



TECHNICAL NOTE

Connecting a broker behind NAT

For the customer connecting an on-site broker

When your MQTT broker runs on a machine that is not reachable from the internet, the hosted Analyzer cannot dial in. Instead your machine opens one **outbound** SSH tunnel. It passes through almost any firewall or NAT, it is encrypted and it opens **no inbound port** on your side. Your broker becomes visible only inside our server and only to the Analyzer.

Three things happen once, in order: you create a key, you send us the public half and after we enable it you start the tunnel.

How it works

Your machine	logiq-vda5050.com
broker on localhost:1883	SSH (user: logiqtun)
ssh -R (outbound, port 22)	
+----->	the Analyzer reaches your broker
	internally, on the port we assign you

The tunnel is started from your side, so it needs nothing opened towards you; SSH encrypts the whole path.

Step 1: Create a key

A dedicated SSH key, with an empty passphrase so the tunnel can run unattended. OpenSSH is built into current Windows, Linux and macOS.

Linux / macOS

```
ssh-keygen -t ed25519 -f ~/.ssh/logiq_tunnel -N "" -C "your-broker"
```

Windows (PowerShell)

```
ssh-keygen -t ed25519 -f "$env:USERPROFILE\.ssh\logiq_tunnel" -C "your-broker"
```

At *Enter passphrase* press Enter twice to leave it empty. If `ssh` is not found, install **OpenSSH Client** under Settings → Apps → Optional features.

Step 2: Send us the public key

Send the contents of the `.pub` file, the public half only, never the file without an extension.

Linux / macOS

```
cat ~/.ssh/logiq_tunnel.pub
```

Windows (PowerShell)

```
Get-Content "$env:USERPROFILE\.ssh\logiq_tunnel.pub"
```

Copy the line (it starts with `ssh-ed25519 ...`) into your reply to us. We enable it and assign you a port, the examples below use **19883**.

Step 3: Start the tunnel

Replace `localhost:1883` with your broker as *seen from your own machine* (if the broker runs on the same computer, `localhost:1883` is right). Replace `19883` with the port we gave you.

Linux / macOS (autossh, auto-reconnect)

Install `autossh` first (`apt install autossh`, `brew install autossh` or `dnf install autossh`), then:

```
autossh -M 0 -f -N \  
  -o "ServerAliveInterval 30" -o "ServerAliveCountMax 3" \  
  -o "ExitOnForwardFailure yes" \  
  -i ~/.ssh/logiq_tunnel \  
  -R 172.17.0.1:19883:localhost:1883 \  
  logiqtun@logiq-vda5050.com
```

Windows (PowerShell, test run)

Runs in the foreground; leave the window open. No error means the tunnel is up.

```
ssh -N -o "ServerAliveInterval=30" -o "ServerAliveCountMax=3" -o "ExitOnForwardFailure=yes" \  
-o "StrictHostKeyChecking=accept-new" -i "$env:USERPROFILE\.ssh\logiq_tunnel" -R \  
172.17.0.1:19883:localhost:1883 logiqtun@logiq-vda5050.com
```

Tell us when it is running, we then see your broker in the Analyzer immediately.

Step 4: Keep it running

For a permanent connection that survives a reboot and reconnects on its own.

Linux (systemd service)

Save as `/etc/systemd/system/logiq-tunnel.service` (replace `USER` with your login), then `systemctl enable --now logiq-tunnel`.

```
[Unit]
Description=LogIQ broker reverse tunnel
After=network-online.target
Wants=network-online.target

[Service]
User=USER
ExecStart=/usr/bin/autossh -M 0 -N \
  -o "ServerAliveInterval 30" -o "ExitOnForwardFailure yes" \
  -i /home/USER/.ssh/logiq_tunnel \
  -R 172.17.0.1:19883:localhost:1883 \
  logiqtun@logiq-vda5050.com
Restart=always
RestartSec=10

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

Windows (script + scheduled task)

Save this loop as `%USERPROFILE%\logiq-tunnel.ps1`:

```
$key = "$env:USERPROFILE\.ssh\logiq_tunnel"
while ($true) {
  ssh -N -o "ServerAliveInterval=30" -o "ServerAliveCountMax=3" -o
  "ExitOnForwardFailure=yes" -o "StrictHostKeyChecking=accept-new" -i $key -R
  172.17.0.1:19883:localhost:1883 logiqtun@logiq-vda5050.com
  Start-Sleep -Seconds 10
}
```

Then register it to start at logon (run once in PowerShell):

```
schtasks /Create /TN "LogIQ Broker Tunnel" /SC ONLOGON /RL LIMITED /F ^
/TR "powershell -NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass -File
%USERPROFILE%\logiq-tunnel.ps1"
```

For a server that runs without anyone logged in, set the task to *Run whether user is logged on or not*.

What we set on our side

In the Analyzer your broker profile points at host `172.17.0.1` , port `19883` , with **TLS off**, the SSH tunnel already encrypts the link. If your broker needs a username and password, tell us and we store them encrypted.

Security

- Only **outbound** SSH (port 22) is used. Nothing is opened, forwarded or exposed on your network and the broker is never reachable from the internet.
- The key can open **only that one** tunnel port, no shell, no file access, no other ports. You can stop the tunnel at any time by closing the window or the service.

Questions about connecting your broker: support@logiq-vda5050.com