

Setting a Linux database server for TScore

If you are running a large competition with tablet scoring it is likely that you will hit the Windows network connections limit. Unless you have a Windows Server Edition (very expensive!) once you connect more than 20 devices to your database server, Windows may start to refuse additional connections. This makes scoring rather 'hit and miss'! You should not even attempt this.

Instead, we advise that you deploy a Linux PC to act purely as a database server. This might be an old or new laptop, a mini-PC, or even a 'Raspberry Pi' device. Although TScore itself isn't supported on Linux, the PostgreSQL database is very fast and reliable on these operating systems.

Regarding hardware configuration, it depends very much on the age of the equipment. For example a 10 year-old i7 laptop with 16GB RAM and a 128GB SSD can easily manage a large event with 30+ tablets. Newer computers with NVMe drives and more modern chipsets & processors should be able to run on 8GB RAM and an i5 or equivalent processor.

We recommend a Linux Mint distribution, as the Cinnamon desktop experience is similar to Windows, and it is built on the latest long-term supported version of Ubuntu. You can of course just use Ubuntu itself (or any other), but technical support for these may be limited if we haven't previously used your particular version.

When setting up Linux, as the server is most likely to be physically isolated from external systems, having it heavily locked down is overkill for most users. For convenience, you may wish to have it set to automatically login on boot, so all you need to do is plug it in and switch it on. Have the 'power options' set so that closing the lid does nothing and then you can just have it unobtrusively working away. A wired connection to the router is the most performant option if available (either directly or via a USB-Ethernet adapter)

So, once you have your baseline Linux operating system installed and running, you need to install PostgreSQL.

Here is a useful guide: <https://www.devart.com/dbforge/postgresql/how-to-install-postgresql-on-linux/>

The simplest route though is to just install it from the default Linux repositories. For MINT v22.3, this will install PostgreSQL v16. If you would prefer to install the newer version 18 (which has some notable performance improvements) skip to the next section in this document.

To install the default version of PostgreSQL, first bring the repositories up to date:

```
sudo apt update
```

Then install the PostgreSQL package:

```
sudo apt install postgresql
```

Follow the prompts for configuration. You should be able to take the defaults. If asked for a PostgreSQL username and password, just use 'postgres' for both.

When that completes, ensure that the PostgreSQL server is running, run the following command:

```
sudo systemctl start postgresql.service
```

To access the PostgreSQL prompt, type:

```
psql
```

To view the PostgreSQL server version running, use the command:

```
psql -V
```

To exit the PostgreSQL prompt, type:

```
\q
```

To check the PostgreSQL service status:

```
sudo systemctl status postgresql
```

To check the network post listener:

```
netstat -nlt | grep 5432
```

Alternative - Installing PostgreSQL manually

If you would prefer to install the newest version of PostgreSQL (currently v18) then you must instead do the following (for more details, see <https://www.postgresql.org/download/linux/ubuntu>):

```
# Import the repository signing key and install common libraries:
sudo apt install curl ca-certificates
sudo install -d /usr/share/postgresql-common/pgdg
sudo curl -o /usr/share/postgresql-common/pgdg/apt.postgresql.org.asc --fail
https://www.postgresql.org/media/keys/ACCC4CF8.asc

# Create the repository configuration file:
sudo sh -c "echo 'deb [signed-by=/usr/share/postgresql-common/pgdg/apt.postgresql.org.asc]
https://apt.postgresql.org/pub/repos/apt noble-pgdg main' > /etc/apt/sources.list.d/pgdg.list"

# Update the package lists and install PostgreSQL version 18:
sudo apt update
sudo apt install postgresql-18

Follow the prompts for configuration. You should be able to take the defaults. If asked for a PostgreSQL
username and password, just use 'postgres' for both.

# Check it has installed and launched ok:
sudo systemctl status postgresql
```

By default, PostgreSQL installs with no external network access. Please follow the instruction in the next section to make it usable by TScore.

PostgreSQL Server Settings

Whichever PostgreSQL installation you have, you will need to alter the configuration in order to use it over a network for TScore.

First configure both the main 'postgresql.conf' and the 'pg_hba.conf' connection files to allow easy remote access to the database server.

These files should be in one of these folders:

```
/etc/postgresql/[version]/main/
```

Or

```
/var/lib/postgresql/[version]/main
```

In **postgresql.conf**, you need to relax the connection constraints to make life simpler. Note that some of the configuration flags below will be commented out (with a # symbol). Simply remove the #.

```
ssl = off #This removes the need for trusted server SSL certificates.
```

Turn on Logging to help debug connection issues:

```
# Logging
logging_collector = on
log_directory = 'log'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_connections = on
log_disconnections = on
log_hostname = on
```

And allow external connections from any address:

```
# Connection settings - make sure these are set
listen_addresses = '*' # or specific IP
port = 5432
```

Now, modify `pg_hba.conf` to allow remote connections from the local network:

```
host    all         all         127.0.0.1/32    trust
host    all         all         192.168.1.0/16  trust
```

Once these have been set and saved, restart the PostgreSQL service

```
sudo systemctl restart postgresql.service
```

You should now be able to connect from TScore as usual.

Note – once it is running and tested ok, you may as well disable all logging:

Just set `logging_collector = off` in `postgresql.conf`