en ligne de commande **psql** une fois connecté en ssh sur LINSERV-INFO-01.

Il est aussi accessible depuis:

- Tous les postes physiques Windows/Linux/Macs du département à Fabron et Sophia.
- Depuis le Wifi à Fabron et Sophia
- Depuis l'extérieur une fois monté le VPN Pedago

Pour vous connecter, utilisez votre identifiant ENT UCA comme utilisateur et comme mot de passe. Par exemple, si votre identifiant est **ab123456**, utilisez **ab123456** comme mot de passe pour vous connecter au serveur PostgreSQL.

Une base de données avec comme nom votre identifiant existe par défaut: ne la supprimez pas!

Votre utilisateur peut créer autant de bases qu'il veut, mais vous êtes sur un serveur partagé: si vous deviez créer une base de données **tp02**, peut-être qu'un autre étudiant l'aurait déjà créée, et vous auriez une erreur si vous aviez à créer une base de données avec ce même nom.

Vous êtes donc invités à préfixer le nom de vos bases de données par votre login afin mieux maîtriser le nommage des bases: par exemple, créez la base **ab123456\_tp02**, au lieu de **tp02**.

### 3. Commandes PostgreSQL en ligne de commande

Postgres possède des commandes spécifiques accessibles démarrant par \ suivi de la commande. Par exemple, voici quelques commandes utiles :

- \? : aide sur les commandes psql
- \h : aide sur les commandes sql
- \q : quitter
- \c : seconnecter à une autre base
- \d : liste des tables
- \dt NOMTABLE : liste le schéma d'une table
- \dn : liste des schémas
- \df : liste des fonctions
- \i FICHIER exécute les commandes SQL de fichier

```
tvert@campus.unice.fr=> \d
Liste des relations

```

Schéma	Nom	Type	Propriétaire
public	astreinte	table	tvert@campus.unice.fr
public	competence	table	tvert@campus.unice.fr
public	competence2	table	tvert@campus.unice.fr
public	entreprises	table	tvert@campus.unice.fr
public	etudiants	table	tvert@campus.unice.fr
public	informaticien	table	tvert@campus.unice.fr
public	stages	table	tvert@campus.unice.fr

Le shell postgresQL reconnaît également les commandes SQL en direct.

Par exemple :

```
CREATE TABLE essai (id int, nom varchar(32));
INSERT INTO essai VALUES (1,'essai PostgreSQL');
SELECT * FROM essai ;
```

Les commandes SQL peuvent être entrées sur plusieurs lignes dans le shell postgres mais elles doivent se terminer par un ; pour l'exécution. L'invite de commande se transforme en → sur plusieurs lignes. Par exemple :

```
INSERT INTO essai
VALUES (1,'essai PostgreSQL');
```

```
tvert@campus.unice.fr=> CREATE TABLE essai (id int, nom varchar(32));
CREATE TABLE
tvert@campus.unice.fr=> INSERT INTO essai
tvert@campus.unice.fr-> VALUES
tvert@campus.unice.fr-> (1,'essai PostgreSQL');
INSERT 0 1
tvert@campus.unice.fr=> SELECT * FROM essai;
 id |      nom
----+-----
  1 | essai PostgreSQL
(1 ligne)
```

Remarque : les commandes spécifiques à Postgres démarrent par \ mais ne

nécessitent pas de ; final.

Taper \q pour revenir sur le serveur linux puis exit pour fermer la console.

```
tvert@campus.unice.fr=> \q  
tvert@campus.unice.fr@LINSERV-INFO-01:~$ exit
```

#### **4. Commandes PostgreSQL en Python**

Les objets connection et cursor sont expliqués dans la documentation en ligne du module psycopg. Les requêtes et réponses passent toutes par ces deux objets.

Exemple :

```
import psycopg2  
  
conn = psycopg2.connect(host="linserv-info-01", dbname="xxxxxxxxxxxx@campus.unice.fr", \  
| | | | | user="xxxxxxxxxxxx@campus.unice.fr", password="xxxxxxxxxxxx")  
cur = conn.cursor()  
  
requete = """CREATE TABLE backlog (id VARCHAR(32) PRIMARY KEY, name VARCHAR(32)); """  
cur.execute(requete)  
conn.commit()  
  
cur.close()  
conn.close()
```

[En savoir plus](#)