SGX Group

From Virtual Machines to Immutable Metal: Migrating from RHV and VMware to Talos

Rushan Ratha – Head of Platform Engineering

17th October 2025

Singapore Exchange

Commodities | Equity Derivatives | Fixed Income | FX | Indices | Securities

Topics

- Legacy Stack
- Breaking Point
- Pivot
- On-premise Architecture
- Migration Path
- VM Migration with KubeVirt
- Battle Scars

SGX Group

The Icebreaker!

- VMs are great but working with them is like moving a piano up a staircase.

- Traditional virtualisation taught us a lot… mostly about emotional resilience.

SGX Group

The Legacy Stack

- RHV and VMWare
  - 100+ hypervisors spread across 6 countries
  - 1000+ VMs
  - A mix of terraform, ansible and ClickOps for day 2 changes.
  - Heavy use of NFS.
  - Upgrades and patching are not seamless. Requires a lot of weekend work between infra and app/services teams.

The Breaking Point

We wanted cattle, not pets — but in a rack, not the cloud.

- RHV End-of-Life looming.
- VMware licensing costs increasing.
- Need a single platform standard to simplify ops and training.
- Bare-metal performance for latency-sensitive workloads.
- Security & immutability: No ssh, no root, no sudoer rules, no audit finding write up!

The Pivot Point 🥁 dum dum duuum… Enter Talos.

- We needed something radically different — not another Linux with bash glue.
- Talos + Omni stood out: Single plane of glass, no SSH, API-driven, declarative to the bone.
- Treats nodes as disposable — rebuild, don't repair.
- Perfect alignment with GitOps and our infrastructure-as-code mindset.
- Every system state (from kernel args to kubelet config) lives in YAML.

Omni Machine Config → Git → CI/CD → Node Bootstrapped

Our Architecture and Tools

- **Multi-zone layout**: split across DC cages each with independent power and network.

- **Redundancy baked in**: MLAGs, dual 25 Gb links per host, fallback mode for isolated operation.

- **Storage**: NFS and Object using Pure; future integration with Portworx for performance and replication.

- **Networking**: Cilium providing eBPF-based overlay, observability, and security.

- **Observability**: VictoriaMetrics stack aggregating metrics from all zones via Vector collectors.

- Other tools:

  - Falco for runtime security.

  - Kyverno for policy management.

  - Cert-Manager

  - External Secrets

SGX Group

The Migration Path

We didn't lift-and-shift — we lifted, stripped, and rebuilt the entire stack.

- **Phase 1** - Buy, rack, stack new hardware.
- **Phase 2** - Bootstrap & GitOps Foundation
  - PXE boot new servers and register into Omni.
  - Bootstrap the cluster.
  - Integrated FluxCD for continuous reconciliation.
  - All cluster and machine configuration stored in Git — no manual tweaks.
- **Phase 3** - Setup all platform workloads using Helm + kustomize overlays.
- **Phase 4** - Container Workload Migration:
  - Move workloads in our cloud environment to on-prem.
  - Verified observability and alert parity before cutover.
- **Phase 5** - VM Workloads:
  - Evaluating KubeVirt and selective re-platforming for critical legacy apps.
  - Gradual migration off hypervisors toward containerized or VM-as-Pod models.

VM Migration with KubeVirt

KubeVirt gave us a way to treat VMs like any other workload — no exceptions, no handholding, just GitOps.

- **Bridging the Gap:**
  - KubeVirt lets us run traditional VMs as Kubernetes workloads.
  - Perfect for apps that couldn't yet be containerised (legacy services, stateful workloads).
- **Migration Workflow**:
  - Export VM definition from RHV or VMware (OVF/OVA).
  - Convert image to QCOW2 and import into our internal image registry.
  - Create KubeVirt VirtualMachine manifest (with labels, StorageClass, network).
  - Deploy through Git — Flux reconciles it into the cluster.
  - Validate network + storage performance parity, then decommission the old VM.

- **Benefits**:
  - Full GitOps lifecycle for VMs — no click-based provisioning.
  - Simplified backups and replication via standard K8s tools (Velero, VolumeSnapshots).
  - Unified monitoring through VictoriaMetrics — VMs and containers on the same plane.
  - Seamless path to eventual containerisation.

The Battle Scars

Engineering Out the Pain (Without Losing Resilience)

- Networking Resilience, Simplified
  - Still using port-channels and MLAGs for high throughput and redundancy.
  - Introduced fallback modes — if a peer fails, each NIC gracefully reverts to individual mode.
  - Now we can destroy and rebuild a host with zero switch-side config changes.
- NFS is not great, but we hope to use Portworx for next-gen storage.
  - Predictable IOPS, zone-level replication, and simplified recovery workflows.
  - In the meantime, we're squeezing every last drop out of our NVMe backplane.
- Lesson Learned
  - Talos removes "quick fixes." You either fix it properly in Git or rebuild cleanly.
  - The discipline hurts at first — then it pays off forever.

The hook: "We didn't simplify by cutting corners — we simplified by making failure predictable."

SGX Group

Thank you

Questions?

Singapore Exchange

Commodities | Equity Derivatives | Fixed Income | FX | Indices | Securities

sgx.com