|  |  |
| --- | --- |
|  | Título do documento  Classe de Cartaz Político |
| Projeto: CEDAP |

|  |  |  |
| --- | --- | --- |
| Responsável | Ação | Data |
| Rene Faustino Gabriel Junior | Criação do Documento | 05/08/2016 |

**Instalando e Configurando o PostgreSQL no Slackware**

Começaremos criando o diretório onde será guardado o source do postgresql e baixando o mesmo :

**# cd /opt/   
# mkdir postgresql   
# cd postgresql   
# wget ftp://ftp.au.postgresql.org/ pub /postgresql/v7.3.4/postgresql-7.3.4.tar.gz**

Ou envie por FTP para a pasta /opt/postgresql

Depois , descompactamos o source e compilamos:

**# tar xvzf postgresql-7.3.4.tar.gz   
# cd postgresql-7.3.4   
# ./configure --prefix=/usr   
# gmake   
# gmake install**

Se tudo tiver dado certo , você vai estar com o postgresql instalado na sua maquina. Os binários vão estar instalados em /usr/bin/ as libs em /usr/lib/postgresql/ os docs em /usr/doc/postgresql/ as manpages em /usr/man. Assim , você não precisa adicionar nada no .bashrc do seu home ou mexer no profile global . Agora vamos adicionar o usuário e grupo para o postgresql poder rodar:

**# groupadd postgres   
# useradd -g postgres -s /bin/bash -d /var/lib/pgsql postgres   
# mkdir -p /var/lib/pgsql/data   
# chown -R postgres:postgres /var/lib/pgsql/**

**Iniciando o postgresql:**

**# su - postgres   
$ initdb -D /var/lib/pgsql/data   
$ postmaster -i -D /var/lib/pgsql/data >logfile 2>&1 &**

Se tudo estiver dado certo , voce poderá dar um ps ax e verá o postgresql rodando perfeitamente. Ex:

**$ ps ax   
$ 17544 pts/3 S 0:00 /usr/bin/postmaster -i -D /var/lib/pgsql/data   
$ 17545 pts/3 S 0:00 postgres: stats buffer process   
$ 17547 pts/3 S 0:00 postgres: stats collector process**

**Criando a database para o usuário postgres:**

**$ createdb postgres   
CREATE DATABASE**

**Logando no banco para fazer as alterações:**

**$ psql postgres   
Welcome to psql 7.3.4, the PostgreSQL interactive terminal .   
  
Type: copyright for distribution terms   
h for help with SQL commands   
? for help on internal slash commands   
g or terminate with semicolon to execute query   
q to quit**  
  
**postgres=#**

Pronto , você já está logando no banco , agora vamos alterar a senha do usuário postgres:

**postgres=# alter user postgres with encrypted password 'sua-senha';   
ALTER USER   
postgres=#q**

Pronto, você alterou a senha do administrador do banco que é o usuário postgres, mude a parte 'sua-senha' para uma senha de sua preferência . E não esqueça do ; no final , se não vai retornar um mensagem de erro .

Agora vamos fazer algumas alterações no arquivo pg\_hba.conf para ele sempre pedir senha quando alguém for logar no banco . Pois por padrão o postgresql não exige senha na hora de logar, e sendo assim qualquer usuário que tiver uma shell no sistema , consegue acesso total ao banco . Editando o arquivo pg\_hba.conf que está em /var/lib/pgsql/data altere as seguintes linhas no final do arquivo :

**local all all trust   
host all all 127.0.0.1 255.255.255.255 trust**

Estas linha libera todas as bases com permissão trust ( em localhost), a permissão trust da acesso as bases sem requisição de senha ( mesmo que o user tenha uma senha ). Troque as palavras trust por md5, deixando assim :

**local all all md5   
host all all 127.0.0.1 255.255.255.255 md5**

**Agora inicie o seu postgresql:**

**$ pg\_ctl stop -D /var/lib/pgsql/data/ -s -m fast   
$ postmaster -i -D /var/lib/pgsql/data >logfile 2>&1 &**

**Agora teste se o seu postgresql esta pedindo senha :**

**$ psql postgres   
Password**

Pronto , agora o seu banco de dados está mais seguro , poi só quem vai logar nele é quem tiver senha . E isso vai lhe poupar umas boas noites de sono :)

Para não ter que ficar digitando isso na manualmente , segue abaixo o script de inicialização, ele vem junto no pacote do source do postgresql, e está dentro do diretório crontrib/start-scripts. Mas como ele não atendia todas as minhas necessidades , por isso fiz as alterações que precisava.

**#! /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   
# Altered by: Cláudio Borges   
  
# $Header: /cvsroot/pgsql-server/contrib/start-scripts/linux,v 1.3 2001/07/30 14:52:42 momjian Exp $   
  
## EDIT FROM HERE   
  
# Installation prefix   
prefix=/usr   
  
# Data directory   
PGDATA="/var/lib/pgsql/data"**

**# Who to run pg\_ctl as, should be "postgres".   
PGUSER=postgres   
  
# Where to keep a log file   
PGLOG="/var/log/postgresql.log"   
  
## STOP EDITING HERE   
  
# Check for echo -n vs echo c   
if echo 'c' | grep -s c >/dev/null 2>&1 ; then   
ECHO\_N="echo -n"   
ECHO\_C=""   
else   
ECHO\_N="echo"   
ECHO\_C='c'   
fi   
  
# The path that is to be used for the script   
PATH=/usr/local /sbin:/usr/local /bin:/sbin:/bin:/usr/sbin:/usr/bin   
  
# What to use to start up the postmaster   
DAEMON="$prefix/bin/pg\_ctl"   
  
set -e   
  
# Only start if we can find pg\_ctl.   
test -f $DAEMON || exit 0   
  
# Parse command line parameters.   
case $1 in   
start)   
$ECHO\_N "Sta! rting PostgreSQL: "$ECHO\_C   
su - $PGUSER -c $prefix/bin/'postmaster -i -D '$PGDATA' &' postgres > $PGLOG 2>&1 &   
  
echo "ok"   
;;   
stop)   
echo -n "Stopping PostgreSQL: "   
su - $PGUSER -c "$DAEMON stop -D '$PGDATA' -s -m fast"   
echo "ok"   
;;   
restart)   
echo -n "Restarting PostgreSQL: "   
su - $PGUSER -c "$DAEMON restart -D '$PGDATA' -s -m fast"   
echo "ok"   
;;   
status )   
su - $PGUSER -c "$DAEMON status -D '$PGDATA'"   
;;   
\*)   
# Print help   
echo "Usage: $0 {start|stop|restart| status }" 1>&2   
exit 1   
;;   
esac   
  
exit 0**

Salve este script como rc.postgresql no diretório /etc/rc.d/ depois adicione a permissão de execução chmod +x rc.postgresql. Depois é so adicionar o rc.postgresql no seu rc. local (faça isso com o usuário root):

**# chmod +x rc.postgresql   
# echo "/etc/rc.d/rc.postgresql start" >> /etc/rc.d/rc.local**

Pronto , agora quando iniciar o seu micro , o postgresql será iniciado automáticamente . Espero ter ajudado alguém que tenha dúvidas em relação a este excelente banco de dados .   
  
Por Claudio Borges