

Replicação de Dados com PostgreSQL

O Postgre, SGBD (Sistema Gerenciador de Banco de Dados) gratuito e muito popular, aceita replicação assíncrona de dados com um servidor master (mestre) e múltiplos servidores slave (escravo). Com a replicação assíncrona os dados não são sincronizados em tempo real entre os dois servidores, através de logs de transação. É permitido apenas um servidor master, com a quantidade de slaves necessária. O servidor principal (mestre) trata todas as transações de leitura e escrita, enquanto os servidores secundários (escravos) apenas replicam os dados do principal, e podem permitir operações apenas de leitura ou nenhuma operação. Se o servidor master falhar um servidor slave pode ser promovido a master e assumir a função, fazendo com que o serviço de dados não seja interrompido. Em uma loja virtual, por exemplo, a interrupção do serviço de banco de dados resulta em prejuízo financeiro, daí a importância de um servidor secundário assumir imediatamente a função do servidor principal que falhou.

O Postgre tem uma função nativa de replicação de dados entre servidores. Como já dito, replicação assíncrona com apenas um servidor master. Esse tutorial ensina de forma prática a configuração para replicação de dados. É bastante aconselhável que os dois computadores utilizados tenham a mesma versão do Postgre e do Windows ou Linux. O exemplo a seguir foi feito em Windows.

Servidor master:

Começaremos configurando nosso servidor mestre. Todos esses comandos devem ser executados com o usuário postgres.

1. Primeiro criaremos um usuário chamado "rep" que pode ser usado apenas para replicação. Dentro da pasta /bin, execute os comandos:

```
Set pguser=postgres
Set pgpassword=postgres
psql -c "CREATE USER rep REPLICATION LOGIN CONNECTION LIMIT 1 ENCRYPTED PASSWORD 'yourpassword';"
```

2. Abra o arquivo /data/pg_hba.conf e adicione uma linha para permitir que o novo usuário tenha acesso a este servidor. Utilize os seguintes parâmetros:

```
host      replication      rep                                IP_Server_Slave/32          md5
```

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# IPv4 local connections:					
host	all	all		0/0	md5
host	replication		rep	192.168.10.13/32	md5
# IPv6 local connections:					
host	all	all		::1/128	md5
# Allow replication connections from localhost, by a user with the					
# replication privilege.					
#host	replication	postgres		127.0.0.1/32	md5
#host	replication	postgres		::1/128	md5

3. Em seguida, vamos abrir o arquivo de configuração principal do postgres, /data/postgresql.conf. Encontre esses parâmetros. Descomente-os se eles forem comentados e modifique os valores de acordo com o que listamos abaixo:

```
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
```

4. Agora reinicie o serviço do postgres.

Servidor slave:

Obs.: É aconselhável instalar o Postgre na raiz do HD do computador slave, para depois o comando `promote` conseguir ler o arquivo `postmaster.pid`, necessário para a realização da promoção. Na pasta "arquivos de programas" muitas vezes o Windows bloqueia o acesso a arquivos. Instale por exemplo em `c:\PostgreSQL`.

1. Parar o serviço no servidor slave da mesma maneira que paramos o master.
2. Editar o arquivo `postgresql.conf`, localizado normalmente na pasta `\data` do SGBD. Encontre esses parâmetros. Descomente-os se eles forem comentados e modifique os valores de acordo com o que listamos abaixo:

```
wal_level = hot_standby
max_wal_senders = 1
wal_keep_segments = 20
hot_standby = on
max_standby_archive_delay = 30s
max_standby_streaming_delay = 30s
```

3. Criar o arquivo `recovery.conf` na pasta `\data` do SGBD com o seguinte conteúdo:

```
standby_mode = 'on'
primary_conninfo = 'host= IP_Server_Master port=5432 user=rep password=yourpassword'
trigger_file='/tmp/pgtrigger'
```

O parâmetro "standby_mode" em "on" ativa o modo slave no servidor. O parâmetro "primary_conninfo" contém uma string de conexão ao servidor master. "trigger_file" contém o nome de um arquivo de gatilho. Se o servidor encontrar esse arquivo se torna master automaticamente.

4. Utilizar o aplicativo `pg_basebackup` encontrado na pasta `\bin` do SGBD para fazer o backup inicial (base) do servidor principal no HD do servidor secundário. Esse backup depois vai sendo atualizado através dos logs de transação do servidor principal.

Utilize a seguinte instrução no prompt de comando do servidor slave:

```
Set pguser=rep
Set pgpassword=yourpassword
pg_basebackup -D /data/ -h 192.168.2.109 -U rep
```

-D diz a pasta local onde será feito o backup.

-h diz o endereço IP do servidor master.

-U diz o nome de usuário no servidor master.

No lugar do endereço IP pode ser usado o nome de HOST do servidor principal.

Após executar esse comando deve aparecer uma mensagem dizendo que o `pg_backup_stop` foi concluído. Nesse caso é criada no HD do servidor slave a pasta `C:\data` que contém o backup dos dados do servidor principal. Você deve apagar o conteúdo de `C:\postgresql\9.*\data` e copiar no lugar as pastas contidas em `C:\data`. *Porém, os arquivos de configuração irão vir com as configurações do Server Master, volte para as o passo 2 e reconfigure os arquivos do Server Slave.*

5. Iniciar o serviço de servidor slave da mesma maneira como iniciou o servidor master.

Se tudo ocorrer bem a replicação começa imediatamente.

Promovendo server slave a master:

Para promover o servidor slave a master em caso de falha do servidor principal, utilize o aplicativo pg_ctl localizado na pasta \bin do SGBD. Utilize no prompt do servidor slave o seguinte comando:

```
pg_ctl promote -D "c:\caminho completo entre aspas duplas\data"
```

Fontes:

- <https://www.digitalocean.com/community/tutorials/how-to-set-up-master-slave-replication-on-postgresql-on-an-ubuntu-12-04-vps>
- <http://robsonfm.blogspot.com/2015/10/replicacao-de-dados-com-postgresql.html>