#### 0. Set the path such that the older postgresql version is default.

[root@d\_server pbspro]# psql -V
psql (PostgreSQL) 9.2.23
[root@d\_server pbspro]# /usr/pgsql-9.3/bin/psql -V
psql (PostgreSQL) 9.3.23
[root@d\_server pbspro]# cat $PBS\_HOME/datastore/PG\_VERSION
9.2

#### 1. Stop PBS and verify if the dataservice has stopped

[root@d\_server pbspro]# /etc/init.d/pbs stop
Stopping PBS
Shutting server down with qterm.
PBS server - was pid: 1792
PBS sched - was pid: 1499
PBS comm - was pid: 1493
Waiting for shutdown to complete
[root@d\_server pbspro]# ps -elf | grep pbs
0 S root 2528 1 0 80 0 - 2264 pipe\_w 06:53 ? 00:00:00 grep --color=auto pbs
[root@d\_server pbspro]# ps -elf | grep post
0 S root 2530 1 0 80 0 - 2264 pipe\_w 06:53 ? 00:00:00 grep --color=auto post
[root@d\_server pbspro]#

#### 2. Change pg\_hba.conf in the existing DB to enable passwordless operations.

[root@d\_server pbspro]# cp $PBS\_HOME/datastore/pg\_hba.conf $PBS\_HOME/datastore/pg\_hba.conf.orig
[root@d\_server pbspro]# chown postgres $PBS\_HOME/datastore/pg\_hba.conf.orig
[root@d\_server pbspro]# sed 's/md5/trust/g' $PBS\_HOME/datastore/pg\_hba.conf > $PBS\_HOME/datastore/pg\_hba.conf.new
[root@d\_server pbspro]# mv -f $PBS\_HOME/datastore/pg\_hba.conf.new $PBS\_HOME/datastore/pg\_hba.conf
[root@d\_server pbspro]#

#### 3. Find the encoding and locale settings used in the existing cluster

[root@d\_server pbspro]# /opt/pbs/sbin/pbs\_dataservice start
Starting PBS Data Service..
[root@d\_server pbspro]# enc\_type=`psql -A -t -p 15007 -d pbs\_datastore -U postgres -c "select pg\_encoding\_to\_char(encoding) from pg\_database where datname = 'pbs\_datastore'"`
[root@d\_server pbspro]# echo $enc\_type
SQL\_ASCII
[root@d\_server pbspro]# lc\_collate=`psql -A -t -p 15007 -d pbs\_datastore -U postgres -c "SHOW LC\_COLLATE"`
[root@d\_server pbspro]# echo $lc\_collate
C
[root@d\_server pbspro]# lc\_ctype=`psql -A -t -p 15007 -d pbs\_datastore -U postgres -c "SHOW LC\_COLLATE"`
[root@d\_server pbspro]# echo $lc\_ctype
C
[root@d\_server pbspro]# /opt/pbs/sbin/pbs\_dataservice stop
Stopping PBS Data Service..
waiting for server to shut down.... done
server stopped
[root@d\_server pbspro]#

#### 4. Rename the datastore folder

[root@d\_server pbspro]# mv $PBS\_HOME/datastore/ $PBS\_HOME/datastore.old

#### 5. Create the new datastore directory, set ownership and permissions

mkdir -p $PBS\_HOME/datastore
chown postgres $PBS\_HOME/datastore
chmod 700 $PBS\_HOME/datastore
cd $PBS\_HOME//datastore

#### 6. Create the new PBS database cluster using the encoding and locale settings found in step 3.

[root@d\_server datastore]# su postgres -c "/usr/pgsql-9.3/bin/initdb -D /var/spool/pbs/datastore -U postgres -E SQL\_ASCII --lc-collate=C --lc-ctype=C"
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/spool/pbs/datastore ... ok
creating subdirectories ... ok
selecting default max\_connections ... 100
selecting default shared\_buffers ... 128MB
creating configuration files ... ok
creating template1 database in /var/spool/pbs/datastore/base/1 ... ok
initializing pg\_authid ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating collations ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
loading PL/pgSQL server-side language ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg\_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

/usr/pgsql-9.3/bin/postgres -D /var/spool/pbs/datastore
or
/usr/pgsql-9.3/bin/pg\_ctl -D /var/spool/pbs/datastore -l logfile start

[root@d\_server datastore]#

7. Change postgresql.conf in the new cluster to use settings below.
checkpoint\_segments = 20
port = 15007
listen\_addresses = '\*'
standard\_conforming\_strings = on
standard\_conforming\_strings = on
logging\_collector = on
log\_directory = 'pg\_log'
log\_filename = 'pbs\_dataservice\_log.%a'
log\_truncate\_on\_rotation = on
log\_rotation\_age = 1440
log\_line\_prefix = '%t'

#### 8. check if we can upgrade the cluster or not, using pg\_upgrade -c from the newer version of postgresql

[root@d\_server datastore]# su postgres -c "/usr/pgsql-9.3/bin/pg\_upgrade -d /var/spool/pbs/datastore.old/ -D /var/spool/pbs/datastore -b /usr/bin -B /usr/pgsql-9.3/bin -p 15007 -P 15007 -c"
Performing Consistency Checks
-----------------------------
Checking cluster versions ok
Checking database user is a superuser ok
Checking database connection settings ok
Checking for prepared transactions ok
Checking for reg\* system OID user data types ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for presence of required libraries ok
Checking database user is a superuser ok
Checking for prepared transactions ok

\*Clusters are compatible\*
[root@d\_server datastore]#

#### 9. If pg\_upgrade -c reports clusters are compatible, perform pg\_upgrade for real this time. If not, try resolving the incompatabilities reported. One such incompatibility is explained at the end of this document.

[root@d\_server datastore]# su postgres -c "/usr/pgsql-9.3/bin/pg\_upgrade -d /var/spool/pbs/datastore.old/ -D /var/spool/pbs/datastore -b /usr/bin -B /usr/pgsql-9.3/bin -p 15007 -P 15007"
Performing Consistency Checks
-----------------------------
Checking cluster versions ok
Checking database user is a superuser ok
Checking database connection settings ok
Checking for prepared transactions ok
Checking for reg\* system OID user data types ok
Checking for contrib/isn with bigint-passing mismatch ok
Creating dump of global objects ok
Creating dump of database schemas
ok
Checking for presence of required libraries ok
Checking database user is a superuser ok
Checking for prepared transactions ok

If pg\_upgrade fails after this point, you must re-initdb the
new cluster before continuing.

Performing Upgrade
------------------
Analyzing all rows in the new cluster ok
Freezing all rows on the new cluster ok
Deleting files from new pg\_clog ok
Copying old pg\_clog to new server ok
Setting next transaction ID and epoch for new cluster ok
Deleting files from new pg\_multixact/offsets ok
Setting oldest multixact ID on new cluster ok
Resetting WAL archives ok
Setting frozenxid and minmxid counters in new cluster ok
Restoring global objects in the new cluster ok
Adding support functions to new cluster ok
Restoring database schemas in the new cluster
ok
Setting minmxid counter in new cluster ok
Removing support functions from new cluster ok
Copying user relation files
ok
Setting next OID for new cluster ok
Sync data directory to disk ok
Creating script to analyze new cluster ok
Creating script to delete old cluster ok

Upgrade Complete
----------------
Optimizer statistics are not transferred by pg\_upgrade so,
once you start the new server, consider running:
analyze\_new\_cluster.sh

Running this script will delete the old cluster's data files:
delete\_old\_cluster.sh
[root@d\_server datastore]#

#### 10. Set the path to use newer postgresql version to be default.

[root@d\_server datastore]# export PATH=/usr/pgsql-9.3/bin/:$PATH
[root@d\_server datastore]#

#### 11. Start the PBS dataservice.

[root@d\_server datastore]# /opt/pbs/sbin/pbs\_dataservice start
Starting PBS Data Service..
[root@d\_server datastore]#

#### 12. Set PGPORT environment variable to use 15007 as the port.

[root@d\_server datastore]# export PGPORT=15007
[root@d\_server datastore]#

#### 13. Analyze the new cluster

[root@d\_server datastore]# su postgres -c "./analyze\_new\_cluster.sh"
This script will generate minimal optimizer statistics rapidly
so your system is usable, and then gather statistics twice more
with increasing accuracy. When it is done, your system will
have the default level of optimizer statistics.

If you have used ALTER TABLE to modify the statistics target for
any tables, you might want to remove them and restore them after
running this script because they will delay fast statistics generation.

If you would like default statistics as quickly as possible, cancel
this script and run:
"/usr/pgsql-9.3/bin/vacuumdb" --all --analyze-only

Generating minimal optimizer statistics (1 target)
--------------------------------------------------
vacuumdb: vacuuming database "pbs\_datastore"
vacuumdb: vacuuming database "postgres"
vacuumdb: vacuuming database "template1"

The server is now available with minimal optimizer statistics.
Query performance will be optimal once this script completes.

Generating medium optimizer statistics (10 targets)
---------------------------------------------------
vacuumdb: vacuuming database "pbs\_datastore"
vacuumdb: vacuuming database "postgres"
vacuumdb: vacuuming database "template1"

Generating default (full) optimizer statistics (100 targets?)
-------------------------------------------------------------
vacuumdb: vacuuming database "pbs\_datastore"
vacuumdb: vacuuming database "postgres"
vacuumdb: vacuuming database "template1"

Done
[root@d\_server datastore]#

#### 14. Stop the PBS dataservice.

[root@d\_server datastore]# /opt/pbs/sbin/pbs\_dataservice stop
Stopping PBS Data Service..
waiting for server to shut down..... done
server stopped
[root@d\_server datastore]#

#### 15. Reset the PBS datauser password.

[root@d\_server datastore]# /opt/pbs/sbin/pbs\_ds\_password -r
---> Updated user password
---> Success
[root@d\_server datastore]#

#### 16. Copy pg\_hba.conf from the old cluster to the new cluster and revoke the passwordless settings.

[root@d\_server datastore]# cp $PBS\_HOME/datastore.old/pg\_hba.conf $PBS\_HOME/datastore/pg\_hba.conf
[root@d\_server datastore]#

#### 17. Start PBS

[root@d\_server datastore]# /etc/init.d/pbs restart
Restarting PBS
Stopping PBS
Killing Server.
PBS server - was pid: 4572
PBS sched - was pid: 4064
PBS comm - was pid: 4049
Waiting for shutdown to complete
Starting PBS
/opt/pbs/sbin/pbs\_comm ready (pid=5500), Proxy Name:d\_server:17001, Threads:4
PBS comm
Creating usage database for fairshare.
PBS sched
Connecting to PBS dataservice....connected to PBS dataservice@d\_server
Licenses valid for 10000000 Floating hosts
PBS server
[root@d\_server datastore]#

#### 18. Verify the PG\_VERSION in the new datastore

[root@d\_server datastore]# cat $PBS\_HOME/datastore/PG\_VERSION
9.3
[root@d\_server datastore]#

19. If needed, remove the older postgresql.
[root@d\_server datastore]# yum remove postgresql-server-9.2.23-3.el7\_4.x86\_64 postgresql-9.2.23-3.el7\_4.x86\_64 postgresql-libs-9.2.23-3.el7\_4.x86\_64
Loaded plugins: fastestmirror, ovl

...

...

...

Removed:
postgresql.x86\_64 0:9.2.23-3.el7\_4 postgresql-libs.x86\_64 0:9.2.23-3.el7\_4 postgresql-server.x86\_64 0:9.2.23-3.el7\_4

Complete!

20. Restart PBS
[root@d\_server datastore]# psql -V
psql (PostgreSQL) 9.3.23
[root@d\_server datastore]# /etc/init.d/pbs restart
Restarting PBS
Stopping PBS
Shutting server down with qterm.
PBS server - was pid: 5826
PBS sched - was pid: 5515
PBS comm - was pid: 5500
Waiting for shutdown to complete
Starting PBS
/opt/pbs/sbin/pbs\_comm ready (pid=6189), Proxy Name:d\_server:17001, Threads:4
PBS comm
Creating usage database for fairshare.
PBS sched
Connecting to PBS dataservice....connected to PBS dataservice@d\_server
Licenses valid for 10000000 Floating hosts
PBS server
[root@d\_server datastore]#

#### 21. Remove the old cluster.

[root@d\_server datastore]# ./delete\_old\_cluster.sh
[root@d\_server datastore]#

#### Miscellaneous

#### Resolution of the issue reported by pg\_upgrade while upgrading from 9.2.23 to 9.3.23 and it's resolution

[root@d\_server datastore]# su postgres -c "/usr/pgsql-9.3/bin/pg\_upgrade -d /var/spool/pbs/datastore.old/ -D /var/spool/pbs/datastore -b /usr/bin -B /usr/pgsql-9.3/bin -p 15007 -P 15007 -c"
Performing Consistency Checks
-----------------------------
Checking cluster versions ok

\*failure\*
Consult the last few lines of "pg\_upgrade\_server.log" for
the probable cause of the failure.

connection to database failed: could not connect to server: No such file or directory
Is the server running locally and accepting
connections on Unix domain socket "/var/spool/pbs/datastore/.s.PGSQL.15007"?

could not connect to old postmaster started with the command:
"/usr/bin/pg\_ctl" -w -l "pg\_upgrade\_server.log" -D "/var/spool/pbs/datastore.old/" -o "-p 15007 -b -c listen\_addresses='' -c unix\_socket\_permissions=0700 -c unix\_socket\_directory='/var/spool/pbs/datastore'" start
Failure, exiting
[root@d\_server datastore]#

[root@d\_server datastore]# mv -f /usr/bin/pg\_ctl{,-orig}
[root@d\_server datastore]# echo '#!/bin/bash' > /usr/bin/pg\_ctl
[root@d\_server datastore]# echo '"$0"-orig "${@/unix\_socket\_directory/unix\_socket\_directories}"' >> /usr/bin/pg\_ctl
[root@d\_server datastore]# chmod +x /usr/bin/pg\_ctl
[root@d\_server datastore]# su postgres -c "/usr/pgsql-9.3/bin/pg\_upgrade -d /var/spool/pbs/datastore.old/ -D /var/spool/pbs/datastore -b /usr/bin -B /usr/pgsql-9.3/bin -p 15007 -P 15007 -c"
Performing Consistency Checks
-----------------------------
Checking cluster versions ok
Checking database user is a superuser ok
Checking database connection settings ok
Checking for prepared transactions ok
Checking for reg\* system OID user data types ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for presence of required libraries ok
Checking database user is a superuser ok
Checking for prepared transactions ok

\*Clusters are compatible\*
[root@d\_server datastore]#