



# PostgreSQL17

## Installation auf Linux-Systemen

### für ASV-BW

Stand: 16.04.2025

#### Kontaktdaten SCS:

Telefon: 0711 89246-0  
E-Mail [sc@schule.bwl.de](mailto:sc@schule.bwl.de)

#### Weitere Informationsquellen:

Im Intranet <https://ov.kv.bwl.de/wdb>  
Im Internet [www.asv-bw.de](http://www.asv-bw.de)

---

## Inhaltsverzeichnis

1. Einführung .....	3
2. Installation von PostgreSQL 17 mit Ubuntu 22 LTS.....	3
2.1 Konfigurationsdateien bearbeiten	4
2.2 PostgreSQL deinstallieren	6
3. pgAdmin4 .....	6
4. Datenbank .....	8
4.1 Backup/Restore	8

## 1 Einführung

### ASV-BW Datenbank

Ab ASV-BW Version 2.30 unterstützt ASV-BW die PostgreSQL-Datenbank bis Version 17.

#### Ziel der Handreichung

Die vorliegende Handreichung führt in die Installation und Administration des relationalen Datenbanksystems PostgreSQL für das Schulverwaltungsprogramm des Landes Baden-Württemberg (ASV-BW) ein. Das Datenbanksystem wird von diesem Programm als Speicherort für sämtliche Daten genutzt, die im Rahmen der Schulverwaltung anfallen. Es bietet dem Programm die Möglichkeit, eine eigene Datenbank zu nutzen, in der die Daten effektiv gespeichert und bedarfsgerecht abgefragt werden können. Das Datenbanksystem kann verschiedene Datenbanken verwalten.

Nach der Installation einer neuen Versionsnummer von PostgreSQL ist ASV-BW ebenfalls neu zu installieren, da es sonst zu Kommunikationsproblemen kommt. Durch die Neuinstallation beider Komponenten verfügen Sie über ein aktuelles System aus Datenbank- und Schulverwaltungssoftware. Ihr letztes Backup von ASV-BW können Sie in die neue Version einspielen.

Wir stellen für PostgreSQL kein Installationspaket zur Verfügung, sondern empfehlen die Installation über den Paketmanager der jeweiligen Distribution.

#### Upgradeinstallation

Es ist sehr aufwändig zwei unterschiedliche PostgreSQL-Versionen parallel einzurichten. Deshalb empfehlen wir, die Datenbank aus dem alten System zu sichern und dann die vorhandene PostgreSQL-Instanz zu deinstallieren. Danach kann die neue Version installiert werden.

## 2 Installation von PostgreSQL 17 mit Ubuntu 22 LTS

Löschen Sie zunächst vorhandene Installationen von PostgreSQL, nachdem die ASV-BW Datenbank gesichert wurde.

```
sudo apt-get --purge remove postgresql
sudo apt-get purge postgresql*
sudo apt-get --purge remove postgresql postgresql-doc postgresql-
common
```

PostgreSQL 17 ist nicht standardmäßig verfügbar. Ergänzen Sie die notwendigen Pakete von PostgreSQL

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

```
wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
tee /etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null
```

Aktualisieren Sie apt

```
sudo apt update
```

Prüfen Sie die vorhandenen Versionen:

```
apt search postgresql |grep postgresql-server
```

```

postgresql-server-dev-10/jammy-pgdg 10.23-7.pgdg22.04+1 amd64
postgresql-server-dev-11/jammy-pgdg 11.22-9.pgdg22.04+1 amd64
postgresql-server-dev-12/jammy-pgdg 12.22-2.pgdg22.04+1 amd64
postgresql-server-dev-13/jammy-pgdg 13.20-1.pgdg22.04+1 amd64
postgresql-server-dev-14/jammy-updates,jammy-security 14.17-0ubuntu0.22.04.1 i386
postgresql-server-dev-15/jammy-pgdg 15.12-1.pgdg22.04+1 amd64
postgresql-server-dev-16/jammy-pgdg 16.8-1.pgdg22.04+1 amd64
postgresql-server-dev-17/jammy-pgdg 17.4-1.pgdg22.04+2 amd64
postgresql-server-dev-17-dbg/jammy-pgdg 17.4-1.pgdg22.04+2 amd64
debug symbols for postgresql-server-dev-17

```

Installieren Sie PostgreSQL

```
sudo apt install postgresql-17 postgresql-client-17 -y
```

Prüfen Sie, ob das Datenbanksystem läuft

```
sudo systemctl status postgresql
```

```

root@Ubuntu22:/# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2025-04-16 11:26:24 CEST; 1min 35s ago
     Main PID: 473092 (code=exited, status=0/SUCCESS)
        CPU: 1ms

```

Prüfen Sie die installierte Version

```
psql --version
```

```

root@Ubuntu22:/# psql --version
psql (PostgreSQL) 17.4 (Ubuntu 17.4-1.pgdg22.04+2)
root@Ubuntu22:/#

```

Melden Sie sich als User postgres an

```
sudo -u postgres psql
```

Ändern Sie das Passwort für Superuser postgres und notieren Sie das neue Passwort.

```
alter user postgres password 'strenggeheim'
```

Verlassen Sie psql

```
\q
```

## 2.1 Konfigurationsdateien bearbeiten

Die Konfiguration befindet sich im Verzeichnis /etc/postgresql/17/main

### Anpassung der postgresql.conf Datei

Tragen Sie ein

- listen\_addresses= '\*'
- port= 5432
- max\_connections = 200
- max\_locks\_per\_transaction= 256 (Zeile 782)

```
#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'          # 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 = 200           # (change requires restart)
#reserved_connections = 0       # (change requires restart)
```

```
#-----
# LOCK MANAGEMENT
#-----

#deadlock_timeout = 1s
max_locks_per_transaction = 256 # min 10
```

### Anpassung der pg\_hba.conf Datei

Ergänzen Sie unter IPv4 local connections die Zeile

```
host all all 0.0.0.0/0 scram-sha-256
```

Die restlichen Einstellungen wie abgebildet:

```
GNU nano 6.2          pg_hba.conf
# DO NOT DISABLE!
# 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           scram-sha-256
host all          all              0.0.0.0/0             scram-sha-256
# 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              peer
host replication all              127.0.0.1/32           scram-sha-256
host replication all              ::1/128                scram-sha-256
```

Danach die Konfiguration neu laden:

```
sudo systemctl restart postgresql
```

## 2.2 PostgreSQL deinstallieren

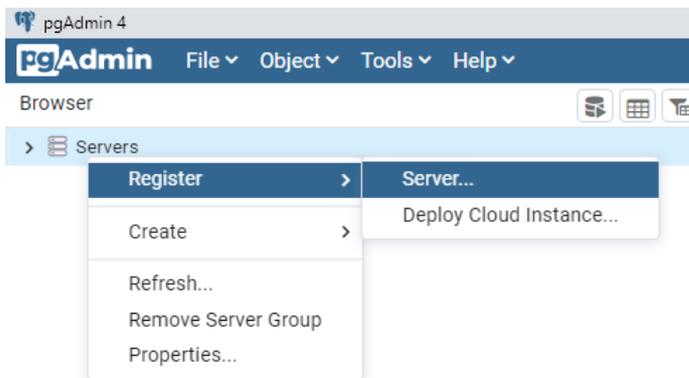
```
sudo apt-get --purge remove postgresql
sudo apt-get purge postgresql*
sudo apt-get --purge remove postgresql postgresql-doc postgresql-
common
```

## 3 pgAdmin4

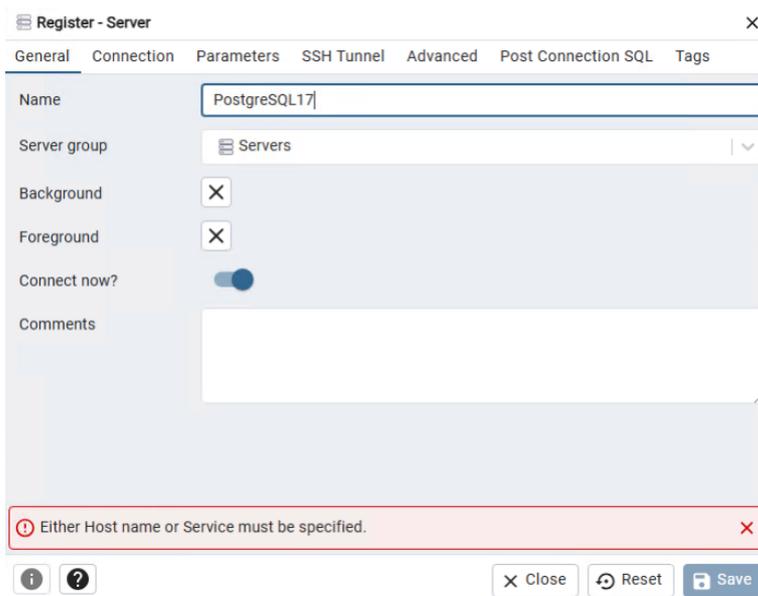
Das Datenbank-Verwaltungstool pgAdmin4 benötigt eine graphische Oberfläche. Wenn der Linux-Server keine graphische Oberfläche hat, oder die Distribution kein Installationspaket anbietet, muss pgAdmin4 auf einem Windows-Rechner im Netzwerk installiert werden.

pgAdmin4 kann sich mit der PostgreSQL-Instanz auf dem Linux-Server verbinden

Mit Rechtsklick auf Servers unter Register > Server wählen

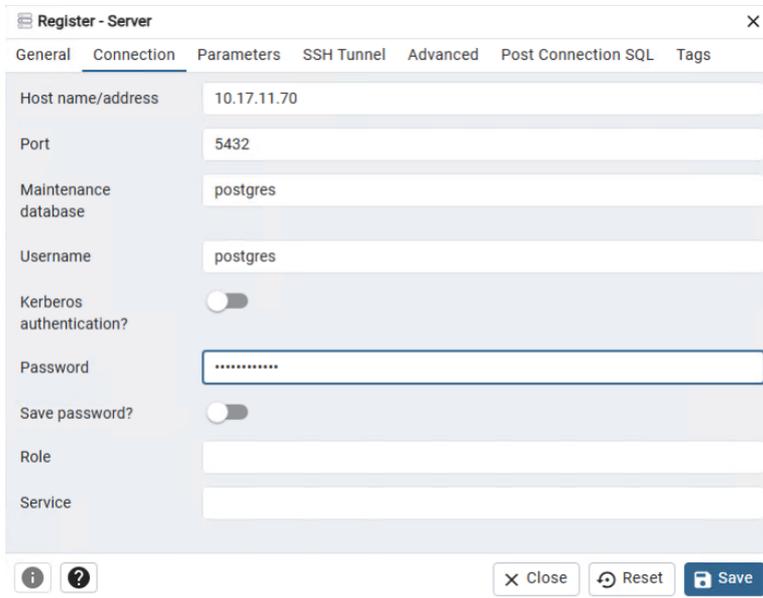


Auf der ersten Registerkarte muss ein Namen eingetragen werden, z.B. PostgreSQL17

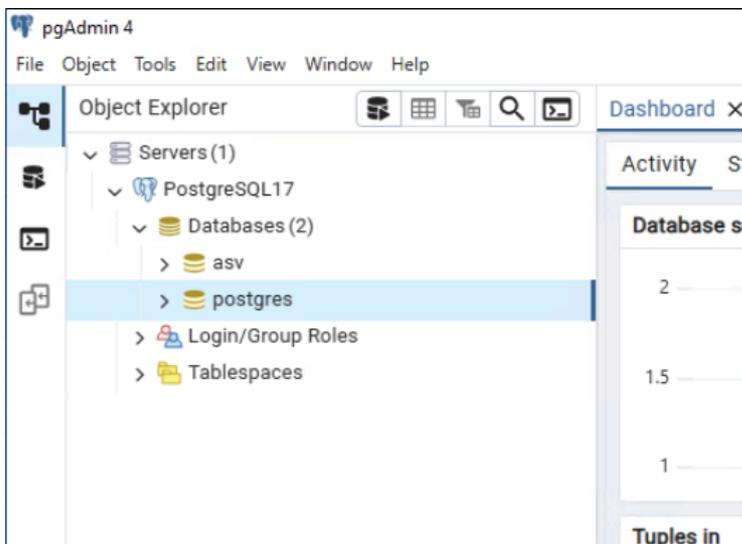


Auf der Registerkarte Connection werden die Verbindungsdaten eingetragen

Host = IP Adresse des Linux-Servers  
 Passwort = Passwort von User postgres



Serververbindung ist hergestellt und die Datenbanken können verwaltet werden.



## 4 Datenbank

### 4.1 Backup/Restore

Die Datensicherung und -wiederherstellung kann zunächst manuell durchgeführt werden mit den jeweiligen Befehlen im bin Verzeichnis der Installation. Auch eine automatische Datensicherung kann eingerichtet werden.

#### **Backup - ein komprimiertes Backup der asv Datenbank erzeugen (als root)**

pg\_dump benötigt Hostname, Port, Benutzername, Optionen, Dateipfad/Bezeichnung, Datenbankname

```
pg_dump -h localhost -p 5432 -U postgres -Fc -f <backup-name>.backup asv
```

Ein so erzeugtes Backup ist nachher auch mit pgAdmin 4 wieder zurückspielbar (in diesem Fall ist die Dateiergung .backup wichtig)

#### **Restore**

pg\_restore benötigt Hostname, Datenbankname, Benutzername, Dateipfad/Bezeichnung der .backup-Datei.

```
pg_restore -h localhost -p5432 -d asv -U postgres -c <backup-name>.backup
```

#### **Automatische Datensicherung mit ASV-BW Updatedienst**

für die Nutzung der automatischen Datensicherung von ASV-BW muss beim Updatedienst in der Konfiguration der Pfad zu den pg-Tools eingetragen werden.

Dazu wird im Verzeichnis /opt/asv/server/update-service/service/configuration eine Datei config.local.ini angelegt.

Dort tragen Sie diese Zeile ein:

```
cdbbackup.db.tools=/usr/bin
```

#### **Automatische Datensicherung mit einem Skript**

Das Skript im hier gezeigten Beispiel erstellt einmal an jedem Wochentag eine Datensicherung der ASV-BW-Datenbank. Die Sicherungen des vergangenen Woche werden jeweils überschrieben. Somit kann man immer auf die Backups einer zurückliegenden Woche zurückgreifen. Eine Anpassung des Skripts auf andere Zeitintervalle ist jederzeit möglich. Zusätzlich wird bei jeder Datensicherung eine backup.log-Datei geschrieben zur Überwachung des Ablaufs.

Legen Sie zunächst das Zielverzeichnis für die Backupdateien an.

```
mkdir /root/backup/  
mkdir /root/backup/asv
```

Erstellen Sie im Verzeichnis /root/backup/asv eine neue Datei, z.B. autobackup

```
touch autobackup
```

Als Shell-Skript muss diese Datei ausführbar gemacht werden

```
chmod +x autobackup
```

Öffnen Sie die die Datei autobackup mit Ihrem Editor und kopieren Sie den folgenden Text hinein:

```
#!/bin/bash
# automatisches Backup für ASV Datenbank
HOMEDIR='/root/backup/asv'
DATABASE=asv
HOSTNAME=127.0.0.1
PORT=5432
#Ab hier keine Änderungen vornehmen
DUMP_FILE=$HOMEDIR/"asv"`date +%w'`.backup.bz2"
echo "-----"
echo "Start: "`date`
echo -e "\n\nasv.backup ($DUMP_FILE):\n"
pg_dump -h $HOSTNAME -p $PORT -U postgres -Fc $DATABASE |bzip2 > $DUMP_FILE
echo -e "\nEnde: "`date` "\n"
```

Passen Sie die Werte für HOME, DATABASE, PORT bei Bedarf auf Ihre Gegebenheiten an.

Damit beim Ausführen der Datensicherung die Passwortabfrage des Benutzers postgres unterdrückt wird, muss im Homeverzeichnis des ausführenden Benutzers die Datei .pgpass erzeugt werden.

```
nano /root/.pgpass
```

In diese Datei werden die Zugangsdaten für User postgres eingefügt.

```
hostname:port:database:username:password
```

Die Datei wird nur berücksichtigt, wenn die Benutzerrechte eingeschränkt sind.

```
chmod 0600 /root/.pgpass
```

In diesem Beispiel sind Wildcards für alle Ports und alle Datenbanken eingetragen, sowie der Benutzername postgres mit dem Passwort postgres:



```
Terminal - scs@debian: ~
Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe
GNU nano 2.2.6 Datei: .pgpass
localhost:*:*:postgres:postgres
```

Für die automatisierte Ausführung des Skripts wird nun ein Cronjob eingerichtet. Dazu muss die Datei crontab editiert werden.

Für root befindet sie sich im Verzeichnis /etc/crontab. Für andere User kann eine crontab-Datei je nach Distribution in das Verzeichnis /var/spool/cron oder /var/spool/cron/crontabs mit dem Usernamen als Dateibezeichnung kopiert werden.

Aufruf mit `crontab -e`

Hier wird tabellenartig das Zeitintervall eingetragen, der Username und der auszuführende Befehl.

In diesem Beispiel wird jeden Tag um 16.00 Uhr ein Backup erstellt und eine log-Datei in das Verzeichnis /root/backup/asv geschrieben.

```
0 16 * * * /root/backup/asv/autobackup >> root/backup/asv/backup.log 2>&1
```

```

scs@localhost:~/var/spool/cron
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
GNU nano 2.0.9 Datei: /etc/crontab

SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# ..... minute (0 - 59)
# | ..... hour (0 - 23)
# | | ..... day of month (1 - 31)
# | | | ..... month (1 - 12) OR jan,feb,mar,apr ...
# | | | | ..... day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
0 16 * * * root /root/backup/asv/autobackup >> /root/backup/asv/backup.log 2>&1

^G Hilfe      ^O Speichern  ^R Datei öffnen ^Y Seite zurück ^K Ausschneiden ^C Cursor
^X Beenden    ^J Ausrichten ^W Wo ist       ^V Seite vor   ^U Ausschn. rüc ^T Rechtschr.
    
```

Nach einer Woche befinden sich im Verzeichnis /root/asv/backup die Sicherungen von jedem Wochentag (0 = Sonntag usw.).

```

scs@localhost:~/~/backup/asv
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[root@localhost asv]# ll
insgesamt 36
-rw-r--r--. 1 root root 725 21. Jun 16:00 asv0.backup.bz2
-rw-r--r--. 1 root root 721 15. Jun 16:00 asv1.backup.bz2
-rw-r--r--. 1 root root 719 16. Jun 16:00 asv2.backup.bz2
-rw-r--r--. 1 root root 718 17. Jun 16:00 asv3.backup.bz2
-rw-r--r--. 1 root root 722 18. Jun 16:00 asv4.backup.bz2
-rw-r--r--. 1 root root 722 19. Jun 16:00 asv5.backup.bz2
-rw-r--r--. 1 root root 721 20. Jun 16:00 asv6.backup.bz2
-rwxrwxrwx. 1 root root 600 22. Jun 07:53 autobackup
-rw-r--r--. 1 root root 3733 22. Jun 08:13 backup.log
[root@localhost asv]#
    
```