

Documentação - Script onedrive-rclone.sh

Visão Geral

Este script realiza um **backup diário** do diretório /DATA para o OneDrive usando rclone, garantindo consistência de dados ao:

- Parar containers Docker (respeitando a ordem do MariaDB).
 - Parar serviços do CasaOS.
 - Executar rclone sync com logs controlados.
 - Reiniciar serviços do CasaOS e containers Docker em ordem.
 - Gerar e rotacionar logs compactados em /DATA.
-

Fluxo Geral do Script

1. **Preparação** - Inicializa variáveis, diretórios e logs.
 2. **Parada de Containers Docker** - Coleta e para todos os containers (MariaDB por último).
 3. **Parada de Serviços CasaOS** - Para todos os serviços casaos*.
 4. **Sincronização com Rclone** - Sincroniza /DATA para onedrive:/RCLONE.
 5. **Reinício de Serviços CasaOS** - Reinicia todos os serviços casaos*.
 6. **Reinício de Containers Docker** - Reinicia containers (MariaDB primeiro, depois outros com delay).
 7. **Compactação e Rotação de Logs** - Compacta logs e mantém apenas 7 versões.
 8. **Limpeza** - Remove arquivos temporários.
-

Configurações Principais

Diretórios e Destinos

Parâmetro	Valor	Descrição
SRC	/DATA	Diretório local para backup
DEST	onedrive:/RCLONE	Destino remoto no OneDrive
LOG_DIR	/var/log/onedrive-rclone	Logs temporários durante execução

Arquivos de Log

Tipo	Local Temporário	Compactado em /DATA	Padrão
Log Principal	/var/log/onedrive-rclone/	/DATA/	onedrive-rclone-YYYY-MM-DD_HH-MM-SS.tar.gz
Log Rclone	/var/log/onedrive-rclone/	/DATA/	rclone-debug-YYYY-MM-DD_HH-MM-SS.tar.gz

Retenção

- **Máximo de versões mantidas:** 7 arquivos compactados de cada tipo.
- **Local:** /DATA/
- **Frequência de limpeza:** Ao final de cada execução.

Controle de Containers Docker

Coleta e Parada

Funcionamento:

```
docker ps --format '{{.ID}} {{.Names}}' > "$DOCKER_LIST"
```

- Lista todos os containers em execução com ID e nome.
- Salva em arquivo temporário para uso posterior.

Ordem de Parada:

1. Para todos os containers **exceto** os que terminam com mariadb:

```
grep -v ' mariadb$' "$DOCKER_LIST" | awk '{print $1}' | xargs docker stop
```

2. Em seguida, para **todos os containers mariadb** (nomes terminando com mariadb):

```
grep ' mariadb$' "$DOCKER_LIST" | awk '{print $1}' | xargs docker stop
```

Objetivo: Garantir que o banco de dados MariaDB seja parado por último, permitindo que outros serviços façam graceful shutdown.

Reinício

Ordem de Reinício:

1. Inicia primeiro os containers com nome terminando em mariadb:

```
grep ' mariadb$' "$DOCKER_LIST" | awk '{print $1}' | xargs docker start
```

- Aguarda 30 segundos para estabilização completa do banco.
2. Inicia os demais containers **um por vez** com delay de 30 segundos:

```
grep -v ' mariadb$' "$DOCKER_LIST" | awk '{print $1}' | while read -r CID; do
    docker start "$CID"
    sleep 30
done
```

Objetivo: Evitar que serviços dependentes do banco tentem conectar antes do MariaDB estar pronto. Delay de 30s entre cada container garante que cada um inicie corretamente.

Controle dos Serviços CasaOS

Parada

```
systemctl list-unit-files --type=service | \
awk '{print $1}' | grep '^casaos' | xargs -r systemctl stop
```

- Lista todos os serviços do systemd.
- Filtra apenas aqueles que começam com casaos.
- Para cada um com systemctl stop.

Reinício

```
systemctl list-unit-files --type=service | \
awk '{print $1}' | grep '^casaos' | xargs -r systemctl start
```

- Mesmo filtro, agora com systemctl start.

Objetivo: Evitar que serviços do CasaOS acessem o storage ou banco de dados durante o backup, garantindo integridade dos dados.

Sincronização com Rclone

Comando Completo

```
rclone sync "$SRC" "$DEST" \
--progress \
--log-level NOTICE \
--log-file "$RCLONE_LOG" \
--stats 1m \
--stats-file-name-length 0 \
--transfers 8 \
--checkers 8 \
--retries 3 \
--timeout 12h \
--onedrive-chunk-size 100M \
--onedrive-no-versions \
--fast-list \
--one-file-system \
--drive-stop-on-upload-limit \
--delete-during
```

Parâmetros Explicados

Parâmetro	Valor	Descrição
--progress	-	Mostra barra de progresso
--log-level	NOTICE	Log simplificado (sem verbosidade por arquivo)
--log-file	\$RCLONE_LOG	Arquivo de log detalhado separado
--stats	1m	Atualiza estatísticas a cada 1 minuto
--transfers	8	Máximo de uploads paralelos (moderado)
--checkers	8	Máximo de verificações paralelas
--retries	3	Tenta 3 vezes antes de falhar
--timeout	12h	Tolera execuções longas
--onedrive-chunk-size	100M	Tamanho de chunk para uploads
--onedrive-no-versions	-	Não mantém versões anteriores no OneDrive
--fast-list	-	Otimiza listagem de muitos arquivos
--one-file-system	-	Não atravessa outros mounts
--drive-stop-on-upload-limit	-	Respeita limites de upload do provedor
--delete-during	-	Deleta remotamente durante sync (não após)

Comportamento de Sincronização

- **Incremental:** Apenas arquivos novos/modificados são enviados.
- **Deleção:** Arquivos removidos em /DATA são também removidos do destino.
- **Verificação:** Tamanho + timestamp + checksum para determinar se arquivo mudou.
- **Resultado esperado:** Primeira execução pode levar horas; execuções subsequentes (diárias) levam minutos.

Logs e Rotação

Criação de Logs Durante Execução

1. **Log Principal** - Captura toda saída do script:
 - Arquivo: /var/log/onedrive-rclone/onedrive-rclone-TIMESTAMP.log
 - Conteúdo: Parada/reinício de containers, serviços, status geral.
2. **Log Rclone** - Detalhes da sincronização:
 - Arquivo: /var/log/onedrive-rclone/rclone-debug-TIMESTAMP.log
 - Conteúdo: Listagens, transferências, erros específicos do rclone.

Compactação

Ao final da execução, ambos os logs são compactados separadamente:

```
tar -czf "/DATA/onedrive-rclone-TIMESTAMP.tar.gz" -C "/var/log/onedrive-rclone" "onedrive-rclone-TIMESTAMP.log"
```

```
tar -czf "/DATA/rclone-debug-TIMESTAMP.tar.gz" -C "/var/log/onedrive-rclone" "rclone-debug-TIMESTAMP.log"
```

Limpeza de Temporários

Após compactação, os arquivos .log são removidos do /var/log/onedrive-rclone.

Rotação (Retenção de 7)

O script mantém apenas os **7 arquivos mais recentes** de cada tipo:

```
# Para Logs principais
```

```
ls -lt "/DATA/onedrive-rclone-".tar.gz | sed -e '1,7d' | xargs -r rm -f
```

```
# Para Logs rclone
```

```
ls -lt "/DATA/rclone-debug-".tar.gz | sed -e '1,7d' | xargs -r rm -f
```

Efeito: Se o script rodar diariamente, mantém logs dos últimos 7 dias de cada tipo em /DATA.

Pré-Requisitos

Binários Necessários

- bash - Shell
- docker - Gerenciamento de containers
- systemctl - Gerenciamento de serviços
- rclone - Sincronização com cloud
- tar - Compactação de logs
- Utilitários POSIX: grep, awk, ls, sed, xargs, find, wc

Permissões

Execute o script com privilégios de root (ou usuario com sudo):

- Gerenciar containers Docker (docker group ou sudo)
- Administrar serviços (sudo systemctl)
- Ler /DATA
- Escrever em /var/log/onedrive-rclone e /DATA

Configuração Rclone

O rclone deve estar configurado com acesso ao OneDrive:

```
rclone config
```

Configure um remote chamado onedrive ou ajuste DEST no script conforme necessário.

Execução e Uso

Tornar Executável

```
chmod +x /caminho/para/onedrive-rclone.sh
```

Execução Manual

```
sudo /caminho/para/onedrive-rclone.sh
```

Ou sem sudo (se o usuário tiver permissões necessárias):

```
./onedrive-rclone.sh
```

Agendamento com Cron

Editar crontab:

```
sudo crontab -e
```

Exemplo: executar todos os dias às 03:00 (madrugada):

```
0 3 * * * /caminho/para/onedrive-rclone.sh >/dev/null 2>&1
```

Exemplo: executar a cada 12 horas:

```
0 */12 * * * /caminho/para/onedrive-rclone.sh >/dev/null 2>&1
```

Agendamento com Systemd Timer

Criar arquivo de serviço (/etc/systemd/system/onedrive-rclone.service):

[Unit]

Description=OneDrive Backup Script

After=network.target

[Service]

Type=oneshot

ExecStart=/caminho/para/onedrive-rclone.sh

User=root

Criar arquivo de timer (/etc/systemd/system/onedrive-rclone.timer):

[Unit]

Description=OneDrive Backup Timer

Requires=onedrive-rclone.service

[Timer]

OnCalendar=03:00

Persistent=true

[Install]

WantedBy=timers.target

Ativar:

```
sudo systemctl enable onedrive-rclone.timer
```

```
sudo systemctl start onedrive-rclone.timer
```

Troubleshooting

Script Não Executa ou Para com Erro

Verificar syntax:

```
bash -n /caminho/para/onedrive-rclone.sh
```

Consultar logs:

```
# Log mais recente (compactado)
```

```
tar -tzf /DATA/onedrive-rclone-*.tar.gz | tail -20
```

```
# Ou se ainda em execução (temporário)
```

```
tail -50 /var/log/onedrive-rclone/onedrive-rclone-*.log
```

Containers Não Sobem Após Execução

Verificar: - O nome do container MariaDB termina com mariadb? `bash docker ps -a | grep -i maria` - Há espaço em disco em /DATA ou /var/lib/docker? `bash df -h /DATA df -h /var/lib/docker`

Rclone Trava ou Demora Muito

Possíveis causas: - Muitos arquivos em /DATA (45k+): normal, pode levar horas na primeira execução. - Limite de API do OneDrive atingido: aumentar `--timeout` ou reduzir `--transfers` e `--checkers`. - Rede instável: verificar conexão de internet.

Teste com dry-run:

```
rclone sync /DATA onedrive:/RCLONE --dry-run --fast-list --progress
```

Logs Crescem Muito

Logs já são compactados e rotacionados. Se ainda crescerem: - Reduzir `--log-level` em rclone (usar NOTICE em vez de INFO). - Aumentar intervalo de `--stats` (ex: `--stats 5m`).

Serviços CasaOS Não Reiniciam

Verificar:

```
systemctl list-unit-files --type=service | grep casaos
```

Se nenhum serviço começar com casaos, remover essa seção do script.

Variáveis de Ambiente (Opcional)

Pode-se passar variáveis ao executar:

```
SRC="/backup/data" DEST="onedrive:/backup" ./onedrive-rclone.sh
```

Ou editar diretamente no script no início da seção # CONFIGURAÇÕES.

Notas de Performance

Primeira Execução

- Pode levar várias horas (toda a árvore de /DATA será listada e enviada).
- É normal ver logs grandes (100MB+) nesta fase.

Execuções Subsequentes (Diárias)

- Levam **minutos** (apenas deltas).
- Logs reduzidos (1-10MB compactados).
- Uso de banda muito menor.

Otimizações Futuras

- Se /DATA crescer muito (>500GB), considerar:
 - Aumentar `--chunk-size`.
 - Reduzir `--stats` (ex: 5m em vez de 1m).
 - Usar `--transfers 4 --checkers 4` (reduzir paralelismo).
-

Suporte e Referências

- **Documentação Rclone:** <https://rclone.org/>
 - **OneDrive com Rclone:** <https://rclone.org/onedrive/>
 - **Docker CLI:** <https://docs.docker.com/engine/reference/commandline/>
 - **Systemd:** <https://www.freedesktop.org/software/systemd/man/>
-

Histórico de Versões

Versão	Data	Alterações
1.0	2026-01-23	Versão inicial com docker stop/start, rclone sync, logs compactados e rotação.

Última atualização: 23 de janeiro de 2026