# autossh Self-healing ssh tunnels

**autossh** supervisor → *spawns* → **ssh** your tunnel or session → *keepalive or test data* → **Remote host** sshd → *ssh exits or stalls* → **Restart** backs off up to the poll time

## CONCEPTS

autossh runs ssh for you: every argument except its own flags goes to ssh unchanged. Get the plain ssh command working first, then put autossh in front.

Use key or agent login: autossh cannot answer a password prompt, and -f makes that impossible.

## RUNNING AUTOSSH

**autossh -M 0** *<host>*
Supervised login session

**autossh -M 0 -N -L** *<lport>*:*<dst>*:*<dport>* *<host>*
Keep a local forward alive

**autossh -M 0 -N -R** *<rp>*:**localhost:***<p>* *<host>*
Keep a reverse tunnel alive

**autossh -M 0 -N -D** *<port>* *<host>*
Keep a SOCKS proxy alive

**autossh -M 0 -f -N** *<ssh args>*
Drop to the background

## RUNNING AUTOSSH (cont.)

**autossh -M 0 -i** *<key>* **-p** *<port>* *<user>*@*<host>*
Choose key and port as for ssh

Prefer an entry in ~/.ssh/config for the forwards and keepalives, then run just autossh -M 0 -f -N <host>.

## MONITORING

**-M 0**
Disable the monitor port, rely on ssh keepalives

**-M** *<port>*
Monitor with a port pair

**-M** *<port>*:*<echo_port>*
Monitor via a remote echo service

**-o ServerAliveInterval=30 -o ServerAliveCountMax=3**
Ssh quits after ~90 s of silence

**-o ExitOnForwardFailure=yes**
Exit ssh if a forward cannot bind

**-o ConnectTimeout=10**
Fail fast when the host is down

## MONITORING (cont.)

**-o BatchMode=yes**
Never prompt: fail instead

On the server side ClientAliveInterval in sshd_config closes stale listening sockets when the client vanishes, so a restarted reverse tunnel can bind again.

## ENVIRONMENT VARIABLES

**AUTOSSH_GATETIME=***<s>*
Seconds ssh must survive to count as started (30)

**AUTOSSH_POLL=***<s>*
Seconds between connection tests (600)

**AUTOSSH_PORT=***<port>*
Monitor port; overrides -M

**AUTOSSH_DEBUG=1**
Debug log, copied to stderr

**AUTOSSH_LOGLEVEL=***<0-7>*
Syslog level, 7 chattiest

**AUTOSSH_LOGFILE=***<file>*
Log to a file instead of syslog

**AUTOSSH_PIDFILE=***<file>*
Write the autossh pid

## ENVIRONMENT VARIABLES (cont.)

Variables go in front of the command (AUTOSSH_GATETIME=0 autossh ...) or in a systemd Environment= line.

## CONTROL AND TROUBLESHOOTING

**kill -USR1** *<pid>*
Kill ssh and start a new one

**kill -HUP** *<pid>*
Prod autossh to retry now after back-off

**kill** *<pid>*
Stop autossh and its ssh child

**pgrep -af autossh**
Find running autossh processes

**AUTOSSH_DEBUG=1 autossh -M 0 -v** *<ssh args>*
Debug autossh and ssh together

If autossh exits at once, ssh failed in the first 30 s (gate time) or with status 1: run the ssh command alone to see the error.

## SERVICES

**systemd unit**

```
[Unit]
Description=Tunnel to <host>
After=network-online.target
Wants=network-online.target

[Service]
User=<user>
Environment=
AUTOSSH_GATETIME=0
ExecStart=/usr/bin/autossh -
M 0 -N \
  -o ServerAliveInterval=30
-o ServerAliveCountMax=3 \
  -o
ExitOnForwardFailure=yes \
  -L
<lport>:localhost:<dport>
<host>
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

**systemctl enable --now autossh-***<name>*
Start the unit now and at boot

**journalctl -u autossh-***<name>* **-f**
Follow the unit's log

Under systemd do not use -f: stay in the foreground so systemd can supervise the process.