

## Postgresql-Aufgabe

### Inhaltsverzeichnis

Installieren Sie einen postgresql-Server sowohl unter Windows als auch unter Linux:.....	1
Windows: .....	1
Linux:.....	1
Erzeugen sie spezielle User unter Postgresql:.....	2
SuperUser: .....	2
User der Datenbanken anlegen, befüllen und auslesen kann .....	2
Benutzer erstellen: .....	2
Dem Benutzer Rechte geben: .....	3
User der nur in einer Datenbank Daten auslesen kann:.....	3
Datenbank erstellen: .....	4
Versuchen Sie unter Windows mit dem psql-Kommando auf Daten unter Linux zuzugreifen:.....	4
Linux einstellen: .....	4
Windows einstellen: .....	5
Ergebnis: .....	7
Von Windows auf Linux: .....	7
Von Linux auf Windows: .....	7

Installieren Sie einen postgresql-Server sowohl unter Windows als auch unter Linux:

Windows:

Im Windows mittels einem Installer PostgreSQL installiert und durchgeklickt

Der Link zum Installer finden sie hier: [Installer](#)

Linux:

```
sudo sh -c 'echo "deb
https://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list'
```

```
wget --quiet -O -  
https://www.postgresql.org/media/keys/ACCC4CF8.asc |  
sudo apt-key add -  
sudo apt-get update  
sudo apt-get -y install postgresql
```

Erzeugen sie spezielle User unter Postgresql:

SuperUser:

Mit dem Befehl `CREATE USER super_user SUPERUSER;` wird der superuser erstellt

```
postgres=# CREATE USER super_user SUPERUSER;  
CREATE ROLE  
postgres=# |
```

Abbildung 1 - Erstellung eines Super\_users in der Windows Terminal (WT)

Mit `\du` werden alle User angezeigt

```
postgres=# \du  
  
               Liste der Rollen  
Rolllenname | Attribute  
-----+-----  
postgres   | Superuser, Rolle erzeugen, DB erzeugen, Replikation, Bypass RLS  
super_user | Superuser  
  
postgres=# |
```

Abbildung 2 - Anzeige aller Benutzer

User der Datenbanken anlegen, befüllen und auslesen kann

Benutzer erstellen:

Einen Benutzer erstellen mit dem Passwort „password“ erstellen (Ein Passwort wird benötigt damit sich der Benutzer überhaupt anmelden kann)

```
postgres=# CREATE ROLE db_user WITH LOGIN PASSWORD 'password';
CREATE ROLE
postgres=# \du
```

Liste der Rollen	
Rollenname	Attribute
db_user	
postgres	Superuser, Rolle erzeugen, DB erzeugen, Replikation, Bypass RLS
super_user	Superuser

```
postgres=# |
```

Abbildung 3 - Passwort für Benutzer festlegen

Dem Benutzer Rechte geben:

Nachdem der Benutzer erstellt ist, kann man ihm mit dem **ALTER** Befehl das Recht geben, dass er Datenbanken erstellen kann

```
postgres=# ALTER ROLE db_user CREATEDB;
ALTER ROLE
postgres=# |
```

Abbildung 4 - Rechtevergabe um eine Datenbank zu erstellen

Berechtigungen zum Ausführen von **SELECT** auf allen Tabellen in zukünftigen Datenbanken

```
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO db_user;
ALTER DEFAULT PRIVILEGES
postgres=# |
```

Abbildung 5 - Rechtevergabe um den Select Befehl überall ausführen zu können

User der nur in einer Datenbank Daten auslesen kann:

```
postgres=# CREATE ROLE only_read_user WITH LOGIN PASSWORD 'password';
CREATE ROLE
postgres=# \du
```

Liste der Rollen	
Rollenname	Attribute
db_user	DB erzeugen
only_read_user	
postgres	Superuser, Rolle erzeugen, DB erzeugen, Replikation, Bypass RLS
super_user	Superuser

```
postgres=# |
```

Abbildung 6 - Benutzer erstellen der nur Leserechte hat

Datenbank erstellen:

Mit **CREATE DATABASE my\_database;**

```
postgres=# CREATE DATABASE my_database;  
CREATE DATABASE
```

Abbildung 7 - Datenbank erstellen

```
postgres=# GRANT CONNECT ON DATABASE my_database TO only_read_user
```

Abbildung 8 - Rechtevergabe um sich auf die Datenbank verbinden zu dürfen

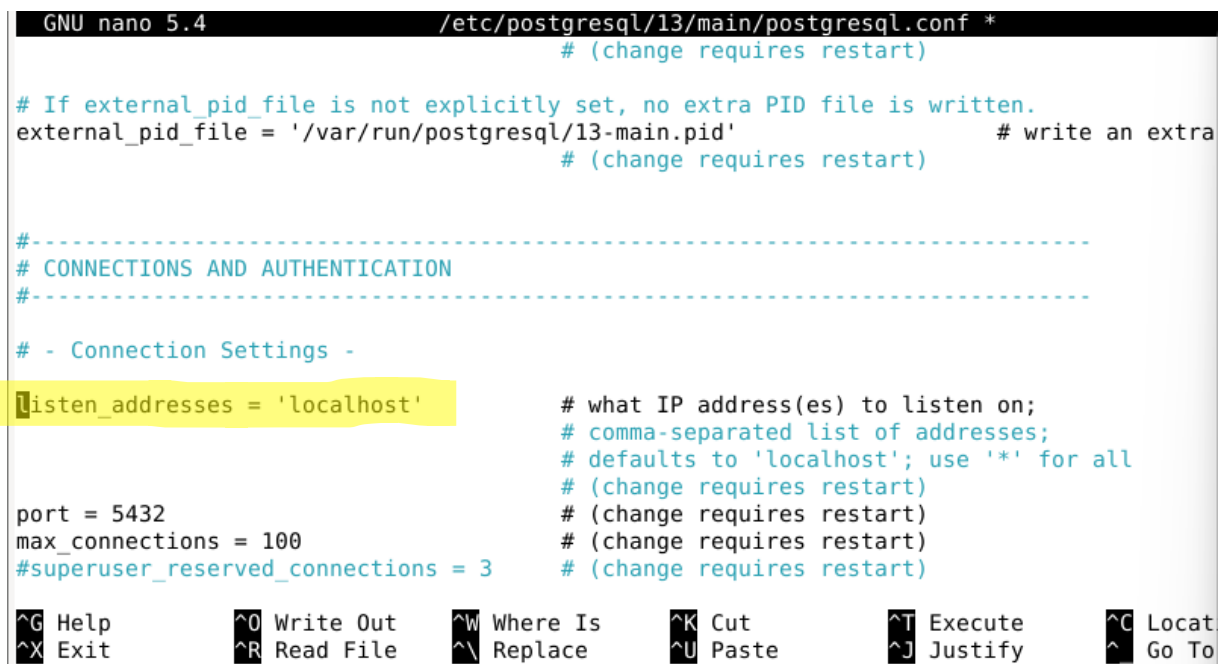
```
postgres=# GRANT USAGE, SELECT ON ALL TABLES IN SCHEMA public TO only_read_user;  
GRANT  
postgres=# |
```

Abbildung 9 - Rechtevergabe um auf alle Tabellen zu selecten

Versuchen Sie unter Windows mit dem psql-Kommando auf Daten unter Linux zuzugreifen:

Linux einstellen:

Mit dem Befehl kommt man ins config-File: „**nano /etc/postgresql/13/main/postgresql.conf**“, wo man dann den Kommentar (#) bei **listen\_addresses = 'localhost'** entfernen muss.



```
GNU nano 5.4 /etc/postgresql/13/main/postgresql.conf *  
# (change requires restart)  
  
# If external_pid_file is not explicitly set, no extra PID file is written.  
external_pid_file = '/var/run/postgresql/13-main.pid' # write an extra  
# (change requires restart)  
  
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
listen_addresses = 'localhost' # what IP address(es) to listen on;  
# comma-separated list of addresses;  
# defaults to 'localhost'; use '*' for all  
# (change requires restart)  
port = 5432 # (change requires restart)  
max_connections = 100 # (change requires restart)  
#superuser_reserved_connections = 3 # (change requires restart)  
  
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Locat  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To
```

Abbildung 10 - Die Einstellungen wo man den Wert für listen\_addresses ändern muss

Danach das 'localhost' mit einem Stern austauschen, um alle Adressen zu erlauben.

Danach in `/etc/postgresql/13/main/pg_hba.conf` die Zeile

<b>host</b>	<b>all</b>	<b>all</b>	<b>127.0.0.1/32</b>	<b>md5</b>
-------------	------------	------------	---------------------	------------

eine neue Zeile darunter hinzufügen:

<b>host</b>	<b>all</b>	<b>all</b>	<b>0.0.0.0/0</b>	<b>md5</b>
-------------	------------	------------	------------------	------------

```

File Edit View Search Terminal Help
GNU nano 5.4 /etc/postgresql/13/main/pg_hba.conf
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres peer
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 md5
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
  
```

Abbildung 11 - Einstellungen wo eine extra Zeile hinzugefügt werden muss

Danach Postgresql neustarten: `service postgresql restart`

Windows einstellen:

Unter "`C:\Program Files\PostgreSQL\16rc1\data\postgresql.conf`"

Die Zeile muss hinzugefügt werden:

<b>host</b>	<b>all</b>	<b>all</b>	<b>0.0.0.0/0</b>	<b>md5</b>
-------------	------------	------------	------------------	------------

```

# IPv6 local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all scram-sha-256
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
host all all 0.0.0.0/0 md5

```

Abbildung 12 - Im Windows Einstellungsfile muss eine extra Zeile am Ende hinzugefügt werden

Auf der Windows Firewall alle PostgreSQL Verbindungen erlauben:

- Eine neue Firewall Regel erstellen
- Port auswählen
- TCP ➔ Bestimmte lokale Ports: 5432

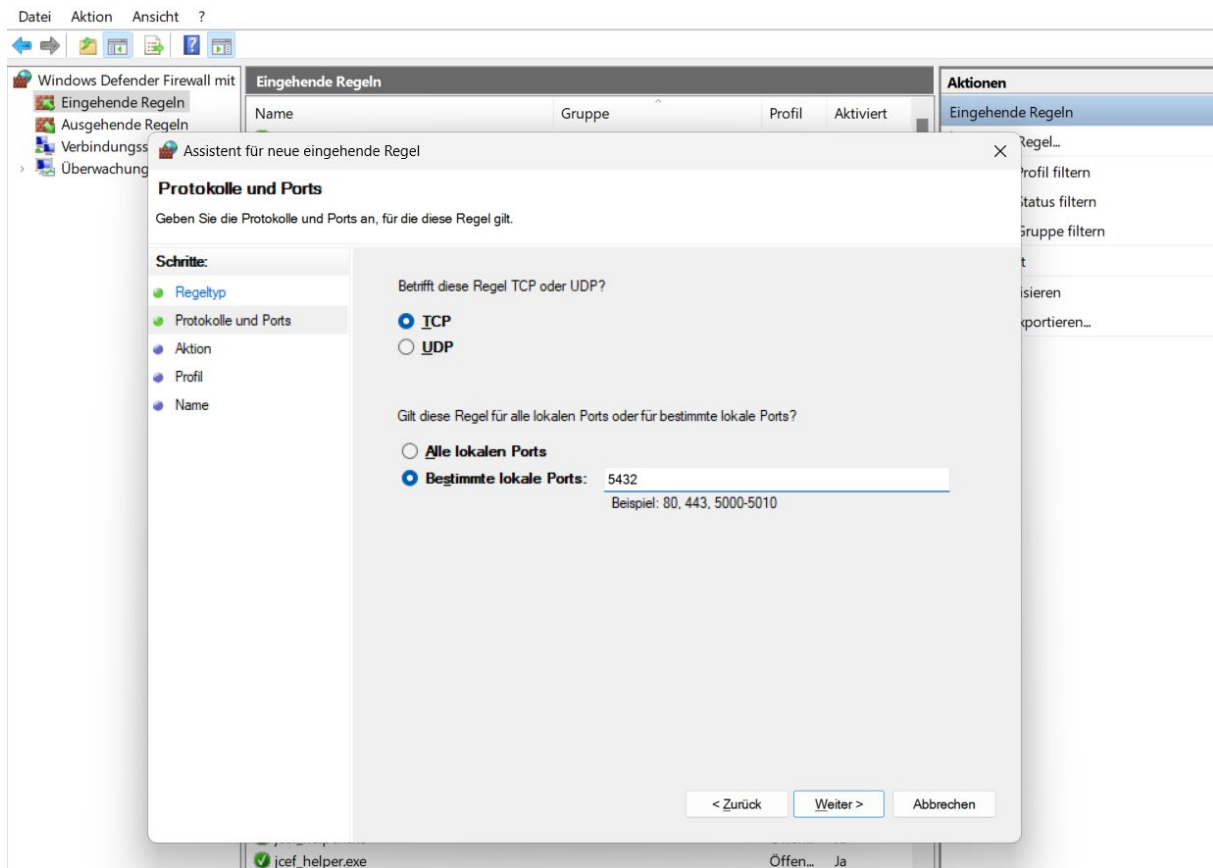


Abbildung 13 - Neue Firewall Regel erstellen und als Port 5432 nehmen

## Ergebnis:

### Von Windows auf Linux:

```
-U .... Benutzer (postgres)
-h .... IP-Adresse (10.0.0.41)
```

Danach das Password eingeben:

```
PS C:\Users\bsulj> psql -U postgres -h 10.0.0.41
Passwort für Benutzer postgres:
psql (16rc1, Server 13.11 (Debian 13.11-0+deb11u1))
Warnung: Konsolencodeseite (850) unterscheidet sich von der Windows-
        Codeseite (1252). 8-Bit-Zeichen funktionieren möglicherweise nicht
        richtig. Einzelheiten finden Sie auf der psql-Handbuchseite unter
        »Notes for Windows users«.
SSL-Verbindung (Protokoll: TLSv1.3, Verschlüsselungsmethode: TLS_AES_256_GCM_SHA384, Komprimierung: aus)
Geben Sie »help« für Hilfe ein.

postgres=# |
```

Abbildung 14 - Postgresql verbindung von Windows auf Linux

### Von Linux auf Windows:

```
-U .... Benutzer (postgres)
-h .... IP-Adresse (10.0.0.41)
```

Danach das Password eingeben:

```
root@Debian-Client-Suljevic:~# psql -U postgres -h 10.0.0.32
Password for user postgres:
psql (13.11 (Debian 13.11-0+deb11u1), server 16rc1)
WARNING: psql major version 13, server major version 16.
        Some psql features might not work.
Type "help" for help.

postgres=# █
```

Abbildung 15 - Postgresql verbindung von Linux auf Windows