GPU SMITH
COMPUTE ENGINEERING

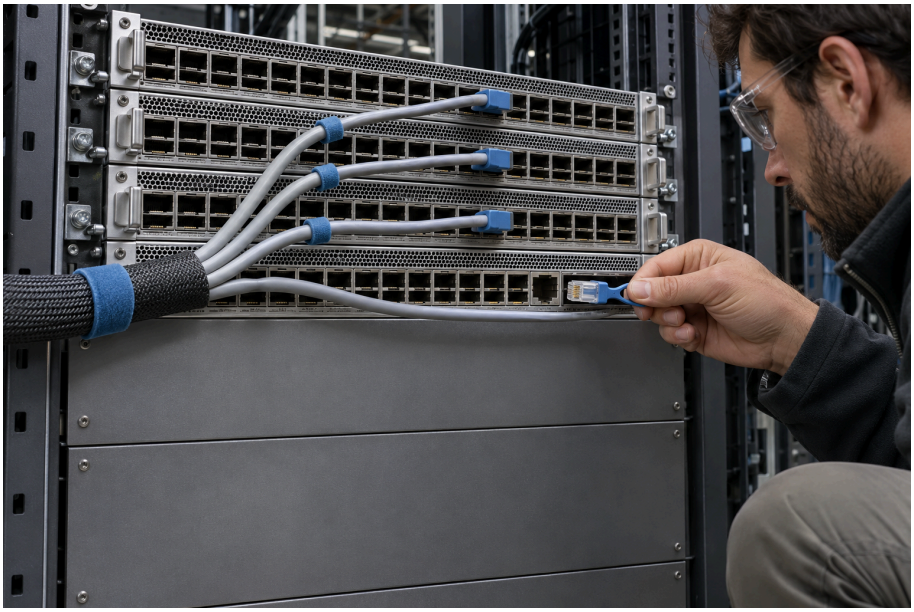
GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

NVIDIA Network Operator 26.7 Upgrade: Spectrum-X Preflight

> Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-11-24 · nvidia network operator 26.7 upgrade · nvidia network operator · spectrum-x kubernetes · spectrum-x ra 2.3 · network operator rollback · doca-ofed driver upgrade · connectx-8 multiplane · kubernetes rdma networking

Original article: https://gpusmith.com/articles/en/nvidia-network-operator-26-7-spectrum-x-upgrade



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes in Network Operator 26.7
4. Pass/Fail Preflight Matrix and Who Should Stay Put
5. Implementation Considerations and Process Changes
6. Rollback Artifacts and Recovery Sequence
7. Procurement Questions for ConnectX-8/9 and Multi-Plane Designs
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Frequently Asked Questions (FAQs)
11. Conclusion

Executive Summary

NVIDIA Network Operator 26.7 is the supported release line for Kubernetes clusters running Spectrum-X Ethernet, while the previous **26.4.x** line is a maintenance version that receives critical bug and CVE fixes only (docs.nvidia.com). NVIDIA's pages print no formal release date. As of November 24, 2026, GitHub's release listing already marks **v26.10.0** as the newest tag (github.com), and the 26.7.1 patch is also listed (github.com). This report qualifies the 26.7 line only and does not assess 26.10.x.

The release adds **Spectrum-X Reference Architecture (RA) 2.3**, makes hardware multiplane with DOCA xPlane generally available on ConnectX-8 SuperNICs, and adds IPv6 rail isolation and readiness conditions. It also carries upgrade traps documented in NVIDIA's release notes: the `spectrumXOptimized.version` field now names a profile ConfigMap instead of a built-in RA version; the Kubernetes floor rises to **1.32**; the Spectrum-X OVS (Open vSwitch) interface type changes from `dpdk` to `doca`; and RA 2.3 requires Kubernetes **1.35** or later (docs.nvidia.com). Upgrading from 26.4.x or earlier is not a drop-in change.

The most consequential trap is silent. A profile that lacks mlxConfig tuning for a NIC device ID applies no Spectrum-X parameters to those NICs and reports no error (docs.nvidia.com). The practical gate is a per-node profile coverage percentage that must read **100%** before production nodes are touched.

Kubernetes lifecycle compounds the decision. Upstream support for Kubernetes 1.34 ended on October 27, 2026 (kubernetes.io), and 1.35, the RA 2.3 minimum, reaches end of life on February 28, 2027, **96 days** after this report's date (kubernetes.io). A cluster on 1.32 needs three sequential minor upgrades to reach 1.35, because the API server may not skip minor versions (kubernetes.io).

NVIDIA publishes no dedicated rollback procedure for 26.7 in the pages reviewed. The documented driver recovery text is to set the driver version in the NicClusterPolicy back to the previous value, and Helm documents no CRD (CustomResourceDefinition) upgrade or removal support (helm.sh). Rollback is therefore built from artifacts captured before the change. No zero-downtime claim is made: loading a new driver removes network interfaces from running pods, so each node needs a maintenance window. Sites that need none of the 26.7 features and run a validated 26.4.x configuration can stay, but only as a time-limited decision,

and the Tech Preview items (Spectrum-X DRA and ConnectX-9 Spectrum-X configuration) should stay out of production. The vendor performance claims for multiplane, such as 98% of theoretical line rate, are NVIDIA's own figures (developer.nvidia.com).

Introduction and Background

This report's scope is a version-specific networking preflight for teams that run Spectrum-X on Kubernetes, separate from GPU Operator DRA (Dynamic Resource Allocation) guidance and from generic RoCE (RDMA over Converged Ethernet) acceptance testing.

The **NVIDIA Network Operator** deploys networking components, including the DOCA-OFED driver (the successor to the MOFED driver stack), Multus, SR-IOV components and the Spectrum-X sub-operators, through one Helm chart and a NicClusterPolicy custom resource. Multus attaches multiple network interfaces to pods (github.com). The chart no longer creates the NicClusterPolicy itself, which matters for rollback because the policy is a separate artifact. NVIDIA's lifecycle page states that upgrades are supported only within a major release or to the next major release (docs.nvidia.com), so a 26.1.x cluster must pass through 26.4.x before 26.7.x.

Three version labels circulate and should not be confused. The Helm chart and GitHub tag **v26.7.0** are the base artifact; the v26.7.0 release was published on GitHub on 28 August (github.com). The 26.7.1 patch backports MOFED wait-node readiness handling (github.com) and updates the DOCA driver container version for the 26.7.x line. The documentation pages carry later footer dates: the 26.7.0 release notes read October 7, 2026 (docs.nvidia.com) and the 26.7.1 set reads November 5, 2026.

The analysis below uses dated vendor documentation and reproducible arithmetic, reports no benchmark results of its own, and is not a substitute for NVIDIA support or for site-specific engineering review.

Key Changes in Network Operator 26.7

Spectrum-X profiles become ConfigMaps

Through 26.4.x, `spectrumXOptimized.version` selected one of a fixed set of RA versions built into the NIC Configuration Operator image. In 26.7.0 the operator carries no built-in profiles, so every Spectrum-X deployment applies a profile before configuring NICs, and the field must match the name of a profile ConfigMap. The release notes describe the change as NIC tuning delivered as a ConfigMap instead of a built-in RA version (docs.nvidia.com).

The migration consequence is concrete. A template that still carries a value such as RA2.1 now resolves to a ConfigMap of that name, and Spectrum-X configuration is not applied until a matching ConfigMap exists (docs.nvidia.com). A template that references a missing profile is accepted by the API and fails only when applied, so admission does not protect the operator. Profile names must be unique across every watched namespace, and if two labeled ConfigMaps share a name the latest-reconciled one wins with no error. NVIDIA's guidance is to apply the profile ConfigMap before or together with the upgrade.

Profiles are not a public download: NVIDIA directs operators to the NVOnline portal for the profile published for their RA (docs.nvidia.com). NVIDIA presents one benefit, that tuning can be revised without a new operator release, which shifts risk from operator upgrades to profile governance.

Kubernetes floor of 1.32 and an RA 2.3 minimum of 1.35

Helm installation and upgrade fail on clusters below Kubernetes 1.32 (docs.nvidia.com). The platform table shows a general range of 1.32 to 1.36. RA 2.3 is narrower: the known-limitations table says it requires 1.35 or later, and the platform table lists upstream Kubernetes 1.35 to 1.36 on Ubuntu 24.04 and 22.04 for that row (docs.nvidia.com). NVIDIA adds that Kubernetes support for the RA is limited to a subset of OS and platform combinations, so Ubuntu 26.04 support added in 26.7.0 should not be read as RA 2.3 validation.

The upstream calendar narrows the window. Kubernetes maintains release branches for the three most recent minor releases (kubernetes.io) with roughly one year of patch support. Versions 1.32 and 1.33 appear in the releases page's end-of-life table with end dates of 2026-02-28 and 2026-06-28 (kubernetes.io). The documented range ends at 1.36, so 1.37, which has a published end-of-life date of October 28, 2027 (kubernetes.io), sits outside it.

OVS interface type moves from dpdk to doca

The release notes state that the Spectrum-X OVS interface type changed from `dpdk` to `doca` and that nodes must run an OVS build with DOCA support (docs.nvidia.com). OVS-DOCA is host provisioning outside the operator's scope.

NVIDIA's DOCA documentation shows the same naming in commands. The 3.6.0 guide creates bridges with `datapath_type=doca` (networking-docs.nvidia.com), adds ports with `type=doca` (networking-docs.nvidia.com), and enables initialization and offload with `doca-init` (networking-docs.nvidia.com) and `hw-offload` (networking-docs.nvidia.com). The 3.2.0 guide still used `type=dpdk` in its virtual-function MAC step (networking-docs.nvidia.com) and noted that OVS-DOCA shares most of its structure with OVS-DPDK (networking-docs.nvidia.com). Upstream OVS documents `type=dpdk` ports for its own DPDK datapath (docs.openvswitch.org), so a stock distribution package is not a substitute. NVIDIA's published documentation contains no explicit migration statement for the rename, so the check is empirical: inspect live bridge and port types on a canary node.

Hardware multiplane, IPv6 rail isolation, readiness and Data Direct

Hardware Multiplane with DOCA xPlane is now GA on ConnectX-8 SuperNICs, scoped to dual-plane and quad-plane B300 and GB300 deployments (docs.nvidia.com). The platform table lists hardware multiplane as recommended alongside software multiplane for RA 2.3. xPlane currently needs a ConnectX-8 SuperNIC with firmware 40.48.1000 or later (docs.nvidia.com). NVIDIA's technical blog says the operating system and collective libraries see a single RoCE device (developer.nvidia.com), and a failed plane is masked, with traffic redirected to the remaining planes in under 3 milliseconds (developer.nvidia.com) (a vendor claim).

The release also adds IPv6 rail isolation through the VRF CNI meta-plugin, automatic Data Direct enablement on Spectrum-X virtual functions, and per-component readiness conditions on NicClusterPolicy and NicNodePolicy. The VRF plugin provides no connectivity by itself and is only useful chained with other plugins

(www.cni.dev), which is why rail isolation depends on Multus, which the project recommends running in its thick-plugin form (github.com).

Tech Preview scope: Spectrum-X DRA and ConnectX-9

The Spectrum-X DRA option is an opt-in setting in NVIDIA Kubernetes Launch Kit, the manifest-generation tool, not an operator CRD field. It exists so that a node's rails can be divided among several workloads, and it requires Kubernetes 1.34 or later and the Tech Preview DRA Driver for SR-IOV (docs.nvidia.com). NVIDIA states that the driver has limited testing and is not recommended for production (docs.nvidia.com). It cannot run alongside the SR-IOV device plugin on the same cluster, enabling it on a cluster with device-plugin workloads is a migration rather than a toggle, and several workloads on one node must take different rails. Upstream, many of the DRA APIs in the `resource.k8s.io` group graduated to GA in Kubernetes v1.34 (kubernetes.io), so the Tech Preview label belongs to NVIDIA's SR-IOV driver, not to the Kubernetes API.

ConnectX-9 follows the same pattern. NVIDIA says it is outside RA 2.3, not covered by an RA-validated configuration, and for evaluation only in Spectrum-X mode (docs.nvidia.com). A profile covering it comes from NVIDIA Support or the account's CPM, and the release adds ConnectX-9 Network Bay configuration as a Tech Preview item.

Other field and schema changes

The `rawNvConfig` field is now limited to 128 entries, and index-range syntax such as a bracketed range is no longer accepted. The `pciPerformanceOptimized.maxAccOutRead` field is deprecated in favor of setting MAX_ACC_OUT_READ through `rawNvConfig`.

Table 1 below consolidates the 26.4 to 26.7 changes. The 26.4 column reflects the 26.4-era documentation and the wording of the 26.7 notes; where NVIDIA's published documentation states no 26.4 value, the cell says so.

Item	26.4.x behavior	26.7.x behavior	Pass/fail test before upgrade
`spectrumXOptimized.version`	Selected a built-in RA value such as RA1.3 to RA2.2	Names a profile ConfigMap; no built-in profiles	A labeled ConfigMap exists for every version string in use
Kubernetes range	26.4.0 added 1.35 support; minimum not stated in NVIDIA's documentation	1.32 to 1.36; install fails below 1.32	All nodes and API servers at 1.32 or later
RA 2.3 Kubernetes minimum	Not applicable	1.35 or later	Cluster at 1.35 or 1.36 if RA 2.3 is targeted
OVS interface type	`dpdk` (implied by the 26.7 wording)	`doca`; DOCA-capable OVS required (networking-docs.nvidia.com)	Live bridges report `datapath_type=doca`
`rawNvConfig`	Index ranges accepted (implied)	128 entries maximum; explicit indexes only	Entry count at or below 128; no ranges
`maxAccOutRead`	Available	Deprecated; use `rawNvConfig`	Value moved to `rawNvConfig`

Item	26.4.x behavior	26.7.x behavior	Pass/fail test before upgrade
Spectrum-X RA	26.4.0 notes cite RA 2.2; platform table lists RA 2.1 as Tech Preview	RA 2.3 GA	Decide target RA before choosing a profile
DRA	Tech Preview multi-workload support	Launch Kit opt-in remains Tech Preview	Keep off in production
Status reporting	Not stated	Readiness conditions and per-node status	Conditions read Ready after rollout

Most changes are schema or environment conditions that can be tested offline, which is why the preflight is written as pass/fail rows. One discrepancy deserves flagging: NVIDIA's pages disagree on which RA the 26.4 line carried, since the 26.4.0 notes cite RA 2.2 while the 26.7 notes describe Tech Preview support for RA 2.1 on 26.4.x. Sites should confirm the RA name in their existing template rather than infer it from a release number.

Pass/Fail Preflight Matrix and Who Should Stay Put

Who should stay on the current operator

Staying is defensible in four situations.

- **Kubernetes below 1.32.** The 26.7 chart cannot be installed or upgraded there.
- **RA 2.3 targeted but Kubernetes below 1.35.** There is no validated path, because RA 2.3 is validated only on 1.35 to 1.36.
- **No DOCA-capable OVS or no NVOnline profile.** These are hard preconditions that sites cannot satisfy by configuration alone.
- **A validated 26.4.x configuration that needs no new feature.** No urgent functional gap exists.

Staying carries a clock. The 26.4.x line is a maintenance version, and 26.1.x and lower receive no updates. The question is "when", not "whether". Because only the next major release is a supported upgrade target, waiting also means any later jump is staged.

The matrix

Table 2 below turns the release-note traps into a preflight with a status label. **Supported** means GA in NVIDIA's tables; **Preview** means Tech Preview, which NVIDIA defines as limited testing and not recommended for production (docs.nvidia.com). A row passes only when its test is met on every node in scope.

#	Check	Pass criterion	Status
1	Kubernetes floor	API servers and kubelets at 1.32 to 1.36	Supported
2	RA 2.3 Kubernetes minimum	1.35 or 1.36	Supported

#	Check	Pass criterion	Status
3	Upstream support remaining	Version not past end of life; 1.35 ends 2027-02-28 (kubernetes.io)	Informational
4	Profile ConfigMap present	One per RA version string, unique names	Supported
5	Profile covers every NIC	Coverage of 100% of device IDs and multiplane modes	Supported
6	OVS-DOCA on every node	Bridges use `datapath_type=doca` (networking-docs.nvidia.com)	Supported
7	Driver path chosen	DOCA-Host or containerized DOCA-OFED, never both	Supported
8	Hardware multiplane prerequisites	ConnectX-8 with firmware 40.48.1000 or later	Supported (GA)
9	Hugepages for OVS-DOCA and xPlane	400 MB plus 150 pages of 2 MB per SuperNIC	Supported
10	`rawNvConfig` shape	At most 128 entries, no ranges	Supported
11	Maintenance path	Maintenance Operator or equivalent drain process (github.com)	Supported
12	Spectrum-X DRA	Kubernetes 1.34 or later; driver not used with device plugin	**Preview**
13	ConnectX-9 Spectrum-X config	Evaluation only	**Preview**
14	Rollback artifacts	Captured and restorable (see Rollback section)	Site-specific

Rows 5 and 6 deserve the most scrutiny because they fail without an error at admission. A missing mlxConfig entry for a multiplane mode leaves those parameters at their device defaults, and NVIDIA's troubleshooting page lists "Spectrum-X parameters silently not applied, no error" as a named failure mode (docs.nvidia.com). It advises confirming that the profile covers every NIC model and mode in the cluster. Rows 12 and 13 exist so that Preview features are labeled in the same grid as Supported ones and are not enabled by accident. The hugepage figures in row 9 come from NVIDIA's components page, which describes the xPlane reservation as a second reservation beyond OVS-DOCA.

Compatibility grid by NIC

Table 3 below summarizes how each NIC family interacts with 26.7 Spectrum-X configuration. Rated speeds are NVIDIA marketing and manual figures and are vendor claims.

NIC family	Rated bandwidth (vendor)	Spectrum-X configuration in 26.7	Hardware multiplane
ConnectX-7	400 Gb/s	Spectrum-X templates accepted	Not listed for hardware multiplane in the platform table (API-validation behavior not verified)

NIC family	Rated bandwidth (vendor)	Spectrum-X configuration in 26.7	Hardware multiplane
BlueField-3 SuperNIC	Up to 400 Gb/s (www.nvidia.com)	NIC mode only	Not listed for hardware multiplane in the platform table
ConnectX-8 SuperNIC	Up to 800 Gb/s (www.nvidia.com)	RA 2.3 validated on Kubernetes 1.35 to 1.36	GA for dual and quad plane on B300 and GB300
ConnectX-9 SuperNIC	800 Gb/s per card in the hardware manual (networking-docs.nvidia.com); 1.6 Tb/s per GPU in marketing (www.nvidia.com)	**Preview**, outside RA 2.3	Not documented for 26.7 (unverified)

The two ConnectX-9 figures need not conflict if 1.6 Tb/s is an aggregate per GPU, but the sources do not say so, so a buyer should ask which figure a quoted system delivers. NVIDIA's 26.7 release notes scope hardware multiplane GA to ConnectX-8, and ConnectX-9 support for it is not documented for 26.7. ConnectX-8 integrates 48 lanes of PCIe Gen6 with an integrated switch (developer.nvidia.com).

Implementation Considerations and Process Changes

Step 1: inventory

Before any change, collect the inputs the preflight needs.

- **Helm values:** export the applied values and chart version and diff them against the 26.7.0 defaults. Helm can reuse the last release's values and merge overrides (helm.sh), which hides new defaults, so compare explicitly.
- **CRDs and custom resources:** list NicClusterPolicy, NicConfigurationTemplate, NicNodePolicy and SpectrumXRailPoolConfig objects and save them as YAML.
- **NIC device IDs:** record the PCI device ID of every NIC and note any BlueField in DPU mode, since the platform table lists BlueField-3 as NIC mode only.
- **OVS build and driver:** record the OVS package, datapath and port types, the driver path in use, and the DOCA-OFED tag. NVIDIA's 26.7 matrix lists doca3.5.0-26.07-0.7.7.0-1 (GA) and doca3.2.2-25.10-2.4.1.0-4 (LTS).
- **Kubernetes state:** record the version of every control-plane and kubelet component. Kubelets may lag the API server by up to three minor versions (kubernetes.io), so mixed fleets are common.

Per-node profile coverage is then a ratio: nodes whose every Spectrum-X NIC device ID and chosen multiplane mode appears in the profile, divided by total Spectrum-X nodes. The pass criterion is 100%.

Step 2: build the order of operations

Kubernetes upgrades and operator upgrades should be separate change windows. If RA 2.3 is the goal and the cluster sits at 1.32, the control plane must move through 1.33, 1.34 and 1.35, because the API server must not skip minor versions (kubernetes.io). Kubernetes recommends being on the latest patch of the current

minor first (kubernetes.io), and 1.35 is the last release to support containerd 1.x (kubernetes.io).

The operator change then has four parts, each with a documented caveat.

- **Profiles first.** Apply the ConfigMaps before or with the upgrade.
- **CRDs.** The chart carries a pre-install and pre-upgrade hook that upgrades the required CRDs (github.com), while Helm alone does not upgrade CRDs (helm.sh). Confirm the hook is not disabled in your values.
- **Helm flags.** When the admission controller is deployed through Helm, `--wait` must be appended to install and upgrade commands (github.com). A server-side dry run simulates the operation against the cluster (helm.sh). GitOps controllers that cannot tell install from upgrade need the chart's keepNCP behavior handled deliberately.
- **Component versions.** A Helm upgrade does not update component versions inside the NicClusterPolicy, so that edit is manual and is the step that triggers driver rollout.

Step 3: maintenance and drain requirements

The documents do not say that every Spectrum-X change drains a node, and the evidence supports narrower statements. When the DOCA-OFED driver loads, existing pods that use NVIDIA NICs lose their network interfaces (docs.nvidia.com). Safe driver loading therefore cordons and drains the node first. Helm upgrade only edits the driver DaemonSet, and the old driver keeps running until the pod is removed or the node reboots. For a manual reload NVIDIA says to remove secondary-network pods from the node first.

NIC configuration is coordinated with maintenance too. The NIC Configuration Operator uses the Maintenance Operator to prepare a node before configuring it, which cordons and drains on request through NodeMaintenance objects (github.com). States such as PendingReboot and PendingNodeMaintenance wait for a window, and a configuration reset triggers a node reboot. On external-BMC platforms an OS-level reboot does not reload NIC firmware, and the operator stack can loop trying to converge unless the firmware-reset setting is configured. Treating Spectrum-X NIC reconfiguration as a drain-and-reboot event is an inference from these statements, not a quoted rule, and it is the safe planning assumption.

Automatic driver upgrade is NVIDIA's preferred method, and its controls set the blast radius: a maxParallelUpgrades value of 0 means all nodes upgrade in parallel, drain and wait timeouts default to 300 seconds, and the drain pod selector must be configured so network-using pods are evicted. On the Kubernetes side, a PodDisruptionBudget that requires zero voluntary evictions makes a drain wait (kubernetes.io), and misbehaving budgets can block drains (kubernetes.io). Each drain command should target a single node, and DaemonSet pods require the ignore flag.

Downtime exposure is nodes times the per-node window, and elapsed time divides by parallelism. **(Hypothetical Example)** For 64 nodes with an assumed 45 minute window each, exposure is 64 x 45 = 2,880 node-minutes (48 node-hours). With a parallelism of 4, elapsed time is 16 waves x 45 minutes = 720 minutes (12 hours). Every input is a placeholder: measure the window on the canary, including reboot and firmware reset time. This arithmetic does not promise zero downtime and does not describe any real deployment.

Step 4: canary rollout and validation

Pick one or two nodes per distinct hardware class (NIC model, OS image, OVS build) and run the full change on them first. Validate in layers.

- **Control plane:** NicClusterPolicy conditions surface component health in machine-readable form, and per-node driver state appears in the nvidia.com/ofed-driver-upgrade-state label.
- **NIC configuration:** the NicDevice target state is "Configuration applied". A PartiallyApplied state means some parameters were skipped as unsupported, which is the visible counterpart of the silent-skip risk.
- **Rails:** SpectrumXRailPoolConfig reports an aggregate and a per-node status. Under hardware multiplane a degraded plane is invisible from inside a pod, and DOCA xPlane is the only place to see it.
- **RDMA throughput:** use the perftest suite. Its latency tests report minimum, median and maximum (github.com), with one-way latency taken as half the round trip (github.com), and bandwidth tests can run for a fixed duration (github.com). RDMA-CM is not VRF-aware, so tests that use it must run inside the rail's VRF context on IPv6 rails. For multi-node collectives, nccl-tests rely on MPI (github.com); its documented multi-node example runs 64 ranks across 8 nodes (github.com).
- **QoS:** compare host and switch settings to the RA values the site was given. NVIDIA's generic Cumulus Linux RoCE documentation is a reference for switch defaults, not a Spectrum-X RA mandate, so the site's RA documents govern.

Record each test with these fields: node, rail, port, test type (link pull, plane failure, node reboot), expected behavior, observed behavior, time to recovery, and pass or fail. Run at least one port failure per rail on the canary. NVIDIA reports a 2.68 millisecond hardware failover (developer.nvidia.com); this is a vendor claim to confirm on the site's own fabric, not to assume.

Rollback Artifacts and Recovery Sequence

NVIDIA's published documentation contains no dedicated rollback page for 26.7. The closest text is troubleshooting guidance: if the new driver pod still fails, change the DOCA-OFED driver version in the NicClusterPolicy to the previous version (docs.nvidia.com). Everything else must be assembled from component behavior, and the gaps should be understood before the window opens.

Why rollback is asymmetric

- **CRDs move forward.** The chart's hook upgrades CRDs on install and upgrade, while Helm documents no support for upgrading or deleting CRDs, citing the danger of unintentional data loss (helm.sh). If a CRD already exists Helm skips it with a warning (helm.sh). No CRD downgrade path appears in NVIDIA's published documentation, so treat the older schema as restorable only from a saved copy.
- **Driver rollout is node-by-node.** Nodes not yet rolled are unaffected by a chart rollback, and rolled nodes are not restored by it.
- **NIC configuration persists on the device.** Resetting configuration triggers a node reboot, and NVIDIA recommends a NIC configuration reset before switching between multiplane configurations. Reversing a tuning change is therefore a reset and reboot, not a chart operation. This is an inference from documented behavior.

- **Firmware order.** When firmware and configuration both apply, the firmware update runs first, so a firmware change is part of the rollback surface.

Artifacts to capture before the window

- **Helm state:** release history and user-supplied values. `helm history` lists revision numbers (helm.sh), and the history limit can be set so revisions are retained (helm.sh).
- **Custom resources:** full YAML for NicClusterPolicy, NicNodePolicy, templates and rail pool objects, plus the old and new profile ConfigMaps.
- **CRD manifests:** the CRDs as installed before upgrade, since no downgrade is documented.
- **Per-node baseline:** NIC non-volatile configuration, firmware versions, OVS bridge and port listing, and loaded driver version.
- **Signed images:** pin the 26.7.0 images. NVIDIA's NGC tag for the operator carries a digital signature (catalog.ngc.nvidia.com).
- **Profile source:** keep the NVOnline profile file for both RA versions, since profiles are not publicly downloadable.

Recovery sequence

1. **Stop the rollout.** Pause automatic driver upgrade by setting autoUpgrade to false, which makes the other upgrade options ignored, and stop any GitOps sync.
2. **Hold affected nodes cordoned.** If the Maintenance Operator is in use, a NodeMaintenance object is deleted to finish the operation (github.com), so keep it in place until the node is validated.
3. **Restore the driver version** in the NicClusterPolicy and recycle driver pods on affected nodes.
4. **Chart rollback if needed.** `helm rollback` takes a release and revision, can skip hooks (helm.sh), and can delete resources created in a failed rollback (helm.sh). Helm can also roll back automatically on failure when asked (helm.sh). Reapply saved CRDs separately.
5. **Restore profile and template.** Reinstate the prior ConfigMap and template version string, then confirm each NicDevice reaches its target state.
6. **Reset and reboot where tuning changed.** Plan this per node and re-run the canary suite before reopening scheduling.

Any step that reloads the driver interrupts pod networking on that node, so rollback is itself a maintenance event.

Procurement Questions for ConnectX-8/9 and Multi-Plane Designs

Buyers evaluating hardware for a 26.7 deployment should ask for written answers, using vendor documentation as the baseline.

- **Which NIC and firmware ships?** Hardware multiplane needs ConnectX-8 with firmware 40.48.1000 or later. NVIDIA strongly advises running all components at the most recent validated configuration, so ask which configuration version the system was validated against.

- **Is the RA profile included?** Profiles come from NVOnline or an NVIDIA CPM. Confirm entitlement and who maintains it.
- **Dual-plane or quad-plane?** NVIDIA describes multiplane as splitting each GPU's SuperNIC across two or more independent network planes (www.nvidia.com), each built as a two-tier fat tree (developer.nvidia.com). Operator GA scope covers B300 and GB300 dual and quad-plane designs.
- **What is the ConnectX-9 plan?** Spectrum-X configuration for ConnectX-9 is Preview and outside RA 2.3. Ask for the support path and profile delivery in writing, and note that ConnectX-9 interoperates with 25 to 800 Gb/s Ethernet switches (networking-docs.nvidia.com).
- **What are the host memory costs?** OVS-DOCA needs 400 MB of hugepages per SuperNIC from DOCA 3.4 (docs.nvidia.com), and xPlane needs at least 150 pages of 2 MB per SuperNIC as a second reservation. That is 700 MB per SuperNIC; with 8 SuperNICs per node (a placeholder count) the reservation is 5,600 MB, about 5.5 GiB.
- **Which performance claims are vendor-only?** NVIDIA claims a 1.6x AI network performance gain over off-the-shelf Ethernet (www.nvidia.com), scaling to 128K GPUs in two tiers (www.nvidia.com), 98% of theoretical line rate in a worst-case benchmark (developer.nvidia.com), and P99 tail latency of 8 to 9 microseconds at 75% load (developer.nvidia.com).
- **Which third-party figures repeat these claims?** ServeTheHome relays Hot Chips figures, including 1.6x RDMA bandwidth (www.servethehome.com), 1.7x fewer scale-out switches (www.servethehome.com) and 64x scale over multi-rail (www.servethehome.com), as NVIDIA claims. No independent benchmark of 26.7 multiplane behavior was found, so acceptance criteria belong in the contract and should be measured on site.
- **How is failure visible?** Ask how plane health reaches the site's monitoring, because a degraded plane is invisible inside the pod. Switch-side adaptive routing is also relevant: the switch generates ECN marks only after its adaptive routing capacity is exhausted (developer.nvidia.com), which changes how congestion tests are read.

Data Analysis and Evidence

This section collects the calculations used above with their inputs, so each can be rerun with site values.

Upstream Kubernetes runway. From November 24, 2026, Kubernetes 1.34 reached end of life 28 days earlier on October 27, 2026 (kubernetes.io). Version 1.35 ends February 28, 2027 (kubernetes.io), 96 days away, and 1.36 ends June 28, 2027 (kubernetes.io), 216 days away. Kubernetes' stated support model is roughly fourteen months per patch series (kubernetes.io). A site that upgrades to 1.35 only to unlock RA 2.3 therefore has about three months before upstream patches stop, unless it moves to 1.36 or buys vendor support. These dates come from the patch-releases page, which can lag the main releases page, so confirm before scheduling.

Kubernetes hops. A cluster at 1.32 needs three control-plane upgrades to reach 1.35, each its own change window (kubernetes.io). Both intermediate versions (1.33 and 1.34) are past end of life as of the report date. Cost is hops times validation time per hop.

Profile coverage. Coverage = covered nodes divided by total Spectrum-X nodes. **(Hypothetical Example)** With 64 nodes, of which 48 carry ConnectX-8 NICs that the profile covers and 16 carry another NIC with no matching mlxConfig entry, coverage is 48 / 64 = 75%. Under the documented behavior the 16 uncovered nodes would apply no Spectrum-X parameters and report no error, so the gate fails until the profile is extended or those nodes leave the Spectrum-X pool.

Hugepage budget. Per SuperNIC, 400 MB for OVS-DOCA plus 150 x 2 MB = 300 MB for xPlane gives 700 MB. Totals scale with the SuperNIC count per node (see the procurement questions). Earlier DOCA releases used 1 GB for OVS-DOCA, which changes the sum for sites on older DOCA.

Downtime exposure. Exposure in node-minutes = nodes x window. Elapsed time = ceiling(nodes / parallelism) x window. Setting maxParallelUpgrades to 0 means no limit, so all nodes are upgraded in parallel; set an explicit value to bound the exposure.

Evidence quality. Every specification quoted is from NVIDIA or an open-source project's own documentation, and performance numbers are labeled vendor claims. The source set contains no independent measurement of 26.7 behavior. An NVIDIA-authored multiplane paper on arXiv frames the architecture around hardware-accelerated load balancing in NICs and switches (arxiv.org).

Implications and Future Directions

Three structural changes in 26.7 point past this release. First, moving tuning into ConfigMaps decouples NIC tuning from operator releases, which shifts risk from operator upgrades to profile governance: who holds the profile, how it is versioned, and how coverage is tested. Teams that treated an RA string as an enum now own a configuration artifact.

Second, the compatibility surface is widening faster than validated combinations. The documented Kubernetes range spans five minor versions, yet RA 2.3 is validated on two, and the newest upstream minor, 1.37, sits outside the range (kubernetes.io). Procurement and platform teams should expect ranges and validated sets to diverge and should treat the validated set as the contract.

Third, many DRA APIs are GA upstream since Kubernetes v1.34 (kubernetes.io) while remaining Preview in NVIDIA's SR-IOV driver. When that driver reaches GA, the device-plugin to DRA move will be a planned project, not a toggle.

The release cadence matters as well. GitHub already lists 26.10.0 as the latest tag (github.com), while 26.7.x remains supported and 26.4.x is in maintenance. Sites planning a 26.4 to 26.7 move should read the 26.10 release notes before committing, because a move to 26.7 followed shortly by another supported hop doubles the change windows. This report did not review 26.10 documentation and makes no claim about it.

A practical planning principle is that the right unit of decision is the node class, not the cluster. Each class (NIC model, OS image, OVS build, firmware) should pass the matrix on its own, and the upgrade should proceed class by class with a recorded canary. That framing also limits exposure to the silent-skip failure mode, because coverage is measured per class before tuning is relied on.

Frequently Asked Questions (FAQs)

Where are the NVIDIA Network Operator 26.7 release notes?

They are published on NVIDIA's documentation site under the v26.7.0 and v26.7.1 documentation sets, with footer dates of October 7 and November 5, 2026 respectively. No formal release date is printed, and GitHub shows the v26.7.0 tag (github.com).

How do I run an NVIDIA Network Operator Spectrum-X upgrade?

Pass the preflight matrix, apply the profile ConfigMaps first, upgrade the chart, update NicClusterPolicy component versions manually, and roll the driver on a canary before the fleet.

How do I upgrade the NVIDIA Network Operator on Kubernetes with Helm?

Use the NGC chart at the target version with a values file, keep the CRD hook enabled (github.com), and append the wait flag if the admission controller is deployed through Helm (github.com). Check the release notes for the target version first.

What does a MOFED or DOCA-OFED driver upgrade involve?

Helm does not restart driver pods, automatic upgrade is the preferred route, and loading a driver removes network interfaces from running pods. Host-installed DOCA-Host and the containerized driver are mutually exclusive per node.

What is the NVIDIA Network Operator rollback procedure?

NVIDIA documents setting the driver version back to the previous value. Chart rollback uses Helm revision numbers (helm.sh), but Helm does not downgrade CRDs (helm.sh). Restore saved CRDs, templates and profiles, and expect a reset and reboot where NIC tuning changed.

How do I configure Spectrum-X on Kubernetes under 26.7?

Apply a profile ConfigMap, reference it by name in the template, set the multiplane mode explicitly because the default is none, and validate with NicDevice and rail pool status.

What belongs on an upgrade checklist?

Table 2 above is the checklist: Kubernetes version, RA 2.3 minimum, profile presence and coverage, OVS-DOCA, driver path, firmware, hugepages, `rawNvConfig` shape, maintenance path, Preview exclusions, and rollback artifacts.

Can I upgrade with zero downtime?

No such promise is supported by the documentation reviewed, because driver loading removes pod network interfaces, so plan per-node maintenance windows.

Conclusion

NVIDIA Network Operator 26.7 is a sound target for Spectrum-X sites that can meet its preconditions, and a poor target for those that cannot. The preconditions are specific: Kubernetes at 1.32 or later, and 1.35 or later for RA 2.3; a DOCA-capable OVS build; a profile ConfigMap that exists for every RA string and covers every NIC device ID and multiplane mode; firmware and hugepage settings that match the multiplane design; and a maintenance process that tolerates per-node driver reloads.

The headline features, RA 2.3, hardware multiplane GA on ConnectX-8, IPv6 rail isolation and readiness conditions, are documented. The Tech Preview items are not production material by NVIDIA's own definition. The key operational point is that a profile coverage gap produces no error, so coverage must be measured, not assumed.

Rollback has to be engineered in advance, because the documentation offers a driver-version reversal and Helm offers revision rollback but no CRD downgrade (helm.sh). The ordering in this report, inventory, preflight, canary, recorded failure tests, then fleet rollout, is the shortest path that keeps each failure visible and reversible. Verify every combination against NVIDIA's current platform-support table and a canary before relying on it.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://docs.nvidia.com/networking/display/kubernetes2670/life-cycle-management.html
2. https://github.com/Mellanox/network-operator/releases
3. https://docs.nvidia.com/networking/display/kubernetes2670/release-notes.html
4. https://kubernetes.io/releases/patch-releases
5. https://kubernetes.io/releases/version-skew-policy
6. https://helm.sh/docs/chart_best_practices/custom_resource_definitions
7. https://developer.nvidia.com/blog/giga-scale-ai-ethernet-evolution-spectrum-x-ethernet-rewrites-rules
8. https://github.com/k8snetworkplumbingwg/multus-cni
9. https://github.com/Mellanox/network-operator/releases/tag/v26.7.0
10. https://docs.nvidia.com/networking/display/kubernetes2670/spectrum-x/spectrum-x-configuration.html
11. https://docs.nvidia.com/networking/display/kubernetes2670/platform-support.html
12. https://kubernetes.io/releases
13. https://networking-docs.nvidia.com/doca/archive/3-6-0/ovs-doca-hardware-acceleration.md
14. https://networking-docs.nvidia.com/doca/archive/3-2-0/ovs-doca-hardware-acceleration.md
15. https://docs.openvswitch.org/en/latest/howto/dpdk
16. https://docs.nvidia.com/networking/display/kubernetes2670/spectrum-x/components.html
17. https://www.cni.dev/plugins/current/meta/vrf
18. https://docs.nvidia.com/networking/display/kubernetes2670/dra-sriov-driver/dra-sriov-driver.html
19. https://kubernetes.io/blog/2025/09/01/kubernetes-v1-34-dra-updates

20. https://github.com/Mellanox/maintenance-operator
21. https://docs.nvidia.com/networking/display/kubernetes2670/spectrum-x/verify-and-troubleshoot.html
22. https://www.nvidia.com/en-us/networking/products/ethernet/supernic
23. https://networking-docs.nvidia.com/connectx9hw/introduction
24. https://developer.nvidia.com/blog/nvidia-connectx-8-supernics-advance-ai-platform-architecture-with-pcie-gen6-connectivity
25. https://helm.sh/docs/helm/helm_upgrade
26. https://kubernetes.io/blog/2025/12/17/kubernetes-v1-35-release
27. https://github.com/Mellanox/network-operator/blob/v26.7.0/deployment/network-operator/README.md
28. https://kubernetes.io/docs/tasks/run-application/configure-pdb
29. https://github.com/linux-rdma/perftest
30. https://github.com/NVIDIA/nccl-tests
31. https://helm.sh/docs/helm/helm_rollback
32. https://catalog.ngc.nvidia.com/orgs/nvidia/mellanox/containers/spectrum-x-operator/network-operator-v26.7.0
33. https://www.nvidia.com/en-us/networking/spectrumx
34. https://www.servethehome.com/nvidia-spectrum-x-ethernet-multiplane-network-architecture-at-hot-chips-2026
35. https://arxiv.org/abs/2605.21187

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the hardware library, GPU references, networking references and vendor research. Contact GPU Smith with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: https://gpusmith.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.