



PostgreSQL15

Installation auf Linux-Systemen

für ASV-BW

Stand: 26.04.2024

Kontaktdaten SCS:

Telefon: 0711 89246-0
E-Mail sc@schule.bwl.de

Weitere Informationsquellen:

Im Intranet <https://ov.kv.bwl.de/wdb>
Im Internet www.asv-bw.de

Inhaltsverzeichnis

1. Einführung	3
2. Installation von PostgreSQL 15 mit openSUSE Leap 15.4.....	3
2.1 Konfigurationsdateien bearbeiten	4
2.2 PostgreSQL deinstallieren	6
3. Installation von PostgreSQL 15 mit Ubuntu 22 LTS.....	6
3.1 Konfigurationsdateien bearbeiten	7
3.2 PostgreSQL deinstallieren	8
4. pgAdmin4	9
5. Datenbank.....	11
5.1 Backup/Restore	11

1 Einführung

ASV-BW Datenbank

Ab ASV-BW Version 2.22 unterstützt ASV-BW die PostgreSQL-Datenbank bis Version 15.

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 15 mit openSUSE Leap 15.4

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

```
zypper rm -u postgresql10
```

Entfernen des data-Verzeichnisses

```
rm -rf /var/lib/pgsql/data
```

Prüfen Sie die verfügbaren Pakete:

```
sudo zypper se --provides libpq5
```

```
localhost:/ # zypper se --provides libpq5
Repository-Daten werden geladen...
Installierte Pakete werden gelesen...

S | Name | Summary | Type
-----|-----|-----|-----
i+ | libpq5 | Gemeinsam genutzte Bibliotheken, die für PostgreSQL-Clients benötigt werden | Paket
  | libpq5-32bit | Gemeinsam genutzte Bibliotheken, die für PostgreSQL-Clients benötigt werden | Paket
  | postgresql14-devel-mini | PostgreSQL client development header files and libraries | Paket
```

```
sudo zypper se --provides postgresql-server
```

zeigt verfügbare PostgreSQL-Versionen an

ASV-BW arbeitet mit PostgreSQL in den Versionen 12 bis 15.

Installieren Sie folgende Pakete:

```
sudo zypper install libpq5
sudo zypper install postgresql15-server
```

Start des Datenbanksystems:

```
sudo systemctl start postgresql
```

Anmelden mit User postgres

```
sudo -u postgres psql
```

Ändern Sie das Passwort von Superuser postgres:

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

psql verlassen:

```
\q
```

2.1 Konfigurationsdateien bearbeiten

Die Konfiguration befindet sich im Verzeichnis `/var/lib/pgsql/data`

Anpassung der `postgresql.conf` Datei

Tragen Sie ein

- `listen_addresses = '*'`
- `port = 5432`
- `max_locks_per_transaction = 100`

```
GNU nano 5.9 /var/lib/pgsql/data/postgresql.conf
# 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 = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)

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

#deadlock_timeout = 1s
max_locks_per_transaction = 100 # min 10
                                # (change requires restart)
#max_pred_locks_per_transaction = 64 # min 10
```

Anpassung der `pg_hba.conf` Datei

Ergänzen Sie unter IPv4 local connections die Zeile

```
host all all 0.0.0.0/0 trust
```

Die restlichen Einstellungen wie abgebildet:

```
# 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

Entfernen der Installation, z.B. PostgreSQL 10

```
zypper rm -u postgresql10
```

Entfernen des data-Verzeichnisses

```
rm -rf /var/lib/pgsql/data
```

3 Installation von PostgreSQL 15 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 15 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 10.23-1.pgdg22.04+1 amd64
postgresql-server-dev-11/jammy-pgdg 11.19-1.pgdg22.04+1 amd64
postgresql-server-dev-12/jammy-pgdg 12.14-1.pgdg22.04+1 amd64
postgresql-server-dev-13/jammy-pgdg 13.10-1.pgdg22.04+1 amd64
postgresql-server-dev-14/jammy-updates 14.6-0ubuntu0.22.04.1 i386
postgresql-server-dev-15/jammy-pgdg 15.2-1.pgdg22.04+1 amd64
postgresql-server-dev-8.2/jammy-pgdg 8.2.23-9.pgdg22.04+1+b1 amd64
postgresql-server-dev-8.3/jammy-pgdg 8.3.23-7.pgdg22.04+1+b1 amd64
postgresql-server-dev-8.4/jammy-pgdg 8.4.22-8.pgdg22.04+1+b1 amd64
postgresql-server-dev-9.0/jammy-pgdg 9.0.23-7.pgdg22.04+1+b1 amd64
postgresql-server-dev-9.1/jammy-pgdg 9.1.24-10.pgdg22.04+1 amd64
postgresql-server-dev-9.2/jammy-pgdg 9.2.24-7.pgdg22.04+1 amd64
postgresql-server-dev-9.3/jammy-pgdg 9.3.25-6.pgdg22.04+1 amd64
postgresql-server-dev-9.4/jammy-pgdg 9.4.26-5.pgdg22.04+1 amd64
postgresql-server-dev-9.5/jammy-pgdg 9.5.25-4.pgdg22.04+1 amd64
postgresql-server-dev-9.6/jammy-pgdg 9.6.24-3.pgdg22.04+1 amd64
postgresql-server-dev-all/jammy-pgdg 247.pgdg22.04+1 amd64
```

Installieren Sie PostgreSQL

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

Prüfen Sie, ob das Datenbanksystem läuft

```
sudo systemctl status postgresql
```

```
scs@UbuntuLTS:/$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor p
   Active: active (exited) since Wed 2023-02-22 11:22:20 CET; 12min ago
     Main PID: 4635 (code=exited, status=0/SUCCESS)
        CPU: 1ms
```

Prüfen Sie die installierte Version

```
psql --version
```

```
scs@UbuntuLTS:/$ psql --version
psql (PostgreSQL) 15.2 (Ubuntu 15.2-1.pgdg22.04+1)
```

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
```

3.1 Konfigurationsdateien bearbeiten

Die Konfiguration befindet sich im Verzeichnis /etc/postgresql/main/15

Anpassung der postgresql.conf Datei

Tragen Sie ein

- listen_addresses= '*'

- port= 5432

- max_locks_per_transaction= 100

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

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100          # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of dire
                                # (change requires restart)
```

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

#deadlock_timeout = 1s
max_locks_per_transaction = 100          # min 10
                                         # (change requires restart)
```

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
```

3.2 PostgreSQL deinstallieren

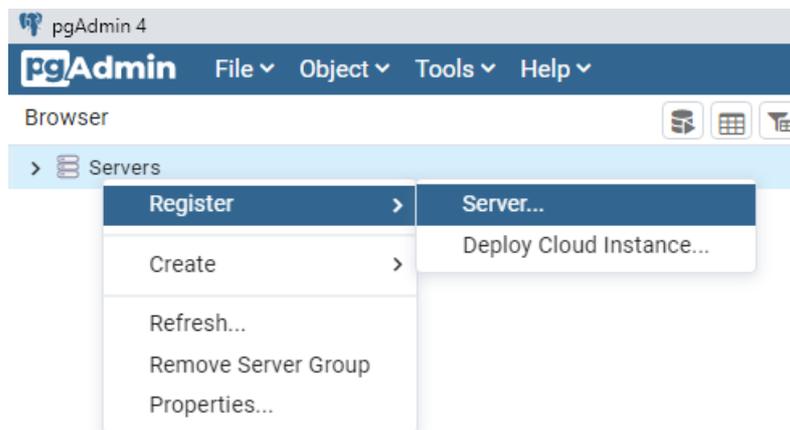
```
sudo apt-get --purge remove postgresql
sudo apt-get purge postgresql*
sudo apt-get --purge remove postgresql postgresql-doc postgresql-
common
```

4 pgAdmin4

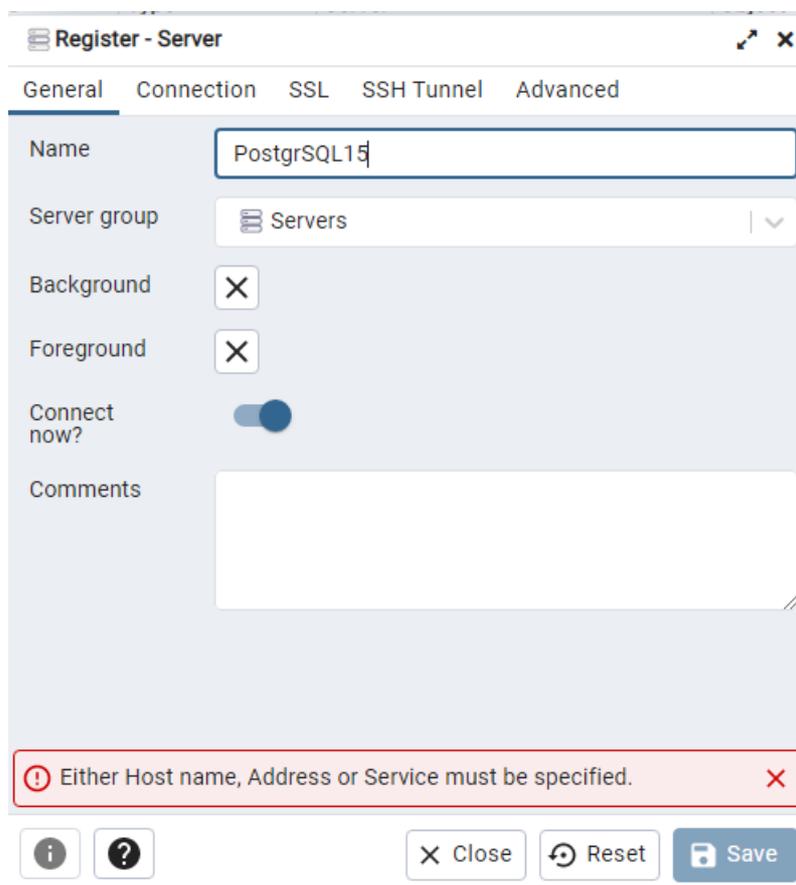
Das Datenbank-Verwaltungstool pgAdmin4 benötigt eine graphische Oberfläche. Wenn der Linux-Server keine graphische Oberfläche hat, oder die Distribution kein Installationspaket anbietet, muss pgAdmin4 auf einem Windows-Rechner im Netzwerk installiert werden.

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

Mit Rechtsklick auf Servers unter Register > Server wählen



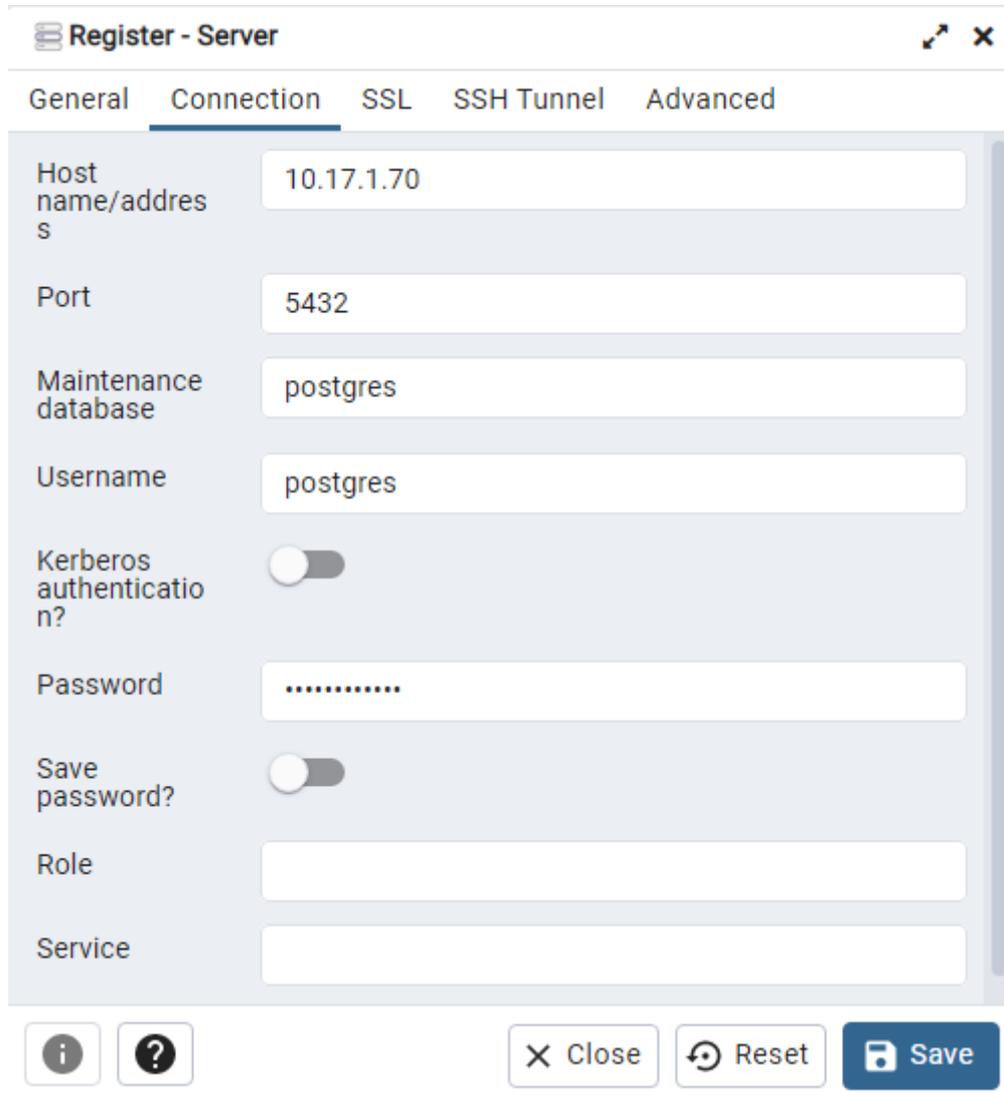
Auf der ersten Registerkarte muss ein Namen eingetragen werden, z.B. PostgreSQL 15



Auf der Registerkarte Connection werden die Verbindungsdaten eingetragen

Host = IP Adresse des Linux-Servers

Passwort = Passwort von User postgres

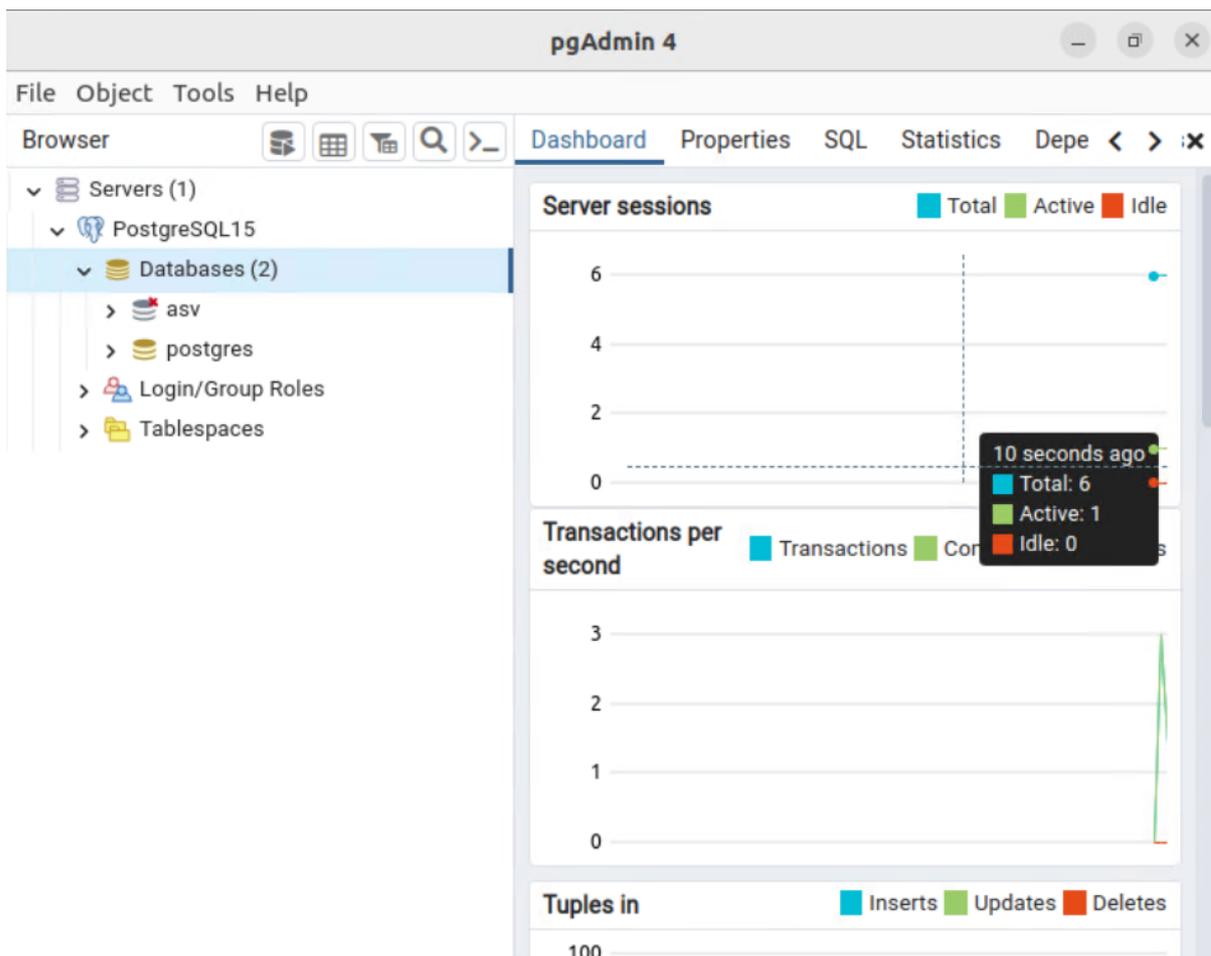


The screenshot shows the 'Register - Server' dialog box in pgAdmin4, with the 'Connection' tab selected. The fields are filled with the following values:

Field	Value
Host name/addresses	10.17.1.70
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	<input type="checkbox"/>
Password
Save password?	<input type="checkbox"/>
Role	
Service	

At the bottom of the dialog, there are buttons for 'Close', 'Reset', and 'Save', along with information and help icons.

Serververbindung ist hergestellt und die Datenbanken können verwaltet werden.



5 Datenbank

5.1 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 III wieder zurückspielbar (in diesem Fall ist die Dateierdung .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:

```
cdbackup.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

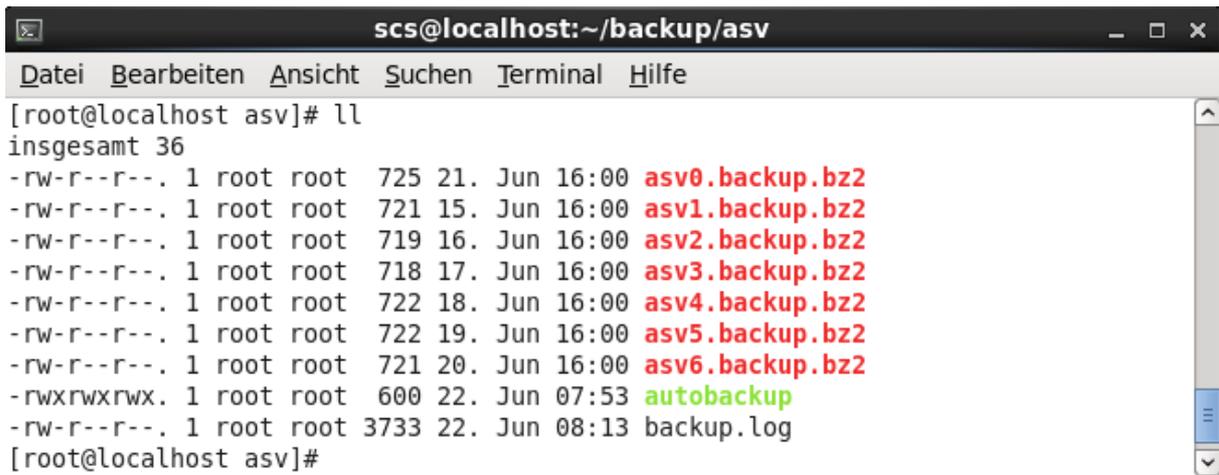
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# 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. rück ^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]#
```

