[](https://medium.com/@plashchynski?source=post_header_lockup)

[Dzmitry Plashchynski](https://medium.com/@plashchynski?source=post_header_lockup)

Aug 12, 2016

**PostgreSQL upgrade on CentOS**

(<https://medium.com/@plashchynski/postgresql-upgrade-on-centos-4c0ddd2f8687>)

In our case we are going to upgrade PostgreSQL 9.5 to 9.6 on CentOS 7.2. However, the following is easily applicable to all Red Hat family (RHEL/CentOS/SL/OL 7) and to all PostgreSQL 9.\* versions. The upgrading flow is well covered in the [official documentation](https://www.postgresql.org/docs/9.5/static/upgrading.html). We’re going to go through the CentOS-specific aspects.

First, you need to install PostgreSQL 9.6 or whatever version you are upgrading to. It’s noteworthy that several of PostgreSQL versions can live together in peace on one CentOS instance. The installation process is well described on the [official website](https://www.postgresql.org/download/linux/redhat/). There you should find the latest *repository RPM* for your OS and the latest PG version, in our case of CentOS 7.2 x86\_64 and PG 9.6 it’s:

[root@database ~]# yum install https://yum.postgresql.org/9.6/redhat/rhel-7.3-x86\_64/pgdg-redhat96-9.6-3.noarch.rpm

Install PG:

[root@database ~]# yum install postgresql96-server postgresql96-contrib

Now PostgreSQL 9.6 is installed on your machine along with another PG version you have before. You can find them both at /usr/pgsql-9.6/ and /usr/pgsql-9.5/ (in case of 9.5).

Create Postgres 9.6 data directory:

[root@database ~]# /usr/pgsql-9.6/bin/postgresql96-setup initdb

Now you have two data directories, /var/lib/pgsql/9.6 and /var/lib/pgsql/9.5 in our case.

Let’s check the ability to upgrade your data from 9.5 to 9.6:

[root@database ~]# su postgres # pg\_upgrade cannot be run as root  
[postgres@database /root]$ cd ~  
[postgres@database ~]$ /usr/pgsql-9.6/bin/pg\_upgrade --old-bindir=/usr/pgsql-9.5/bin/ --new-bindir=/usr/pgsql-9.6/bin/ --old-datadir=/var/lib/pgsql/9.5/data/ --new-datadir=/var/lib/pgsql/9.6/data/ --check # check only, won't change any data

You should see “*\*Clusters are compatible\**” as a sign that all is ok and we are ready to move on. Perhaps you will see something like: “*Performing Consistency Checks on Old Live Server*”. That’s because your old version of the server is running, so you should shut it down before upgrading. Yes, that means downtime. Although zero-downtime upgrade is possible, it’s significantly more difficult to perform. If zero-downtime is important for you, consider [upgrading via replication](https://www.postgresql.org/docs/9.5/static/upgrading.html#UPGRADING-VIA-REPLICATION).

Stop the running server as root:

[root@database ~]# service postgresql-9.5 stop

Now back to the *postgres* user and run:

[postgres@database ~]$ /usr/pgsql-9.6/bin/pg\_upgrade --old-bindir=/usr/pgsql-9.5/bin/ --new-bindir=/usr/pgsql-9.6/bin/ --old-datadir=/var/lib/pgsql/9.5/data/ --new-datadir=/var/lib/pgsql/9.6/data/

The whole process takes a while. It’s about 5 minutes for a 50GB database on a good machine. If all goes well, you need to transfer all your settings from old pg\_hba.conf, postgresql.conf, and other configuration files to new ones.

Pay particular attention to *listen\_addresses* and *max\_connections* options of postgresql.conf as default values for these options are especially far from commonly used in production.

Sometimes you can simply copy these files:

[root@database ~]# cp /var/lib/pgsql/9.5/data/postgresql.conf /var/lib/pgsql/9.6/data/postgresql.conf  
[root@database ~]# cp /var/lib/pgsql/9.5/data/pg\_hba.conf /var/lib/pgsql/9.6/data/pg\_hba.conf

Sometimes you can’t or you shouldn’t. Especially when a new PG version introduced new options and/or refused old ones. Be careful on that. Anyway, be sure to save distribution versions of these files, so you can use them as a guide later.

Now hopefully you can start the new version of the server. Cross your fingers and run as root:

[root@database ~]# service postgresql-9.6 start

Enable the service to auto start on launch:

[root@database ~]# systemctl enable postgresql-9.6

The *pg\_upgrade* script creates two new scriptions in its working directory, *analyze\_new\_cluster.sh* and *delete\_old\_cluster.sh* Depending on the size of your database probably a good idea to run *analyze\_new\_cluster.sh*:

[postgres@database ~]$ ./analyze\_new\_cluster.sh

Check if all works well and if so, you can safely uninstall the old version of PG and remove the old data directory using *delete\_old\_cluster.sh*

Keep your software up to date. It is always easier to upgrade recent maintained versions rather than ancient fossils. Good luck.