

Configuration

In order to set up a database connection, modify the file `/etc/rstudio/database.conf`. The file contains documentation about how to use it, and you can simply uncomment any lines that are relevant to your configuration. Note that because the file can contain password data, this file must be user read/write **only** (file mask 600).

SQLite

By default, Posit Workbench creates a SQLite database for you automatically under the `/var/lib/rstudio-server` directory. For single-node installations, this is sufficient, but as stated before, will not be sufficient for load balanced deployments.

You should **never** specify a SQLite directory that is on shared storage, such as NFS. Per the [SQLite documentation](#), this can cause data corruption.

Sample configuration:

Listing 1 `/etc/rstudio/database.conf`

```
provider=sqlite

# Directory in which the sqlite database will be written
directory=/var/lib/rstudio-server
```

PostgreSQL

If you wish to use PostgreSQL, you must create an empty database for Posit Workbench to connect to. You must not share this database with other products or services. If running an HA setup, it is strongly recommended to run the database on a separate server than the Posit Workbench services to ensure maximum availability. The minimum supported version is PostgreSQL 11. It requires a 1Ghz processor with 2GB of RAM, and 4GB of disk space.

Posit Workbench requires PostgreSQL to be configured for password-based authentication. For more details on this and other PostgreSQL fundamentals, see the support article [Install and Configure PostgreSQL for Posit Workbench / RStudio Server Pro](#).

Sample configuration:

Listing 2 /etc/rstudio/database.conf

```
# Note: when connecting to a PostgreSQL database, a default empty rstudio database must first be created
provider=postgresql

# Specifies the host (hostname or IP address) of the database host
host=localhost

# Specifies the database to connect to
database=rstudio

# Specifies the TCP port where the database is listening for connections
port=5432

# Specifies the database connection username
username=rstudio

# Specifies the database connection password. This may be encrypted with the secure-cookie-key
# The encrypted password can be generated using the helper command rstudio-server encrypt-password
# It is strongly recommended that you encrypt the password!
password=test

# Specifies the maximum amount of seconds allowed before a database connection attempt
# is considered failed. The default if not specified is 10 seconds. Corresponds to the
# PostgreSQL "connect_timeout=" connection string parameter.
connection-timeout-seconds=12
```

PostgreSQL connection URIs are also supported if preferred. If specifying additional options other than the ones provided above, such as `sslmode`, the use of a URI is required.

For example:

Available PostgreSQL connection string parameters are documented in the [official PostgreSQL documentation](#).

Listing 3 /etc/rstudio/database.conf

```
provider=postgresql

# Specifies the connection URL in the form of a postgresql:// connection URL. This can be used
# to set special database settings that are not available with the other parameters. If set,
# override any other postgresql parameters that have been set, with the exception of the password.
# the URI is supported as a convenience but we strongly recommend using the separate password
# always replace any password specified in the URI.
connection-uri=postgresql://rstudio@localhost:5432/rstudio?sslmode=allow&options=-csearch_path=
```

i Note

The password in `connection-uri` may contain characters that may need to be URL-encoded to work properly. Avoid encoding the password by using the separate `password` field in the configuration.

SSL

SSL for the PostgreSQL database can be used with Posit Workbench by specifying `sslmode=allow` within the `connection-uri` parameter. The `connection-uri` mode of configuration must be used to specify additional database connection options such as these, beyond the simple name/value pairs that are supported.

For example:

Listing 4 /etc/rstudio/database.conf

```
provider=postgresql
connection-uri=postgresql://postgres@localhost:5432/rstudio?sslmode=allow
```

Schemas

If you need to, you can tell Posit Workbench to restrict itself to keeping its tables within a specific schema. You control this by giving PostgreSQL a search path as part of the URL by adding `options=-csearch_path=<schema-name>` to the `connection-uri`. If it's the only item you're adding, separate it from the rest of the URL with `?` (just like the `sslmode` item above). Otherwise, separate it from other items with `'&'`.

For example:

```
provider=postgresql
connection-uri=postgresql://postgres@localhost:5432/rstudio?sslmode=allow&options=-csearch_p
```

Posit Workbench will refuse to start when given a schema that does not already exist. The schema must be owned by the connecting user or by a group that contains the connecting user.

PostgreSQL account

When setting up your PostgreSQL database for use with Posit Workbench, ensure that you do not use the default `postgres` user account that comes with a standard installation. This is to ensure that your database is secure. Always ensure that whichever account that is used to access the database contains a strong password - do **not** use an account that has no password! You should also ensure that only one PostgreSQL user has access to the Posit Workbench database for maximum security.

PostgreSQL password encryption

A plain-text password in the `password` or `connection-uri` options of the `/etc/rstudio/database.conf` file must only be used temporarily for testing purposes. A warning will be present in Posit Workbench log output when a plain-text password is being used.

We strongly recommend encrypting the password using the command `rstudio-server encrypt-password`. This way, if you have to backup your configuration, save it to a repository or share it with Posit Support, your PostgreSQL password will be protected.

Use the following steps to encrypt the PostgreSQL password:

- Remove the password from the `connection-uri` option if defined in the `database.conf` file.
- Run the command `sudo rstudio-server encrypt-password` and enter the PostgreSQL password.
- Copy the resulting encrypted password printed in the terminal.
- Add or replace the `password` option in the `database.conf` file using the encrypted password copied above.
- Restart Posit Workbench. Confirm it operates normally. You should no longer see a warning about plain-text password in Posit Workbench logs.

Note

Alternatively, you can also “pipe” your password to the `rstudio-server encrypt-password` command to skip the prompt. Useful when the password is already stored elsewhere. For example:

```
cat passwordfile | sudo rstudio-server encrypt-password
```

The password encryption uses the `secure-cookie-key` value. By default Posit Workbench generates this key during installation and stores it in `/var/lib/rstudio-server/secure-cookie-key`.

The same key value must be used for both encryption and decryption. If the key value used to encrypt the PostgreSQL password changes, the password must be re-encrypted with the new key and updated in `/etc/rstudio/database.conf`.

If preparing Posit Workbench configuration files on one system for deployment on other system(s), you must manually generate a key and store it in `/etc/rstudio/secure-cookie-key`, then encrypt the password again, update `/etc/rstudio/database.conf`, and ensure this `secure-cookie-key` file is deployed along with other Posit Workbench configuration files. The technique for creating this file is described in [Generating a key](#).

PostgreSQL connection testing and troubleshooting

Once the settings have been entered in `/etc/rstudio/database.conf`, use the `sudo rstudio-server verify-installation` command to test connectivity and quickly view errors and warnings.

Connection pool

Posit Workbench will create a pool of connections to the database at startup. The size of this pool defaults to the number of logical CPUs on the host running Posit Workbench, up to 20. It is not generally recommended that you adjust the size of this pool manually unless you need to address a specific problem, such as exceeding a connection limit on the database or experiencing delays in Posit Workbench caused by unavailable connections.

If you do need to adjust the size of the pool, you can do so by setting `pool-size` in `database.conf` as follows:

Listing 5 `/etc/rstudio/database.conf`

```
pool-size=5
```
