BACKUPS LOCAIS E REMOTOS COM RSYNC

Thiago Mendonça

- Nerd;
- Gerente de TI da Terrali Móveis Planejados;
- Ativista do Movimento Software Livre;
- Usuário Debian e Trisquel.

RSync

- Sincronizador de arquivos para destinos locais e remotos.
- Diversas abordagens de sincronização.
- Usa menos banda para tráfego remoto de dados devido a compressão dos dados.
- Fácil de implementar.

Estrutura dos exemplos:

- Sincronização entre diretórios montados localmente:

  rsync -arzvPh --delete /origem/ /destino/local/

- Sincronização entre diretórios remotos:

  rsync -arzvPh --delete -e ssh /origem/ servidor.remoto:diretoriodestino/

Quebrando o comando

- -a – arquivar, preserva todos os atributos dos arquivos;
- -z – comprimir, consome menos recursos de rede, porém, uma carga maior na CPU;
- -r – recursividade, força a busca recursiva por arquivos e diretórios em níveis inferiores do diretório de origem.
- -v – verboso, habilita o modo verboso na execução do comando. Não é necessário para o arquivo final;

Quebrando o comando

- -P – progresso, exibe pelo modo verboso o progresso do envio/recebimento de cada arquivo durante a transferência;
- -h – deixa a saída de texto mais amigável para leitura.
- --delete – sincroniza arquivos apagados na origem, apagando no destino também.
- -e – especifica, no nosso, caso a enviar os arquivos externamente via ssh.

Recuperando dados salvos com RSync

- Backup:
  rsync -arzvPh --delete /diretorio1/ /diretorio2/
- Recuperação:
  rsync -arzvPh --delete /diretorio2/ /diretorio1/
- Caso tenha perdido até o diretório de origem:
  rsync -arzvPh --delete /diretorio2 /diretorio1/

  (omitir a ***/*** do diretório a ser restaurado faz o truque)

Automatizando a tarefa

Para automatizar o backup será preciso:

- 1 script bash simples contendo o comando completo;
- Ajustar as permissões de execução do script;
- Otimizar o acesso ao host remoto;
- Adicionar ao cron tab da máquina responsável por executar o script a tarefa a execução do script para todos os dias às 17:00h.

1º – script.sh

```
#!/bin/bash

# Cria o arquivo de log

echo “” > rsync-`date +”%d-%m-%Y_%H”`.log

# Comando RSync

rsync -azvrPh --delete --log-file=rsync-`date +”%d-%m-%Y_
%H”`.log -e ssh origem/ servidor.remoto:diretoriodestino/
```

Tornar o script.sh executável

chmod +x script.sh

Acesso ao servidor remoto via ssh

O Problema:

- Com o acesso padrão: ssh usuário@servidor temos de digitar a senha do usuário. Podemos passar essa senha pelo script, porém não é seguro.

Solução:

- Trocar chaves entre as máquinas para conexão sem senha segura.

Como fazer?

- Criar as chaves locais:

  ssh-keygen

- Enviar a chave pública para o host remoto:

  ssh-copy-id -i ~/.ssh/id_rsa.pub usuario@maquina-remota

- Testar se deu certo:

  ssh usuario@maquinaremota

Último passo - CronTab

- Dê um crontab -e;
- Selecione o editor de sua preferencia;
- Adicione a linha:
  * 17 * * * /caminho/para/o/script.sh
- Salve o documento e feche ele;
- Digite crontab -l e verifique se a tarefa nova está listada.

E ficamos por aqui.

Contato:

- E-mail: thiago@acesso.me
- Twitter: _tarkun_
- Diaspora: t_f_mendonca@diasporabr.com.br
- Blog: acesso.me