FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

DR Failover on OpenStack — Why the Replica IP Never Answers Until You Set Allowed-Address-Pairs

A field story about a two-data-centre DR design that looked finished on paper — the cloud network rule almost nobody checks before failover day — and the port check I now run before any IP takeover is signed off.

Daanial Farrukh · 22 November 2026 · https://quabin.com/openstack-dr-failover-allowed-address-pairs/

The design was on one page, and it was clean.

"Two data centres. Stretched layer 2 between them. The replication tool watches the primary, and if it dies, the DR server takes over the same IP. Cloudflare keeps one A record. Nothing to change in DNS."

It was a good design. Fewer moving parts than anything else we had looked at.

I asked one question.

"Which layer decides whether the DR server is allowed to use that IP?"

"The replication tool. It sends the ARP and the switch learns the new MAC."

"On a physical switch, yes. These are OpenStack VMs. The switch is Neutron."

A pause.

"Neutron will block it?"

"By default, yes. And it will not tell anyone."

That's the trap. **On OpenStack, an IP takeover does not fail with an error. It fails with silence.** The DR server believes it owns the IP. The OS shows the IP on the interface. The replication console shows a successful failover. And not one packet for that IP gets in or out of the DR port.

What Allowed-Address-Pairs Actually Is

Every VM in OpenStack connects to the network through a **port**. A port has a MAC address and one or more fixed IPs, handed out by Neutron.

When **port security** is on (the default), Neutron adds anti-spoofing rules to that port. **Anti-spoofing means the port only lets through traffic that carries the port's own MAC and its own fixed IPs.** Anything else is dropped at the virtual switch, before it leaves the hypervisor.

That is a good default. It stops one tenant VM from pretending to be another.

It is also exactly what an IP failover does. The flow during a takeover looks like this:

Primary fails → DR server adds the service IP → DR sends a gratuitous ARP → neighbours update their ARP cache → traffic for the IP goes to the DR MAC

On OpenStack, the second arrow is where it stops. The service IP is a fixed IP on the **primary's** port, not on the DR port. So from Neutron's point of view, the DR server is now spoofing:

DR server → gratuitous ARP for the primary's IP → DR port anti-spoofing → dropped

Allowed-address-pairs is the Neutron setting that tells a port: these extra IPs (and optionally MACs) are legitimate on this port too. Add the service IP to the DR port as an allowed address pair, and the same gratuitous ARP is let through.

Where this bites:

- IP failover in replication and DR tools (gratuitous ARP takeover between a source and a target server).
- Virtual IPs with keepalived / VRRP for load balancers and database pairs.
- Any cluster where a "floating" service IP moves between nodes inside the guest OS, not through the cloud API.

Why Nobody Sees It in Testing

Most DR tests on a new design check the replication, not the takeover.

The replica is in sync. The DR server boots. You log in to it on its own IP. Everything works, because everything you touched used the DR server's own fixed IP — the one Neutron already knows.

The service IP is the one thing you cannot test without a real failover, and a real failover is usually the last step, scheduled for "after go-live", if at all.

And the usual signs of health all lie on the day:

- The replication console reports the failover as complete.
- ip addr on the DR server shows the service IP.
- The DR server can still reach the internet on its own IP.

Only clients trying to reach the service IP notice. From their side, the old MAC is gone and the new one never answers.

The Simpler Design That Made This the Only Question

Before the stretched L2 was on the table, the design had more moving parts. They are worth naming, because each one was dropped for a reason:

1. **Moving an OpenStack floating IP.** A floating IP lives in one region's router. It cannot cross to the other data centre.
2. **A cloud health check plus DNS failover.** A second detector can disagree with the replication tool, and a status check on the instance does not catch a network loss.
3. **A post-failover script that calls the Cloudflare API.** It needs an API token stored in plain text on a server that will itself be replaced at failover.

4 **A Cloudflare load balancer with failover pools.** It works, it costs a small monthly fee, and it was the answer until layer 2 between the sites became possible.

When the design reached two scripts on two machines plus a token, the feedback was simple: "looks complex." That was fair.

With stretched layer 2, the source and the DR server sit on the same segment. The replication tool's own IP failover moves the service IP. Cloudflare keeps one A record that never changes. No scripts for traffic at all.

Fewer moving parts means fewer things to fail on the worst day. But it also means one rule carries the whole design. Here that rule is the allowed address pair.

How to Actually Fix This

The commands below use the standard openstack client. Run the read-only ones first. The change steps are marked.

Step 1: Find Both Ports

You need the port IDs, not the server names.

```
openstack port list --server PRIMARY-UUID
```

```
openstack port list --server DR-UUID
```

Note the port that carries the service IP on the primary, and the port on the same stretched network on the DR server. If the two ports are on different networks, stop here: IP failover with ARP only works inside one layer 2 segment.

Step 2: Read What the DR Port Allows Today

```
openstack port show DR-PORT-ID -c fixed_ips -c allowed_address_pairs -c port_security_enabled
```

A port that is not ready shows port_security_enabled as True and an empty allowed_address_pairs. That port will drop the service IP at failover.

Plain takeaway: if the service IP is not in that output, the failover has never been possible.

Step 3: Add the Service IP as an Allowed Address Pair (change)

This is the fix. It changes the DR port only.

```
openstack port set --allowed-address ip-address=203.0.113.20 DR-PORT-ID
```

Here 203.0.113.20 stands for the service IP that Cloudflare points at. No MAC is given, so Neutron allows the IP with the DR port's own MAC — which is what a gratuitous ARP from the DR server will carry.

Do the same on the **primary's** port with the same IP if the service will ever fail back. After a failback, the primary takes the IP again, and by then the IP may no longer be the fixed IP it started with — for example, if someone rebuilt the port.

Step 4: Or Turn Port Security Off (change, and wider)

The other option is to disable port security on both ports. Neutron requires the security groups to be removed first:

```
openstack port set --no-security-group --disable-port-security DR-PORT-ID
```

I do not recommend it as the first choice. It removes the security groups from that port as well, so the VM now relies on its own host firewall for everything. Allowed-address-pairs keeps the security groups and opens exactly one IP.

Step 5: Confirm the Change on the Port

```
openstack port show DR-PORT-ID -c allowed_address_pairs
```

You should see the service IP with the DR port's MAC. If the list is empty, the command went to the wrong port, or your project's policy does not allow you to set address pairs. In that case, the cloud team must do it.

Step 6: Test the Takeover Before Go-Live

This step is where most tutorials stop and where mine starts. A port setting is a promise. Only a takeover proves it.

In a maintenance window, run a planned failover in the replication tool. Then, from a third VM on the same stretched segment:

```
ping -c 3 203.0.113.20
```

```
ip neigh show 203.0.113.20
```

The MAC in the ip neigh output must be the **DR port's MAC** (from Step 2). If ping answers and the MAC is the DR MAC, the takeover works end to end.

If ping fails and ip neigh shows FAILED or still shows the primary's MAC, the gratuitous ARP never got out. Go back to Step 2 on the DR port.

This failure is not a bug. It is the verification.

Then check from the outside, through Cloudflare, that the site answers. The public path must reach the stretched segment. If the public entry point is a floating IP NAT in the first data centre only, the internal IP moves and the public path does not.

The Fence: Power Off the Primary Through the Cloud, Not the OS

An IP takeover is only safe if the old primary is really gone. Two servers on one segment with the same IP is worse than an outage.

Many replication tools have an option like "shut down the source server" at failover. Think about when it runs: at failover, because the DR server cannot reach the primary. If it cannot reach it, it cannot shut it down either.

The fence must go through the layer below the operating system. On OpenStack that is the compute API of the first data centre. One pre-failover script on the DR server is enough:

```
openstack --os-cloud dc1 server stop PRIMARY-UUID
```

Use the **UUID, not the name**. At a full-server failover the DR server can take over the primary's hostname, so a name-based command may end up pointing at the wrong machine.

Two more notes on the job settings I used in this design:

- Automatic failover on, with monitoring every 5 seconds and failover after 5 missed checks. That is roughly 25 seconds of silence before the takeover starts.
- IP failover on, with the service IP on the stretched segment.

Still verify one thing in your own tool before you rely on it: that the job type you use (full server, files, application) really exposes the pre-failover script option. Not every job type does.

The Checklist Before You Sign Off an IP Takeover on OpenStack

1. **Same layer 2 segment.** openstack port list --server <UUID> on both servers shows ports on the same network.
2. **Service IP allowed on the DR port.** openstack port show <DR-PORT> -c allowed_address_pairs lists it.
3. **Service IP allowed on the primary port for failback.** Same command on the primary port.
4. **Security groups still in place.** openstack port show <PORT> -c security_group_ids is not empty, unless you chose Step 4 on purpose.
5. **Fence by UUID through the other data centre's API.** openstack --os-cloud dc1 server show <UUID> -c status works from the DR server.
6. **Takeover proven.** ip neigh show <SERVICE-IP> from a third VM shows the DR MAC after a planned failover.
7. **Public path moves too.** The public entry point lands on the stretched segment, not on a NAT in one site.

What I Tell Every Customer Now

When a DR design on any cloud says "the DR server takes over the IP", I ask one question:

"Show me the DR port's allowed address pairs."

If the answer is a list with the service IP in it, and a planned failover has already shown the DR MAC from a third VM, they're in good shape. If the answer is "the replication tool handles that", we're not done yet.

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

Why does IP failover not work on OpenStack?

Neutron port security adds anti-spoofing rules to every port, so a port only passes traffic for its own MAC and fixed IPs. When the DR server takes over the primary's IP and sends a gratuitous ARP, the DR port drops it. Add the service IP to the DR port as an allowed address pair.

What does allowed-address-pairs do in OpenStack Neutron?

It adds extra IP addresses, and optionally MAC addresses, that a port is allowed to use on top of its fixed IPs. Anti-spoofing stays on for everything else. It is the standard way to run virtual IPs for keepalived, VRRP and replication tools that fail over an IP.

Should I disable port security instead of using allowed-address-pairs?

Usually not. Disabling port security also removes the security groups from that port, so the VM depends on its own firewall alone. Allowed-address-pairs opens exactly one IP and keeps the security groups in place.

Can an OpenStack floating IP move between two data centres?

No. A floating IP belongs to the router of one region or site. For failover between data centres you need a stretched layer 2 segment with IP takeover, a DNS change, or a load balancer in front of both sites.

How do I test that the DR server really took over the IP?

Run a planned failover, then from a third VM on the same segment ping the service IP and run ip neigh show for it. The MAC must be the DR port's MAC. If the entry shows FAILED or the old MAC, the gratuitous ARP was dropped.

Why fence the primary through the cloud API and not from the replication tool?

At failover the DR server cannot reach the primary, so a shutdown request from the tool cannot reach it either. A stop call to the first data centre's compute API works even when the primary's network is down. Use the server UUID, because the DR server may take over the hostname.

Read online: https://quabin.com/openstack-dr-failover-allowed-address-pairs/