

Moving from NVDS to VDS

NSX can be deployed on an ESXi host using two different kinds of virtual switches: the NSX Virtual Switch (NVDS) or the vSphere Distributed Switch (VDS). The VDS-based model supersedes the other, and the NVDS is being deprecated as of NSX 4.0. This document focuses on the transition NVDS to VDS.

Table of Contents

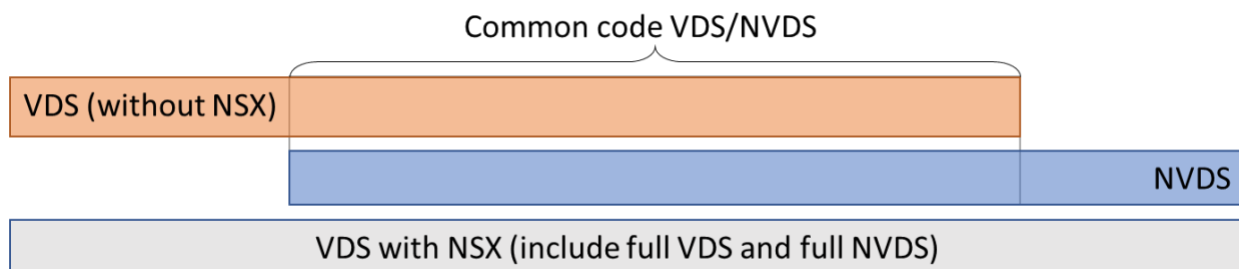
- Introduction: from VDS to NVDS and back 2
- VDS model vs. NVDS model 2
 - VDS without NSX 3
 - NVDS configuration model 4
 - VDS running NSX 6
 - Summary 7
- Converting NVDS to VDS 7
 - Preparing a VDS in vCenter 7
 - Evacuate VMs from NVDS-based hosts 7
 - Replace the NVDS hostswitch 8
- Manual migration 8
 - Prepare the VDS in vCenter 11
 - Put a host in maintenance mode 11
 - Free one physical uplink from the NVDS 11
 - Assign the free uplink to the VDS and migrate the vmkernel interfaces 13
 - Removing the NVDS 14
 - Install NSX on the VDS hostswitch 17
- Converting NVDS to VDS using the NSX manager CLI 19
 - Procedure overview 22
 - Run the pre-check 22
 - Apply the topology 23
 - Migrate the ESXi hosts 24
- Converting NVDS to VDS using API 27
 - Procedure overview 30
 - Run the pre-check 30
 - Apply the topology 31
 - Migrating the hosts 33
 - Migrating by cluster 34
- Converting NVDS to VDS while upgrading 37
 - Procedure overview 39
 - Run the pre-check 39
 - Apply the topology 42
 - Prepare the baseline for upgrading the ESXi hosts 43
 - Upgrade and migrate the hosts 47

Introduction: from VDS to NVDS and back

NSX-T was developed from the ground up as a completely independent networking and security platform. One of its goals was that we could run it on all types of hypervisors, natively on non-virtualized servers or in the cloud. NSX was supposed to be independent of vCenter too, but if that latter was initially an optional compute manager, it became quickly obvious that all our customers running NSX on ESXi were also using vCenter. As of NSX 3.0, we have introduced the capability of running NSX on the top of vCenter's native virtual switch, the vSphere Distributed Switch (VDS), instead of using the NSX Virtual Distributed Switch (NVDS.) Running NSX on VDS has the following benefits:

- Simpler install. Existing vSphere customers do not need migrating to NVDS to install NSX.
- Compatibility with many third-party applications that only expect VDS dvportgroups (the NVDS presented NSX segments as "opaque networks" in vCenter and many applications were not retrofitted to handle those.)
- Single product to maintain and enhance

This transition to running NSX on VDS was relatively straightforward because the NVDS was in fact based on VDS code. The name NVDS was in fact chosen to highlight this close relationship. The following diagram tries to represent the overlap between the code of the NVDS and the VDS. The NVDS is nothing more than a VDS, stripped from its capability to handle standard dvportgroups, but with the added NSX functionality:



When running NSX on VDS, we are just adding the NSX capability to a fully functional VDS (or, from another angle, we are adding back the capability of running standard VDS dvportgroups on to the original NSX virtual switch.) Whenever you have a question about feature parity, rest assured that all NSX features available on NVDS are also available on VDS prepared for NSX: the whole NVDS is in fact included in this model!

That said, when running NSX on NVDS, NSX segments look like "opaque networks" in vCenter, whereas they look like "NSX dvportgroups" when running NSX on VDS. This leads to few operational differences that are treated thoroughly in KB articles [79872](#) and [82487](#) and that are not the focus of this document.

VDS model vs. NVDS model

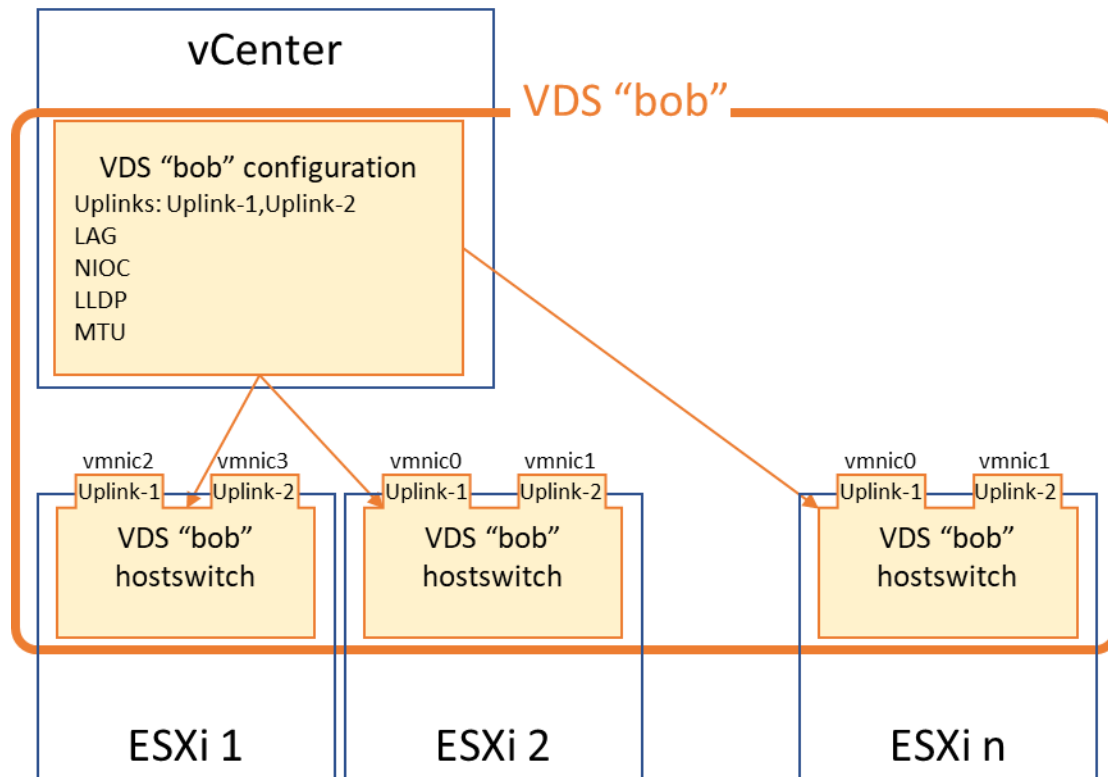
In most cases, converting an existing NVDS deployment to VDS is straightforward. The configuration model of the NVDS is however more elaborate than the VDS one and there are scenarios where an NVDS will have to be translated into multiple different VDS. This part details the configuration model for our different virtual switches, a good investment of your time that will surely pay off when migrating your NVDS manually or with the tool embedded in NSX.

VDS without NSX

The vSphere Distributed Switch has in fact two components:

- A configuration object, stored in vCenter
- A data plane component (a virtual switch) instantiated on the ESXi hosts

Both components are often individually referred to as “the VDS”, but it will be useful to make a clear distinction between them and for that, I will call the former “VDS configuration” and the latter “VDS hostswitch”.



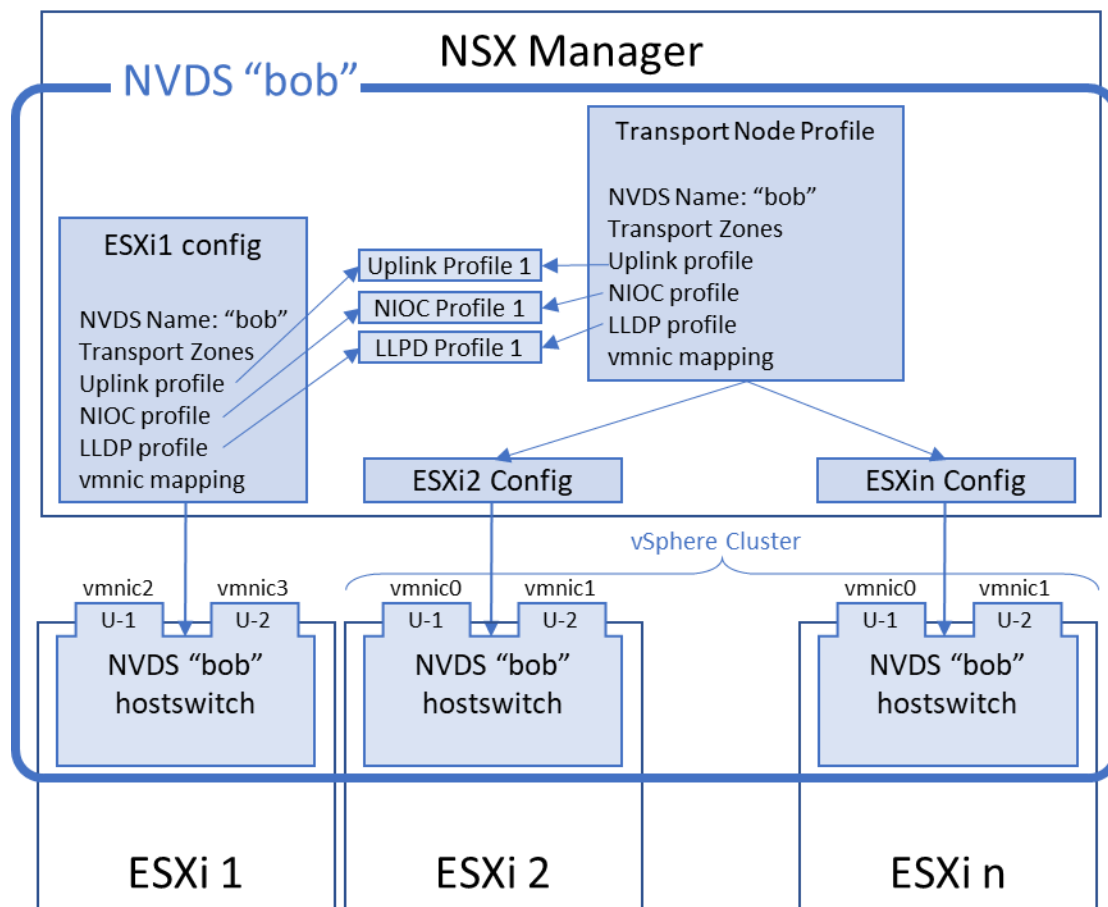
What is very specific to the VDS is that the same configuration is applied to every hostswitch. The only thing that is configured on a per-host basis and could potentially differ between hosts is the mapping between the VDS uplinks names (in the above diagram Uplink-1 and Uplink-2) and the host physical NICs. The diagram above is representing this possibility, with ESXi 1 having its VDS uplinks mapped to physical uplinks vmnic2 and vmnic3, while the other hosts have their VDS uplinks mapped to vmnic0 and vmnic1. Otherwise, notable parameters like the definition of Link Aggregated Groups (LAGs), Network I/O Control parameters, link discovery protocol settings (like Link-Level Discovery Protocol, or Cisco Discovery Protocol) and Maximum Transmission Unit (MTU) are applied consistently to all VDS hostswitches.

Finally, the VDS is uniquely identified by a name. In the above example, vCenter would not let the user create another VDS named “bob”. This is a significant difference with the NVDS, as we will see below.

NVDS configuration model

In NSX, there is no global configuration for an NVDS. There is a dedicated configuration in the NSX manager for the NVDS hostswitch of each ESXi host. To avoid individual configuration of many similar hosts, a template, called a Transport Node Profile (TNP) can be applied to all the hosts of a vSphere cluster. The use of a TNP requires NSX to be linked to vCenter, in order to identify the clusters.

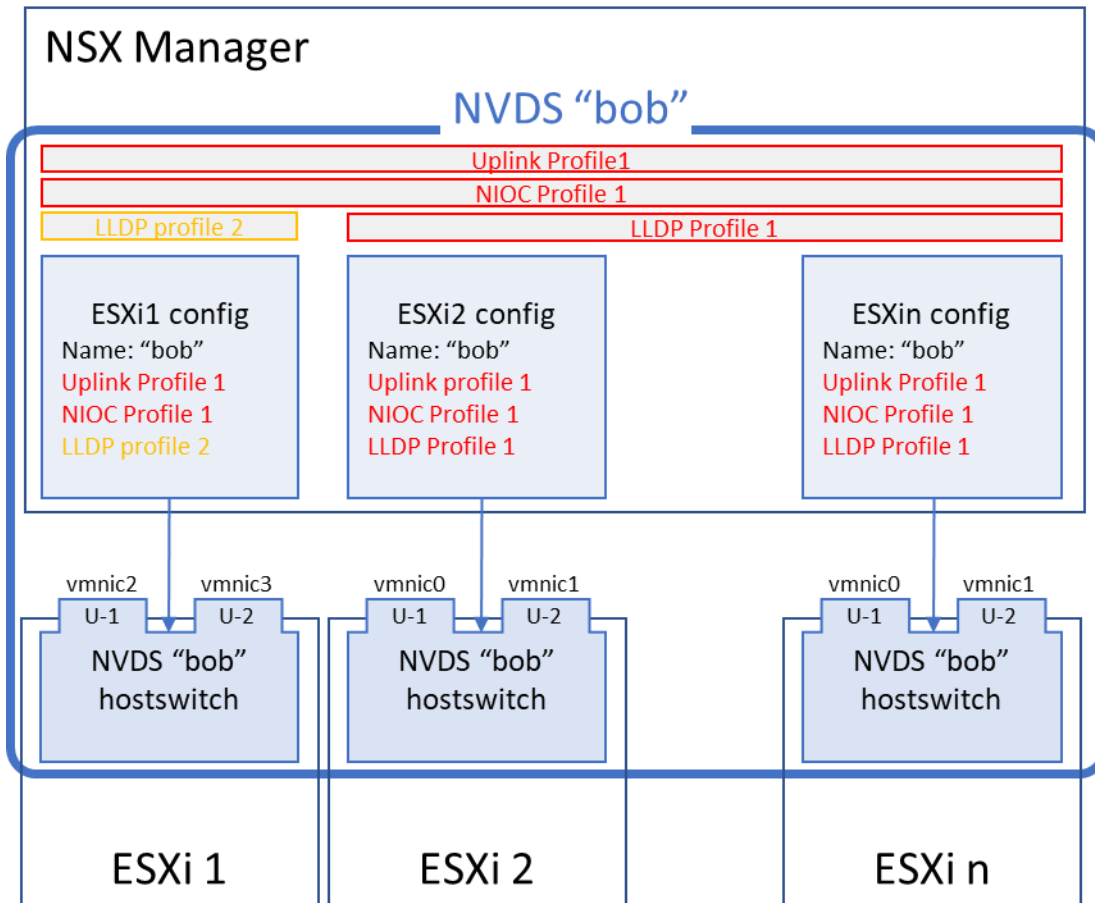
The following diagram shows the configuration for an NVDS called “bob” across multiple ESXi hosts. ESXi1 is configured individually, while ESXi2 to ESXi n are part of a cluster to which a TNP is applied.



Let us first focus on the configuration of the NVDS hostswitch for ESXi1. The arrows are representing references to global configuration objects. For example, the uplink profile section of the NVDS hostswitch for ESXi1 is referring to a global configuration object called “Uplink Profile 1”. Whatever is configured in “Uplink Profile 1” is applied to the NVDS hostswitch of ESXi1. If the administrator modifies “Uplink Profile 1”, the changes are propagated to the NVDS hostswitch of ESXi1 (and to all host hostswitches referring to “Uplink Profile 1”). The same applies to “NIOC Profile 1” and “LLDP Profile 1”. Without going into the detail of those global profiles, you can see that they represent the global configuration for the NVDS.

The Transport Node Profile is very similar the configuration of ESXi1 hostswitch. The TNP can be seen as a template that is pushed to the individual configuration of the ESXi hosts (ESXi2 to ESXi_n) which are part of the vSphere cluster to which the TNP was applied. Note that the administrator can still make changes to the individual configuration of an ESXi host, overriding the TNP configuration.

The following diagram illustrates the flexibility introduced by the NVDS configuration model. As mentioned earlier, the global profiles can be seen as a global configuration for the NVDS. However, it is always possible to carve out an exception to this global configuration.



Here, all the NVDS hostswitches are following the exact same configuration, except ESXi1, which is using "LLDP Profile 2" instead of "LLDP Profile 1". This is still the same NVDS "bob" spanning all those ESXi hosts, it is just that it does not have a consistent configuration applied to each hostswitch. This illustrates the main issue that can happen when moving to a VDS. The VDS can provide all the features of the NVDS, but its configuration is monolithic. Here, NVDS "bob" cannot be converted to a unique VDS "bob", because a unique VDS "bob" can only have a single LLDP configuration applied to all hosts ESXi1 to ESXi_n. The solution will be to either drop the exception for ESXi1 (by configuring it to use "LLDP Profile 1" instead of "LLDP Profile 2"), or to create a dedicated VDS "bob1" for ESXi1, and VDS "bob2" for the other hosts.

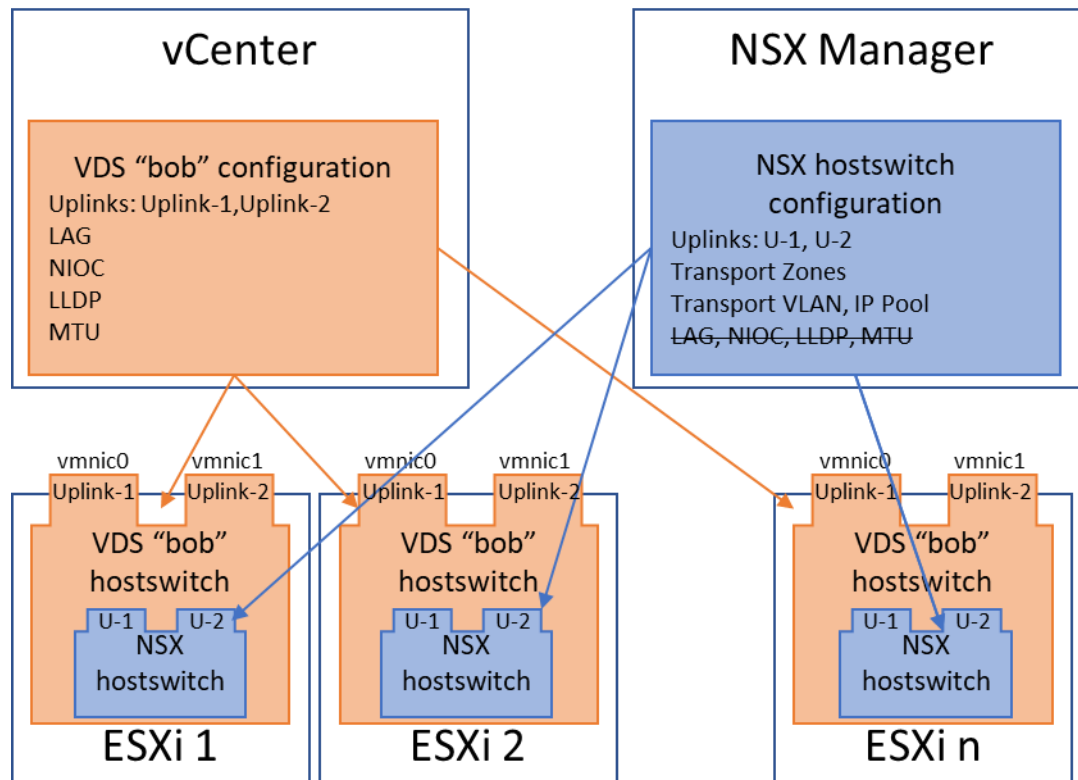
The next part focuses on the configuration of a VDS running NSX and will help identify the NVDS configuration that result in the creation of multiple VDS during migration.

VDS running NSX

Running NSX on VDS means “enhancing” a regular VDS hostswitch with NSX-specific capabilities. An unofficial way to represent this would be to show an NVDS hostswitch instantiated inside the standard VDS hostswitch. All the parameters that were common to VDS and NVDS are now part of the VDS. Those parameters, represented in the following diagram, are:

- The hostswitch Maximum Transmission Unit (MTU)
- The Network IO Control (NIOC) configuration
- The LLDP configuration. NSX only supported LLDP, going back to the VDS allows LLDP and Cisco Discovery Protocol (CDP.)
- The Link Aggregation Groups (LAGs) definition

The following diagram shows some ESXi hosts configured for NSX on the top of a VDS. The NSX functionality is represented as an embedded NSX hostswitch inside the VDS hostswitch. The VDS configuration is represented in orange, the NSX configuration in blue. The NSX configuration parameters that have been migrated to the VDS configuration are struck through.



In the NSX on VDS model, the blue NSX configuration still benefits from the flexibility of the NVDS. For example, two ESXi hosts can have different transport zones while still running the same VDS "bob". This is because the transport zone configuration belongs to the NSX and is still configured per-host. On the other hand, the ESXi hosts running VDS "bob" must have the same MTU, because the MTU is a VDS parameter, stored in the monolithic VDS configuration in vCenter.

Summary

When moving an NVDS to a VDS running NSX, some virtual switch parameters that once belonged to the NVDS are now part of the monolithic VDS configuration. Those parameters are:

- LAG definition (part of uplink profile in NSX)
- MTU (part of uplink profile in NSX)
- LLDP
- NIOC

Any discrepancy in the value of those parameters must result in the creation of separate VDS, even if the original hosts had an NVDS hostswitch with the same name. This is because the VDS configuration can only accommodate for a global value for those parameters, not a per-host value.

Converting NVDS to VDS

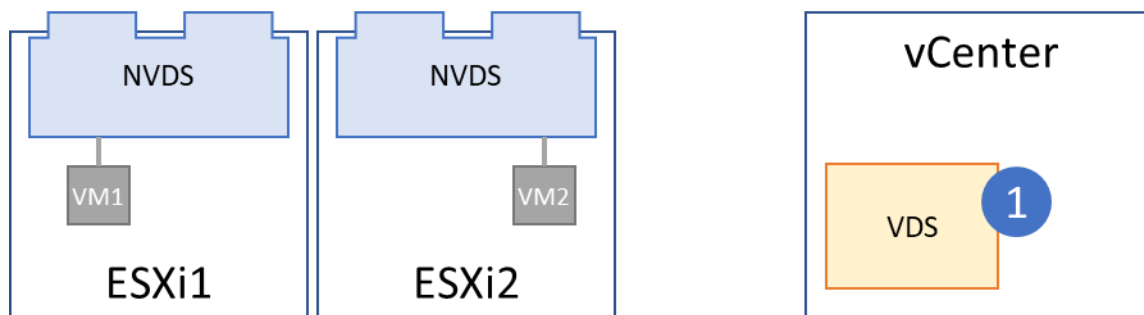
There are many ways of converting an NSX deployment based on NVDS to a VDS-based one. This document will describe step-by-step migrations:

- without help from the NSX migration tool
- CLI based, using the NSX migration tool
- API based, using the NSX migration tool
- And finally, UI based, using the NSX migration tool and the vCenter Upgrade Manager.

All those different methods are following the same steps, detailed below.

Preparing a VDS in vCenter

We are going to assume that the NVDS is going to be migrated to a single VDS (migrating to multiple VDS is just a repeat of this procedure for each VDS.) This VDS must be prepared in vCenter. When using the NSX migration tool, a new VDS will always be created for the purpose of the migration.

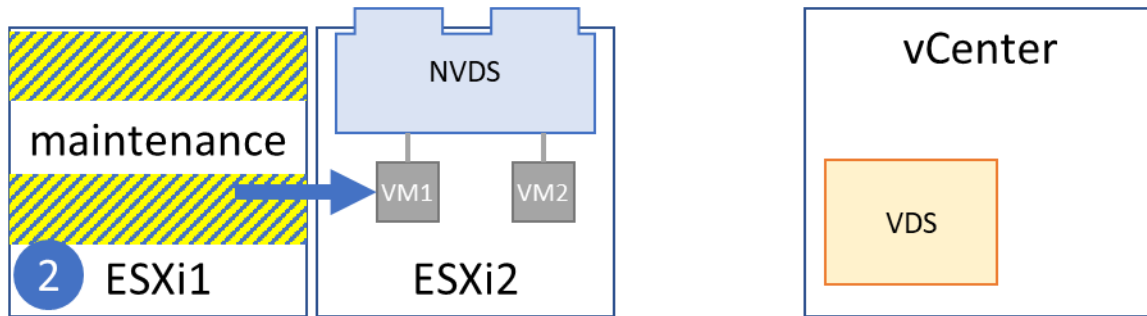


Some dvportgroups might have to be provisioned on the VDS. If some vmkernel interfaces were attached to NSX VLAN segments on the NVDS, they need to be moved to a dvportgroup backed by the same VLANs on the VDS.

Evacuate VMs from NVDS-based hosts

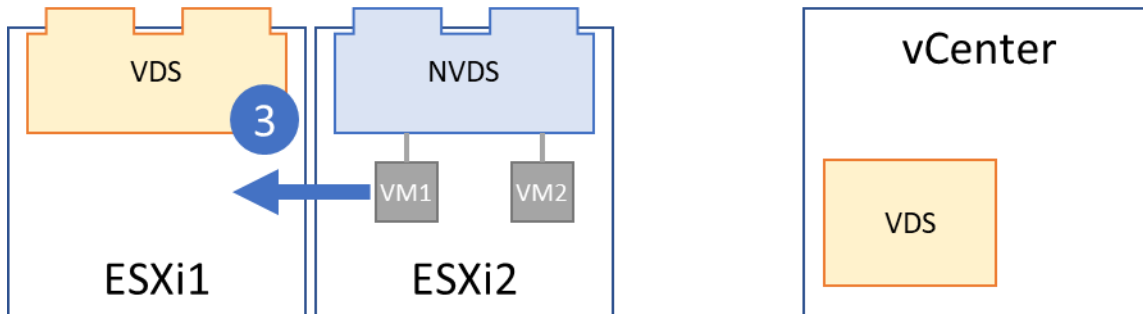
Once the VDS configuration is properly setup in vCenter, it is time to start transitioning the NVDS hostswitches on the ESXi transport nodes to a VDS hostswitch. The operation would be disruptive for the payloads running on the host, they must be evacuated. Typically, the ESXi host is part of a cluster

configured with the vSphere Distributed Resource Scheduler (DRS), and putting the host in maintenance mode will lead the DRS to relocate the VMs.



Replace the NVDS hostswitch

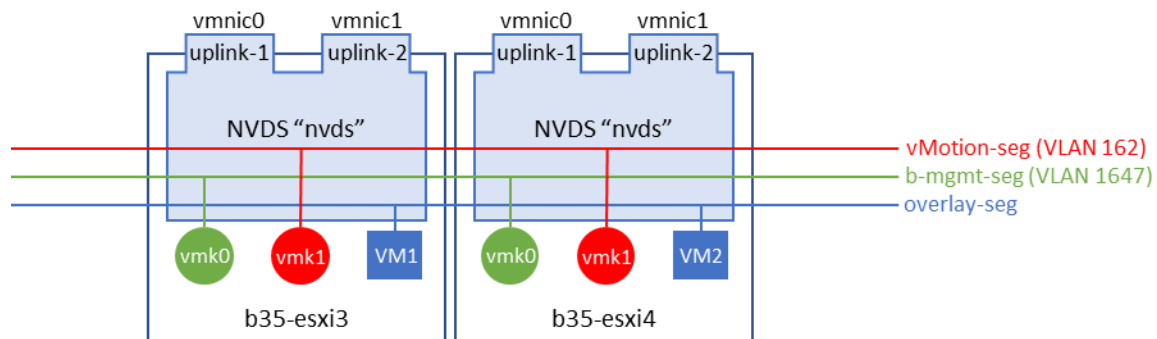
Finally, the NVDS hostswitch is replaced by a VDS hostswitch prepared for NSX. This operation can be performed in multiple steps (first deploy the VDS hostswitch, then prepare it for NSX.)



Once the VDS hostswitch is in place, the host can exit the maintenance mode and VMs can be moved back to it. There is absolutely no problem running NSX with a mix of hosts prepared with VDS and NVDS. The NVDS hostswitch of the other hosts are converted to a VDS hostswitch following the same procedure.

Manual migration

In this example, my ESXi hosts have two uplinks, both belonging to a single NVDS hostswitch.



The hosts have two vmkernel interfaces each:

- vmk0: management interface attached to NSX VLAN segment “b-mgmt-seg” (which appears as an opaque network in vCenter), backed by VLAN 1647
- vmk1: vmotion interface, attached to NSX VLAN segment “vmotion-seg”, backed by VLAN 162

This represents the most complex migration scenario possible as we need to relocate the management interface without losing connectivity.

The hosts have been prepared with a TNP called “NVDS TNP”, represented in the following screenshot:

Edit Transport Node Profile - NVDS TNP



Name* NVDS TNP

Description

+ ADD SWITCH

▼ nvds

Type* N-VDS VDS

Mode* Standard ⓘ
 ENS Interrupt ⓘ
 Enhanced Datapath ⓘ

Name* ⓘ nvds

Transport Zone* nsx-overlay-transportzone x nsx-vlan-transportzone x
[OR Create New Transport Zone](#)

NIOC Profile x
[OR Create New NIOC Profile](#)

Uplink Profile* nsx-default-uplink-hostswitch-profile
[OR Create New Uplink Profile](#)

LLDP Profile LLDP [Send Packet Disabled] x

IP Assignment (TEP)* Use IP Pool

IP Pool* TEP-Pool

Teaming Policy Uplink Mapping

Uplinks	Physical NICs
🔌 uplink-1	vmnic0
🔌 uplink-2	vmnic1

PNIC only Migration No

Network Mappings for Install [2 Mappings](#)

Network Mappings for Uninstall [Add Mapping](#)

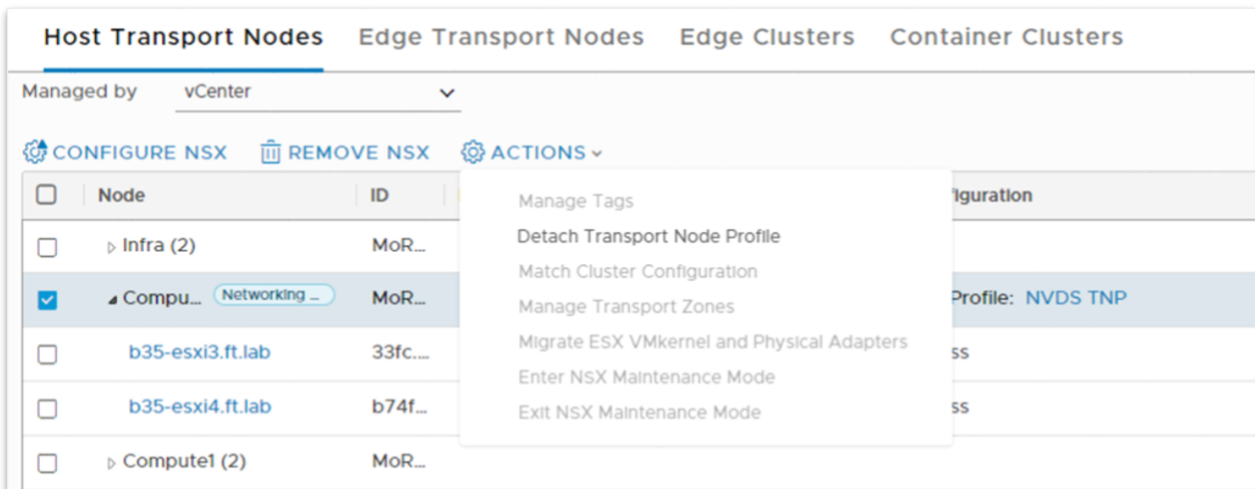
CANCEL SAVE

Prepare the VDS in vCenter

We are creating the configuration for a new VDS called “newVDS” in vCenter. Again, we are going to assume that the NVDS can be migrated to a single VDS. If there was the need for multiple VDS, this would just require running the same kind of migration procedure for each of those VDS.

We need to create a dvportgroup “management-dvpg”, backed by VLAN 1647 and corresponding to “b-mgmt-seg” and a dvportgroup “vmotion-dvpg” backed by VLAN 162 and corresponding to “vmotion-seg”.

Finally, we are going to disconnect the ESXi hosts from the TNP of the cluster. This will have no effect on NSX, it is just that the TNP would interfere with our migration procedure, and we will need to replace it with a VDS-TNP at the end of it anyway. The following screenshot shows how to detach the TNP:



Put a host in maintenance mode

The goal here is to evacuate virtual machines that would be running on the first host we are migrating. The assumption is that the cluster is properly configured with the DRS, otherwise the migration of the VMs using NSX segments must be made manually.

Free one physical uplink from the NVDS

Connectivity to the host management interface depends on the NVDS. It's not possible to simply delete the NVDS and create a replacement VDS. We need to configure a VDS hostswitch providing connectivity to the vmkernel interfaces of the host before we can delete the NDVS.

The NVDS has two redundant uplinks: we are going to free one of them and assign it to the VDS.

Removing an uplink from the NVDS is achieved by editing the NVDS configuration in the NSX manager. The host had uplink-1 mapped to vmnic0 and uplink-2 mapped to vmnic1, we are going to delete the

mapping of uplink-2 in the NVDS configuration, as shown in the screenshot below:

NSX Installation

- 1 Host Details
- 2 Configure NSX**

Configure NSX

+ ADD SWITCH

nvds

Type ***** N-VDS VDS

Mode ***** Standard ⓘ ENS Interrupt ⓘ Enhanced Datapath ⓘ

Name ***** ⓘ nvds

Transport Zone ***** nsx-overlay-transportzone x nsx-vlan-transportzone x

[OR Create New Transport Zone](#)

NIOC Profile nsx-default-nioc-hostswitch-profile x [OR Create New NIOC Profile](#)

Uplink Profile ***** uplinkProfile1 [OR Create New Uplink Profile](#)

LLDP Profile LLDP [Send Packet Disabled] x

IP Assignment (TEP) ***** Use IP Pool

IP Pool ***** TEP-Pool

Teaming Policy Uplink Mapping

Uplinks	Physical NICs
🔗 uplink-1	vmmnic0
🔗 uplink-2	Device Name/Key x

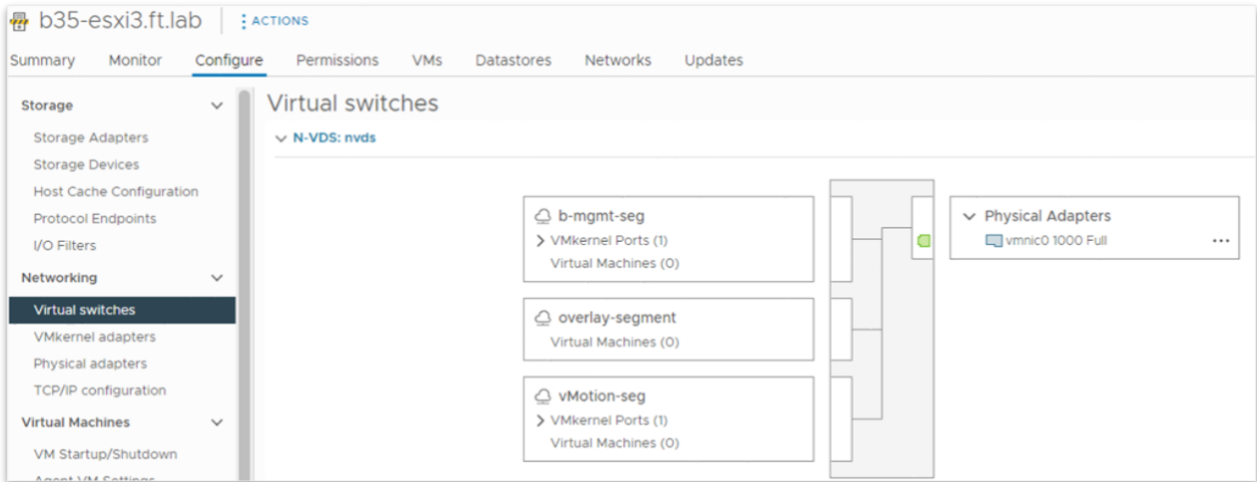
PNIC only Migration No

Network Mappings for Install **2 Mappings**

Network Mappings for Uninstall **Add Mapping**

CANCEL **PREVIOUS** **FINISH**

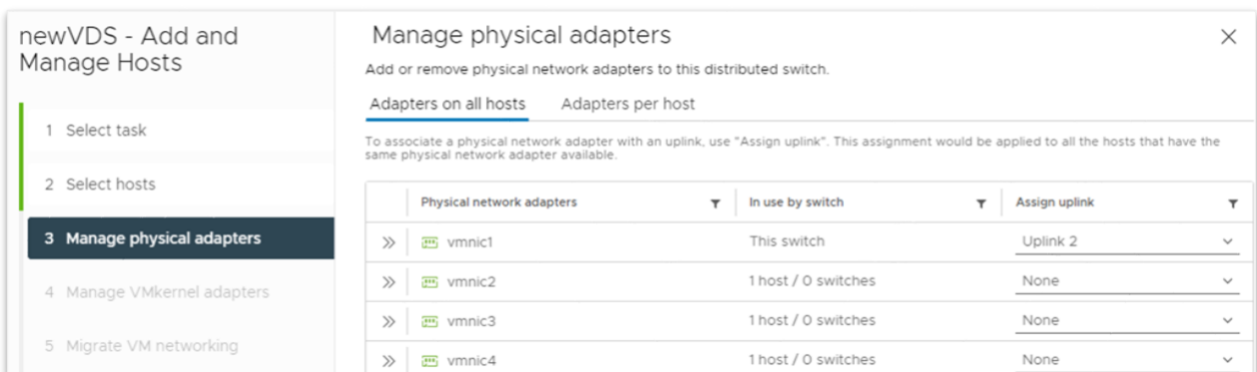
After the physical uplink vmnic1 is removed, the host just lost redundancy, but it kept full connectivity via vmnic0.



Assign the free uplink to the VDS and migrate the vmkernel interfaces

The ESXi host now has a free physical uplink vmnic1, that we are going to use to install a VDS hostswitch. At the same time, we will migrate the vmkernel interfaces to the appropriate dvportgroups we have pre-created on the VDS.

Those steps are easily and safely achieved thanks to the vCenter UI. In the “Add and Manage Hosts” menu for “newVDS”, assign newVDS “Uplink-2” to vmnic1:



Then associate vmk0 and vmk1 to their respective dvportgroups “management-dvpg” and “vmotion-dvpg”:

newVDS - Add and Manage Hosts

- 1 Select task
- 2 Select hosts
- 3 Manage physical adapters
- 4 Manage VMkernel adapters
- 5 Migrate VM networking
- 6 Ready to complete

Manage VMkernel adapters

Manage and assign VMkernel network adapters to the distributed switch.

Adapters on all hosts Adapters per host

To assign vmkernel network adapter to port group, click on the arrow or "Assign port group" button. This assignment would be applied to all hosts that have the same vmkernel network adapter available.

Name	NSX port group ID	Distributed switch	Actions
b-mgmt-seg	/infra/segments/b-mgmt-seg	newVDS	ASSIGN
management-dvpg	--	newVDS	UNASSIGN
overlay-segment	/infra/segments/overlay-seg	newVDS	ASSIGN
vmotion-dvpg	--	newVDS	ASSIGN
vMotion-seg	/infra/segments/vMotion-seg	newVDS	ASSIGN

At the end of this step, we have two hostswitches on the host, each with a single uplink. The vmkernel interfaces are attached to the VDS, so we can now safely delete the NVDS hostswitch:

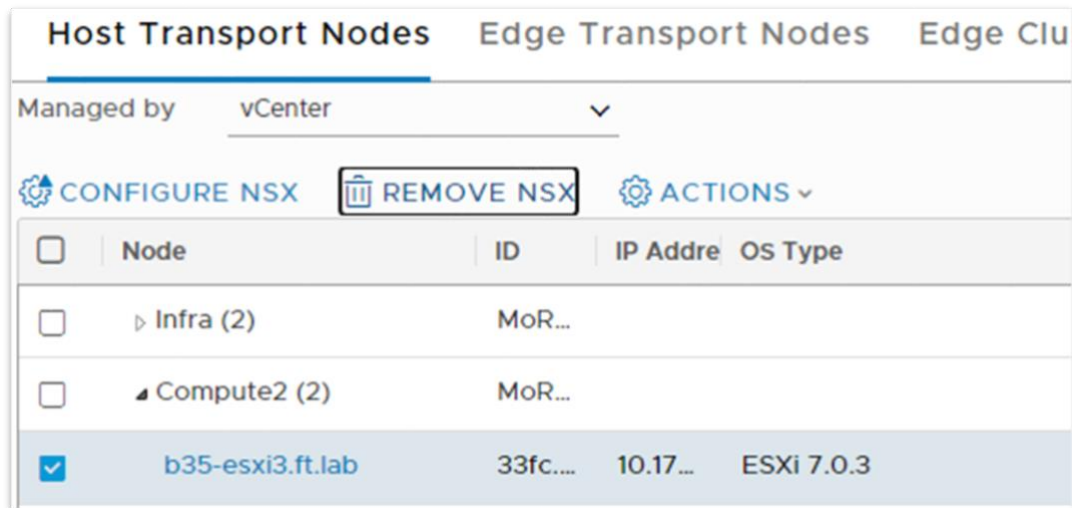
The screenshot shows the vSphere configuration interface for host 'b35-esxi3.ft.lab'. The 'Configure' tab is active, and the 'Virtual switches' section is expanded. Under 'Distributed Switch: newVDS', two port groups are shown: 'management-dvpg' (VLAN ID: 1647) and 'vmotion-dvpg' (VLAN ID: 162). Both are connected to 'newVDS-DVUplinks-55'. Under 'N-VDS: nvds', three port groups are shown: 'b-mgmt-seg', 'overlay-segment', and 'vMotion-seg'. These are connected to 'Physical Adapters' (vmnic0 1000 Full).

Removing the NVDS

Here, we have multiple options to get rid of the NVDS hostswitch.

Option 1: delete the NVDS from the NSX manager

The simplest one is to outright delete it from the NSX manager. This is achieved in a single step, as illustrated by the following screenshot:



The issue with removing NSX from the ESXi host is that the NSX manager will have to push the NSX binaries again, when preparing the host for NSX using the VDS. Preparing the VDS for NSX will thus be slower than if we could simply re-configure NSX for using the VDS instead of the NVDS.

Option 2: reconfigure the NVDS

The NVDS of the host is attached to the default VLAN and overlay transport zone, “nsx-vlan-transportzone” and “nsx-overlay-transportzone” respectively. There cannot be two hostswitches attaching to the same transport zones on a host. As a result, we cannot configure the VDS to attach to those transport zones while the NVDS is also attached to them. A simple workaround is to create a temporary transport zone to which we attach the NVDS. In the following screenshot, we are editing the NVDS configuration for the host so that it attaches to a newly created “Temporary TZ” transport zone. This temporary VLAN transport zone is just used during this step of the process and can be deleted once all the hosts have been migrated to VDS.

Note also that we have removed the “Network Mappings for install” that were previously configured: we do not want to move vmkernel interfaces back to this NVDS!

NSX Installation

- 1 Host Details
- 2 Configure NSX

Configure NSX

? ×

+ ADD SWITCH

v
nvds

Type* N-VDS VDS

Mode* Standard ⓘ ENS Interrupt ⓘ Enhanced Datapath ⓘ

Name* ⓘ

Transport Zone* x v

OR Create New Transport Zone

NIOC Profile x v

OR Create New NIOC Profile

Uplink Profile* v

OR Create New Uplink Profile

LLDP Profile x v

Teaming Policy Uplink Mapping

Uplinks	Physical NICs
<input type="text" value="uplink-1"/>	vmnic0
<input type="text" value="uplink-2"/>	

PNIC only Migration No

Network Mappings for Install Add Mapping

Network Mappings for Uninstall Add Mapping

CANCEL
PREVIOUS
FINISH

Thanks to this step, we can proceed with the installation of NSX on the VDS hostswitch as if we had deleted the NVDS hostswitch, except this time, the NSX binaries will not be removed and the configuration of the VDS for NSX will be much faster.

Install NSX on the VDS hostswitch

The configuration of NSX on a VDS is very similar to the creation of an NVDS hostswitch. The following screenshot is providing an example of this configuration:

NSX Installation

- 1 Host Details
- 2 Configure NSX

Configure NSX

+ ADD SWITCH

nvds

Type • N-VDS VDS

Mode • Standard ⓘ
 ENS Interrupt ⓘ
 Enhanced Datapath ⓘ

Name • newVDS

Transport Zone • nsx-overlay-transportzone x
nsx-vlan-transportzone x

OR Create New Transport Zone

Uplink Profile • nsx-default-uplink-hostswitch-profile

OR Create New Uplink Profile

IP Assignment (TEP) • Use IP Pool

IP Pool • TEP-Pool

Teaming Policy Uplink Mapping

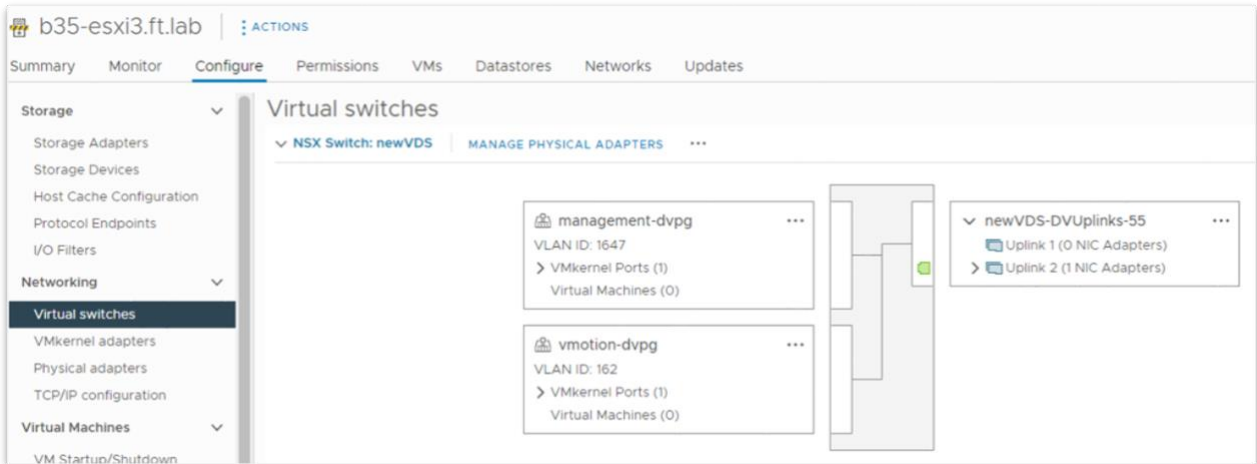
Uplinks	VDS Uplinks
uplink-1	Uplink 1
uplink-2	Uplink 2

CANCEL PREVIOUS FINISH

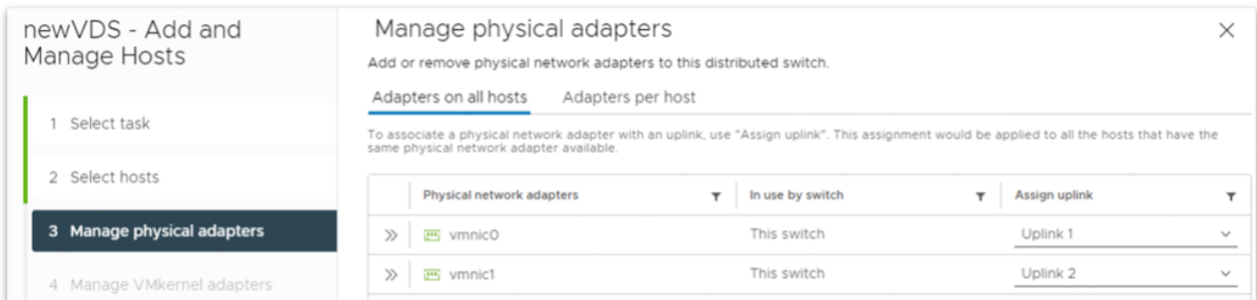
There is a significant difference in the way the uplinks are assigned on a VDS vs. NVDS. The above screenshot shows that “uplink-1” is mapped to “Uplink 1” instead of vmnic0. This configuration follows

the model described in the first part of the paper, where installing NSX on a VDS is represented as instantiating an internal NSX hostswitch inside the VDS hostswitch. Here, we are mapping the uplinks of this NSX hostswitch (“uplink-1”) to the uplink of the VDS hostswitch (“Uplink 1”).

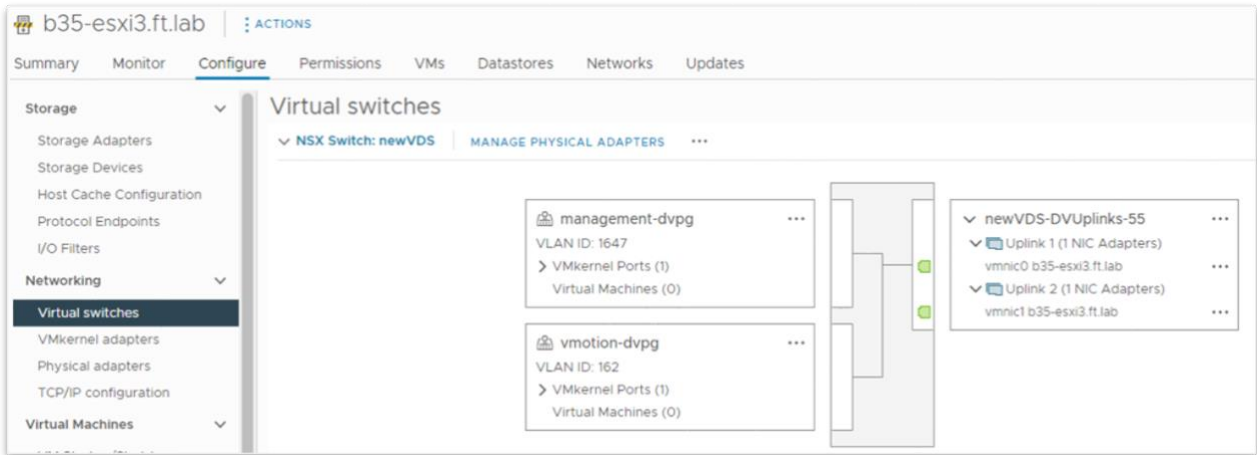
Applying this configuration on the host will prepare the VDS for NSX. The following screenshot shows “newVDS” as an “NSX Switch”:



Another interesting fact in the above screenshot is that the VDS still only has a single physical uplink (vmnic1) mapped to it. Because vmnic0 is now free (the NVDS has disappeared because it was replaced with the VDS), we can assign vmnic0 to the VDS and restore full network redundancy to our host:



The screenshot below shows the final state for the host, where the “newVDS” replaces the NVDS.

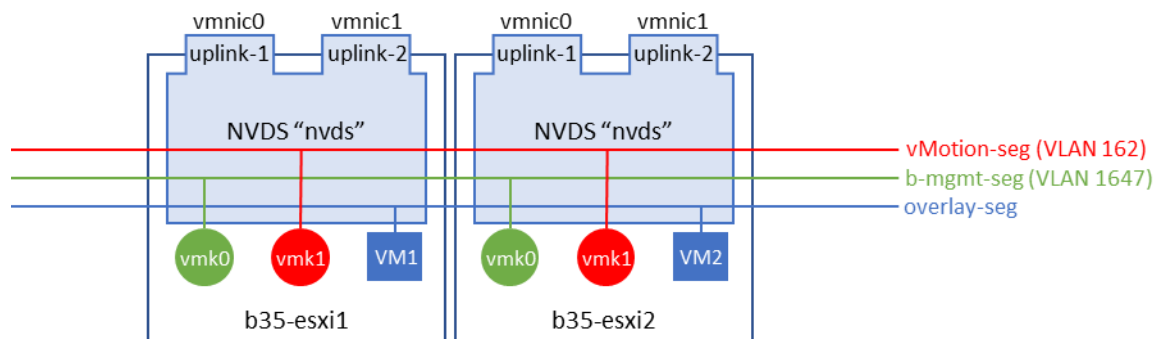


This host can exit maintenance mode and be fully operational. The migration procedure needs to be applied to the other hosts, until all NVDS instances have been removed. When the cluster has been converted to VDS, it is possible to (optionally) create a TNP matching their common configuration and apply it to the cluster.

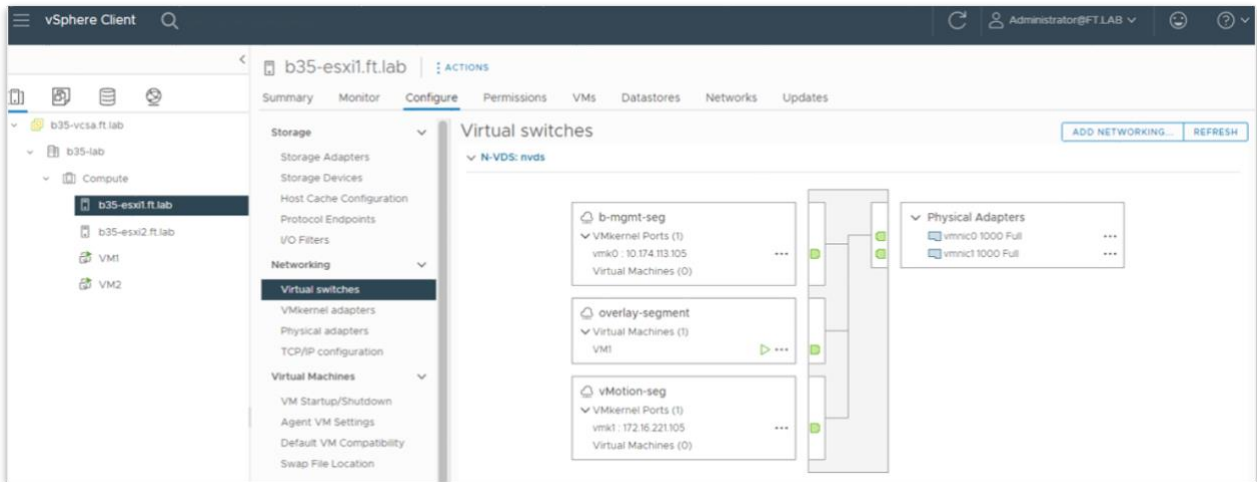
Converting NVDS to VDS using the NSX manager CLI

Initial lab configuration:

- two ESXi hosts part of a cluster called “Compute” with DRS enabled
- a TNP (named “NVDS TNP”) is applied to the cluster, ensuring a consistent NSX configuration across hosts
- an NVDS hostswitch (named “nvds”) with two uplinks (uplink-1/uplink-2 mapped to vmnic0/vmnic1) is provision on each host
- three NSX segments are configured in the NSX manager and available on the hosts:
 - b-mgmt-seg, VLAN segment backed by VLAN 1647, to which management vmk0 is attached
 - vMotion-seg, VLAN segment backed by VLAN 162, to which vMotion vmk1 is attached
 - overlay-seg, an overlay segment to which VM1/VM2 are attached



Following is a snapshot of the virtual switch configuration for host b35-esxi1, as it is shown in vCenter:



In the NSX UI, we can see that the two transport nodes are successfully deployed:

Host Transport Nodes		Edge Transport Nodes	Edge Clusters	Container Clusters				
Managed by vCenter								
CONFIGURE NSX REMOVE NSX ACTIONS								
<input type="checkbox"/>	Node	ID	IP Addresses	OS Type	NSX Configuration	NSX Version	Host Swit	Tunnels
<input checked="" type="checkbox"/>	Compute (2) Networking & Security	MoRe...			Applied Profile: NVDS TNP			
<input type="checkbox"/>	b35-esxi1.ft.lab	10b4...	10.174.11...	ESXi 7.0.3	Success	3.2.1.0.0.1...	1	↑1
<input type="checkbox"/>	b35-esxi2.ft.lab	2c63...	10.174.11...	ESXi 7.0.3	Success	3.2.1.0.0.1...	1	↑1

The following screenshot is representing the TNP applied to the Compute cluster. This shows most of the NSX relevant configuration that is applied to the NVDS.

Edit Transport Node Profile - NVDS TNP



Name* NVDS TNP

Description

+ ADD SWITCH

▼ nvds

Type* N-VDS VDS

Mode* Standard ⓘ
 ENS Interrupt ⓘ
 Enhanced Datapath ⓘ

Name* ⓘ nvds

Transport Zone* nsx-overlay-transportzone x nsx-vlan-transportzone x ▼

[OR Create New Transport Zone](#)

NIOC Profile x ▼

[OR Create New NIOC Profile](#)

Uplink Profile* nsx-default-uplink-hostswitch-profile ▼

[OR Create New Uplink Profile](#)

LLDP Profile LLDP [Send Packet Disabled] x ▼

IP Assignment (TEP)* Use IP Pool ▼

IP Pool* TEP-Pool ▼

Teaming Policy Uplink Mapping

Uplinks	Physical NICs
🔌 uplink-1	vmnic0
🔌 uplink-2	vmnic1

PNIC only Migration No

Network Mappings for Install [2 Mappings](#)

Network Mappings for Uninstall [Add Mapping](#)

CANCEL

SAVE

Procedure overview

When replacing the NVDS hostswitch with a VDS using the NSX migration tool from the NSX manager CLI, the first step is, of course, to log onto the NSX manager console via ssh. From there, the administrator will first

- run a pre-check: the migration tool will determine whether it can perform the migration and will return its suggested final topology.
- Apply the topology: this creates a VDS according to the topology computed in the previous step.
- Migrate the hosts. This is the most labor intensive phase, where the NVDS hostswitches are going to be swapped with VDS hostswitches following the configuration of the VDS created in the previous step.

Run the pre-check

The CLI based migration start with a pre-check. The pre-check analyzes the current NSX configuration and determines whether it can build a VDS-based topology to which the current deployment can be migrated. Note that this command is safe at any time. It does not perform any change to the running configuration.

```
b35-nsxmgr> vds-migrate precheck  
Precheck Id: 0583e314-d0c2-4b50-80c8-e2031325ff7b
```

After the “pre-check” command is run, the administrator can get a detail of the suggested topology:

```
b35-nsxmgr> vds-migrate show-topology  
Precheck Id: 0583e314-d0c2-4b50-80c8-e2031325ff7b
```

```
NVDS: nvds(86d09684-0e75-4819-8ae6-12441d2df186)
```

```
Compute Manager Topology:
```

```
[  
  {  
    "compute_manager_id": "3dc0a6aa-535e-49fb-bba9-9d50e38a9958",  
    "dvswitch": [  
      {  
        "data_center_id": "datacenter-3",  
        "vds_name": "CVDS-nvds-datacenter-3-0",  
        "vmknic": [  
          "vmk0",  
          "vmk1"  
        ],  
        "transport_node_id": [  
          "10b4765f-332b-4f58-b2f7-e68022b6822d",  
          "2c639a3d-eabc-4f94-b018-b8984142b86d"  
        ],  
        "id": "2a43476c-fd42-4118-b148-0d3c292470d1",  
        "_protection": "NOT_PROTECTED"  
      }  
    ]  
  }  
]
```

]

Here, we can see that the migration tool is suggesting the creation of a VDS named “CVDS-nvds-datacenter-3-0”. The output acknowledges that there are two vmkernel interfaces to take care of and that the VDS would apply to two transport nodes.

This example is very straightforward, but your real-life scenario can be much more complex, and you might not be satisfied with the (potentially several) VDS showing up in the migration tool topology.

There is not much you can do to edit the suggested configuration from the CLI. You could potentially change the name of the VDS, but the complexity of the procedure is not worth the benefit: you can easily edit the VDS name in the vCenter UI after the migration is complete.

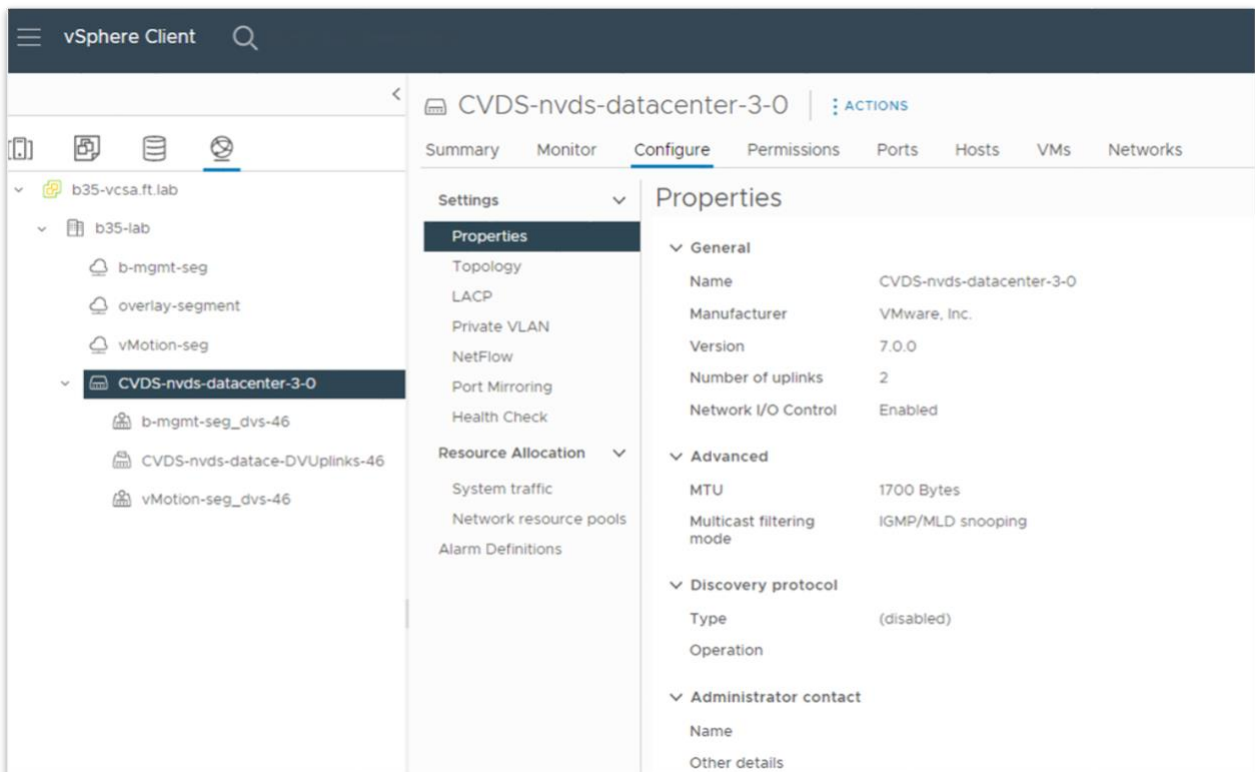
If you are not satisfied with the proposed topology, the right procedure is to modify your existing NSX environment and re-run the pre-check. You can do this as many times as you like.

Apply the topology

The next step in the migration consists of applying the topology, using the “apply-topology” command:

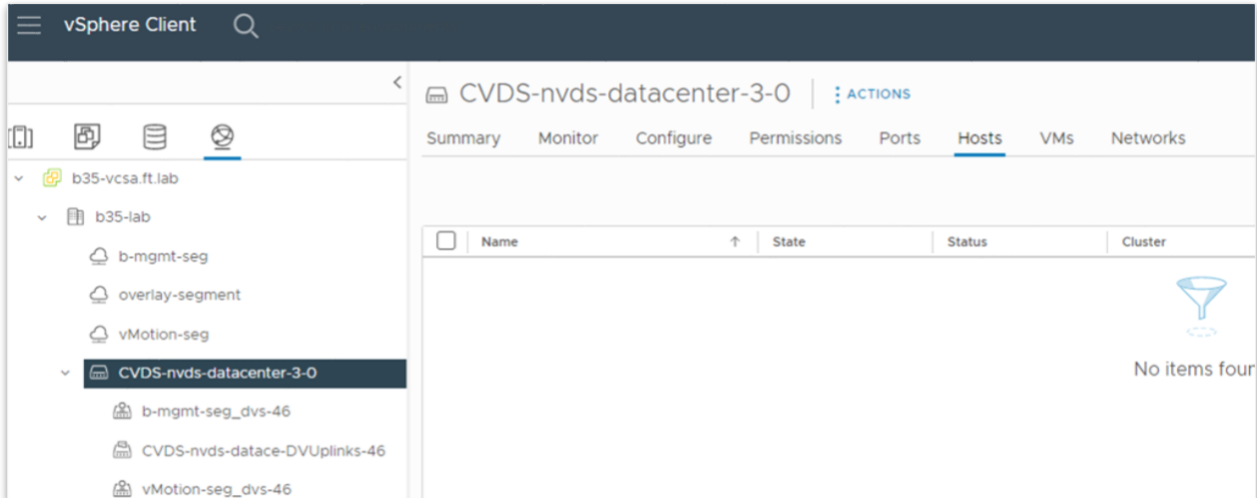
```
b35-nsxmgr> vds-migrate apply-topology  
Apply topology successfully.
```

This command creates the VDS described in the topology. This does not start the migration of the hosts yet. Here are screenshots of the result in vCenter UI:



The VDS has been automatically created using the parameters that were set in the original NVDS that it is bound to replace. The “Discovery protocol” configuration, or the MTU are examples of those in the above snapshot.

The migration tool has also automatically provisioned some dvportgroups matching the NSX VLAN segments to which the vmkernel interfaces were attached.



Here, “b-mgmt-seg_dvs-46” and “vMotion-seg_dvs-46” are dvportgroups backed by the same VLANs as the original NSX VLAN segments “b-mgmt-seg” and “vMotion-seg”. Those dvportgroups will provide connectivity to the vmkernel interfaces when the NVDS (and its opaque networks) is removed.

Migrate the ESXi hosts

The final step is to migrate the NVDS hostswitch to VDS hostswitch on the ESXi hosts. The procedure can be started with the following command, specifying the name of the cluster:

```
b35-nsxmgr> vds-migrate esxi-cluster-name Compute
```

```
NVDS Migrate Done:
```

```
 2 Transport-Nodes Migrate Successfully, ID list ['2c639a3d-eabc-4f94-b018-b8984142b86d', '10b4765f-332b-4f58-b2f7-e68022b6822d']
```

```
 0 Transport-Nodes Migrate Failed, ID list {}
```

```
 0 Transport-Nodes Migrate Skipped, ID list {}
```

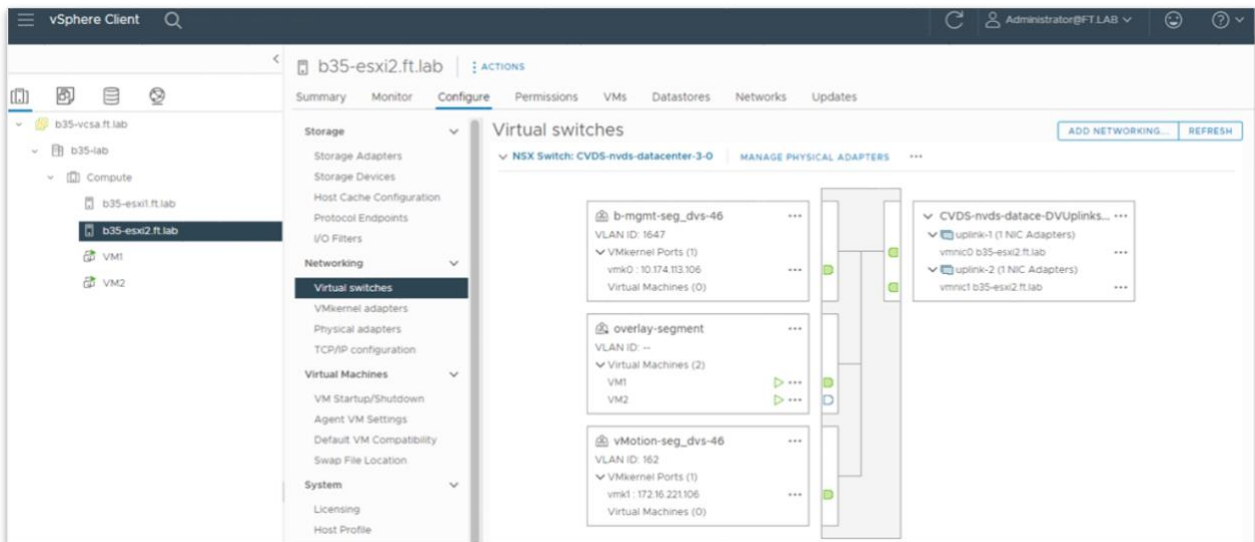
```
More info see /var/log/nsx-cli/nsxcli.log
```

```
b35-nsxmgr>
```

This step is really where the migration tool saves time. You it automates all the migration activities on the hosts for you. The following screenshot show the kind of tasks that vCenter reports. This list is not exhaustive, but you can see hosts entering/exiting the maintenance mode, virtual machine being migrated etc.

Task Name	Target	Status	Details
Exit maintenance mode	b35-esxi1.ft.lab	Completed	
Reconfigure vSphere Dis...	CVDS-nvds-datacen...	Completed	
Migrate virtual machine	VM2	Completed	Migrating Virtual Machine active state
Migrate virtual machine	VM1	Completed	Migrating Virtual Machine active state
Enter maintenance mode	b35-esxi1.ft.lab	Completed	Changing Virtual Machine host and resource pool association
Exit maintenance mode	b35-esxi2.ft.lab	Completed	
Reconfigure vSphere Dis...	CVDS-nvds-datacen...	Completed	
Migrate virtual machine	VM2	Completed	Migrating Virtual Machine active state
Enter maintenance mode	b35-esxi2.ft.lab	Completed	Migrating Virtual Machine active state
Remove vSphere Distrib...	vds	Completed	
Add Distributed Port Gro...	CVDS-nvds-datacen...	Completed	
Add Distributed Port Gro...	CVDS-nvds-datacen...	Completed	
Update network I/O cont...	CVDS-nvds-datacen...	Completed	
Reconfigure vSphere Dis...	CVDS-nvds-datacen...	Completed	
Create a vSphere Distrib...	b35-lab	Completed	

The following screenshot show the final virtual switching configuration on one of the ESXi hosts. You can see that the VDS is presented as a “NSX Switch”, which means that it is a VDS prepared for NSX. The vmkernel interfaces are attached to the appropriate dvportgroups. Now that the migration is over, you can also change the VDS name easily.



Finally, the following screenshot shows the content of the new VDS-based TNP that has been created by the migration tool and applied to the cluster:

Edit Transport Node Profile - MigratedTNP-Nv... ? ×

Name * MigratedTNP-Nvds2Cvds-Migration-NVDS TNP

Description MigrateToCvdsTask autogenerated TransportNodeProfile

+ ADD SWITCH

CVDS-nvds-datacenter-3-0

Type * N-VDS VDS

Mode * Standard ⓘ
 ENS Interrupt ⓘ
 Enhanced Datapath ⓘ

Name * vCenter CVDS-nvds-datacenter-3-0

Transport Zone * nsx-overlay-transportzone nsx-vlan-transportzone

OR Create New Transport Zone

Uplink Profile * nsx-default-uplink-hostswitch-profile

OR Create New Uplink Profile

IP Assignment (TEP) * Use IP Pool

IP Pool * TEP-Pool

Teaming Policy Uplink Mapping

Uplinks	VDS Uplinks
🔌 uplink-1	uplink-1
🔌 uplink-2	uplink-2

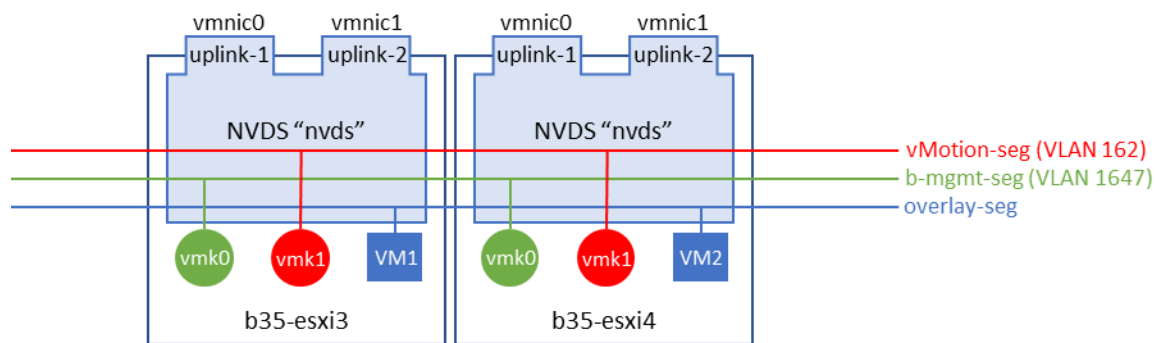
CANCEL

SAVE

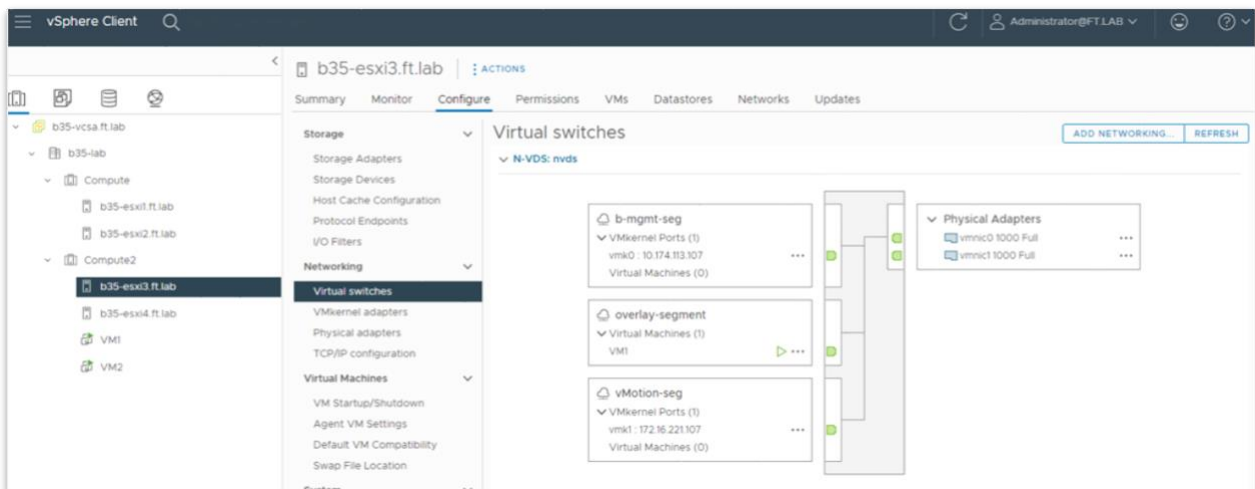
Converting NVDS to VDS using API

Initial lab configuration:

- two ESXi hosts (b35-esxi3 and b35-esxi4) part of a cluster called “Compute2” with DRS enabled
- a TNP (named “NVDS TNP”) is applied to the cluster, ensuring a consistent NSX configuration across hosts
- an NVDS hostswitch (named “nvds”) with two uplinks (uplink-1/uplink-2 mapped to vmnic0/vmnic1) is provision on each host
- three NSX segments are configured in the NSX manager and available on the hosts:
 - b-mgmt-seg, VLAN segment backed by VLAN 1647, to which management vmk0 is attached
 - vMotion-seg, VLAN segment backed by VLAN 162, to which vMotion vmk1 is attached
 - overlay-seg, an overlay segment to which VM1/VM2 are attached



Following is a snapshot of the virtual switch configuration for host b35-esxi3, as it is shown in vCenter:



In the NSX UI, we can see that the two transport nodes are successfully deployed:

Host Transport Nodes								Edge Transport Nodes	Edge Clusters	Container Clusters	
Managed by								vCenter		▼	
<input type="checkbox"/> CONFIGURE NSX <input type="checkbox"/> REMOVE NSX <input type="checkbox"/> ACTIONS ▼											
<input type="checkbox"/>	Node	ID	IP Addresses	OS Type	NSX Configuration	NSX Version	Host Swit				
<input type="checkbox"/>	▲ Compute2 (2) Networking & Security	MoRe...			Applied Profile: NVDS TNP						
<input type="checkbox"/>	b35-esxi3.ft.lab	048e...	10.174.113...	ESXi 7.0.3	● Success	3.2.1.0.0.19...	1				
<input type="checkbox"/>	b35-esxi4.ft.lab	5b56...	10.174.113...	ESXi 7.0.3	● Success	3.2.1.0.0.19...	1				

The following screenshot is representing the TNP applied to the Compute cluster. This shows most of the NSX relevant configuration that is applied to the NVDS.

Edit Transport Node Profile - NVDS TNP



Name* NVDS TNP

Description

+ ADD SWITCH

▼ nvds

Type* N-VDS VDS

Mode* Standard ⓘ
 ENS Interrupt ⓘ
 Enhanced Datapath ⓘ

Name* ⓘ nvds

Transport Zone* nsx-overlay-transportzone x nsx-vlan-transportzone x ▼

[OR Create New Transport Zone](#)

NIOC Profile x ▼

[OR Create New NIOC Profile](#)

Uplink Profile* nsx-default-uplink-hostswitch-profile ▼

[OR Create New Uplink Profile](#)

LLDP Profile LLDP [Send Packet Disabled] x ▼

IP Assignment (TEP)* Use IP Pool ▼

IP Pool* TEP-Pool ▼

Teaming Policy Uplink Mapping

Uplinks	Physical NICs
🔌 uplink-1	vmnic0
🔌 uplink-2	vmnic1

PNIC only Migration No

Network Mappings for Install [2 Mappings](#)

Network Mappings for Uninstall [Add Mapping](#)

CANCEL

SAVE

Procedure overview

The API-based tool for migrating NVDS to VDS is following the same step as its CLI-based version.

- run a pre-check: the migration tool will determine whether it can perform the migration and will return its suggested final topology.
- Apply the topology: this creates a VDS according to the topology computed in the previous step.
- Migrate the hosts: an API call will transition a transport node with no active workload from NVDS hostswitch to VDS hostswitch.

There are two versions of the APIs for the migration tool, manager and policy API. Both are supported but we will show our example based on policy APIs, as they are the preferred way of configuring NSX.

The following table shows a summary of the relevant APIs:

URT Workflow:		
Name	Proton API	Policy API
CreateNvdsUpgradePrecheck	POST /nvds-urt/precheck	PUT /policy/api/v1/infra/nvds-urt/precheck
GetNvdsUpgradePrecheckId	GET /nvds-urt/precheck	GET /policy/api/v1/infra/nvds-urt/precheck
GetNvdsUpgradeReadinessCheckSummary	GET /nvds-urt/status-summary/<precheck-id>	GET /policy/api/v1/infra/nvds-urt/status-summary/<precheck-id>
GetRecommendedVdsTopology	GET /nvds-urt/topology/<precheck-id>	GET /policy/api/v1/infra/nvds-urt/topology/<precheck-id>
NvdsUpgradeCleanup	POST /nvds-urt/topology?action=cleanup	PUT /policy/api/v1/infra/nvds-urt/topology?action=cleanup
SetTargetVdsTopology	POST /nvds-urt/topology?action=apply	PUT /policy/api/v1/infra/nvds-urt/topology?action=apply
UnsetTargetVdsTopology	POST /nvds-urt/topology?action=revert	PUT /policy/api/v1/infra/nvds-urt/topology?action=revert

Migrator Apis:		
MigrateHostTransportNodeFromNvdsToVds	POST /transport-nodes/<transport-node-id>?action=migrate_to_vds	Post /infra/sites/<site-id>/enforcement-points/<enforcementpoint-id>/host-transport-nodes/<host-transport-node-id>?action=migrate_to_vds

Run the pre-check

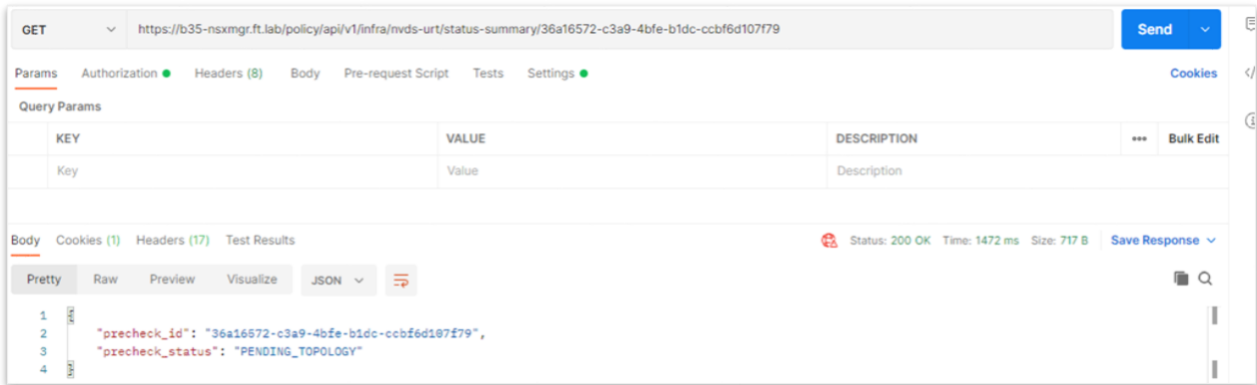
The pre-check analyzes the current NSX configuration and determines whether it can build a VDS-based topology to which the current deployment can be migrated. Note that this command is safe at any time. It does not perform any change to the running configuration.

The screenshot shows a REST client interface with the following details:

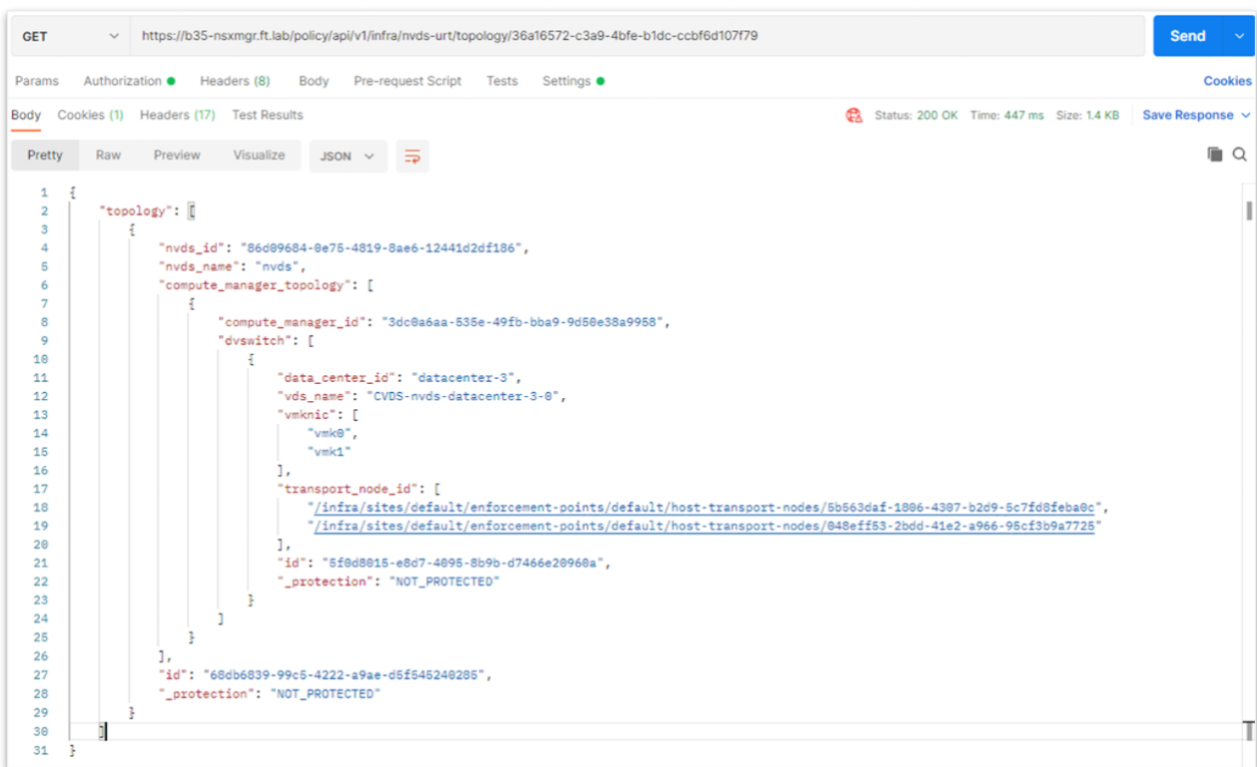
- URL: `https://b35-nsxmgr.ft.lab/policy/api/v1/infra/nvds-urt/precheck`
- Method: `PUT`
- Send button: `Send`
- Params: Authorization, Headers (9), Body, Pre-request Script, Tests, Settings
- Query Params: Empty
- Body: `200 OK 220 ms 675 B`
- Response (JSON):

```
{  "precheck_id": "36a16572-c3a9-4bfe-b1dc-ccb6d107f79"}
```

The above command returns immediately. Even if the computation of the topology is fast, an example of a second API call used to validate whether the topology is ready is shown below:



Once the topology is ready, you can retrieve it using the API call represented below:

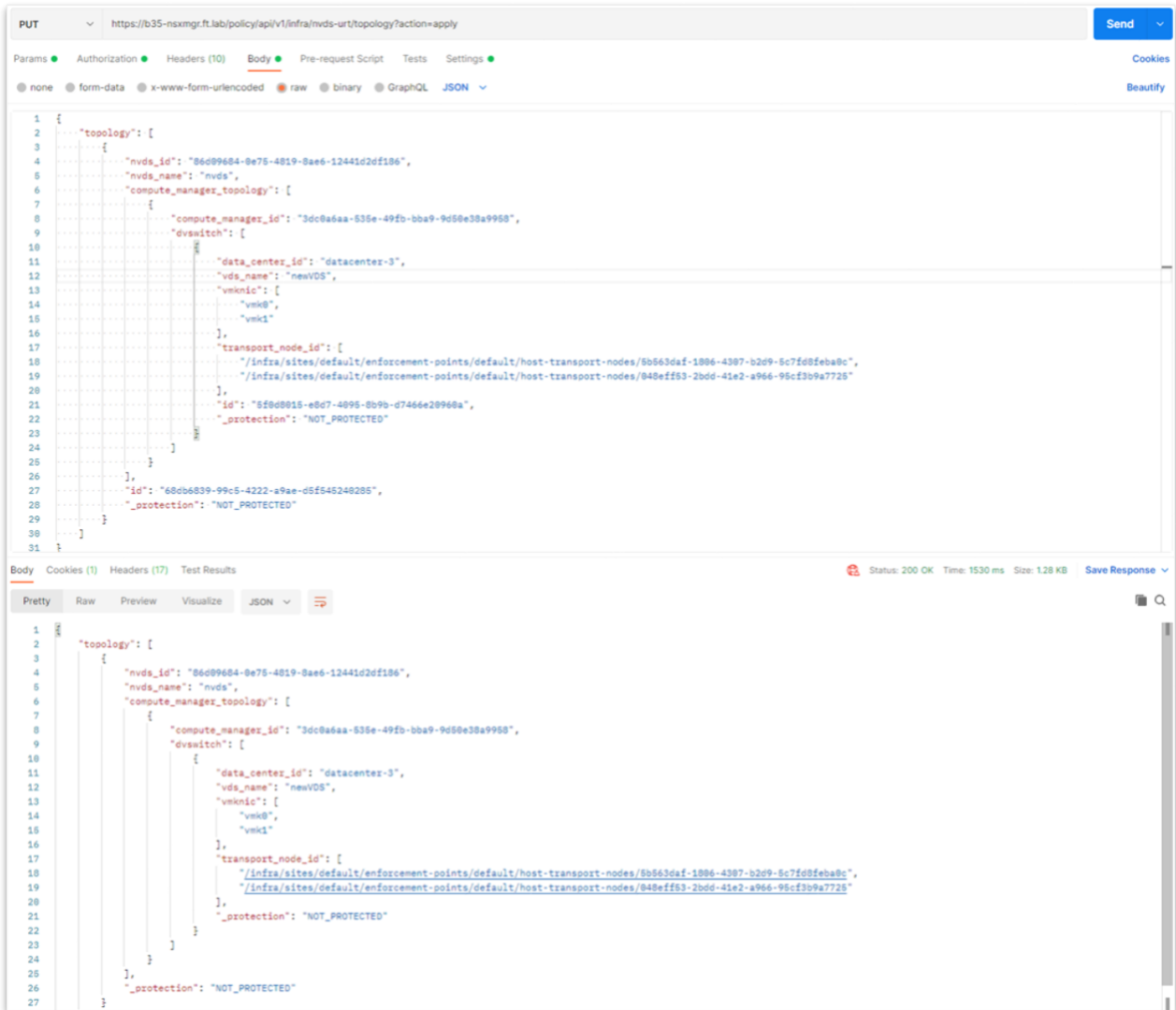


The format of the output is very similar to the one displayed with the CLI version of the tool.

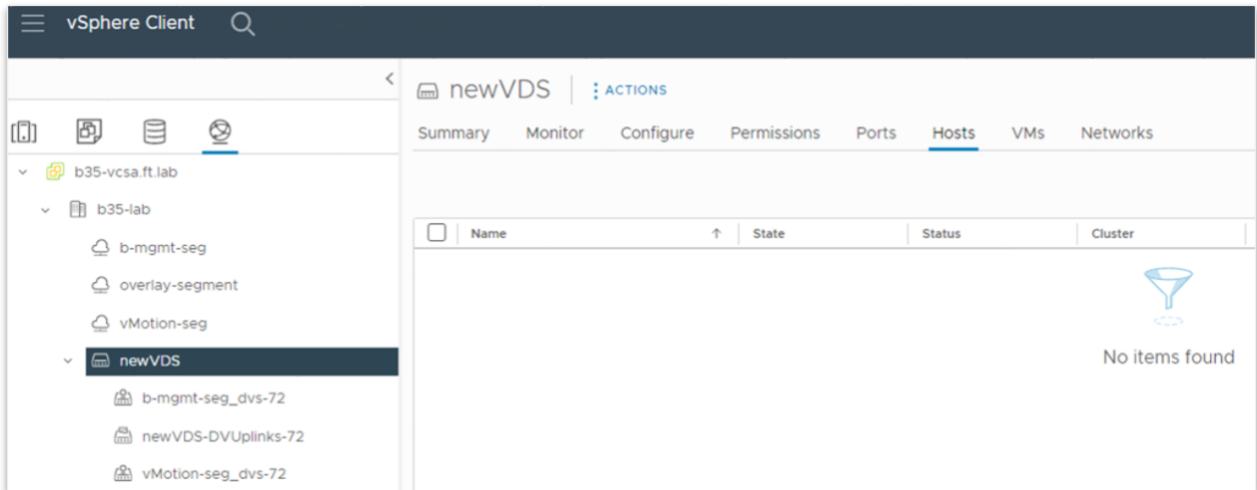
Here, the migration tool is suggesting the creation of a VDS named “CVDS-nvds-datacenter-3-0”. The output acknowledges that there are two vmkernel interfaces to take care of and that the VDS would apply to two transport nodes. The “transport_node_id”s listed here are important as we will need them to migrate the respective hosts.

Apply the topology

Applying the topology computed in the previous step is achieved by the following API call. The topology is passed in the body of the following PUT API call. Note that we took the opportunity to update the



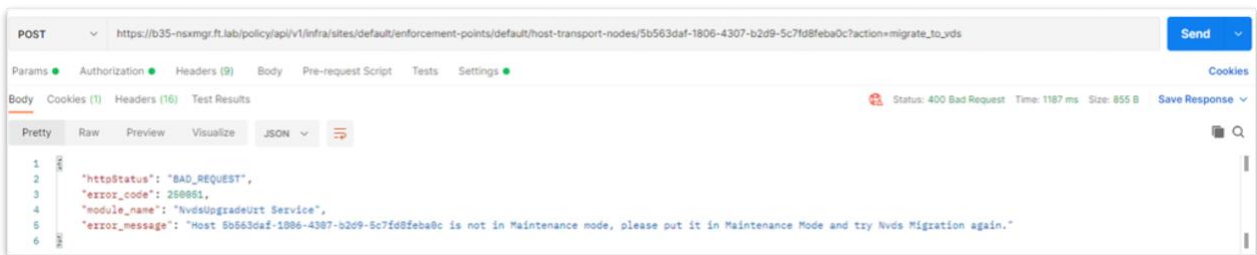
At that stage, the VDS is created in vCenter. The following vCenter UI screenshot shows the new VDS as well as the two dvportgroups that are created in preparation for the migration of the vmkernel interfaces:



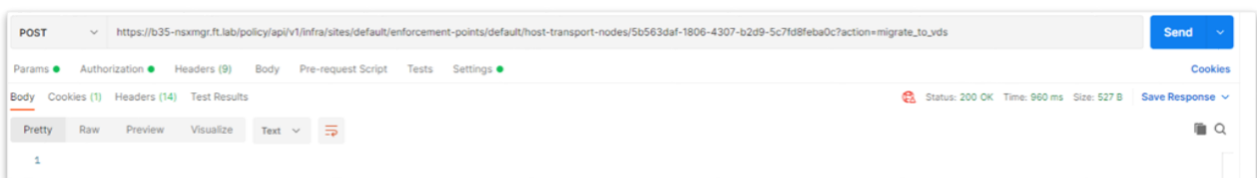
Migrating the hosts

In this phase, the hosts get their NVDS hostswitch replaced with a VDS hostswitch. This must be done by an explicit API call for each ESXi transport node.

It is the responsibility of the administrator to make sure that all running virtual machines have been evacuated and that the ESXi host is in maintenance mode. The call to migrate to VDS would fail with the following error message if not:



When the host is ready to be migrated, this API call realizes the transition to VDS hostswitch:



A call requesting the status will confirm that the migration of the host is complete, as shown below:

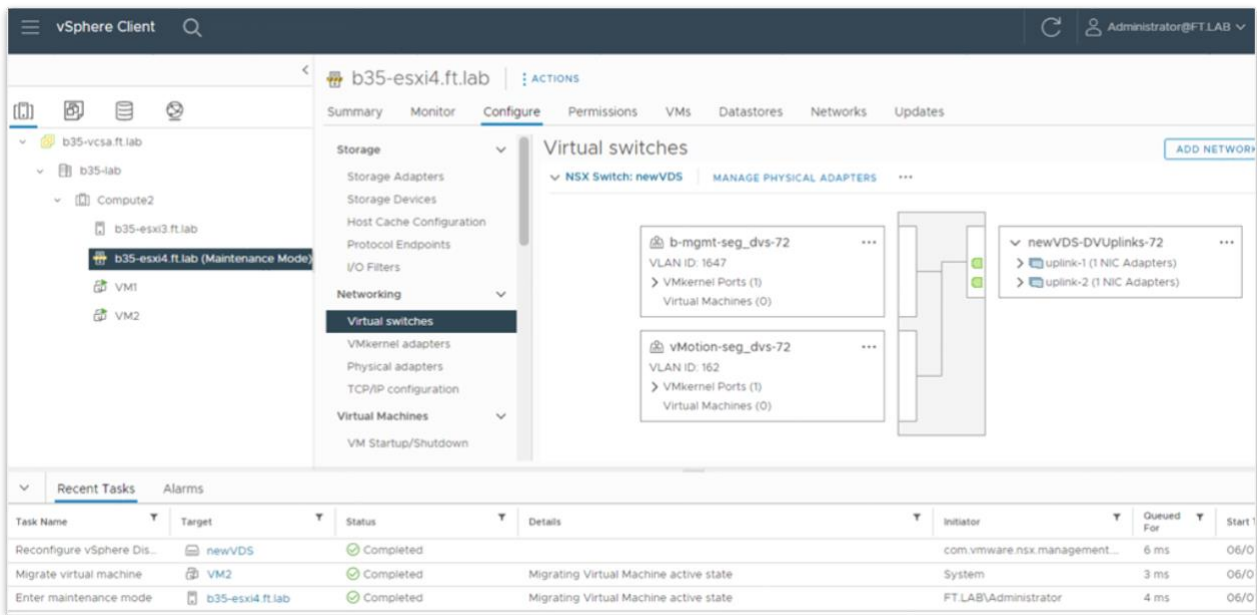
```

GET https://b35-nsxmgr.ft.lab/policy/api/v1/infra/nvds-urt/status-summary/36a16572-c3a9-4bfe-b1dc-ccb6d107f79

Params Authorization Headers (8) Body Pre-request Script Tests Settings
Body Cookies (1) Headers (17) Test Results
Pretty Raw Preview Visualize JSON
1
2 "precheck_id": "36a16572-c3a9-4bfe-b1dc-ccb6d107f79",
3 "precheck_status": "READY",
4 "migration_state": [
5   {
6     "host": "/infra/sites/default/enforcement-points/default/host-transport-nodes/5b563daf-1806-4307-b2d9-5c7fd8fba8c",
7     "overall_state": "SUCCESS",
8     "ip_address": "10.174.113.108",
9     "upgrade_stage": "TN_MIGRATION_COMPLETED",
10    "_protection": "NOT_PROTECTED"
11  },
12  {
13    "host": "/infra/sites/default/enforcement-points/default/host-transport-nodes/048eff53-2bdd-41e2-a966-95cf3b9a7725",
14    "overall_state": "UPGRADE_READY",
15    "ip_address": "10.174.113.107",
16    "_protection": "NOT_PROTECTED"
17  }
18 ]
19

```

Of course, the change is also visible in vCenter:



Repeat this step for all the VDS-based hosts in order to complete the migration.

Migrating by cluster

The API-based migration model includes the unique capability of migrating a single vSphere cluster. This feature was initially introduced for VCF and hidden, but its related APIs are now publicly available. This paragraph covers an example of API-driven migration of a single cluster. The example is based on a minimal lab with two ESXi transport nodes prepared using a transport node profile.

- Run a pre-check, specifying explicitly the cluster. The "cluster_ids" parameter is a table of cluster IDs:

POST <https://<nsx-manager>/api/v1/nvds-urt/precheck-by-clusters>

```
{
  "cluster_ids": [{"4711ec5f-1f89-49b9-a12e-b879df593138:domain-c10"}]
}
```

- You can see the status of the pre-check using the following API call:

GET: <https://nsxmgr-01a.corp.local/policy/api/v1/nvds-urt/status-summary-by-cluster/d531ffe8-0056-4ab7-a72e-8be63424140c>

Returns:

```
{
  "precheck_id": "d531ffe8-0056-4ab7-a72e-8be63424140c",
  "precheck_status": "READY",
  "migration_state": [
    {
      "host": "7152f462-31d1-4bf3-bf1e-4ef0f997e26a",
      "overall_state": "UPGRADE_READY",
      "ip_address": "192.168.110.53",
      "_protection": "NOT_PROTECTED"
    },
    {
      "host": "88e006d5-a8bc-45f6-b499-8974521cdb5a",
      "overall_state": "UPGRADE_READY",
      "ip_address": "192.168.110.54",
      "_protection": "NOT_PROTECTED"
    }
  ]
}
```

- The following API call details the resulting topology:

GET: <https://nsxmgr-01a.corp.local/policy/api/v1/nvds-urt/topology-by-cluster/d531ffe8-0056-4ab7-a72e-8be63424140c>

Returns:

```
{
  "topology": [
    {
      "nvds_id": "d4a51c72-056e-438c-a6be-30e8a101e89d",
      "nvds_name": "nsxHostSwitch",
      "compute_manager_topology": [
        {
          "compute_manager_id": "4711ec5f-1f89-49b9-a12e-b879df593138",
          "dvswitch": [
            {
              "data_center_id": "datacenter-3",
              "cluster_id": "4711ec5f-1f89-49b9-a12e-b879df593138:domain-c10",

```

```

        "vds_name": "CVDS-nsxHostSwitch-datacenter-3-Cluster-02a-0",
        "vmknic": [],
        "transport_node_id": [
            "7152f462-31d1-4bf3-bf1e-4ef0f997e26a",
            "88e006d5-a8bc-45f6-b499-8974521cdb5a"
        ],
        "id": "69f8c326-cb39-41e8-bd97-2f6841cdbeb1",
        "_protection": "NOT_PROTECTED"
    }
]
}
],
"id": "03fa8971-55d3-4f31-abc6-4084bf0f58e1",
"_protection": "NOT_PROTECTED"
}
]
}

```

- The topology is then applied using the following API call:

POST: <https://nsxmgr-01a.corp.local/policy/api/v1/nvds-urt/topology/?action=apply>

BODY:

```

{
  "topology": [
    {
      "nvds_id": "d4a51c72-056e-438c-a6be-30e8a101e89d",
      "nvds_name": "nsxHostSwitch",
      "compute_manager_topology": [
        {
          "compute_manager_id": "4711ec5f-1f89-49b9-a12e-b879df593138",
          "dvswitch": [
            {
              "data_center_id": "datacenter-3",
              "cluster_id": "4711ec5f-1f89-49b9-a12e-b879df593138:domain-c10",
              "vds_name": "CVDS-nsxHostSwitch-datacenter-3-Cluster-02a-0",
              "vmknic": [],
              "transport_node_id": [
                "7152f462-31d1-4bf3-bf1e-4ef0f997e26a",
                "88e006d5-a8bc-45f6-b499-8974521cdb5a"
              ],
              "id": "69f8c326-cb39-41e8-bd97-2f6841cdbeb1",
              "_protection": "NOT_PROTECTED"
            }
          ]
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "id": "03fa8971-55d3-4f31-abc6-4084bf0f58e1",
  "_protection": "NOT_PROTECTED"
}
]
}

```

- Remember that the administrator is responsible for putting the hosts in maintenance mode and evacuate the VMs before migrating them. Here is the API call for migrating a host in the appropriate state:

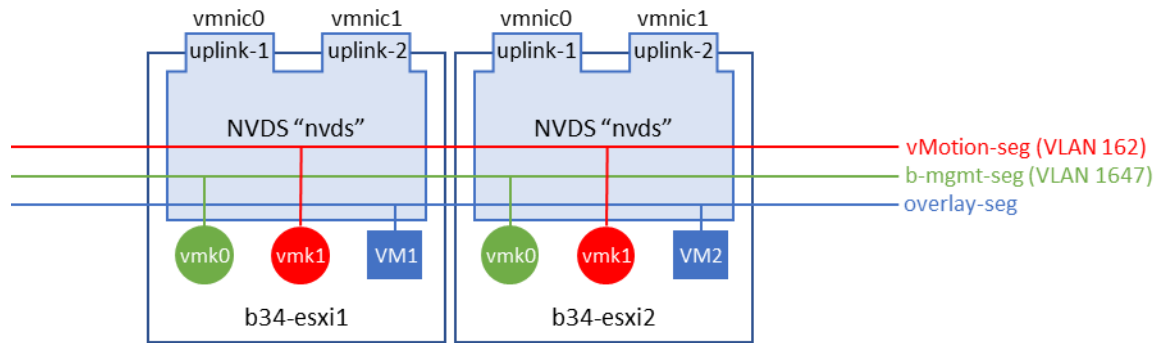
[https://nsxmgr-01a.corp.local/policy/api/v1/infra/sites/default/enforcement-points/default/host-transport-nodes/7152f462-31d1-4bf3-bfle-4ef0f997e26a/?action=migrate to vds](https://nsxmgr-01a.corp.local/policy/api/v1/infra/sites/default/enforcement-points/default/host-transport-nodes/7152f462-31d1-4bf3-bfle-4ef0f997e26a/?action=migrate%20to%20vds)

Converting NVDS to VDS while upgrading

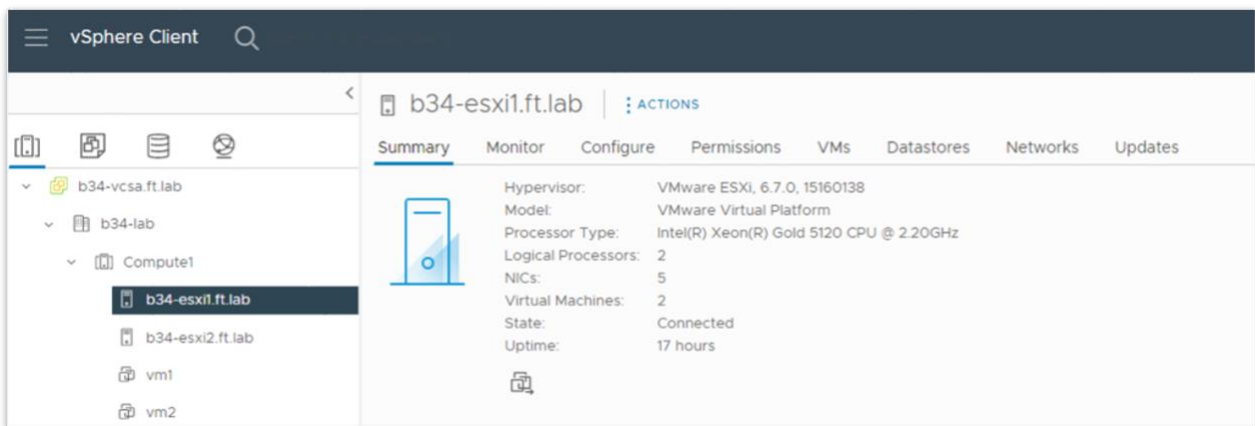
This final example documents the migration from NVDS to VDS while upgrading vSphere on the ESXi host. Both ESXi upgrade and NVDS to VDS migration require a given host to be put in maintenance mode. This migration model allows performing those two disruptive operations together, during the same maintenance window for the host.

In this environment:

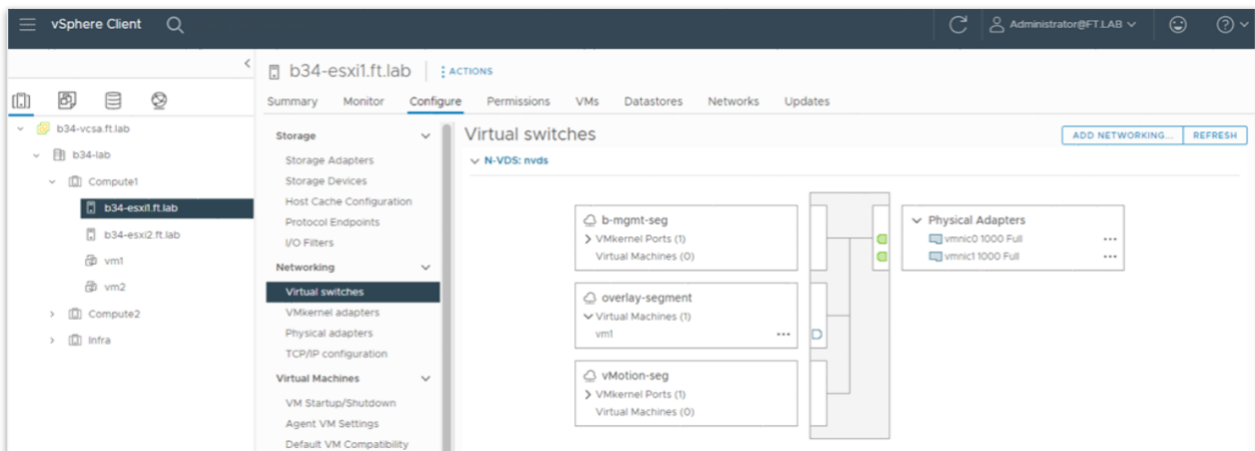
- The NSX manager is running NSX 3.2.1
- vCenter is at 7.0u3d
- two ESXi hosts (b35-esxi1, b35-esxi2) are running ESXi 6.7 and are part of a cluster called “Compute1” with DRS enabled
- a TNP (named “NVDS TNP2”) is applied to the cluster, ensuring a consistent NSX configuration across the hosts.
- an NVDS hostswitch (named “nvds”) with two uplinks (uplink-1/uplink-2 mapped to vmnic0/vmnic1) is provision on each host
- three NSX segments are configured in the NSX manager and available on the hosts:
 - b-mgmt-seg, VLAN segment backed by VLAN 1647, to which management vmk0 is attached
 - vMotion-seg, VLAN segment backed by VLAN 162, to which vMotion vmk1 is attached
 - overlay-seg, an overlay segment to which VM1/VM2 are attached
-



The following screenshot show that b34-esxi1 is initially running vSphere 6.7:



The virtual switch configuration for host b35-esxi1, as it is shown in vCenter:



Finally, a view from the NSX manager UI, showing both ESXi hosts prepared as transport nodes with “NVDS TNP2”:

Host Transport Nodes								Edge Transport Nodes	Edge Clusters	Container Clusters	
Managed by								vCenter			
<a>CONFIGURE NSX <a>REMOVE NSX <a>ACTIONS											
<input type="checkbox"/>	Node	ID	IP Address	OS Type	NSX Configuration	NSX Version	Host Swi				
<input type="checkbox"/>	▸ Infra (2)	MoR...									
<input checked="" type="checkbox"/>	▾ Compute1 (2) Networking & Security	MoR...	Applied Profile: NVDS TNP2								
<input type="checkbox"/>	b34-esxi1.ft.lab	632f...	10.174.11...	ESXi 6.7.0	Success	3.2.1.0.0.19801961	1				
<input type="checkbox"/>	b34-esxi2.ft.lab	175b...	10.174.11...	ESXi 6.7.0	Success	3.2.1.0.0.19801961	1				

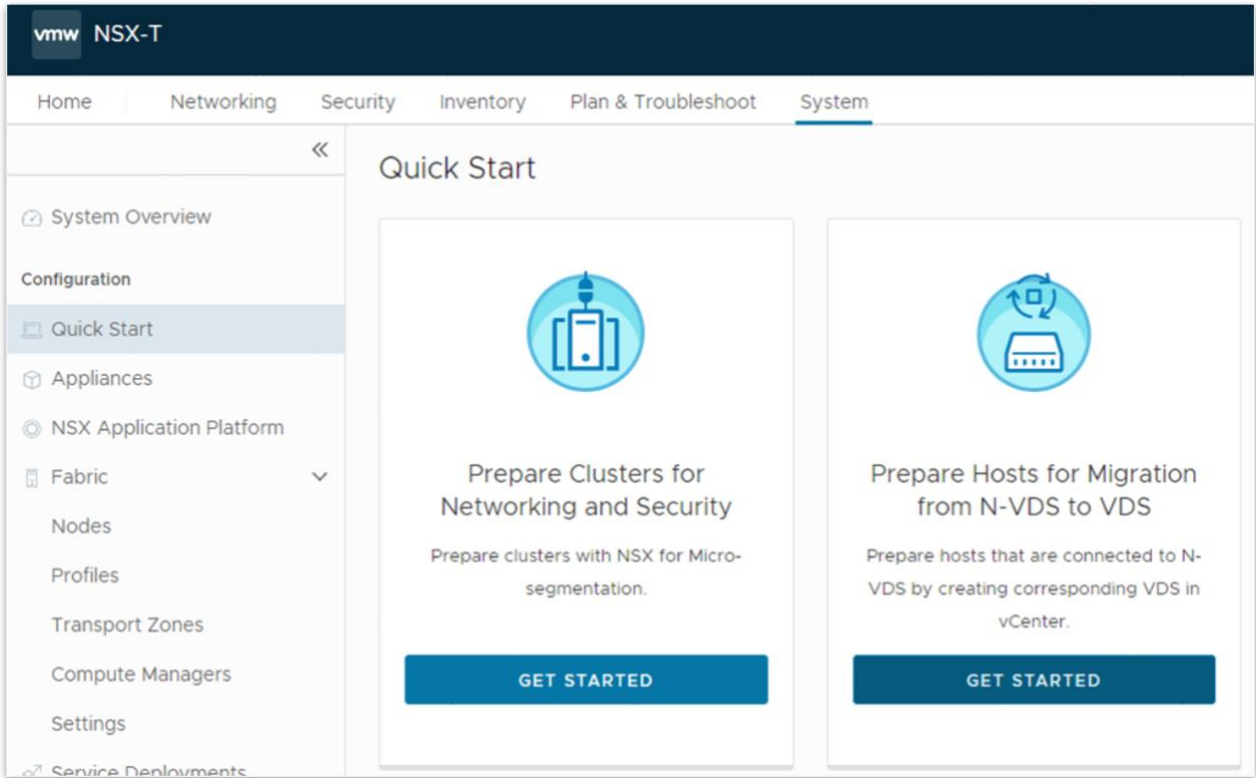
Procedure overview

The migration is very similar to the other models, except that the last step, where the ESXi hosts are individually migrated from NVDS to VDS, is part of the ESXi upgrade procedure driven by vCenter.

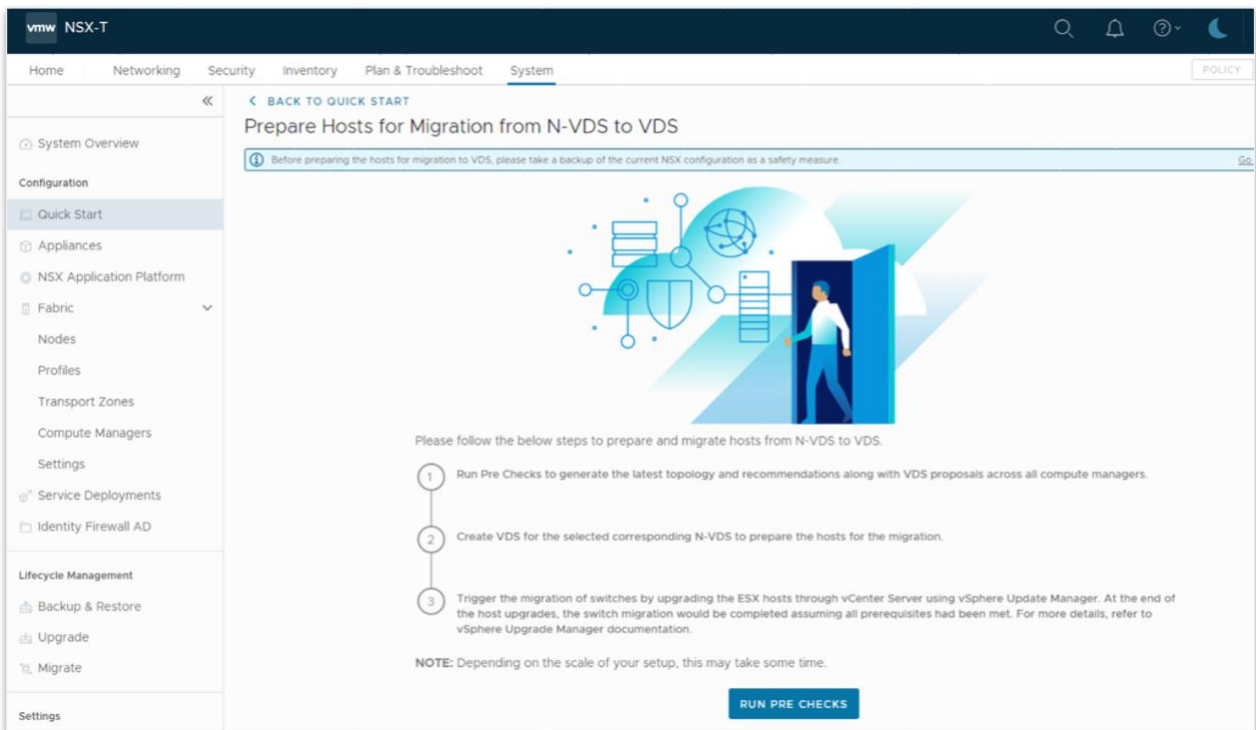
- run a pre-check: In the NSX UI, launch a pre-check where the migration tool determines whether it can perform the migration.
- Apply the topology: this step creates a VDS according to the topology computed in the previous step.
- Prepare a baseline in vSphere Upgrade Manager (VUM) that result in the upgrade of an ESXi host to 7.0u3. The baseline will include additional NSX related binaries.
- Start the “reconciliation”, i.e. the upgrade procedure, from vCenter. The ESXi hosts will be upgraded from vSphere 6.7 to 7.0u3 and their NVDS will be converted to VDS in the same step.

Run the pre-check

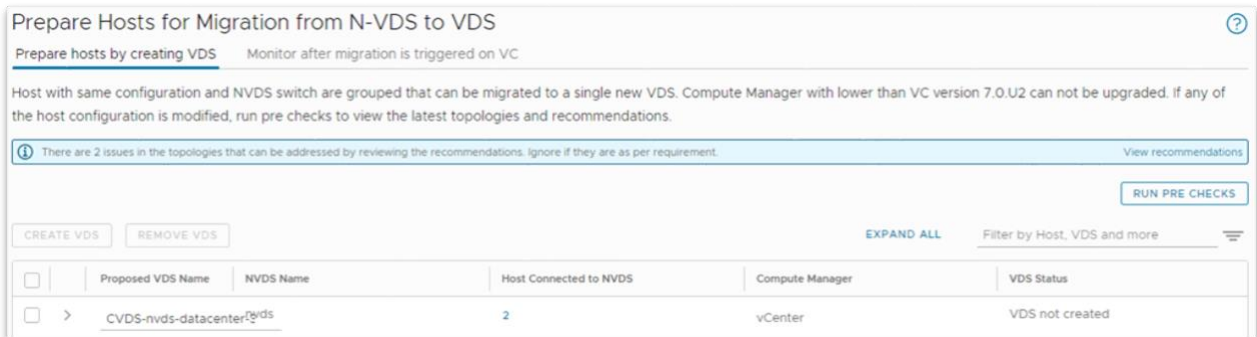
From the NSX UI, click on the “Quick Start” tab and select “GET STARTED” in the “Prepare Hosts for Migration from N-VDS to VDS” box, as represented in the screenshot below:



From there, you reach a screen providing a summary of the procedure (see screenshot below.) Click on “RUN PRE CHECKS”.



After the pre-check is run, you get a proposed VDS as well as some information concerning the hosts impacted by the migration. In the screenshot below, in the column “Host Connected to NVDS”, there is a link to a page showing the hosts.



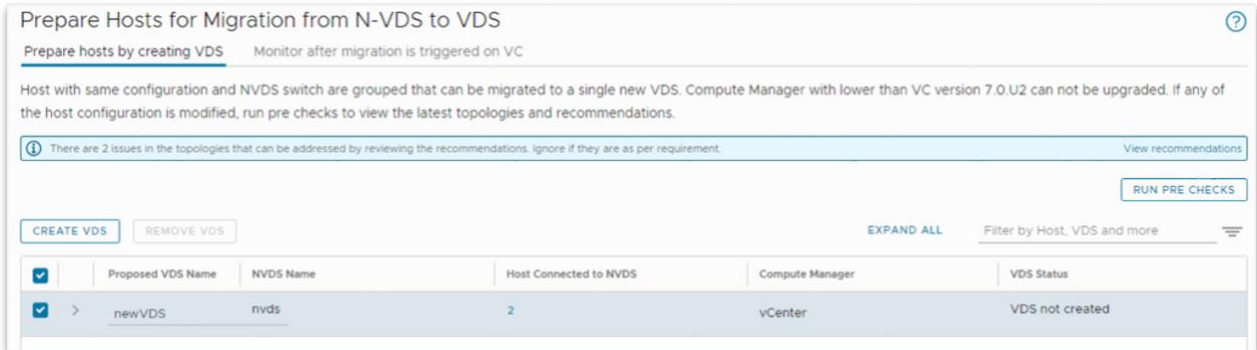
Note that the pre-check is similar to the ones we did for the other versions of the migration tool (CLI and API.) You can log into the NSX manager console and issue a “vds-migrate show-topology” to get a summary of the proposed VDS:

```
b34-nsxmgr> vds-migrate show-topology
Precheck Id: 992cc696-fa90-44e1-b988-b70e6720e2b6

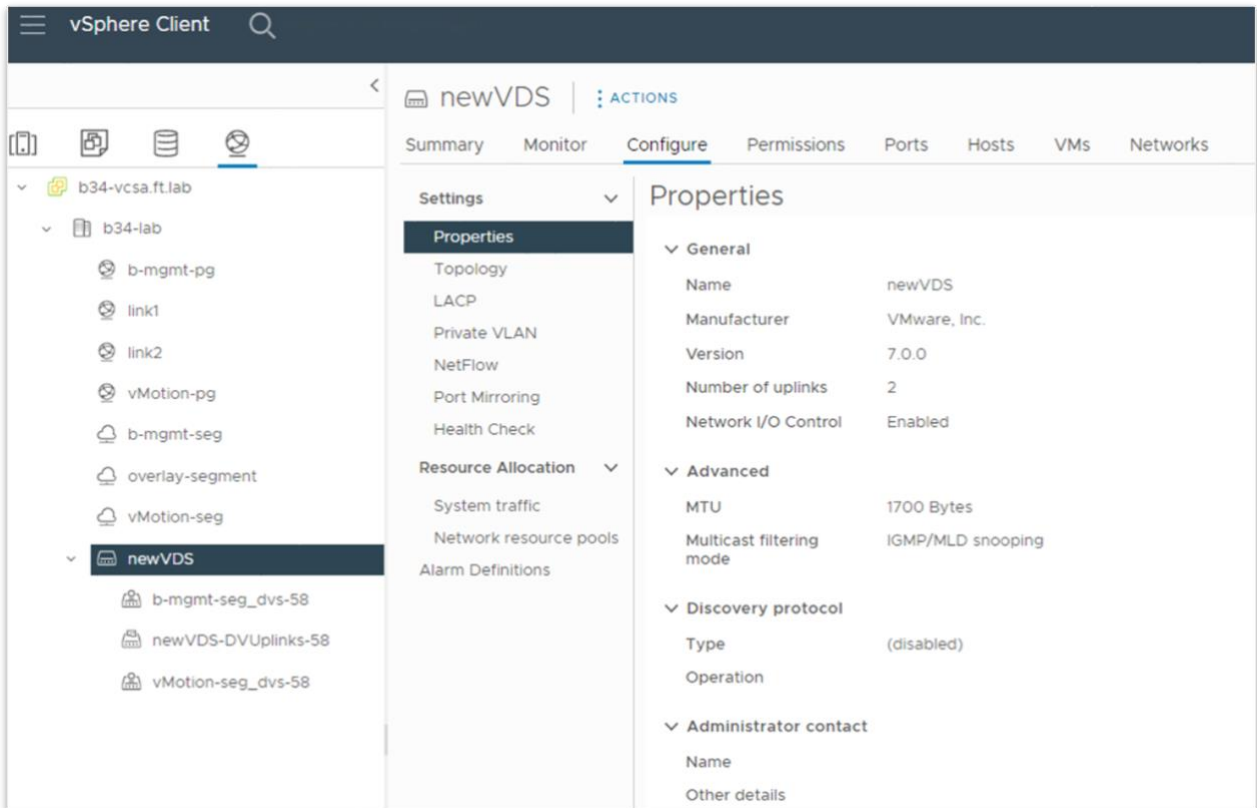
NVDS: nvds (dba83685-43f2-4a90-aca8-f41932ec4950)
Compute Manager Topology:
[
  {
    "compute_manager_id": "4ae87275-6a95-4668-8356-43ed768f56e5",
    "dvswitch": [
      {
        "data_center_id": "datacenter-3",
        "vds_name": "CVDS-nvds-datacenter-3-0",
        "vmknic": [
          "vmk0",
          "vmk1"
        ],
        "transport_node_id": [
          "175b5f0e-1a27-4290-84aa-fc51a3689b93",
          "632f6e96-2d59-410d-a218-3a818895a44b"
        ],
        "id": "adfcdb71-5954-4470-a3b1-9de9ca7893a6",
        "_protection": "NOT_PROTECTED"
      }
    ]
  }
]
```

Apply the topology

Before creating the VDS computed during pre-check, the administrator has the option to change its name. Here, we updated the previously suggested “CVDS-nvds-datacenter-3-0” name to “newVDS” directly from the NSX Manager UI. The administrator still has the option to make changes to the NSX or vCenter configuration and re-run the pre-checks.



After clicking the “CREATE VDS” button in the NSX UI (as represented in the above screenshot), the new VDS “newVDS” is created in vCenter. Below is the result in vCenter UI:



Prepare the baseline for upgrading the ESXi hosts

This part requires having access to the VMware download pages

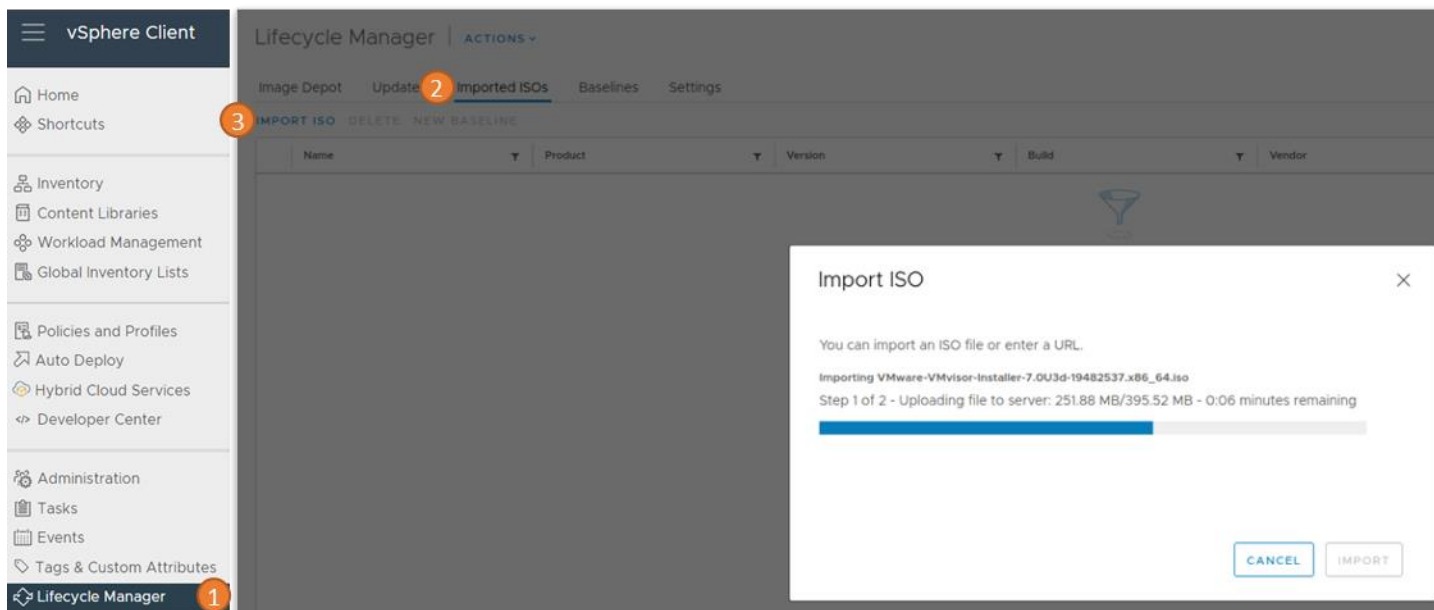
(https://customerconnect.vmware.com/downloads/#all_products)

For this specific example, we will download:

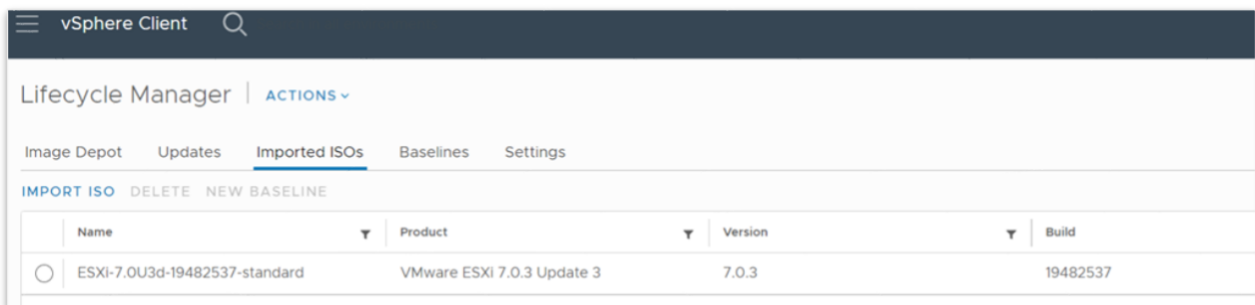
- The iso for ESXi vSphere 7.0u3d, *VMware-VMvisor-Installer-7.0U3d-19482537.x86_64.iso*
- The NSX Kernel Module for VMware ESXi 7.0, *nsx-lcp-3.2.1.0.0.19801961-esx70.zip* (in the NSX 3.2.1 download section)

Create an upgrade baseline for upgrading the ESXi host

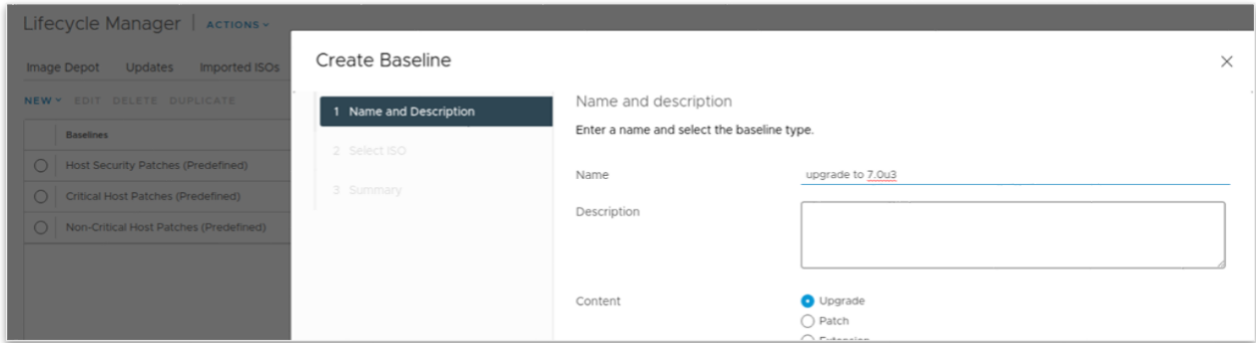
Click on “Lifecycle Manager” in the left tab of the vCenter UI, then click on “Imported ISOs”. Finally, click on “IMPORT ISO”, and upload ESXi VMvisor Installer. Check the screenshot below:



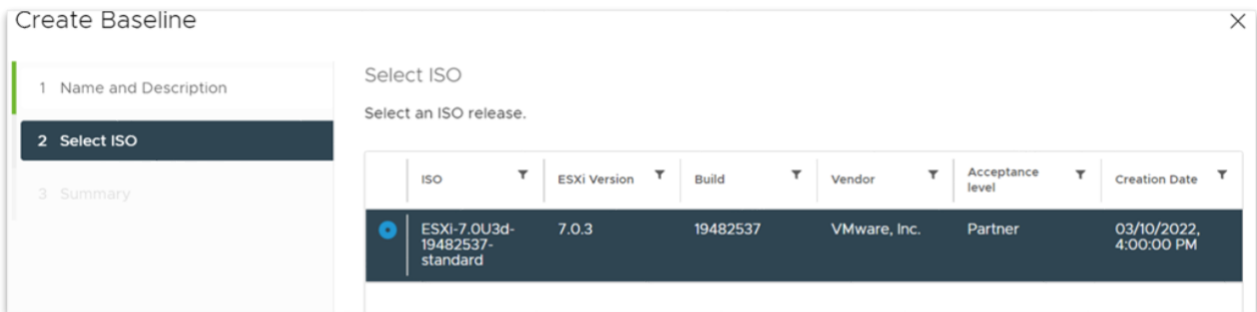
After the iso is uploaded, it is shown as available in the Lifecycle Manager:



Then, click on the top “Baselines” tab, then “NEW→Baseline”, enter a name for the new baseline (“upgrade to 7.0u3” in the screenshot below.)



Click on NEXT, select the iso file

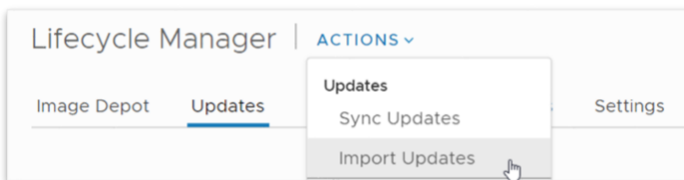


Click NEXT then FINISH.

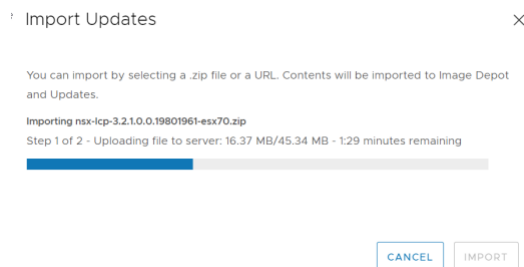
Create a vSphere 7.0 upgrade baseline for NSX 3.2

Now, we're going to create a specific baseline for including some additional NSX-specific binaries that are necessary for the upgrade to ESXi 7.0u3

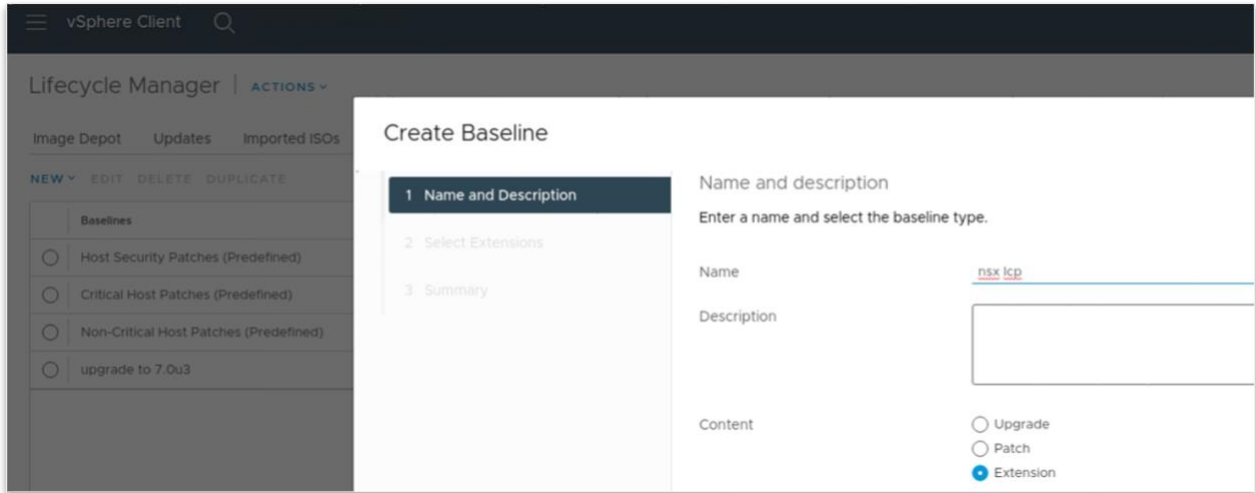
In the Lifecycle Manager, click on "Updates" then "Import Updates" as shown below:



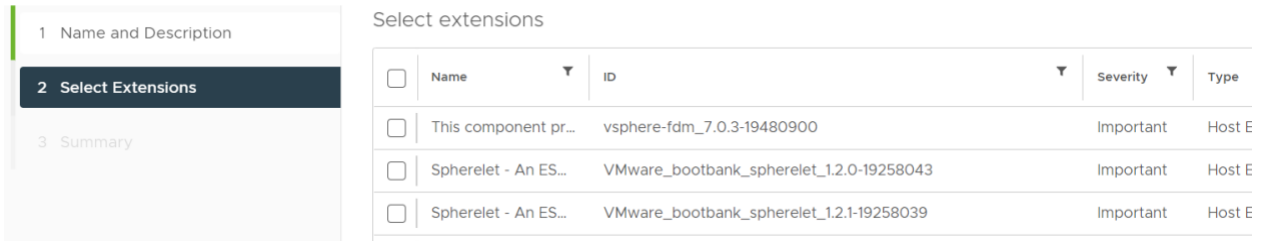
Then select for import the NSX Kernel Module for VMware ESXi 7.0, nsx-icp-3.2.1.0.0.19801961-esx70.zip, that we had downloaded earlier on.



Finally, we're going to create an "extension" baseline for this NSX Kernel Module. Click on "Baselines", the "NEW→Baseline". Enter a name then select "Extension" type:



In the next screen, a list of extension is presented:



You can filter on the "Name" column to quickly find the NSX LCP Bundle we are looking for:

Create Baseline

<input type="checkbox"/>	Name	ID
<input checked="" type="checkbox"/>	NSX LCP Bundle	vmw-nsx-lcp_3.2.1.0.0.19801961-esx70

Select this “NSX LCP Bundle” that we have uploaded in the previous step:

<input type="checkbox"/>	Name	ID	Severity	Type
<input checked="" type="checkbox"/>	NSX LCP Bundle	vmw-nsx-lcp_3.2.1.0.0.19801961-esx70	Moderate	Host Extension

Click NEXT and FINISH

Create a baseline group for the update

We are now going to combine the two baselines created in the previous steps into a single baseline group. In “Baseline”, select “NEW→Baseline Group” this time. In the following screenshot, the baseline group is called “upgrade to 7.0u3d with nsx”.

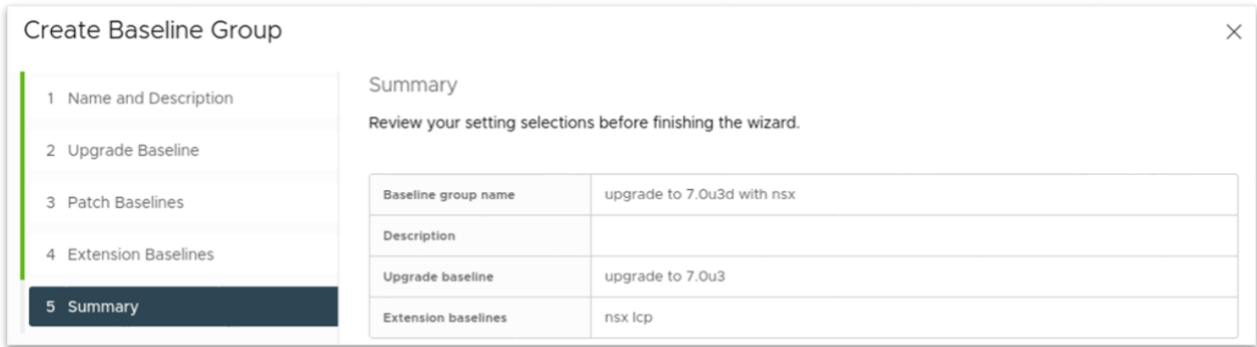
Name and description

Enter a unique name for this baseline group.

Name:

Description:

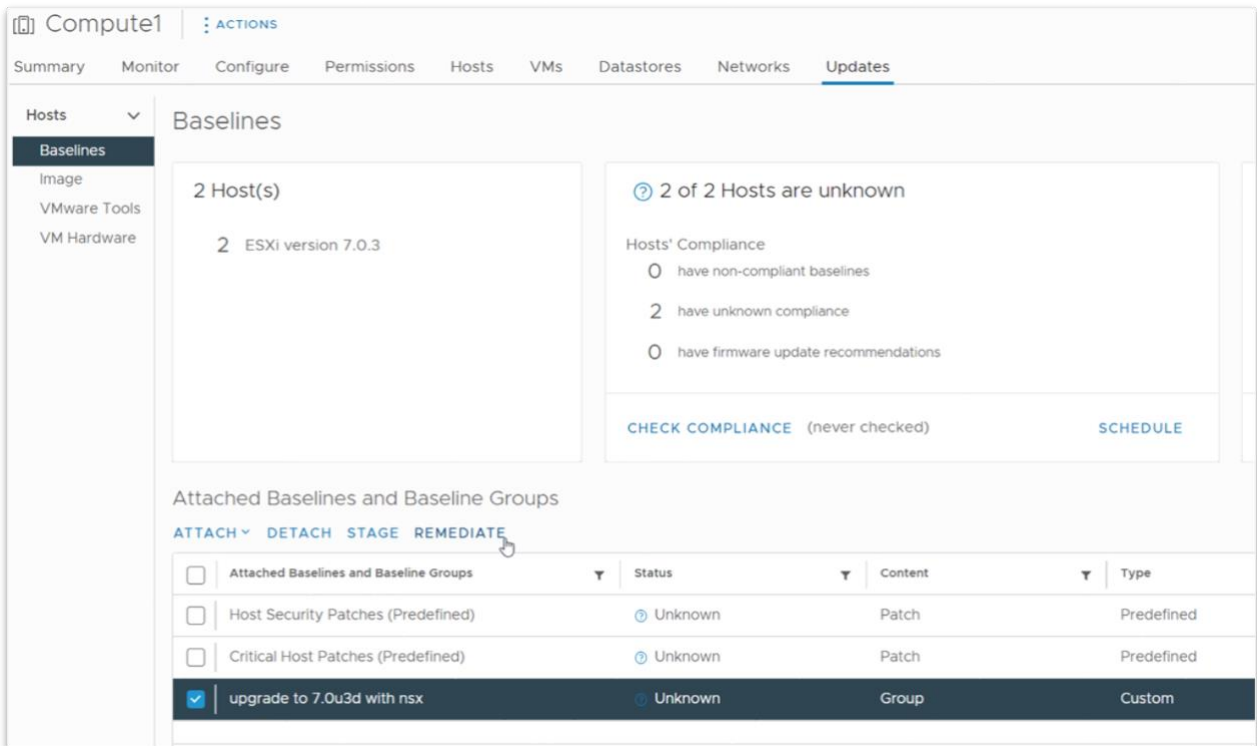
In the “Upgrade Baseline” section, select the first baseline we had created (“upgrade to 7.0u3”). Skip the “Patch Baselines” and in the “Extension Baselines”, select the extension baseline we had created (“nsx lcp”).



Click FINISH to complete the creation of the baseline group.

Upgrade and migrate the hosts

Thanks to the preparation we had made in the NSX UI, the upgrade of the ESXi hosts will also include a NVDS to VDS migration step. In order to start the upgrade, we are going to select the “Compute1” cluster, select “Updates”, then attach the baseline group that we have created in the previous step.



Finally, by click “Remediate”, the upgrade is started in vCenter:

Remediate | Compute1 with upgrade to 7.0u3d with nsx

Cluster is ready to remediate

No actions will be performed automatically

2 hosts will remediate

<input checked="" type="checkbox"/>	Host Name	Version	In Maintenance Mode	Patches	Extensions	Remediation Status
<input checked="" type="checkbox"/>	b34-esxi1.ft.lab	6.7.0	No	0 (0 Staged)	0 (0 Staged)	✓ Ready
<input checked="" type="checkbox"/>	b34-esxi2.ft.lab	6.7.0	No	0 (0 Staged)	0 (0 Staged)	✓ Ready

2 EXPORT 2 Hosts

- > Install ISO VMware ESXi Release 7.0.3, Build 19482537
- > Scheduling Options: Will remediate immediately
- > Remediation settings

For each host, the process of evacuating VMs, putting hosts in maintenance mode, upgrading, migrating to VDS and finally exiting the maintenance mode is started...

The following screenshot shows the result on b34-esxi1:

vSphere Client Administrator@FT.LAB

b34-esxi1.ft.lab

Summary Monitor **Configure** Permissions VMs Datastores Networks Updates

Storage Adapters Storage Devices Host Cache Configuration Protocol Endpoints I/O Filters

Networking

Virtual switches

VMkernel adapters Physical adapters TCP/IP configuration

Virtual Machines VM Startup/Shutdown Agent VM Settings Default VM Compatibility Swap File Location

Virtual switches

NSX Switch: newVDS

b-mgmt-seg_dvs-58
VLAN ID: 1647
VMkernel Ports (1)
Virtual Machines (0)

overlay-segment
VLAN ID: --
Virtual Machines (2)

vMotion-seg_dvs-58
VLAN ID: 162
VMkernel Ports (1)
Virtual Machines (0)

newVDS-DVUplinks-58
uplink-1 (1 NIC Adapters)
uplink-2 (1 NIC Adapters)