

# Mount Sharepoint

Set up an MS App password for rclone on <https://mysignins.microsoft.com/security-info>

```
sudo apt-get install rclone  
  
mkdir -p ~/.config/rclone/  
vi ~/.config/rclone/rclone.conf
```

[~/.config/rclone/rclone.conf](#)

```
[whatevername]  
type = webdav  
url = https://domainname.sharepoint.com/sites/sitename  
vendor = sharepoint  
user = firstname.lastname@example.com  
pass = xxxx
```

```
rclone config  
#edit existing site and add the app password
```

Mount using the following, where “whatevername” is the name of the sharepoint site and “~/sharepoint/whatevername” is the desired mount point:

```
rclone --vfs-cache-mode writes mount whatevername: ~/sharepoint/whatevername
```

If this works, run it in daemon mode:

```
rclone --vfs-cache-mode writes mount whatevername: ~/sharepoint/whatevername  
--daemon
```

To stop the mount, use:

```
fusermount -u /path/to/local/mount  
#or  
fusermount -uz /path/to/local/mount
```

## Use in fstab

to use in fstab, create wrapper using:

```
sudo ln -s /usr/bin/rclone /sbin/mount.rclone
```

then it can be used in fstab like:

```
whatevname: /path/to/mount/point rclone user,rw,noauto,nofail,_netdev,x-  
systemd.automount,args2env,vfs_cache_mode=writes,rc.config=/etc/rclone.conf,  
cache_dir=/var/cache/rclone 0 0
```

or better use rclonefs.py script

<https://forum.rclone.org/t/auto-mount-from-fstab-entry-with-systemd-options/23897>

[/usr/local/bin/rclonefs.py](#)

```
#!/usr/bin/env python3  
from os import environ  
from time import sleep  
import sys  
from subprocess import check_call, check_output,  
CalledProcessError, TimeoutExpired, Popen, PIPE  
  
def write_to_journal(string):  
    check_output(['systemd-cat'], input=string.encode('utf-8'))  
  
write_to_journal('rclone-script started')  
  
# Save passed arguments  
remote_mount = sys.argv[1]  
local_mount = sys.argv[2]  
sys_args = sys.argv[4].split(',')  
  
# Must add path to environment in order for rclone mount to work  
env = environ.copy()  
output_raw = check_output(['whereis', 'rclone'],)  
##path = output_raw.decode('utf-8').split(''  
'')[1].removesuffix('/rclone')  
path = "/usr/bin"  
env['PATH'] = path  
  
# If path extraction does not work enter desired path here  
# env['PATH'] = '/usr/bin'  
  
options = []  
discarded_options = []  
  
# Load options for rclone mount to consume  
for arg in sys_args:  
    if not arg.startswith('rc.'):  
        discarded_options.append(arg)  
        continue  
    options.append(f'--{arg[3:]}')  
  
write_to_journal(f'rclone-script options: {options}')  
write_to_journal(f'rclone-script discarded options:  
{discarded_options}')
```

```
# Mount folder, if this doesn't work try with full path for rclone
output = Popen(
    ['rclone', 'mount', *options, remote_mount, local_mount ], env=env,
    stdout=PIPE, stderr=PIPE, close_fds=True, shell=False,
)
try:
    output.wait(1)
except TimeoutExpired as e:
    # Must wait for mount to be seen, in order for systemd not to
    complain
    write_to_journal('rclone-script rclone mount success')
    for tries in range(1, 20):
        try:
            check_call(['mountpoint', local_mount], env=env)
            write_to_journal(f'rclone-script mount found after {tries}
tries')
            break
        except CalledProcessError:
            write_to_journal('rclone-script mount not found, sleeping')
            sleep(0.1)
    else:
        write_to_journal('rclone-script mount not found, not looking
anymore')
        sys.exit(1)

    sys.exit(0)
else:
    write_to_journal(f'rclone-script mount failed. ErrMsg:
{output.stderr.read().decode("utf-8")}')
    sys.exit(1)
```

```
sudo chmod 755 /usr/local/bin/rclonefsy
```

Then use

```
gdrive: /mnt/gdrive fuse.rclonefsy user,noauto,x-
systemd.automount,_netdev,x-systemd.mount-timeout=30,x-systemd.idle-
timeout=10min,rc.allow-other,rc.fast-list,rc.drive-export-
formats=.link.html,rc.config=/home/user/.config/rclone/rclone.conf 0 0
```

For troubleshooting use:

```
journalctl -b | grep rclone
```

# systemd service

<https://gist.github.com/kabili207/2cd2d637e5c7617411a666d8d7e97101>

rclone@.service

```
# User service for Rclone mounting
#
# Place in ~/.config/systemd/user/
# File must include the '@' (ex rclone@.service)
# As your normal user, run
#   systemctl --user daemon-reload
# You can now start/enable each remote by using rclone@<remote>
#   systemctl --user enable rclone@dropbox
#   systemctl --user start rclone@dropbox

[Unit]
Description=rclone: Remote FUSE filesystem for cloud storage config %i
Documentation=man:rclone(1)
After=network-online.target
Wants=network-online.target
AssertPathIsDirectory=%h/mnt/%i

[Service]
Type=notify
ExecStart= \
  /usr/bin/rclone mount \
    --config=%h/.config/rclone/rclone.conf \
    --fast-list \
    --vfs-cache-mode writes \
    --vfs-cache-max-size 100M \
    --log-level INFO \
    --log-file /tmp/rclone-%i.log \
    --umask 022 \
    --allow-other \
    %i: %h/mnt/%i
ExecStop=/bin/fusermount -u %h/mnt/%i

[Install]
WantedBy=default.target
```

From:  
<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:  
<http://wuff.dyndns.org/doku.php?id=linux:sharepoint&rev=1637767360>

Last update: **2023/05/29 11:53**



