PREAMBLE

This guide is valid only for Synology with Intel inside X64 CPU.
My Synology model is DS1618+ with Intel Atom 3538;
6Gb RAM
1xSSD 120Gb used for index with PostgreSQL (volume1)
5x12Tb NAS HDD for studies images (volume2)
DSM 6.2.1-23284Update4
IP: 192.168.10.55
Install third part software : nano and midnight commander
For edit any files in windows use   NOTEPAD++

INSTALL ORTHANC SERVER

1. Install PostgreSQL 9.6.12 (including PostgreSQL contrib)

By default Synology come with PostgreSQL 9.3.22 (port 5432) , this is too old version for running Orthanc 1.5.6 and plugin-postgresql 3.2.

You will not be able to upgrade PostgreSQL 9.3.22 to PostgreSQL 9.6.12 because DSM use version 9.3.22 and an upgrade will make DSM unstable or defective; some application inside of DSM use PostgreSQL - for example "mediaserver".

So you have to install a new instance of PostgreSQL in parallel with the existing one (port 5433).

1.1 Compile a version of PostgreSQL - in this case 9.6.12

You will need to compile on another machine (I used Debian 9 for compile)
Manually create a folder

```
mkdir -p /home/share/build_dir
```

Download source:
https://ftp.postgresql.org/pub/source/v9.6.12/postgresql-9.6.12.tar.gz

```
cd /home/share/
wget
https://ftp.postgresql.org/pub/source/v9.6.12/postgresql-9.6.12.tar.gz
gunzip postgresql-9.6.12.tar.gz
tar xf postgresql-9.6.12.tar
```

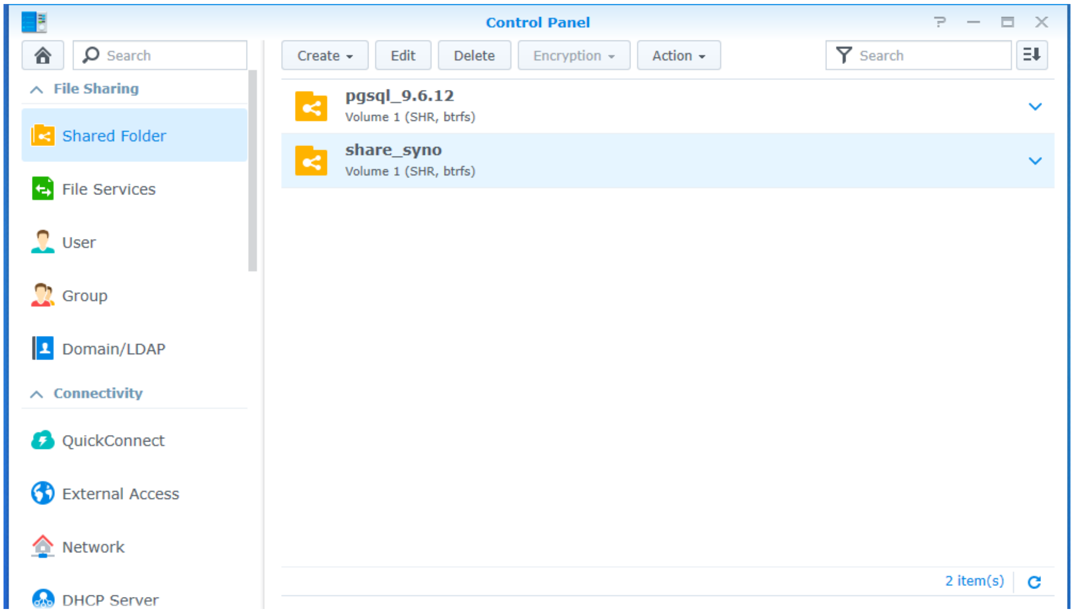
Will result a folder:`/home/share/postgresql-9.6.12`

```
cd /home/share/build_dir
root@debian:/home/share/build_dir#/home/share/postgresql-9.6.12/configur
e --prefix=/home/share/build_dir/
make world
make install-world
```

1.2 Copy compiled PostgreSQL 9.6.12 to Synology

From the web interface of synology you will need to create 2 folders:
- pgsql_9.6.12 (on volume1)
- share_syno (on volume1)

For both of them admin have read/write rights.



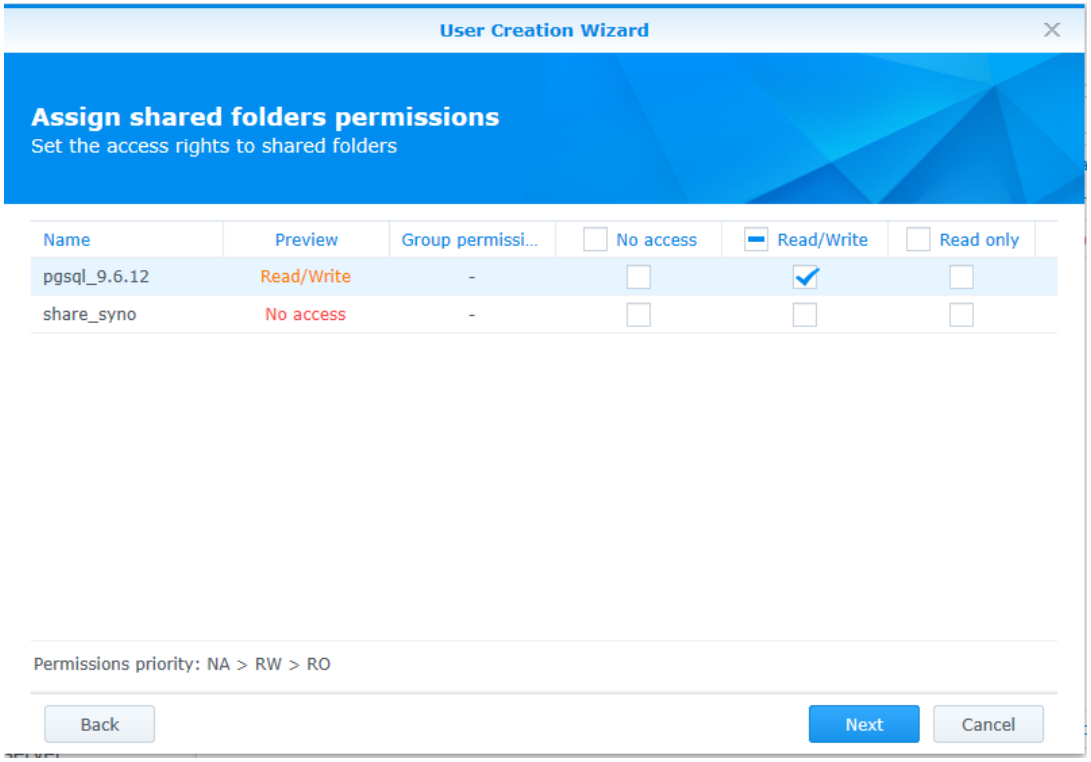
pgsql_9.6.12
Volume 1 (SHR, btrfs)

share_syno
Volume 1 (SHR, btrfs)

```
root@debian:~# scp -r /home/share/build_dir/ admin@192.168.10.55:/volume1/share_syno
```

1.3 Install PostgreSQL 9.6.12 on Synology

From Synology web interface create user “`postgres2`” with read/write rights to the `pgsql_9.6.12` folder



User Creation Wizard

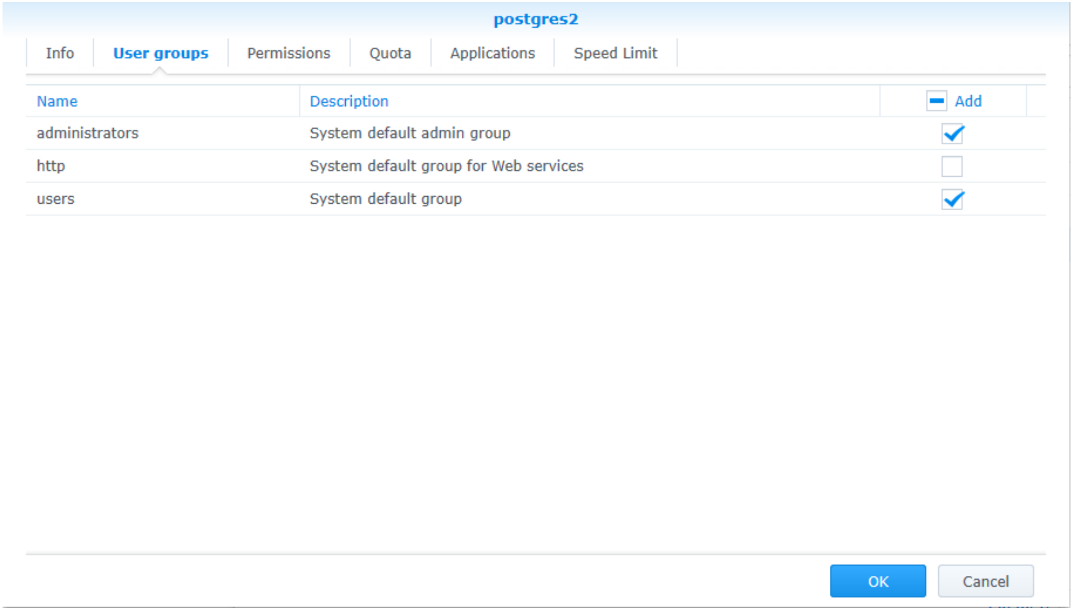
Assign shared folders permissions

Set the access rights to shared folders

Name	Preview	Group permissi...	□ No access	▬ Read/Write	□ Read only
pgsql_9.6.12	Read/Write	-	□	☑	□
share_syno	No access	-	□	□	□

Permissions priority: NA > RW > RO

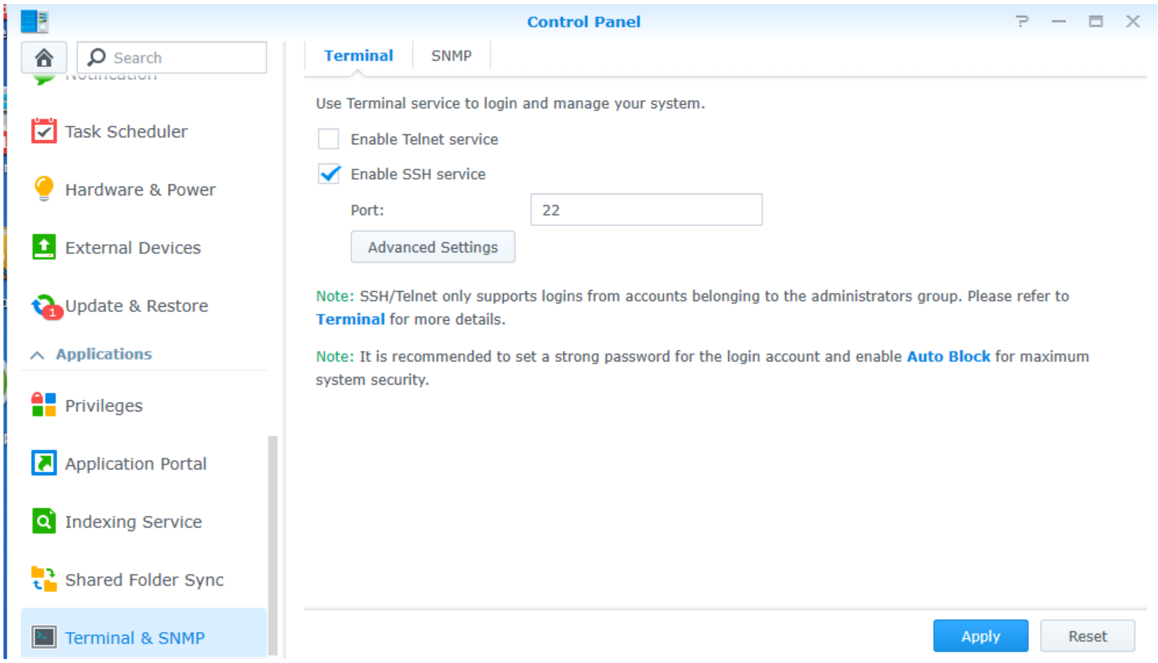
And check - “System default admin group”



postgres2

Name	Description	▬ Add
administrators	System default admin group	☑
http	System default group for Web services	□
users	System default group	☑

- activate SSH on synology



Use Terminal service to login and manage your system.

□ Enable Telnet service

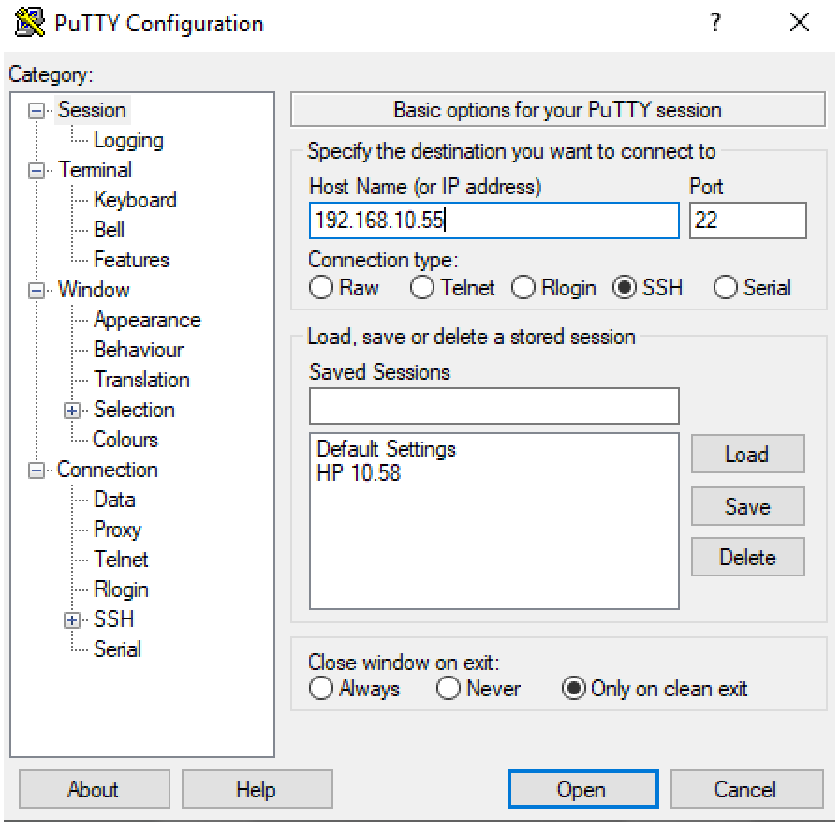
☑ Enable SSH service

Port: 22

Note: SSH/Telnet only supports logins from accounts belonging to the administrators group. Please refer to **Terminal** for more details.

Note: It is recommended to set a strong password for the login account and enable **Auto Block** for maximum system security.

Use PUTTY for SSH to Synology



Specify the destination you want to connect to

Host Name (or IP address): 192.168.10.55

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

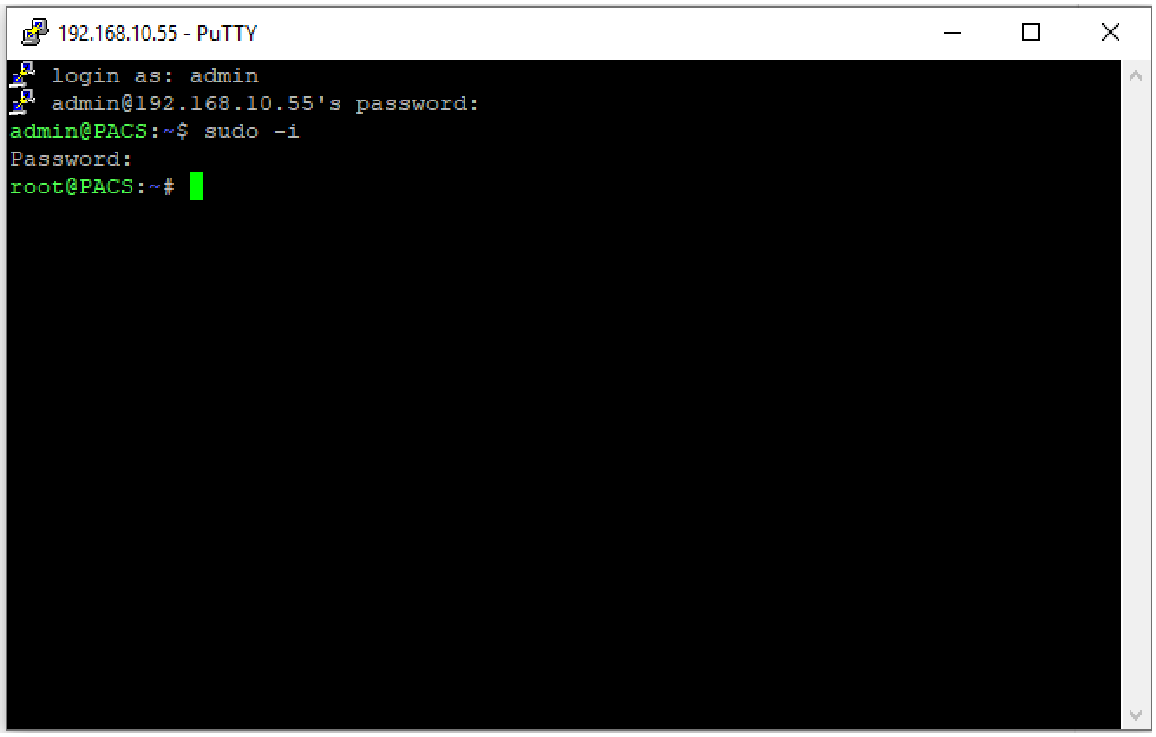
Load, save or delete a stored session

Saved Sessions

Default Settings
HP 10.58

Close window on exit: ○ Always ○ Never ● Only on clean exit

Login as root



```
login as: admin
admin@192.168.10.55's password:
admin@PACS:~$ sudo -i
Password:
root@PACS:~#
```

```
root@PACS:~# mkdir -p /volume1/pgsql_9.6.12/data

root@PACS:~# chown postgres2 /volume1/pgsql_9.6.12/data -R
```

Edit file: `/etc/passwd`

```
root@PACS:~# nano /etc/passwd
```

Look for :

```
postgres2:x:1026:100::/var/services/homes/postgres2:/sbin/nologin
```

and change in:

```
postgres2:x:1026:100::/volume1/pgsql_9.6.12/data:/bin/sh
```

```
reboot
```

After restart check again this file to be sure about saved changes.

INITDB

Exit from root

```
root@PACS:~# exit
logout
admin@PACS:~$ cd /volume1/pgsql_9.6.12/
admin@PACS:/volume1/pgsql_9.6.12$
sudo -u postgres2 bin/initdb -D /volume1/pgsql_9.6.12/data/
```

Will get a response:

```
admin@PACS:/volume1/pgsql_9.6.12$ sudo -u postgres2 bin/initdb -D
/volume1/pgsql_9.6.12/data/
The files belonging to this database system will be owned by user "postgres2".
This user must also own the server process.
The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".
Data page checksums are disabled.
fixing permissions on existing directory /volume1/pgsql_9.6.12/data ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... 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:
   bin/pg_ctl -D /volume1/pgsql_9.6.12/data/ -l logfile start
```

1.4 Start PostgreSQL

Create a file:`start.sh`

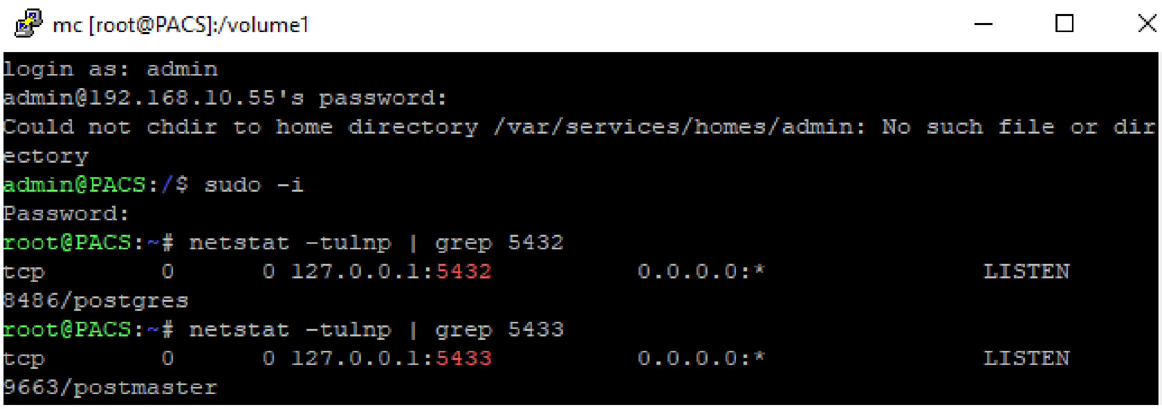
root@PACS:~# nano /volume1/pgsql_9.6.12/data/start.sh

(THE CONTENT OF THE start.sh FILE IS AT THE END OF THIS GUIDE) and run this file

```
cd /volume1/pgsql_9.6.12/data
./start.sh start
```

Check status of the 5432 and 5433 ports

```
netstat -tulnp | grep 5432
netstat -tulnp | grep 5433
```



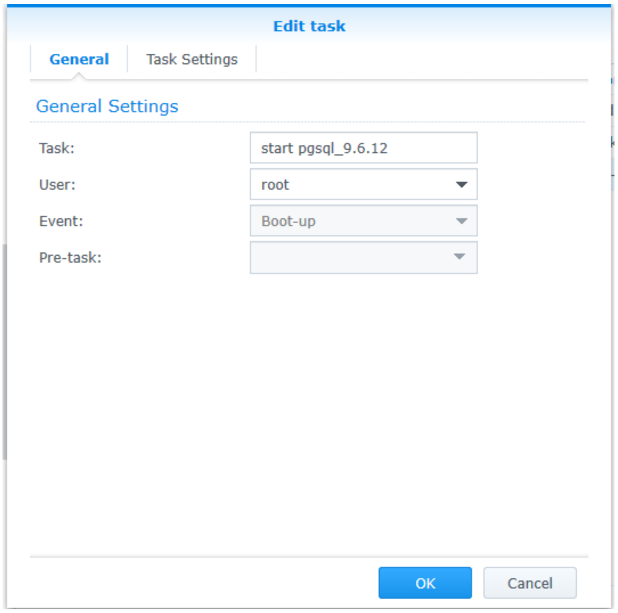
```
login as: admin
admin@192.168.10.55's password:
Could not chdir to home directory /var/services/homes/admin: No such file or dir
ectory
admin@PACS:/$ sudo -i
Password:
root@PACS:~# netstat -tulnp | grep 5432
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN
8486/postgres
root@PACS:~# netstat -tulnp | grep 5433
tcp        0      0 127.0.0.1:5433          0.0.0.0:*               LISTEN
9663/postmaster
```

Automatic start PostgreSQL after reboot

In Task Scheduler :

Create -> Triggered Task -> User-defined script

enable at boot-up



General Settings

Task: start pgsql_9.6.12

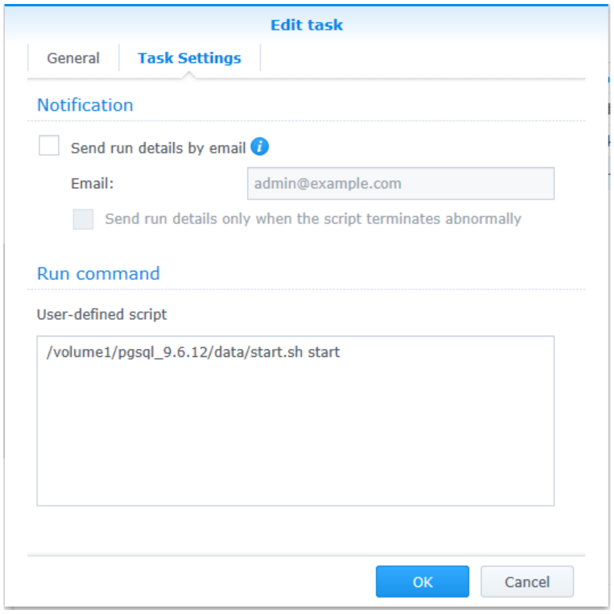
User: root

Event: Boot-up

Pre-task:

task setting -> Run command

/volume1/pgsql_9.6.12/data/start.sh start



Notification

□ Send run details by email

Email: admin@example.com

□ Send run details only when the script terminates abnormally

Run command

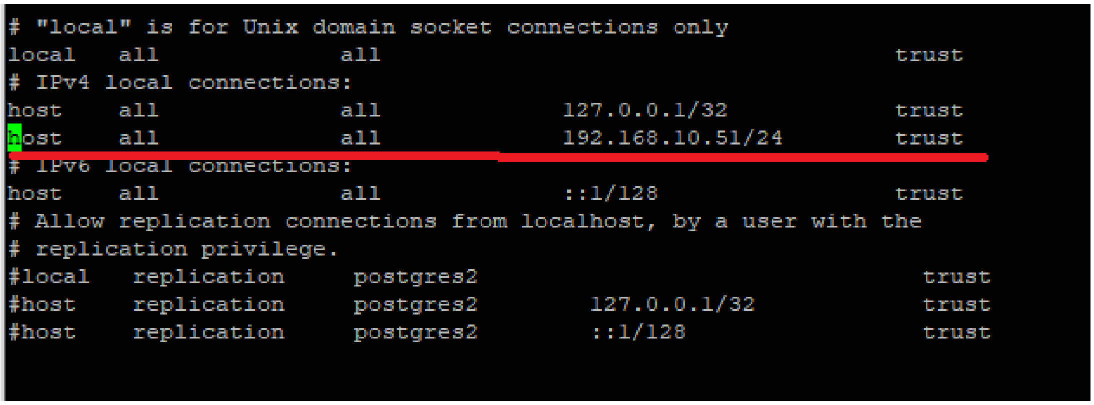
User-defined script

```
/volume1/pgsql_9.6.12/data/start.sh start
```

2.Manage PostgreSQL with PgAdmin3

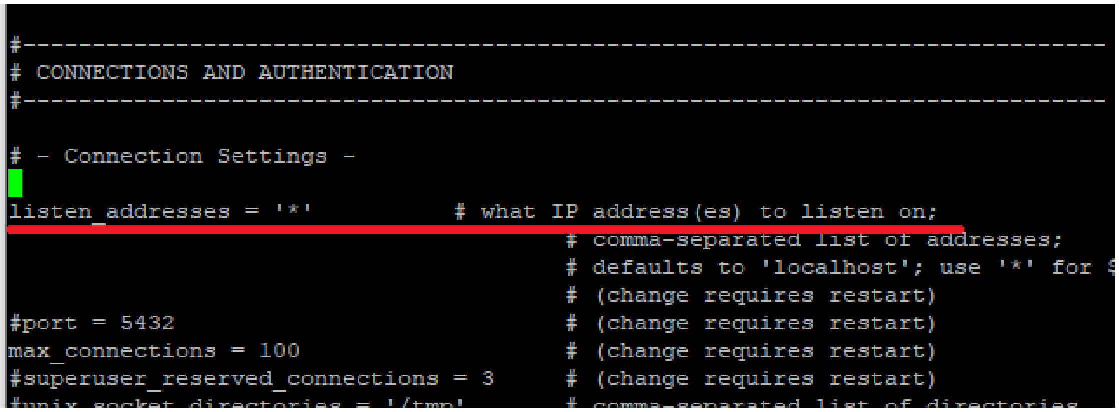
-download PgAdmin3 and install on your computer (in my case IP: 192.168.10.51)
- edit this two files

```
root@PACS:~# nano /volume1/pgsql_9.6.12/data/pg_hba.conf
```



```
# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
host    all             all             192.168.10.51/24        trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres2                               trust
#host    replication     postgres2        127.0.0.1/32            trust
#host    replication     postgres2        ::1/128                 trust
```

```
root@PACS:~# nano /volume1/pgsql_9.6.12/data/postgresql.conf
```



```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for
                                        # (change requires restart)
#port = 5432                            # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/tmp'       # comma-separated list of directories
```

After start PgAdmin3 you will need to crate a server with your Synology IP adress and login on port 5433.
Manually create
ORTHANC_DB
ORTHANC_USER
ORTHANC_PASS
and define this in the server_config file
Install Orthanc 1.3.0 from synology and stop this package after install.
Update Orthanc to 1.5.6 and pugin-postgresql index adapt server-config file to new configuration and start package again.

3.THE CONTENT OF THE start.sh file

```
#!/bin/sh

# chkconfig: 2345 98 02
# description: PostgreSQL RDBMS

# This is an example of a start/stop script for SysV-style init, such
# as is used on Linux systems.  You should edit some of the variables
# and maybe the 'echo' commands.
#
# Place this file at /etc/init.d/postgresql (or
# /etc/rc.d/init.d/postgresql) and make symlinks to
#   /etc/rc.d/rc0.d/K02postgresql
#   /etc/rc.d/rc1.d/K02postgresql
#   /etc/rc.d/rc2.d/K02postgresql
#   /etc/rc.d/rc3.d/S98postgresql
#   /etc/rc.d/rc4.d/S98postgresql
#   /etc/rc.d/rc5.d/S98postgresql
# Or, if you have chkconfig, simply:
# chkconfig --add postgresql
#
# Proper init scripts on Linux systems normally require setting lock
# and pid files under /var/run as well as reacting to network
# settings, so you should treat this with care.

# Original author:  Ryan Kirkpatrick <pgsql@rkirkpat.net>

# contrib/start-scripts/linux

## EDIT FROM HERE
# change the port
PGPORT=5433
# Installation prefix
# change prefix
prefix=/volume1/pgsql_9.6.12
# add the correct library path and then edit the
# case statement below to include it for the su -c command
LD_LIBRARY_PATH=$prefix/lib

# Data directory
# change the data dir
PGDATA=$prefix/data

# Who to run the postmaster as, usually "postgres".  (NOT "root")
# change the user
PGUSER=postgres2

# Where to keep a log file
PGLOG="$PGDATA/serverlog"

# It's often a good idea to protect the postmaster from being killed by the
# OOM killer (which will tend to preferentially kill the postmaster because
# of the way it accounts for shared memory).  Setting the OOM_SCORE_ADJ value
# to -1000 will disable OOM kill altogether.  If you enable this, you probably
# want to compile PostgreSQL with "-DLINUX_OOM_SCORE_ADJ=0", so that
# individual backends can still be killed by the OOM killer.
#OOM_SCORE_ADJ=-1000
# Older Linux kernels may not have /proc/self/oom_score_adj, but instead
# /proc/self/oom_adj, which works similarly except the disable value is -17.
# For such a system, enable this and compile with "-DLINUX_OOM_ADJ=0".
#OOM_ADJ=-17

## STOP EDITING HERE

# The path that is to be used for the script
PATH=$prefix/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# What to use to start up the postmaster.  (If you want the script to wait
# until the server has started, you could use "pg_ctl start -w" here.
# But without -w, pg_ctl adds no value.)
DAEMON="$prefix/bin/postmaster"

# What to use to shut down the postmaster
PGCTL="$prefix/bin/pg_ctl"

set -e

# Only start if we can find the postmaster.
test -x $DAEMON ||
{
    echo "$DAEMON not found"
    if [ "$1" = "stop" ]
    then exit 0
    else exit 5
    fi
}


# Parse command line parameters.
case $1 in
  start)
    echo -n "Starting PostgreSQL: "
    test x"$OOM_SCORE_ADJ" != x && echo "$OOM_SCORE_ADJ" > /proc/self/oom_score_adj
    test x"$OOM_ADJ" != x && echo "$OOM_ADJ" > /proc/self/oom_adj
    su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $DAEMON -D '$PGDATA' -p $PGPORT &"
>>$PGLOG 2>&1
    echo "ok"
    ;;
  stop)
    echo -n "Stopping PostgreSQL: "
    su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $PGCTL stop -D '$PGDATA' -s -m fast"
    echo "ok"
    ;;
  restart)
    echo -n "Restarting PostgreSQL: "
    su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $PGCTL stop -D '$PGDATA' -s -m fast -w"
    test x"$OOM_SCORE_ADJ" != x && echo "$OOM_SCORE_ADJ" > /proc/self/oom_score_adj
    test x"$OOM_ADJ" != x && echo "$OOM_ADJ" > /proc/self/oom_adj
    su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $DAEMON -D '$PGDATA' -p $PGPORT &"
>>$PGLOG 2>&1
    echo "ok"
    ;;
  reload)
      echo -n "Reload PostgreSQL: "
      su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $PGCTL reload -D '$PGDATA' -s"
      echo "ok"
      ;;
  status)
    su - $PGUSER -c "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $PGCTL status -D '$PGDATA'"
    ;;
  *)
    # Print help
    echo "Usage: $0 {start|stop|restart|reload|status}" 1>&2
    exit 1
    ;;
esac

exit 0
```

4.THE CONTENT OF THE server_config FILE

```
{
        "Name" : "SUP",
        "StorageDirectory" : "/volume2/orthanc/OrthancStorage",
        "HttpPort" : 8042,
        "DicomAet" : "ORTHANC",
        "DicomPort" : 4242,
        "RemoteAccessAllowed" : true,
        "SslEnabled" : false,
        "AuthenticationEnabled" : true,
        "RegisteredUsers" : {
                "orthanc" : "orthanc"
        },
"PostgreSQL" : {
                "EnableIndex" : true,
                "Port" : 5433,
                "Host" : "localhost",
                "Database" : "orthanc_db",
                "Username" : "orthanc_user",
                "Password" : "orthanc_pass"
        },
        ],
        "Plugins" : [
                "/var/packages/Orthanc/target/usr/share/orthanc/plugins/"
        ]
}
```

Thanks for their information and help:
https://sondregronas.com/managing-postgresql-on-a-synology-server/

https://toggen.com.au/blog/it-tips/synology-ds415-creating-a-pycrypto-extensioned-postgres-instance-for-postpooks