Why cloud migrations often fail – and how to get them right

Highlights

- The urgency and complexity of leaving traditional virtualization platforms
- Why migrations get stuck
- Exits doesn't have to be painful

The urgency—and complexity—of leaving traditional virtualization platforms

Across industries, enterprises are moving fast to reduce dependency on **traditional virtualization platforms**. The reasons are well documented: licensing uncertainty, rising costs and a growing desire to consolidate around open, cloud-native infrastructure. Red Hat® OpenShift® Virtualization, whether deployed on premises or through Red Hat OpenShift Service on AWS (ROSA), has emerged as a strong contender for this next chapter.

Yet even with the right platform identified, many migrations falter before they gain momentum. Not because the destination is wrong—but because the journey is harder than expected.

"We planned for a 3-month phased migration. We ended up stalling halfway—not because of the app or the VM—but because networking kept breaking," said an IT infrastructure lead at a Fortune 100 enterprise.

This story is not isolated. It's a pattern.

Why migrations get stuck: It’s not the app, it’s the network

At a glance, moving from traditional virtualization platforms to Red Hat OpenShift seems like a standard lift-and-shift. You export the VM, import it into the new environment, test it and move on.
But real-world applications aren’t self-contained. They are deeply interconnected systems: a front-end service depending on an auth API, which talks to a user profile service, which writes to a shared database and emits events to a Kafka queue. All of it is stitched together over a network stack that has grown organically over the years.
Here’s what happens when you try to move one part of that system:
The app can no longer find its backend service—the DNS name resolves, but the IP route fails.
A firewall rule, written years ago, blocks egress from the new platform.
Traffic flows fine, but latency doubles—breaking timeouts and retries.
Logs and monitoring aren’t wired into the new environment yet. Debugging is guesswork.
And worst of all: during migration, you’re running in a **hybrid state**—with some workloads still on **traditional virtualization platforms** and others on Red Hat OpenShift. That hybrid state is where most outages occur and it's where most teams either stall or roll back entirely.

What if you could migrate without breaking connections?

Let’s walk through a typical phased migration journey and see how **IBM® Hybrid Cloud Mesh**, with its **Virtual Application Networking (VAN)** capability, changes the playbook.

H3: Phase 1: Discovery and planning

Most migrations start with discovery:
Which applications are eligible to move?
What are their dependencies?
Which systems will they still need to talk to after moving?
But most discovery processes stop at the infrastructure level—IP addresses, firewall zones or basic dependency diagrams. What teams need is a **service-level view**: which workloads talk to which APIs, databases or external services and how.
You can build a service-centric view of your application topology. This view helps you understand not just traffic patterns, but identity and context (for example, “service A talks to service B through this route, over this protocol”). This visibility helps decide **what to move together** and **what to decouple**.

H3: Phase 2: Pilot migration

A small, noncritical service is chosen to test the waters—maybe a background job service or a staging instance. The VM is moved to Red Hat OpenShift Virtualization. It spins up correctly. Success? Not quite.
The service is now unreachable—either because DNS names don't resolve correctly in the new environment or because IP-based policies block traffic. Networking teams scramble to write patch rules, open ports or adjust routes—but no one's confident it's production-ready.
VAN overlays extend your enterprise network across both environments. Services retain their identity, regardless of location. The migrated service can "see" its dependencies just like before. No new firewall rules. No rewriting addresses. And everything is encrypted, authenticated and observable out of the box.

Phase 3: Scaling the migration

Once the pilot is successful, the migration moves to core services. This phase is where most enterprises slow down—because now downtime has a business cost.
One team moves their app, but it still depends on a data service owned by another team—still on traditional virtualization platforms. Any break in connectivity means:

Failed transactions
Broken customer sessions
Disrupted SLAs

With Hybrid Cloud Mesh, you're no longer dependent on full cutovers. You can move in phases, knowing that VAN maintains persistent, zero-trust connectivity across environments. Teams are decoupled—one can move without waiting for the other. The mesh takes care of bridging the gap.

Phase 4: Operationalization and optimization

By now, most of the workloads are on Red Hat OpenShift—but operations haven't caught up. Network observability is patchy. Engineers still worry about how to route external traffic into the new environment, or how to monitor east-west traffic across the mesh. Because VAN treats networking as an app-centric overlay, everything from ingress policies to service discovery is managed in a single, unified control plane.
You can route by app name, identity or policy—not by brittle IPs and ports. Engineers regain confidence. Operations stabilize. Costs drop—because what previously took 10 people to manage now needs only two.

A composite example: One company's journey

Let's imagine a company running 1,200 VMs, hosting everything from customer-facing portals to batch jobs and analytics pipelines. Facing budget constraints and licensing shifts, they decide to move to Red Hat OpenShift Virtualization.
They start with a pilot—a microservice that handles document generation. It works well… until it tries to call the back end that stores those documents. The back end is still on VMware, and their call times out. Debugging reveals that the call is failing due to a misconfigured NAT rule.
Their engineers try to fix it—but now the front end times out. DNS entries haven't propagated. They're caught in a whack-a-mole of network fixes.
So they pause. They deploy Hybrid Cloud Mesh. They define their application boundaries as services, not machines. With VAN, their document service can call its back end without worrying where it lives.

The result?
Migrations proceed without emergency control rooms
Rollbacks are rare
SLA breaches disappear
The final migration is completed 2 months ahead of schedule

The big picture: Your traditional virtualization plaforms exit doesn't have to be painful

The push away from traditional virtualization platforms is real. But the path forward doesn't have to be risky or expensive.

Hybrid Cloud Mesh changes how migrations are done—not by replacing your virtualization stack, but by removing the connectivity friction that breaks migrations.

You can:
Migrate progressively, without breaking dependencies
Connect services across platforms without rewriting the network
Cut engineering overhead and reduce risk

Keep your business running—even while your infrastructure shifts

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the
United States of America
April 2025