FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

Veeam "Target Gate Has Rejected Connection" — the Cloud Gateway With No Path to the Repository

A field note about a new private-link Veeam cloud gateway that looked healthy in every console and still rejected every backup job — the network leg nobody draws, and the three checks I now run before a gateway goes into a pool.

Daanial Farrukh · 25 November 2026 · https://quabin.com/veeam-cloud-gateway-target-gate-rejected-connection/

The idea was a good one. Some backup customers already sat on our own network, connected over a private layer-2 link. Their backups were still going out over the internet and coming back in through the public gateway.

"Why are we sending on-net traffic out to the WAN?" someone asked in the design review.

Nobody had a good answer. So we built a second Veeam Cloud Connect gateway — a small Windows VM, 2 cores, 4 GB of RAM — with one leg on the private customer link and one leg on management. No internet needed. Full 10 Gb to the customers who used it.

The build went cleanly. The gateway joined the backup server. The service provider console showed it. The customer's management agent connected over the private link on the first try.

"Green everywhere," the engineer said. "Let's move the first tenant."

We moved the first tenant. The first job ran.

Target gate has rejected connection. Unable to establish target connection.

We ran it again. Same error. A second tenant. Same error.

"The agent is connected," the engineer said. "I can see it right there. How can the gateway reject it?"

I asked one question. "Where does the gateway send the data after it accepts it?"

He opened the network drawing. It showed the customer, the private link, the gateway. It stopped there.

That one cost us a full day.

That's the trap. **A cloud gateway is not the end of the path. It is the middle.** The console shows you the leg from the customer to the gateway. It does not show you the leg from the gateway to the repository, and that is the leg that carries every byte of backup data.

What a Veeam Cloud Gateway Actually Does

A cloud gateway is a relay. It is the only thing a Cloud Connect tenant talks to from outside. It does not store anything. It accepts the tenant's connection on one port and passes the traffic on to the servers that do the real work.

In a typical service-provider build the flow looks like this:

```
Tenant agent → Cloud gateway (TCP 6180) → Backup server (control)
Tenant agent → Cloud gateway (TCP 6180) → Repository (data)
```

So one gateway carries two very different kinds of traffic through the same port:

- **Control and management traffic.** The tenant's management agent and the service provider console talk to the backup server through the gateway. This is small, chatty traffic on the management network.
- **Backup data.** The actual blocks of the backup. The gateway opens a connection to the repository's data mover and streams the data there. In our build the repositories sat on a separate backup data network, not on management.
- **The redirect.** When a tenant connects, Cloud Connect decides which gateway in the tenant's pool should carry the session. The tenant dials one address and may be handed to another.

Here is the part that trips people up. Those two kinds of traffic leave the gateway on **different networks**. Management goes out the management leg. Backup data goes out the backup data leg. If the gateway has the first leg and not the second, half of it works perfectly.

Our new gateway had three things planned on paper:

- a private-link NIC facing the customers, port 6180, NAT mode
- a management NIC, on the same network as the backup server and the console
- a NIC on the backup data network, facing the repositories

The third one was never added. The VM had two NICs, not three.

Why Every Console Said "Healthy"

This is the part worth slowing down for. Every check we ran said the gateway was fine — and every check was telling the truth.

The management agent connected because that conversation runs over the management network. The gateway could reach the backup server and the console there with no problem. The service provider console saw the agent, saw the company, saw the policy.

On the management agent box, this command reported a healthy connection to the provider:

```
veeamconsoleconfig -s
```

The backup server saw the gateway service as up, because it checks the service over management too.

None of these checks asked the only question that mattered for a backup job: **can this gateway open a connection to the repository?**

When a job starts, the tenant hands its data to the gateway. The gateway tries to reach the repository's data mover on the backup data network. It has no interface there and no route there. The connection

fails, and the gateway reports the only thing it knows: the target rejected it.

So the error message points at the "target gate". The cause is one hop further: the repository leg.

A healthy control channel with failing jobs means the repository leg, not the tenant leg. That one sentence would have saved us a day.

How to Actually Fix This

Step 1: Read the symptom correctly

Before you touch anything, put the two facts side by side:

- Management side: the agent is connected, the console shows the company, the policy applies.
- Data side: every job fails with "Target gate has rejected connection. Unable to establish target connection."

If both are true, stop looking at the customer's firewall, the tenant's password and the private link. The tenant reached the gateway. The gateway could not reach the repository.

Plain takeaway: the tenant leg is proven by the management agent. The data leg is not.

Step 2: List the gateway's interfaces

On the gateway (Windows), list every IPv4 address:

```
Get-NetIPAddress -AddressFamily IPv4 | Select-Object InterfaceAlias, IPAddress, PrefixLength
```

You should see one address on each network the gateway needs: the customer-facing link, management, and the backup data network. In our case there were only two. No address on the backup data subnet at all.

If the address is there, check the prefix length too. A /29 on one box and a /24 on another means they disagree about who is "local". The box with the small mask sends traffic for the "outside" addresses to its default gateway instead — often the public one. We saw that on the same platform with a new repository: the backup server's data NIC was a /29, the new repository was just outside it, and the backup server tried to reach it over its public default route.

Plain takeaway: count the NICs, then read the masks.

Step 3: Add the missing leg

In the hypervisor, add a NIC to the gateway VM on the backup data network. Give it a static address on that subnet, with **no default gateway** — the only default route should stay on the leg that needs it.

Before you pick the address, check that nobody else holds it. We keep a small register of every address on the backup data network, and it already had one double-booking in its history. A duplicate IP on a backup network gives you a second, stranger failure on top of the first one.

On Windows, a duplicate shows up in the address state:

```
Get-NetIPAddress -InterfaceAlias "Ethernet 3" | Select-Object IPAddress, AddressState
```

Preferred means the address is yours. Duplicate means someone else already answers on it.

Plain takeaway: one new NIC, one static address, no second default gateway.

Step 4: Prove the repository leg from the gateway itself

This step is where most tutorials stop and where mine starts.

Do not test from your laptop. Do not test from the backup server. Test **from the gateway**, to each repository, on the repository data mover port:

```
Test-NetConnection <REPO_IP> -Port 6162
```

Expected output ends with:

```
TcpTestSucceeded : True
```

Run it for every repository that a tenant in this gateway's pool can land on. In a later build on the same platform this check paid for itself on the first run: the new gateway reached the backup server on 6169 and the new repository on 6162 — and failed to reach an older repository, because that repository was still on a /29 that did not include the gateway's address.

If you see TcpTestSucceeded : False and ping says "Destination host unreachable", there is no layer-2 or routing path. That is not a Veeam error. It is the verification doing its job.

Then test the control leg the same way:

```
Test-NetConnection <BACKUP_SERVER_IP> -Port 6169
```

Plain takeaway: every leg is tested from the box that has to use it.

Step 5: Put the gateway in its own pool and assign it

Tenant routing in Cloud Connect is by **gateway pool**, not by anything on the customer's machine. A new private-link gateway should sit in its own pool, so only the tenants that are on the private link get sent there.

In the backup server: Cloud Gateways › Gateway Pools › add a pool with only the new gateway.

In the service provider console: Companies › [company] › Edit › Bandwidth › Gateway pool › "Selected gateway pool", then pick the new pool.

The tenant keeps dialling whatever address it already has. Cloud Connect redirects it to a gateway from its assigned pool. Nothing changes on the customer's server.

This is also why a gateway that is missing a leg can hurt tenants who never asked for it. If you drop the new gateway into an existing pool, Cloud Connect may redirect any tenant in that pool to it, and their jobs start failing too.

Plain takeaway: a new gateway goes into a new pool, and only after Step 4 passes.

Step 6: Run one real job

After the new NIC and the pool assignment, start one backup job for one tenant on the new pool. Watch it to the end. A finished job with a new restore point on the repository is the only proof that both legs

work.

Then — and only then — move the next tenant.

The Gateway Checklist I Use Now

1. **Draw both legs.** Customer → gateway, and gateway → repository. If the drawing stops at the gateway, it is not finished.
2. **Count the NICs against the drawing.** Get-NetIPAddress -AddressFamily IPv4 on the gateway.
3. **Read the masks.** Every box on the backup data network uses the same prefix length.
4. **Check the address register before assigning.** AddressState must be Preferred.
5. **Test from the gateway, to every repository.** Test-NetConnection <REPO_IP> -Port 6162.
6. **Test the control leg.** Test-NetConnection <BACKUP_SERVER_IP> -Port 6169.
7. **New gateway, new pool.** Never add an untested gateway to a pool that live tenants use.
8. **One real job before the next tenant.** A restore point, not a green icon.

One more note. When a gateway sits in a different data centre from a repository, latency on the backup network will be higher — we saw about 21 ms between two sites. That is distance, not a fault. Know your normal before you call something slow.

What I Tell Every Customer Now

When a customer or a partner tells me a new gateway, proxy or replication relay is "up and connected", I ask:

"From that box, can you reach the place where the data ends up?"

If the answer is "yes, here is the TcpTestSucceeded : True to every repository", we're in good shape. If the answer is "the agent is connected", we're not done yet.

A green management channel tells you the box is reachable. It tells you nothing about where the box can send your data.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

What does "Target gate has rejected connection" mean in Veeam Cloud Connect?

The tenant reached the cloud gateway, but the gateway could not open the onward connection it needed for the job. In a service-provider build that is usually the connection from the gateway to the repository data mover. Check that the gateway has an interface and a route on the network where the repositories live.

Why does the management agent connect if backup jobs fail?

Management and control traffic go from the gateway to the backup server and the provider console, usually over the management network. Backup data goes from the gateway to the repository, often over a separate backup data network. A gateway with only the management leg passes every management check and fails every job.

Which ports should a Veeam cloud gateway reach?

Tenants reach the gateway on TCP 6180. From the gateway, test the backup server on TCP 6169 and each repository data mover on TCP 6162 with Test-NetConnection. Run the tests from the gateway itself, not from another server.

Do I need to change anything on the customer's server to use a new gateway?

No. Cloud Connect routes a tenant by its gateway pool. Put the new gateway in its own pool, then assign that pool to the company in the service provider console. The tenant keeps dialling the address it has and is redirected to a gateway from its assigned pool.

Should a new cloud gateway be added to an existing gateway pool?

Only after it has passed the repository and control-leg tests. Any tenant in the pool can be redirected to any gateway in it, so an untested gateway in a shared pool can break jobs for customers who never moved.

Why is the repository reachable from the backup server but not from the gateway?

Each box has its own interfaces, masks and routes. A repository on a subnet the gateway has no leg on, or a /29 mask on one side that leaves the gateway's address outside the "local" range, gives a path from one server and none from another. Always test from the box that will carry the data.

Read online: https://quabin.com/veeam-cloud-gateway-target-gate-rejected-connection/