NSX-T 2.5 ESXi Transport Node Preparation

- This informal document describes the NSX installation/removal/recovery procedures for an ESXi host. This is not meant to be a reference guide, it is focusing on a simple case that can be easily extended to most scenarios.
- Everything described here is based on NSX-T 2.5 and will probably be outdated as soon as the next release is out because we're still actively working on simplifying this workflow.
- I welcome your feedback; will fix errors you report and add important cases that I have missed. This is not however an offer to troubleshoot your install problem as I just cannot provide help directly at the scale of VMware.

We're going to focus on a very simple scenario starting from an ESXi with two uplinks, one of them used attached to a default vSphere Standard Switch (VSS). We're going to setup an NSX Virtualized Distributed Switch (N-VDS) on this host so that:

- The N-VDS owns the two physical uplinks
- The N-VDS is configured for both overlay and VLAN traffic
- The unique management vmkernel interface vmk0 is migrated to the N-VDS

The following diagram represents the initial and final states:

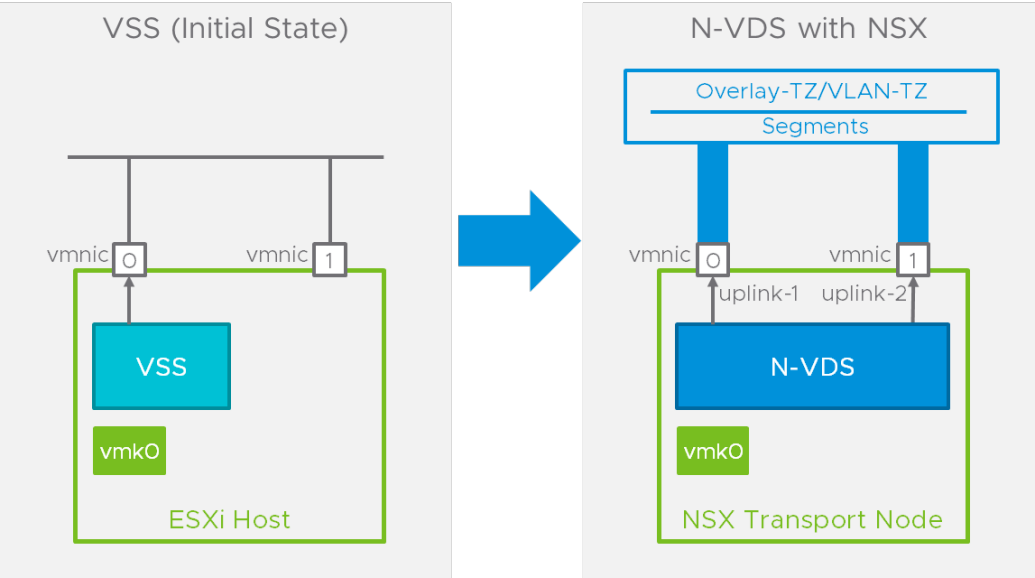


Figure 1: Transport Node preparation, VSS to N-VDS

The document also includes a section showing how other installation scenarios can be derived from this simple case. Note however that the configuration of Edge and KVM transport nodes is out of the scope of this document.

<v3, 3/2/2020>

Table of Content

(don't be intimidated by the document size, there are a lot of screenshots, and you probably only need to follow part 2 😉)

1 The theory

The part just summarizes the definitions and parameters that will be exposed to during the install, it is by no means an introduction to NSX-T. You can skip directly to section 2 if you're already familiar with NSX-T.

1.1 Transport Nodes

A transport node is a device running NSX-T's data plane. This document describes the installation of the NSX virtual switch, called the N-VDS, on an ESXi host. Because the N-VDS is a component forwarding traffic for NSX (meaning, implementing NSX data plane), we are in fact turning our ESXi host into a transport node.

1.2 Zones and Segments

Transport zones are NSX objects grouping transport nodes. When creating a transport node, one needs to attach it to one or more transport zones. NSX defines two kinds of transport zones:

- overlay transport zones and
- VLAN transport zones.

Within a transport zone, the NSX administrator can create segments. Segments are the base component of NSX connectivity, representing a virtual layer 2 broadcast domain. Segments are of two types, depending on the transport zone on which they are created. As you might guess, a segment created in a VLAN transport zone is a VLAN segment. That means that the virtual layer 2 broadcast domain is backed by a VLAN in the physical networking infrastructure. The overlay segments, defined in overlay transport zones, are virtual layer 2 broadcast domains implemented by a collection of IP tunnels between transport nodes. Those tunnels are setup and maintained automatically by NSX over the physical networking infrastructure.

When the NSX administrator creates a segment in a transport zone, it is immediately made available to all the transport nodes attached to the transport zone. As a result, the transport zone is an object representing the span of the virtual network. If I create a VLAN transport zone "VLAN-TZ", and within this transport zone I define a "Management" segment, all the transport nodes attached to VLAN-TZ see the Management segment. Via vCenter, the administrator can directly attach VM vnics or vmkernel interfaces to this Management segment on those transport nodes.

1.3 Tunnel End Points (TEPs), IP Pools

As mentioned in the above section, the NSX overlay model relies on IP tunnels between transport nodes. Each end of an IP tunnel must have an IP address and the Tunnel End Point, or TEP, is the object to which this IP address is assigned. A given transport node can have multiple TEPs in order to establish tunnels from several different physical interfaces.

The transport node we're going to configure in this example is attached to an overlay transport zone, which means that it will have at least one TEP. The TEPs are automatically created by NSX but the administrator must provide a way of assigning them an IP address. There are currently three different ways of assigning IP addresses to TEPs on a transport node.

1. You can statically define the range of IP addresses used by the transport node. This method is not very flexible because you need to configure each transport nodes individually so that their addresses don't collide.
2. You can use DHCP. When NSX creates a TEP, it will negotiate an IP address with a DHCP server available in the physical networking infrastructure.
3. You can define some IP pools from which NSX will assign the TEP addresses. We'll use this method in our example.

1.4 Uplink Profile and Teaming Policy

As suggested earlier, installing NSX on an ESXi host means creating an N-VDS. The uplink profile defines the characteristics of the N-VDS. It includes:

- The Maximum Transmission Unit (MTU) of the N-VDS. This is the size of the largest frame the N-VDS can send on the transport node physical uplink. This MTU needs to be larger than the MTU defined on the VMs' vnics (typically 1500 bytes) in order to accommodate for the overlay encapsulation overhead. The physical networking infrastructure must be able to support the N-VDS MTU. We recommend setting jumbo frames in the physical network and we'll use 9000 as the MTU of our N-VDS.
- The transport VLAN: this is the VLAN ID that will be used to tag overlay (tunnel) traffic on the uplinks.
- Link Aggregation Groups (LAGs) can be optionally defined in the uplink profile. We won't use any in this example.
- Teaming policy: this object is probably the most important object in the uplink profile. It defines both the name of the uplinks of the N-VDS as well as the way they are used for redundancy and traffic load balancing. The name of the uplinks is arbitrary, it will be matched to vmnics on the host we are preparing as a transport node.

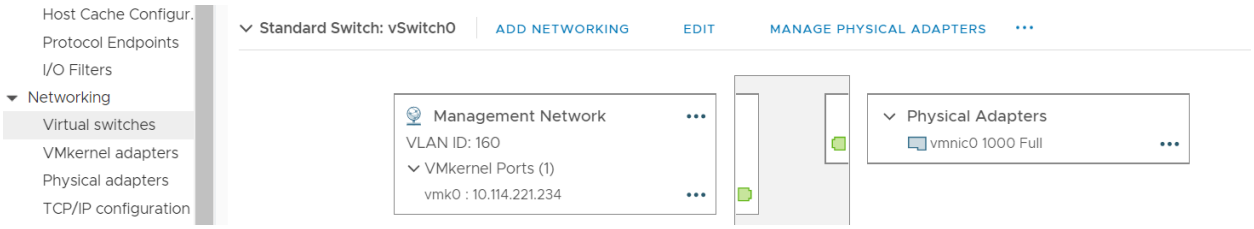
1.5 Compute Manager

Compute manager is pretty much synonymous with vCenter at that stage. The administrator can connect NSX to one or more vCenters in order to collect information from vCenter about the hosts. The information extracted from vCenter makes NSX configuration simpler.

2 Install Step-by-Step Example

My lab includes four ESXi hosts in a "compute" cluster. They have identical configuration.

Each host has a single VSS called vSwitch0 (the default name) with vmk0 attached to "Management Network". Management traffic for vmk0 is sent with VLAN ID 160 on a unique physical uplink vmnic0. The host has an additional vmnic1, which is currently unused.

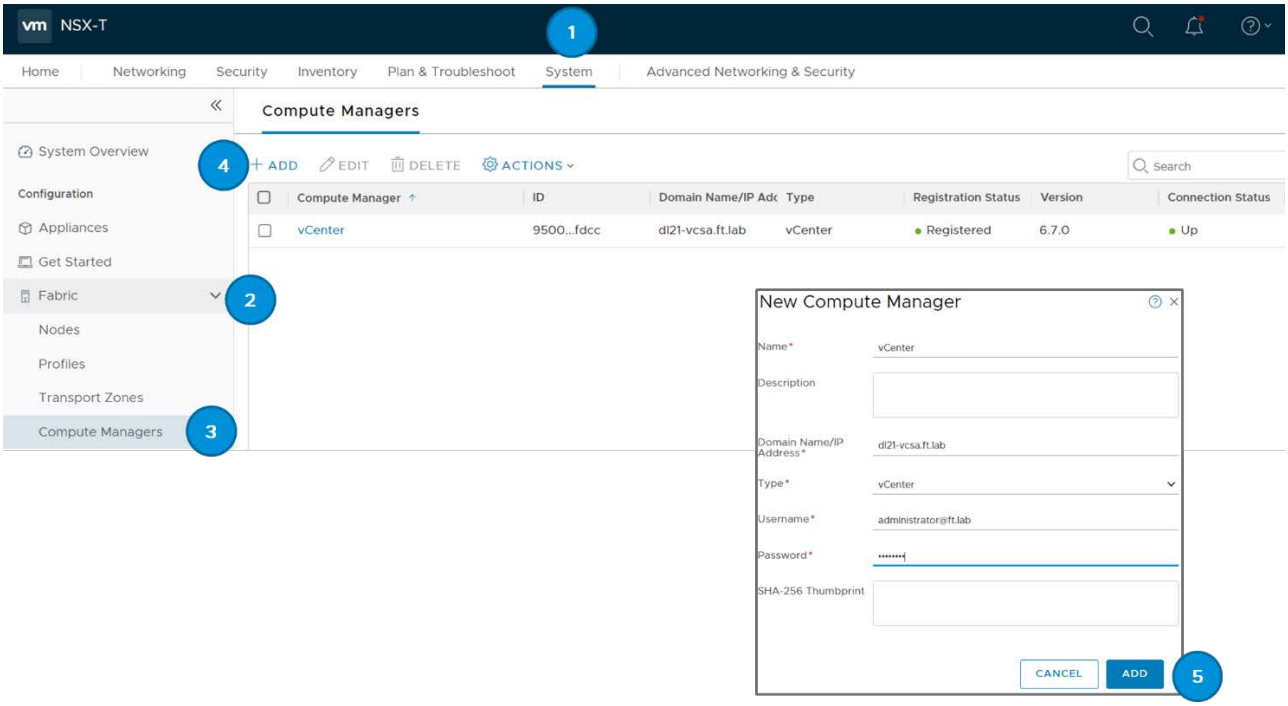


Both vmnic0 and vmnic1 are attached to physical switches. The corresponding ports are “trunks” (i.e. they accept VLAN tagged traffic) and allow at least traffic for VLAN 160, where a default gateway for the management interfaces (vmk0) of the different ESXi hosts can be reached.

2.1 Preliminary work

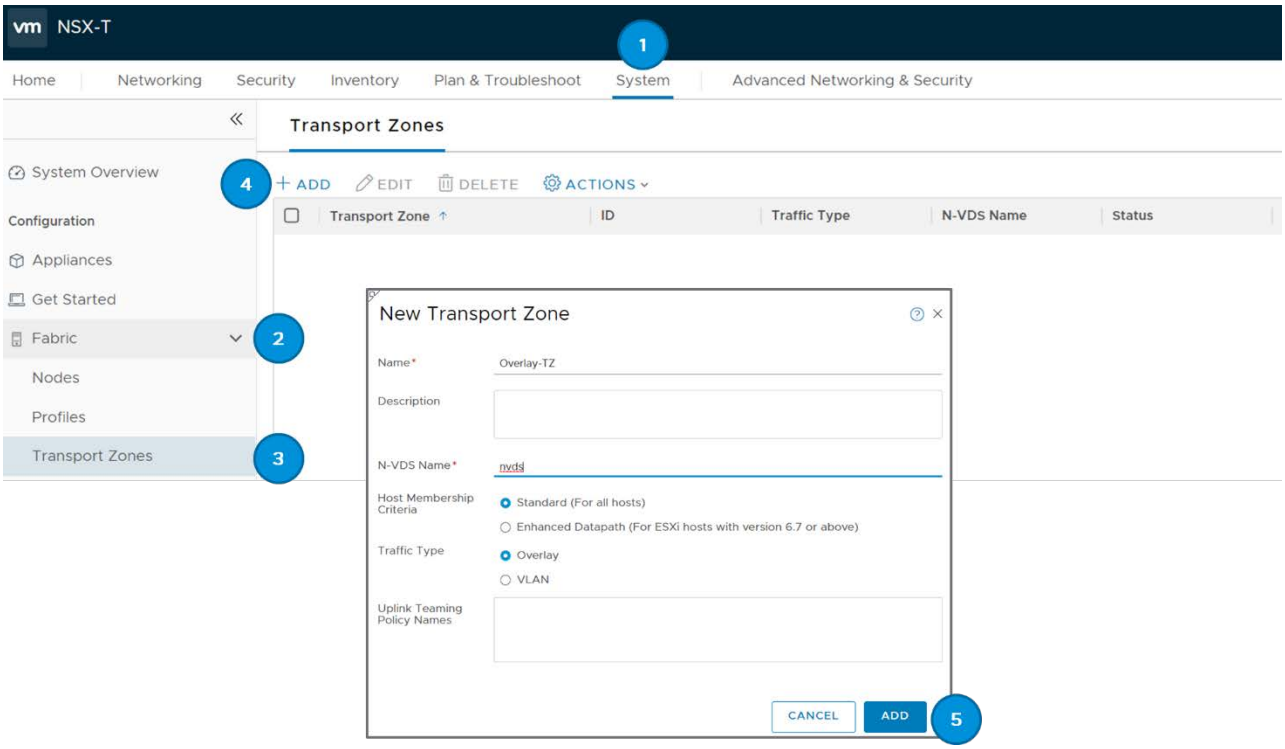
2.1.1 Connecting vCenter to NSX

In order to access its host and cluster information, we are going to register vCenter into NSX. Navigate to (1) System/(2) Fabric/(3) Compute Managers and click “+ ADD”(4). Fill in the information requested in the pop-up window and click (5)“ADD”.



2.1.2 Creating a VLAN and overlay transport zone

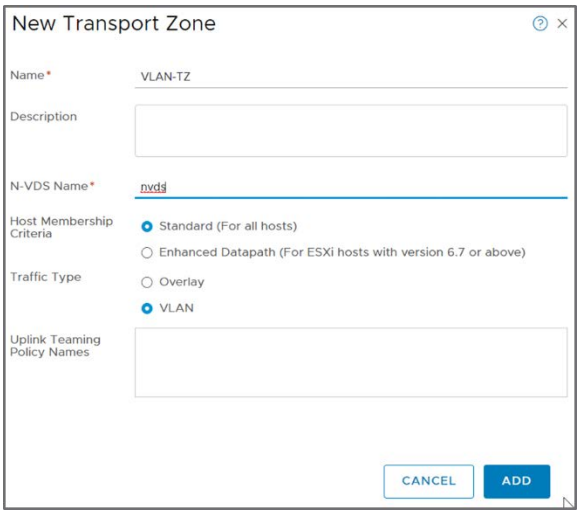
We want to attach our hosts to both VLAN and overlay segments, we’re thus going to create two transport zones, one of each kind. The following diagram show the creation of the overlay transport zone. Navigate to (1) System/(2) Fabric/(3) Transport Zones and click (4)“+ ADD”



Parameters you will see in the "New Transport Zone" dialog box:

- Name: the name of the transport zone. We'll call our first transport zone "Overlay-TZ"
- N-VDS Name: This is the name of the N-VDS that will be created on the transport node when it is attached to the transport zone. We're working on deprecating the use of this field in future releases, don't worry too much about it. Here, we're going to call our N-VDS: "nvds".
- Host Membership Criteria: Frankly, I'm not sure how we came up with the name of this field. This is where you specify whether the N-VDS created on the transport node is a "standard" N-VDS or an "Enhanced Datapath" N-VDS. We'll leave the default, "standard", here. Just for your information, the Enhanced Datapath version of the N-VDS is DPDK-based and optimized for low-latency, high packet rate forwarding. This might look tempting, but this N-VDS is optimized for very specific network-centric workloads. It requires very specific NICs on the host and does not support some hardware accelerations beneficial for generic VMs. Discussing this Enhanced Datapath N-VDS is completely out of the scope of this document and I recommend you only use the standard mode unless you know exactly what the Enhanced Datapath mode really does.
- Traffic Type: whether the transport zone is overlay or VLAN. For Overlay-TZ, we'll obviously select "Overlay".

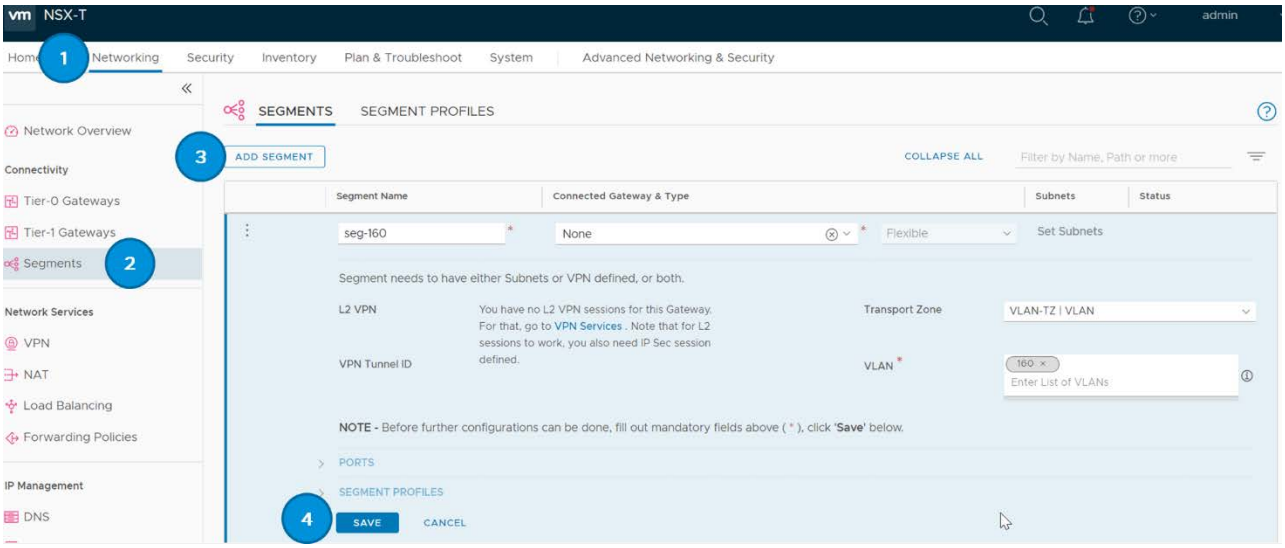
Create the second transport zone with the same parameters except for the name, "VLAN-TZ" and the traffic type "VLAN":



Note that we have used the same N-VDS name because we want both our transport zones to connect to the same N-VDS (called “nvds”).

2.1.3 Creating a VLAN segment for management

The vmk0 interface of our ESXi host is connected to VLAN 160 in the physical infrastructure. When migrating vmk0 to the N-VDS, we need to connect it to a segment backed by VLAN 160 so that it can still function properly. We’re thus going to create the appropriate segment in our VLAN transport zone “VLAN-TZ”. Navigate to (1) Networking/(2) Segments, click (3)“ADD SEGMENT”:



We’ll name our management segment “seg-160”. Make sure you select “VLAN-TZ” as a transport zone (that’s what determines we’re creating a VLAN segment) and “160” in the VLAN box. Click save when done.

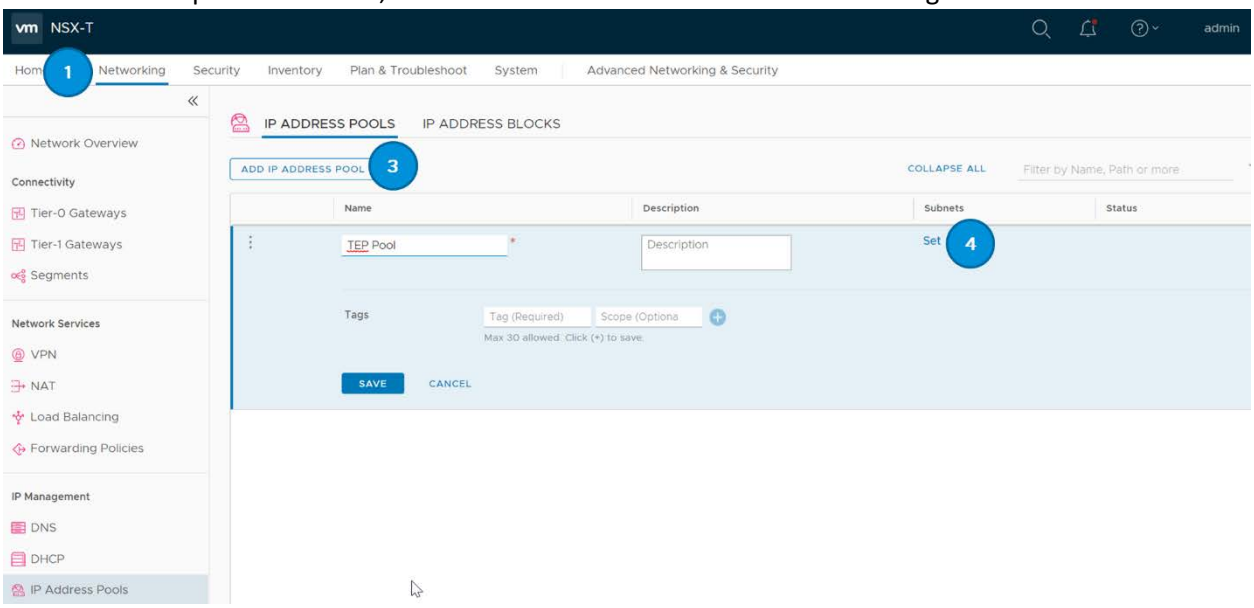
2.1.4 Creating an IP pool for the TEPs

Our transport nodes will attach to an overlay transport zone. That means that we’ll be creating tunnels between the transport nodes. In order to create tunnels, we need tunnel end points, and those TEPs

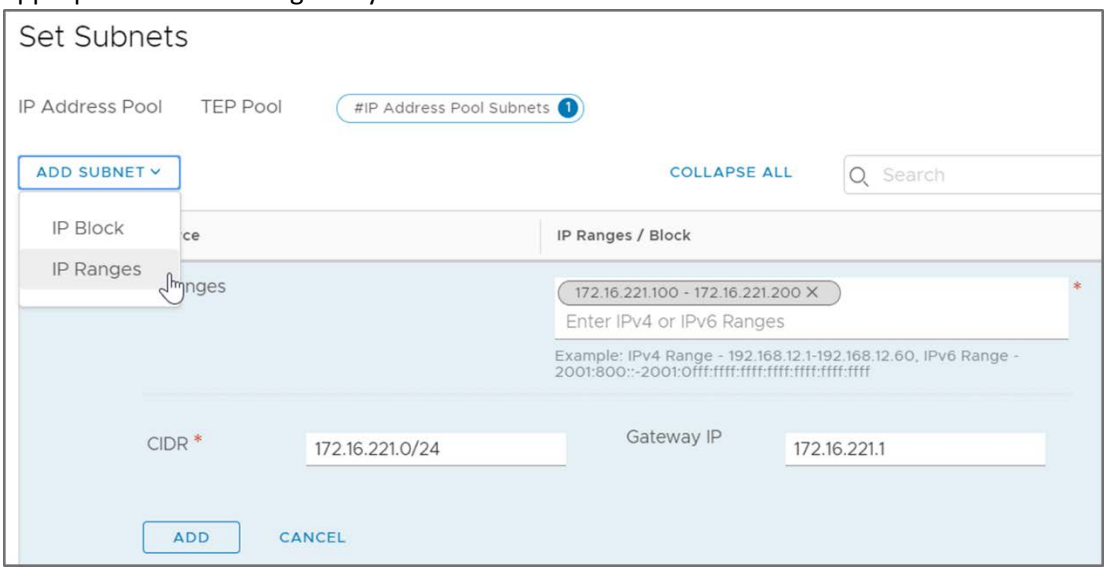
must be assigned some IP addresses routable in the physical network. In this step, we're going to create a pool of IP addresses that NSX can assign to those TEPs.

Navigate to (1) Networking/(2) IP Address Pools and click on (3)"ADD IP ADDRESS POOL".

We'll name the pool "TEP Pool", then click on "set" to enter an IP address range.



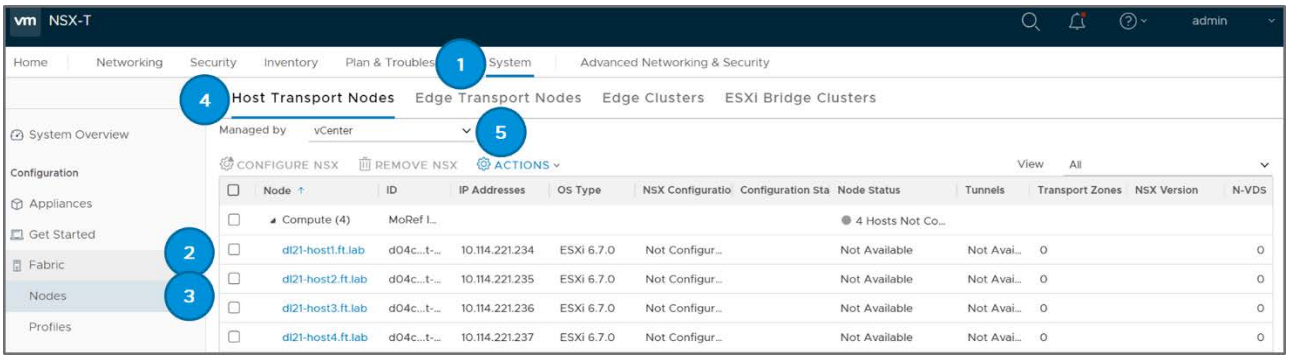
In the "Set Subnets" dialog box, represented below, click ADD SUBNET/IP Ranges and enter an appropriate address range for your lab.



Again, the range of IP addresses must be routable in the physical infrastructure. In this example, all the ESXi host are in the same cluster with L2 connectivity between their uplinks. As a result, we'll only need a single IP pool. It is common for hosts in different racks to be in different subnets. In that case, an IP pool will be needed for each rack, and the default gateways in those pools must be set appropriately so that the TEPs of the host can communicate with each other between racks.

2.2 Preparing a single transport node

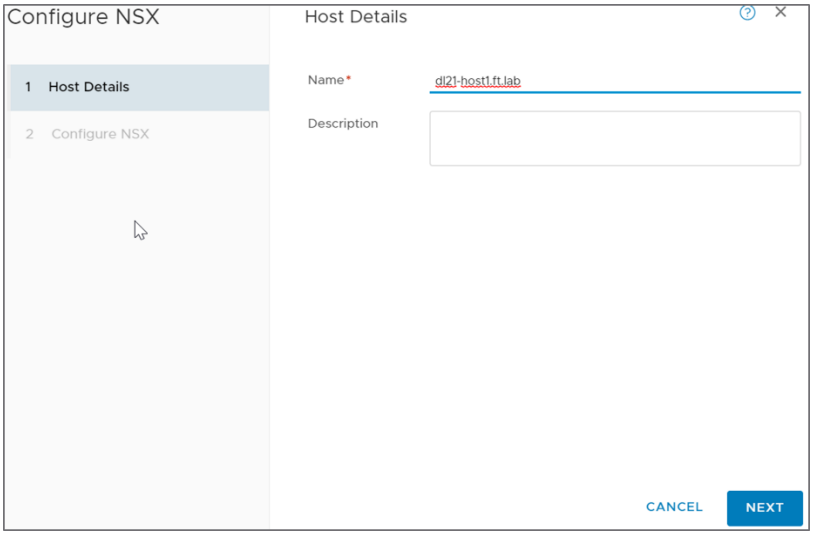
Everything we've done so far was just groundwork so that we had all the required objects for our transport node creation. It's going to be straightforward from there. First, we'll select the host. For that, navigate to (1) System/(2) Fabric/(3) Nodes/(4) Host Transport Nodes. In the "Managed by" drop down menu (5), select vCenter. Thanks to our linking of vCenter to NSX, all cluster/host information is available on the resulting screen:



2.2.1 Selecting the host

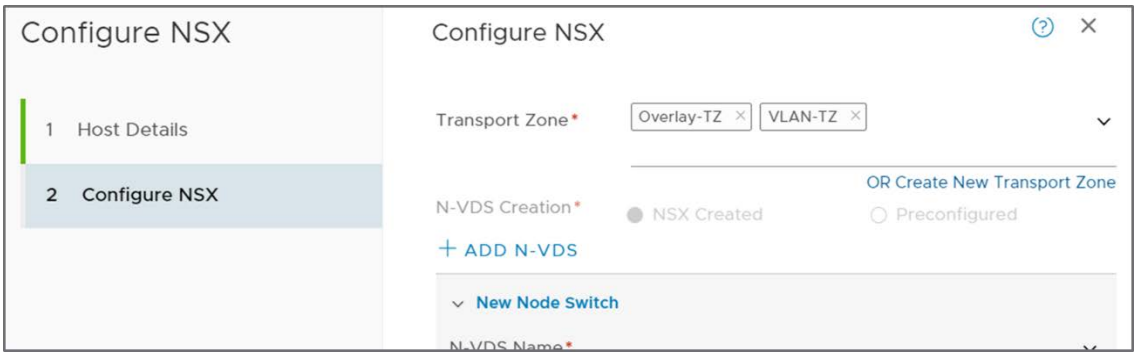
We're going to select the first host (here dl21-host1.ft.lab) and click on "CONFIGURE NSX":

Click next on the first dialog box:



2.2.2 Selecting the transport zones

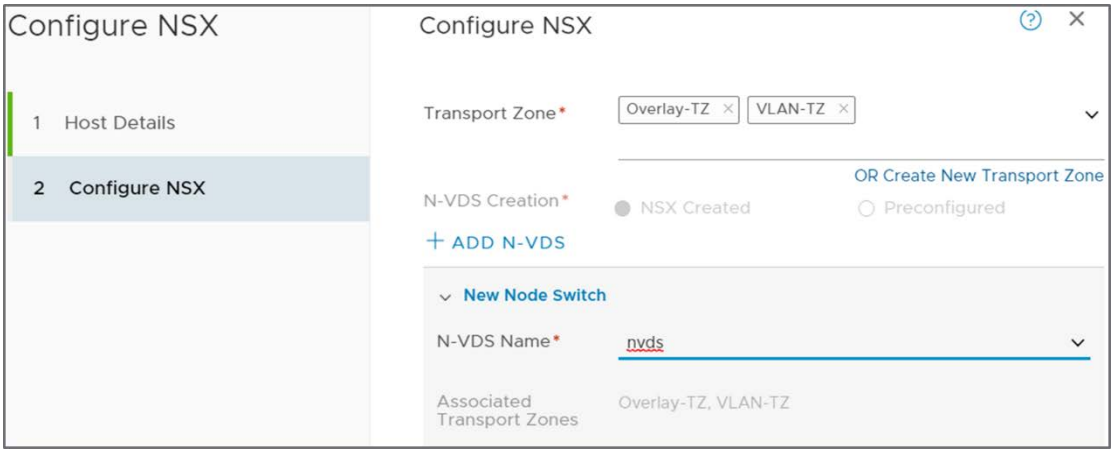
The top of the next screen allows specifying the transport zones we want to connect our transport node to. We'll select the two transport zones created earlier: Overlay-TZ and VLAN-TZ:



A transport node needs at least one N-VDS. The bottom part of the dialog box, under the section "New Node Switch", will guide us through the creation of the first N-VDS. We could add multiple N-VDS by clicking on "+ ADD N-VDS", but we'll only create one in this example.

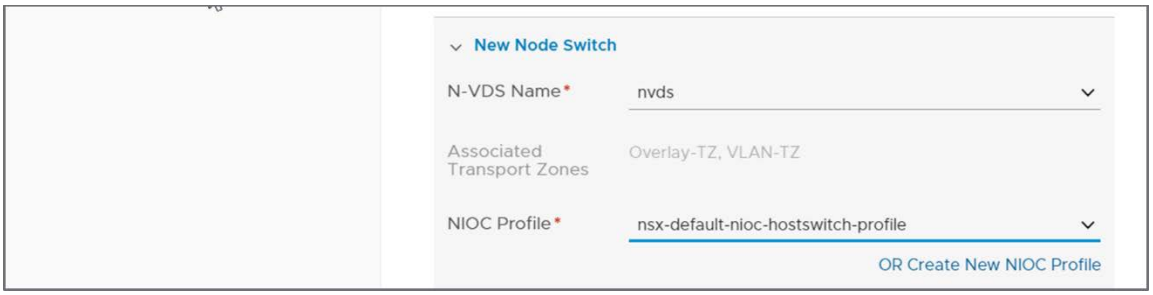
2.2.3 Setting the N-VDS name

The "N-VDS Name" field is a drop-down menu listing all the N-VDS names derived from the list of transport zones we have selected in the previous step. Here, both our transport zones have a similar N-VDS name, so we'll select the unique option, "nvds". As soon as you do that, the UI is showing you the associated transport zones this N-VDS is connecting to:



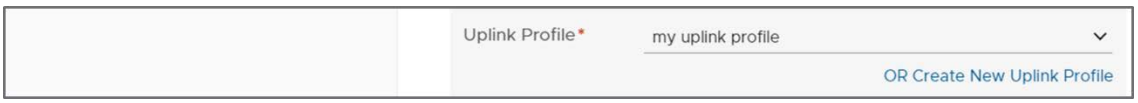
2.2.4 NIOC Configuration

The next field is about the NIOC profile that we're going to apply on the uplinks of our N-VDS. Network I/O Control is a feature allowing the user to specify shares, limits and bandwidth reservation on different class of traffic. The detail of this capability (equivalent to NIOCv3 in vSphere) is out of scope of this document. We'll just select the default NIOC profile:

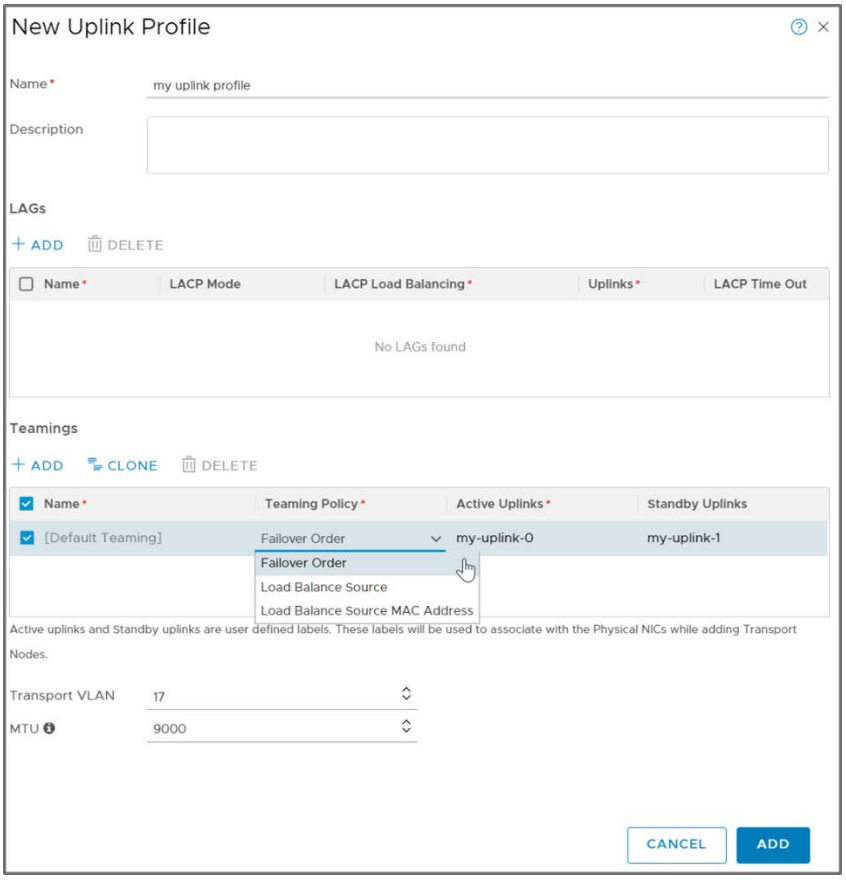


2.2.5 Uplink Profile

Next configuration is the uplink profile. We're going to click on "OR Create New Uplink Profile" instead of using a default one because -1- you're very likely to put some non-default values specific to your environment in there, -2- it gives us an opportunity to go over the different fields in the uplink profile.



We'll name this profile "my uplink profile".



The uplink profile gives you the opportunity to define LAGs. We won't be defining any in this example.

Under the section "Teamings", we'll only define the default teaming. A teaming lists the N-VDS uplinks and details how they are configured for redundancy and traffic load balancing.

We're going to select "Failover Order" as a teaming policy. That means that the N-VDS is going to send/receive traffic on the uplink in the "Active Uplinks" column first, while maintaining the uplinks in the "Standby Uplinks" as standby. No traffic will be sent/received on standby uplinks unless the active uplink fails, in which case it is replaced by one uplink from the standby list.

The other two teaming policies, "Load Balance Source" and "Load Balance Source MAC Address" are only available on ESXi transport nodes, not on KVM transport nodes. They allow "pinning" traffic to one of the multiple uplinks defined in the "Active Uplinks" column, based on the origin of the traffic (vnic or source mac address.) Those teaming policies would lead NSX to create one TEP per uplink, as overlay traffic would be sent/received concurrently on all the uplinks.

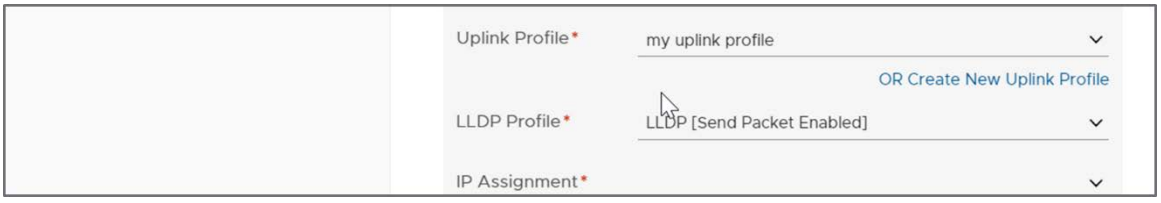
In the "Active Uplinks" (resp. "Standby Uplinks"), we'll enter the name "my-uplink-0" (resp. "my-uplink-1"). I chose those names so that it is obvious they are custom names with no meaning. The uplink names are just variables: you could use "bob" and "jane" if you wanted to. Those variables will be assigned a physical port a little bit further in the transport node creation dialog box.

Finally, the uplink profile defines the transport VLAN and the MTU of the N-VDS. Those two parameters (especially the transport VLAN) are the reason you are unlikely to use the default uplink profile defined in the system:

- The transport VLAN defines the VLAN ID in the 802.1Q tag that will be added to the overlay traffic on the uplinks of the host. If you chose "0" as a VLAN ID, the overlay traffic will be sent untagged. The overlay traffic is originated by the TEPs. Remember that we set an IP Pool for the TEP IP addresses a little bit earlier? The VLAN ID specified here must match the VLAN ID on which the subnet defined in the IP Pool is available in the physical infrastructure.
  I have chosen an arbitrary VLAN 17 to carry overlay traffic in this example, please use a value appropriate to your physical network environment.
- The MTU is the MTU of the N-VDS created using this uplink profile. Because of the overhead of the GENEVE encapsulation that NSX-T is using, we need at least 1600 bytes. Be aware that, depending on the features used in NSX-T, the minimum MTU might actually need to be larger. Our simple recommendation is to enable jumbo frames anywhere in the physical network backing NSX. Here, I've put 9000 bytes but again, feel free to use whatever value over 1600 that is appropriate for your physical network infrastructure.

2.2.6 LLDP Profile

We're done with the definition of the uplink profile, let's get back to the transport node creation dialog box. After selecting the uplink profile, the next field is the LLDP profile. LLDP is the Link Layer Discovery Protocol, an IEEE protocol allowing the exchange of some administrative information with Layer 2 neighbors. The two profiles that can be selected in the drop-down menu just decide whether we want to send LLDP frames to our neighbors. We will receive LLDP frames either way. I've selected "LLDP [Send Packet Enabled]" only because I'm not paranoid, but in the end, the choice you make here is not critical.

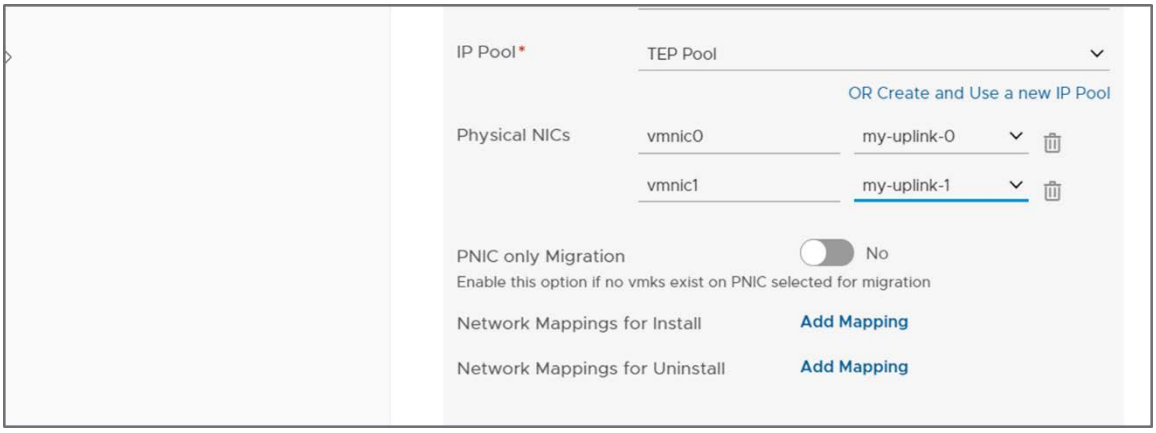


2.2.7 IP Assignment

The following “Assignment” field is about the way TEPs IP addresses are assigned. We’ve already discussed the three options and we’ll select the IP pool we have created earlier on:

2.2.8 Physical NICs mapping

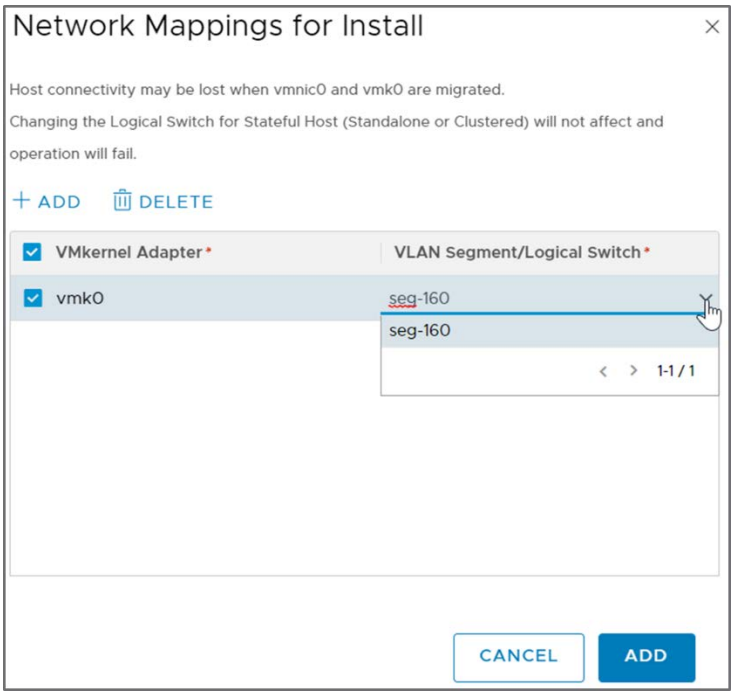
The next part is about the assignment of physical NICs to the uplink names defined in the uplink profile. We had defined “my-uplink-0” as active and “my-uplink-1” as standby, now is the time to explicitly define what “my-uplink-0” (and resp. “my-uplink-1”) means on *this* host. We’re going to assign physical uplink vmnic0 to my-uplink-0 and vmnic1 to my-uplink-1.



We’ll talk about the “PNIC only Migration” switch in section 3.3 “Installing NSX while keeping infrastructure traffic on VSS/VDS” later in the document. For now, leave this unchanged to “No”.

2.2.9 Network Mappings for Install

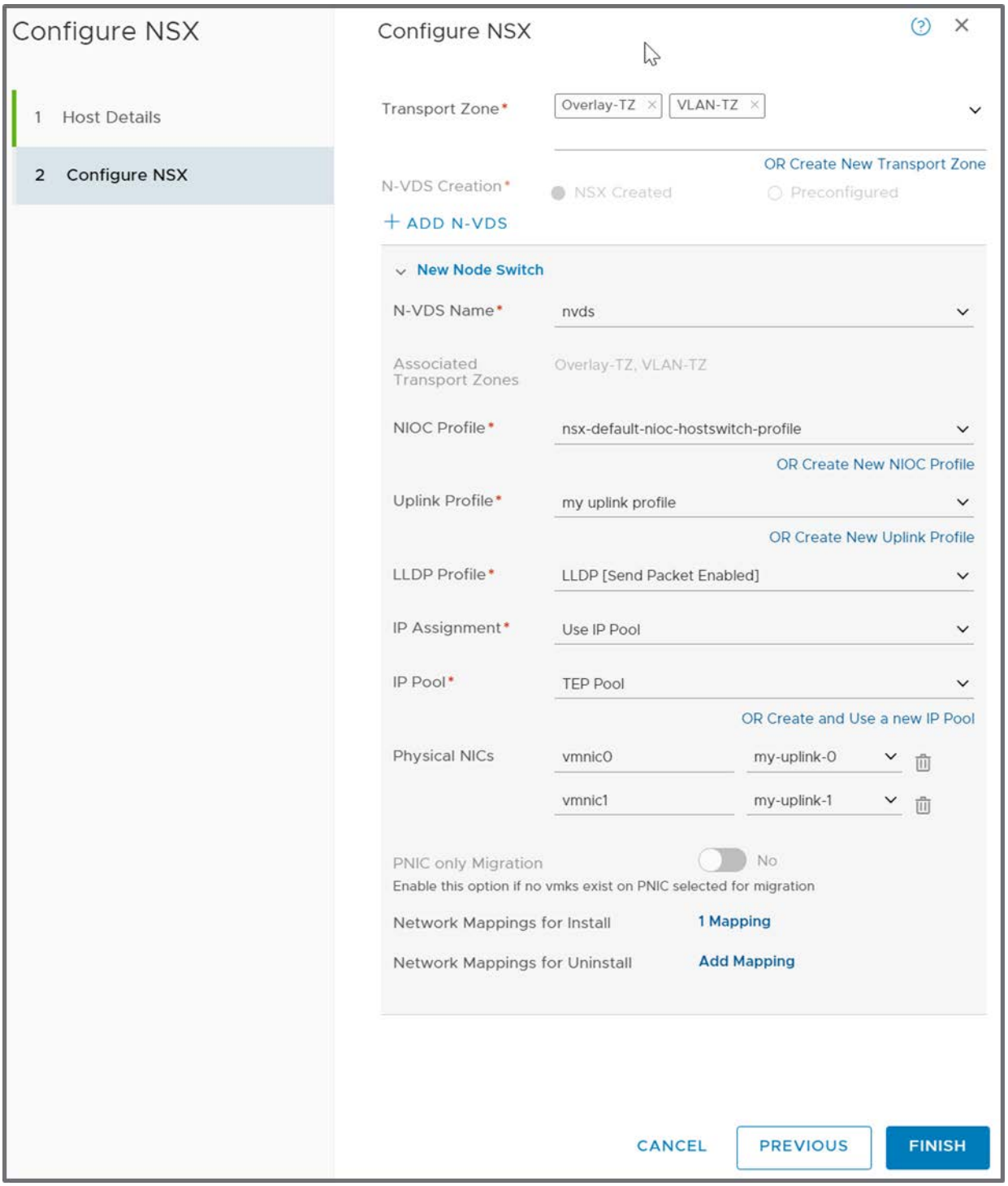
Because we’re migrating both vmnic0 and vmnic1 to the N-VDS, the original VSS to which our management vmk0 is attached is going to lose network connectivity to the outside world. We thus need to migrate this vmk0 to the N-VDS at the same time we create the transport node. This migration is defined in “Network Mappings for Install”. Click on “Add Mapping” in order to get the following dialog box:



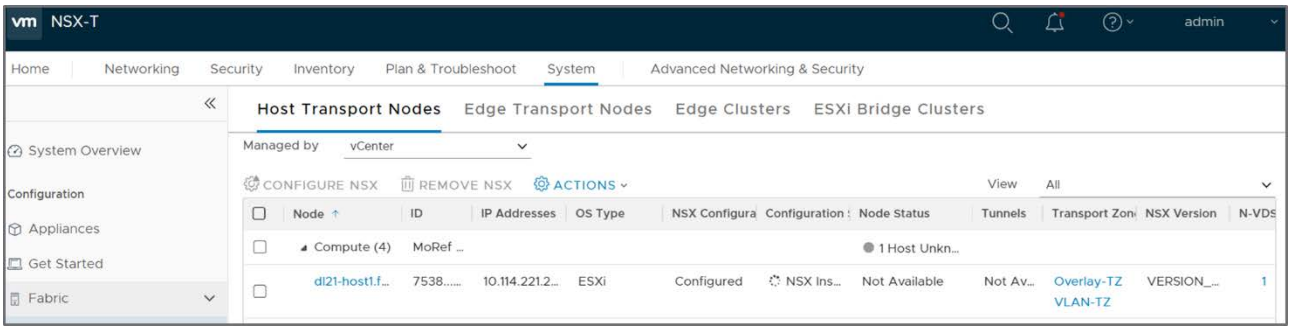
Here, click "+ ADD" to create a mapping between vmk0 and segment "seg-160" that we had created for this purpose in a previous step.

2.2.10 Starting the install

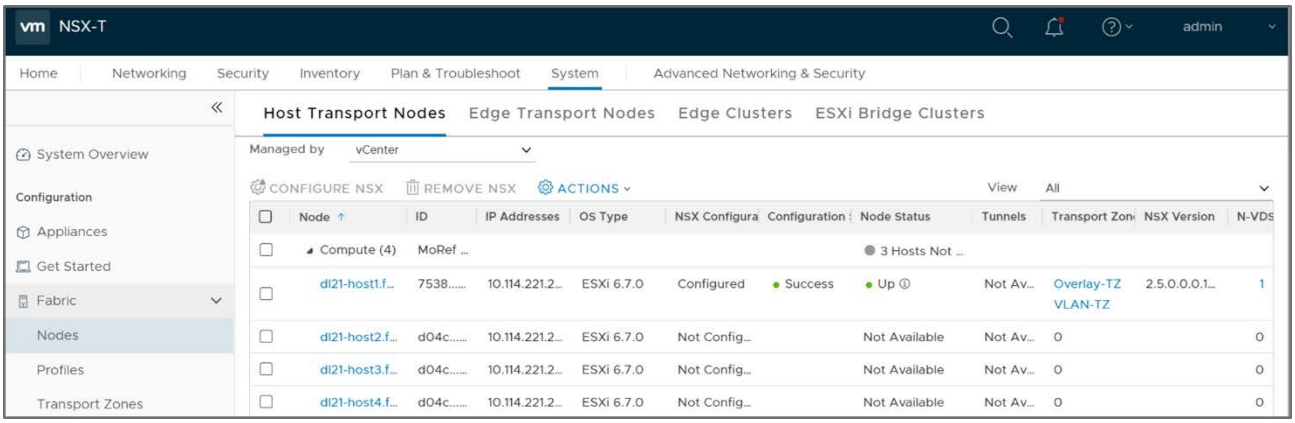
That's the last step, we're ready to perform our transport node creation. Here is a full picture of the dialog box before we click "FINISH" and start the installation:



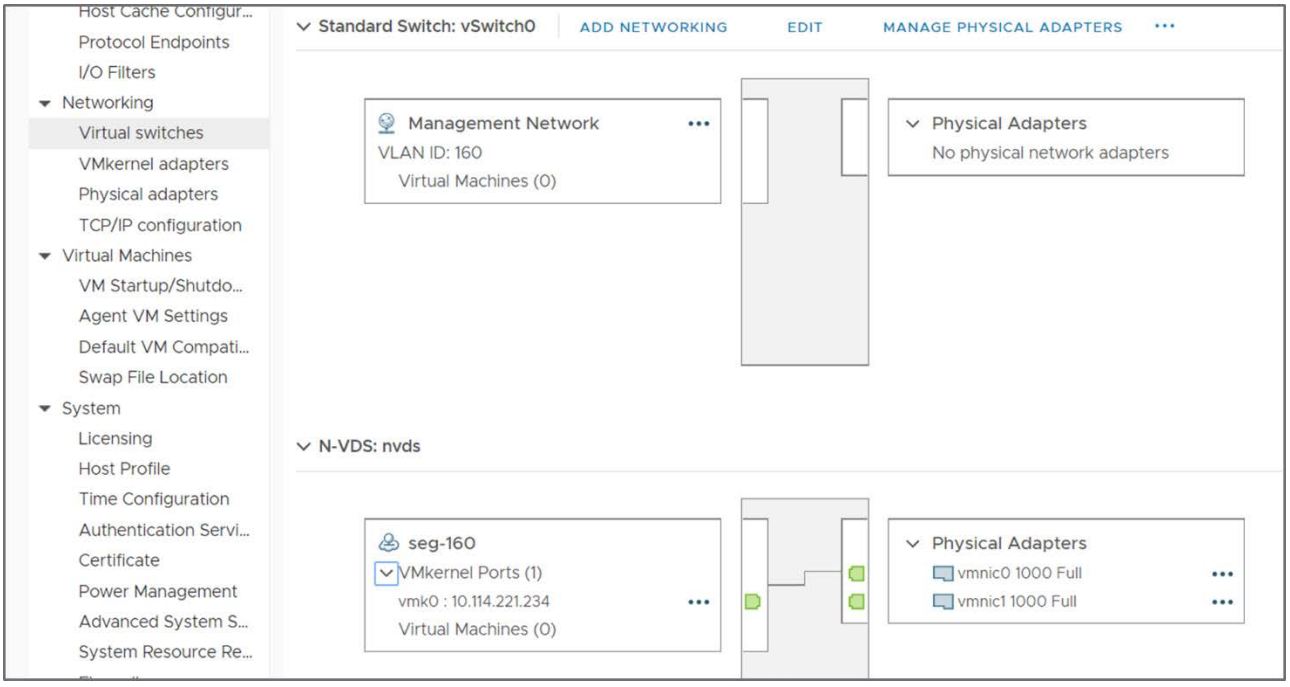
As soon as we hit FINISH, the transport node configuration starts:



And hopefully, after few minutes, this is a success:



This is how our new N-VDS looks in vCenter for that host:



You'll notice that our original VSS (vSwitch0) has not been deleted. It's just that we've taken its vmnics and vmk0 and moved them to a new vswtich. You can potentially delete the VSS, but it might make sense to keep it if you want to configure some mapping for the uninstallation of NSX. Whether you

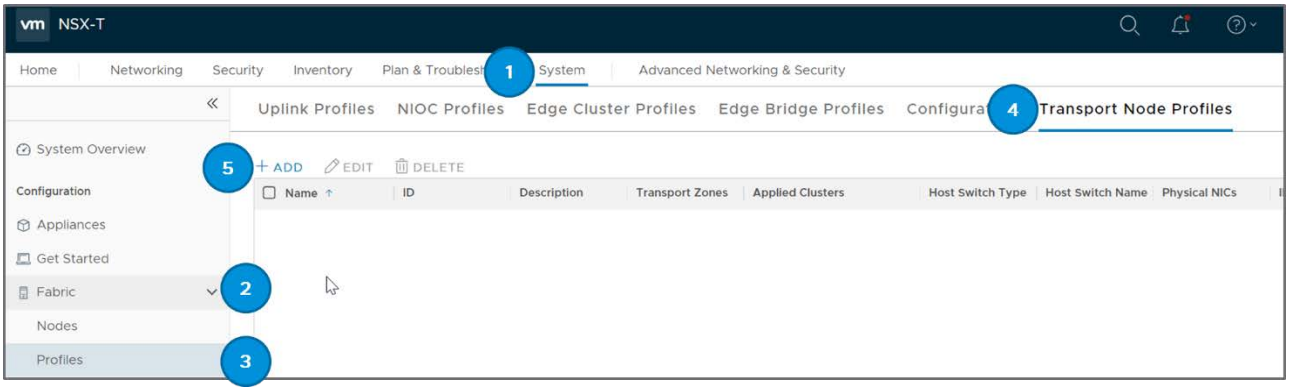
started the NSX installation from a VSS or a VDS, I recommend you read section 4 below “Uninstall, Step-by-Step Example” before removing the original virtual switch.

2.3 Installation using a Transport Node Profile (TNP)

The transport node installation as described in the previous section can be performed from API and entirely automated, but NSX already offers a way of preparing multiple transport nodes at once thanks to the Transport Node Profile (TNP). The TNP is a kind of template summarizing all the parameters that we have manually entered when configuring an individual transport node in the previous section. The TNP can be then applied to a whole vCenter cluster, resulting in the installation of NSX on all the hosts in the cluster. Of course, all the hosts in the cluster must be fine with the same configuration: same vmkernel interfaces, same management VLAN, same vmnics etc. In this part, we’re going to turn into transport nodes the remaining hosts in the compute cluster of my data center in a single shot, using a TNP.

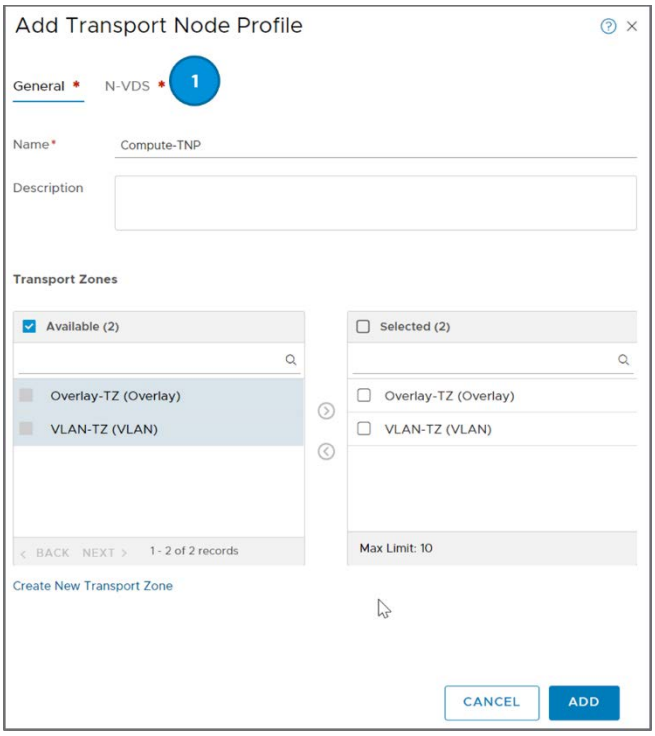
2.3.1 Create a new TNP

To create a transport node profile, navigate to: (1) System/(2) Fabric/(3) Profiles/(4) Transport Node Profiles and click on (5)“+ ADD”:



2.3.2 Assigning a name and selecting the transport zones

On the first screen, we need to assign a name to the TNP and select the transport zones to which the transport nodes will attach. Here, we’re creating “Compute-TNP” and select both our transport zones.



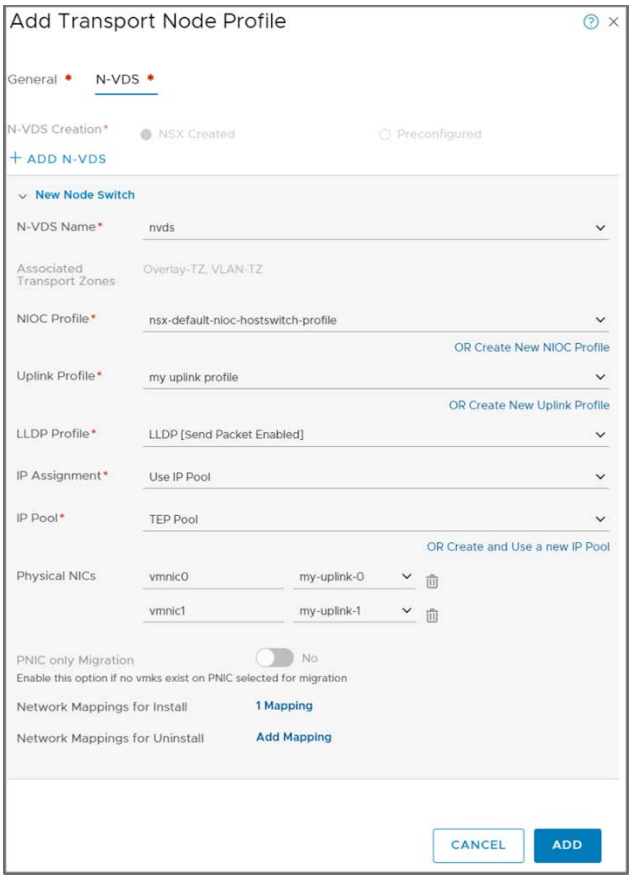
Click on “N-VDS” (1) to carry on the TNP configuration (don’t click “ADD” yet).

2.3.3 Setting the other parameters

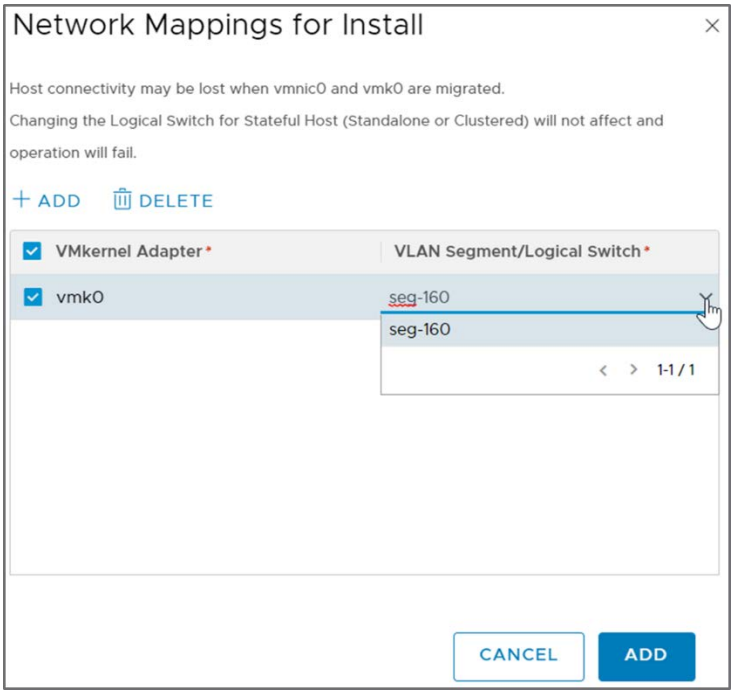
This screen is very similar to the individual transport node creation we saw in the previous part. The only differences are:

- The transport zones have been selected in the previous screen
- The IP assignment can only be “IP Pool” or “DHCP”. This is because network ranges have to be entered on a per host basis and the TNP is meant to be applied to several different hosts.
- The vmnic selection is not a drop down menu, the user has to type in the exact name.

We’re going to enter exactly the same values as the ones we’ve used in the previous section:



Clicking on “Add Mapping” in the “Network Mappings for Install” we’re also going to specify the migration of vmk0 to seg-160:

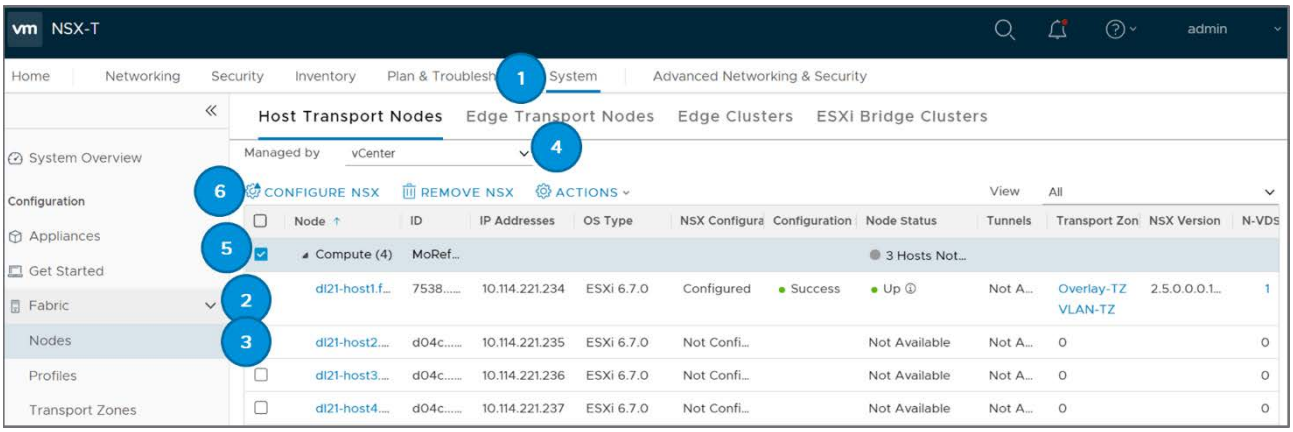


Finally, click “ADD” to eventually save the new transport node profile.

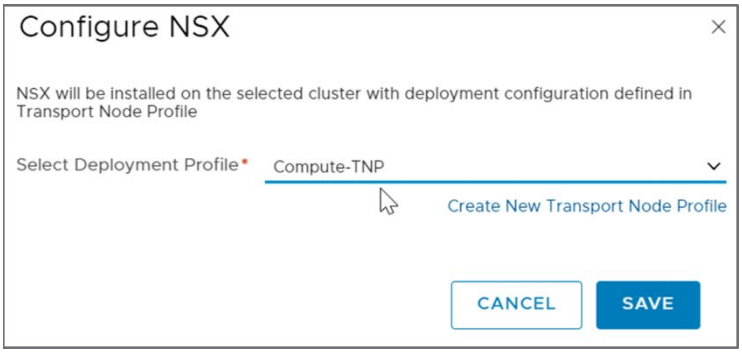
2.3.4 Applying the TNP to a cluster

Now we have created a transport node profile, we can apply it to our compute cluster. To select the compute cluster, navigate to: (1) System/(2) Fabric/(3) Nodes, in the “Managed by” field (4), select “vCenter”: this will retrieve the cluster information from our compute manager.

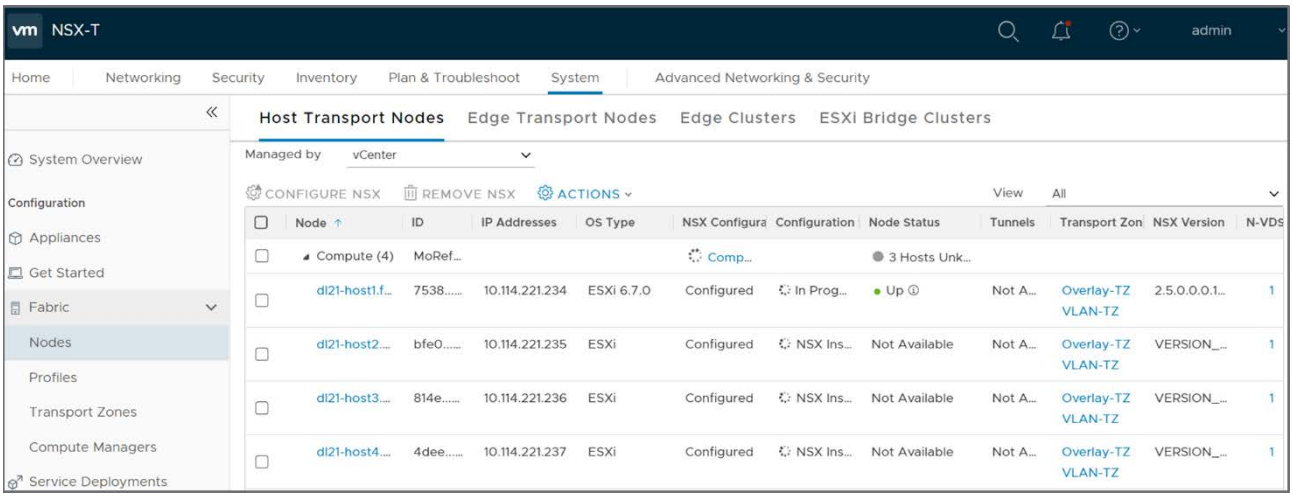
Then select the checkbox in front of the “Compute” cluster (5). The “Configure NSX” option is now available, click on it (6).



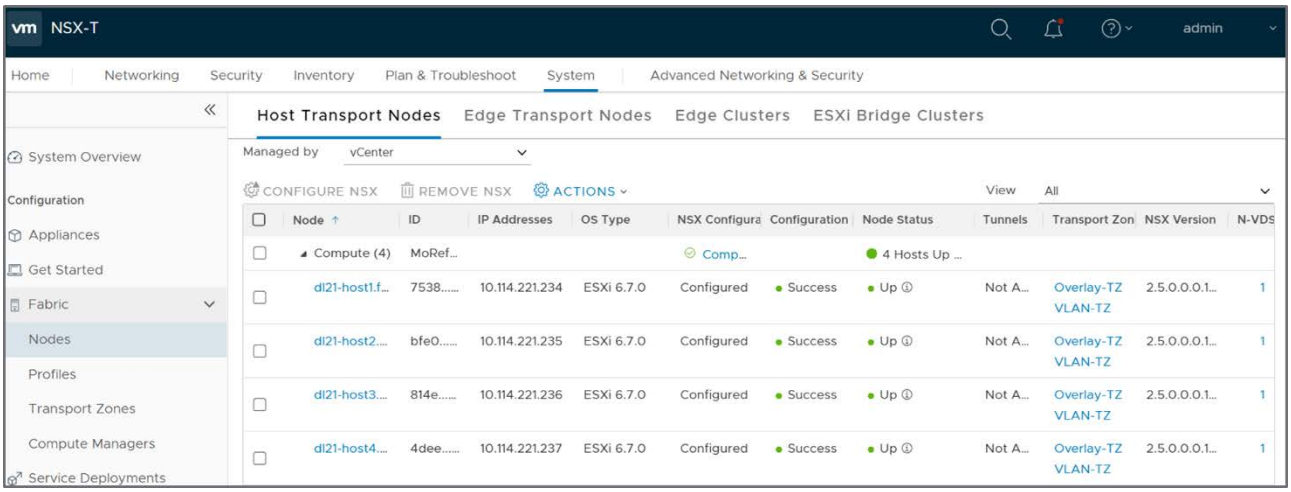
The “Configure NSX” dialog box opens, you can now select a TNP to apply to this cluster. Choose the only option, our newly created “Compute-TNP” and click “SAVE”



As soon as you press “SAVE”, the TNP is applied to the compute cluster and the installation of NSX on all the hosts in the cluster is initiated:



Note that dl21-host1, which was prepared in the previous section, is already shown as "up". The TNP is applied to dl21-host1, but it will not result in any change because the TNP is just re-applying the same configuration. After few minutes, all the hosts are available for NSX:

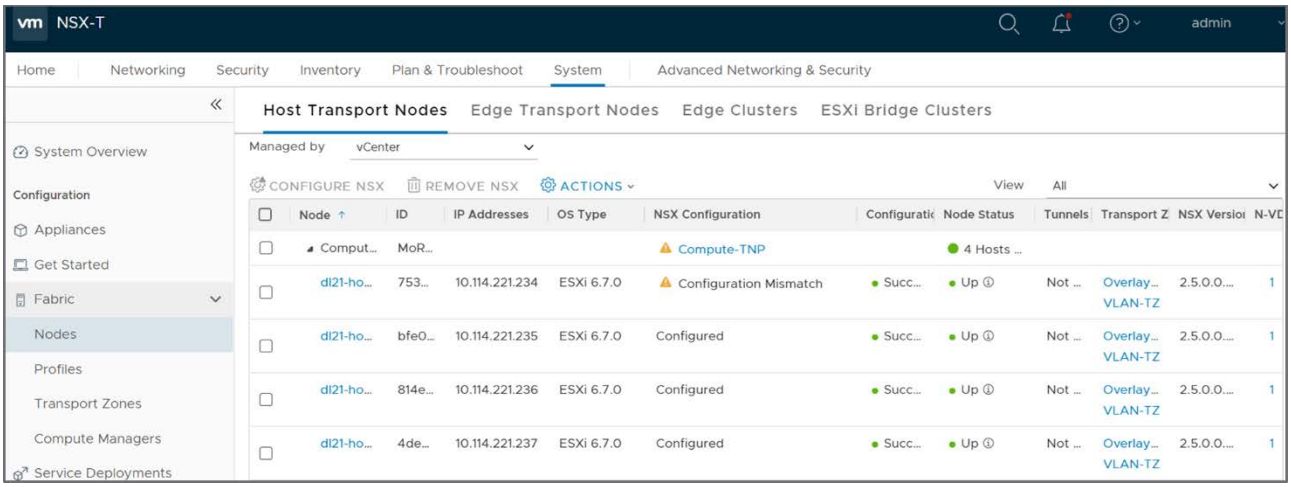


2.3.5 Some few additional things about the TNP

I presented the TNP as a kind of template, but when applied to a cluster, it is more than that. In fact, the TNP is permanently linked to the cluster and any change made to the TNP will immediately be applied to the whole cluster.

If the vCenter admin moves a host into the compute cluster, the Compute-TNP is immediately applied to this host. If the vCenter admin moves a host out of the cluster, it retains its NSX configuration but will not be associated to the TNP anymore. That means that further modification to the TNP will not affect the host moved out of the cluster.

Finally, even if the goal of the TNP is to apply a consistent NSX configuration to all the hosts in a cluster, it is still possible to override the TNP on a per-host basis. For example, if I select dl21-host1 and click on "Configure NSX", any configuration change I make will only be applied to this host. To illustrate this capability, I've removed the second uplink (vmnic1) from dl21-host1:



Because it has been manually edited, host dl21-host1 is now showing as a "Configuration Mismatch" and the TNP itself is shown as having an exception (orange triangle icon).

From there one, dl21-host1 is decoupled from the TNP, as if it was removed from the cluster. Changes to the TNP will not affect dl21-host1. I can click on the little orange triangle icon in front of "Compute-TNP" and I will be given the option to reapply the TNP to all the hosts in the cluster. Re-adding vmnic1 to dl21-host1 would also make it in sync with the TNP again.

2.4 Miscellaneous recommendations

While writing this paper, I installed and uninstalled NSX tens of times. The good news is that when you're following the configuration path that I've been describing, it has been consistently working fine. Here are some few points you need to be aware of when installing NSX:

2.4.1 Make sure you have the proper ESXi version

Failure to do so will result in an installation failure and I must admit that the error messages that the UI retrieves from ESXi is not necessarily clearly pointing to a software incompatibility.

2.4.2 Management vmkernel migration warning

Migrating the management vmkernel interfaces is the trickiest part of the installation in my opinion.

You absolutely need to be sure that the management interface will maintain connectivity after migration. The target VLAN-backed segment must have the same network access as the source portgroup/dvportgroup where the vmk interface was initially.

Here is a limitation you need to be aware of and that I learnt the hard way. It's typical for vmk0 to be associated with a physical port, most likely vmnic0. That means that vmk0 will use vmnic0's mac address. Because of that, you need to migrate vmk0 to the N-VDS along with vmnic0. If you don't do this, NSX will stop the migration procedure and warn you anyway. Unfortunately, there is still a corner case that NSX will not catch in NSX-T 2.5. **If the teaming policy for the VLAN segment where vmk0 is attached puts vmnic0 in standby state, the host will lose connectivity.** The next release will prevent this. If you end up losing connectivity during migration, this is a possible cause for the issue. Check part 5.1.2 "Removing one vmnic from the N-VDS" for a recovery procedure from the host console.

2.4.3 Inconsistency in the vmkernel mappings will fail your install

This is especially painful when using a transport node profile, as errors can be applied to several hosts in one shot. Make sure that your "Network Mappings for Install" only refer to existing vmk and that all vmks that need to be migrated are part of a "Network Mappings for Install". In fact, the recommendation also partly applies to "Network Mapping for Uninstall" (procedure we'll see in section 4, "Uninstall, Step-by-Step Example"): make sure you don't specify vmks that don't exist. The consequence for any inconsistency is not as dramatic as messing up the management vmkernel migration, however, you might have to waste time cleaning up the configuration.

3 Other scenarios

The use case I have chosen to illustrate in the previous step-by-step example is in fact almost the most complex migration scenario:

- All the vmnics of the host have been migrated to the N-VDS in one shot
- The migration includes the critical migration of the management vmkernel interface (here vmk0).
- The transport node attaches to both a VLAN and an overlay transport zone.

This section explores some variations over the basic scenario. You'll see that in most cases, it's a matter of removing steps from the basic scenario (which makes me wonder if the term "basic scenario" is appropriate!)

3.1 Starting from the VDS instead of VSS

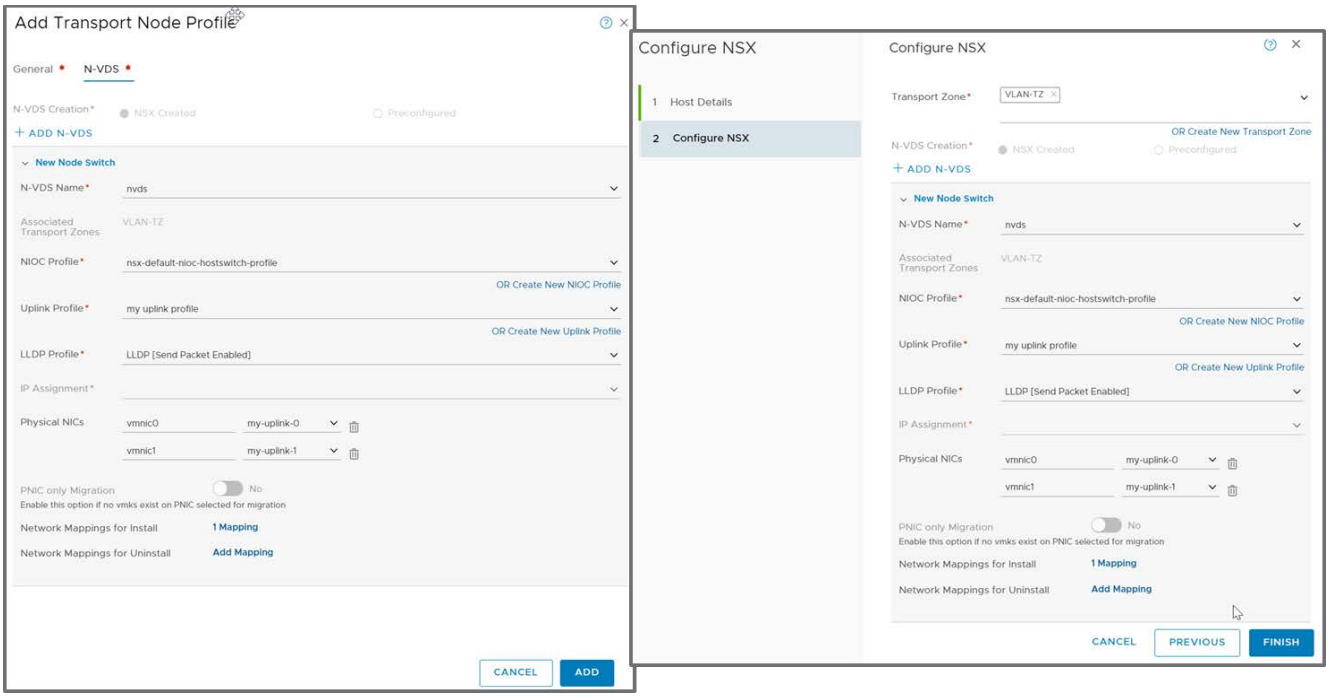
The basic scenario started from a VSS, as it is the default state for an ESXi host. However, it is also very common to install NSX on the top of an existing environment based on vCenter and the VDS.

The transport node preparation does not depend on the initial virtual switch type, so the configuration steps described in the first part would be unchanged if we had used a VDS instead of a VSS as a starting point.

3.2 Installing NSX for VLAN networking or overlay networking only

The use of NSX-T for VLAN micro-segmentation is quite popular and does not require setting up an overlay. Some steps described in the basic scenario are thus superfluous:

- There is no need to create an overlay transport zone.
- Because that case does not involve tunnels, there is no TEP and thus no need for an IP address pool.
- If you only attach your transport node to VLAN transport zones, the UI grey out the "IP Assignment field" in the transport node preparation (and this whether it is TNP-based on individual TN preparation, see screenshots below.)

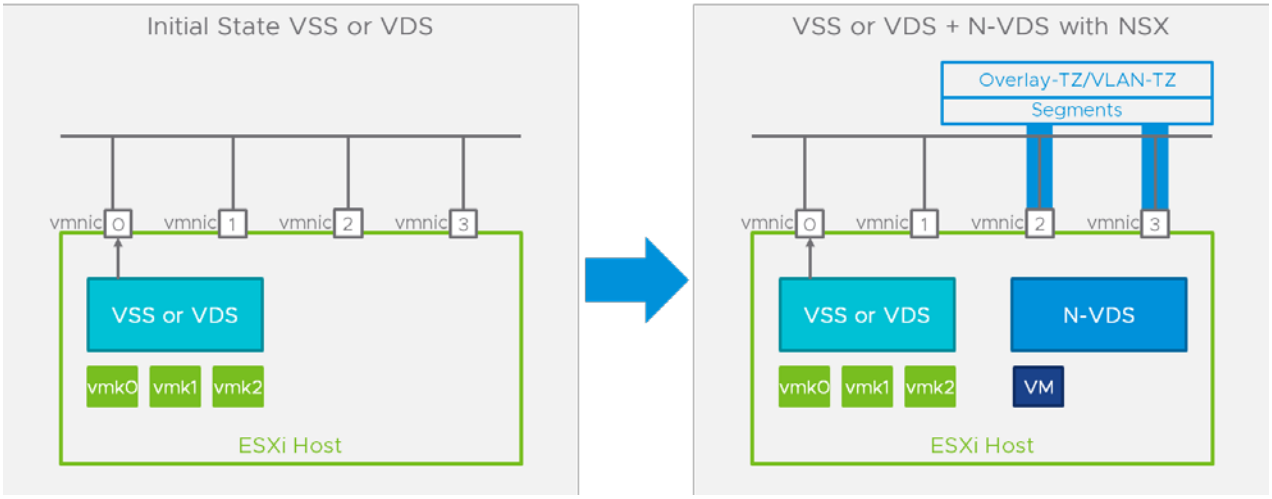


3.3 Installing NSX while keeping infrastructure traffic on VSS/VDS

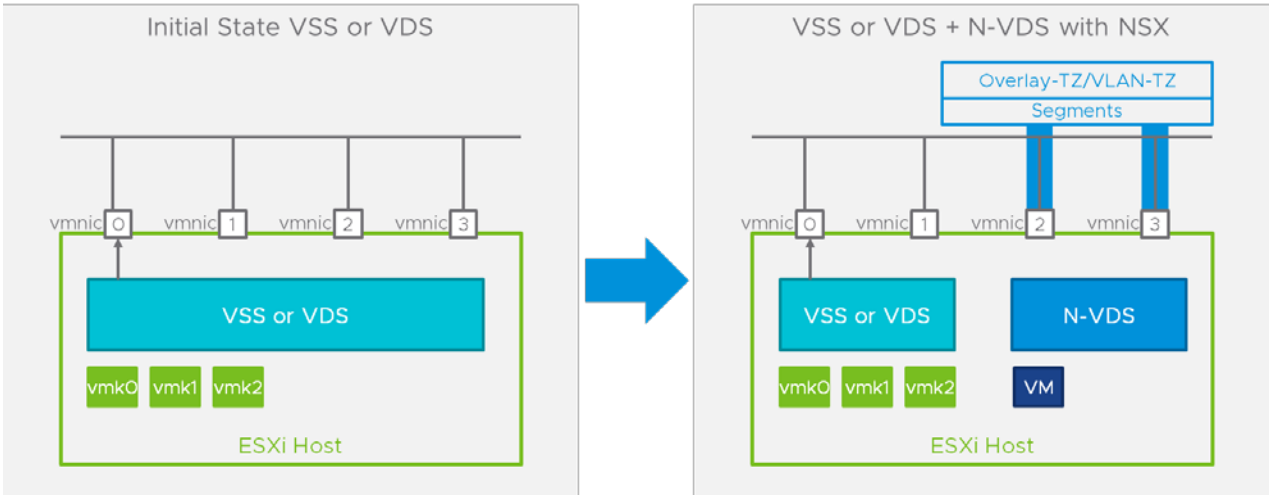
In the current VMware model, a physical uplink can only belong to a single virtual switch. In the basic scenario, as we only have two physical uplinks, we can only deploy a single virtual switch with uplink redundancy on this host. Because we want to install NSX, this implies that the single virtual switch owning the two uplinks must be the N-VDS.

Now, if most modern servers now come with two high speed physical adapters, it is still quite common to have hosts with four or more physical uplinks. In that case, it's possible to dedicate a redundant pair of uplinks to the N-VDS, while keeping a VSS or VDS with another pair of redundant uplinks for infrastructure traffic (by infrastructure traffic, I mean traffic to/from vmkernel interfaces like management, storage, vmotion etc.)

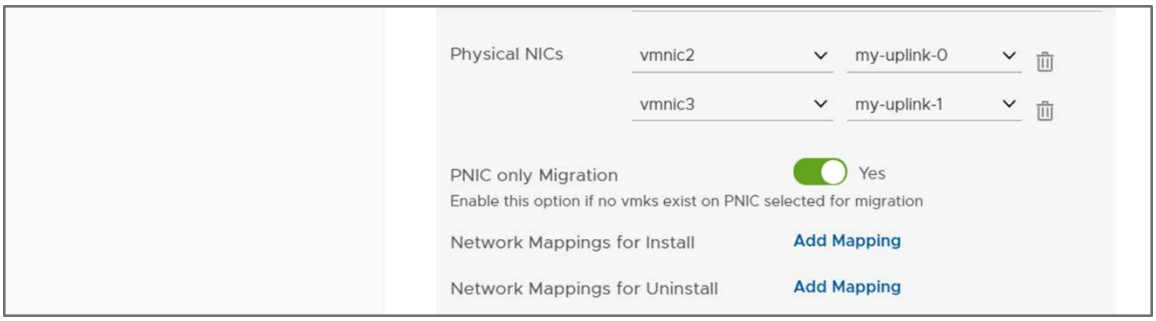
The following diagram represent the installation of an N-VDS on two available uplinks of a host already configured with a VDS or a VSS for its management traffic. The benefit of this model is that we can now use NSX and all its networking features for VM traffic, while keeping the infrastructure traffic unchanged. The only difference with the basic scenario here is that you don't have to configure the migration of any vmkernel interface (no "Network Mapping for Install".) Of course, here, you also have to assign vmnic2 and vmnic3 to the N-VDS uplinks instead of using vmnic0 and vmnic1:



Note that in the above scenario, we are installing the N-VDS on unused vmnics. The following example is almost similar, but this time vmnic2 and vmnic3 previously belonged to the VSS or VDS:



In this specific case, where we are migrating vmnics from one vswitch to the N-VDS *without* migrating any vmkernel interface, we must set the “PNIC only Migration” flag in the transport node creation procedure. The following screenshot shows you how this would look like in the UI:



I must admit that I’m not a fan of this flag. It was added to make sure that the user did not end up doing an NSX installation without forgetting the vmk migration. We’ll get rid of this in the next release anyway.

3.4 Multiple vmkernel interfaces

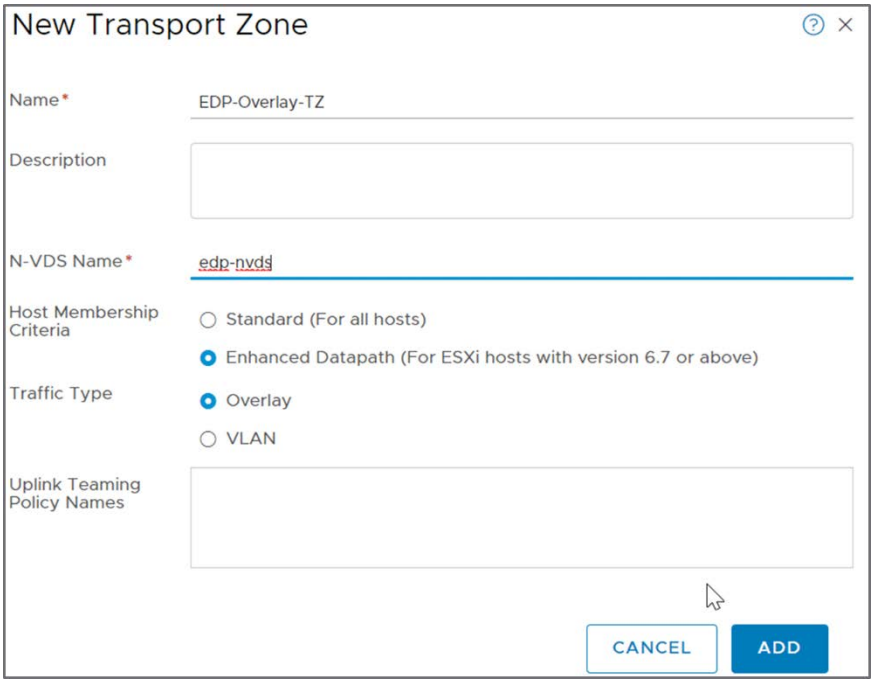
The basic example only handles the migration of the management vmkernel interface to the N-VDS. This is however the most critical one, as a failure to configure this migration properly could make the host unreachable. Migrating multiple vmkernel interfaces is just a matter of entering additional "Network Mapping for Install" in the UI.

3.5 Transport node with multiple N-VDS

In the overwhelming majority of cases, a single N-VDS per host is necessary when deploying NSX. A common specific scenario where it makes sense having multiple N-VDS on a host is when both a standard and an Enhanced Datapath N-VDS are required on the same transport node.

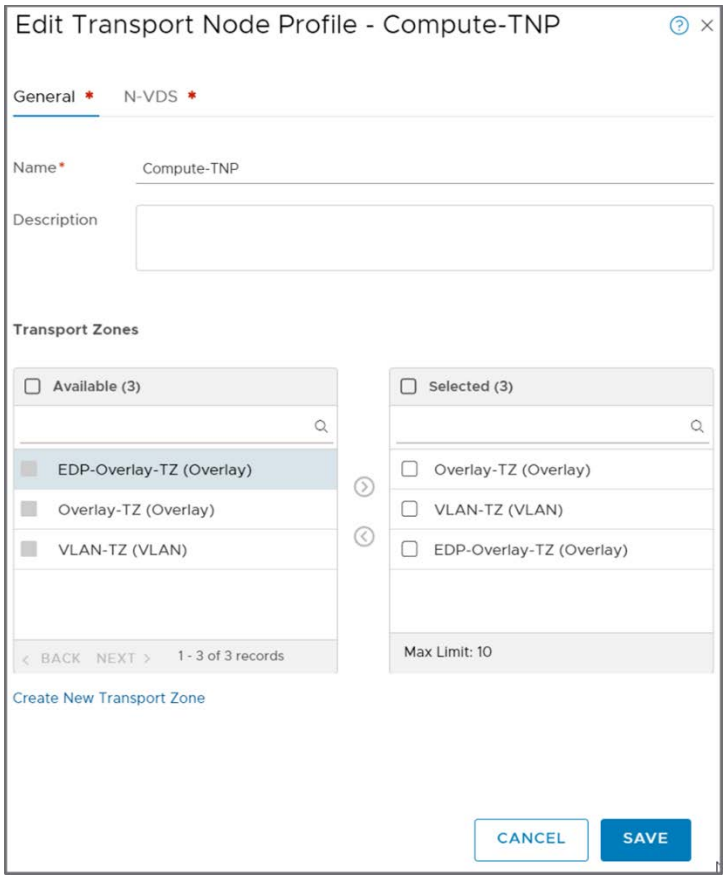
The configuration for multiple N-VDS is just an extension of the basic use case. You're just going to join the transport node to some additional transport zone(s) with a different N-VDS name. Then, in the transport node configuration, you just need to click "+ ADD N-VDS" and configure the additional N-VDS. Here are some few screenshots showing the configuration of the additional parameters needed for running an additional N-VDS called "edp-nvds" on the top of the "nvds" of our basic scenario.

First, I need a new transport zone, with a different N-VDS name (here "edp-nvds") and set for Enhanced Datapath:

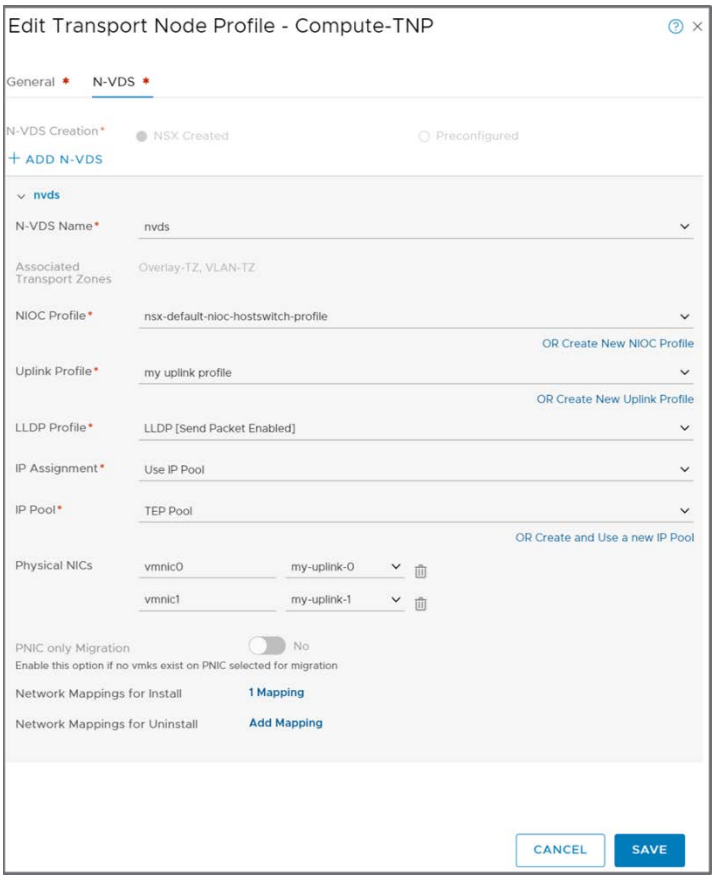


Now, let's edit the current "Compute-TNP".

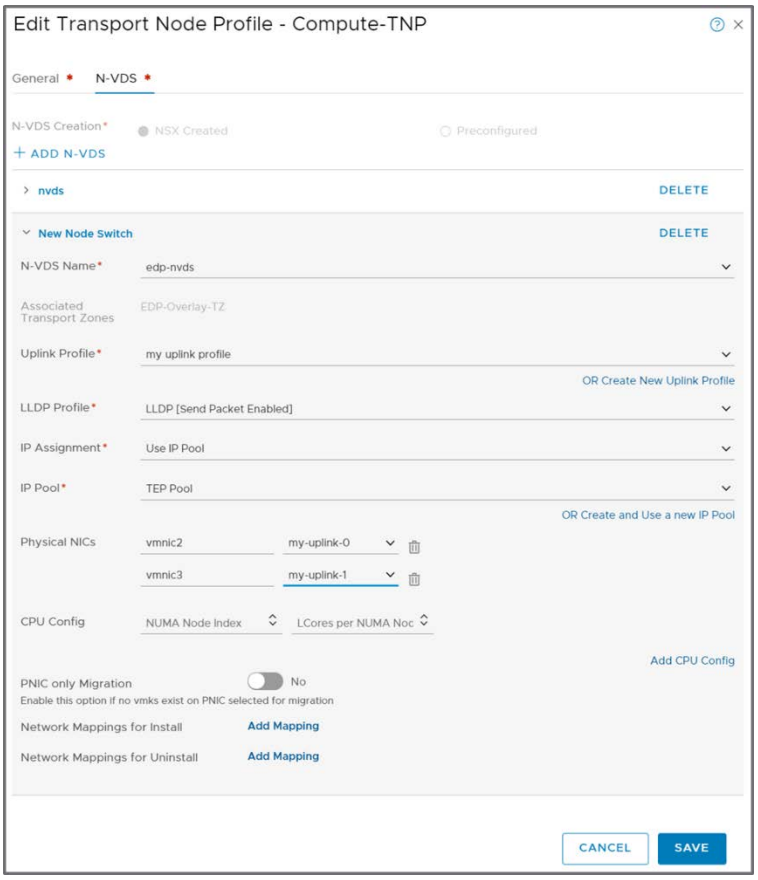
First, we're going to add the new transport zone to it:



Now, let's click on N-VDS tab at the top of the dialog box. The following screenshot is showing the configuration for the N-VDS called "nvds" that we had set up earlier (you'll notice by the way this is N-VDS is now showing as "nvds" in the UI.)



Now, let's click on "+ ADD N-VDS" and configure our new N-VDS by selecting "edp-nvds" in the N-VDS Name drop down menu:



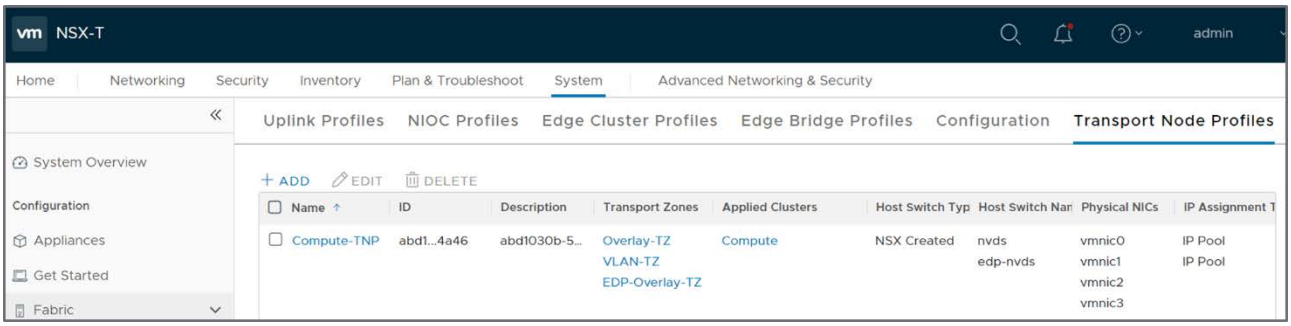
You'll notice that the "Associated Transport Zones" has been automatically set to "EDP-Overlay-TZ" based on the N-VDS name.

I kept the exact same configuration for Uplink Profile, LLDP Profile. There is no NIOC profile because it is not currently supported on Enhanced Datapath N-VDS.

You'll notice that I also kept the same IP Assignment as the one I had for "nvds". This is not desirable, but this is currently a limitation NSX-T. When multiple N-VDS are configured on a host, there is a single IP stack for all their TEPs. We're working on lifting this restriction, but it's not committed in the roadmap yet.

I mapped vmnic2 and vmnic3 to "my-uplink-0" and "my-uplink-1" from the uplink profile. This is because vmnic0 and vmnic1 are already assigned to "nvds" and we cannot share uplinks between virtual switches.

I'm going to skip over the "CPU Config", which is Enhanced Datapath N-VDS specific and just "SAVE" the updated transport node profile. The following screenshot shows the result in the UI:



Note that as soon as I saved the transport node profile, it got applied to the hosts in the compute cluster. I would have been glad to show you the result but the realization failed in my lab because the Enhanced Datapath N-VDS has strict NIC requirements and does not support the VMXNET3 NICs of my nested environment… that seems to be the perfect example of why you should only be using the Enhanced Datapath N-VDS if you know what you're doing!

4 Uninstall, Step-by-Step Example

Uninstalling NSX is a matter of:

- Migrating vmnics to a VDS or VSS
- Migrating vmkernel interfaces to dvportgroups/portgroups
- Deleting the NVDS and removing the NSX bits

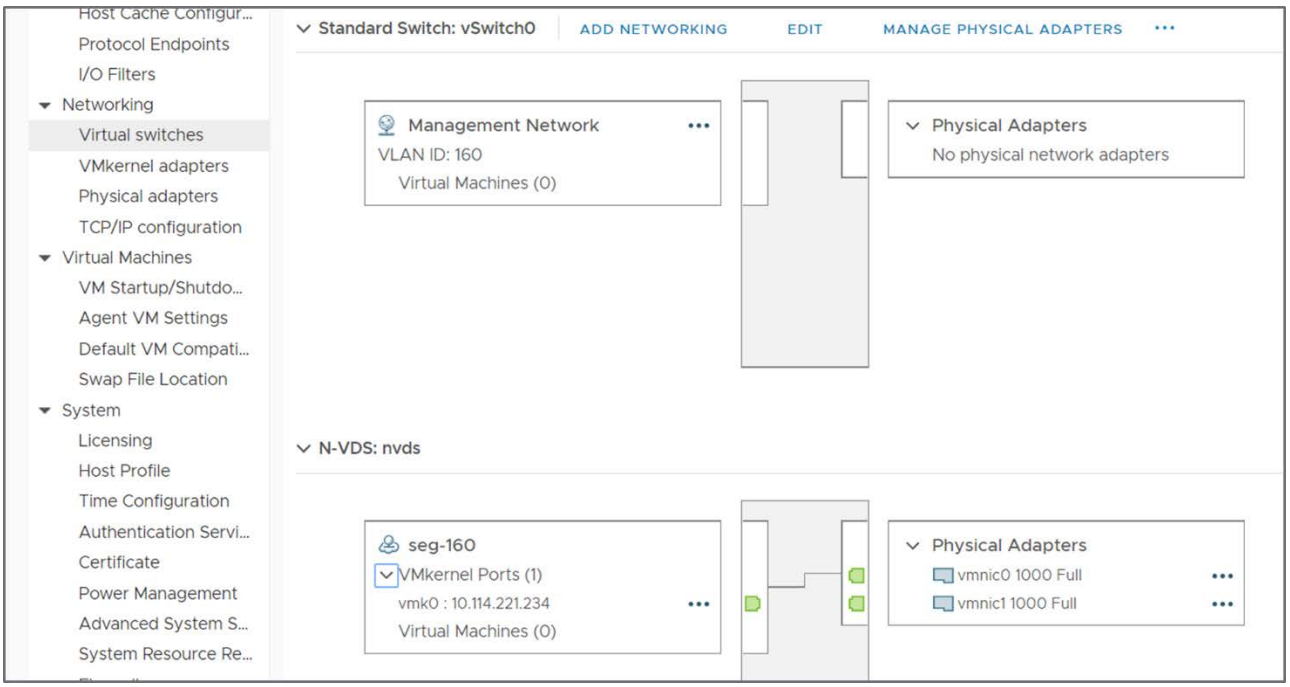
Whether it is for an individual host or a transport node profile, the parameters for the uninstall procedure are stored behind the "Mapping for Uninstall" link. Here, you'll be able to configure two kinds of mappings.

- vmk to portgroup/dvportgroups: this will migrate the specified vmks as well as identify the virtual switch (VSS or VDS) to where you're reverting to. Notice that the procedure thus assumes that this VSS or VDS is already present on the host.
- Physical port mapping: this determines the vmnics that you're assigning to the virtual switch identified by the portgroup/dvportgroup specified above.

Of course, it is important that the vmkernel interfaces (or at least the management vmkernel interface) have connectivity through the (dv)portgroup/physical uplinks specified in this mapping.

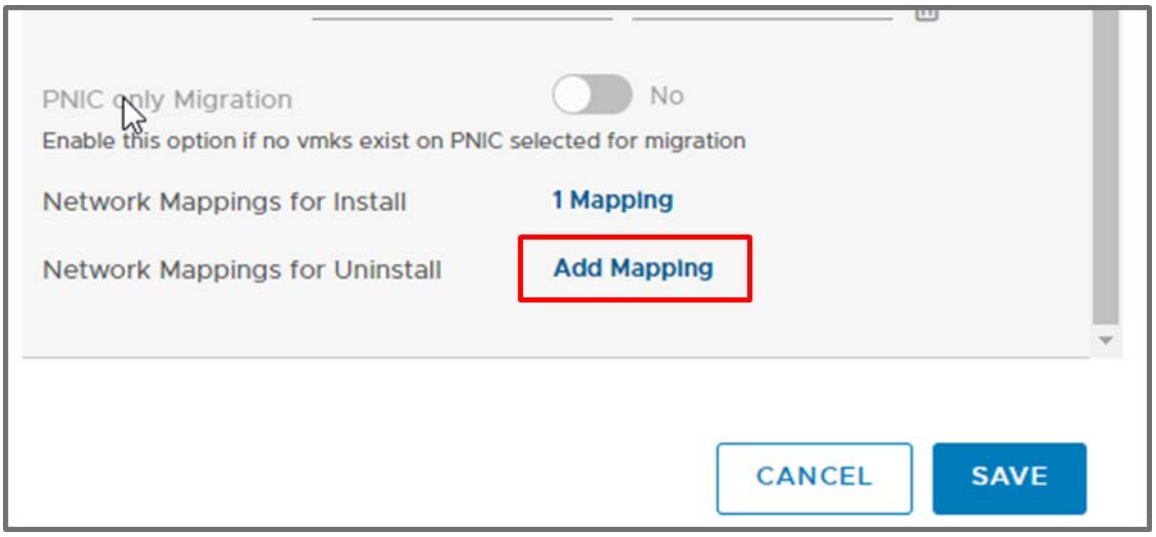
4.1 Migrating back to a VSS

Our basic example started from hosts with a VSS. We did not delete the VSS nor its management portgroup, so they're still showing up on the hosts:

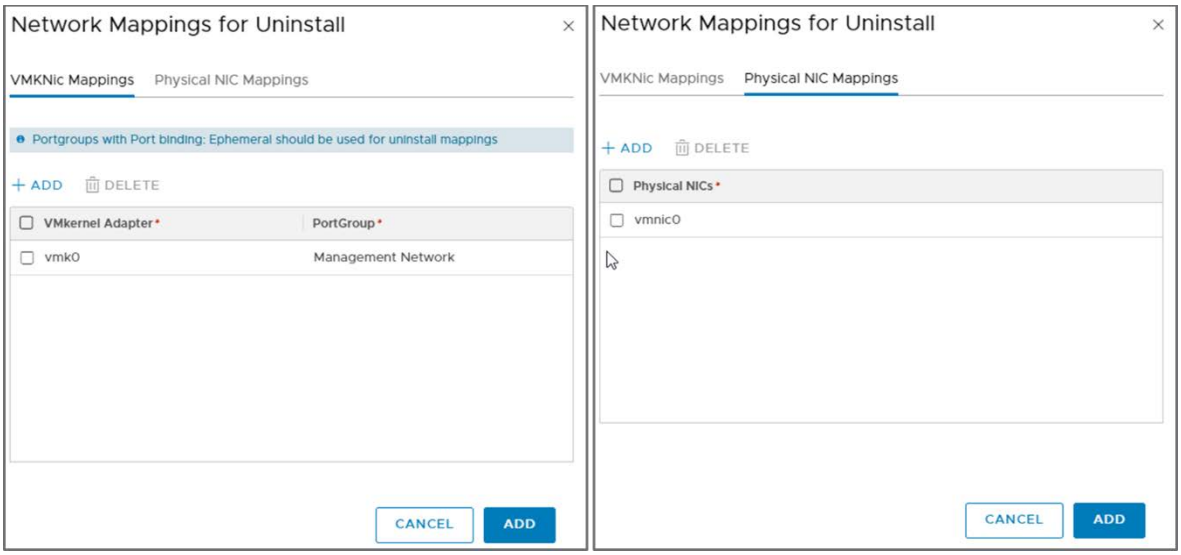


The beauty with the VSS is that, should you have deleted it and/or its portgroup(s), you could easily recreate it from vCenter or the host UI anyway.

We're going to edit the Compute-TNP to configure the mappings:

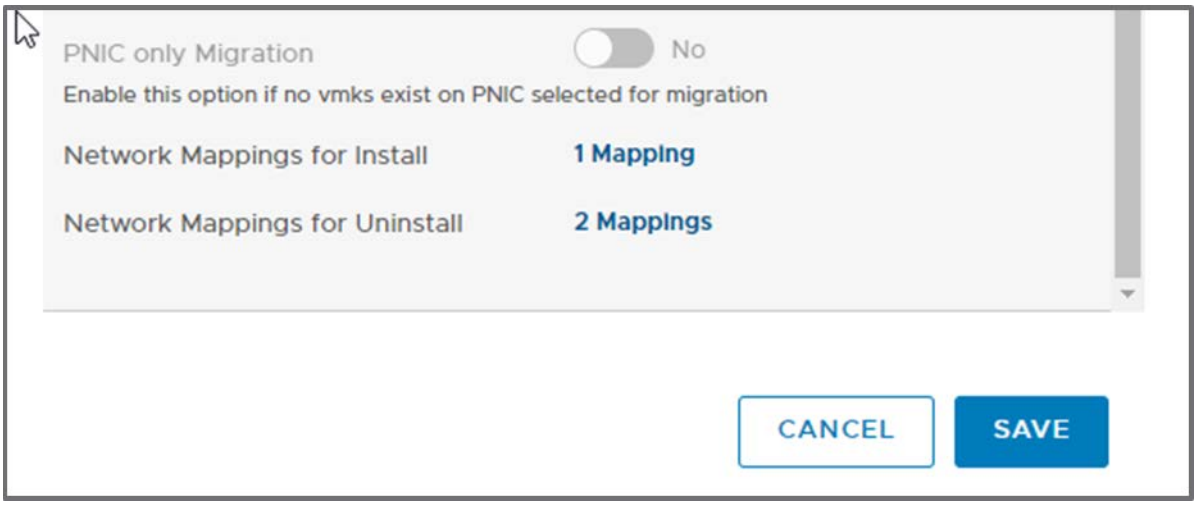


In the following dialog box, we're going to enter the vmk and vmnic mappings necessary for uninstall:

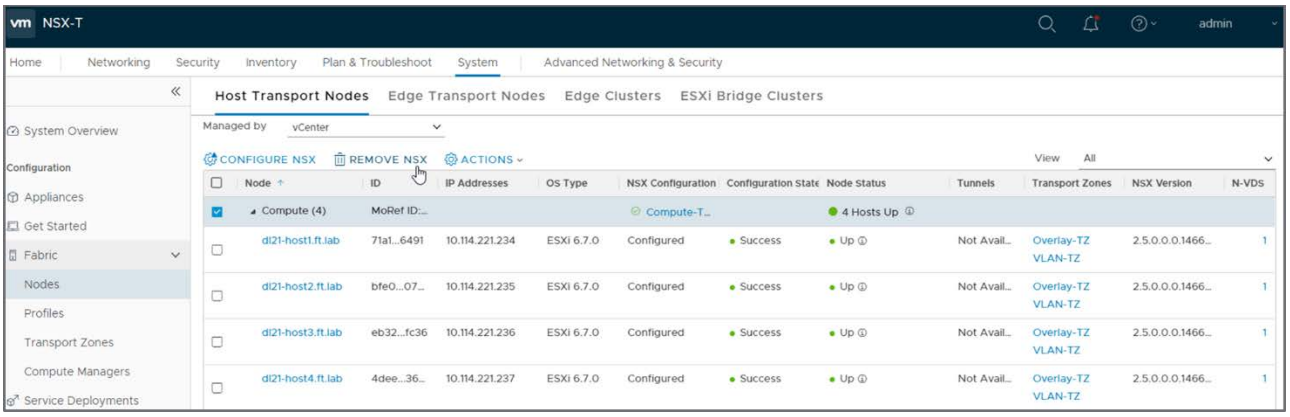


The “VMKNIc Mappings” section is configured to migrating vmk0 back to its original “Management Network” portgroup. The “Management Network” belongs to vSwitch0, the VSS on the host. The “Physical NIC Mappings” section will assign vmnic0 to that VSS. Note that I did not even bother to migrate vmnic1 to the VSS. When the N-VDS is deleted, vmnic1 will just be freed.

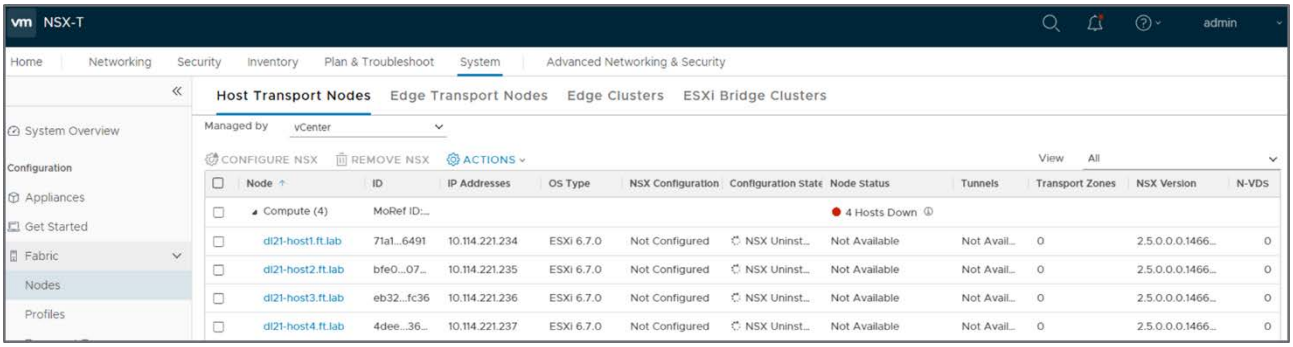
Once the mappings are defined, we can save the TNP.



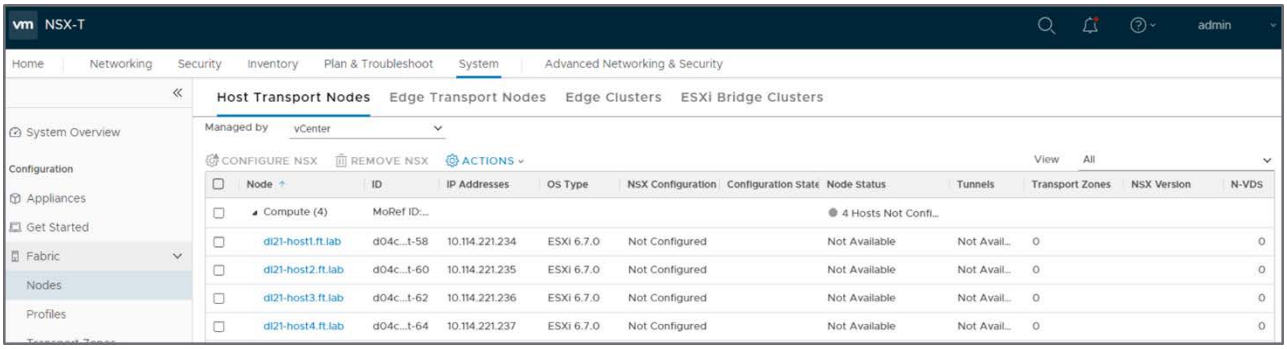
At that stage, the hosts following the TNP have the proper uninstall mappings configured, but the uninstall procedure is not started. In order to uninstall NSX from the whole cluster, we need to go back to the “Host Transport Nodes” list (in System/Fabric/Nodes).



There, we select check the box corresponding to the “Compute” cluster and click on “REMOVE NSX”. Immediately, the NSX Uninstall process is started for the whole cluster:



After few minutes, everything is cleaned up and we’re back to our initial state:



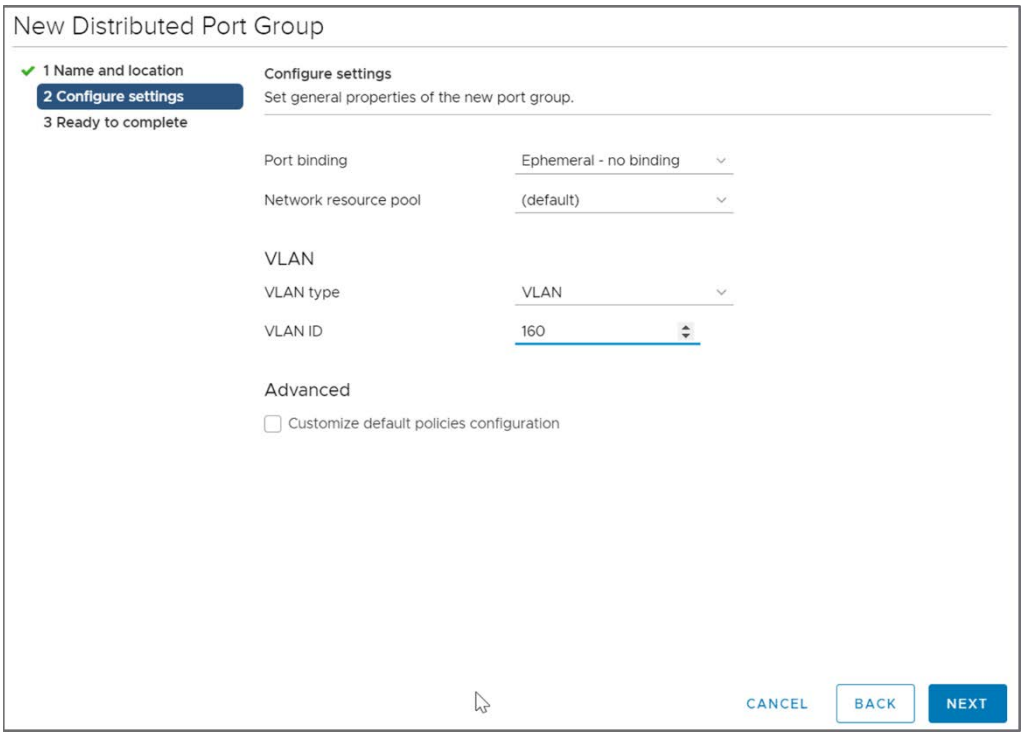
4.2 Migrating back to a VDS

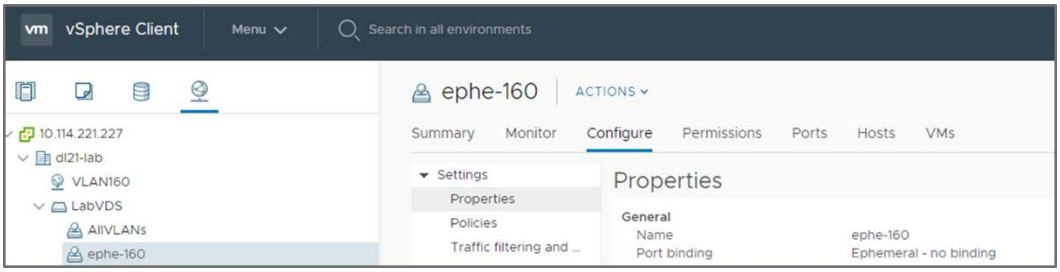
Migrating back to a VDS is pretty much the same procedure as what is documented for the VSS in the previous section. There are however two significant differences:

- vCenter will not allow you to add a host to a VDS if vmk0 is owned by the N-VDS. Apparently, the VDS migration wizard is not capable of understanding that the host gets its connectivity via the N-VDS and it prevents you from adding the VDS, fearing that the operation would make the host unreachable. What this means is that you will not be able to re-install easily a VDS if you did not have a VDS before NSX installation or if you removed the VDS from the host after NSX installation.

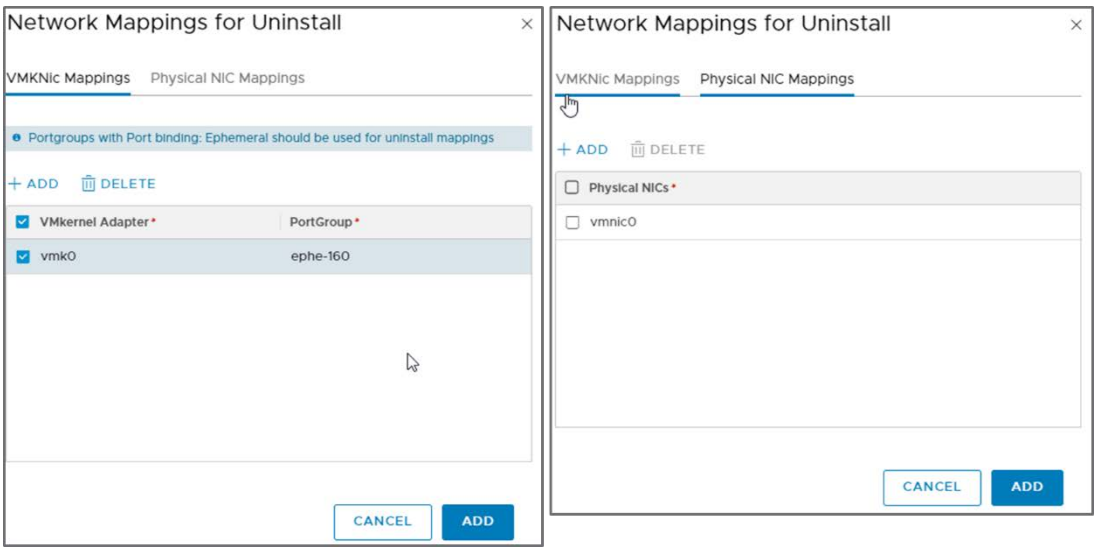
- The vmknics must be mapped to “ephemeral” dvportgroups. This is not the default kind of dvportgroup, so even if you already had some dvportgroups for your vmks, you will have to create some “ephemeral” version of them.

For the purpose of documenting a migration back to VDS, I’ve created a VDS in my lab and an ephemeral dvportgroup called “ephe-160” for vmk0:





As expected the uninstall configuration is very similar to the one we followed for the VSS:



Save the TNP and you can now uninstall NSX and move your vmk back to VDS.

Because the VDS must exist on the transport node before the uninstall (if vmk0 is attached to the N-VDS) and because you need to configure dedicated ephemeral dvportgroups, I don't think there is a lot of value in migrating back directly to a VDS. Even if you plan on running a VDS after uninstalling NSX, it looks simpler to migrate back to a VSS, then use the nice and polished wizard from vCenter to move from VSS to VDS.

When playing around with the uninstall procedure, I once hit the issue where vmnic0 associated to vmk0 was blocked by the teaming policy. This problem is not specific to NSX and can be simply avoided by only specifying vmnic0 in the Physical NIC Mappings for uninstall (of course, here I use the example of vmk0 associated to vmnic0, this could be a different combination in your environment.)

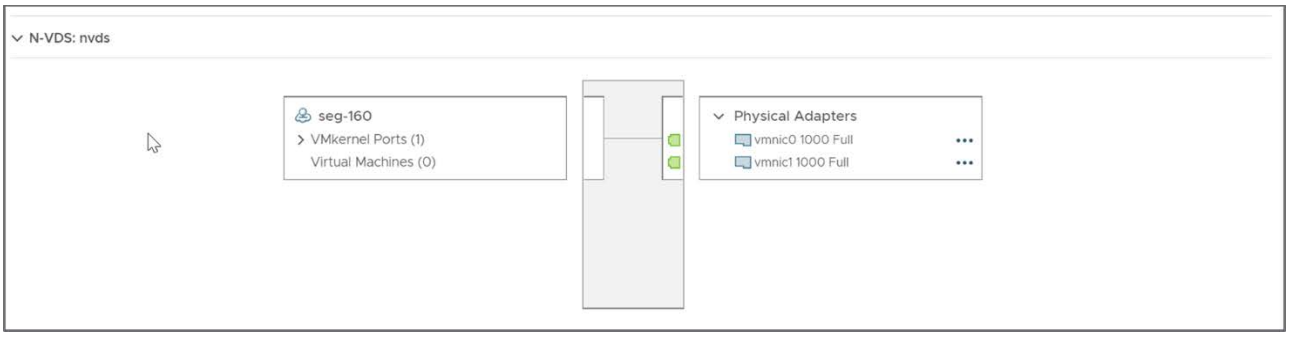
5 Recovery

If you're reading this, you have lost connectivity to your host and there is nothing you can do from vCenter or the NSX Manager UI. This part offering some options for recovering your host from the console. Two kinds of cleanups might be necessary:

1. From the ESXi console, re-assign the management interface and a physical uplink in order to recover connectivity to the host.
2. Remove a stale reference to the unreachable transport node from NSX

5.1 Reconfiguring the host from the ESXi shell

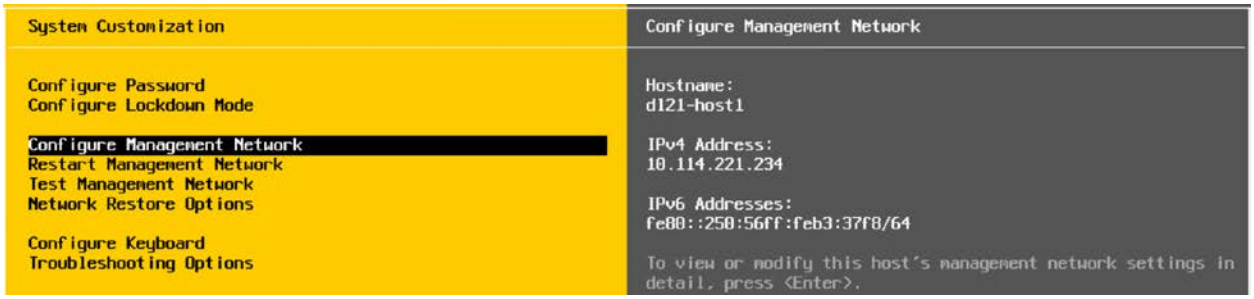
As an example, I'm going to use dl21-host1 currently successfully prepared as an NSX transport node.



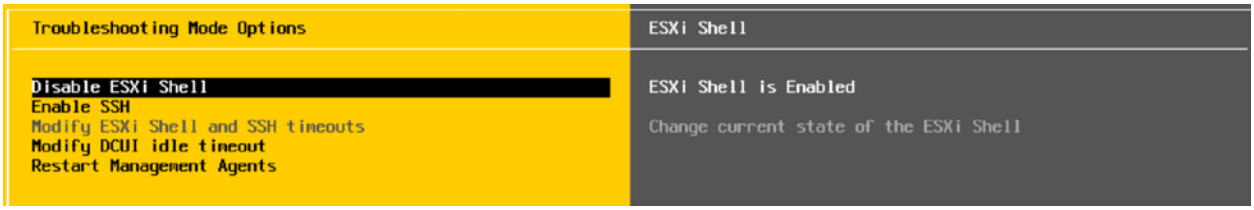
I'm going to remove vmk0 and the vmnics from the N-VDS and configure a VSS with an operational management interface from the host console. Note, because this host is perfectly healthy and accessible from NSX, the preferred way to uninstall NSX is of course using the procedure detailed in the previous parts.

5.1.1 Connecting to the ESXi Shell

Log into the Direct Console User Interface (DCUI). In this state, where the management interface is attached to NSX, there is in fact no way to reconfigure the management network from DCUI. The "Configure Management Network" option just does not work here (note, we're working on improving that.)



Select "Troubleshooting Options" then "Enable ESXi Shell"



From there, press Alt-F1 to get the login screen:

```
      ESXi 6.7.0 http://www.vmware.com
      Copyright (c) 2007-2019 VMware, Inc.

d121-host1.ft.lab login: root
Password:
The time and date of this login have been sent to the system logs.

WARNING:
   All commands run on the ESXi shell are logged and may be included in
   support bundles. Do not provide passwords directly on the command line.
   Most tools can prompt for secrets or accept them from standard input.

VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
[root@d121-host1:~]
```

We're now going to see how to move vmk and vmnic back to a VSS from the console. The following information in mainly coming from this KB article 71080 (https://kb.vmware.com/s/article/71080).

5.1.2 Removing one vmnic from the N-VDS

The first procedure I would like to demonstrate is about removing a vmnic from the N-VDS. Remember the warning I gave part 2.4.2 "Management vmkernel migration warning"? Let's assume the management interface of the host is associated to vmnic0 but that the N-VDS teaming policy has set vmnic1 as active and vmnic0 as standby. This resulted in host being unreachable. A simple way for re-gaining connectivity is to remove vmnic1 from the N-VDS so that it uses vmnic0 as active uplink.

Use the following nsxdp-cli command to get some information about the NICs attached to the N-VDS:

```
nsxdp-cli vswitch instance list
```

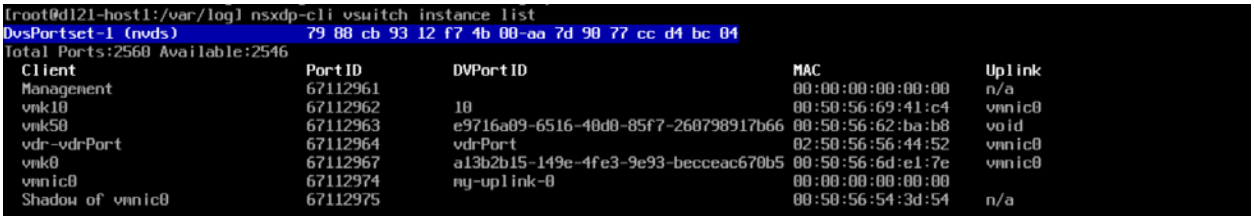
```
[root@d121-host1:/var/log] nsxdp-cli vswitch instance list
DvsPortset-1 (nvds)             79 88 cb 93 12 f7 4b 00-aa 7d 90 77 cc d4 bc 04
Total Ports:2560 Available:2545
  Client                  PortID      DVPortID                             MAC               Uplink
  Management              67112961                                         00:00:00:00:00:00 n/a
  vmk10                   67112962    10                                   00:50:56:69:41:c4 vmnic0
  vmk50                   67112963    e9716a09-6516-40d0-85f7-260798917b66 00:50:56:62:ba:b8 void
  vdr-vdrPort             67112964    vdrPort                              02:50:56:56:44:52 vmnic0
  vmnic1                  67112965    my-uplink-1                          00:00:00:00:00:00
  Shadow of vmnic1        67112966                                         00:50:56:57:17:2d n/a
  vmk0                    67112967    a13b2b15-149e-4fe3-9e93-becceac670b5 00:50:56:6d:e1:7e vmnic0
  vmnic0                  67112968    my-uplink-0                          00:00:00:00:00:00
  Shadow of vmnic0        67112969                                         00:50:56:54:3d:54 n/a
```

From the above output we can identify:

DvsPortset-1	this is the Switchname for the N-VDS (this is an ID identifying a VDS in ESXi. The N-VDS looks like a VDS here.)
nvds	this is the N-VDS name, as we have configured it in the transport zones
my-uplink-0, my-uplink-1	the name of the uplinks as defined in the uplink profile. We can also see their mapping to vmnic0 and vmnic1 respectively.

With those parameters identified, we can use the following esxcfg-vswitch command to remove vmnic1 uplink from the N-VDS.

```
esxcfg-vswitch -Q vmnic1 -V my-uplink-1 DvsPortset-1
```



You can see from the output of the second command that vmnic1 has disappeared.

5.1.3 Recreating a management interface on a VSS

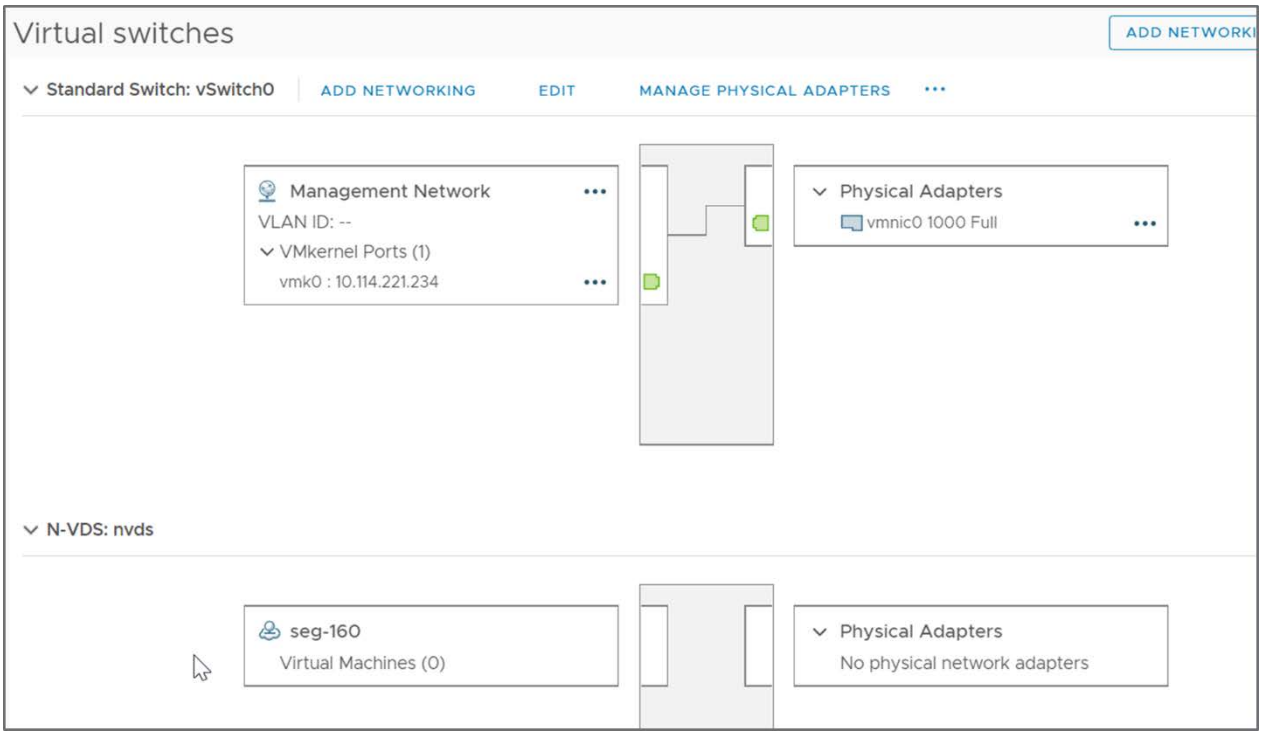
Let's move one step further in our example and remove the remaining vmnic0 from the host. Now, it is completely unreachable.

```
esxcfg-vswitch -Q vmnic0 -V my-uplink-0 DvsPortset-1
```

In order to retrieve connectivity, we're going to recreate a VSS and management port group, if they were absent. Then delete the management interface from the N-VDS and recreate it on the VSS.

`# esxcfg-vswitch -a vSwitch0`	Create a VSS called vSwitch0. This is the default VSS name, the command will just fail if there is already one such vSwitch0 existing on the host.
`# esxcfg-vswitch -L vmnic0 vSwitch0`	Adding vmnic0 to vSwitch0. You could also add vmnic1, but we just need an uplink for recovery.
`# esxcfg-vswitch -A "Management Network" vSwitch0`	Create a portgroup for management traffic.
`# esxcli network ip interface remove –interface-name=vmk0`	Delete vmk0 from the N-VDS
`# esxcfg-vmknic -a -i 10.114.221.234 -n 255.255.255.192 -p "Management Network"`	Recreate vmk0 on vSwitch0/"Management Network". Here 10.114.221.234 is the IP address of vmk0 and 255.255.255.192 is the subnet mask.
`# esxcfg-vswitch -v 160 -p "Management Network" vSwitch0`	The management network of my lab is on VLAN 160. This command sets the VLAN ID associated to the management portgroup.
`# esxcfg-route -a default 10.114.221.193`	Finally, we add a default gateway to the default IP stack where vmk0 is defined.

At that stage, we have recovered connectivity to the outside world through vSwitch0.



5.1.4 Cleaning up NSX bits from the host

A single nsxcli command deletes all NSX components on the host:

```
nsxcli -c del nsx
```

The command can only be executed once because it deletes itself from the host.

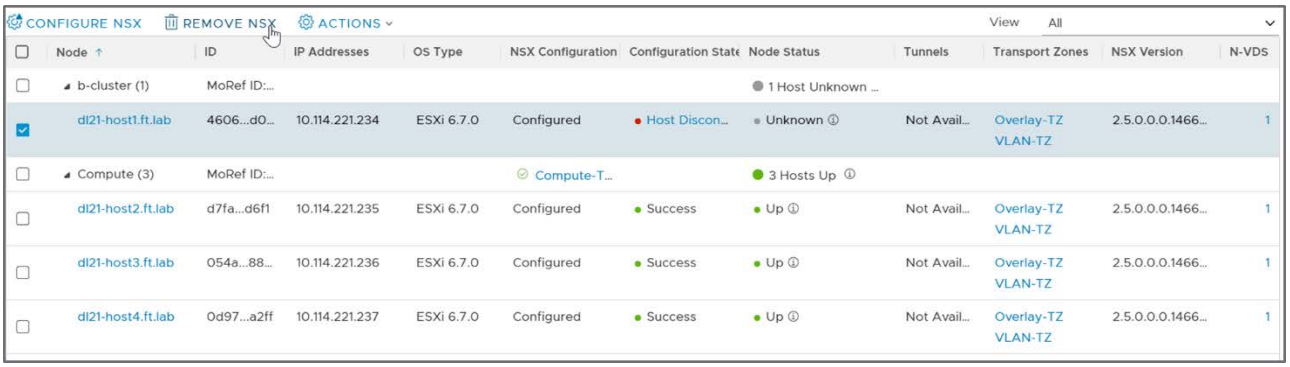
```
[root@dl21-host1:~] nsxcli -c del nsx
Terminated
```

```
[root@dl21-host1:~] nsxcli -c del nsx
-sh: nsxcli: not found
```

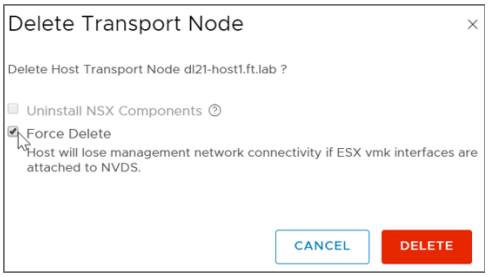
5.2 Removing stale transport nodes from NSX using the UI

Suppose that the host is not reachable any more and that it had to be reset to a simple VSS configuration, as shown in the previous part. This host might still be showing up as an unreachable transport node in NSX.

In order remove it from NSX, select the transport node from the list and click on “REMOVE NSX”. If your host is part of a cluster to which a transport node profile is attached, the option to remove NSX on a single transport node will not appear. A simple trick to work this around is to create a dummy cluster and move the host to it. For example, here, I’ve created “b-cluster” and moved dl21-host1 to it from vCenter. Now, the option to remove NSX is available:



The “Delete Transport Node” dialog box appear:

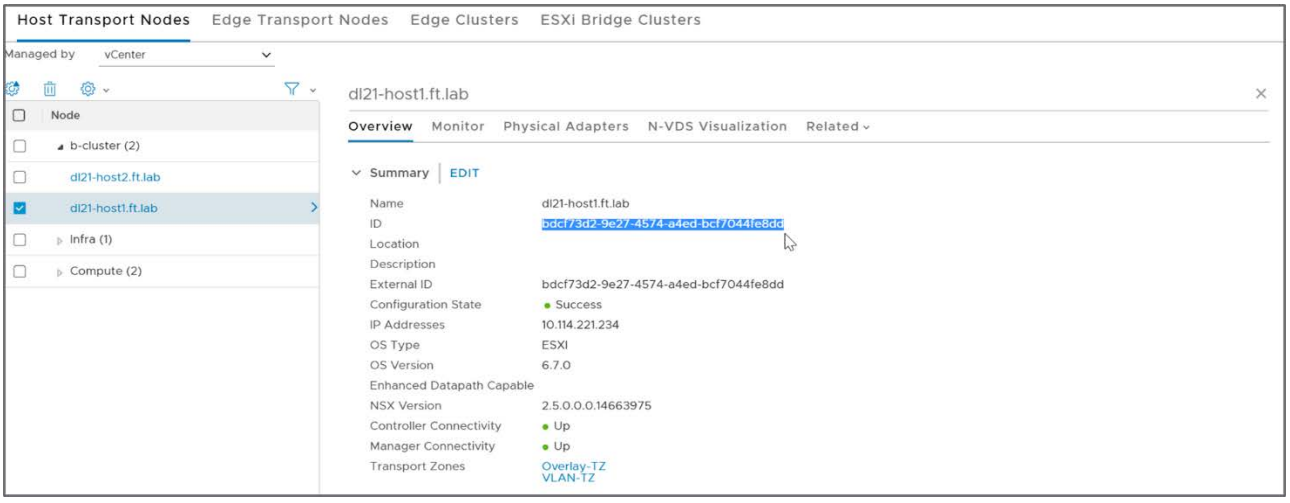


In this scenario, I’ve assumed that we had lost connectivity to this dl21-host1. The option “Uninstall NSX Components”, which requires connectivity to the host is thus unavailable. Check “Force Delete” to remove this transport node from NSX.

Note that it is possible that in order to be able to delete the transport node using the above procedure, you first need to try to “Resolve” the error showing up in the “Configuration Status” column. Just click on the error message to get this “Resolve” option.

5.3 Removing stale transport nodes from NSX using the API

At last, I need to mention the API-based nuclear option for deleting a transport node. Retrieve the ID of the host:

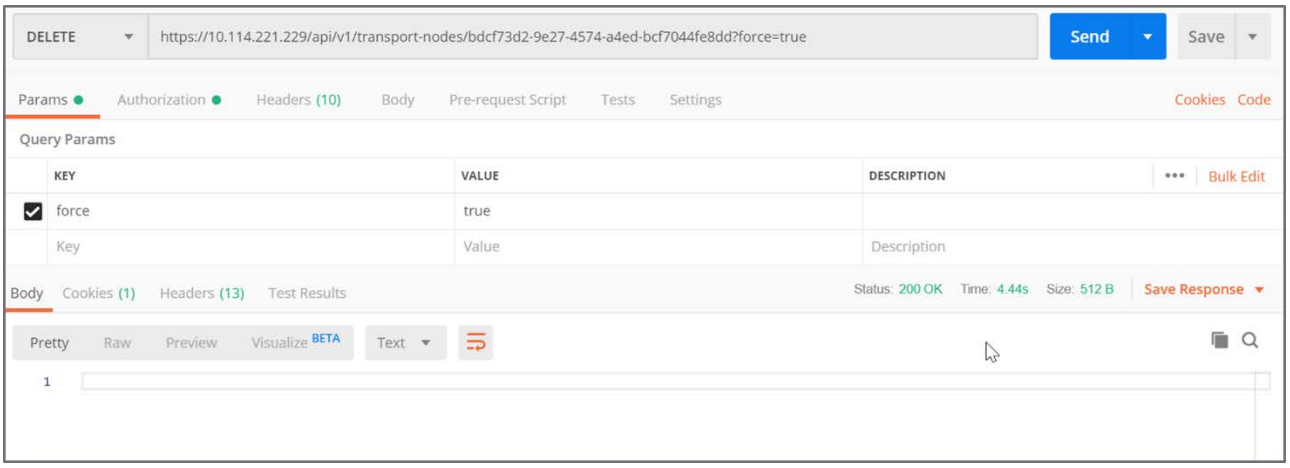


Then, execute the following DELETE API call:

```
https://<nsx-manager-ip>/api/v1/transport-nodes/<transport-node-id>?force=true&unprepare_host=false
```

where

- `nsx-manager-ip` is the FQDN or IP address of the NSX manager appliance
- `transport-node-id` is the ID retrieved in the previous step.



This will delete the transport node, even if the cluster is part of a TNP.