

- Guides & Tutorials (https://www.linode.com/docs/)
- » Database Management Systems (https://www.linode.com/docs/databases/)
 - » PostgreSQL (https://www.linode.com/docs/databases/postgresql/)
 - » How to Back Up Your PostgreSQL Database

How to Back Up Your PostgreSQL Database

Updated Monday, July 29, 2019 by Jared Kobos

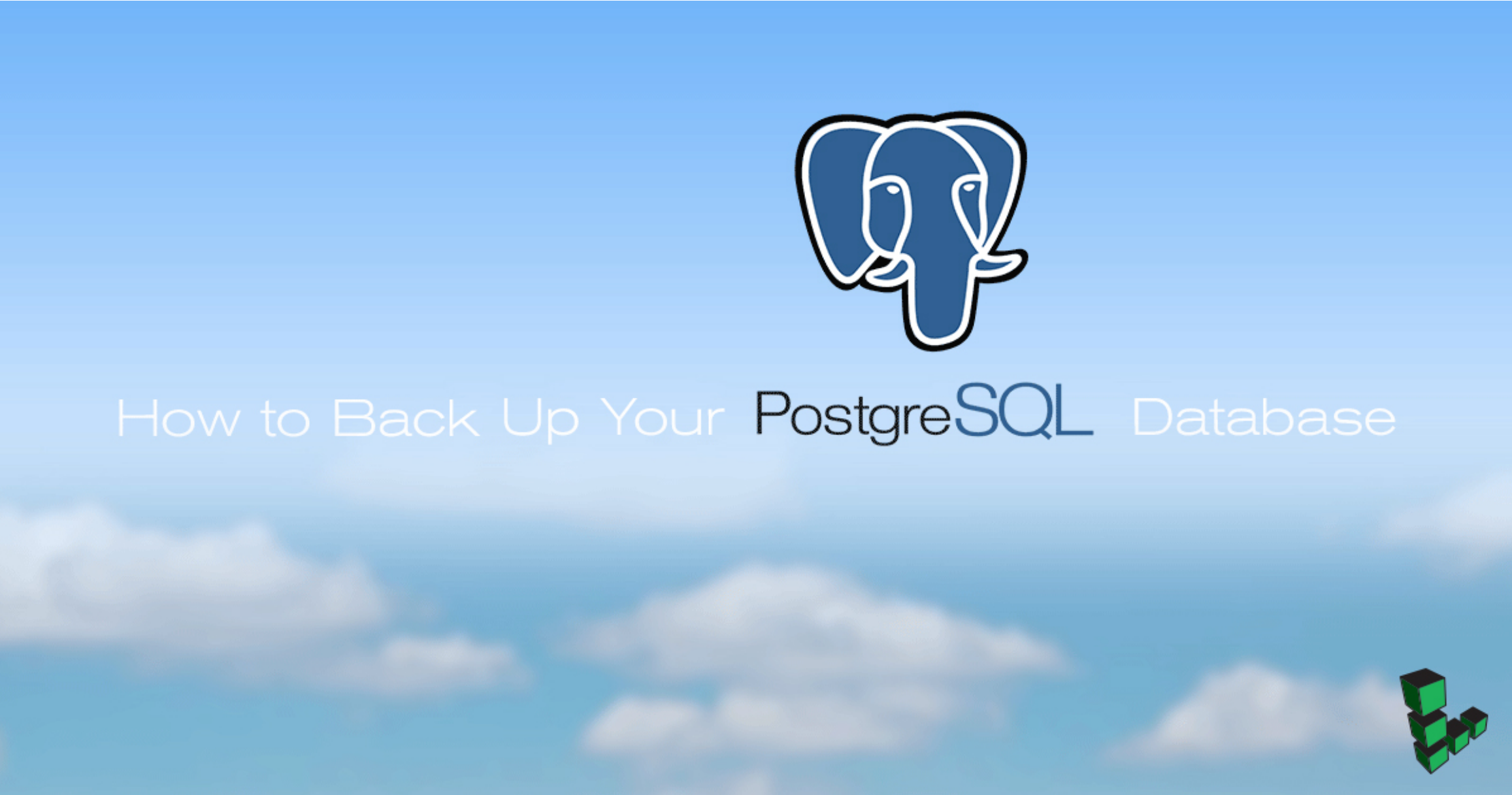
Written by Jared Kobos

Try this guide (https://login.linode.com/signup?promo=DOCS20AA00X1) to receive \$20 at signup on a new account.



Contribute on GitHub

Report an Issue (https://github.com/linode/docs/issues/new?title=How%20to%20Back%20Up%20Your%20PostgreSQL%20Database%20Proposed%20Changes&body=Link%3A+https%3A%2F%2Flinode.com%2Fdocs%2Fdatabases%2Fpostgresql-how-to-back-up-your-postgresql-database%2F%0A%23%23%20Issue%0A%0A%23%23%20Suggested%20Fix%0A&labels=inaccurate+guide)+View+File(https://github.com/linode/docs/blob/master/docs/databases/postgresql/how-to-back-up-your-postgresql-database/index.md)+Edit+File(https://github.com/linode/docs/edit/develop/docs/databases/postgresql/how-to-back-up-your-postgresql-database/index.md)



If you are using PostgreSQL in a production environment, it is important to take precautions to ensure that your users’ data is not lost. By frequently backing up your database, and/or automating backups with a cron task, you will be able to quickly restore your system in the event that your database is lost or corrupted. Fortunately, PostgreSQL includes tools to make this task simple and easy to manage.

Before You Begin

You should have a working installation of PostgreSQL on your system before beginning this guide. Go through our [How to Install PostgreSQL on Ubuntu](/docs/databases/postgresql/how-to-install-postgresql-on-ubuntu-16-04/) guide (/docs/databases/postgresql/how-to-install-postgresql-on-ubuntu-16-04/) to install PostgreSQL and create a sample database.

Note

The steps in this guide require root privileges. Be sure to run the steps below as `root` or with the `sudo` prefix. For more information on privileges, see our [Users and Groups](/docs/tools-reference/linux-users-and-groups/) (/docs/tools-reference/linux-users-and-groups/) guide.

One-Time SQL Dump

Single Database

PostgreSQL provides the `pg_dump` utility to simplify backing up a single database. This command must be run as a user with read permissions to the database you intend to back up.

1. Log in as the `postgres` user:

```
su - postgres
```

2. Dump the contents of a database to a file by running the following command. Replace `dbname` with the name of the database to be backed up.

```
pg_dump dbname > dbname.bak
```

The resulting backup file, `dbname.bak` , can be transferred to another host with `scp` or stored locally for later use.

3. To demonstrate restoring lost data, delete your example database and create an empty database in its place:

```
dropdb dbname
createdb dbname
```

4. Restore the database using `psql` :

```
psql test < dbname.bak
```

There are several options for the backup format:

- `*.bak` : compressed binary format
- `*.sql` : plaintext dump
- `*.tar` : tarball

Remote Database

Just as `psql` allows you to connect to a remote host, `pg_dump` can be run from a client computer to back up data on a remote server. Use the `-h` flag to specify the IP address of your Linode and `-p` to identify the port on which PostgreSQL is listening:

```
pg_dump -h 198.51.100.0 -p 5432 dbname > dbname.bak
```

All Databases

Because `pg_dump` only creates a backup of one database at a time, it does not store information about database roles or other cluster-wide configuration. To store this information, and back up all of your databases simultaneously, you can use `pg_dumpall` .

1. Create a backup file:

```
pg_dumpall >pg_backup.bak
```

2. Restore all databases from the backup:

```
psql -f pg_backup.bak postgres
```

Automate Backups with a Cron Task

You may want to set up a cron job so that your database will be backed up automatically at regular intervals. The steps in this section will set up a cron task that will run `pg_dump` once every week.

1. Make sure you are logged in as the `postgres` user:

```
su - postgres
```

2. Create a directory to store the automatic backups:

```
mkdir -p ~/postgres/backups
```

3. Edit the crontab to create the new cron task:

```
crontab -e
```

4. Add the following line to the end of the crontab:

crontab

1	0 0 * * 0 pg_dump -U postgres dbname > ~/postgres/backups/dbname.bak
---	--

5. Save and exit from the editor. Your database will be backed up at midnight every Sunday. To change the time or frequency of the updates, see our [Schedule Tasks with Cron \(/docs/tools-reference/tools/schedule-tasks-with-cron/\)](#) guide.

Next Steps

PostgreSQL also offers more advanced ways to back up your databases. The official docs (<https://www.postgresql.org/docs/9.1/static/continuous-archiving.html>) describe how to set up continuous archiving and point-in-time recovery. This is a much more complex process, but it will maintain a constant archive of your database and make it possible replay PostgreSQL’s logs to recover the state of the database at any point in the past.

This method can also be helpful if you have a very large database although continuously archiving a large database consumes resources. Since the process is ongoing, there is no need to make frequent and time consuming full backups.

More Information

You may wish to consult the following resources for additional information on this topic. While these are provided in the hope that they will be useful, please note that we cannot vouch for the accuracy or timeliness of externally hosted materials.

- PostgreSQL Documentation (<https://www.postgresql.org/docs/9.1/static/>)

Join our Community

Find answers, ask questions, and help others. (<https://www.linode.com/community/questions/>)

Was This Guide Helpful?

Your feedback is important to us. Let us know if this guide helped you find the answer you were looking for by taking this survey. (<https://delighted.com/t/iVc7nAYL?referrer=%2Fdocs%2Fdatabases%2Fpostgresql%2Fhow-to-back-up-your-postgresql-database%2F>)

This guide is published under a CC BY-ND 4.0 (<https://creativecommons.org/licenses/by-nd/4.0>) license.

Write for Linode.

We're always expanding our docs. If you like to help people, can write, and have expertise in a Linux or cloud infrastructure topic, learn how you can contribute (https://www.linode.com/lp/write-for-linode?utm_source=library&utm_medium=contribute-link&utm_campaign=write-for-linode) to our library.

Get started in the Linode Cloud today.

(<https://login.linode.com/signup>)



© 2019 All rights reserved.