

# Backup com Rsync e Rclone - Shell Script

Gerado pelo ChatGPT

Claro! Vou te mostrar um exemplo de shell script que usa `rsync` para fazer backup local e `rclone` para fazer backup em um serviço de armazenamento em nuvem. Esse script é útil para quem deseja manter backups locais e na nuvem com uma única execução.

```
### Exemplo de Script de Backup com `rsync` e `rclone`
```

Este script faz o backup de um diretório local para um local de backup e, em seguida, usa `rclone` para enviar o backup para um serviço de armazenamento em nuvem.

```
```bash
```

```
#!/bin/bash
```

```
# Configurações
```

```
SOURCE_DIR="/caminho/do/diretorio/origem" # Diretório que você deseja fazer backup
```

```
LOCAL_BACKUP_DIR="/caminho/do/backup/local" # Diretório onde o backup local será salvo
```

```
RCLONE_REMOTE="nome_do_remote:backup" # Configuração do rclone (ex: gdrive:backup)
```

```
# Data e hora atuais para adicionar aos nomes dos arquivos
```

```
now=$(date +"%Y-%m-%d_%H-%M-%S")
```

```
# Nome do arquivo de backup
```

```
BACKUP_FILE="backup_$(now).tar.gz"
```

```
# Função para verificar se um comando está disponível
```

```
check_command() {
```

```
    command -v "$1" >/dev/null 2>&1 || { echo >&2 "O comando $1 não está instalado. Saindo.";
```

```
    exit 1; }
```

```
}
```

```
# Verifica se rsync e rclone estão instalados
```

```
check_command rsync
```

```
check_command rclone
```

```
# Cria o diretório de backup local, se não existir
```

```
mkdir -p "$LOCAL_BACKUP_DIR"
```

### # Cria o backup local com rsync

```
echo "Iniciando backup local com rsync..."
rsync -avzhP --delete "$SOURCE_DIR/" "$LOCAL_BACKUP_DIR/"
echo "Backup local concluído."
```

### # Compacta o backup local

```
echo "Compactando o backup..."
tar -czf "$LOCAL_BACKUP_DIR/$BACKUP_FILE" -C "$LOCAL_BACKUP_DIR" "$(basename
"$SOURCE_DIR")"
echo "Compactação concluída."
```

### # Envia o backup para o serviço de armazenamento em nuvem usando rclone

```
echo "Enviando o backup para o armazenamento em nuvem com rclone..."
rclone copy "$LOCAL_BACKUP_DIR/$BACKUP_FILE" "$RCLONE_REMOTE/"
echo "Backup na nuvem concluído."
```

### # Limpa arquivos de backup antigos, mantendo os últimos 5 backups

```
echo "Removendo backups antigos..."
find "$LOCAL_BACKUP_DIR" -type f -name "backup_*.tar.gz" -mtime +30 -exec rm {} \;
echo "Limpeza concluída."
```

### # Mensagem final

```
echo "Processo de backup completo. Arquivo de backup local em:
$LOCAL_BACKUP_DIR/$BACKUP_FILE"
```
```

### ### Como usar o script:

1. **Crie o arquivo do script:**

```
```bash
nano backup_script.sh
```
```

2. **Cole o conteúdo do script no editor e salve o arquivo (Ctrl + X, depois Y, e Enter).**

3. **Dê permissão de execução ao script:**

```
```bash
chmod +x backup_script.sh
```
```

4. **Execute o script:**

```
```bash
sudo ./backup_script.sh
```
```

### ### Explicação dos Comandos e Funções:

- `rsync -av --delete "$SOURCE_DIR/" "$LOCAL_BACKUP_DIR/"`: Sincroniza o diretório de origem com o diretório de backup local. O `--delete` garante que os arquivos deletados no diretório de origem também sejam removidos do backup.
- `tar -czf "$LOCAL_BACKUP_DIR/$BACKUP_FILE"`: Compacta o diretório de backup local em um arquivo `.tar.gz`.
- `rclone copy "$LOCAL_BACKUP_DIR/$BACKUP_FILE" "$RCLONE_REMOTE/"`: Envia o arquivo de backup compactado para o serviço de armazenamento em nuvem configurado no `rclone`.
- `find "$LOCAL_BACKUP_DIR" -type f -name "backup_*.tar.gz" -mtime +30 -exec rm {} \;`: Remove backups locais antigos, mantendo apenas os últimos 30 dias de backups.

### ### Configuração do `rclone`:

#### 1. Instale o `rclone`:

```
```bash
sudo apt-get install rclone
```
```

#### 2. Configure o `rclone`:

```
```bash
rclone config
```
```

Siga as instruções para configurar o `rclone` para o serviço de armazenamento em nuvem de sua escolha.

#### 3. Verifique a configuração do `rclone`:

```
```bash
rclone ls nome_do_remote:
```
```

Substitua os caminhos e nomes conforme necessário para se adequar ao seu ambiente. Se precisar de mais ajustes ou tiver perguntas, estou à disposição!

---

Revision #1

Created 2 September 2024 15:32:19 by Administrador

Updated 2 September 2024 15:39:54 by Administrador