autossh Self-healing ssh tunnels

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.

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

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

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

MONITORING (cont.)

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

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

ENVIRONMENT VARIABLES

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

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

CONTROL AND TROUBLESHOOTING

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

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
```