

INSTALANDO O POSTGRESQL NO LINUX

Faça o download do instalador da versão Postgresql 8.2.4-1 acessando o endereço:

<http://gcom.ipad.com.br/postgresql-8.2.4.tar.bz2>

Baixe o arquivo postgresql-8.2.4-1.zip.

Por favor, leia o FAQ http://pginstaller.projects.postgresql.org/FAQ_windows.html antes de começar.

Instalação passo a passo

1 Descompacte e compile o arquivo de instalação

```
$ tar -zxvf postgresql-8.2.4.tar.gz
$ cd postgresql-8.2.4
$ ./configure --prefix=/usr/local/pgsql --without-readline
$ make
# make install
# cp /usr/local/pgsql/bin/* /usr/bin
```

Compile o pacote do dblink

```
$ cd /contrib/dblink
$ make
# make install
```

OBS: /usr/local/pgsql - diretório onde será instalado o PostgreSQL.

2 Crie o usuário Postgres:

```
# adduser postgres
```

3 Crie o diretório onde os dados serão armazenados:

```
# mkdir /usr/local/pgsql/data
```

4 Dê permissão ao usuário postgres:

```
# chown postgres data -v
```

5 Agora efetuaremos o login com o usuário postgres para fazer as devidas configurações e criação da base de dados.

```
# su - postgres
$ cd /usr/local/pgsql/
$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

6 Testaremos se o PostgreSQL está realmente funcionando, digite o comando abaixo:

```
$ /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data
```

Para poder parar o comando acima, execute o famoso CONTROL+C.

7 Agora copiaremos o script de inicialização do PostgreSQL que vem por padrão na sua respectiva pasta.

Basta entrar na pasta onde você baixou o PostgreSQL e dar o seguinte comando:

```
# cp contrib/start-scripts/linux /etc/init.d/postgresql  
# chmod +x /etc/init.d/postgresql
```

8 Com o comando acima estamos colocando o script de inicialização do postgresql para iniciá-lo, pará-lo e restartá-lo.

Como o PostgreSQL trabalha com sockets podemos ativar o TCP/IP na própria inicialização sem alterar nenhuma configuração dele, bastando apenas editar o arquivo abaixo:

/etc/init.d/postgresql

Como está:

```
echo -n "Starting PostgreSQL: " su - $PGUSER -c "$DAEMON -D '$PGDATA' &" »$PGLOG  
2>&1
```

Como deve ficar:

```
echo -n "Starting PostgreSQL: " su - $PGUSER -c "$DAEMON -i -D '$PGDATA' &" »$PGLOG  
2>&1
```

OBS: Note que foi acrescentado apenas um -i na linha acima.

Iniciando Postgres:

/etc/init.d/postgresql start

Parando Postgres:

/etc/init.d/postgresql stop

Restartando Postgres:

/etc/init.d/postgresql restart

Verificando status do Postgres:

/etc/init.d/postgresql status

Verificando se o postgres esta rodando:

```
# netstat -avnp |grep 5432
```

Resultado seria algo parecido com isso:

```
tcp 0 0 10.18.0.243:5432 10.18.0.243:49843 ESTABELECIDAS132/postgres: gsan  
tcp 0 0 10.18.0.243:5432 10.18.0.243:49842 ESTABELECIDAS131/postgres: gsan  
tcp 0 0 10.18.0.243:5432 10.18.0.243:49841 ESTABELECIDAS130/postgres: gsan
```

9 Altere arquivo o postgres.conf

```
$ vi /usr/local/pgsql/data/postgres.conf
```

Mude o parâmetro:

```
De: listen_addresses = '127.0.0.1'
```

```
Para: listen_addresses = '*'
```

10 Agora verifique a configuração do pg_hba.conf

Verificar se existe a entrada da sua rede local no pg_hba.conf para o acesso ao banco conforme documentação do PostgreSQL.

11 Para visualizar as databases existentes:

```
$ psql -l postgres | postgres | LATIN1  
template0 | postgres | LATIN1  
template1 | postgres | LATIN1
```

OBS: Lembrando que para executar o comando psql você não poderá estar como usuário root, deverá usar o usuário criado anteriormente.

12 Crie a pasta indices para armazenar os índices das tabelas:

```
$ mkdir /usr/local/pgsql/data/indices
```

13 Conecte com a base de dados postgres:

```
$ psql postgres
```

14 Criar os grupos padrão:

```
postgres=# CREATE ROLE pg_users NOSUPERUSER NOINHERIT NOCREATEDB  
NOCREATEROLE;  
postgres=# CREATE ROLE pg_aplic NOSUPERUSER NOINHERIT NOCREATEDB  
NOCREATEROLE;
```

15 Crie os usuários padrão:

```
postgres=# CREATE ROLE gsan_admin LOGIN  
ENCRYPTED PASSWORD 'md5e25cb13bee657fd472ea261733d52b12'  
NOSUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE;
```

```
postgres=# CREATE ROLE gsan_batch LOGIN  
ENCRYPTED PASSWORD 'md5a28a39f8784453813c2aa5403da3949d'  
NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;  
GRANT pg_aplic TO gsan_batch;
```

```
postgres=# CREATE ROLE gsan_olap LOGIN  
ENCRYPTED PASSWORD 'md53a5ee6d07b205e698e22b46cbe086335'  
NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;  
GRANT pg_aplic TO gsan_olap;
```

```
postgres=# CREATE ROLE gsan_online LOGIN
ENCRYPTED PASSWORD 'md54e64a47f54d3289600f400a939abcf86'
NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;
GRANT pg_aplic TO gsan_online;
```

16 Crie as bases de dados gsan_comercial e gsan_gerencia:

```
postgres=# CREATE DATABASE gsan_comercial
WITH OWNER = gsan_admin
ENCODING = 'LATIN1'
TABLESPACE = pg_default;
```

```
postgres=# CREATE DATABASE gsan_gerencia
WITH OWNER = gsan_admin
ENCODING = 'LATIN1'
TABLESPACE = pg_default;
```

17 Crie tablespace dos índices:

```
postgres=# CREATE TABLESPACE indices OWNER postgres
LOCATION '/usr/local/pgsql/data/indices';
```

A localização da pasta dos índices vai depender de cada instalação, dependendo do local ou volume reservado para este armazenamento.

18 Desconecte da base de dados postgres:

```
postgres=# \q
```

19 Restaure a base gsan_comercial a partir de um backup:

```
$ pg_restore -d gsan_comercial < gsan_comercial.backup
```

20 Restaure a base gsan_gerencia a partir de um backup:

```
$ pg_restore -d gsan_gerencia < gsan_gerencia.backup
```

REFERÊNCIAS

<http://www.vivaolinux.com.br/>

<http://www.postgresql.org.br/>

Para retornar, clique no link [RETORNAR](#)

From:

<https://www.gsan.com.br/> - Base de Conhecimento de Gestão Comercial de Saneamento

Permanent link:

https://www.gsan.com.br/doku.php?id=instalacao_no_linux&rev=1395952159

Last update: 31/08/2017 01:11



