sshfs Mount over SSH

CONCEPTS

sshfs speaks SFTP over a normal ssh login: if ssh <host> works, sshfs <host>: <dir> works. Run it as your own user, not root.

Your programs ls, cp, editor
↓ FUSE
sshfs local mountpoint
↓ ssh + sftp
sftp-server on the remote host
↓ disk
Remote dir /srv/data

MOUNT AND UNMOUNT

sshfs *<user>*@*<host>*:*<dir>* *<mountpoint>*
Mount a remote directory

sshfs *<host>*: *<mountpoint>*
Mount the remote home directory

fusermount3 -u *<mountpoint>*
Unmount (Linux)

sshfs -p *<port>* *<host>*: *<mountpoint>*
Use a non-default SSH port

sshfs -o IdentityFile=*<key>* *<host>*: *<mountpoint>*
Pick the key to log in with

sshfs -o ro *<host>*: *<mountpoint>*
Mount read-only

sshfs -C *<host>*: *<mountpoint>*
Compress traffic (slow links)

findmnt -t fuse.sshfs
List active sshfs mounts

RECONNECT AND HANGS

sshfs -o reconnect,ServerAliveInterval=15 *<host>*: *<mountpoint>*
Recover after the link drops

pkill -f 'sshfs.*<mountpoint>'
Kill a hung sshfs, freeing stuck programs

RECONNECT AND HANGS (cont.)

A frozen mount (ls hangs, cd stalls) usually means a dead connection with no keepalive: set ServerAliveInterval together with reconnect.

PERMISSIONS AND SYMLINKS

sshfs -o allow_other *<host>*: *<mountpoint>*
Let other local users use the mount

sshfs -o allow_other,default_permissions *<host>*: *<mountpoint>*
Enforce file modes locally too

sshfs -o idmap=user *<host>*: *<mountpoint>*
Map remote user's IDs to yours

SPEED AND CACHING

sshfs -o dcache_timeout=*<s>* *<host>*: *<mountpoint>*
Cache directory data longer (default 20 s)

MOUNTING AT BOOT

mkdir -p *<mountpoint>*
Create the mountpoint first

fstab line

```
<user>@<host>:<dir> <mountpoint>
fuse.sshfs \
  _netdev,reconnect,ServerAliveInterval=
15,IdentityFile=<key> 0 0
# one line in the real file
```

RECIPES

Mount a server for the day

```
mkdir -p ~/mnt/box
sshfs -o
reconnect,ServerAliveInterval=15
<user>@<host>:/srv/data ~/mnt/box
ls ~/mnt/box
fusermount3 -u ~/mnt/box
```