

## Installation

### Download essential rpm's (RHEL):

```
postgresql-8.4.2-1PGDG.rhel5.x86_64.rpm
postgresql-libs-8.4.2-1PGDG.rhel5.x86_64.rpm
postgresql-devel-8.4.2-1PGDG.rhel5.x86_64.rpm
postgresql-server-8.4.2-1PGDG.rhel5.x86_64.rpm
postgresql-docs-8.4.2-1PGDG.rhel5.x86_64.rpm
```

### Install

```
rpm -ivh postgres*
```

### Initialize database cluster

```
# mkdir /var/lib/pgsql/data
# chown postgres /var/lib/pgsql/data
# su - postgres
$ initdb -D /var/lib/pgsql/data
# service postgresql start
$ psql
```

```
# select * from pg_database;
```

Tip: As an alternative to the -D option, you can set the environment variable PGDATA.

## Backup & Recovery

3 types of backup approaches

- \* SQL dump
- \* File system level backup
- \* Continuous archiving

### Dumps (exp/imp)

Backup:

```
pg_dump mydatabase > mydump.out => exp
pg_dumpall > myfulldump.out => exp full=Y
```

Restore:

```
pg_restore -d mydatabase mydump.out
psql mydatabase < mydump.out
psql -f myfulldump.out postgres
```

Copy database to another node:

```
pg_dump -h host1 mydatabase | psql -h host2 mydatabase
```

### Filesystem backup:

```
tar -cf backup.tar /var/lib/pgsql/data
! database server have to shutdown cleanly!
```

### Taking advantage of WAL logging:

- WAL Logs stored by default in \$PGDATA/pg\_xlog
- database can be restored by "replaying" transaction logs
- hot standby concept can be achieved with WAL logs
- do not try to combine WAL logs with pg\_dump files for restoring in another location, FS dump is needed to be able to recover WAL log

### Standalone hot backup:

Ensure parameters are defined and directories exist:

```
archive_mode=on
archive_command = 'test ! -f
/var/lib/pgsql/backup_in_progress || cp -i %p
/var/lib/pgsql/archive/%f < /dev/null'
```

```
touch /var/lib/pgsql/backup_in_progress
psql -c "select pg_start_backup('hot_backup');"
tar -cf /var/lib/pgsql/backups/backup.tar /var/lib/pgsql/data/
psql -c "select pg_stop_backup();"
rm /var/lib/pgsql/backup_in_progress
tar -rf /var/lib/pgsql/backup.tar /var/lib/pgsql/archive/
```

## START/STOP Postgresql

### Start:

```
#su -c 'pg_ctl -w start -D /var/lib/pgsql/data -l /var/lib/pgsql/pgstartup.log' postgres
on RHEL: # service postgresql start
$ pg_ctl -w start -D /var/lib/pgsql/data -l /var/lib/pgsql/pgstartup.log
```

### Stop:

```
# su -c 'pg_ctl -w stop -D /var/lib/pgsql/data -m fast' postgres
on RHEL: # service postgresql stop
$ pg_ctl -w stop -D /var/lib/pgsql/data -m smart => shutdown normal
$ pg_ctl -w stop -D /var/lib/pgsql/data -m fast => shutdown immediate
$ pg_ctl -w stop -D /var/lib/pgsql/data -m immediate => shutdown abort
```

### Status:

```
$ pg_ctl status -D /var/lib/pgsql/data
$ ps -ef | grep postgres | grep -v grep
```

## Connecting to database

### Local connections as postgres OS user

```
psql [dbname] [[username]] => sqlplus / as sysdba
postgres=# \connect template1 postgres => connect postgres@template1
```

### Remote connections

```
psql -h 10.0.0.20 -p 5432 -U myuser -d mydatabase => sqlplus myuser@mydatabase
psql -h 10.0.0.20 -p 5432 -l => list all databases listening on the host/port, may need
to add entry to pg_hba.conf/pg_ident.conf
Default DB listen port - 5432
```

## Important file locations

### Configuration files

```
$PGDATA - on RHEL /var/lib/pgsql/data - location of the database
$PGDATA/postgresql.conf => init.ora
$PGDATA/pg_hba.conf => (sort of sqlnet.ora)
$PGDATA/pg_ident.conf => OS/DB user mapping file
$PGDATA/pg_xlog => log_archive_dest
```

### Server Logfiles

```
$PGDATA/pg_log/postgresql-%a.log
/var/lib/pgsql/pgstartup.log
```

### Directories

```
Executables - /usr/bin
Libraries - /usr/lib
Documentation - /usr/share/doc/postgresqlx.y.z
Data - /var/lib/pgsql/data
Backup area - /var/lib/pgsql/backup
Templates - /usr/share/pgsql
Procedural Languages - /usr/lib/pgsql
Development Headers - /usr/include/pgsql
Other shared data - /usr/share/pgsql
```

## Important Init parameters

### File locations

```
show config_file; => show parameters spfile;
show all; => show parameters;
show data_directory; => show db file dir location
show hba_file;
show ident_file;
show log_directory;
show log_destination;
show external_pid_file;
```

### Memory/resources

```
show effective_cache_size;
show shared_buffers;
show vacuum_mem;
show wal_buffers;
```

### Server settings

```
show listen_addresses;
show port;
show server_encoding;
```

## Performance

```
pg_stat_% => v$views
Get explain plan for query
EXPLAIN ANALYZE SELECT *
FROM MY_TABLE;
Update stats on table
ANALYZE MY_TABLE;
Vacuum table
VACUUM MY_TABLE;
Vacuum & Analyze whole DB
VACUUM ANALYZE;
Analyze - update stats
Vacuum - housekeep after
heavy insert/delete ops
```

Pythian  
love your data