**Purpose:**

Databases grow over time, sometimes outgrowing the space on their original file system. When they’re located on the same partition as the rest of the operating system, this can also potentially lead to I/O contention.

This tutorial will guide you through relocating PostgreSQL’s data directory.

## Prerequisites

To complete this guide, you will need:

* An Ubuntu 18.04 server with a non-root user with sudo privileges.
* PostgreSQL installed on your server.

In this example, we’re moving the data to a block storage device mounted at /mnt/volume\_nyc1\_01. If you are using Block Storage on DigitalOcean, [this guide](https://www.digitalocean.com/community/tutorials/how-to-use-block-storage-on-digitalocean) can help you mount your volume before continuing with this tutorial.

| **Step** | **Major Activity** | **References, Forms and Details** |
| --- | --- | --- |
| **1** | Enter psql as a superuser:  $ sudo -u postgres psql | **sudo -u postgres psql**  is the standard logon |
| **2** | Determine active PostgreSQL version data directory   * SHOW data\_directory * exit psql with “\q” | * Output will show the path. For this example   /var/lib/postgresql/10/main   * It is recommended that the path include a version number, to allow for multiple PostgreSQL instances and separate data directories |
| 3 | Switch to superuser   * sudo -i | * Where permitted |
| **3** | Stop PostgreSQL   * on pre-RHEL/Centos7:   service postgresql-{ver} stop   * on newer systems:   systemctl stop postgresql-{ver} | * Substitute actual version for {ver} * ensures the integrity of the data |
| **4** | Confirm PostgreSQL is stopped   * pre-RHEL/Centos7:   service postgresql-{ver} status | * newer systems:   sudo systemctl status postgresql-{ver} |
| **5** | Copy the directory as superuser  rsync -av /var/lib/postgresql /mnt/volume\_nyc1\_01 | * Be very careful. Rsync is fast, but it can overwrite files with impunity * flag -a preserves the permissions and other directory properties * flag -v provides verbose output * source is:   /var/lib/postgresql   * destination is:   /mnt/volume\_nyc1\_01 |
| **6** | Once the copy is complete, we’ll rename the current folder with a .bak extension  mv /var/lib/postgresql/10/main  /var/lib/postgresql/10/main.bak |  |
| **7** | Locate postgresql.conf   * edit to point to new file location, * save * close \*.conf   data\_directory = '/mnt/volume\_nyc1\_01/postgresql/10/main' | * Postgesql.conf may be found in /etc tree, or /var tree |
| 8 | Restart the PostgreSQL server   * pre-RHEL/Centos7:   service postgresql-{ver} start | * newer systems:   sudo systemctl start postgresql-{ver} |
| 9 | Confirm PostgreSQL is started   * pre-RHEL/Centos7:   service postgresql-{ver} status | * newer systems:   sudo systemctl status postgresql-{ver} |
| 10 | Access database and confirm directory  sudo -u postgres psql  Check the value for the data directory again:  postgres# SHOW data\_directory; | * In this example, should see   /mnt/volume\_nyc1\_01/postgresql/10/main |