Migrating VM from OVA to OCP Virtualization at Cisco Unified Edge

Contents

Executive summary

Many existing workloads at edge locations are still delivered as OVA-based virtual machines from VMware environments. As customers adopt Cisco Unified Edge with Red Hat, they need a migration approach that fits edge operational realities, particularly limited WAN bandwidth and repeated deployment across multiple sites.

This white paper describes two validated approaches for migrating a RHEL-like guest VM from OVA to OpenShift Virtualization at Cisco Unified Edge on OpenShift 4.19+:

- MTV + NFS, where the OVA is staged on NFS and migrated with the Migration Toolkit for Virtualization (MTV)
- QCOW2 + HTTP, where the OVA is converted centrally and imported into the edge cluster over HTTP

When NFS is available at the edge location, the MTV-based workflow provides a direct migration path. When WAN constraints matter, or when the same image must be deployed across multiple edge locations, the QCOW2 + HTTP approach is usually the more practical operating model.

This white paper is a focused extension of the Cisco CVD program for Cisco Unified Edge and should be read alongside the Cisco Unified Edge with Red Hat Design Guide.

Introduction

Across edge deployments, some applications are already delivered as containers, while others are still packaged and operated as virtual machines. Cisco Unified Edge with Red Hat is built for edge sites where space, power, and on-site IT support are limited. Its compact design and cloud-based management make it easier to deploy, manage, and scale across many remote locations. It also lets customers run both containers and virtual machines on the same platform, so they can support new and existing applications together. At the same time, many organizations are evaluating alternatives to VMware ESXi, so migrating existing OVA-based virtual machines becomes a practical design question for edge deployments.

The intended audience is solution architects. The paper emphasizes architecture decisions, operational fit, and use-case guidance rather than detailed procedures. Implementation details are kept brief and focus on the validated migration patterns rather than step-by-step execution.

The guidance applies to an edge OpenShift Virtualization cluster. Validation was performed on Single Node OpenShift (SNO), but the migration patterns are relevant more broadly to the validated Cisco Unified Edge deployment options, from single-node footprints to OpenShift compact cluster designs.

The scope is intentionally narrow:

- RHEL-like guest virtual machines are used as the example workload.
- OpenShift 4.19+ and MTV 2.11 are the validated software versions.
- The target VM uses the default pod network.
- Successful migration is validated by booting the guest and accessing it through serial console or VNC in OpenShift Virtualization.

A common technical element in both migration approaches is virt-v2v, an open-source tool that converts virtual machines from other hypervisors into a form suitable for KVM-based platforms. In the MTV-based workflow, MTV orchestrates virt-v2v as part of the migration process. In the QCOW2 + HTTP workflow, virt-v2v is run directly on a conversion host to produce the disk image that OpenShift Virtualization later imports.

This paper does not cover Windows guest migration, Multus networking, security and compliance architecture, lifecycle management, or support ownership models.

The edge migration challenge

OVA remains a common packaging format for virtual machines exported from VMware environments. As these environments move to OpenShift Virtualization, the question is not only how to move a VM image, but also how to do it in a way that fits the operating model of the edge.

At edge locations, several constraints shape the migration design:

- WAN connectivity may be limited, intermittent, or shared with production traffic.
- Remote sites may not host the same supporting infrastructure available in a central data center.
- The same workload may need to be deployed repeatedly across many geographically distributed sites.
- Operational tasks must often be centralized and executed with minimal local dependencies.

OpenShift Virtualization does not consume an OVA package as a runtime artifact. The cluster needs a bootable VM disk image on persistent storage and a VirtualMachine definition describing the guest configuration. The migration workflow must bridge the source artifact format used in VMware environments and the target artifact model used by OpenShift Virtualization.

The key decision for solution architects is which migration pattern best fits the available infrastructure, WAN conditions, and rollout scale.

Migration approach 1: OVA on NFS with MTV

The first validated approach uses MTV to migrate a VMware-exported OVA into OpenShift Virtualization. The OVA file is placed on an NFS share that MTV can access. MTV then discovers the OVA, applies the required network and storage mappings, orchestrates conversion using virt-v2v, and creates the target virtual machine in OpenShift Virtualization.

The workflow is as follows:

1. Export the VM from VMware as OVA.
2. Place the OVA on an NFS share accessible to the migration workflow.
3. Create the MTV OVA provider, NetworkMap, StorageMap, and migration plan.
4. Run the migration.
5. Validate that the destination PVC and VirtualMachine are created successfully.
6. Boot the VM and verify console or VNC access.

MTV handles source discovery, target mapping, conversion orchestration, and VM creation as part of one migration process, which removes the need for a separate image-preparation pipeline at headquarters.

Network conversion is handled separately from disk conversion. The NetworkMap translates the source network definitions discovered from the OVA into the destination network configuration on OpenShift Virtualization. For the scope of this paper, the destination is the default pod network. Guest OS network settings should still be validated after boot because MAC addresses and interface identities may change during migration.

The NFS dependency is also the main limitation of this approach. NFS is not optional in this model. If the edge location can reach NFS only through WAN interfaces, the design becomes less practical, and that mismatch becomes more visible as site count increases.

This approach is validated for VMware-exported OVA files representing RHEL-like guests and is not the preferred path for virtual appliances.

MTV + NFS fits best when the required NFS infrastructure is available at the edge location under acceptable network conditions and the migration scope is limited to one site or a small number of sites.

Migration approach 2: pre-convert to QCOW2 and import through HTTP

The second validated approach separates conversion from deployment. The VM is first converted from OVA to QCOW2 at headquarters or a regional data center using virt-v2v. The resulting QCOW2 image is hosted over HTTP, and the edge OpenShift Virtualization cluster imports it using CDI as part of a VirtualMachine manifest workflow.

The workflow is as follows:

1. Export the VM from VMware as OVA.
2. Convert the OVA to QCOW2 at headquarters or a regional data center.
3. Host the QCOW2 image on an HTTP server reachable from the edge location.
4. Create a VirtualMachine manifest that references the image URL.
5. Apply the manifest to the edge OpenShift Virtualization cluster.
6. Allow CDI to import the image into persistent storage.
7. Boot the VM and verify console or VNC access.

Conversion is centralized while deployment remains distributed. The image can be prepared once and reused across multiple sites. Each edge cluster needs only HTTP reachability to the image host and the ability to apply the VM manifest during a planned deployment window.

This approach does introduce a central image-preparation task. Someone must run virt-v2v, validate the resulting QCOW2 image, host it, and maintain the VirtualMachine manifest template used at the edge. In practice this is often a reasonable tradeoff because it moves conversion work into a central environment where it is easier to manage and repeat.

Network handling is straightforward in this approach because the target VM definition is created directly for OpenShift Virtualization. The VM is attached to the default pod network using the standard masquerade binding. The guest operating system should still be validated after boot, because interface naming, DHCP behavior, or guest network configuration may differ from the original VMware environment after conversion.

One practical sizing detail: the requested PVC size should be slightly larger than the original virtual disk size. In the validated example, a nominal 20 GiB source disk required a 21 GiB request on the target storage class. This reflects usable-capacity differences after storage-class overhead is accounted for during CDI import, not a virt-v2v conversion issue.

The following example shows the key elements of the VirtualMachine manifest: the HTTP source URL for the QCOW2 image, the storage request with a slight size buffer, the virtio disk bus, the masquerade network binding for the default pod network, and runStrategy: Always so the VM starts automatically once CDI completes the import.

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: rhel9-edge
spec:
  runStrategy: Always
  dataVolumeTemplates:
    - metadata:
        name: rhel9-edge-dv
      spec:
        source:
          http:
            url: http://<image-host>/<vm-name>.qcow2
        storage:
          resources:
            requests:
              storage: 21Gi
          storageClassName: <storage-class>
  template:
    spec:
      domain:
        devices:
          disks:
            - name: rootdisk
              disk:
                bus: virtio
          interfaces:
            - name: default
              masquerade: {}
      networks:
        - name: default
          pod: {}
      volumes:
        - name: rootdisk
          dataVolume:
            name: rhel9-edge-dv
```

QCOW2 + HTTP fits a centralized distribution operating model. It is the better choice when WAN constraints matter, when the same image must be deployed across multiple edge locations, or when a cleaner separation between image preparation and site deployment is operationally useful.

Comparison and recommended use cases

The two approaches are designed for different edge conditions. Table 1 summarizes the practical tradeoffs as a quick reference.

Table 1. Solution comparison

Dimension	OVA on NFS with MTV	QCOW2 over HTTP with CDI
Source artifact	VMware-exported OVA	Pre-converted QCOW2
Primary infrastructure dependency	NFS share accessible to MTV	HTTP server reachable from edge cluster
Conversion location	Performed within MTV migration workflow	Performed centrally before edge deployment
Edge dependency on NFS	Required	Not required
Suitability for slow WAN links	Limited	Strong
Operational simplicity for a small number of sites	Strong	Moderate
Operational scalability across many sites	Moderate	Strong
Best-fit workload example	Supported RHEL-like guest with NFS available at the edge location	Supported RHEL-like guest deployed repeatedly across distributed edge locations
Recommended use case	Single-site or limited-site migration where NFS is available at the edge location	Multi-site rollout or WAN-constrained environment where HTTP is more practical

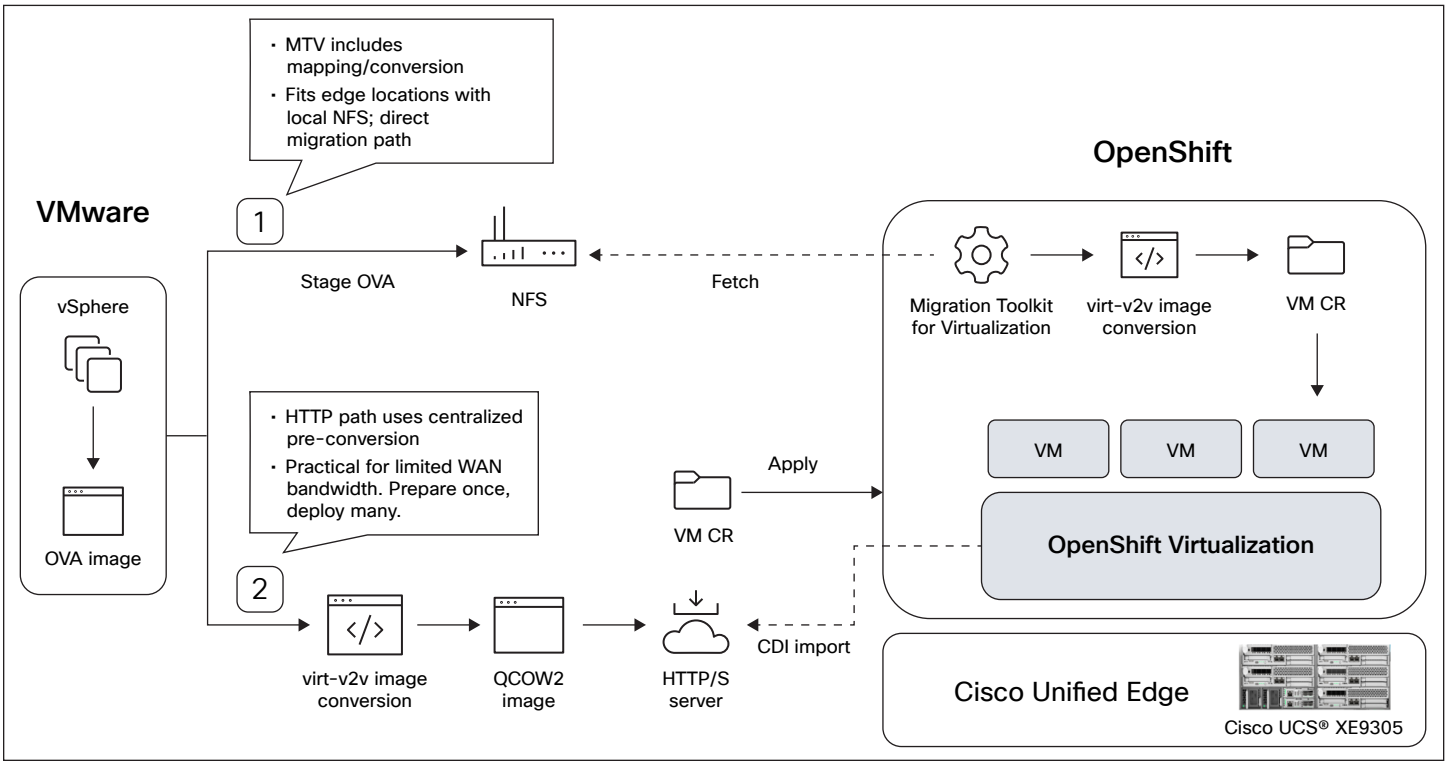

Figure 1. Workflow comparison

Use MTV + NFS when the edge location has access to an NFS server under acceptable network conditions and the migration scope is limited to one site or a small number of sites. Use QCOW2 + HTTP when NFS is available only across WAN interfaces, when WAN bandwidth is limited, or when the same VM image must be deployed to multiple edge locations.

For Cisco Unified Edge multi-site designs specifically, the QCOW2 + HTTP approach aligns better with centralized operations and Cisco Intersight®-driven automation, where the same image preparation and deployment workflow needs to repeat reliably across many edge locations.

Validation of successful migration

For this paper, the validation target is intentionally simple. The goal is to confirm that the VM has been migrated into OpenShift Virtualization and is usable in the target environment.

The validated success criteria are as follows:

- The target VirtualMachine object is created successfully.
- The VM boots on OpenShift Virtualization.
- The operator can access the guest by serial console or VNC through OpenShift Virtualization.
- The guest uses the default pod network for the validation scenario.

This level of validation is appropriate for a concise architecture white paper because it confirms the essential result of the migration without turning the document into a detailed operations guide.

Conclusion

Migrating an OVA-based virtual machine into OpenShift Virtualization at Cisco Unified Edge is more than a format-conversion step. The right approach depends on where supporting infrastructure is located, what the WAN path to the edge looks like, and how broadly the workload needs to be deployed. Both validated approaches address real edge operating conditions, and neither requires changes to the guest workload itself.

This paper covers one specific migration scenario within the broader Cisco Unified Edge platform. For platform architecture, deployment options, and the management context provided by Cisco Intersight, refer to the Cisco Unified Edge with Red Hat Design Guide.

References

- Cisco Unified Edge with Red Hat Design Guide
- OpenShift Virtualization 4.19 documentation
- Migration Toolkit for Virtualization 2.11 documentation

Authors

Shixiong Shang: Technical Marketing Engineer, Cisco

Tom Qian: Solutions Engineer, Cisco

Jonathan Wong: Solutions Architect, Red Hat

C11-6317351-00 07/26