pg\_dumpall -p port > $backup\_location/dumpall.sql

plain format:

pg\_dump -p port -U user\_name db\_name > $backup\_location/dump\_postgres\_db.sql

Compressed Format:

pg\_dump -p port -U user\_name dbname -Fc -f $backup\_location/dump.dmp

Where -p - port,   
 -U – user  
 -Fc - Format Compressed (you can use tar by using -Ft)  
 -f - dumpfile.

* To Take the dump of a table/sequence

pg\_dump -p port -U user\_name -t table\_name db\_name > $backup\_location/dump\_test\_table.sql

Where -t - tablename/sequencename

#!/bin/bash

set -e

psql -U postgres -q "SELECT pg\_start\_backup('foo')" # ‘foo’ is a label which identifies this backup.

tar cfz /backup/$(date +%Y%m%d).tar.gz /var/lib/pgsql/data

psql -U postgres -q "SELECT pg\_stop\_backup()"

#!/bin/bash

set -e

pg\_basebackup -D /backup/$(date +%Y%m%d) -Ft -X fetch

local replication postgres peer

$ pg\_basebackup -x -D /path/to/backupdir

pg\_receivewal -D /log/archive -h master -S backup

#!/bin/bash

find /var/backups/basebackup -type f -mtime +30 -print0 | xargs -0 -r /bin/rm

find /var/backups/wal -type f -mtime +7 -print0 | xargs -0 -r /bin/rm

# restoring a custom format file  
$ pg\_restore -U db\_user -d db\_name\_new -v -1 db\_name.dmp

# restore a single table from the dump  
$ pg\_restore -U db\_user -d db\_name\_new --table=mytable -v -1 db\_name.dmp

# restore a single function from the dump  
$ pg\_restore -U db\_user -d db\_name\_new --function=myfunc -v -1 db\_name.dmp

$ pg\_ctl stop postgresql

$ sudo rm -rf /path/to/data/directory/\*

$ tar -xvC /path/to/data/directory -f /path/to/dumpdir/base.tar.gz

$ pg\_ctl start postgresql

restore\_command = 'cp /tmp/demo/archive/%f "%p"'

pg\_ctl -D /restored/data/directory start