

Rclone 文档部分翻译

作者: [HaujetZhao](#)

原文链接: <https://ld246.com/article/1600853705300>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 id="介绍">介绍</h2>

<h3 id="关于-rclone">关于 rclone</h3>

<p>Rclone 是一个用于管理云存储的命令程序。它特性丰富，可作为的云服务提供商的 web 界面代品。有超过 40 个云存储产品支持 rclone，包括 S3 对存储、商用&个人云存储服务、和标准传输协议。</p>

<p>对于 Unix 命令例如 rsync、cp、mv、mount、ls、ncdu、tree、rm、cat，Rclone 有强大的效命令。Rclone 的相似的语法也支持 shell 管道，并且有 <code>--dry-run</code> 保护选项。可以以命令行使用、以脚本使用、以 API 使用。</p>

<p>用户们把 rclone 称作“云存储的瑞士军刀”和“来自魔法的无可替代的科技”</p>

<p>Rclone 可以看管好你的数据，它时刻保存时间戳并校验 checksums。在有限的带宽下传输；可断续连接、或者配额对象可被重启，直到最后一个完好的文件被传输。你可以检查你文件的完整性。可能的时候，rclone 会让服务端传输文件，以最小化从服务端一个位置移动到另一个位置所需的带宽无需占用本地存储。</p>

<p>虚拟后台包装本地和云文件，用以 加密，缓存，分块 和 合并。</p>

<p>Rclone 在 Windows、macOS、linux、FreeBSD 上 挂载 任何本地、云、虚拟文件系统作为磁盘，并且能以 SFTP，HTTP，WebDAV，FTP 和 DLNA 协议提供文件夹在线访问服务。</p>

<p>Rclone 是受 rsync 启发，用 go 语言编写的成熟、开源的软件。友好的社区熟悉各种用例。官方 Ubuntu、Debian、Fedora、Brew、Chocolatey 仓库都有 rclone。如果想要最新版本，推荐从 downloading from rclone.org 下载。</p>

<p>Rclone 已在 Linux、Windows、Mac 上被广泛使用。第三方开发者使用 rclone 命令行或 API 发了创新性的备份、恢复、GUI、商用 进程解决方案。</p>

<p>Rclone 是一个可以干云存储交互重活的工具。</p>

<h3 id="Rclone-可以为你做什么-">Rclone 可以为你做什么？</h3>

<p>Rclone 可以帮你：</p>

备份（和加密）文件到云存储

从云存储恢复（和解密）文件

镜像云数据到另一个云服务，或者与本地存储协作

迁移数据到云上，或者在不同云存储商之间

挂载多个、加密的、缓存的或分开的云存储为一个磁盘

使用 lsf，ljson，size

ncdu 分析和统计云上的数据
 统一文件系统，使多个本地或者云存储为展示一个盘

 <h3 id="特性">特性</h3>

 传输

 为了保证文件完整性，时刻校验 MD5, SHA1 哈希值
 在文件中保存时间戳
 操作可以在任何时刻重启
 可以在两个不同的云服务商间工作
 可以使用多线程下载到硬盘

 Copy 复制新的或者更改了的文件云存储
 Sync 单向同步，使得两边相等
 Move 将本地文件移动到云存储并在校验完成后删除本地文件
 Check 检查哈希值以查看有没有失或新增的文件
 Mount 挂载云存储为磁盘
 Serve 通过 HTTP/WebDav/FTP/SFTP/dlna 协议提供文件在线访问服务
 实验性的网页 GUI

 <h3 id="支持的云服务">支持的云服务</h3>
 <p>(其实还有许多服务商，基于例如 WebDAV or S3 的公用协议，就不在这里列出了)</p>

 1Fichier Home Config
 Alibaba Cloud (Aliyun) Object Storage System (OSS) Home Config
 Amazon Drive (See note) See note

=<https://ld246.com/forward?goto=https%3A%2F%2Fwww.amazon.com%2Fclouddrive>" target="_blank" rel="nofollow ugc"> Home Config

Amazon S3 Home Config

Backblaze B2 Home Config

Box Home Config

Ceph Home Config

Citrix ShareFile Home Config

C14 Home Config

DigitalOcean Spaces Home Config

Dreamhost Home Config

Dropbox Home Config

FTP Home Config

Google Cloud Storage Home Config

Google Drive Home Config

Google Photos Home Config

k" rel="nofollow ugc"> Config

- HTTP Home Config
- Hubic Home Config
- Jottacloud Home Config
- IBM COS S3 Home Config
- Koofr Home Config
- Mail.ru Cloud Home Config
- Memset Memstore Home Config
- Mega Home Config
- Memory Home Config
- Microsoft Azure Blob Storage Home Config
- Microsoft OneDrive Home Config
- Minio Home Config
- Nextcloud Home Config
- OVH Home Config

OpenDrive Home onfig

OpenStack Swift Home Config

Oracle Cloud Storage Home Config

ownCloud Home Config

pCloud Home Config

premiumize.me Home Config

put.io Home Config

QingStor Home Config

Rackspace Cloud Files Home Config

rsync.net Home Config

Scaleway Home Config

Seafile Home Config

SFTP Home Config

StackPath Home Config

rget="_blank" rel="nofollow ugc"> Config
SugarSync Home Config
Tardigrade Home Config
Tencent Cloud Object Storage (COS) Home Config
Wasabi Home Config
WebDAV Home Config
Yandex Disk Home Config
The local filesystem Home Config

<p>相关链接</p>

主页
源代码和问题追踪的 Github 页面
Rclone 论坛
下载

<h2 id="安装">安装</h2>
<h3 id="安装-">安装</h3>
<p>Rclone 是一个 go 程序，所以只有一个二进制可执行文件。</p>
<h3 id="快速开始">快速开始</h3>

下载 相关的二进制文件
从下载的压缩包提取出 <code>rclone</code> 或者 <code>rclone.exe</code> 二进制文件
运行 <code>rclone config</code> 以配置。详情参照 rclone config docs 。

<h2 id="使用">使用</h2>

<h3 id="配置">配置</h3>

<p>首先，你需要配置 rclone。因为对象存储系统的配置比较复杂，这些被放在了 config 文件中。详情查看 <code>--config</code> 一节，以学习如何找到 config 文件并选择它的保存位置) </p>

<p>最简单配置的的方法就是用 config 选项运行 rclone: </p>

<pre><code class="highlight-chroma">rclone config

</code></pre>

<p>查看下面的链接以获得详细的不同服务商的配置步骤: </p>

1Fichier

Alias

Amazon Drive

Amazon S3

Backblaze B2

Box

Cache

Chunker - 从其它远程云存储透明地分割大文件

Citrix ShareFile

Crypt - 加密其它的远程云存储

DigitalOcean Spaces

Dropbox

FTP

Google Cloud Storage

Google Drive

Google Photos

HTTP

Hubic

Jottacloud / GetSky.no

Koofr

Mail.ru Cloud

Mega

- Memory
- Microsoft Azure Blob Storage
- Microsoft OneDrive
- OpenStack Swift / Rackspace Cloudfiles / Memset Memstor
- OpenDrive
- Pcloud
- premiumize.me
- put.io
- QingStor
- Seafile
- SFTP
- SugarSync
- Tardigrade
- Union
- WebDAV
- Yandex Disk
- The local filesystem

<h3 id="用法">用法</h3>

<p>Rclone 将文件夹保持原来结构从一个存储系统系统同步到另一个系统。 </p>

<p>它的语法类似于这样： </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl" style="background-color: #ffffcc;">语法: [选项] 子命令 &lt;参数> &lt;参数...&gt;</span></span></code></pre>
```

</code></pre>

<p>源 和 目标路径 都由你所给的【 config 文件中保存存储系统名字加上子路径】指定，例如：“dirve:myfolder” 表示查看保存的 Google Drive 中的 my older 路径。 </p>

<p>你可以在 config 文件中定义许多存储路径。 </p>

<p>请在学习 rclone 的时候使用 <code>-i</code> / <code>--interactive</code> 标志以避免意外数据损失。 </p>

<h3 id="子命令">子命令</h3>

<p>rclone 使用子命令系统，例如： </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl" style="background-color: #ffffcc;">rclone ls remote:path ## 列出远程路径下的文件</span></span></code>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl" style="background-color: #ffffcc;">rclone copy /local
```

path remote:path ## 复制 /local/path 到远程路径

```
</span></span><span class="highlight-line"><span class="highlight-cl">rclone sync -i /local/path remote:path ## 同步 /local/path 到远程路径
```

```
</span></span></code></pre>
```

<p>主要的 rclone 命令（按最常使用的频率）：</p>

rclone config - 进入交互式配置面

rclone copy - 从 source 复制到 dest，跳过已复制的

rclone sync - 使 source 和 dest 步，只修改 dest

rclone move - 将文件从 source 动到 dest

rclone delete - 删除一个路径下文件

rclone purge - 删除路径和它包的文件

rclone mkdir - 创建文件夹（如它不存在）

rclone rmdir - 删除文件夹

rclone rmdirs - 删除目录下所有空的路径

rclone check - 检查 source 和 dest 文件是否一致

rclone ls - 列出所有对象（包括大小路径）

rclone lsd - 列出路径下的文件夹、器

rclone lsl - 列出所有对象（包括大小修改时间和路径）

rclone md5sum - 用路径下所有物体生成一个 md5 文件

rclone sha1sum - 用路径下所有物体生成一个 sha1 文件

rclone size - 返回总文件大小和文数量

rclone space - 返回源和目的地的总空间使用量

[>rclone version](https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_version%2F) - 显示版本号
rclone cleanuprclone deduperclone authorizerclone catrclone copytorclone genautocompleterclone gendocsrclone listremotesrclone mountrclone movetorclone obscurerclone cryptcheckrclone about
<p>查阅 命令索引 以获得完整列表</p>
<h3 id="复制单个文件">复制单个文件</h3>
<p>rclone 一般同步或复制文件夹, 不过, 如果 source 指向一个文件, rclone 会只复制这个文件。dest 必须指向一个文件夹 - 如果目标路径不是文件夹, rclone 会给出 <code>Failed to create file system for "remote:file": is a file not a directory</code> 的错误提示。</p>
<p>例如, 假设你有一个远程地址 remote, 其中有个文件叫 <code>test.jpg</code>, 然后你以用下述命令复制: </p>
<pre><code class="highlight-chroma">rclone copy remote:test.jpg /tmp/download</code></pre>
<p>T 文件 <code>test.jpg</code> 会补复制到 <code>/tmp/download</code>.</p>

这也等同于当 `/tmp/files` 只包含下述这一个文件时

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">test.jpg</span></span></code></pre>
```

指定：

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy --files-from /tmp/files remote:/tmp/download</span></span></code></pre>
```

我们的推荐当复制单个文件时使用 `copy`，而不是 `sync`。它们作用相似，不过 `copy` 会使用更少的内存。

远程路径的语法

输入 rclone 的远程路径语法如下：

-path-to-dir

这指向的是本地系统的路径

在 Windows 上只有本地路径才可以用 `\` 代替 `/`，非本地路必须使用 `/`

这些路径不必要在开头加上 `/` - 如果没有 `/` 的话表示的是前路径的相对路径

remote:path/to/dir

这表示 config 文件中定义（使用 `rclone config` 定义）的 `remote:` 上的 `path/to/dir` 路径

remote:/path/to/dir

在大多数后端这等同于 `remote:path/to/dir` 并且这种形势更受推荐。在极少的远程系统（ftp、sftp、dropbox 商用版）上这会指向一个不同的路径。在这些系统上，开头没有 `/` 的路径一指向 “home” 路径，而加上 `/` 则表示根目录。

:backend:path/to/dir

这是创建远程路径的一种高级形式。`backend` 应当是后端（config 文件中的 `type`）的名称或前缀，所有的后端配置都应该被提供在命令行（或者环境变量中）。

例如：

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone lsd --http-url https://pub.rclone.org :http:</span></span></code></pre>
```

要列出 `https://pub.rclone.org/` 根目录下的所有文件夹。

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone lsf --http-url https://example.com :http:path/to/dir</span></span></code></pre>
```

要列出 `https://example.com/path/to/dir/` 目录下的所有文件夹和文件。

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy --http-url https://example.com :http:path/to/dir /tmp/dir</span></span></code></pre>
```

要复制 `https://example.com/path/to/dir/` 目录下的所有文件夹和文件到 `/tmp/dir`。

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy --sftp-host example.com :sftp:path/to/dir /tmp/dir</span></span></code></pre>
```

要使用 sftp 从 `example.com` 的相对路径 `path/to/dir` 下载文件和文件夹到 `/tmp/dir`。

有效的 remote 名字

- remote 名字只可以包含 `0-9`、`A-Z`、`a-z`、`_`、`-` 和空格
- remote 名字不能以 `-` 开头

引用和-shell" >引用和 shell</h3>

<p>当你在电脑上输入命令的时候，你是在 shell 中使用命令行。这以操作系统特定的方式解释各种符。</p>

<p>这里要讲一些陷阱，以帮助不熟悉 shell 规则的用户</p>

Linux---OSX" >Linux / OSX</h4>

<p>如果你输入的名字包含空格或 shell 元字符（例如 <code>*</code>, <code>?</code>, <code>\$</code>, <code>'</code>, <code>"</code> 等），则必须用单引号将其引起来。默认使用引号 <code>'</code>。</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy 'Important files?' remote:backup</span></span></code></pre>
```

<p>如果要发送 <code>'</code>，则需要使用 <code>"</code>，例如</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy "O'Reilly Reviews" remote:backup</span></span></code></pre>
```

<p>引用元字符的规则很复杂，如果您需要完整的详细信息，则必须查阅 Shell 的手册页。</p>

Windows" >Windows</h4>

<p>如果您的名字中有空格，则需要将其放在 <code>"</code> 中，例如</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy "E:\folder name\folder name\folder name" remote:backup</span></span></code></pre>
```

<p>如果单独使用根目录，则不要引用该根目录（有关原因，请参见 #464），例如：</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy E:\ remote:backup</span></span></code></pre>
```

复制名称中带有--的文件或目录" >复制名称中带有 <code>:</code> 的文件或目录</h3>

<p>rclone 使用 <code>:</code> 标记一个 remote 名称。但是，这是非 Windows 操作系统中的效文件名组件之一。远程名称解析器只会搜索直到第一个 <code>/</code> 的 <code>:</code>，因此，如果您需要对这样的文件或目录执行操作，请使用以 <code>/</code> 开头的完整路径，使用 <code>.</code> 作为当前目录的前缀。</p>

<p>因此，要将名为 <code>sync:me</code> 的目录同步到名为 <code>remote:</code> 的程目录，请使用</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone sync -i ./sync:me remote:path</span></span></code></pre>
```

<p>或者</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone sync -i /full/path/to/sync:me remote:path</span></span></code></pre>
```

服务器端复制" >服务器端复制</h3>

<p>大多数远程服务器（但不是全部-请参阅 [概述]）支持服务器端复制。</p>

<p>这意味着，如果您要将一个文件夹复制到另一个文件夹，rclone 不会下载所有文件再重新上传们；rclone 将指示服务器将它们复制到位。</p>

<p>例如</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone copy s3:oldbucket s3:newbucket</span></span></code></pre>
```

<p>将 <code>oldbucket</code> 的内容复制到 <code>newbucket</code> 而无需下载并重新上传。</p>

<p>当服务器不支持服务器端复制时，rclone 会下载并重新上传。</p>

<p>服务器端复制会被 <code>sync</code> 和 <code>copy</code> 使用，并且在使用 <code>-v</code> 标志时将在日志中区分出来。如果服务器端不支持直接移动，<code>move</code> 令也会使用服务器端复制功能，这是通过发送服务器端复制命令，然后删除原来文件来完成的，这样下载和重新上传要快得多。</p>

<p>仅当 remote 名称相同时，才尝试服务器端复制。</p>

<p>此功能可以被用在编写脚本，进行高效地进行备份时。例如</p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone sync -i remote:current-backup remote:previous-backup</span></span><span class="highlight-line"><span class="highlight-cl">rclone sync -i /pat</span></span></code></pre>
```

<p>Rclone 有许多选项可以控制其行为。</p> <p>带有参数的选项可以通过两种方式传递值： <code>--option=value</code> 或 <code>--option value</code>，但是，布尔（<code>true</code> / <code>false</code>）选项的行为与他选项略有不同，因为 <code>--boolean</code> 会将选项设置为 <code>true</code>，而没该选项时，则将其设置为 <code>false</code>，也可以指定 <code>--boolean=false</code> 或 <code>--boolean=true</code>，请注意，<code>--boolean false</code> 无效 - 这将会被析为 <code>--boolean</code> 和 <code>false</code> 两个额外的命令行参数。</p> <p>使用 TIME 的选项，使用 go time 解析器解析。例如 <code>300ms</code>、<code>-1.5h</code>、<code>2h45m</code>，有效的单位有：<code>ns</code>、<code>us</code>（<code>µs</code>）、<code>ms</code>、<code>s</code>、<code>m</code>、<code></code></p> <p>使用 SIZE 的选项默认用 kByte，可以用 b, k, M, G, T, P 后缀。</p> <p>使用 <code>copy</code>、<code>sync</code>、<code>move</code> 时，任何补盖、删除的文件会被移到这个文件夹并保持它的层级结构。</p> <p>如果设了 <code>--suffix</code>，那么被移动的文件会加上这个后缀。如果有相同路径的文，那么会覆盖掉。</p> <p>例如：</p> ``` <code class="highlight-chroma">rclone sync /path/to/local remote:current --backup-dir remote:old</code></pre> ``` <p>会同步 <code>/path/to/local</code> 到 <code>remote:current</code>，但任何被更或删除的文件会被移动到 <code>remote:old</code></p> <p>如果你从一个脚本运行 rclone，或许你想将今天的日期作为 <code>--backup-dir</code> 文件夹的名字，用于存储旧文件，或者你想将今天的日期传递到 <code>--suffix</code> 选项。</p> <p>详见 <code>--compare-dest</code> 和 <code>--copy-dest</code>。</p> <p>向外连接时，绑定的本地 ip。可以是 ipv4 地址 (1.2.3.4)、ipv6 地址 (1234::789A)、域名地址。如果域名无法被解析，或者解析出多个 ip，那会返回错误。</p> <p>控制带宽。两种方式：一个限制、时间表。默认是 0，表示没有限制。</p> ``` <code class="highlight-chroma">--bwlimit 10M</code></pre> ``` ``` <code class="highlight-chroma">--bwlimit "08:00,512 12:00,10M 13:00,512 18:00,30M 23:00,off"</code></pre> ``` ``` <code class="highlight-chroma">--bwlimit "Mon-00:00,512 Fri-23:59,10M Sat-10:00,1M Sun-20:00,off"</code></pre> ``` <p>列出文件不受带宽限制选项影响。</p> 原文链接：[Rclone 文档部分翻译](#)

<p>这里的单位是 Byte/s, 不是 bit/s。</p>
<h4 id="--bwlimit-file-BANDWIDTH-SPEC">--bwlimit-file=BANDWIDTH_SPEC</h4>
<p>略</p>
<h4 id="--buffer-size-SIZE">--buffer-size=SIZE</h4>
<p>每个 --transfer 用多少内存来缓冲</p>
<h4 id="--check-first">--check-first</h4>
<p>略</p>
<h4 id="--checkers-N">--checkers=N</h4>
<p>同步运行多少个比较任务。默认是 8</p>
<h4 id="-c----checksum">-c, --checksum</h4>
<p>只用文件大小来判断是否是同一个文件。</p>
<h4 id="--compare-dest-DIR">--compare-dest=DIR</h4>
<p>使用 sync, copy, move 时, 将 DIR 中的文件向 dest 中的文件比较每个文件, 如果有的文件相, 就不复制。</p>
<p>用 sync 时, 必须使用与 dest 相同的 remote。比较路径不能与目标路径重叠。</p>
<h4 id="--config-CONFIG-FILE">--config=CONFIG_FILE</h4>
<p>一般配置文件在 home 路径下的 <code>.config/rclone/rclone.conf</code> 中, 当 <code>XDG_CONFIG_HOME</code> 被设置时, 路径变为 <code>\$XDG_CONFIG_HOME/rclone/rclone.conf</code> </p>
<p>如果执行 rclone 的目录下有个 <code>rclone.conf</code> 文件, 会优先用它。</p>
<p>运行 <code>rclone config file</code> 可以看到默认配置文件在哪里。</p>
<h4 id="--contimeout-TIME">--contimeout=TIME</h4>
<p>设置连接 timeout。</p>
<pre><code class="highlight-chroma">5s 10m 3h30m</code></pre>
<p>默认是 1min。</p>
<p>如果超过 timeout 还没得到服务器响应, 就会认为无法连接。</p>
<h4 id="--copy-dest-DIR">--copy-dest=DIR</h4>
<p>使用 sync, copy, move 时, 将 DIR 中的文件向 dest 中的文件比较每个文件, 如果有的文件相, 就在服务端将文件从 DIR 复制到 dest。对增量备份有用。</p>
<h4 id="--dedupe-mode-MODE">--dedupe-mode MODE</h4>
<p>dedupe 命令下用。可选 <code>interactive</code>, <code>skip</code>, <code>first</code>, <code>newest</code>, <code>oldest</code>, <code>rename</code>。默认是 <code>interactive</code>。</p>
<h4 id="--disable-FEATURE-FEATURE--">--disable FEATURE,FEATURE,...</h4>
<p>例如: </p>
<pre><code class="highlight-chroma">--disable move,copy</code></pre>
<p>查询: </p>
<pre><code class="highlight-chroma">--disable help</code></pre>
<p>debug 用的</p>
<h4 id="-n----dry-run">-n, --dry-run</h4>
<p>尝试运行一下, 但不做出任何实质性修改, 用来看一下, rclone 都会做哪些动作。对 sync 的检很有用。</p>
<h4 id="--expect-continue-timeout-TIME">--expect-continue-timeout=TIME</h4>
<p>当 request 有 "Expect: 100-continue" header 时, 指定完全写 request header 后, 等服务器的第一个 response header 的等待时间。</p>
<h4 id="--ignore-case-sync">--ignore-case-sync</h4>
<p>忽略文件名的大小写。</p>
<h4 id="--ignore-checksum">--ignore-checksum</h4>

<p>忽略比较文件数量来判断是否传输出了问题。一般若出问题时，会提示：“corrupted on transfer” </p>
<h4 id="--ignore-existing">--ignore-existing</h4>
<p>跳过所有已存在的目标文件</p>
<h4 id="--ignore-size">--ignore-size</h4>
<p>忽略比较文件大小来判断是不是不同的文件。</p>
<h4 id="-l----ignore-times">-l, --ignore-times</h4>
<p>不管目标文件是什么状态，都上传所有文件。</p>
<h4 id="--immutable">--immutable</h4>
<p>当 source 和 destination 文件都是不可修改的。只新建和删除。如果两个文件不匹配，会出错
<code>Source and destination exist but do not match: immutable file modified</code>.</p>

<h4 id="--leave-root">--leave-root</h4>
<p>rmdir 时，不删除根目录。</p>
<h4 id="--log-file-FILE">--log-file=FILE</h4>
<p>将日志输出到 FILE</p>
<h4 id="--log-format-LIST">--log-format LIST</h4>
<p>可配置项： <code>date</code>, <code>time</code>, <code>microseconds</code>, <code>longfile</code>, <code>shortfile</code>, <code>UTC</code>. 默认是 “<code>date</code>, <code>time</code>” .</p>
<h4 id="--log-level-LEVEL">--log-level LEVEL</h4>
<p>DEBUG 等同于 -vv</p>
<p>INFO 等同于 -v</p>
<p>NOTICE 默认</p>
<p>ERROR 等同于 -q</p>
<h4 id="--use-json-log">--use-json-log</h4>
<p>用 json 格式的 log</p>
<h4 id="--low-level-retries-NUMBER">--low-level-retries NUMBER</h4>
<h4 id="--max-backlog-N">--max-backlog=N</h4>
<h4 id="--max-delete-N">--max-delete=N</h4>
<p>限制删除文件的最大数量</p>
<h4 id="--max-depth-N">--max-depth=N</h4>
<p>递归层级。（对 purge 无效）</p>
<p>例如 <code>rclone --max-depth 1 ls remote:path</code> 只会列出第一层级的文件。</p>
<p>历史遗留原因，lsd 默认用 --max-depth 1</p>
<h4 id="--max-duration-TIME">--max-duration=TIME</h4>
<p>超过指定时间，停止 schedule</p>
<h4 id="--max-transfer-SIZE">--max-transfer=SIZE</h4>
<h4 id="--modify-window-TIME">--modify-window=TIME</h4>
<p>当检查一个文件是否被修改过，这是允许的最大的时间差。</p>
<p>默认是 1ns 。对 os x 是 1s</p>
<h4 id="--multi-thread-cutoff-SIZE">--multi-thread-cutoff=SIZE</h4>
<p>如果下载的文件到本地后台大于这个大小，用多线程下载。默认是 250M</p>
<p>只对目标是本地时有用。</p>
<h4 id="--multi-thread-streams-N">--multi-thread-streams=N</h4>
<p>多线程时，线程的最大数量。默认是 4。</p>
<h4 id="--no-check-dest">--no-check-dest</h4>
<p>不检查 destination 。会有以下效果：</p>

目标文件不被列出，最少使用 api
文件总会被传输
在允许的网盘上，会产生重复文件
推荐使用 --retries 1 ，否则失败时，需要重新传输所有文件。

<h4 id="--no-gzip-encoding">--no-gzip-encoding</h4>
<h4 id="--no-traverse">--no-traverse</h4>
<h4 id="--no-update-modtime">--no-update-modtime</h4>
<h4 id="--order-by-string">--order-by string</h4>
<p>控制文件处理顺序，类别有：</p>

size
name
modtime

<p>可以在逗号后附加的方法有：</p>

ascending 或 asc
descending 或 desc

<p>默认方法是 ascending</p>
<p>例子：</p>

--order-by size,desc
--order-by modtime,ascending

<p>当然有一点点局限和例外，先不译了</p>
<h4 id="--password-command-SpaceSepList">--password-command SpaceSepList</h4>
<h4 id="-P----progress">-P, --progress</h4>
<p>提供上传和下载进度。每 500ms 更新一次</p>
<h4 id="-q----quiet">-q, --quiet</h4>
<p>尽可能不做输出</p>
<h4 id="--retries-int">--retries int</h4>
<p>重试次数。默认是 3。设为 1 即为禁用。</p>
<h4 id="--retries-sleep-TIME">--retries-sleep=TIME</h4>
<p>重试前的等待时间。默认是 0</p>
<h4 id="--size-only">--size-only</h4>
<p>只检查大小。不检查时间。</p>
<h4 id="--stats-TIME">--stats=TIME</h4>
<h4 id="--stats-file-name-length-integer">--stats-file-name-length integer</h4>
<h4 id="--stats-log-level-string">--stats-log-level string</h4>
<h4 id="--stats-one-line">--stats-one-line</h4>
<h4 id="--stats-one-line-date">--stats-one-line-date</h4>
<h4 id="--stats-one-line-date-format">--stats-one-line-date-format</h4>
<h4 id="--stats-unit-bits-bytes">--stats-unit=bits|bytes</h4>
<h4 id="--suffix-SUFFIX">--suffix=SUFFIX</h4>
<p>用 sync, copy, move 时，将任何本来要被覆盖或删除的文件添加一个后缀。</p>
<p>与 --backup-dir 搭配使用</p>
<h4 id="--suffix-keep-extension">--suffix-keep-extension</h4>
<p>用 --suffix=SUFFIX 时，保持后缀名。</p>
<h4 id="--syslog">--syslog</h4>
<h4 id="--syslog-facility-string">--syslog-facility string</h4>
<h4 id="--tpslimit-float">--tpslimit float</h4>
<h4 id="--tpslimit-burst-int">--tpslimit-burst int</h4>
<h4 id="--track-renames">--track-renames</h4>
<p>默认使用 sync 时，如果你在本地重命名了一个文件，rclone 会先删除远程文件，再上传新文。
</p>
<p>用上这个，就可以追踪重命名的文件。</p>
<p>用 size 和 hash 来匹配。</p>

<p>与 --no-traverse 不兼容</p>
 <p>与 --delete-before 不兼容</p>
 <h4 id="--delete--before-during-after-">--delete-(before,during,after)</h4>
 <p>决定同步时，远程的多余文件是在上传的什么阶段被删掉的。默认是 after</p>
 <h4 id="--fast-list">--fast-list</h4>
 <h4 id="--timeout-TIME">--timeout=TIME</h4>
 <p>如果 idle 超过指定时间，认为连接中断</p>
 <h4 id="--transfers-N">--transfers=N</h4>
 <p>最大同步传输的文件数量。默认是 4</p>
 <h4 id="-u----update">-u, --update</h4>
 <p>跳过远程修改时间比本地还新的文件。</p>
 <h4 id="--use-mmap">--use-mmap</h4>
 <h4 id="--use-server-modtime">--use-server-modtime</h4>
 <h4 id="-v---vv---verbose">-v, -vv, --verbose</h4>
 <h4 id="-V---version">-V, --version</h4>
 <p>打印版本</p>
 <h3 id="SSL-TLS-选项">SSL/TLS 选项</h3>
 <h4 id="--ca-cert-string">--ca-cert string</h4>
 <h4 id="--client-cert-string">--client-cert string</h4>
 <h4 id="--client-key-string">--client-key string</h4>
 <h4 id="--no-check-certificate-true-false">--no-check-certificate=true/false</h4>
 <h3 id="配置文件加密">配置文件加密</h3>
 <p>你的配置文件包含了网盘的登陆信息，也就是说，需要保证 .rclone.conf 的安全。</p>
 <p>如果你的工作环境不是安全的，你可以给你的配置文件加密</p>
 <p>每次要用这个 config 文件时都要输入密码</p>
 <p>执行 <code>rclone config</code> </p>
 <pre><code class="highlight-chroma">>rclone configCurrent remotes:e) Edit existing remoten) New remoted) Delete remotes) Set configuration passwordq) Quit confige/n/d/s/q></code></pre>
 <p>跳转 <code>s</code> </p>
 <pre><code class="highlight-chroma">e/n/d/s/q> sYour configuration is not encrypted.If you add a password, you will protect your login information to cloud services.a) Add Passwordq) Quit to main menu</code></pre>
 <p>选 <code>a</code> </p>
 <pre><code class="highlight-chroma">a/q> aEnter NEW config</code></pre>

ration password:

```
</span></span><span class="highlight-line"><span class="highlight-cl">password:
</span></span><span class="highlight-line"><span class="highlight-cl">Confirm NEW pas
word:
</span></span><span class="highlight-line"><span class="highlight-cl">password:
</span></span><span class="highlight-line"><span class="highlight-cl">Password set
</span></span><span class="highlight-line"><span class="highlight-cl">Your configuration
is encrypted.
```

```
</span></span></code></pre>
```

<p>如果丢失密码，就永远无法找回里面的数据。</p>

<p>使用 XSalsa20 和 Poly1305 加密和验证。</p>

<p>如果你的环境是安全的，你可以设置 <code>RCLONE_CONFIG_PASS</code> 环境变量为你密码，就不用每次都输密码了。</p>

<p>你可以写这样一个脚本保存为 <code>set-rclone-password</code>：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight"
cl">#!/bin/echo Source this file don't run it
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">read -s RCLONE_
CONFIG_PASS
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">export RCLONE_
CONFIG_PASS
```

```
</span></span></code></pre>
```

<p>之后只要运行 <code>source set-rclone-password</code> 再设置密码，就可以把密码写进环境变量。</p>

--dump flag,flag,flag</h5> --dump headers</h5> --dump bodies</h5> --dump requests</h5> --dump responses</h5> --dump auth</h5> --dump filters</h5> --dump goroutines</h5> --dump openfiles</h5> <p>可用：</p> <p>详见 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Ffilterin 原文链接：[Rclone 文档部分翻译](#)

%2F" target="_blank" rel="nofollow ugc">过滤章节.</p>
<h3 id="远程控制">远程控制</h3>
<p>如果要查看如何远程控制：</p>

<code>--rc</code>
任何开头带 <code>--rc-</code> 的命令

<p>参见 远程控制章节.</p>
<h3 id="日志">日志</h3>
<h3 id="退出码">退出码</h3>
<h4 id="退出码如下->">退出码如下：</h4>

<code>0</code> - 成功
<code>1</code> - 语法或语法错误
<code>2</code> - 无法归类的错误
<code>3</code> - 没找到文件夹
<code>4</code> - 没找到文件
<code>5</code> - 临时性错误 (多试几次可能会解决) (重新尝试可解决的错误)
<code>6</code> - 不是太严重了，错误 (例如 dropbox 的 461 错误) (重新尝试也无法解决的错误)
<code>7</code> - 致命错误 (多次尝试也无法解决，例如账户被停用) (致命错误)
<code>8</code> - 传输超限 - 由 --max-transfer 设置的限制达到了
<code>9</code> - 操作成功，但是没有文件被传输

<h3 id="环境变量">环境变量</h3>
<p>rclone 可以被环境变量控制配置。你可以用环境变量来设置一些选项的默认值。</p>
<h4 id="选项">选项</h4>
<p>每个选项都可以由环境变量设定默认值</p>
<p>要找到环境变量的名字，首先，取选项的完整名字，这样开头的两个减号去掉，将 <code>--</code> 改为 <code>_</code>，将所有小写变为大写，在开头加上 <code>RCLONE_</code></p>
<p>例如，要设置 <code>--stats 5s</code>，就设置环境变量 <code>RCLONE_STATS=5s</code></p>
<p>又或者要设置磁盘使用回收站，就设置 <code>RCLONE_DRIVE_USE_TRASH=true</code></p>
<h4 id="配置文件">配置文件</h4>
<p>你可以在配置文件中的单个 remote 基础上，为其值设置默认值。</p>
<p>如果要使用此功能，则需要找到所需的配置项的名称。</p>
<p>最简单的方法是手动运行 <code>rclone config</code>，然后在配置文件中查看值是什么（以通过查看 <code>rclone help</code> 中的 <code>--config</code> 帮助来找到该配置文件）</p>
<p>要找到环境变量的名称，您需要设置，将 <code>RCLONE_CONFIG_</code> + remote 名称 + <code>_</code> + 配置文件中的选项的名称（大写）。</p>
<p>例如，要在不带配置文件的情况下配置名为 <code>mys3:</code> 的 S3 远程服务器（使用设置环境变量的 unix 方式）：</p>
<pre><code class="highlight-chroma">\$ export RCLONE_CONFIG_MYS3_TYPE=s3
\$ export RCLONE_CONFIG_MYS3_ACCESS_KEY_ID=XXX
\$ export RCLONE_CONFIG_MYS3_SECRET_ACCESS_KEY=XXX
\$ rclone lsd MYS3:
 -1 2016-09-1 12:54:21 -1 my-bucket

</code>
</pre>


```

</span></span><span class="highlight-line"><span class="highlight-cl">$ rclone listremot
s | grep mys3
</span></span><span class="highlight-line"><span class="highlight-cl">mys3:
</span></span></code></pre>
<p>注意，如果你想通过创建变量的方式创建一个 remote，就必须用上述方式创建 <code>..._TYP
</code> 变量。</p>
<h4 id="优先权">优先权</h4>
<p>后端配置的各种不同方法会以下述顺序读取，并且第一人有效值会被最终采用：</p>
<ul>
<li>命令行中提供的值，例如 <code>--drive-use-trash</code>。</li>
<li>特定于 remote 的环境变量，例如 <code>RCLONE_CONFIG_MYREMOTE_USE_TRASH</cod
>（请参见上文）。</li>
<li>特定于后端的环境变量，例如 <code>RCLONE_DRIVE_USE_TRASH</code>。</li>
<li>配置文件，例如 <code>use_trash = false</code></li>
<li>默认值，例如 <code>>true</code> - 这个值无法更改</li>
</ul>
<p>因此，如果在配置文件中同时提供了 “--drive-use-trash”，并且设置了环境变量 “RCLONE_
RIVE_USE_TRASH”，则命令行标志将优先。</p>
<p>对于非后端配置，顺序如下：</p>
<ul>
<li>命令行中提供的值，例如 <code>--stats 5s</code></li>
<li>环境变量，例如 <code>RCLONE_STATS=5s</code></li>
<li>默认值，例如 <code>1m</code> - 这个值无法更改。</li>
</ul>
<h4 id="其他环境变量">其他环境变量</h4>
<ul>
<li><code>RCLONE_CONFIG_PASS</code> 设置你的配置文件的访问密码（详情见<a href="htt
s://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fdocs%2F%23configuration-enc
ryption" target=" blank" rel="nofollow ugc">配置文件加密</a>）</li>
<li><code>HTTP_PROXY</code>、<code>HTTPS_PROXY</code>、<code>NO_PROXY</c
ode>（或者这些项的小写）。</li>
<li>在 https 请求上 <code>HTTPS_PROXY</code> 相对于 <code>HTTP_PROXY</code> 有
高优先权</li>
<li>环境变量值可以是一个完整的 URL 或者 <code>host[:port]</code>（当使用 http 时）</li>
</ul>
<li><code>RCLONE_CONFIG_DIR</code> - rclone 在子进程中<strong>设置</strong> 此变
，指向保存配置文件的目录，以便使用 config 文件。</li>
</ul>
<h2 id="GUI-实验性-">GUI（实验性）</h2>
<p>Rclone 可以提供基于 Web 的 GUI（图形用户界面）。目前这还处于实验阶段，因此情况可能
发生变化。</p>
<p>在终端中运行此命令，rclone 将下载，然后在 Web 浏览器中显示 GUI：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone rcd --rc-web-gui
</span></span></code></pre>
<p>这将产生这样的日志，并且为了服务 GUI，rclone 需要继续运行：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">2019/08/25 11:40:14 NOTICE: A new release for gui is present at https://github.com/rclon
/rclone-webui-react/releases/download/v0.0.6/currentbuild.zip
</span></span><span class="highlight-line"><span class="highlight-cl">2019/08/25 11:40:
4 NOTICE: Downloading webgui binary. Please wait. [Size: 3813937, Path : /home/USER/.cach
/rclone/webgui/v0.0.6.zip]

```

2019/08/25 11:40:6 NOTICE: Unzipping

2019/08/25 11:40:6 NOTICE: Serving remote control on http://127.0.0.1:5572/

```
</span></span></code></pre>

<p>假设您正在计算机上本地运行 rclone, 您可以将 rclone 和 GUI 分开运行 - 有关详细信息, 请参下文。</p>



<p>如果您希望检查更新, 则可以在命令行中添加 <code>--rc-web-gui-update</code> </p>



<p>如果发现 GUI 损坏, 则可以通过添加 <code>--rc-web-gui-force-update</code> 来强制更新。</p>



<p>默认情况下, rclone 将打开您的浏览器。添加 <code>--rc-web-gui-no-open-browser</code> 以禁用此功能。</p>



### <p>GUI 打开后, 您将看到具有总体概述的仪表板。</p> <p>在左侧, 您将看到一系列查看按钮, 您可以单击以下按钮: </p> - <li>Dashboard - 主要概述</li> - <li>Configs - 检查和创建新配置</li> - <li>Explorer - 查看、下载文件, 和上传文件到云存储系统</li> - <li>Backend - 查看或更改后端配置</li> - <li>Log out</li> </ul> <p>(更多文档和入门视频即将发布!) </p> <p>当你运行 <code>rclone rcd --rc-web-gui</code> 之后会发生这些事: </p> - <li>rclone 开始运行, 但只运行远程控制 API ("rc")</li> - <li>API 绑定到了本地地址, 并有一个自动生成的用户名和密码。</li> - <li>如果 api 包缺失, 那么 rclone 就会下载它</li> - <li>rclone 会从 api 包通过 api 的相同端口开始服务文件</li> - <li>rclone 会打开浏览器并使用 <code>login_token</code> 所以它会直接登陆进去</li> </ul> <p><code>rclone rcd</code> 可以使用任何 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Frc%2F%23supported-parameters" target="_blank" rel="nofollow ugc">rc page</a> 中记录的选项</p> <p>选项 <code>--rc-web-gui</code> 是下述的简写: </p> - <li>如果需要的话就下载 web GUI</li> - <li>检查验证</li> - <li><code>--rc-user gui</code> </li> - <li><code>--rc-pass &lt;random password></code> </li> - <li><code>--rc-serve</code> </li> </ul> <p>这些选项可以被强制覆盖</p> <p>另外请参考 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcmmands%2Frclone_rcd%2F" target="_blank" rel="nofollow ugc">rclone rcd documentation</a>.</p> <p>例如使用以下选项, GUI 可以通过 ssl 使用 httpasswd 文件在一个公共端口上服务: </p> - <li><code>--rc-web-gui</code> </li> - <li><code>--rc-addr :443</code> </li> - <li><code>--rc-htpasswd /path/to/httpasswd</code> </li>


```

- <code>--rc-cert /path/to/ssl.crt</code>
- <code>--rc-key /path/to/ssl.key</code>

如果你想在位于 <code>/rclone</code> 的代理之后运行 GUI，你可以设置以下选项: </p></div><div data-bbox="77 146 643 299" data-label="Text"><code>--rc-web-gui</code> <code>--rc-baseurl rclone</code> <code>--rc-htpasswd /path/to/htpasswd</code> <p>或者不是用 htpasswd，你只是想单个用户和密码地使用: </p><code>--rc-user me</code> <code>--rc-pass mypassword</code> </div><div data-bbox="77 299 912 363" data-label="Text"><h3 id="项目">项目</h3><p>开发地址: rclone/rclone-webui-react repository.</p></div><div data-bbox="77 363 912 409" data-label="Text"><p>欢迎反馈和贡献 :-)</p><p>如果有任何问题可以到这里提问: rclone forum.</p></div><div data-bbox="77 409 912 502" data-label="Text"><h3 id="笔记">笔记</h3><p>我在使用 <code>rclone rcd --rc-web-gui</code> 时会遇到无法读取和添加 config 的问题所以手动运行 GUI。</p><p>首先，到 rclone/rclone-webui-react 下载最新的 release，解压，得到网站文件夹。</p></div><div data-bbox="77 502 916 595" data-label="Text"><p>然后运行如下命令: </p><pre><code class="highlight-chroma">rclone rcd "D:\Users\Haujet\Desktop\build" --rc-user gui --rc-pass pass</code></pre><p>这表示: 运行远程控制，对 <code>D:\Users\Haujet\Desktop\build</code> 下的 web-gui 提供服务，登陆用户名 <code>gui</code>，登陆密码 <code>pass</code></p></div><div data-bbox="77 595 446 609" data-label="Text"><h2 id="Rclone-命令">Rclone 命令</h2></div><div data-bbox="77 609 313 624" data-label="Text"><h3 id="总览">总览</h3></div><div data-bbox="77 624 900 656" data-label="Text"><p>这是 rclone 中所有命令的索引。运行 <code>rclone --help</code> 可以查看该命令的帮助:</p></div><div data-bbox="77 656 906 887" data-label="Table"><table><thead><tr><th>命令</th><th>描述</th></tr></thead><tbody><tr><td>rclone</td><td>显示 rclone 的命令、选项、后端</td></tr><tr><td></td></tr></tbody></table></div><div data-bbox="715 936 929 951" data-label="Page-Footer"><p>原文链接: Rclone 文档部分翻译</p></div>

```

2Frclone_about%2F" target="_blank" rel="nofollow ugc">rclone about</a></td>
<td>获得 remote 的详情信息（例如使用量）</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_authorize%2F" target="_blank" rel="nofollow ugc">rclone authorize</a></td>
<td>remote 授权</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_backend%2F" target="_blank" rel="nofollow ugc">rclone backend</a></td>
<td>运行特定于后端的命令</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_cat%2F" target="_blank" rel="nofollow ugc">rclone cat</a></td>
<td>连接任意文件，并送到标准输出</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_check%2F" target="_blank" rel="nofollow ugc">rclone check</a></td>
<td>检查 source 和 remote 文件是否相同</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_cleanup%2F" target="_blank" rel="nofollow ugc">rclone cleanup</a></td>
<td>如果可以，清空 remote</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_config%2F" target="_blank" rel="nofollow ugc">rclone config</a></td>
<td>进入交互式配置程序</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_config_create%2F" target="_blank" rel="nofollow ugc">rclone config create</a></
d>
<td>用名字、类型、选项，创建一个新的 remote</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_config_delete%2F" target="_blank" rel="nofollow ugc">rclone config delete</a></
d>
<td>删除一个已存在的 remote name.</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_config_disconnect%2F" target="_blank" rel="nofollow ugc">rclone config disconnec
</a></td>
<td>从 remote 断开用户连接</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_config_dump%2F" target="_blank" rel="nofollow ugc">rclone config dump</a></t

```



```

>
<td>将配置文件以 JSON 输出</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_edit%2F" target="_blank" rel="nofollow ugc">rclone config edit</a></td>
<td>进入交互式配置页面</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_file%2F" target="_blank" rel="nofollow ugc">rclone config file</a></td>
<td>显示配置文件地址</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_password%2F" target="_blank" rel="nofollow ugc">rclone config password</a></td>
<td>更新已存在的 remote 的密码</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_providers%2F" target="_blank" rel="nofollow ugc">rclone config providers</a></td>
<td>以 json 形式列出所有的服务商和选项</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_reconnect%2F" target="_blank" rel="nofollow ugc">rclone config reconnect</a></td>
<td>重新对 remote 授权</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_show%2F" target="_blank" rel="nofollow ugc">rclone config show</a></td>
<td>打印配置文件或配置文件中的一个 remote</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_update%2F" target="_blank" rel="nofollow ugc">rclone config update</a></td>
<td>更新一个已存在的 remote 的选项</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_config_userinfo%2F" target="_blank" rel="nofollow ugc">rclone config userinfo</a></td>
<td>打印 remote 的已登录的用户信息</td>
</tr>
<tr>
<td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_copy%2F" target="_blank" rel="nofollow ugc">rclone copy</a></td>
<td>将文件从 source 复制到 dest, 跳过已复制的文件</td>
</tr>

```

```

<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_copyto%2F" target="_blank" rel="nofollow ugc">rclone copyto</a> </td>
<td>将文件从 source 复制到 dest，跳过已复制的文件</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_copyurl%2F" target="_blank" rel="nofollow ugc">rclone copyurl</a> </td>
<td>将 url 文件复制到 dest</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_cryptcheck%2F" target="_blank" rel="nofollow ugc">rclone cryptcheck</a> </td>
<td>检查已加密 remote 的完整性</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_cryptdecode%2F" target="_blank" rel="nofollow ugc">rclone cryptdecode</a> </t
>
<td>返回未加密的文件名</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_dedupe%2F" target="_blank" rel="nofollow ugc">rclone dedupe</a> </td>
<td>交互性的找到重复的文件名，然后删除或重命名他们</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_delete%2F" target="_blank" rel="nofollow ugc">rclone delete</a> </td>
<td>删除一个目录下的内容</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_deletefile%2F" target="_blank" rel="nofollow ugc">rclone deletefile</a> </td>
<td>删除 remote 的单个文件</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_genautocomplete%2F" target="_blank" rel="nofollow ugc">rclone genautocomplet
</a> </td>
<td>为一个给定的 shell 输出完成脚本</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_genautocomplete_bash%2F" target="_blank" rel="nofollow ugc">rclone genautoc
mplete bash</a> </td>
<td>输出 rclone 的 bash 完成脚本</td>
</tr>
<tr>
<td> <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands
2Frclone_genautocomplete_fish%2F" target="_blank" rel="nofollow ugc">rclone genautoco
mplete fish</a> </td>
<td>输出 rclone 的 fish 完成脚本</td>
</tr>

```

rclone genautocomplete zsh	输出 rclone 的 zsh 完成脚本
rclone gendocs	为 rclone 输出 markdown 形式的文档到指定目录
rclone hashsum	为指定目录的所有对象生成一个 hashsum 文件
rclone link	生成指向文件或文件夹的公共链接
rclone listremotes	列出配置文件中所有的 remote
rclone ls	列出指定目录中所有的对象以及其目录、大小
rclone lsd	列出目录中所有的文件夹、容器、桶
rclone lsf	以 remote:path 的格式列出文件夹内所有的对象
rclone lsjson	以 json 的格式列出文件夹内所有的对象
rclone lsl	将指定文件夹内的所有对象的修改时间、大小、路径一并列出
rclone md5sum	

<td>为指定目录的所有对象生成一个 md5sum 文件</td> </tr> <tr> <td>rclone mkdir</td> <td>如果路径不存在，则创建这个路径</td> </tr> <tr> <td>rclone mount</td> <td>在一个挂载点，将远程路径挂载为文件系统</td> </tr> <tr> <td>rclone move</td> <td>将文件从 source 移动到 dest</td> </tr> <tr> <td>rclone moveto</td> <td>将文件或文件夹从 source 移动到 dest</td> </tr> <tr> <td>rclone ncdu</td> <td>用一个基于文本的交互界面浏览 remote</td> </tr> <tr> <td>rclone obscure</td> <td>将 rclone.conf 中存储的密码变为 base64 格式</td> </tr> <tr> <td>rclone purge</td> <td>删除路径和它包含的文件</td> </tr> <tr> <td>rclone rc</td> <td>对正在运行的 rclone 运行一个命令</td> </tr> <tr> <td>rclone rcat</td> <td>复制标准输出到 remote 的一个文件</td> </tr> <tr> <td>rclone rcd</td> <td>运行 rclone，但只接收远程控制命令</td> </tr> <tr> <td><a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands

rclone rm	如果目录为空则移除
rclone rmdir	
rclone rmdirs	移除该目录下的所有空文件夹
rclone serve	用指定的协议搭建一个 remote 服务器
rclone serve dlna	通过 DLNA 协议搭建一个可访问的服务器指向 remote:path
rclone serve ftp	通过 ftp 协议搭建一个可访问的服务器指向 remote:path
rclone serve http	通过 http 协议搭建一个可访问的服务器指向 remote:path
rclone serve restdb	为了 restdb's REST API 搭建一个可访问的服务器指向 remote
rclone serve sftp	通过 sftp 协议搭建一个可访问的服务器指向 remote
rclone serve webdav	通过 webdav 协议搭建一个可访问的服务器指向 remote
rclone settier	改变 remote 的对象的存储类
rclone sha1sum	为指定目录的所有对象生成一个 sha1sum 文件

rclone size	打印目标路径的所有对象的总数量和总大小
rclone sync	将 source 同步到 dest, 只改动 dest 的文件
rclone touch	创建一个新文件或改变文件的修改时间
rclone tree	用文件数的形式列出远程路径所有文件
rclone version	显示版本号

id="rclone-mount">rclone mount</h3> 这里说下, 在 windows 上挂载需要安装 [winfsp](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Fbillziss-gh%2Fwinfsp%2Freleases) ``` <code class="highlight-chroma">https://github.com/billziss-gh/winfsp/releases/download/v1.7B1/winfsp-1.7.20038.msi</code></pre> ``` <code class="highlight-chroma">rclone mount OneDrive:/ H: -cache-dir E:\OneDrive -vfs-cache-mode writes</code></pre> 在一个挂载点, 将远程路径挂载为文件系统 id="概要">概要</h4> rclone mount 允许 Linux、FreeBSD、macOS 和 Windows 使用 FUSE 将任何 Rclone 的云存系统挂载为文件系统。 首先使用 rclone config 设置你的 remote。使用 rclone ls 等命令检查是否配置正确。 你可以用前台模式或后台（守护进程）模式运行挂载。挂载默认以前台模式运行, 使用 --daemon 选项以指定它用后台模式运行。后台模式只支持 Linux 和 OSX, 在 Windows 上只能运行前台模式 在 Linux/macOS/FreeBSD 上, 要确保 /path/to/local/mount 是一个已存的空文件夹: ``` <code class="highlight-chroma">rclone mount remote:path/to/files /path/to/local/mount</code></pre> 或者在 Windows 上, 要确保 X: 是一个未使用的盘符或者一个不存在的路径 ``` <code class="highlight-chroma"> ``` ``` ``` ```

```
cl">rclone mount remote:path/to/files X:
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">rclone mount rem
te:path/to/files C:\path\to\nonexistent\directory
```

```
</span></span></code></pre>
```

<p>当运行在后台模式时，用户需要手动停止挂载（见下文）</p>

<p>当程序在前台模式被结束，挂载会自动停止</p>

<p>挂载操作有可能失败，例如挂载点正忙。当这个发生时，用户就有责任手动停止挂载。</p>

<p>手动停止挂载：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">## Linux
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">fusermount -u /pa
h/to/local/mount
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">## OS X
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">umount /path/to/
ocal/mount
```

```
</span></span></code></pre>
```

<p>注意：从 rclone 1.52.2 开始，rclone mount 现在在某些平台上需要 Go 1.13 以上的版本，取决于所需的 FUSE 库。</p>

<p>要在 Windows 上运行 rclone mount，你需要下载和安装 WinFsp</p> <p>WinFsp 是一个开源的 Windows 文件系统，可以让 Windows 轻松地写用户层文件系统。它提供 FUSE 模拟层，rclone 同时使用后者与 cgofuse。两个包由 Bill Zissimopoulos 所写，他也给 rclone 在 Windows 上的 mount 的实现提供了莫大的帮助。</p> <p>注意，Administrator 创建的新盘对其它用户是不可见的（包括被提升为管理员权限的用户）。以如果你以管理员命令行启动了一个 Windows Drive，然后从 Explorer 访问，是看不见这个磁盘的</p> <p>最简单的方法是以普通用户的 cmd 窗口运行启动这块盘。或者从 SYSTEM 用户启动（使用 the WinFsp.Launcher infrastructure），这会创建每个人都有访问权的 Drive，或者你也可以使用 the nssm service manager</p> <p>默认，rclone 会将 remote 挂载为普通磁盘。不过，你也可以将它挂载为网络磁盘。</p> <p>不像其它系统，Windows 为网络磁盘提供一种不同的文件系统类型，Windows 和其它程序对网络磁盘普通磁盘时会有所不同：在网络磁盘中，许多 I/O 操作会被优化，因为考虑到了网络所带来高延迟和低稳定性。</p> <p>虽然许多人更偏好让网络硬盘以普通硬盘的形式挂载，但这可能引起问题。例如：程序在 Explore 中对挂载的 remote 执行操作时无法正常运行、卡住、发生错误。如果你经历过这些，那么请考虑将 clone remotes 挂载为网络磁盘，因为 Windows 认为普通磁盘应当速度快并且稳定，而网络磁盘就远不行。详情查看 限制一节。</p> <p>在 <code>mount</code> 后添加 <code>--fuse-flag --VolumePrefix=\server\share</code> 选项，如果你挂载了多个 remote 的话，将 <code>share</code> 替换为其它的名字，否则挂载会发生冲突，你的挂载文件系统会重叠。</p> <p>阅读更多关于磁盘映射的信息</p> 原文链接：[Rclone 文档部分翻译](#)

限制

如果没有使用 `--vfs-cache-mode`，那么就只能顺序写入，只能在读取的时候找。这就意味当文件在 rclone mount 上，又没有 `--vfs-cache-mode writes` 或 `--vfs-cache-mode full` 着许多应用无法工作，详见 [文件缓存](https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_mount%2F%23file-caching&target=_blank&rel=nofollow&ugc=)

基于 bucket 的 remote（例如 swift、S3、google 计算存储、B2、Hubic）不支持空文件的概念，所以如果空文件夹在缓存中被清除，那么就会清空。

目前只支持 Linux, FreeBSD, OS X and Windows

rclone mount 相较于 rclone sync/copy

文件系统期待百分百稳定，但是云存储远没有这样稳定。rclone sync/copy 执行时会多次尝试但是 rclone mount 在进行本地复制和上传时无法以相同的方式重试。查阅 [文件缓存](https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fcommands%2Frclone_mount%2F%23file-caching&target=_blank&rel=nofollow&ugc=) 以获得更多。

属性缓存

你可以使用 `--attr-timeout` 属性设置内核缓存文件夹属性（大小、修改时间的时间）。

默认是 1s，可以避免内核给 rclone 太多的 callback。

理论上，对于文件系统，0s 是可以在内核的控制外修改的最正确的数值。不过这会引来不少问题，如 [rclone 占用太多内存](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Frclone%2Fissues%2F2157&target=_blank&rel=nofollow&ugc=)，[rclone not serving files to amba](https://ld246.com/forward?goto=https%3A%2F%2Fforum.rclone.org%2Ft%2Frclone-1-39-s-1-40-mount-issue%2F5112&target=_blank&rel=nofollow&ugc=) 和 [列出文件夹占用太多时间](https://ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2Frclone%2Fissues%2F2095%23issuecomment-371141147&target=_blank&rel=nofollow&ugc=)。

内核可以缓存文件信息持续 `--attr-timeout` 指定的时间。在这个窗口，如果程文件改变长度，你可能会见到崩溃。它会最终以截断文件或者垃圾文件显示。使用 `--attr-timeout 1s` 不太可能会发生（也不一定就不发生）。你设置的 `--attr-timeout` 越高，就越有可能发生。所以 1s 是最佳的数值了。

如果你设置比较高（10s 或者 1m），那么内核会更少地给 rclone callback，会更高效，但也更可能崩溃。

If files don't change on the remote outside of the control of rclone then there is no chance of corruption.

This is the same as setting the attr_timeout option in mount.fuse.

过滤器

提醒下，rclone 过滤器也可以用在挂载点可见文件的子集上。

systemd

当让 rclone 作为 systemd service 运行时，可以使用 Type=notify。这时，在挂载点被成功配之后 service 会进行启动状态。在这个模式下，拥有 rclone mount service 的 units 会立即看见所文件和文件夹。

分块读取

`--vfs-read-chunk-size` 会启用从 source 分块读取文件，对于某些 remote，过递增的次序的请求，对于某些 remote，可以减少使用的下载配额。

当 `--vfs-read-chunk-size-limit` 被指定，并且比 `--vfs-read-chunk-size` 大，那么每个打开的文件的 chunk size 会增倍，直到达到指定的数值。-1 会禁用这个制，chunk size 会无限增长。

如果设置了 `--vfs-read-chunk-size` 和 `--vfs-read-chunk-size-limit`，接下来的部分就会被下载：0-100M, 100M-200M, 200M-300M, 300M-400M and so on。当指定了 `--vfs-read-chunk-size-limit 500M`，那么结果就是：0-100M, 100-300M, 300M-700M, 700M-1200M, 1200M-1700M and so on

只有在 `--vfs-cache-mode <full>` 时才会开始，因为如果设置了 `--vfs-cache-mode full`，那么文件就总是会在打开之间被复制到 vfs 缓存。

<h4 id="VFS---虚拟文件系统-Virtual-File-System-">VFS - 虚拟文件系统 (Virtual File System) </h4>

<p>这个命令使用了 VFS 层，这将 rclone 使用的云存储系统对象转换为像是磁盘文件系统的系统。</p>

<p>云存储对象有许多与磁盘文件不一样的属性 - 你无法扩展它们，也无法在中间增添内容，所以要 VFS 层来处理它。因为至今没有一个所谓正确的方法，所以有了许多选项 (如下所述) </p>

<p>VFS 层也使用了文件夹缓存 - 缓存文件和文件夹信息 (不包含数据) 在内存中。</p>

<h4 id="VFS-文件夹缓存">VFS 文件夹缓存</h4>

<p>使用 <code>--dir-cache-time</code> 选项，你可以控制后台文件夹应该多长时间刷新和不刷新，通过挂载做的更改会立刻出现或者使缓存无效</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">--dir-cache-time duration  缓存文件夹目录的时间 (默认是 5m0s)</span></span><span class="highlight-line"><span class="highlight-cl">--poll-interval duration  拉取变化之间等待的时间</span></span></code></pre>
```

<p>不过，在云端存储、网页交互界面、另外一个 rclone 上发生的更改，只有在文件夹缓存被清空才会更新，如果后端支持拉取，在拉取时间段之内会完成变动的更新。</p>

<p>你可以发送一个 <code>SIGHUP</code> 信号给 rclone，让他清空文件夹缓存，不管已等了多长时间。假设只有一个 rclone 正在运行，你可以这样重置缓存：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">kill -SIGHUP $(pidof rclone)</span></span></code></pre>
```

<p>如果你设置了 rclone 的 远程控制，那么你可以使用 rclone rc 更整个文件夹缓存：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone rc vfs/forget</span></span></code></pre>
```

<p>或者单个文件、文件夹：</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone rc vfs/forget file=path/to/file dir=path/to/dir</span></span></code></pre>
```

<h4 id="VFS-文件缓冲">VFS 文件缓冲</h4>

<p><code>--buffer-size</code> 决定了用多少内存提前缓冲数据。</p>

<p>每个打开的文件总会尽量保存一定量的数据在内存中，这部分缓存的数据会与打开的文件绑定，且不会被分享。</p>

<p>这个选项是每个打开的文件可占用内存的上限，这个缓冲只会将内存用于下载完成但是还没有被取的数据，如果缓冲为空，那么只会用一点点内存。</p>

<p>rclone 用于缓冲所使用的最大的内存取决于 <code>--buffer-size * open files</code> </p>

<h4 id="VFS-文件缓存">VFS 文件缓存</h4>

<p>这些选项控制了 VFS 文件缓存。为了使 vfs 层与普通文件系统兼容，文件缓存就是必要的。以些兼容性为代价，可以将这个选项关掉。</p>

<p>例如，如果你想要同时读取和写入一个文件，也就需要启用 vfs 缓存。详见下文。</p>

<p>注意 vfs 缓存与后端缓存是区别开的，有时候你可能发现需要一个、另一个，或者两者。</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">--cache-dir string  用作rclone缓存的文件夹</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-cache-mode CacheMode  缓存模式 off|minimal|writes|full (默认是 off)</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-cache-max-age duration  文件在缓存中的最长时间. (默认是 1h0m0s)</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-cache-max-size SizeSuffix  缓存文件夹的最大总大小. (默认是 off)</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-cache-poll-interval duration  拉取旧对象的时间间隔. (默认是 1m0s)</span></span></code></pre>
```

--vfs-write-back d
ration 当使用缓存时在使用文件后回写文件的间隔. (默认是 5s)
</code></pre>
<p>如果运行 rclone 时带上 <code>-vv</code>, 那么就会打印出文件缓存的位置。文件被存储
用户缓存文件区域, 这是由系统定义的, 不过可以被 <code>--cache-dir</code> 控制, 或者设置
境变量修改。</p>
<p>缓存有 4 种模式, 模式越高, 兼容性越强, 占用空间越大</p>
<p>注意文件只有在被关闭后等待了 <code>--vfs-write-back</code> 定义的时间后没有被访问
, 才会写回远程端。</p>
<p>如果使用了 <code>--vfs-cache-max-size</code>, 那么在两种情况下, 缓存会超过这个大
: 首先是只有每间隔 <code>--vfs-cache-poll-interval</code> 时间才会检查, 其次是已打开的文
无法被清除。</p>
<h5 id="--vfs-cache-mode-off">--vfs-cache-mode off</h5>
<p>这是默认模式, 在这种模式下, 软件会直接读写远程文件, 这就意味着有些操作无法进行: </p>

打开文件时无法同时读取和写入
写入模式打开的文件无法被查找
要以写权限打开的已存在的文件需要有 O_TRUNC
要以读权限打开的已存在的有 O_TRUNC 的文件只能以写权限打开
只有当文件支持 O_TRUNC 才能打开文件写入
文件打开模式 O_APPEND, O_TRUNC 不被支持, 会忽略
如果上传失败, 则无法重试

<h5 id="--vfs-cache-mode-minimal">--vfs-cache-mode minimal</h5>
<p>This is very similar to "off" except that files opened for read AND write will be buffered to
disk. This means that files opened for write will be a lot more compatible, but uses the minima
disk space.</p>
<p>These operations are not possible</p>

写入模式打开的文件无法被查找
要以写权限打开的已存在的文件需要有 O_TRUNC
文件打开模式 O_APPEND, O_TRUNC 不被支持, 会忽略
如果上传失败, 则无法重试

<h5 id="--vfs-cache-mode-writes">--vfs-cache-mode writes</h5>
<p>这种模式下打开文件读取时, 仍然是直接从远端读取, 写入文件会先被缓存到磁盘。</p>
<p>这个模式应该支持所有正常的文件系统操作。</p>
<p>如果上传失败, 他会在试验性的递增间隔最大达到 1 分钟进行重试。</p>
<h5 id="--vfs-cache-mode-full">--vfs-cache-mode full</h5>
<p>在这个模式下, 读写操作都被缓存到本地磁盘, 读取远程文件也会被写入到缓存。</p>
<p>在这个模式下, 缓存中的文件是稀疏文件, rclone 会追踪每一份他下载的文件</p>
<p>所以如果应用只读取文件的开头, rclone 就只会缓存这个文件的开头部分。这些文件在缓存中
以他们的原始大小存在的, 但是在这些文件中, 只有被下载的部分才有数据(稀疏文件)。</p>
<p>这个模式应该支持所有的正常文件系统操作, 否则与--vfs-cache-mode 写入相同。</p>
<p>当读取文件的时候 rclone 会读取 --buffer-size 加上超前读取 --vfs-read-ahead 大小的文件。-
buffer-size 被存在内存, --vfs-read-ahead 缓存在磁盘。</p>
<p>使用这个模式的时候, 我们推荐 --buffer-size 不要太大, 如果需要的话 --vfs-read-ahead 可
设置大点。</p>
<p>重要提醒 不是所有的文件系统都支持稀疏文件, 尤其是 FAT/exFAT 不支
。如果缓存文件夹是在一个不支持稀疏文件的硬盘上, rclone 就会表现非常糟糕, 并且当检测到的时
会在日志中记录一个错误。</p>
<h4 id="VFS-性能">VFS 性能</h4>
<p>这些选项可以用于出于性能或其它原因启用或禁用 vfs 的一些特性。</p>
<p>尤其是 S3 和 Swift 可以从 --no-modtime (或者用 --use-server-modtime 使用安静的效果)

选项受益很多，因为每次读取修改时间都要进行单独的请求。

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">--no-checksum  转或下载时不要比较大小
</span></span><span class="highlight-line"><span class="highlight-cl">--no-modtime
不要读取和写入修改时间（可以加速某些操作）
</span></span><span class="highlight-line"><span class="highlight-cl">--no-seek
允许文件中查找
</span></span><span class="highlight-line"><span class="highlight-cl">--read-only
读挂载
</span></span></code></pre>
```

当 rclone 从远端读取文件，他会分块读取。rclone 不是请求整个文件，而是读取指定的块。这有利的，因为某些云提供商将读取作为请求的所有数据，而不是所传递的所有数据。

rclone 会从 `--vfs-read-chunk-size` 开始持续倍增缓存大小，直到达到最大值 `--vfs-read-chunk-size-limit`。除被 `--vfs-read-chunk-size-limit` 设置为 off，这时缓存大小将没有限制。

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">--vfs-read-chunk-size SizeSuffix  分块读取源文件. (默认是 128M)
</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-read-chunk-s
ze-limit SizeSuffix  最大缓存大小 (默认是 "off")
</span></span></code></pre>
```

有时，rclone 不会按顺序传递读写操作，相比于查找，rclone 会直接等待一小段时间，让正确序恢复正常。这些选项只有在不使用磁盘缓存时有用。

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">--vfs-read-wait duration  Time to wait for in-sequence read before seeking. (default 20ms
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">--vfs-write-wait d
uration  Time to wait for in-sequence write before giving error. (default 1s)
```

```
</span></span></code></pre>
```

VFS 大小写敏感性

Linux 文件系统是大小写敏感的，两个文件名可能只有大小写区别，打开文件时，大小写必须正确。

文件系统在现代的 Windows 是大小写不敏感但是大小写保留的：虽然不管大小写都能打开一个件，但是文件名的大小写会被保留，且可供程序查询。在同一个文件夹内不允许两个文件名只有大小有区别。

通常在 macOS 文件系统是大小写不敏感的，虽然可以让他大小写敏感，但是这一般不是默认。

`--vfs-case-insensitive` 挂载选项控制 rclone 如何处理这两种情况。如果是 `false`，rclone 将原本文件名传递到挂载文件系统，如果为 `true`，rclone 会使用一个 **fixup** 手段，解释如下：

用户可能会指定打开、删除或重命名一个文件名，现在文件名原来挂载的文件系统中存储的不一。如果这个参数指向一个已经存在的名字完全一样的文件，那么就会使用磁盘上已经存在的文件大小。但是如果具有完全一样文件名的文件没有被找到，而有一个文件，其文件名只有大小写不同，这时 rclone 会明显地 fixup 这个名字。只有当一个存在的文件被请求时才会发生，由 rclone 新创建的文名大小写敏感性由底层挂载的文件系统控制。

注意，运行 rclone 的操作系统 (target) 的大小写敏感性可能和 rclone 挂载的文件系 (source) 的大小写敏感性不同。这个选项控制 **fixup** 是否被使用以满足 target 的需要。

如果这个选项在命令行中提供了，那么它的默认值就取决于运行 rclone 的系统：在 Windows 和 macOS 上为 true，在其他情况下为 false。如果这个选项在没有值的情况下被提供，那他的默认值是 true

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone mount remote:path /path/to/mountpoint [flags]
</span></span></code></pre>
```

选项-


```
<highlight-n">s</span><span class="highlight-p"></span><span class="highlight-w">  
/>daemon</span><span class="highlight-w">  
/>as</span><span clas  
="highlight-w"> </span><span class="highlight-n">a</span><span class="highlight-w"> <  
<span class="highlight-n">daemon</span><span class="highlight-w"> </span><spa  
class="highlight-p">( </span><span class="highlight-n">background</span><span class="  
ightlignt-w"> </span><span class="highlight-n">mode</span><span class="highlight-p">).  
/>  
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla  
s="highlight-w">    </span><span class="highlight-o">--</span><span class="highlight-n"  
>daemon</span><span class="highlight-o">-</span><span class="highlight-n">timeout<  
span><span class="highlight-w"> </span><span class="highlight-n">duration</span><sp  
n class="highlight-w">      </span><span class="highlight-n">Time</span><span clas  
="highlight-w"> </span><span class="highlight-n">limit</span><span class="highlight-w"  
 </span><span class="highlight-n">for</span><span class="highlight-w"> </span><span c  
ass="highlight-n">rclone</span><span class="highlight-w"> </span><span class="highligh  
-n">to</span><span class="highlight-w"> </span><span class="highlight-n">respond</sp  
n><span class="highlight-w"> </span><span class="highlight-n">to</span><span class="h  
ghlight-w"> </span><span class="highlight-n">kernel</span><span class="highlight-w">  
/><span class="highlight-p">( </span><span class="highlight-n">not</span><span class=
```

```

span class="highlight-o">>--</span><span class="highlight-n">time</span><span class="hi
hlight-w"> </span><span class="highlight-n">duration</span><span class="highlight-w">
</span><span class="highlight-n">Time</span><span class="highlight-w"> </spa
><span class="highlight-n">to</span><span class="highlight-w"> </span><span class="hi
hlight-n">cache</span><span class="highlight-w"> </span><span class="highlight-n">dir
ctory</span><span class="highlight-w"> </span><span class="highlight-n">entries</span>
<span class="highlight-w"> </span><span class="highlight-n">for</span><span class="hi
hlight-p">.</span><span class="highlight-w"> </span><span class="highlight-p">(</span>
<span class="highlight-n">default</span><span class="highlight-w"> </span><span class
"highlight-err">5</span><span class="highlight-n">m0s</span><span class="highlight-p"
)</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n"
>dir</span><span class="highlight-o">--</span><span class="highlight-n">perms</span>
span class="highlight-w"> </span><span class="highlight-n">FileMode</span><span class
"highlight-w"> </span><span class="highlight-n">Directory</span><span clas
="highlight-w"> </span><span class="highlight-n">permissions</span><span class="highl
ght-w"> </span><span class="highlight-p">(</span><span class="highlight-n">default</s
an><span class="highlight-w"> </span><span class="highlight-err">0777</span><span cla
s="highlight-p">)</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n"
>file</span><span class="highlight-o">--</span><span class="highlight-n">perms</span>
<span class="highlight-w"> </span><span class="highlight-n">FileMode</span><span class
="highlight-w"> </span><span class="highlight-n">File</span><span class="hi
hlight-w"> </span><span class="highlight-n">permissions</span><span class="highlight-
"> </span><span class="highlight-p">(</span><span class="highlight-n">default</span>
span class="highlight-w"> </span><span class="highlight-err">0666</span><span class="h
ghlight-p">)</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n"
>fuse</span><span class="highlight-o">--</span><span class="highlight-n">flag</span><
pan class="highlight-w"> </span><span class="highlight-n">stringArray</span><span clas
="highlight-w"> </span><span class="highlight-n">Flags</span><span class="h
ghlight-w"> </span><span class="highlight-n">or</span><span class="highlight-w"> </sp
n><span class="highlight-n">arguments</span><span class="highlight-w"> </span><span class="highlight-n">to</span><span class="highlight-w"> </span><span class="highlight-
">be</span><span class="highlight-w"> </span><span class="highlight-n">passed</span>
<span class="highlight-w"> </span><span class="highlight-n">direct</span><span class=
highlight-w"> </span><span class="highlight-n">to</span><span class="highlight-w"> </
pan><span class="highlight-nn">libfuse</span><span class="highlight-n">WinFsp</span>
<span class="highlight-p">.</span><span class="highlight-w"> </span><span class="highl
ght-n">Repeat</span><span class="highlight-w"> </span><span class="highlight-n">if</
pan><span class="highlight-w"> </span><span class="highlight-n">required</span><span class="highlight-p">.</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n"
>gid</span><span class="highlight-w"> </span><span class="highlight-n">uint32</span>
<span class="highlight-w"> </span><span class="highlight-n">Override</
pan><span class="highlight-w"> </span><span class="highlight-n">the</span><span clas
="highlight-w"> </span><span class="highlight-n">gid</span><span class="highlight-w">
</span><span class="highlight-n">field</span><span class="highlight-w"> </span><span class="highlight-n">set</span><span class="highlight-w"> </span><span class="highlight
n">by</span><span class="highlight-w"> </span><span class="highlight-n">the</span><

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```
s="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n">volname</span><span class="highlight-w"> </span><span class="highlight-n">string</span></span><span class="highlight-w"> </span><span class="highlight-n">Set</span></span><span class="highlight-w"> </span><span class="highlight-n">the</span></span><span class="highlight-w"> </span><span class="highlight-n">volume</span></span><span class="highlight-w"> </span><span class="highlight-n">name</span></span><span class="highlight-w"> </span><span class="highlight-p">(</span><span class="highlight-n">not</span></span><span class="highlight-w"> </span><span class="highlight-n">supported</span></span><span class="highlight-w"> </span><span class="highlight-n">by</span></span><span class="highlight-w"> </span><span class="highlight-n">all</span></span><span class="highlight-w"> </span><span class="highlight-n">OSe</span></span><span class="highlight-p">).</span><span class="highlight-w"> </span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-o">--</span><span class="highlight-n">write</span></span><span class="highlight-o">-</span><span class="highlight-n">back</span></span><span class="highlight-o">-</span><span class="highlight-n">cache</span></span><span class="highlight-w"> </span><span class="highlight-n">Makes</span></span><span class="highlight-w"> </span><span class="highlight-n">kernel</span></span><span class="highlight-w"> </span><span class="highlight-n">buffer</span></span><span class="highlight-w"> </span><span class="highlight-n">writes</span></span><span class="highlight-w"> </span><span class="highlight-n">before</span></span><span class="highlight-w"> </span><span class="highlight-n">sendi</span></span><span class="highlight-w"> </span><span class="highlight-n">them</span></span><span class="highlight-w"> </span><span class="highlight-n">to</span></span><span class="highlight-w"> </span><span class="highlight-n">rclone</span></span><span class="highlight-p">.</span><span class="highlight-w"> </span><span class="highlight-n">Without</span></span><span class="highlight-w"> </span><span class="highlight-n">this</span></span><span class="highlight-p"></span><span class="highlight-w"> </span><span class="highlight-n">writethrough</span></span><span class="highlight-w"> </span><span class="highlight-n">caching</span></span><span class="highlight-w"> </span><span class="highlight-n">is</span></span><span class="highlight-w"> </span><span class="highlight-n">used</span></span><span class="highlight-p">.</span><span class="highlight-w"> </span></span></span></code></pre>
<p>查阅 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fflags%2" target="_blank" rel="nofollow ugc">global flags page</a> 以查看没有在这里列出的全局选项。</p>
<h3 id="rclone-serve">rclone serve</h3>
<p>用指定的协议搭建服务，以供访问 remote。</p>
<h4 id="概要">概要</h4>
<p>rclone serve 被用于通过指定的协议搭建服务，以供访问 remote。这个命令需要使用子命令以定协议，例如：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone serve http remote:</span></span></code></pre>
<p>每个子命令都有他自己的选项，你可以查看他们的帮助：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone serve &lt;protocol> [opts] &lt;remote> [flags]</span></span></code></pre>
<h4 id="选项--">选项</h4>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">-h, --help  help for serve</span></span></code></pre>
<p>查阅 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fflags%2" target="_blank" rel="nofollow ugc">global flags page</a> 以查看没有在这里列出的全局选项。</p>
<h4 id="另外参阅">另外参阅</h4>
```


 rclone - 显示命令帮助、选项和后端
 rclone serve dlna - 通过 DLNA 协议提供访问 remote:path 的服务
 rclone serve ftp - 通过 sftp 协议提供访问 remote:path 的服务
 rclone serve http - 通过 http 协议提供访问 remote:path 的服务
 rclone serve restic - 为 restic's REST API 提供访问 remote 的服务
 rclone serve sftp - 通过 sftp 协议提供访问 remote 的服务
 rclone serve webdav - 通过 webdav 协议提供访问 remote 的服务

<h3 id="rclone-serve-dlna">rclone serve dlna</h3>
<p>通过 <code>DLNA</code> 协议提供访问 <code>remote:path</code> 的服务</p>
<h4 id="概要--">概要</h4>
<p><code>rclone serve dlna</code> 是一个服务于 rclone remote 存储的媒体文件的 DLNA 媒体服务端，许多设备，例如 Xbox 和 PlayStation，可以在 LAN 下自动发现这个服务器，并从上播放视、音频。VLC 也是支持的。服务发现使用 UDP 多重广播包（UDP multicast packets）（SSDP），此只在 LAN 下工作。</p>
<p>Rclone 会列出 remote 中的所有文件，而不会基于媒体格式或后缀名过滤文件。另外，不提供转码支持。这意味着一些播放器可能会显示他们不能正确播放的文件。</p>
<h4 id="服务器选项">服务器选项</h4>
<p><code>--addr</code> 指定监听的地址和端口 例如：<code>--addr 1.2.3.4:8000</code> <code>--addr :8080</code> 监听所有 ip。</p>
<p><code>--name</code> 可以选择一个友好的服务器名字，默认是 <code>rclone(hostname)</code> </p>
<p><code>--log-trace</code> 和 <code>-vv</code> 结合起来可以为所有的 UPNP 传输开启外的 debug 记录。</p>
<h3 id="rclone-serve-ftp">rclone serve ftp</h3>
<p>通过 <code>FTP</code> 协议提供访问 <code>remote:path</code> 的服务</p>
<h4 id="概要---">概要</h4>
<p><code>rclone serve ftp</code> 实施一个基本的 <code>ftp</code> 服务，通过 <code>ftp</code> 协议提供访问 <code>remote</code> 的服务。他可以通过 ftp 客户端、浏览器进行浏览，你也可以先建一个 <code>ftp</code> 类型的 <code>remote</code> 去读写他。</p>
<h4 id="服务器选项-">服务器选项</h4>
<p><code>--addr</code> 指定监听的地址和端口 例如：<code>--addr 1.2.3.4:8000</code> <code>--addr :8080</code> 监听所有 ip。默认只监听 <code>localhost</code>。用 <code>0</code> 可以让系统自动选择端口</p>
<h5 id="认证">认证</h5>
<p>默认服务无需登陆</p>
<p>你可以用 <code>--user</code> 和 <code>--pass</code> 选项设置一个用户</p>
<h3 id="rclone-serve-http">rclone serve http</h3>
<p>通过 <code>HTTP</code> 协议提供访问 <code>remote:path</code> 的服务</p>
<h4 id="概要----">概要</h4>

`rclone serve http` 实施一个基本的 `http` 服务，通过 `http` 协议提供访问 `remote` 的服务。他可以通过浏览器进行浏览，你也可以先建一个 `http` 类型的 `remote` 去读写他。

你可以使用 filter 选项（例如 `--include`、`--exclude`）去制服务哪些文件

这个服务会 log errors，用 `-v` 以查看 logs。

`--bwlimit` 被用于文件传输，使用 `--stats` 控制统计打印。

服务器选项

`--addr` 指定监听的地址和端口 例如：`--addr 1.2.3.4:8000`
`--addr :8080` 监听所有 ip。默认只监听 `localhost`。用 `0` 可以让系统自动选择端口

`--server-read-timeout` 和 `--server-write-timeout` 控制传输读写时间。

`--max-header-bytes` 设置在 http 头接收的最大字节数

`--baseurl` 控制服务的 url 前缀。默认从根。如果用了 `--baseurl "/rclone"` 那么 rclone 会从 `"/rclone/"` 开始服务。对于使用代理很有用。rclone 默认会插入 `/`，所以 `--baseurl "rclone"` `--baseurl "/rclone"` 和 `--baseurl "/rclone/"` 效果是一样的

`--template` 允许用户为 http 和 webdav 服务功能指定自定义标记模板。服务器将以下要在模板中使用的标记导出到服务器页面：

参数	描述
.Name	文件或文件夹的全路径
.Title	Directory listing of .Name
.Sort	The current sort used. This is changeable via ?sort= parameter
	Sort Options: namedirfirst,name,size,time (default namedirfirst)
.Order	The current ordering used. This is changeable via ?order= parameter
	Order Options: asc,desc (default asc)

```

<td>.Query</td>
<td>Currently unused.</td>
</tr>
<tr>
<td>.Breadcrumb</td>
<td>Allows for creating a relative navigation</td>
</tr>
<tr>
<td>-- .Link</td>
<td>The relative to the root link of the Text.</td>
</tr>
<tr>
<td>-- .Text</td>
<td>The Name of the directory.</td>
</tr>
<tr>
<td>.Entries</td>
<td>Information about a specific file/directory.</td>
</tr>
<tr>
<td>-- .URL</td>
<td>The 'url' of an entry.</td>
</tr>
<tr>
<td>-- .Leaf</td>
<td>Currently same as 'URL' but intended to be 'just' the name.</td>
</tr>
<tr>
<td>-- .IsDir</td>
<td>Boolean for if an entry is a directory or not.</td>
</tr>
<tr>
<td>-- .Size</td>
<td>Size in Bytes of the entry.</td>
</tr>
<tr>
<td>-- .ModTime</td>
<td>The UTC timestamp of an entry.</td>
</tr>
</tbody>
</table>
<h5 id="认证">认证</h5>
<p>默认服务无需登陆</p>
<p>你可以用可设置多个用户的 htpasswd 文件, 或只用 <code>--user</code> 和 <code>--pass</code> 选项设置一个用户.</p>
<p>用 <code>--htpasswd /path/to/htpasswd</code> 提供 htpasswd 文件。它使用标准 apach 格式, 支持 MD5, SHA1 和 Bcrypt 用于基本的验证。</p>
<p>创建一个 htpasswd 文件: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">touch htpasswd
</span></span><span class="highlight-line"><span class="highlight-cl">htpasswd -B htpas
wd user
</span></span><span class="highlight-line"><span class="highlight-cl">htpasswd -B htpas
wd anotherUser

```


</code></pre>

<p>可以在 rclone 运行的时候更新 htpasswd 文件。</p>

<p>使用 <code>--realm</code> 设置授权 realm.</p>

<h5 id="SSL-TLS">SSL/TLS</h5>

<p>默认用 http。如果你想用 https，需要用 <code>--cert</code> 和 <code>--key</code> 签。如果你想做客户端的验证，还需要用 <code>--client-ca</code></p>

<p><code>--cert</code> 应当是 PEM 编码的 certificate 或连起来的一串有 CA 的 certificate</p>

<p><code>--key</code> 应当是 PEM 编码的私钥</p>

<p><code>--client-ca</code> 应当是 PEM 编码的用户证明</p>

<h3 id="rclone-serve-restic">rclone serve restic</h3>

<p>为 restic's REST API 提供访问 remote 的服务</p>

<h4 id="概要-----">概要</h4>

<p><code>rclone serve restic</code> 实施一个基本的 <code>restic's REST backend API</code> 服务，通过 <code>http</code> 协议提供访问 <code>remote</code> 的服务。这可允许 restic 使用 rclone 让 restic 不直接支持的云服务提供商作为一个数据存储器。</p>

<p>Restic 是一个用于做备份的命令行工具。</p>

<p>这个服务器会记录错误，使用 <code>-v</code> 查看 logs。</p>

<p><code>--bwlimit</code> 被用于文件传输，使用 <code>--stats</code> 控制统计打印。</p>

<h4 id="为-restic-的使用配置-rclone">为 restic 的使用配置 rclone</h4>

<p>首先 从你选择的云服务提供商设置 1 个 remote</p>

<p>在你设置好 remote 之后，检查它是否运行，例如使用 <code>rclone lsd remote:</code>，能你的 remote 不叫 remote，替换为你的那个名字就行。</p>

<p>然后启动 rclone restic server: </p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">rclone serve restic -v remote:backup</span></span></code></pre>
```

<p>这个服务会运行在 <code>localhost:8080</code>，你可以使用 <code>--addr</code> 改地址。</p>

<p>你或许希望开机就启动这个服务。</p>

<h4 id="为使用-rclone-而配置-restic">为使用 rclone 而配置 restic</h4>

<p>现在你可以 跟随 restic 指南 设置好 restic</p>

<p>注意你需要 restic 版本大于 0.8.2，来与 rclone 交互</p>

<p>对于上述例子，你需要使用 <code>http://localhost:8080/</code> 作为 rest 服务器的 url</p>

<p>例如: </p>

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ export RESTIC_REPOSITORY=rest:http://localhost:8080/</span></span><span class="highlight-line"><span class="highlight-cl">$ export RESTIC_</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl">ASSWORD=yourpassword</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl">$ restic init</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl">created restic bac</span></span><span class="highlight-line"><span class="highlight-cl">end 8b1a4b56ae at rest:http://localhost:8080/</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl"></span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl"></span></span><span class="highlight-line"><span class="highlight-cl">Please note that k</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl">nowledge of your password is required to access</span></span></pre>
```

```
<span class="highlight-line"><span class="highlight-cl">the repository. Los</span></span><span class="highlight-line"><span class="highlight-cl">ng your password means that your data is</span></span></pre>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">irrecoverably lost.
</span></span><span class="highlight-line"><span class="highlight-cl">$ restic backup /p
th/to/files/to/backup
</span></span><span class="highlight-line"><span class="highlight-cl">scan [/path/to/file
/to/backup]
</span></span><span class="highlight-line"><span class="highlight-cl">scanned 189 direc
ories, 312 files in 0:00
</span></span><span class="highlight-line"><span class="highlight-cl">[0:00] 100.00% 38
128 MiB / 38.128 MiB 501 / 501 items 0 errors ETA 0:00
</span></span><span class="highlight-line"><span class="highlight-cl">duration: 0:00
</span></span><span class="highlight-line"><span class="highlight-cl">snapshot 45c8fdd
saved
</span></span></code></pre>
<h5 id="多个仓库">多个仓库</h5>
<p>提醒一下，你可以使用这个终端 host 多个仓库，通过在 url 后边添加文件夹路径，注意这些 <strong>必须</strong> 以 <code></code> 结束。例如：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ export RESTIC_REPOSITORY=rest:http://localhost:8080/user1repo/
</span></span><span class="highlight-line"><span class="highlight-cl">## backup user1 s
uff
</span></span><span class="highlight-line"><span class="highlight-cl">$ export RESTIC_R
EPOSITORY=rest:http://localhost:8080/user2repo/
</span></span><span class="highlight-line"><span class="highlight-cl">## backup user2 s
uff
</span></span></code></pre>
<h5 id="私有仓库">私有仓库</h5>
<p><code>--private-repos</code> 选项可以被用于将用户限制为以 <code>/&lt;username&gt;
</code> 路径开头的存储库。</p>
<h4 id="服务器选项---">服务器选项</h4>
<p><code>--addr</code> 指定监听的地址和端口 例如：<code>--addr 1.2.3.4:8000</code>
<code>--addr :8080</code> 监听所有 ip。默认只监听 <code>localhost</code> 。用 <code>
0</code> 可以让系统自动选择端口</p>
<p><code>--server-read-timeout</code> 和 <code>--server-write-timeout</code> 控制
传输读写时间。</p>
<p><code>--max-header-bytes</code> 设置在 http 头接收的最大字节数</p>
<p><code>--baseurl</code> 控制服务的 url 前缀。默认从根。如果用了 <code>--baseurl "/rcl
one"</code> 那么 rclone 会从 <code>"/rclone/"</code> 开始服务。对于使用代理很有用。rclone
默认会插入 <code>/</code>，所以 <code>--baseurl "rclone"</code> <code>--baseurl "/rc
lone"</code> 和 <code>--baseurl "/rclone/"</code> 效果是一样的</p>
<p><code>--template</code> 允许用户为 http 和 webdav 服务功能指定自定义标记模板。服
器将以下要在模板中使用的标记导出到服务器页面：</p>
<table>
<thead>
<tr>
<th>参数</th>
<th>描述</th>
</tr>
</thead>
<tbody>
<tr>
<td>.Name</td>
<td>文件或文件夹的全路径</td>
</tr>
<tr>

```

```

<td>.Title</td>
<td>Directory listing of .Name</td>
</tr>
<tr>
<td>.Sort</td>
<td>The current sort used. This is changeable via ?sort= parameter</td>
</tr>
<tr>
<td></td>
<td>Sort Options: namedirfirst,name,size,time (default namedirfirst)</td>
</tr>
<tr>
<td>.Order</td>
<td>The current ordering used. This is changeable via ?order= parameter</td>
</tr>
<tr>
<td></td>
<td>Order Options: asc,desc (default asc)</td>
</tr>
<tr>
<td>.Query</td>
<td>Currently unused.</td>
</tr>
<tr>
<td>.Breadcrumb</td>
<td>Allows for creating a relative navigation</td>
</tr>
<tr>
<td>-- .Link</td>
<td>The relative to the root link of the Text.</td>
</tr>
<tr>
<td>-- .Text</td>
<td>The Name of the directory.</td>
</tr>
<tr>
<td>.Entries</td>
<td>Information about a specific file/directory.</td>
</tr>
<tr>
<td>-- .URL</td>
<td>The 'url' of an entry.</td>
</tr>
<tr>
<td>-- .Leaf</td>
<td>Currently same as 'URL' but intended to be 'just' the name.</td>
</tr>
<tr>
<td>-- .IsDir</td>
<td>Boolean for if an entry is a directory or not.</td>
</tr>
<tr>
<td>-- .Size</td>
<td>Size in Bytes of the entry.</td>

```

```

</tr>
<tr>
<td>-- .ModTime</td>
<td>The UTC timestamp of an entry.</td>
</tr>
</tbody>
</table>
<h5 id="认证--">认证</h5>
<p>默认服务无需登陆</p>
<p>你可以用可设置多个用户的 htpasswd 文件, 或只用 <code>--user</code> 和 <code>--pass
/code> 选项设置一个用户.</p>
<p>用 <code>--htpasswd /path/to/htpasswd</code> 提供 htpasswd 文件。它使用标准 apach
格式, 支持 MD5, SHA1 和 Bcrypt 用于基本的验证。</p>
<p>创建一个 htpasswd 文件: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">touch htpasswd
</span></span><span class="highlight-line"><span class="highlight-cl">htpasswd -B htpas
wd user
</span></span><span class="highlight-line"><span class="highlight-cl">htpasswd -B htpas
wd anotherUser
</span></span></code></pre>
<p>可以在 rclone 运行的时候更新 htpasswd 文件。</p>
<p>使用 <code>--realm</code> 设置授权 realm.</p>
<h5 id="SSL-TLS-">SSL/TLS</h5>
<p>默认用 http。如果你想用 https , 需要用 <code>--cert</code> 和 <code>--key</code>
签。如果你想做客户端的验证, 还需要用 <code>--client-ca</code></p>
<p><code>--cert</code> 应当是 PEM 编码的 certificate 或连起来的一串有 CA 的 certificate</
>
<p><code>--key</code> 应当是 PEM 编码的私钥</p>
<p><code>--client-ca</code> 应当是 PEM 编码的用户证明</p>
<h5 id="选项---">选项</h5>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">--addr string          IPaddress:Port or :Port to bind server to. (default "localhost:
080")
</span></span><span class="highlight-line"><span class="highlight-cl">--append-only
disallow deletion of repository data
</span></span><span class="highlight-line"><span class="highlight-cl">--baseurl string
Prefix for URLs - leave blank for root.
</span></span><span class="highlight-line"><span class="highlight-cl">--cert string
SSL PEM key (concatenation of certificate and CA certificate)
</span></span><span class="highlight-line"><span class="highlight-cl">--client-ca string
Client certificate authority to verify clients with
</span></span><span class="highlight-line"><span class="highlight-cl">-h, --help
help for restic
</span></span><span class="highlight-line"><span class="highlight-cl">--htpasswd string
htpasswd file - if not provided no authentication is done
</span></span><span class="highlight-line"><span class="highlight-cl">--key string
SSL PEM Private key
</span></span><span class="highlight-line"><span class="highlight-cl">--max-header
bytes int      Maximum size of request header (default 4096)
</span></span><span class="highlight-line"><span class="highlight-cl">--pass string
Password for authentication.
</span></span><span class="highlight-line"><span class="highlight-cl">--private-repo
users can only access their private repo

```



```

</span></span><span class="highlight-line"><span class="highlight-cl"> --realm string
    realm for authentication (default "rclone")
</span></span><span class="highlight-line"><span class="highlight-cl"> --server-read-t
meout duration Timeout for server reading data (default 1h0m0s)
</span></span><span class="highlight-line"><span class="highlight-cl"> --server-write-
imeout duration Timeout for server writing data (default 1h0m0s)
</span></span><span class="highlight-line"><span class="highlight-cl"> --stdio
    run an HTTP2 server on stdin/stdout
</span></span><span class="highlight-line"><span class="highlight-cl"> --template str
ng User Specified Template.
</span></span><span class="highlight-line"><span class="highlight-cl"> --user string
    User name for authentication.
</span></span></code></pre>
<p>查阅 <a href="https://ld246.com/forward?goto=https%3A%2F%2Frclone.org%2Fflags%2
" target="_blank" rel="nofollow ugc">global flags page</a> 以查看没有在这里列出的全局选
。 </p>
<h3 id="rclone-serve-sftp">rclone serve sftp</h3>
<p>通过 <code>sFTP</code> 协议提供访问 <code>remote:path</code> 的服务</p>
<h4 id="概要-----">概要</h4>
<p><code>rclone serve sftp</code> 实施一个基本的 <code>sftp</code> 服务，通过 <code>sftp</code> 协议提供访问 <code>remote</code> 的服务。他可以通过 sftp 客户端使用，你也可以先建一个 <code>sftp</code> 类型的 <code>remote</code> 去读写他。</p>
<p>你可以使用 filter 选项（例如 <code>--include</code>、<code>--exclude</code>）去制服务哪些文件</p>
<p>这个服务会 log errors，用 <code>-v</code> 以查看 logs。</p>
<p><code>--bwlimit</code> 被用于文件传输，使用 <code>--stats</code> 控制统计打印。</p>
<p>你必须提供一些用于验证的方法，可以使用 <code>--user/--pass</code>、一个公钥验证文（使用 <code>--authorized-keys</code> 指派，默认与 ssh 的相同）、一个 <code>--auth-proy</code>、或者设置 <code>--no-auth</code>。</p>
<p>提醒下这个也实施了几个 shell 命令，所以他可以为 rclone sftp 后台提供 md5sum/sha1sum/d 信息。意味着当与 rclone sftp backend 配对时，它支持 SHA1SUMs、MD5SUMs、和 about 命。</p>
<p>如果你不施加 <code>--key</code> 那么 rclone 会自动生成一个，并且将它缓存以供之后使。</p>
<p>默认这个服务器会绑定 <code>localhost:2022</code>，如果你希望他可以被从外部访问，么就使用 <code>--addr :2022</code></p>
<p>注意默认的 <code>--vfs-cache-mode off</code> 对于 rclone sftp backend 是可以的，但对于其它 SFTP 客户端可能就用不了。</p>
<h3 id="rclone-serve-webdav">rclone serve webdav</h3>
<p>通过 <code>webdav</code> 协议提供访问 <code>remote:path</code> 的服务</p>
<h4 id="概要-----">概要</h4>
<p><code>rclone serve webdav</code> 实施一个基本的 <code>webdav</code> 服务，通过 <code>webdav</code> 协议在 HTTP 上提供访问 <code>remote</code> 的服务。他可以通过 webdav 客户端、浏览器进行浏览，你也可以先建一个 <code>webdav</code> 类型的 <code>r mote</code> 去读写他。</p>
<h4 id="Webdav-选项">Webdav 选项</h4>
<h5 id=""><code>--etag-hash</code></h5>
<p>控制 ETag header。没有这个选项的话，ETag 默认基于 ModTime 和 对象大小</p>
<p>如果这个选项被设置为 <code>auto</code>，那么 rclone 会选择后台第一个支持的 hash，者你可以使用一个 hash 的名字例如：“MD5” 或 “SHA-1”</p>
<p>使用 <code>rclone hashsum</code> 以查看整个列表。</p>
<h4 id="服务器选项----">服务器选项</h4>
<p><code>--addr</code> 指定监听的地址和端口 例如：<code>--addr 1.2.3.4:8000</code>

```

`--addr :8080` 监听所有 ip。默认只监听 `localhost` 。用 `0` 可以让系统自动选择端口

`--server-read-timeout` 和 `--server-write-timeout` 控制传输读写时间。

`--max-header-bytes` 设置在 http 头接收的最大字节数

`--baseurl` 控制服务的 url 前缀。默认从根。如果用了 `--baseurl "/rclone"` 那么 rclone 会从 `"/rclone/"` 开始服务。对于使用代理很有用。rclone 默认会插入 `/` , 所以 `--baseurl "rclone"` `--baseurl "/rclone"` 和 `--baseurl "/rclone/"` 效果是一样的

`--template` 允许用户为 http 和 webdav 服务功能指定自定义标记模板。服务器将以下要在模板中使用的标记导出到服务器页面:

参数	描述
.Name	文件或文件夹的全路径
.Title	Directory listing of .Name
.Sort	The current sort used. This is changeable via ?sort= parameter
	Sort Options: namedirfirst,name,size,time (default namedirfirst)
.Order	The current ordering used. This is changeable via ?order= parameter
	Order Options: asc,desc (default asc)
.Query	Currently unused.
.Breadcrumb	Allows for creating a relative navigation
-- .Link	The relative to the root link of the Text.

```

</tr>
<tr>
<td>-- .Text</td>
<td>The Name of the directory.</td>
</tr>
<tr>
<td>.Entries</td>
<td>Information about a specific file/directory.</td>
</tr>
<tr>
<td>-- .URL</td>
<td>The 'url' of an entry.</td>
</tr>
<tr>
<td>-- .Leaf</td>
<td>Currently same as 'URL' but intended to be 'just' the name.</td>
</tr>
<tr>
<td>-- .IsDir</td>
<td>Boolean for if an entry is a directory or not.</td>
</tr>
<tr>
<td>-- .Size</td>
<td>Size in Bytes of the entry.</td>
</tr>
<tr>
<td>-- .ModTime</td>
<td>The UTC timestamp of an entry.</td>
</tr>
</tbody>
</table>
<h5 id="认证---">认证</h5>
<p>默认服务无需登陆</p>
<p>你可以用可设置多个用户的 htpasswd 文件, 或只用 --user 和 --pass 选项设置一个用户.</p>
<p>用 --htpasswd /path/to/htpasswd 提供 htpasswd 文件。它使用标准 apach 格式, 支持 MD5, SHA1 和 Bcrypt 用于基本的验证。</p>
<p>创建一个 htpasswd 文件: </p>


```
<code class="highlight-chroma">touch htpasswd
htpasswd -B htpas
wd user
htpasswd -B htpas
wd anotherUser
</code></pre>
<p>可以在 rclone 运行的时候更新 htpasswd 文件。</p>
<p>使用 --realm 设置授权 realm.</p>
<h5 id="SSL-TLS--">SSL/TLS</h5>
<p>默认用 http。如果你想用 https, 需要用 --cert 和 --key 签。如果你想做客户端的验证, 还需要用 --client-ca</p>
<p><code>--cert</code> 应当是 PEM 编码的 certificate 或连起来的一串有 CA 的 certificate</p>
<p><code>--key</code> 应当是 PEM 编码的私钥</p>
<p><code>--client-ca</code> 应当是 PEM 编码的用户证明</p>

```


```

过滤-包含和排除

注意: `purge` 不适用 `filter`

最简单的试用方法是 `ls` 或者 `--dry-run` 与 `-v` 搭配

Patterns

使用 "file globs" 进行匹配

如果 pattern 以 `/` 开头那么只匹配目录树顶层, 相对于 `remote` 的根目录。如果不以 `/` 开头, 那就匹配路径的末尾, 但它会匹配整个路径元素:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">file.jpg - 匹配 "file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 匹配 "directory/file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "directory/afile.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">/file.jpg - 匹配 "file.jpg" in the root directory of the remote
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "directory/file.jpg"
</span></span></code></pre>
```

即使在 windows 上, 也不能用 `\`

`*` 匹配 `\` 以外的任何字符串:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">*jpg - 匹配 "file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 匹配 "directory/file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "file.jpg/something"
</span></span></code></pre>
```

`**` 匹配任何字符串, 包括 `/`:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">dir/** - 匹配 "dir/file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 匹配 "dir/dir/dir2/file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "directory/file.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 不匹配 "adirectory/file.jpg"
</span></span></code></pre>
```

`?` 匹配 `\` 以外的任何字符:

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">l?ss - 匹配 "less"
</span></span><span class="highlight-line"><span class="highlight-cl">    - 匹配 "lass"
</span></span><span class="highlight-line"><span class="highlight-cl">    - doesn't match "floss"
</span></span></code></pre>
```

`[` 和 `]` 构成类匹配, 如 `[a-z]` `[aeiou]` `[[:alpha:]]` 详见 <https://ld246.com/forward?goto=https%3A%2F%2Fgolang.org%2Fpkg%2Fregexp%2Fsyntax%2F> `go regexp docs`

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">h[ae]llo - matches "hello"
</span></span></code></pre>
```



```

</span></span><span class="highlight-line"><span class="highlight-cl">      - matches "
allo"
</span></span><span class="highlight-line"><span class="highlight-cl">      - doesn't ma
ch "hullo"
</span></span></code></pre>
<p><code>{</code> 和 <code>}</code> 构成选择器，应包含逗号分隔的一系列 pattern 。这些
patterns 可以包含通配符：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{one,two}_potato - matches "one_potato"
</span></span><span class="highlight-line"><span class="highlight-cl">      - matc
es "two_potato"
</span></span><span class="highlight-line"><span class="highlight-cl">      - doesn
t match "three_potato"
</span></span><span class="highlight-line"><span class="highlight-cl">      - doesn
t match "_potato"
</span></span></code></pre>
<p><code>\</code> 用于转义</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">\*.jpg      - matches "*.jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">\\.jpg      - match
s ".jpg"
</span></span><span class="highlight-line"><span class="highlight-cl">\[one\].jpg - mat
hes "[one].jpg"
</span></span></code></pre>
<p>patterns 都是大小写敏感的（除了 --ignore-case 使用时）</p>
<p>filter globs 只能用于 flags，不能用于 remotes 。例如：<code>rcclone copy "remote:dir*.jp
"/path/to/dir</code> 不会生效，应该这样写 <code>rcclone --include "*.jpg" copy remote:dir
path/to/dir</code></p>
<h4 id="目录">目录</h4>
<p>rcclone 会保持 track 能匹配所有 file pattern 的文件夹</p>
<p>例如有这样一个 include rule</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">/a/*.jpg
</span></span></code></pre>
<p>那么会同步这个规则的文件夹：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">/a/
</span></span></code></pre>
<p>如果在任何规则后加上一个 <code></code> ，那么就只会匹配文件夹。</p>
<p>目录匹配对于基于 bucket 的远程地址无效（例如 s3, google compute storage），因为它们
有文件夹的概念。</p>
<h4 id="rsync-和--rcclone-patterns-的区别">rsync 和 rclone patterns 的区别</h4>
<p>rcclone 使用 bash 风格 <code>{a,b,c}</code> 的 glob 来匹配，而 rsync 不是。</p>
<h3 id="规则如何被应用">规则如何被应用</h3>
<p>每个文件从上到下的一条条尝试匹配规则</p>
<p>如果所有规则都不匹配，那么默认就包含这个文件</p>
<p>例如：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">- secret*.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">+ *.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">+ *.png
</span></span><span class="highlight-line"><span class="highlight-cl">+ file2.avi
</span></span><span class="highlight-line"><span class="highlight-cl">- *
</span></span></code></pre>

```

<p>会包含: </p>

 <code>file1.jpg</code>

 <code>file3.png</code>

 <code>file2.avi</code>

<p>排除: </p>

 <code>secret17.jpg</code>

non <code>*.jpg</code> and <code>*.png</code>

<h3 id="添加过滤规则">添加过滤规则</h3>

<h4 id="重复选项">重复选项</h4>

<p>你可以重复以下选项来添加多个不同的规则</p>

 <code>--include</code>

 <code>--include-from</code>

 <code>--exclude</code>

 <code>--exclude-from</code>

 <code>--filter</code>

 <code>--filter-from</code>

<p>不要同时使用 <code>--include*</code> 和 <code>--exclude*</code> , 可能有意外结果。尝试使用 <code>--filter*</code> </p>

<p>相同类型的会在一起按以上的顺序处理, 无论原始参数的顺序如何。</p>

<p>如果要混合 include 和 exclude 的顺序, 用 <code>--filter</code> </p>

<h4 id="-"><code>--exclude</code> </h4>

<p>添加单条的排除规则。可以重复。例如: </p>

<p><code>--exclude *.bak</code> </p>

<h4 id="--"><code>--exclude-from</code> </h4>

<p>从文件添加排除规则。可以重复。例如: </p>

<pre> <code class="highlight-chroma">## a sample exclude rule file

*.bak

file2.jpg

</code></pre>

<h4 id="---"><code>--include</code> </h4>

<p>添加单条的包含规则。可以重复。例如: <code>--include *.{png,jpg}</code> </p>

<p>会在 filter list 末尾添加一个隐含的 <code>--exclude *</code> 这意味着你可以混合 <code>--include</code> 和 <code>--include-from</code> 与其它 filters , 但你必须在 include 描述包含所有你想包含的文件。</p>

<h4 id="----"><code>--include-from</code> </h4>

<pre> <code class="highlight-chroma">## a sample include rule file

*.jpg

*.png

file2.avi

</code></pre>

<h4 id="-----"><code>--filter</code> </h4>

<pre> <code class="highlight-chroma">--filter "- *.bak"

</code></pre>

<h4 id="-----"><code>--filter-from</code> </h4>

<pre> <code class="highlight-chroma">

```

cl">## a sample filter rule file
</span></span><span class="highlight-line"><span class="highlight-cl">- secret*.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">+ *.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">+ *.png
</span></span><span class="highlight-line"><span class="highlight-cl">+ file2.avi
</span></span><span class="highlight-line"><span class="highlight-cl">- /dir/Trash/**
</span></span><span class="highlight-line"><span class="highlight-cl">+ /dir/**
</span></span><span class="highlight-line"><span class="highlight-cl">## exclude everyt
ing else
</span></span><span class="highlight-line"><span class="highlight-cl">- *
</span></span></code></pre>
<p>这个规则会包含所有 jpg 和 png 文件，排除所有匹配 secret*.jpg 的文件，并包含 <code>file2
avi</code>，也会包含 <code>dir</code> 中的一切（除了 <code>/dir/Trash</code>）。</
>
<h4 id="-----"><code>--files-from</code></h4>
<p>有一个 <code>files-from.txt</code> 文件：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">## comment
</span></span><span class="highlight-line"><span class="highlight-cl">file1.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">subdir/file2.jpg
</span></span></code></pre>
<p>这样运行：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">rclone copy --files-from files-from.txt /home/me/pics remote:pics
</span></span></code></pre>
<p>结果：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">/home/me/pics/file1.jpg      → remote:pics/file1.jpg
</span></span><span class="highlight-line"><span class="highlight-cl">/home/me/pics/s
bdir/file2.jpg → remote:pics/subdir/file2.jpg
</span></span></code></pre>
<h4 id="-----"><code>--min-size</code></h4>
<h4 id="-----"><code>--max-size</code></h4>
<h4 id="-----"><code>--max-age</code></h4>
<h4 id="-----"><code>--min-age</code></h4>
<h4 id="-----"><code>--delete-excluded</code></h4>
<p>这个选项很危险。最好先用 --dry-run -v 运行一下。</p>
<p>这会在目标路径删除所有排除在规则外的文件。</p>
<h4 id="-----"><code>--dump filters</code></h4>
<h4 id="-----"><code>--ignore-case</code></h4>
<h3 id="基于一个文件而排除">基于一个文件而排除</h3>
<p>用 <code>--exclude-if-present</code></p>
<p>举例，你有以下目录结构：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">dir1/file1
</span></span><span class="highlight-line"><span class="highlight-cl">dir1/dir2/file2
</span></span><span class="highlight-line"><span class="highlight-cl">dir1/dir2/dir3/file

</span></span><span class="highlight-line"><span class="highlight-cl">dir1/dir2/dir3/.ig
ore
</span></span></code></pre>
<p>执行：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">rclone sync --exclude-if-present .ignore dir1 remote:backup

```

</code></pre>

<p>这样 dir3 就不会被同步了</p>

<p>这个选项不能多重添加，只能用一個排除文件。</p>