Replica Postgres V9.5 master/slave

A. BUSSANI

Approved by: ……

Dr. Paola Del Negro

INDICE:

Abstract

Avendo la necessità di:

- creare un backup dei dati (e degli utenti del db)
- alleggerire il server oceano dalle chiamate in lettura da parte del server web nettuno,

si è reso necessario creare una replica mater/slave del database Postgres installato su oceano.
Si è scelto di installare il server di replica su un server virtuale nominato lagunamarano.

1. Nuova installazione postgresql sul server slave (lagunamarano)

Documentazione disponibile online;
http://tecadmin.net/install-postgresql-server-on-ubuntu/

Aggiungere i repository per la versione scelta (9.5)

```
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release
-cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
$ wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key
add -
```

Installazione

```
$ sudo apt-get update
$ sudo apt-get install postgresql postgresql-contrib
```

Test di connessione

Dopo l'installazione del server PostgreSQL, per default viene creata l'utenza postgres, con lo stesso nome viene creata una role e un utente di sistema
Quindi usando l'utenza postgres si tenta di connettersi al server:

```
$ sudo su - postgres
$ psql
```

Essendo loggati si controlla le informazioni di connessione:

```
postgres-# \conninfo
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in
"/var/run/postgresql" at port "5432".
```

Per disconnettersi:

```
postgres-# \q
```

2. Controllo servizio (ubuntu)

service --status-all

```
[ + ]  acpid
[ + ]  apache2
[ - ]  apparmor
[ ? ]  apport
[ + ]  atd
[ ? ]  console-setup
[ - ]  cron
[ - ]  dbus
[ ? ]  dns-clean
[ - ]  friendly-recovery
[ - ]  grub-common
[ ? ]  irqbalance
[ ? ]  killprocs
[ ? ]  kmod
[ ? ]  mysql
[ ? ]  networking
[ ? ]  ondemand
[ + ]  openvpn
[ + ]  postgresql
[ ? ]  pppd-dns
[ + ]  procps
[ ? ]  rc.local
[ + ]  resolvconf
[ - ]  rsync
[ + ]  rsyslog
[ ? ]  screen-cleanup
[ ? ]  sendsigs
[ - ]  ssh
[ - ]  sudo
[ - ]  sysstat
[ + ]  udev
[ ? ]  umountfs
[ ? ]  umountnfs.sh
[ ? ]  umountroot
[ - ]  unattended-upgrades
[ - ]  urandom
[ - ]  x11-common
```

```
/etc/init.d/postgresql status
9.5/main (port 5432): online
```

3. Rimozione postgresql (ubuntu)

Se ci dovessero essere dei problemi, la disinstallazione si puo' fare cosi':
Documentazione online:

https://help.ubuntu.com/community/PostgreSQL

```
apt-get --purge remove postgresql\*
apt-get purge postgr*
apt-get autoremove postgr*
```

ed inoltre:

```
rm -r /etc/postgresql/
rm -r /etc/postgresql-common/
rm -r /var/lib/postgresql/
userdel -r postgres
groupdel postgres
```

4. Copia 9.3 to 9.5 porta 5432→5433 – server master

Avendo installato la nuova versione 9.5 ed essendo installata la versione 9.3 sul master, si rende necessario aggiornare la versione del server postgres su oceano.
Si provvede quindi ad installare la versione 9.5 aggiornando i repository e scaricando la versione opportuna, provvedendo a variare la porta di ascolto del server da 5432 a 5433 (in modo da poter eseguire contemporaneamente i due server – vecchio e nuovo – per testarne le funzionalità).

Per l'aggiornamento dalla version 9.3 alla 9.5:

```
[postgres@oceano 9.5]$ /usr/pgsql-9.5/bin/pg_upgrade -b /usr/pgsql-9.3/bin/
-B /usr/pgsql-9.5/bin/ -d /usr2/pgsql/9.3/data -D /usr2/pgsql/9.5/
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  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 invalid "line" user columns                    ok
Creating dump of global objects                             ok
Creating dump of database schemas
                                                            ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  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
Copying old pg_multixact/offsets to new server              ok
Deleting files from new pg_multixact/members                ok
```

```
Copying old pg_multixact/members to new server              ok
Setting next multixact ID and offset for new cluster        ok
Resetting WAL archives                                      ok
Setting frozenxid and minmxid counters in new cluster       ok
Restoring global objects in the new cluster                 ok
Restoring database schemas in the 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
```

Per la chiusura/riavvio del servizio:

```
/etc/init.d/postgresql-9.3 stop
/etc/init.d/postgresql-9.5 stop
/etc/init.d/postgresql-9.5 start
```

5. Test uso porta postgres

```
nmap -v -p 5432 127.0.0.1

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2016-07-13 12:22 CEST
Initiating SYN Stealth Scan against oceano.ogs.trieste.it (127.0.0.1) [1 port]
at 12:22
Discovered open port 5432/tcp on 127.0.0.1
The SYN Stealth Scan took 0.00s to scan 1 total ports.
Host oceano.ogs.trieste.it (127.0.0.1) appears to be up ... good.
Interesting ports on oceano.ogs.trieste.it (127.0.0.1):
PORT     STATE SERVICE
5432/tcp open  postgres

Nmap finished: 1 IP address (1 host up) scanned in 0.010 seconds
               Raw packets sent: 1 (44B) | Rcvd: 2 (88B)
```

```
[root@oceano data]# nmap -v -p 5433 127.0.0.1

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2016-07-13 12:22 CEST
Initiating SYN Stealth Scan against oceano.ogs.trieste.it (127.0.0.1) [1 port]
at 12:22
Discovered open port 5433/tcp on 127.0.0.1
The SYN Stealth Scan took 0.00s to scan 1 total ports.
Host oceano.ogs.trieste.it (127.0.0.1) appears to be up ... good.
Interesting ports on oceano.ogs.trieste.it (127.0.0.1):
PORT     STATE SERVICE
```

```
5433/tcp open  unknown

Nmap finished: 1 IP address (1 host up) scanned in 0.010 seconds
               Raw packets sent: 1 (44B) | Rcvd: 2 (88B)
```

6. Rimozione vecchia versione da caricamento all'avvio e impostazione nuova version all'avvio

```
/sbin/chkconfig
postgresql-9.3  0:off   1:off   2:on    3:on    4:on    5:on    6:off
postgresql-9.5  0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

```
usage:   chkconfig --list [name]
         chkconfig --add <name>
         chkconfig --del <name>
         chkconfig [--level <levels>] <name> <on|off|reset|resetpriorities>
```

```
chkconfig postgresql-9.3 off
chkconfig postgresql-9.5 on
```

7. Test script di backup

Esiste uno script di backup che esegue un dumpall dell'intero db e lo comprime, inserendolo successivamente sul server di backup pacifico.

```
/storage/sire/users/abussani/bck_postgres.sh
```

Lo script viene eseguito ogni domenica alle 22 e 22 – utenza root

```
22 22 * * 0 /storage/sire/users/abussani/bck_postgres.sh
```

8. Cambio password

sudo -u postgres psql
# \password
messa su entrambi la stessa:
password

9. Su server master – host: oceano – per replica

Documentazione online:
https://www.digitalocean.com/community/tutorials/how-to-set-up-master-slave-replication-on-postgresql-on-an-ubuntu-12-04-vps

Sul master server creare un utenza per la replica

```
sudo su - postgres

psql -c "CREATE USER rep REPLICATION LOGIN CONNECTION LIMIT 1 ENCRYPTED PASSWORD
'password';"
```

utenza rep
password
password

Inserimento server slave in conf su master

```
/usr2/pgsql/9.3/data/postgresql.conf
```

legge da dove poter accettare connessioni da questo file:

```
/usr2/pgsql/9.3/data/pg_hba.conf
```

aggiungere (non alla fine del file) le permission per il server slave di replica

```
host    replication     rep     140.105.65.140/32   trust
```

poi nel file `postgresql.conf`

```
listen_addresses = 'localhost,IP_address_of_THIS_host'
wal_level = 'hot_standby'
archive_mode = on
archive_command = 'cd .'
max_wal_senders = 1
hot_standby = on
```

Eseguiti i cambiamenti sul postgresql.conf su oceano si riavvia il servizio

```
 /etc/init.d/postgresql-9.5 restart
Stopping postgresql-9.5 service:                           [  OK  ]
Starting postgresql-9.5 service:                           [  OK  ]
```

10.Su server di backup – host: lagunamarano – per replica

```
service postgresql stop
* Stopping PostgreSQL 9.5 database server
```

```
cd /usr/share/postgresql/9.5/
```

Aggiornare il file degli accessi:

`pg_hba.conf`

```
local   all         all                               trust
host    all         all         127.0.0.1/32          trust
host    all         all         140.105.70.44/32      trust
host    all         all         140.105.65.65/32      trust
host    all         all         140.105.70.146/32     trust
```

Aggiungere verso la fine del file:

```
host    replication rep           140.105.70.44/32      trust
```

Aggiungere le seguenti linee all'interno del file di configurazione:

`postgresql.conf`

You can use the same configuration options you set for the master server, modifying only the IP address to reflect the slave server's address:

```
listen_addresses = 'localhost,104.105.65.140'
wal_level = 'hot_standby'
archive_mode = on
archive_command = 'cd .'
max_wal_senders = 1
hot_standby = on
```

11. Replicating the Initial database:

Prima che lo slave possa replicare il master, necessitiamo di costruire il database sul quale verranno eseguite man mano gli aggiornamenti. Lo slave si occupa di leggere i file di log del master e di eseguire le stesse operazioni sul suo database, è necessario quindi che i due siano inizialmente identici.

Sul server master possizmo usare dei comandi di backup. Poi ci serv trasferire i dati al server slave e poi chiudere il comando di apertura del backup.

Usando l'utenza postgres

```
psql -c "select pg_start_backup('initial_backup');"

rsync -cva --inplace --exclude=*pg_xlog* /usr2/pgsql/9.5/data/
140.105.65.140:/var/lib/postgresql/9.5/main/

psql -c "select pg_stop_backup();"
```

Può accadere che ci siano degli errori, controllare che non siano file importanti.

Ora i dati del master dovrebbero essere sullo slave.

Ora è necessario configurare un file di recovery sul nostro server slave

I file di configurazione si dovrebbero trovare in:

```
/etc/postgresql/9.5/main/
```

non in

```
/var/lib/postgresql/9.5/main
```

Qui creiamo un file di recovery

```
recovery.conf
```

```
standby_mode = 'on'
primary_conninfo = 'host=140.105.70.44 port=5432 user=rep password=password'
trigger_file = '/tmp/postgresql.trigger.5432'
```

L'ultima linea nel file,`trigger_file`, è la parte interessante dell'intera configurazione. Se create il file all'interno della macchina slave, il database server si riconfigurerà agendo come master.

Questa operazione romperà il normale flusso delle operazioni, specialmente se il master continua a funzionare, ma è quello che è necessario quando il master si interrompe. Questo permette allo slave di iniziare ad accettare delle modifiche sul db. E' possible risolvere quindi risolvere il problema e rimettere in funzionamento il server master.

Per eseguire il server:

```
service postgresql start
```

Controllare i log se ci sono problemi

```
less /var/log/postgresql/postgresql-9.5-main.log
```

Inoltre controllare anche la connessione generata dallo start

All'interno della directory pg_log controllare il file log .opts

```
cat postmaster.opts
/usr/lib/postgresql/9.5/bin/postgres "-D" "/var/lib/postgresql/9.5/main" "-c"
"config_file=/etc/postgresql/9.5/main/postgresql.conf"
```

modifico i file di config …

Documentazione online su come è possibile risolvere alcuni problemi che possono accaddere:

http://dba.stackexchange.com/questions/16781/postgresql-9-1-hot-backup-error-the-database-system-is-starting-up

12.Ubuntu boot start postgresql

Utenza root:

```
update-rc.d postgresql enable
update-rc.d: warning:  start runlevel arguments (none) do not match postgresql
Default-Start values (2 3 4 5)
update-rc.d: warning:  stop runlevel arguments (none) do not match postgresql
Default-Stop values (0 1 6)
 Enabling system startup links for /etc/init.d/postgresql ...
 Removing any system startup links for /etc/init.d/postgresql ...
   /etc/rc0.d/K21postgresql
   /etc/rc1.d/K21postgresql
   /etc/rc2.d/S19postgresql
   /etc/rc3.d/S19postgresql
   /etc/rc4.d/S19postgresql
   /etc/rc5.d/S19postgresql
   /etc/rc6.d/K21postgresql
 Adding system startup for /etc/init.d/postgresql ...
   /etc/rc0.d/K21postgresql -> ../init.d/postgresql
   /etc/rc1.d/K21postgresql -> ../init.d/postgresql
   /etc/rc6.d/K21postgresql -> ../init.d/postgresql
   /etc/rc2.d/S19postgresql -> ../init.d/postgresql
   /etc/rc3.d/S19postgresql -> ../init.d/postgresql
   /etc/rc4.d/S19postgresql -> ../init.d/postgresql
```

```
/etc/rc5.d/S19postgresql -> ../init.d/postgresql
```

13.Postgresql php e pool di connessioni

Documentazione online:
https://wiki.postgresql.org/wiki/Replication,_Clustering,_and_Connection_Pooling

Il server nettuno dovrà contenere una corretta gestione degli accessi, per ora il file connesso.php si connette al server di replica e se il server di replica è bloccato, effettua la richiesta su oceano.