NetVisor OS
Virtual Netvisor Deployment Guide

Arista Networks

www.arista.com

NetVisor OS Virtual Netvisor Deployment Guide, version 2022.7.0.2
DOC-05951-01

Headquarters	Support	Sales
5453 Great America Parkway		
Santa Clara, CA 95054	+1-408 547-5502	+1-408 547-5501
USA	+1-866 476-0000	+1-866 497-0000
+1-408-547-5500	**support@arista.com**	**sales@arista.com**
www.arista.com		

Table of Contents

Summary

The Arista Networks' Unified Cloud Fabric provides a fully advanced networking solution with a complete Layer 2-3 feature set which can fit in many environments such as Enterprise Data Centers or Small to Large Cloud Providers. Based on a distributed management and control plane, the fabric offers a very flexible solution for scale-out IP designs as well as for secured multitenant architectures.

With the new era of Hybrid Cloud, operators must adapt and find clever solutions to offer more services, with embedded security while minimizing the total cost of ownership and operations. Arista Networks addresses this requirement with the introduction of Virtual Networks (vNET), a logical construct in Netvisor, which slices the Fabric in multiple zones or virtual PODs (vPOD). A vNET is defined by a set of resources from the physical layer (ports) up to the management plane, with delegated administration to different vNET administrators.

The distributed nature of the Arista Networks fabric makes it unique and easy to scale with embedded network services. For some services which require more CPU resources than forwarding capacity, it is more efficient to centralize the service in a fabric node which has a larger CPU/Memory configuration instead of consuming resources on a white box switch. And, considering the scale factor for some use cases, creating and managing tenants is CPU intensive, hence offloading this task to a compute node part of the fabric is the best design.

To accommodate this increased demand of multitenant infrastructures, Arista Networks uses Virtual Netvisor, a Virtual Machine running Netvisor and joining the fabric, to optimize operations that require more compute power in the fabric.

This document describes Virtual Netvisor and how to use it in a multitenant infrastructure or a secured network with several security zones which must be implemented and spread across the network.

Overview

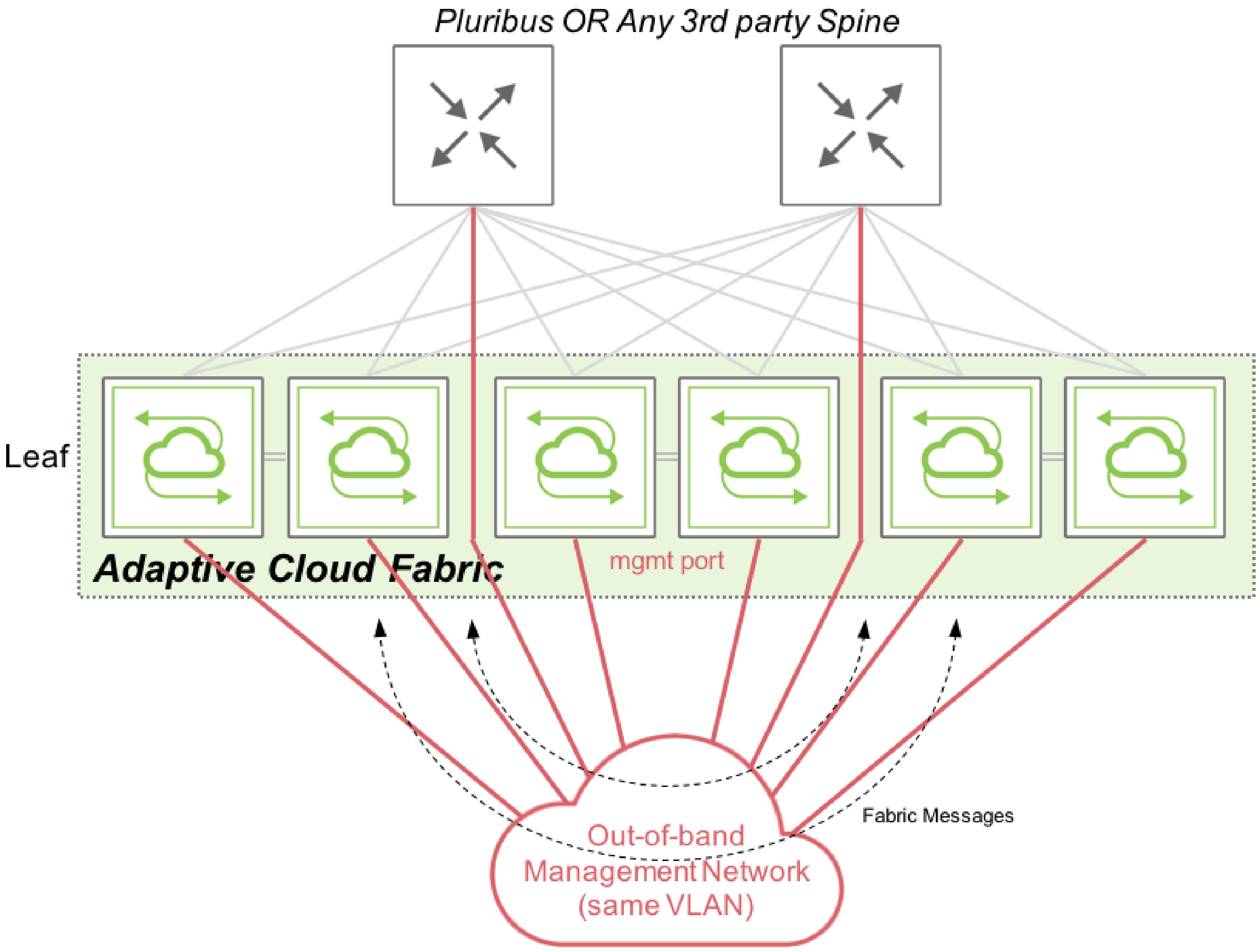
Arista Networks Unified Cloud Fabric Overview

Arista Networks’ Unified Cloud Fabric is deployed in two different ways, using the out-of-band management network or using the in-band network (bridged or routed network). Both options have pros and cons, and the decision to deploy one model or another really depends on the network constraints and the use case.

The following section describes the two typical fabric architectures you can create, with or without Virtual Netvisor deployed and added to the fabric.

Management Fabric

A Management fabric uses the out-of-band management network to send fabric messages and form a fabric. The following diagram shows the route used by fabric messages in a management fabric using out-of-band ports on Arista Networks switches:



Management Fabric Communications

Overview (cont'd)

Netvisor sends and receives fabric messages, essentially keepalive and configuration messages, over the out-of-band network which are consumed by other fabric nodes. These messages ensure each and every node within a fabric, along with other fabric nodes, can initiate/receive a configuration instruction to/from other nodes.

Inband Fabric

An in-band fabric uses the in-band network to send fabric messages and form the fabric. Netvisor provides two options for the in-band network forming the network:

- The underlay network is a Layer 2 network (using Arista Networks clusters and VLAGs for redundancy):

A VLAN to transport fabric messages across the L2 network, similarly to what is performed with a management fabric but using an in-band VLAN (VLAN 1 being the default VLAN used for the fabric).

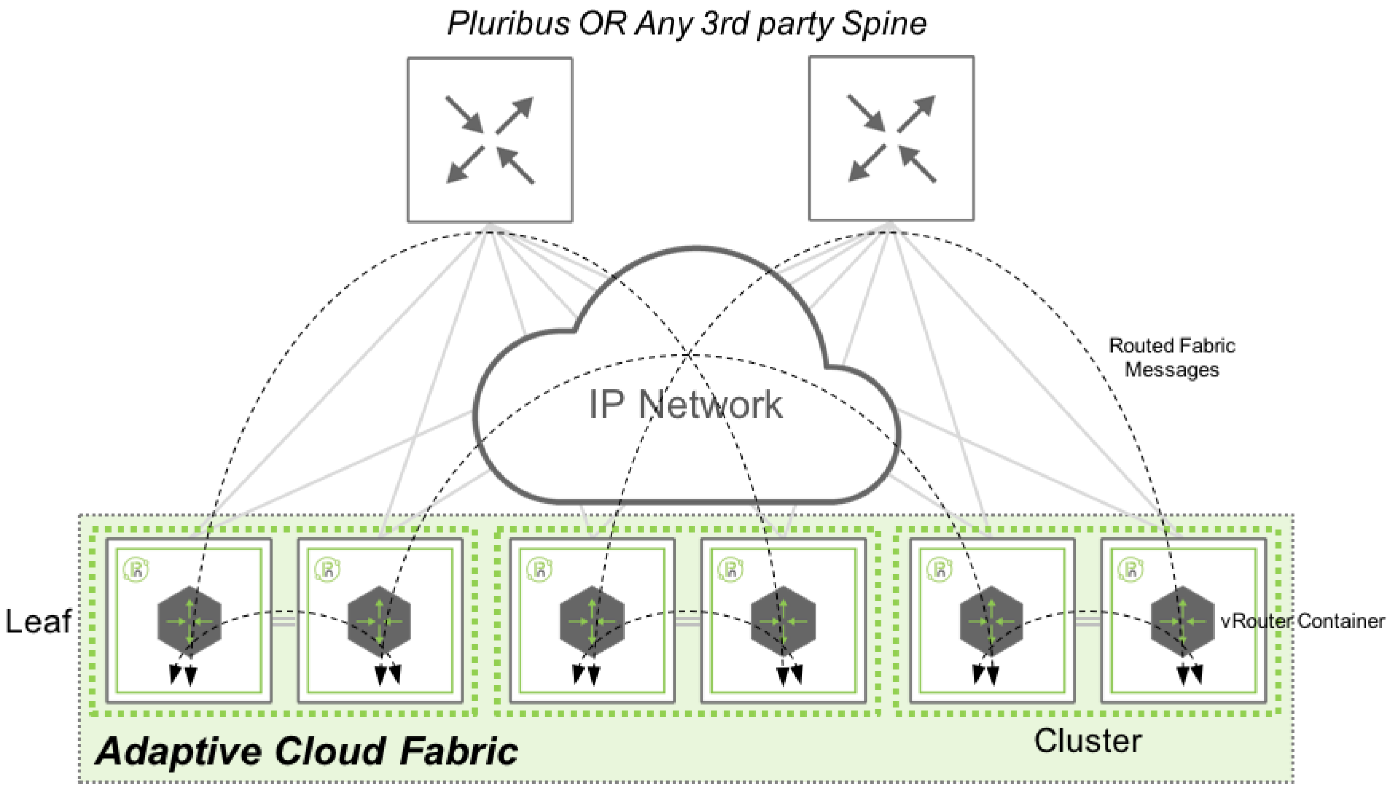
- The underlay network is a Layer 3 network (using IGP and ECMP for redundancy):

The local in-band (fabric) subnet, unique per fabric node, advertises to other nodes using the underlay routing protocol. Once IP connectivity is established between Arista Networks switches, they can join the same fabric and exchange fabric messages over this routed network.

> **Note:** Although an in-band fabric is supported using a Layer 2 or Layer 3 underlay network, Arista Networks strongly recommends implementing a Fabric over a Layer 3 network when configuring an in-band fabric, as Netvisor extends the fabric to a remote site.

Overview (cont'd)

The following diagram shows the path taken by fabric messages in an in-band fabric using a Layer 3 underlay network on Arista Networks switches:



In-band Fabric Communications with Layer 3 Underlay Network

Netvisor sends and receives fabric message over in-band network, by using the local vRouter and consumed by other fabric nodes. In this architecture, Fabric messages are routed from one switch to another using the IGP (BGP, OSPF or Static) configured by the network administrator. This design gives the flexibility to extend a fabric from a local site to geographically dispersed locations, as it relies on standard routing protocols with no specific requirements on latency, jitter or MTU.

Virtual Netvisor (vNV) Overview

Netvisor provides Virtual Netvisor as a virtual edition of Arista Networks Netvisor and provides a partial set of features like a regular physical switch.

The form factor of Virtual Netvisor is a Virtual Machine running a base Linux OS (Ubuntu) and Arista Networks Netvisor on top to bring a data center class network feature set.

Another Fabric Node

Virtual Netvisor is running Netvisor like a regular node, consequently, it is another fabric node participating in the distributed management and control plane. Despite a limited set of functionalities available in Virtual Netvisor (no L2 functions for instance), we use this node as part of an existing node composed of physical switches to provide a powerful compute fabric node for compute intensive control plane activities.

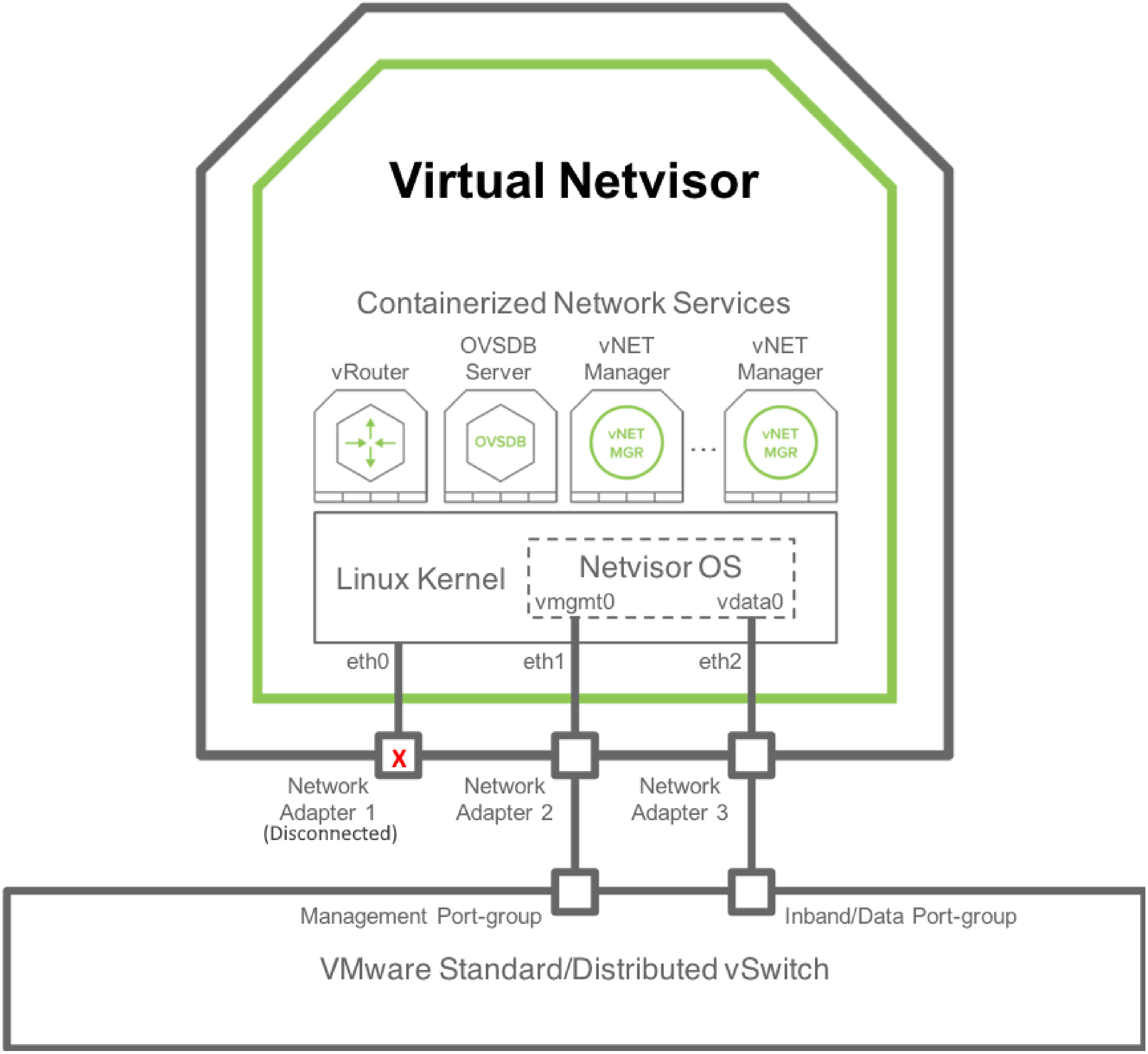
Creating **Containerized Network Functions** such as vNET Manager or OVSDB Server which consume a tremendous amount of CPU/Memory resources make **vNV** ideal when used in a pure compute instance.

When Virtual Netvisor joins an existing fabric with physical nodes, it receives and processes all fabric transactions like a regular node.

Overview (cont'd)

Software Architecture

The following figure shows an overview of vNV software architecture:



vNV Software Architecture

Overview (cont'd)

vNV is a Virtual Machine with the following components:

- Linux OS (current 5.1.0 release uses Ubuntu 16.04 with a Linux Kernel 4.15).
- 3 Kernel network interfaces:
    - eth0: not used. This is “Network Adapter 1” in the VM configuration in VMware vSphere Web Client.
    - eth1: used by Netvisor for the management port/interface. This is “Network Adapter 2” in the VM configuration in VMware vSphere Web Client.
    - eth2: used by Netvisor for the data/in-band port/interface. This is “Network Adapter 3” in the VM configuration in VMware vSphere Web Client.
- 2 OVS bridges (not represented in this diagram) connecting Netvisor to the Linux Kernel (see OpenVSwitch documentation for more details):
    - Management bridge (called nvm-mgmt-br).
    - In-band bridge (called nvm-data-br).
- Netvisor running as a daemon in the Host Kernel space (nvOSd) and using the following interfaces:
    - vmgmt0: connected through Management OVS bridge to eth1 interface in the Linux Kernel. This is the Netvisor Management port/interface seen in the CLI/API.
    - vdata0: connected through in-band OVS bridge to eth2 interface in the Linux Kernel. This is the Netvisor In-band/Data port/interface seen in the CLI/API.
- Linux Containers to host Network services such as:
    - A vRouter: provides L3 connectivity and dynamic L3 control plane (for dynamic routing).
    - An OVSDB Server: exposes an OVSDB interface to 3rd party SDN controller.
    - vNET Managers: provides a fully isolated management view for each tenant admin in the context of a multitenancy fabric architecture.

Joining a Fabric with vNV

Virtual Netvisor (vNV) is a regular Netvisor node and hence requires to be connected to the same management or in-band network to join an existing fabric.

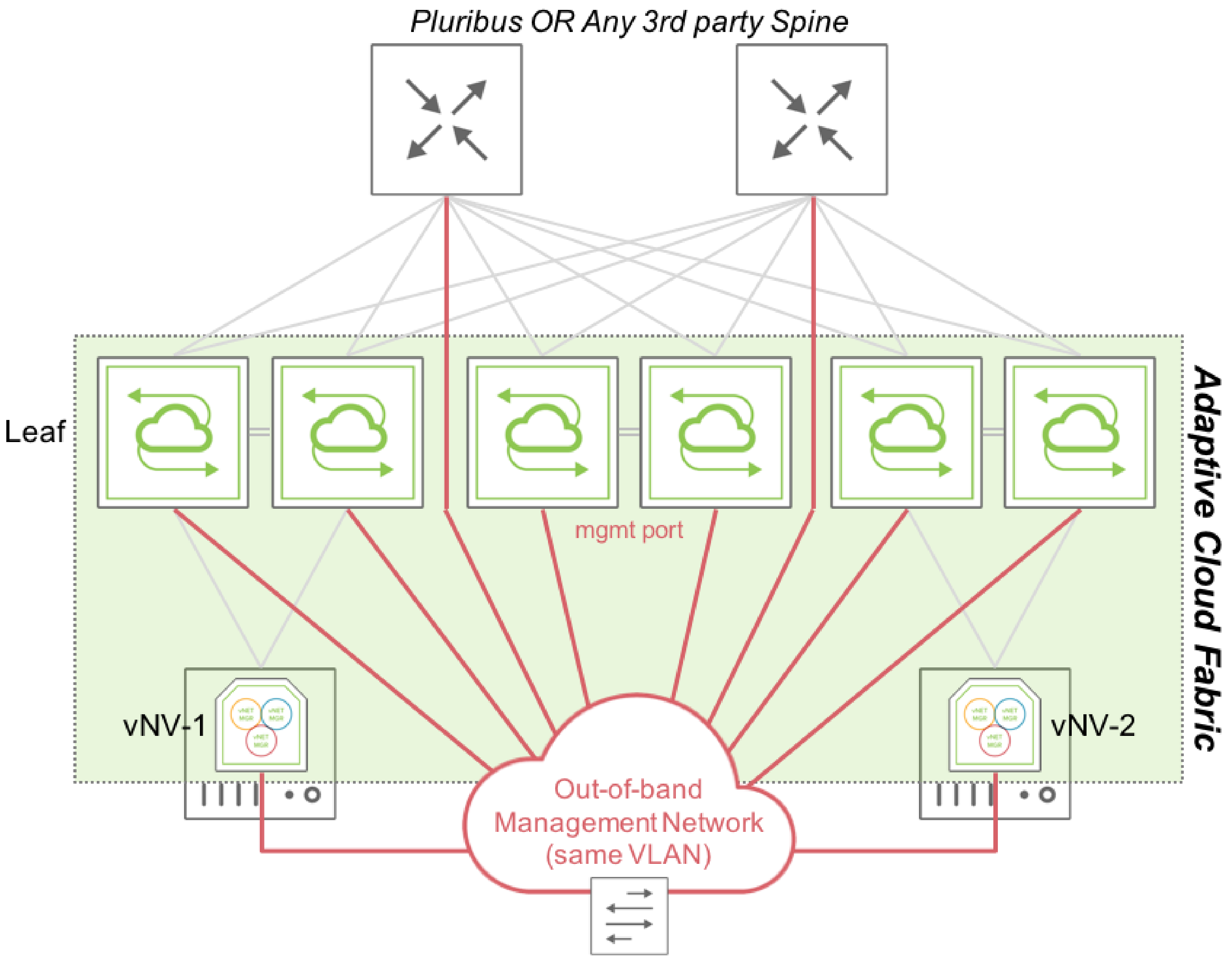
As vNV is a Virtual Machine (VM), the way you can connect it to the network is slightly different from a physical switch.

This section describes the supported topologies to deploy vNV and join an existing fabric of physical switches.

Management Fabric

Flat Management Domain

Using the management network, vNV is installed and deployed in the following topology:


Pluribus OR Any 3rd party Spine
Leaf
Adaptive Cloud Fabric
mgmt port
vNV-1
vNV-2
Out-of-band Management Network (same VLAN)

Fabric Nodes in a 'Flat" Management Network

Overview (cont'd)

In this topology, two vNVs deployed in two VMware ESXi servers are connected to two different clusters of Arista Networks switches or to the same cluster.

vNV-1 and vNV-2 use only their Management interface to communicate with the rest of the fabric, as they are not expected to forward any in-band traffic. Instead, they provide more power to the fabric and perform compute intensive tasks which some white box switches cannot handle without degrading performances in the network.

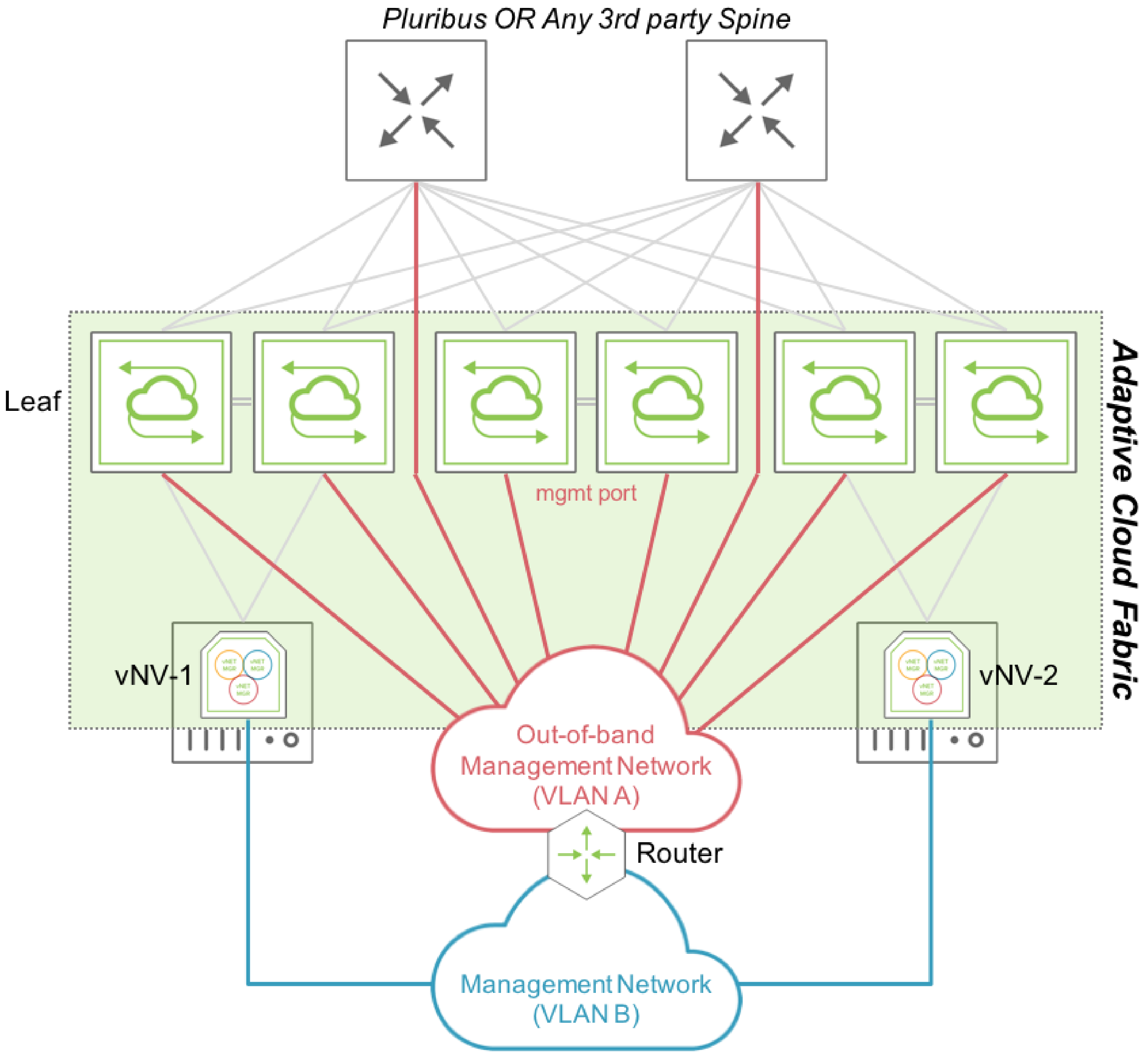
Routed Management Domain

In certain cases, it might be complicated to connect ESXi hypervisors in the out-of-band network directly (in the same VLAN or bridge domain):

- Servers with a limited number of interfaces.
- Physical restrictions.
- Security restrictions.

In this case, create a Management fabric using an IP network to connect fabric nodes together. The only requirement is IP reachability between fabric nodes.

The following diagram is an example of a Management fabric where physical switches are connected to the Red Management network and ESXi servers are connected to another Blue Management network. If these two networks are routed between each other, vNVs join the existing Management fabric formed by physical switches.

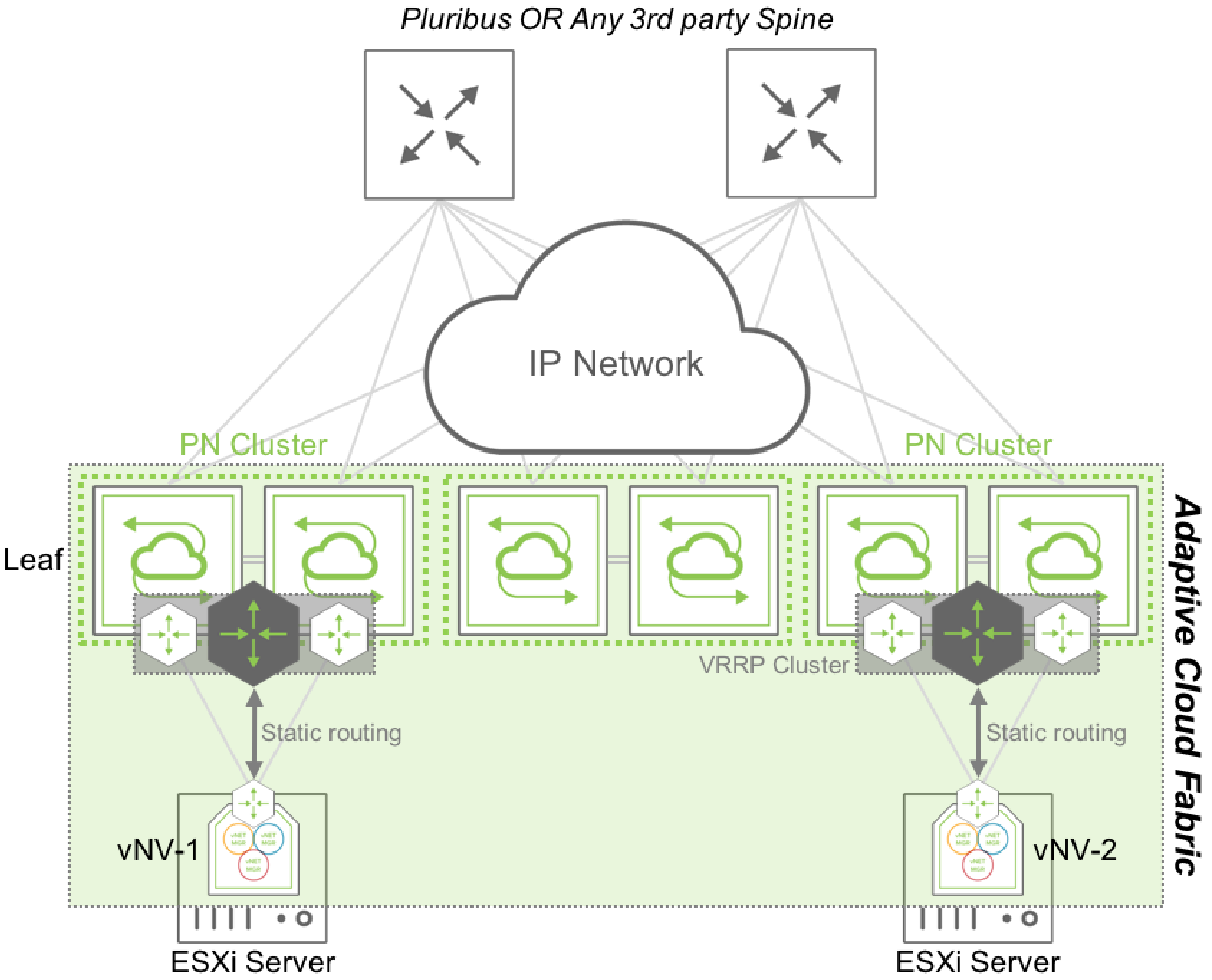

Pluribus OR Any 3rd party Spine
Leaf
Adaptive Cloud Fabric
mgmt port
vNV-1
vNV-2
Out-of-band Management Network (VLAN A)
Router
Management Network (VLAN B)

Fabric Nodes in a Routed Management Network

In this topology, vNV-1 and vNV-2 use only their Management interface to communicate with the rest of the fabric and fabric messages are routed to communicate with other nodes part of the same fabric, as opposed to the previous topology where all nodes were sitting in the same broadcast domain.

In-band Fabric Using VRRP

Using the in-band network with a Layer 3 underlay network, vNV is installed and deployed in the following topology:



Fabric Nodes in a Routed In-band Network using VRRP

In this topology, two vNVs are deployed in two VMware ESXi servers, connected to two different clusters of Arista Networks switches or to the same cluster.

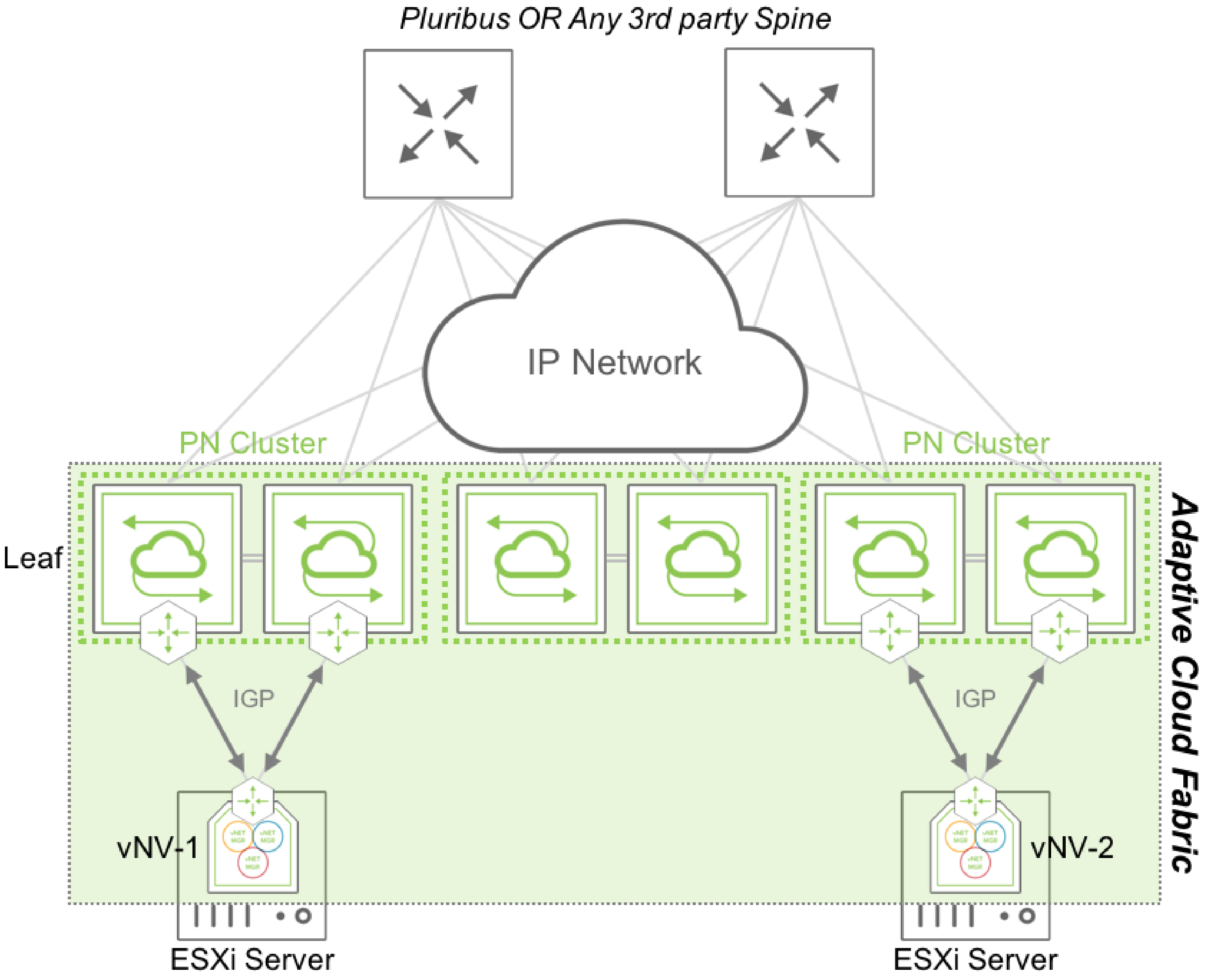
vNV-1 and vNV-2 use their in-band interface to communicate with the rest of the fabric and use the top of the rack (ToR) switches to communicate with other nodes. The two ToR switches are configured in a VRRP cluster and redistribute this local subnet in the IGP to advertise it to other fabric nodes. Each vNV have a default route pointing to this VRRP Virtual IP (VIP) and communicate with any other nodes' in-band interface in the network. This option is the easy way to setup a hybrid fabric (physical + virtual) leveraging the underlay Layer 3 network.

In-band Fabric Using IGP

In certain cases, it is useful to run an IGP between vNV instances and the fabric nodes to provide a fully dynamic layer 3 architecture without relying on First Hop Redundancy Protocols (like VRRP, HSRP, GLBP, etc.).

Although there are no restrictions on where to run the IGP between vNVs and the physical switches, Arista Networks recommends running this IGP at the ToR. When vNV is running on ESXi servers and these servers are directly connected to the ToR switches to avoid implementing multi-hop routing protocols between vNVs and the Spine switches, Arista Networks recommends creating a L3 IGP adjacency with the first ToR switches.

Following that recommendation, vNV is installed and deployed in the following topology:



Fabric Nodes in a Routed In-band Network using an IGP

Overview (cont'd)

In this topology, two vNVs are deployed in two VMware ESXi servers, connected to two different clusters of Arista Networks switches or to the same cluster.

vNV-1 and vNV-2 use their in-band interface to communicate with the rest of the fabric and use the top of the rack (ToR) switches to communicate with other nodes.

The two ToR switches are configured to form an IGP adjacency with the local vNV, with the same configuration which would be applied for any other L3 node in the fabric (there’s no difference because it is a Virtual Machine). The two vNV instances are configured with two different VLANs to connect to the ToR switches (one for each) and form an OSPF adjacency or BGP peering with the local ToR switches.

A simple network advertisement in the routing protocol (or connected routes redistribution) advertises vNVs’ local in-band interface to the rest of the network, ensuring fabric messages are delivered across the L3 network to other nodes.

Virtual Network (vNET) Overview

A Virtual POD (vPOD) is a logical construct in a shared infrastructure where a tenant can only see and use resources which are allocated to this tenant. A vPOD is composed of compute, storage and network resources and is created at the orchestration layer.

In a Arista Networks’ Unified Cloud Fabric, a vPOD translates to a Virtual Network (vNET), a collection of isolated network resources and services (defined by the Fabric administrator) associated with an independent management domain, allowing each vNET administrator to manage his/her own zone/environment.

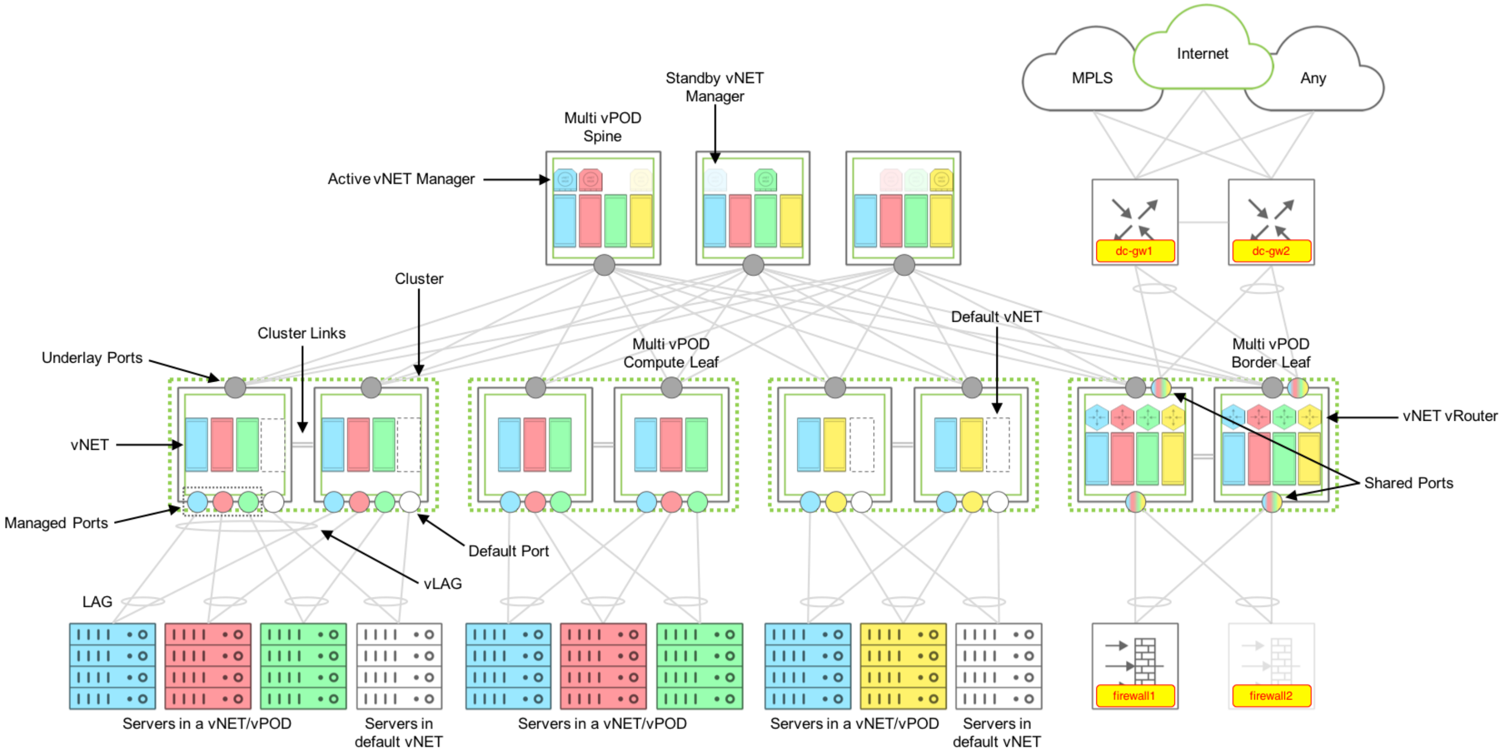
Virtual Networks (vNETs) is Arista Networks’ advanced solution to dealing with multi-tenancy requirements in a secure and versatile manner that goes beyond basic VLAN and VRF standards. In basic terms vNETs are separate resource management spaces and operate in the data plane as well as in the control plane.

vNET Architecture

By creating a vNET in the Unified Cloud Fabric, the network resources are partitioned and dedicated to a single tenant with fully delegated management capabilities (optional). This partition of the fabric is configured either on one switch, a cluster of two switches, or the entire fabric by leveraging the “scope”, depending on where this vNET is required to deliver an isolated service to a tenant.

The figure below shows how a Multitenant leaf-spine design is achieved with a Arista Networks’ Unified Cloud Fabric:

Overview (cont'd)



Multitenant Fabric Design Components

Components

A multitenant fabric is built with the following components:

Default vNET

The Default vNET is the default partition on a switch used when the switch has not been configured with additional vNETs. This default partition “owns” all the resources on a Netvisor switch:

- Physical Ports.
- Default VLAN space.
- Default Routing table/domain (if enabled on the switch).
- The VTEP function.
- Any L2/L3 entries learned through static and dynamic protocols.
- Any additional container created.

The default vNET is not configured by the fabric administrator, it is a partition created as soon as the switch becomes part of a fabric. There is no vNET Manager container associated to the default vNET (see below for the description of a vNET Manager).

The default vNET has the following naming convention: **[Fabric Name]-global**.

Tenant vNET

A Tenant vNET is a partition which carves out the software and the hardware resources from the switch such as:

- VLAN IDs.
- MAC addresses.
- vRouter or VRFs.
- Remote VTEPs (Virtual Tunnel End Points).
- Physical ports.

A vNET is created with different scopes:

- Local: locally created on the switch and not usable by any other switch in the same fabric.
- Cluster: created on a cluster pair which share the same Layer 2 domain.
- Fabric: partition created on all switches in the same fabric.

Once a vNET is created, resources are associated to it when new objects are created in the fabric (VLANs, vRouters, etc.).

Although a vNET could be compared to a VRF, it goes beyond the VRF capabilities as it not only provides an isolated L3 domain, but it also offers the capability to have VLAN and MAC overlapping between vNETs. In other words, it is a complete isolation of the data plane and control plane, along with the management plane if there is a vNET Manager associated to it (see below for the description of a vNET Manager).

Each vNET has a single point of management. As the fabric administrator, one can create vNET and assign ownership of each vNET to individuals with responsibility for managing those resources and with separate usernames and passwords for each vNET managed.

There are two types of vNETs a Fabric Administrator can create, a Public vNET and a Private vNET:

Public vNET

A Public vNET is using the default 4K VLAN space available on a switch. In this case, the default vNET and any other vNET created share the 4094 VLAN IDs available on the platform with no option to have overlapping between VLAN IDs across vNETs.

By default, a vNET is created as Public vNET, although with Overlay networks and Multitenancy requirements, it makes more sense now to use Private vNETs almost exclusively.

Private vNET

A Private vNET is using an independent 4K VLAN space, where the vNET Administrator can create any VLAN ID he wants, regardless of whether that ID is used in the default vNET or in another Tenant vNET.

As the hardware architecture on Open Networking platforms does not offer unlimited resources, tables between the default vNET and any other vNET created still must be shared. Arista Networks provides a way to limit the amount of VLANs which are created in a Private vNET to avoid a situation where a tenant creating too many VLANs prevents other tenants to create VLANs as there are no hardware resources available on the switches.

To create this independent 4K VLAN ID space for each tenant, a mapping between two different types of VLANs is required, Public VLANs and Private VLANs, called vNET VLANs.

vNET VLANS

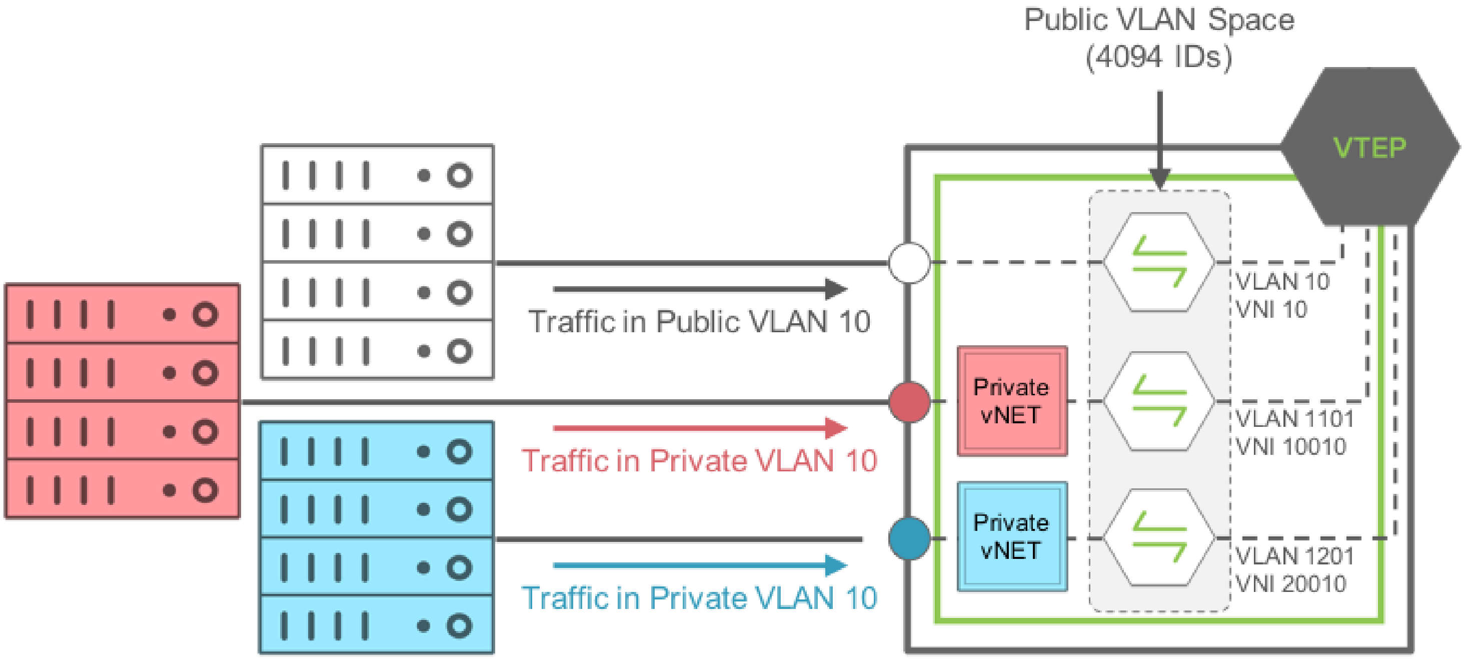
Public VLAN

A Public VLAN is a VLAN which uses the default VLAN ID space on a switch. When a Fabric administrator creates a VLAN on any Netvisor switch, without any vNET implemented (only the default vNET), it consumes VLAN IDs in the Public VLAN space.

Private VLAN

A Private VLAN is a user-defined VLAN ID in a private space, which has nothing to do with the actual VLAN ID (in the public space) used on the switch to learn L2 entries for this bridge domain. When the switch receives a frame with an 802.1q tag using the Private VLAN ID, it translates this VLAN ID to an internal VLAN ID in the 4K Public VLAN space.

When a Private VLAN must be transported over a VXLAN infrastructure, it requires a VNI to be associated to it so it is identified as unique in the shared (or public) Layer 2 domain. The following diagram depicts the relationship between Public and Private VLANs in a Private vNET implementation:



Private VLAN-to-Public VLAN Mapping with Private vNETs

In this example, traffic coming on a Managed port from the RED Private vNET, using Private VLAN ID 10, is mapped internally to the Public VLAN ID 1101.

The same thing happens to traffic coming on a Managed port in the BLUE Private vNET, using Private VLAN ID 10 as well, it is mapped internally to the Public VLAN ID 1201. The mapping is done automatically in Netvisor by defining a pool of usable Public VLAN IDs for Private vNETs.

> **Note:** Traffic from the RED Private VLAN 10 and BLUE Private VLAN 10 are**NOT** bridged at all as they are not in the same bridge domain internally.

Finally, traffic coming on a port on the default vNET, using VLAN 10 (so Public VLAN 10), is mapped internally to the same Public VLAN 10.

vNET Ports

Along with the VLAN and MAC table isolation, a complete isolation for each tenant at the physical port level is also required. To achieve this isolation at the access ports, the Unified Cloud Fabric introduces different types of ports to address different scenarios in a shared network.

Managed Port

A Managed port is an access port which is associated to one tenant only. This port is dedicated to a vNET and cannot be shared with other vNETs configured on the same switch. It is typically used to connect, and isolate resources dedicated to a tenant (like servers or any resource which is not an active L2 node).

> **Note:** As an access port, a Managed port does not support Spanning-Tree. L2 loop prevention mechanism is implemented through another software feature called Block-Loops.

A Managed port is configured as an “access port” or a “Dot1q Trunk port”, if the VLANs carried on the port are part of the same Private vNET. It transports only Private VLANs belonging to the same vNET. It also supports Link Aggregation (802.3ad) on a single switch or with two different switches part of the same cluster (called Virtual LAG (vLAG) with Arista Networks switches).

Shared Port

A Shared port is an access port which can serve different tenants on the same port where a shared resource is connected. This port is not dedicated to a vNET and is shared with other vNETs configured on the same switch. It is typically used to connect shared resources like Gateways, Firewalls or even hypervisors hosting several VMs servers from different tenants.

A Shared port is always configured as a “Dot1q Trunk port” transporting Public VLANs only, from the same vNET or different vNETs on the same port. It also supports Link Aggregation (802.3ad) on a single switch or with two different switches part of the same cluster (called Virtual LAG (vLAG) with Arista Networks switches).

Underlay Port

In a vNET design, the underlay network topology is shared by all vNETs configured in the fabric. To avoid dedicating one Underlay port per tenant to connect Leaf switches to Spine switches, Arista Networks has abstracted Underlay ports from the vNET to leave their management to the Fabric administrator.

Underlay ports are the ports which interconnect Spine and Leaf switches or two switches part of the same availability Cluster. These ports are not configurable from the Private vNETs and are configured automatically based on the underlay design defined by the Fabric administrator.

vNET Manager

A vNET Manager is a container created on a fabric node and associated to a vNET to provide a dedicated management portal for the vNET. A vNET Manager is a completely stateless object, but it is deployed in high-availability mode to ensure the tenant never loses management access to the vNET. A vNET Manager in high-availability mode can have two different states:

- Active vNET Manager: elected container to manage resources allocated to a vNET. This container is the one used by the vNET Administrator to configure resources via CLI or API allocated by the Fabric administrator to this vNET.
- Standby vNET Manager: container in standby mode, sharing the same IP address with the Active vNET Manager container to offer high availability when the Active container fails or becomes inaccessible.

A vNET Manager is not mandatory to create a vNET, it is just providing a dedicated management portal to the tenant.

A vNET Manager container is an object local to the switch where it is created. In other words when the Fabric Administrator creates a vNET with a fabric scope, the vNET Manager associated to this vNET, is only created on the switch where the command is instantiated.

A vNET Manager is created on any switch in a fabric, if local resources (compute, memory and storage) are available.

vNET vRouter

A vNET vRouter is a network service container which provides L3 services with dynamic routing control plane to a vNET. A vNET is created:

- Without a vRouter container (L2-only vNET).
- With one single vRouter (L3 vNET).
- With more than one vRouter (Multi-VRF model in a vNET).

The first vRouter associated to a vNET builds the vNET (default) routing table. In some designs, a vNET can require more than one vRouter to support multiple routing tables, this is achieved by creating additional vRouter containers associated to the same vNET.

Unlike in a traditional multi-VRF router, multiple vRouter containers ensures a complete isolation between routing instances, within the same vNET, with a real resources isolation between routing domains.

In the future, each vRouter will extend to support multi-VRF to help maximizing the scale of VRFs without depending on resource-consuming vRouter containers.

A vRouter in a vNET supports the same feature set a vRouter created in the default vNET.

vNET Administrator

A vNET administrator (called vNET-admin) account is created when a vNET Manager container is created. This account has full read/write access to its associated vNET.

A vNET-admin account can only access his own vNET Manager and can only see resources belonging to its vNET. He’s responsible for the creation/deletion of the following:

- vNET VLANs/VNIs.
- LAGs/vLAGs for Managed ports associated to the vNET.
- vNET vRouters.
- VXLAN Tunnels for the vNET.
- L3 routing protocols for the vNET.

Fabric Administrator

The Fabric administrator (called fabric-admin) is the root account equivalent on a Linux distribution. It has all privileges to create, modify or delete any resource created on any fabric node. Besides having the same privileges that a vNET-admin has, the Fabric-admin can also manage the following fabric attributes:

- Fabric nodes.
- Clusters.
- vNETs.
- All Ports.
- VTEPs.
- vFlows (Security/QoS).
- Software upgrades.

vNET Manager High Availability

When deploying vNV to provide vNET Manager, you have two options to secure the management instance of the vNET:

- On a Single Site using VRRP between vNET Managers.
- On Multiple Sites with two independent instances of vNET Managers.

Single Site

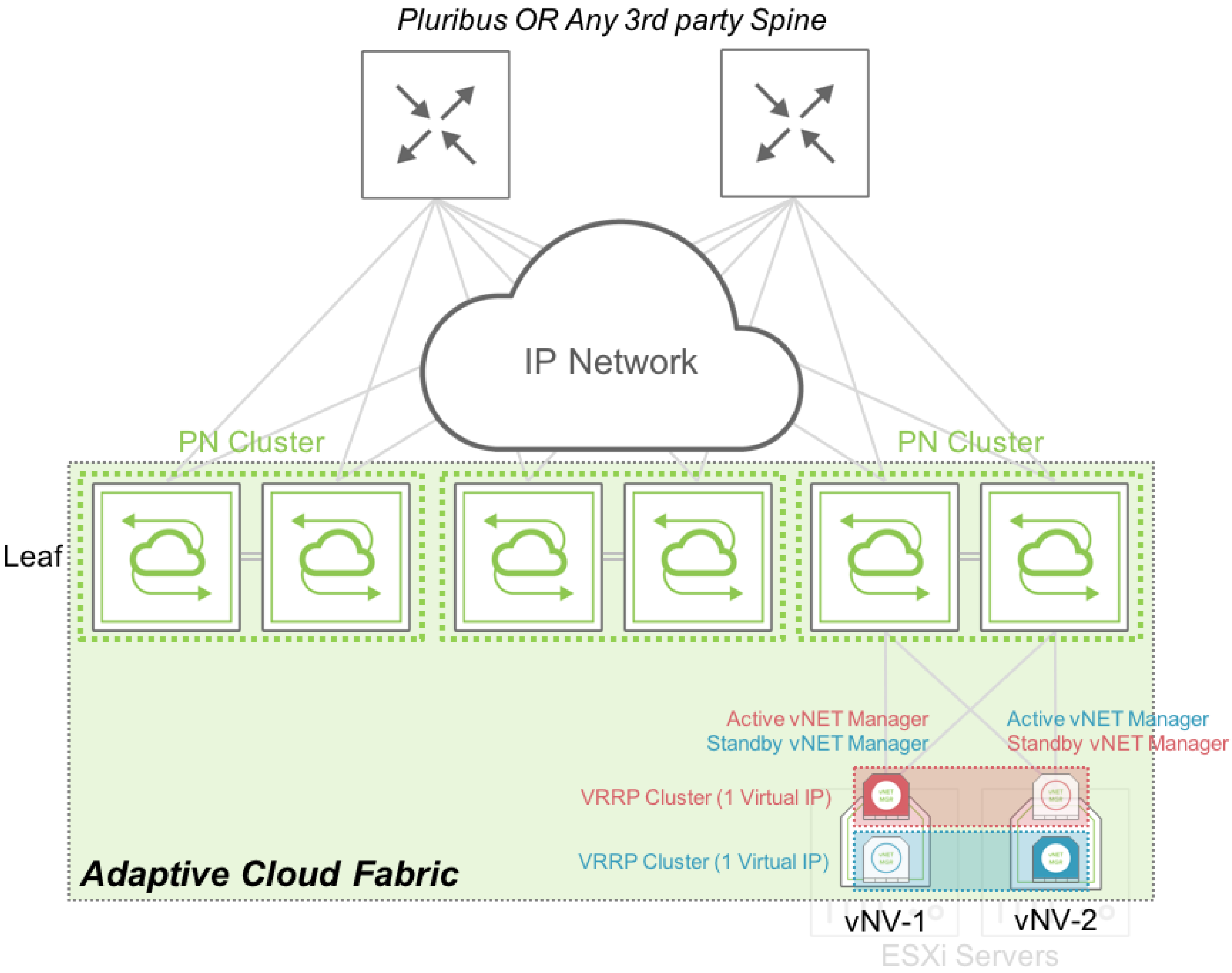
When deploying vNVs in one single site, having Layer 2 domains across switches or racks is common, either through standard VLANs and VLAGs or through VXLANs and tunnels.

In this configuration, you can easily create 2 vNVs sharing the same broadcast domain and run VRRP protocol on their vNET Manager Interface.

Using VRRP between 2 vNVs offers the following benefits:

- One single IP to manage a vNET (easier to map one IP to DNS entries).
- Redundancy is offered by the protocol itself.
- Active/Standby model.

The following diagram shows how we can easily provide HA between vNVs when they are deployed on two servers sitting in the same rack:

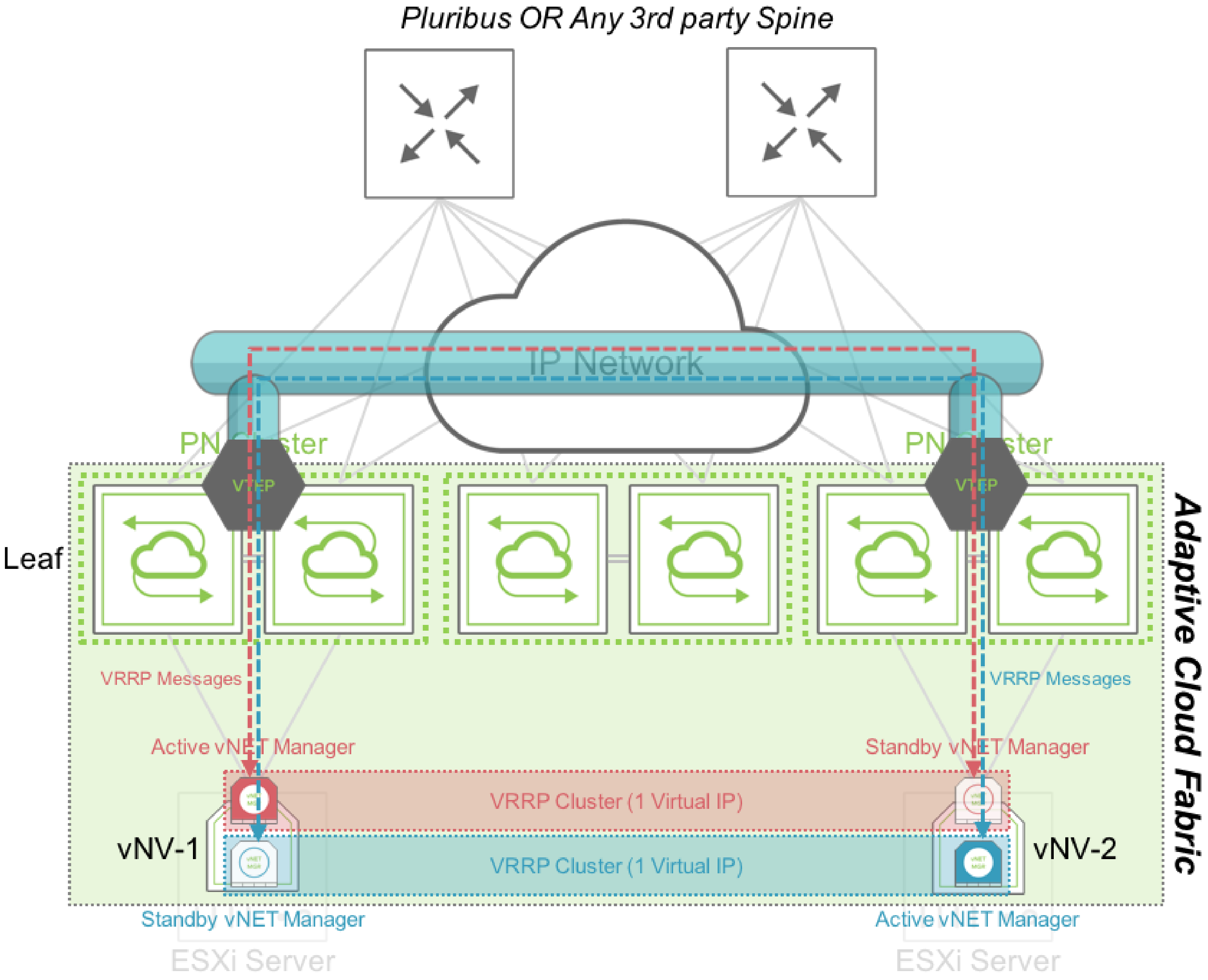


vNET Manager Redundancy – Active/Standby in a Rack

Overview (cont'd)

In this case, both vNVs can exchange VRRP messages locally through the cluster of PN switches at the top of the rack. The vNET administrator has one single IP address to use to connect to his vNET. In case of failure or maintenance of one of the vNV instance, the same IP address is active on the remaining node.

The following diagram shows how we can provide HA between vNVs when they are deployed on two servers sitting in two different racks/clusters and connected through VXLAN tunnels:



vNET Manager Redundancy – Active/Standby between Racks

In this case, both vNVs can exchange VRRP messages through the VXLAN tunnels created between clusters of PN switches at the top of the rack. The vNET administrator has one single IP address to use to connect to his vNET. In case of failure or maintenance of one of the vNV instance, the same IP address is active on the remaining node.

Multiple Sites (with L3 connectivity only)

When deploying vNVs across sites/datacenters, having Layer 2 domains across sites might not be an option and so, you need to rely on a basic IP connectivity between sites. Arista Networks Fabric is perfectly suitable in this kind of multiple sites design along with vNV HA options.

When vNVs are deployed in two different locations, the options mentioned in the previous section still apply, but it might be preferable to separate the IP domains and have two distinct IP addresses for each vNV instance (w/o VRRP).

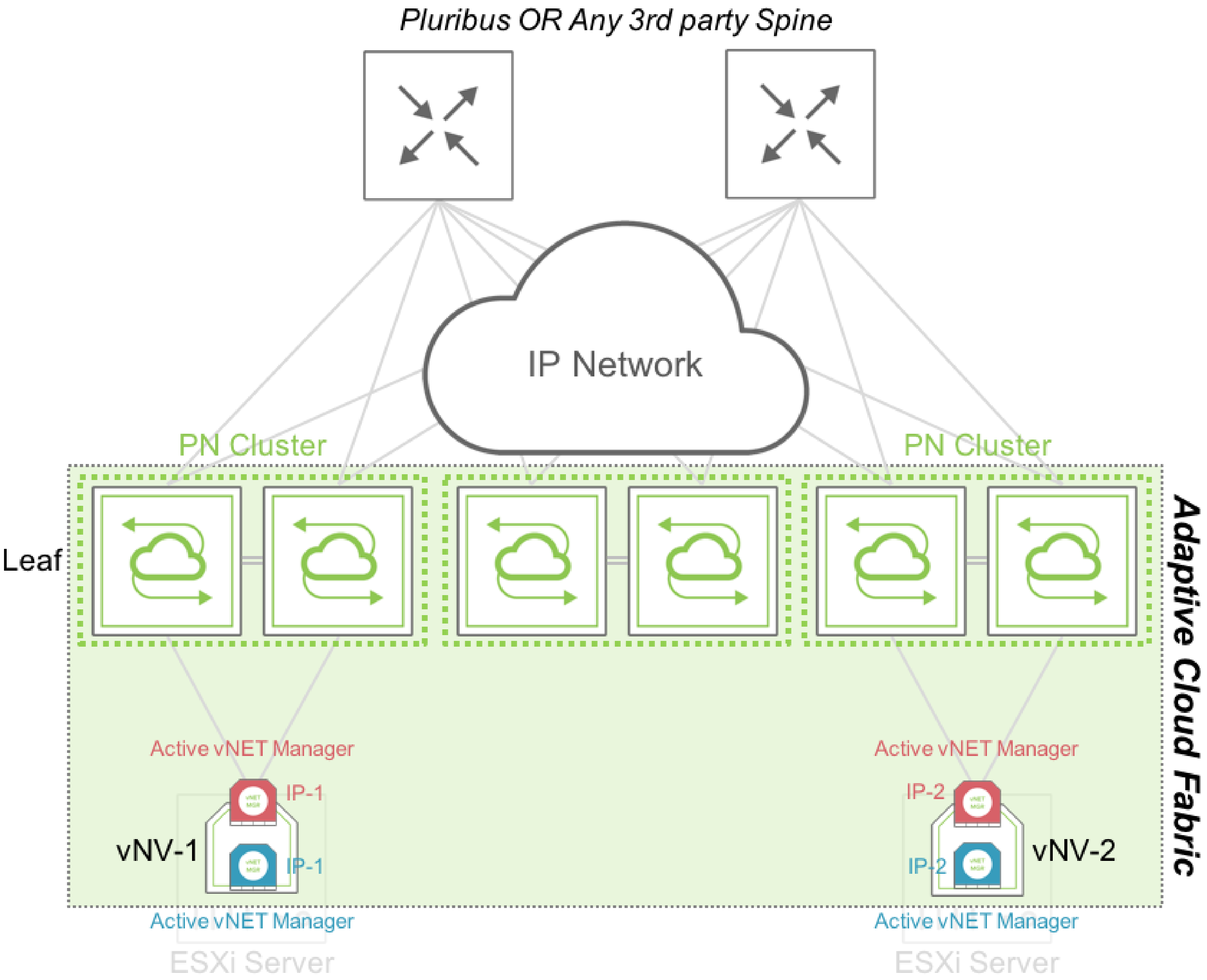
In this configuration, you can easily create 2 vNVs sharing only the knowledge of the vNET but won’t use any control plane messages to offer a single IP address to manage the vNET, both are active at the same time.

Using two distinct vNVs instances without VRRP offers the following benefits:

- Redundancy without L2 dependency.
- Active/Active model with built-in redundancy in the fabric.

The following diagram shows how we can easily provide HA between vNVs when they are deployed in two different racks or Datacenters with L3 connectivity only:

Overview (cont'd)



vNET Manager Redundancy – Active/Active between Racks or Datacenters

Glossary

Glossary of Arista NetVisor UNUM and Arista NetVisor OS Terms

To review the **Glossary of Arista NetVisor UNUM and Arista NetVisor OS Terms**, please refer to to the online document.

vNV Deployment Scenarios and NetVisor UNUM

vNV Deployment Scenarios and Arista NetVisor UNUM

As the Arista Networks Unified Cloud Fabric™ scales to accommodate more services, such as multi-tenancy and analytics, additional resources can be beneficial. In these cases, centralizing the services in a fabric node with more CPU and memory than a typical network switch provides a powerful and cost-effective means of scaling.

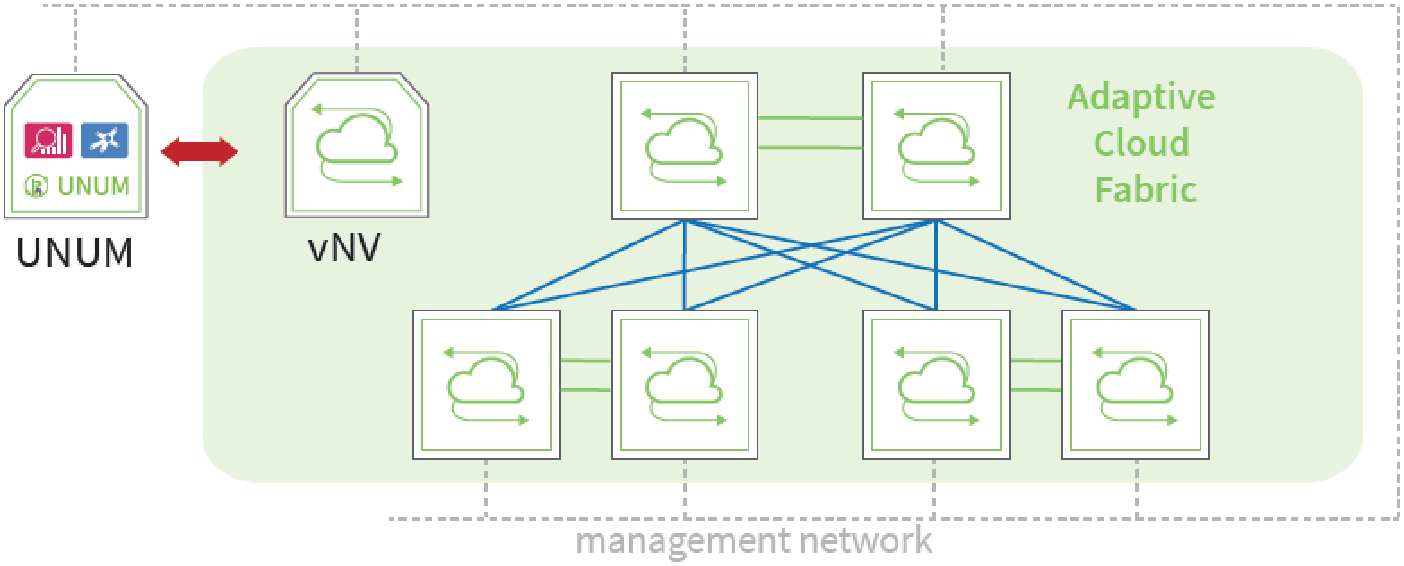
Enter Virtual Netvisor or vNV, a virtual machine running Arista Networks' NetVisor OS operating system. Once deployed and added to a fabric, vNV enables administrators to offload multiple vNET managers and other services from physical network nodes, freeing up switch resources.

Deployment Options and Use Cases

Virtual Netvisor installs on a VMware ESXi server as a virtual machine. Used as a separate fabric node, vNV connects to the fabric through the management network and supports both greenfield and brownfield environments.

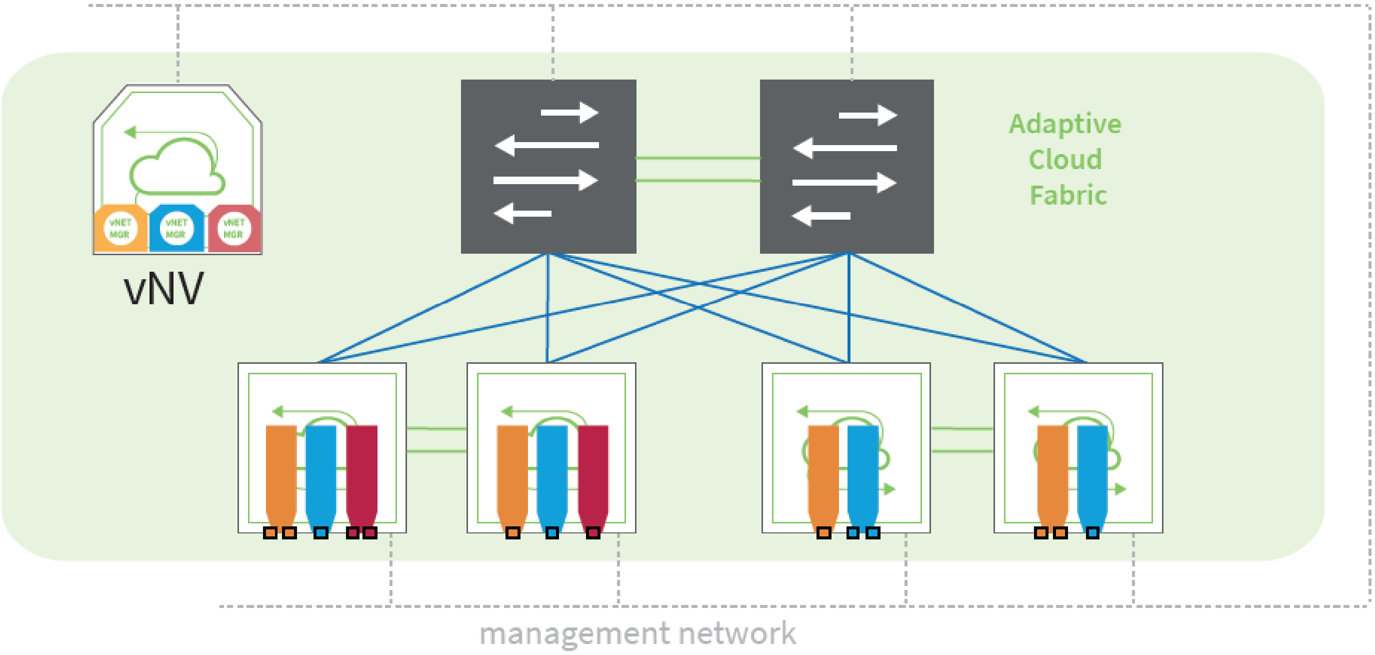
Virtual Netvisor typically deploys with UNUM™, Arista Networks' fabric management, analytics, and automation platform, to move resource consumption from a fabric node known as a seed switch onto the more powerful vNV virtual machine. vNV is especially helpful in environments with high numbers of traffic flows.

Virtual Netvisor - Six Switch, Leaf and Spine Fabric in UNUM



Virtual Netvisor - Deployment Scenario Example 1

Virtual Netvisor - Arista Unified Cloud Fabric with Third-party Spines and three vNET Managers hosted in a Virtual Netvisor Instance



Virtual Netvisor - Deployment Scenario Example 2

Deploy Virtual Networks with vNV

Deployment Workflow

You can create vNETs with vNV by using these high-level steps.

Step 1 Make sure all VMware prerequisites are met. For details, see the following sections:

- Supported VMware vSphere ESXi Hypervisor Versions
- ESXi Host Prerequisites for vNV

Step 2 Build your management and/or underlay network, depending on the topology you want to implement and create a fabric (management or inband).

Step 3 Download the**Arista Networks vNV OVA** from your **PN Cloud** portal.

Step 4 Deploy the **vNV OVA** image in your **VMware vSphere** environment and power on the **VM**.

Step 5 Connect to the console or Linux shell (using SSH) on **vNV-1** to run the switch-setup. If you're not using a **DHCP** server, configure a static **IP** and **default gateway** as appropriate, along with **DNS**, **NTP** and **time zone**.

> **Note:** Ensure you enable all host ports by default during the switch setup.

Step 6 Ensure **vNV-1** has IP connectivity to other nodes (either **L2** or **L3**, depending on the design implemented).

Step 7 Join **vNV-1** with the fabric.

Step 8 (Optional) - Create a range of VLANs to be used to map private VLANs to public VLANs.

Step 9 Create a **vNET** with a **vNET Manager** and a **vNET Admin** user.

Step 10 Associate Managed Ports to this **vNET**.

Step 11 (Optional) - Associate Shared Ports to this **vNET**.

In the event you want to provide full redundancy for the vNET Manager container, you instantiate a second vNV (vNV-2) by repeating **Steps 3 to 6** and configure either VRRP between both vNVs or simply use 2 different IP addresses.

Prerequisites for deploying vNETs with vNV

Supported VMware® vSphere ESXi Hypervisor Versions

Arista Networks vNV is supported on the following VMware ESXi Hypervisor and vCenter Server versions:

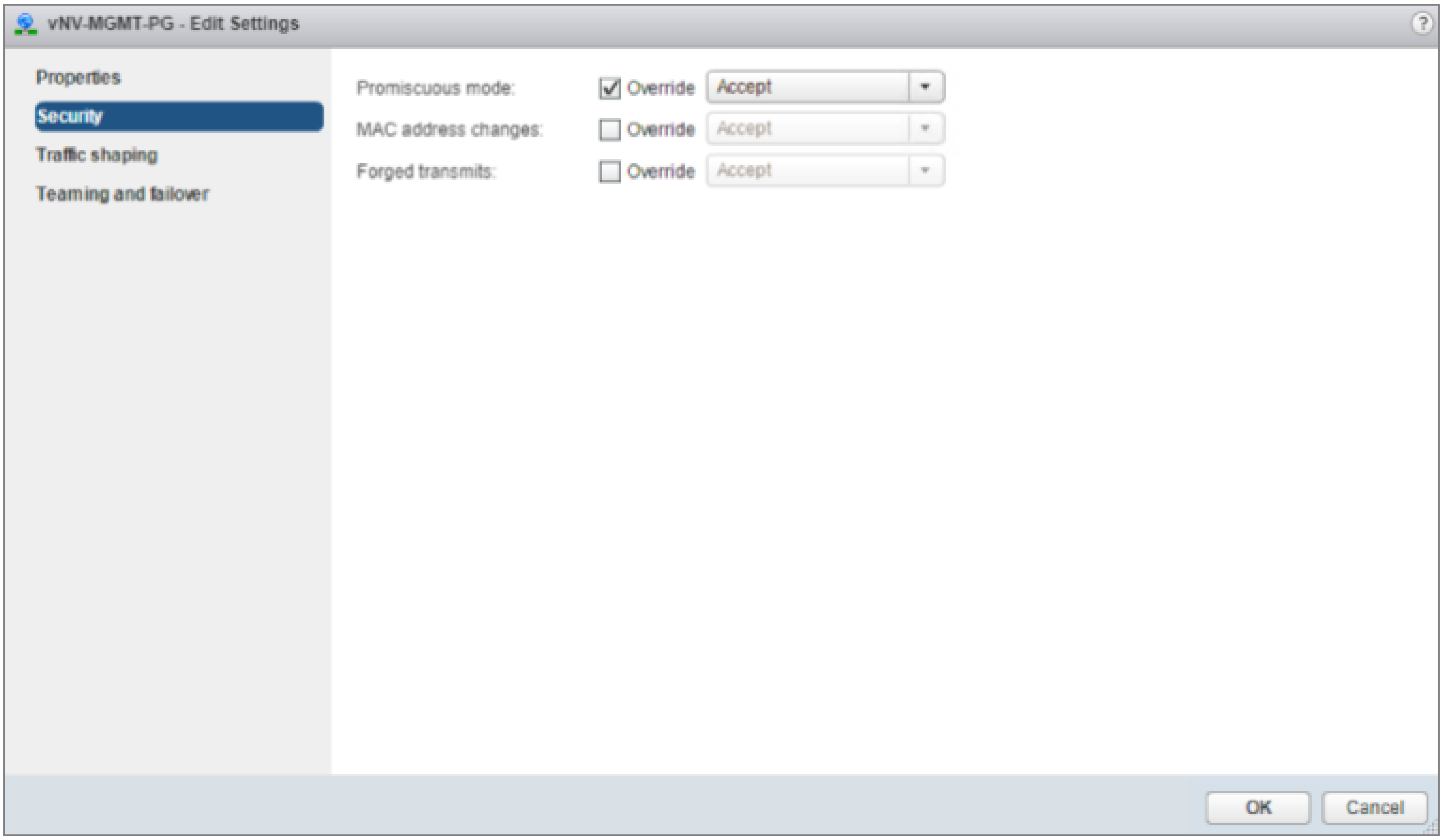
- 6.5.x
- 6.7.x
- 7.0.x

ESXi Host Prerequisites for vNV

VMware ESXi hosts have the following prerequisites:

- You have installed and prepared vCenter Server for host management using the instructions from VMware.
- You have VMware vSphere Client installed or vCenter integration Plug-in in your browser.
- A specific license is not required for the vCenter Server to deploy vNV OVA images.
- You must configure at least two Port-groups:
    - The Management Port-group (with Promiscuous mode, MAC address changes and Forged transmits parameters set to “Accept”) using one VLAN.

Deploying Virtual Networks with vNV (cont'd)



vNV-MGMT-PG - Edit Settings

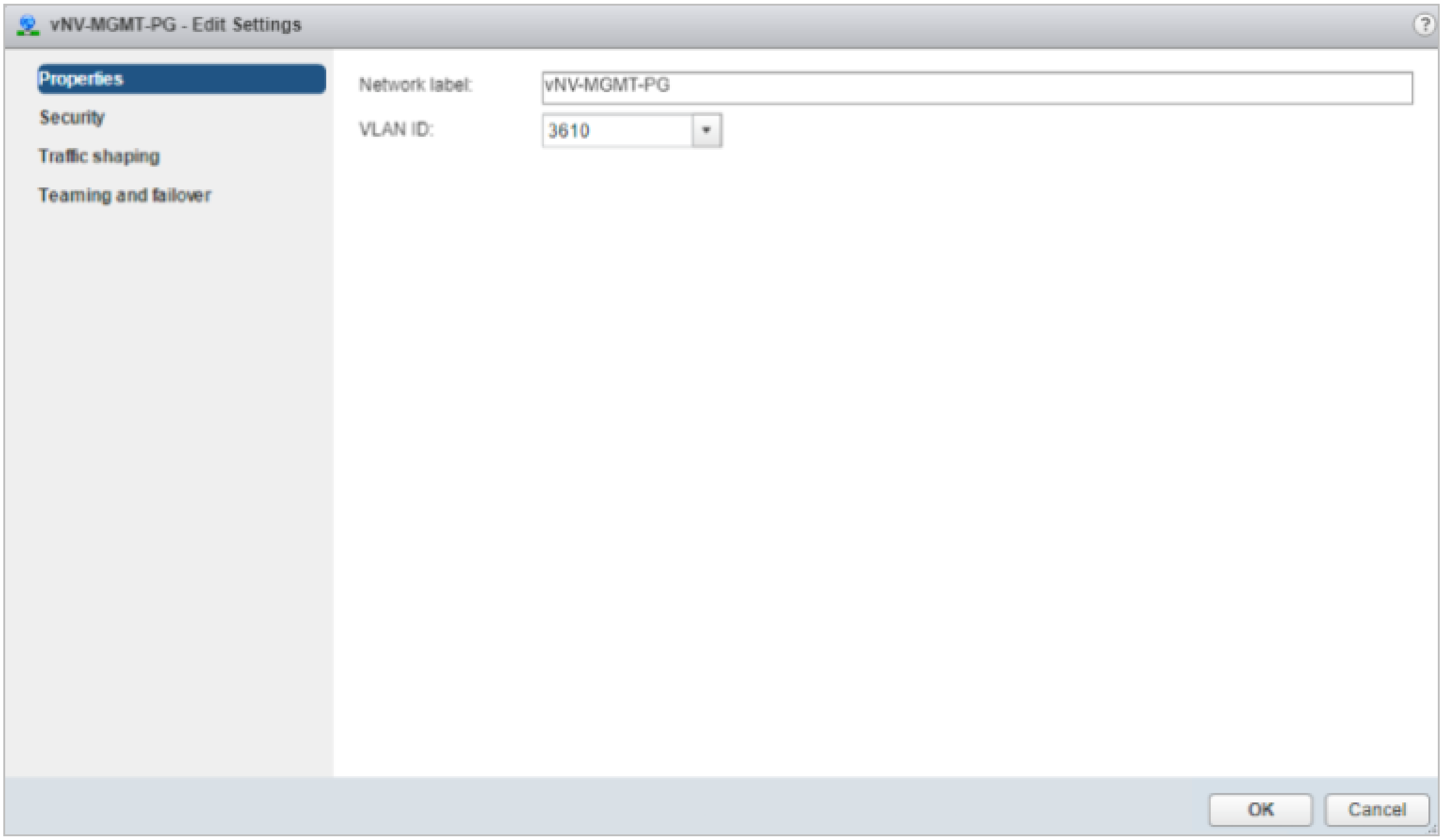
Promiscuous mode: ☑ Override Accept

MAC address changes: □ Override Accept

Forged transmits: □ Override Accept

Management Port-group Security Configuration on a Standard vSwitch

Deploying Virtual Networks with vNV (cont'd)



vNV-MGMT-PG - Edit Settings

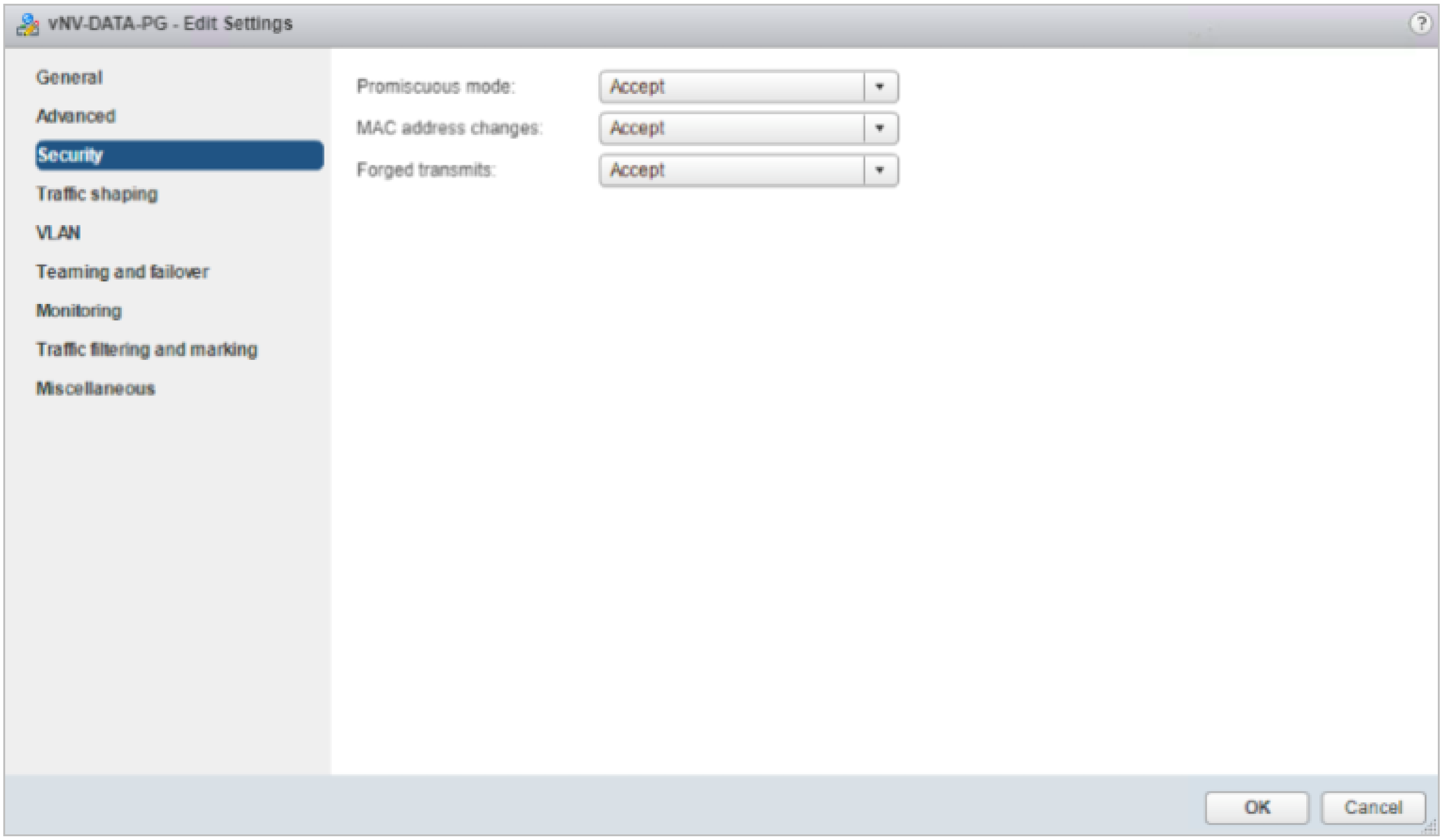
Network label: vNV-MGMT-PG

VLAN ID: 3610

Management Port-group VLAN Configuration on a Standard vSwitch

Deploying Virtual Networks with vNV (cont'd)

- In-band/Data Port-group (with Promiscuous mode, MAC address changes and Forged transmits parameters set to “Accept”) using a one VLAN or an 802.1q trunk, depending on the design.



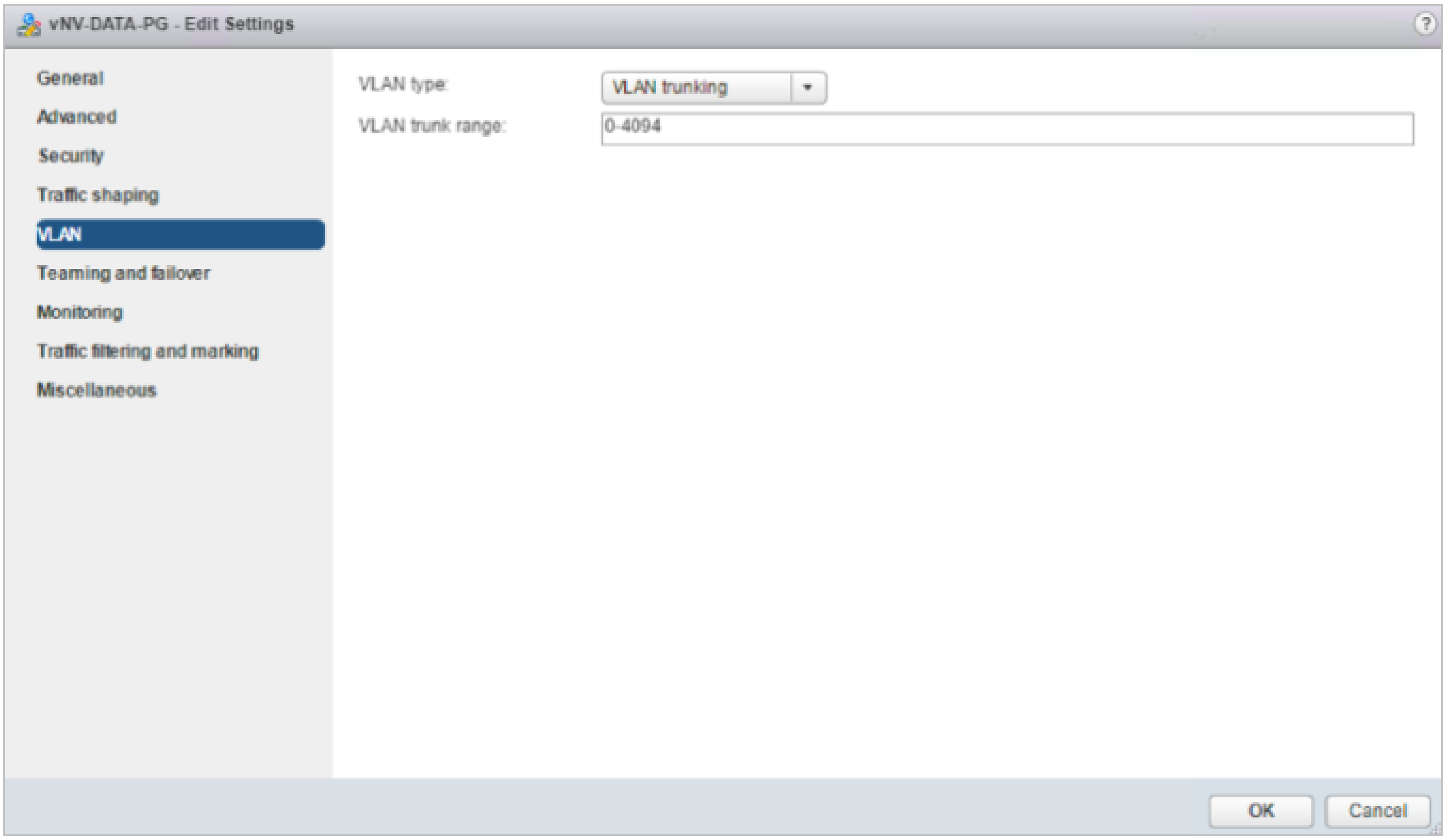
Promiscuous mode: Accept

MAC address changes: Accept

Forged transmits: Accept

Data/In-band Port-group Security Configuration on a Distributed vSwitch

Deploying Virtual Networks with vNV (cont'd)



VLAN type: VLAN trunking

VLAN trunk range: 0-4094

Data/In-band Port-group VLAN Configuration on a Distributed vSwitch

Deploying Virtual Networks with vNV (cont'd)

- You have two physical NICs on each host for redundancy. Deployment is also possible with one physical NIC.
- If you are using a set of switches, make sure the inter-switches links carry all relevant VLANs, including the Management VLAN. The uplink should be a port, single trunk or VLAG, with all VLAN traffic configured on the host.
- Ensure the VMs to be used for vNV meet the minimum requirements listed in the following table:

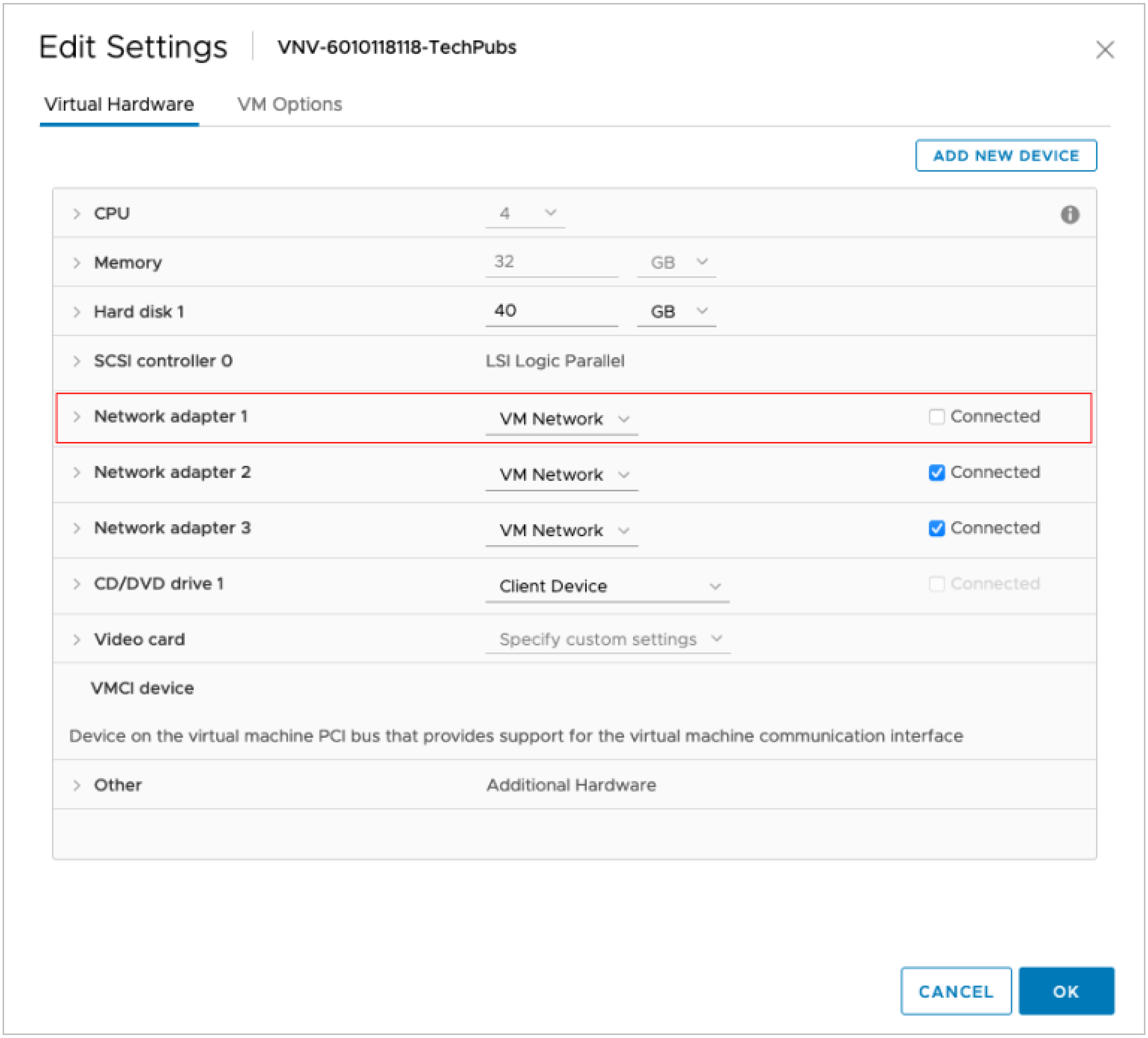
Function	Component	Minimum Requirement
vNV	Hardware Version	9 and higher
	Platform	64-bit
	Type	Other 64-bit Linux
	Processor	4 vCPU (2 cores hyper threaded)
	Memory	32 GB
	Disk	40 GB in Thin Provisioning
	CPU Speed	> 2 GHz

Virtual Machine Minimum Requirements

Note: A first interface is created on vNV in addition to the two others used for **Management** and **In-band**. Do **NOT** use this interface, you must disconnect it during the deployment phase, it is not used by Netvisor.

Deploying Virtual Networks with vNV (cont'd)

Network Adapter 1 Disabled



Edit Settings | VNV-6010118118-TechPubs

Virtual Hardware | VM Options

CPU	4	
Memory	32 GB	
Hard disk 1	40 GB	
SCSI controller 0	LSI Logic Parallel	
Network adapter 1	VM Network	□ Connected
Network adapter 2	VM Network	☑ Connected
Network adapter 3	VM Network	☑ Connected
CD/DVD drive 1	Client Device	□ Connected
Video card	Specify custom settings	
VMCI device		
Other	Additional Hardware	

Device on the virtual machine PCI bus that provides support for the virtual machine communication interface

vNV VM Configuration - Disable Network Adapter 1

Fabric Nodes Prerequisites

Fabric nodes have the following prerequisites:

- Fabric nodes must run the same Netvisor version to ensure a consistent and predictable behavior across the fabric.
- If you plan to deploy a management fabric with an L3 boundary to connect to vNVs, make sure to configure the default gateway on all switches.
- If you plan to deploy an in-band fabric with VRRP to provide a redundant default gateway to vNVs, ensure this VLAN is created and tagged on ports going to the ESXi servers where vNVs is deployed both on the cluster links and the subnet is redistributed (or advertised) in the IGP.

Deploying vNV

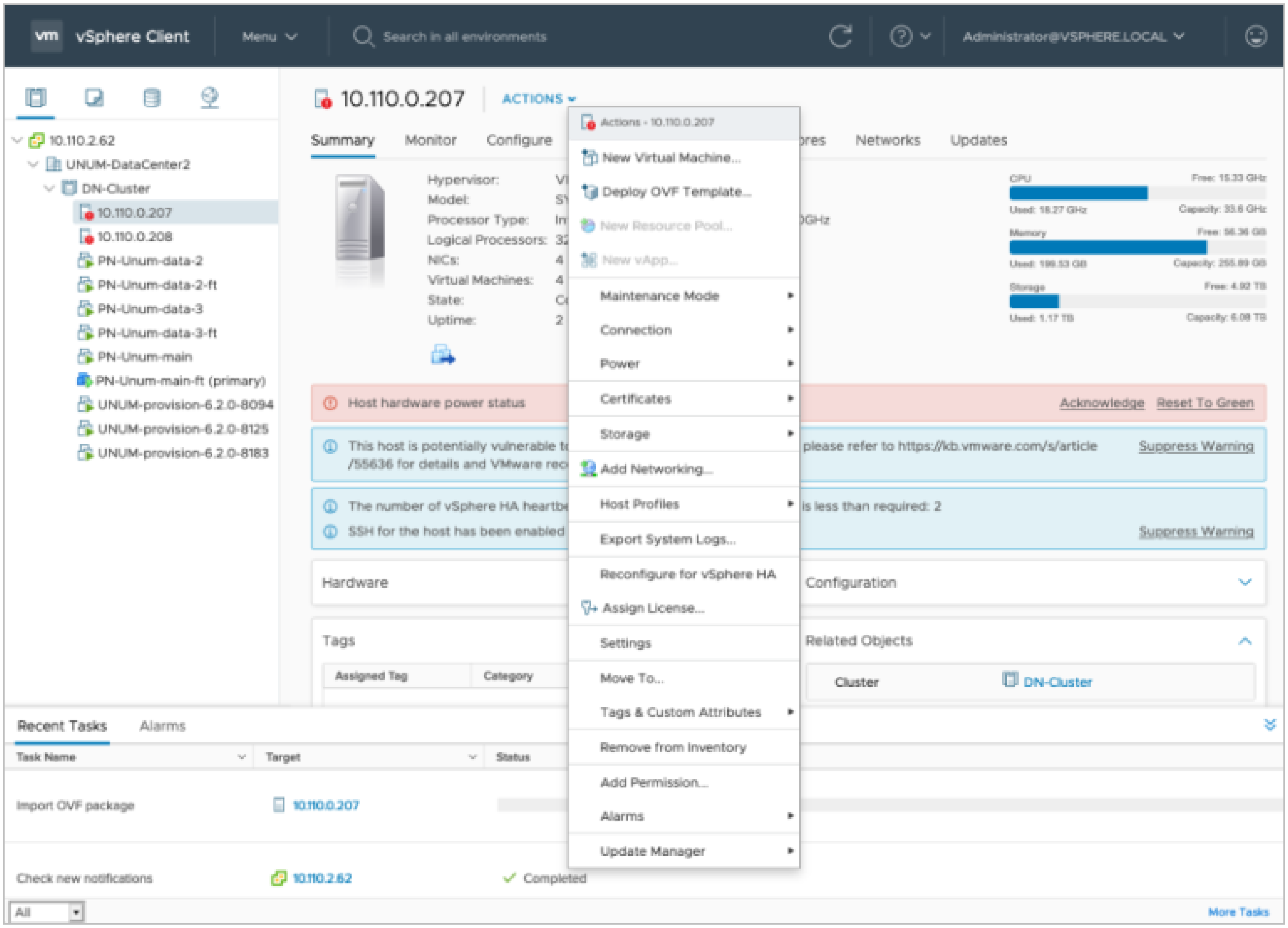
Before you begin this procedure, you must do the following:

- Know the location and image name of the OVA image required for the installation.
- You have already read the Prerequisites for deploying vNETs with vNV.
- For detailed information about using the Deploy OVF Template wizard, see the vSphere Virtual Machine Administration Guide.
- You have the following information available for creating a vNV VM and mapping the required port groups:
    - A name for the new vNV that is unique within the inventory folder and up to 80 characters.
    - The hostname and the inventory folder that contains the vNV you plan to use.
    - The name and location of the VM datastore you plan to use.
    - The names of the network port groups the VM you plan to use.
    - The vNV IP address or an FQDN needed to access the VM.

Deploying Virtual Networks with vNV (cont'd)

Procedure:

Step 1 From the vSphere Web Client, find the host, cluster and/or resource pool where you want to deploy and right click > **Deploy OVF Template...**



Hypervisor: VI...
Model: SY...
Processor Type: In...
Logical Processors: 32
NICs: 4
Virtual Machines: 4
State: Co...
Uptime: 2

...GHz

CPU Free: 15.33 GHz
Used: 18.27 GHz Capacity: 33.6 GHz
Memory Free: 56.36 GB
Used: 199.53 GB Capacity: 255.89 GB
Storage Free: 4.92 TB
Used: 1.17 TB Capacity: 6.08 TB

Host hardware power status

This host is potentially vulnerable t... please refer to https://kb.vmware.com/s/article /55636 for details and VMware rec...

The number of vSphere HA heartbe... is less than required: 2

SSH for the host has been enabled

Assigned Tag	Category

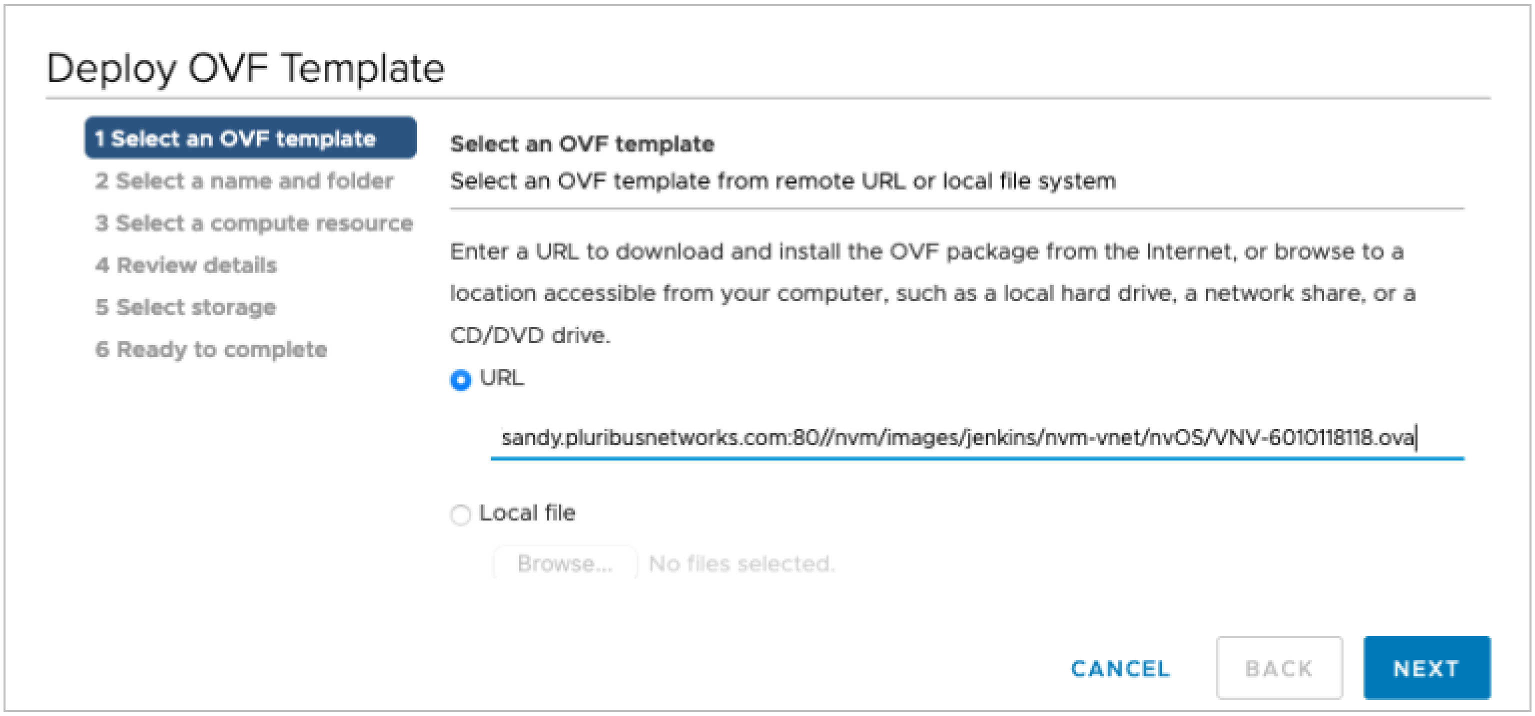
Cluster	DN-Cluster

Task Name	Target	Status
Import OVF package	10.110.0.207	
Check new notifications	10.110.2.62	✓ Completed

Deploy OVF Template

Deploying Virtual Networks with vNV (cont'd)

Step 2 In the Template screen, specify the location of the **OVA** file as a **URL** or **Local File** and click **Next**.



Deploy OVF Template

Select an OVF template

Select an OVF template from remote URL or local file system

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

● URL

sandy.pluribusnetworks.com:80//nvm/images/jenkins/nvm-vnet/nvOS/VNV-6010118118.ova

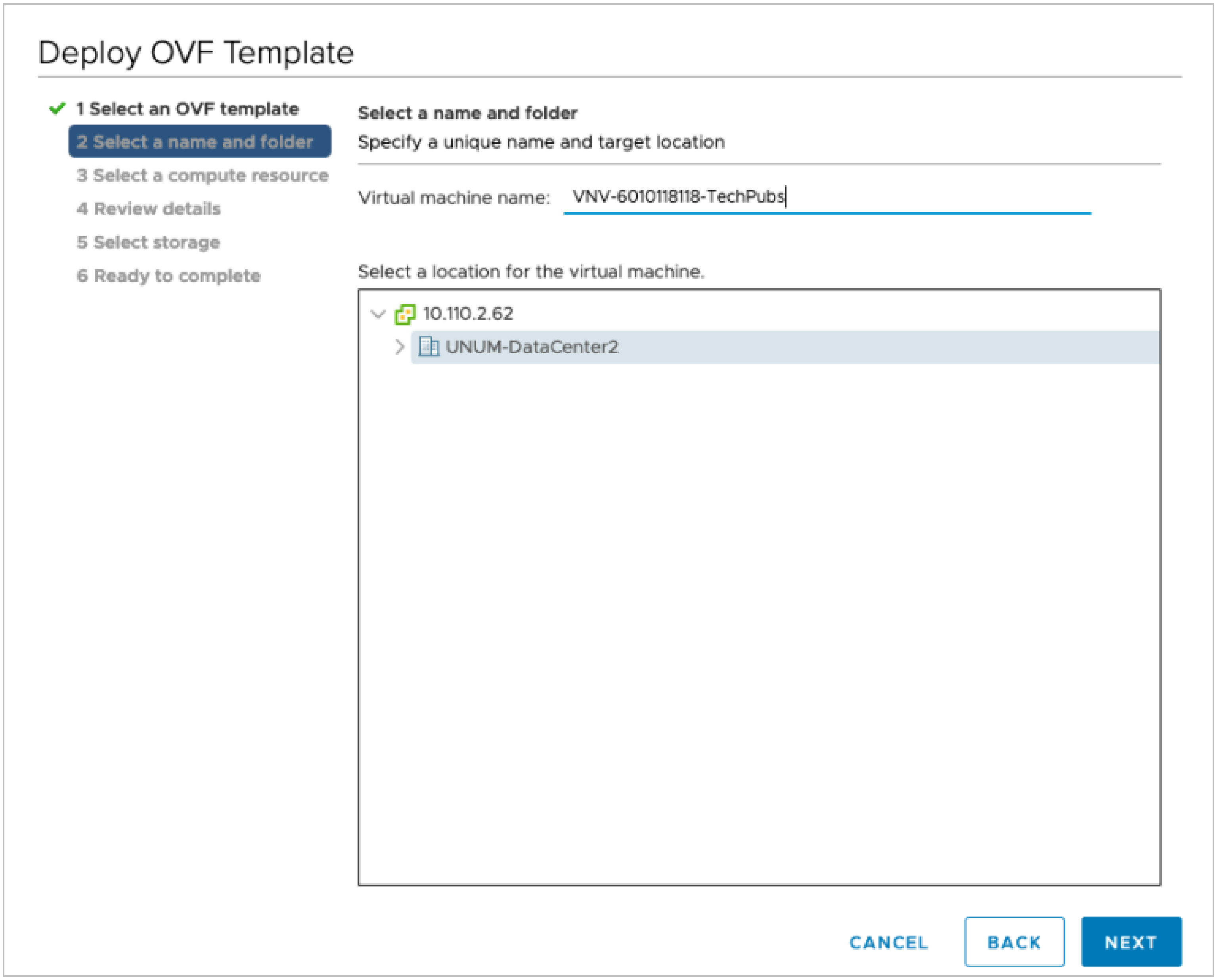
○ Local file

Browse... No files selected.

Select OVF Template

Deploying Virtual Networks with vNV (cont'd)

Step 3 In the **Name** and **Location** screen, specify the **Name** of your **vNV VM** and select datacenter or folder in the **Inventory** and click **Next**.



Deploy OVF Template

✓ 1 Select an OVF template

2 Select a name and folder

3 Select a compute resource

4 Review details

5 Select storage

6 Ready to complete

Select a name and folder

Specify a unique name and target location

Virtual machine name: VNV-6010118118-TechPubs

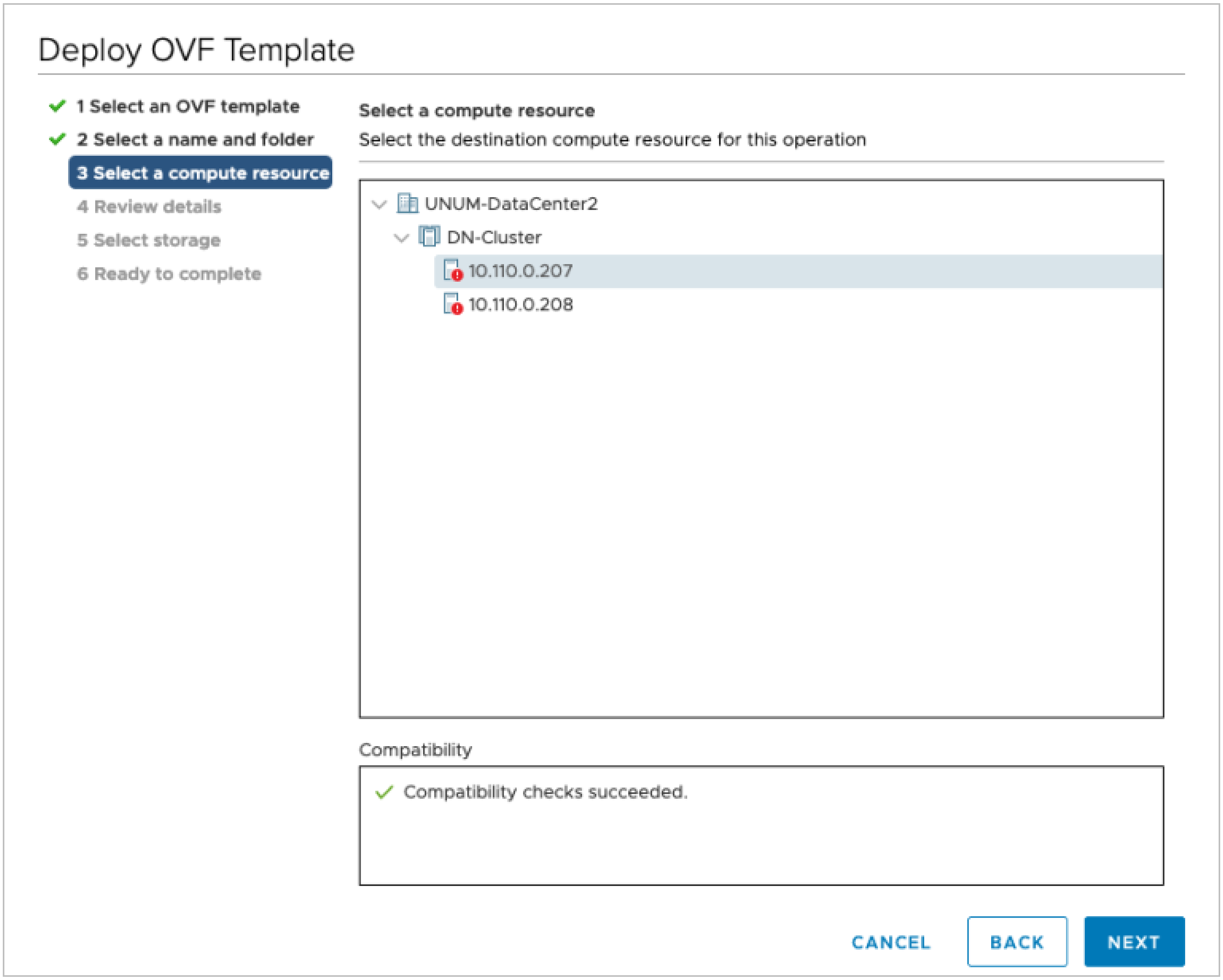
Select a location for the virtual machine.

- 10.110.2.62
  - UNUM-DataCenter2

Select Name and Location

Deploying Virtual Networks with vNV (cont'd)

Step 4 In the **Select a Compute Resource** screen, select host or cluster or resource pool and click **Next**.



Deploy OVF Template

✓ 1 Select an OVF template

✓ 2 Select a name and folder

3 Select a compute resource

4 Review details

5 Select storage

6 Ready to complete

Select a compute resource

Select the destination compute resource for this operation

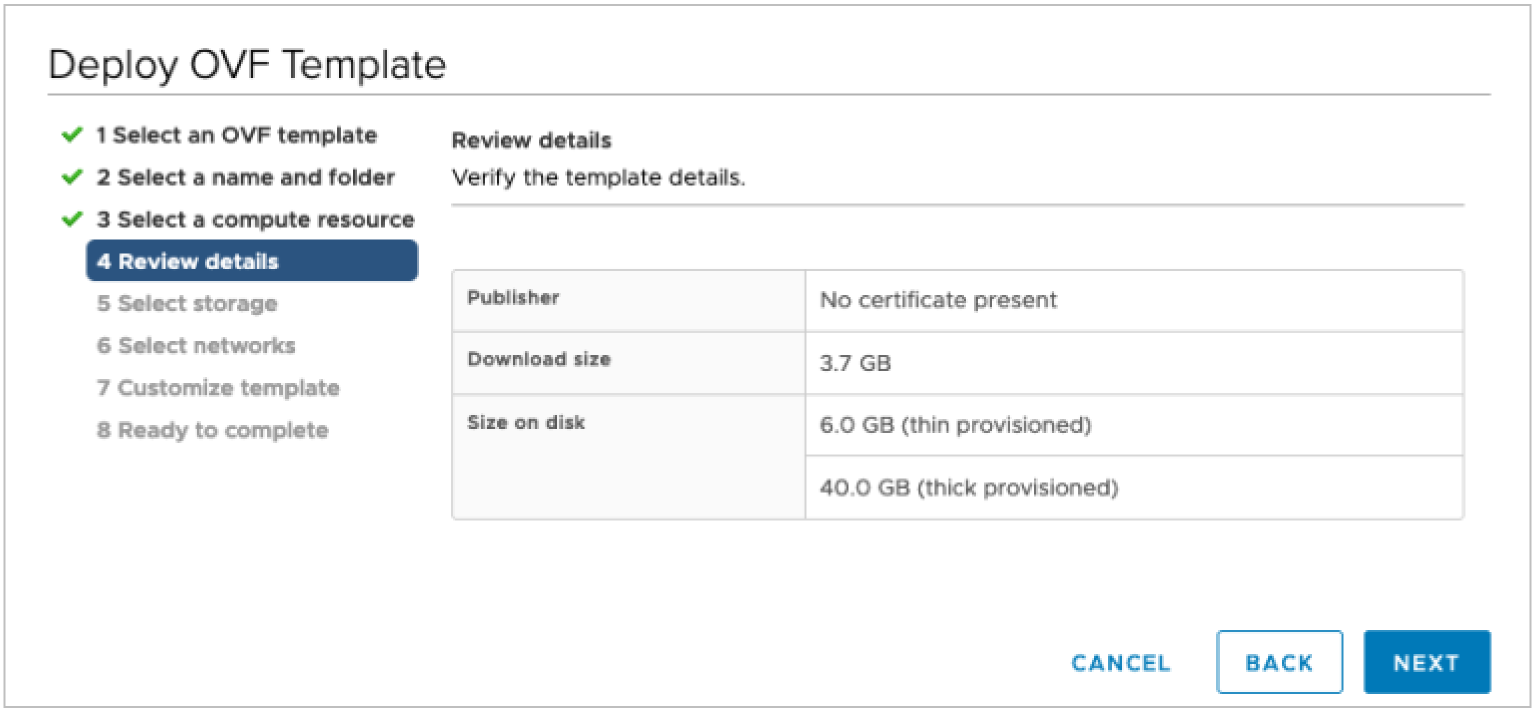
- UNUM-DataCenter2
  - DN-Cluster
    - 10.110.0.207
    - 10.110.0.208

Compatibility

✓ Compatibility checks succeeded.

Select Host or Cluster

Step 5 In the **Review Details** screen, verify details and click **Next**.



Deploy OVF Template

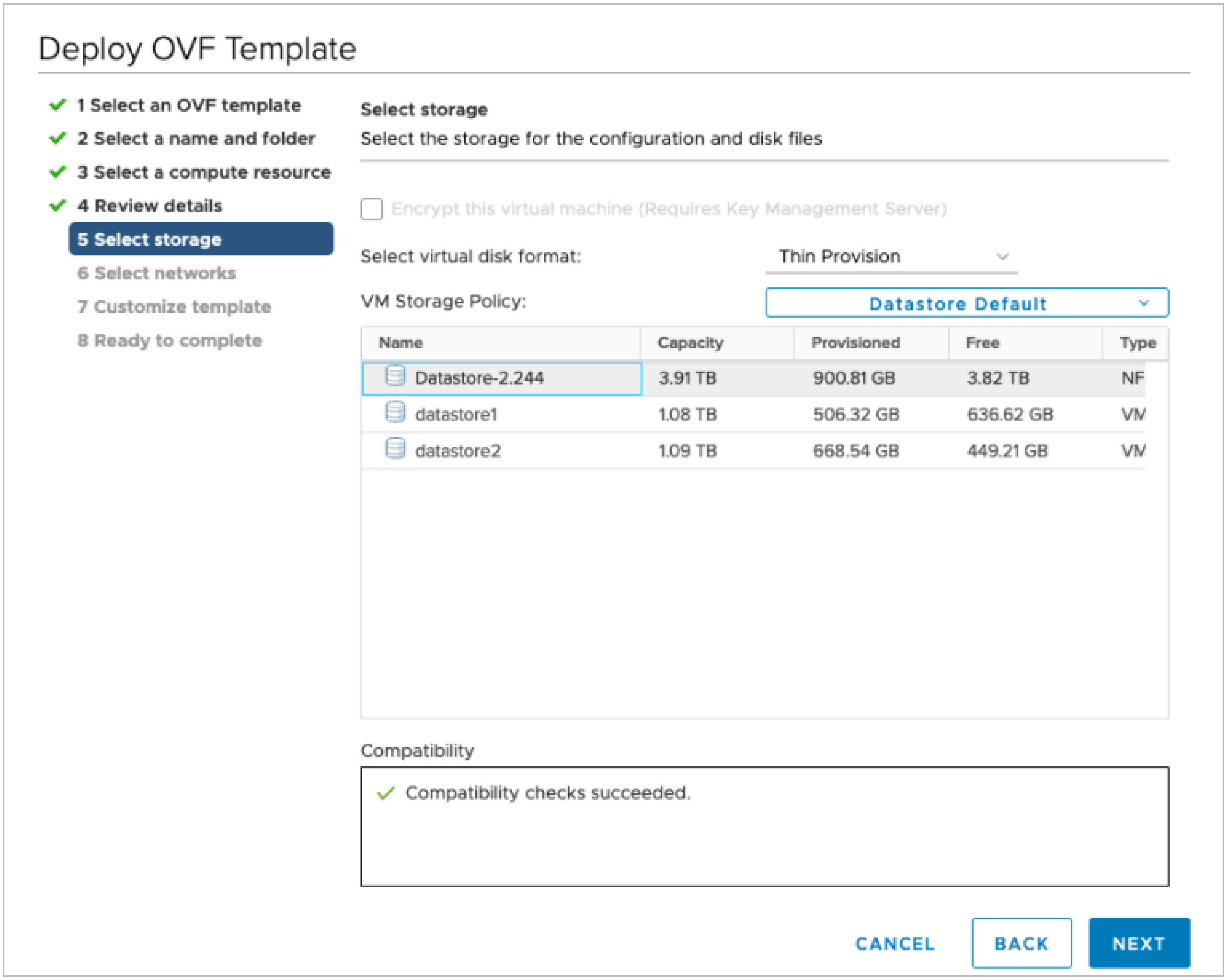
Review details

Verify the template details.

Publisher	No certificate present
Download size	3.7 GB
Size on disk	6.0 GB (thin provisioned)
	40.0 GB (thick provisioned)

Review Details

Step 6 In the **Select Storage** screen, select **Thin Provision** from Select virtual disk format list, select the desired **Datastore** and click **Next**.



Deploy OVF Template

Select storage

Select the storage for the configuration and disk files

□ Encrypt this virtual machine (Requires Key Management Server)

Select virtual disk format: Thin Provision

VM Storage Policy: Datastore Default

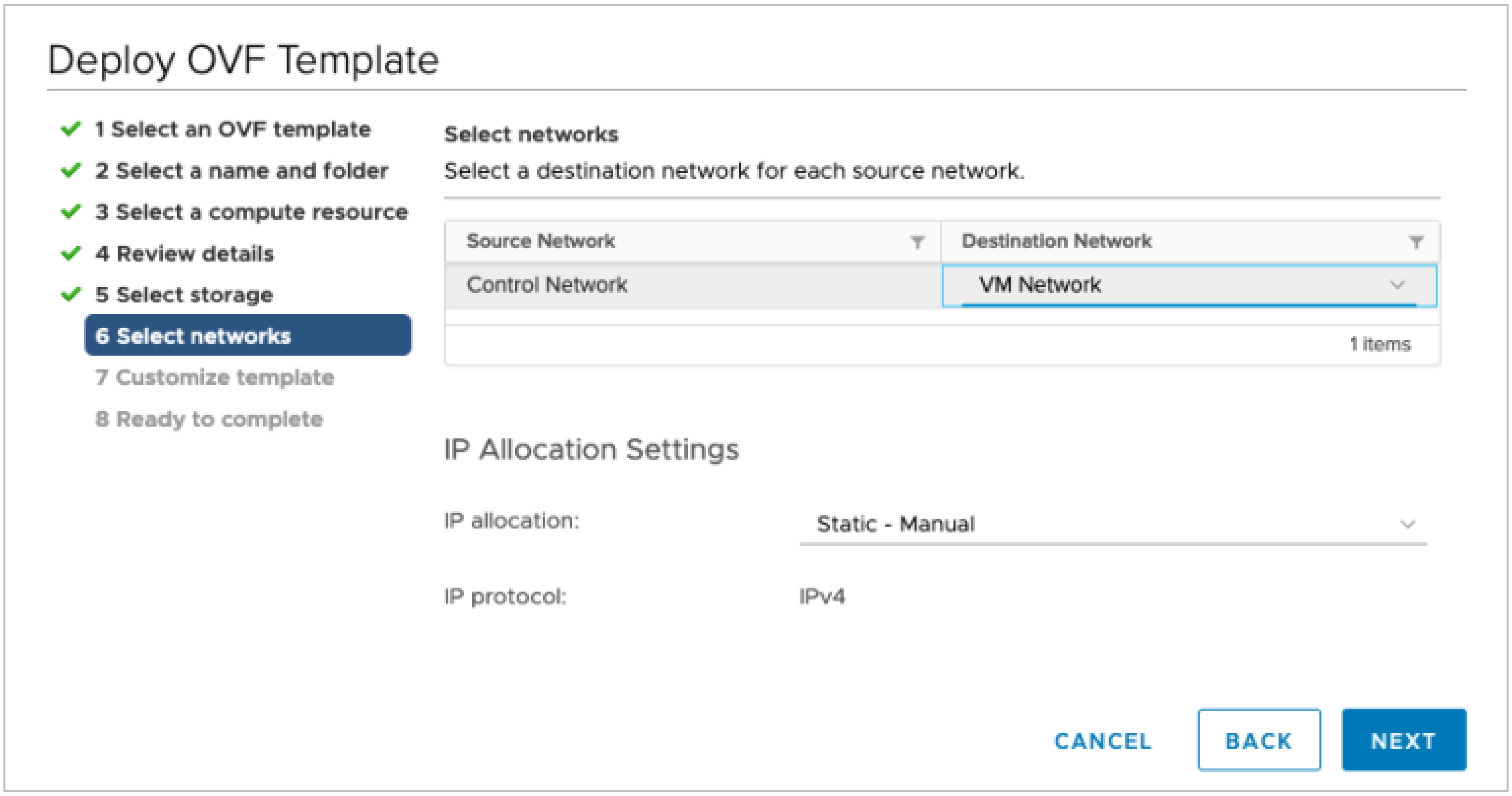
Name	Capacity	Provisioned	Free	Type
Datastore-2.244	3.91 TB	900.81 GB	3.82 TB	NF
datastore1	1.08 TB	506.32 GB	636.62 GB	VM
datastore2	1.09 TB	668.54 GB	449.21 GB	VM

Compatibility

✓ Compatibility checks succeeded.

Select Storage

Step 7 In the Networks screen, select **VM Network** from **Destination Network**.



Deploy OVF Template

- ✓ 1 Select an OVF template
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Review details
- ✓ 5 Select storage
- 6 Select networks
- 7 Customize template
- 8 Ready to complete

Select networks

Select a destination network for each source network.

Source Network	Destination Network
Control Network	VM Network

1 items

IP Allocation Settings

IP allocation: Static - Manual

IP protocol: IPv4

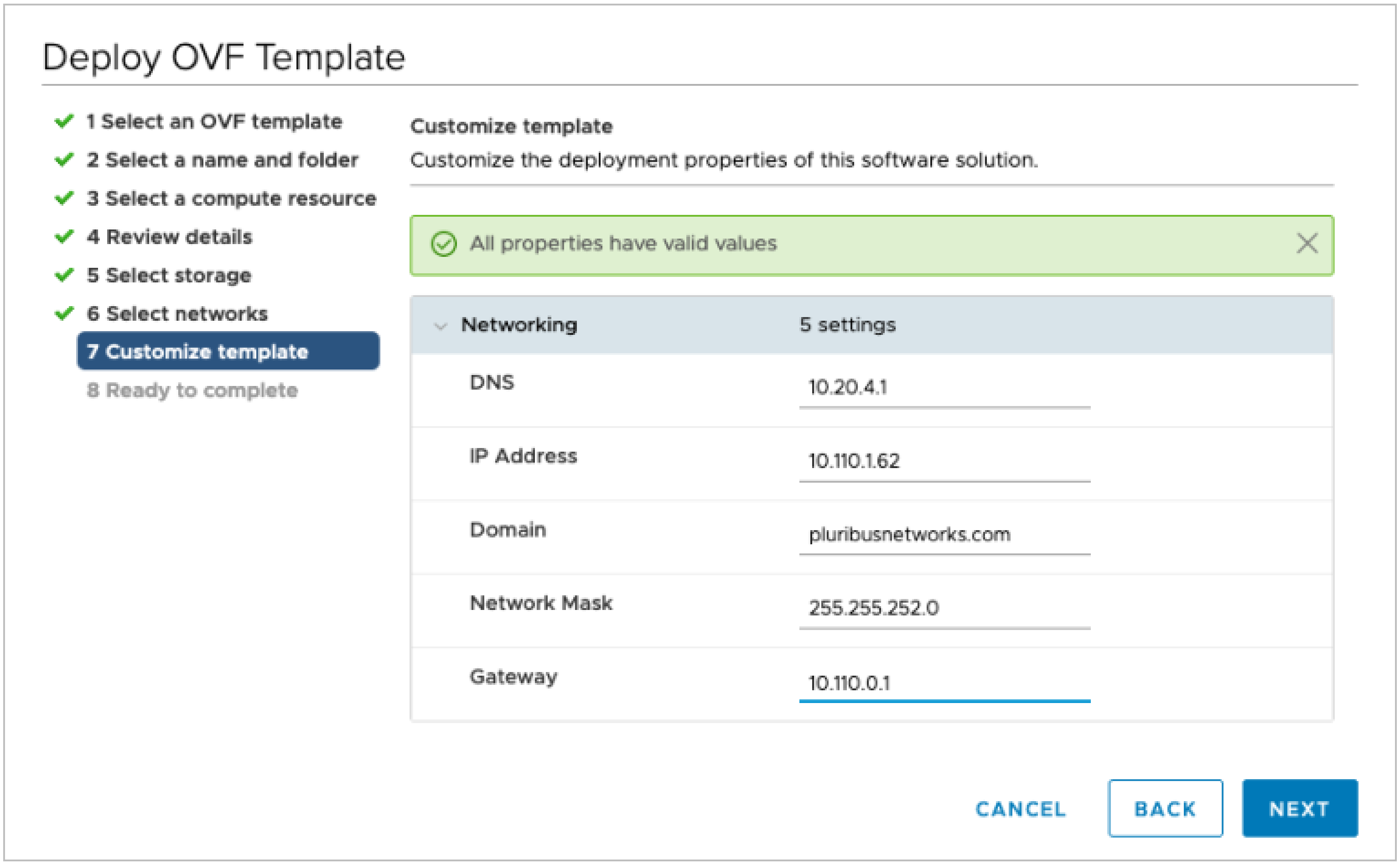
Select Networks

Select the desired type of network configuration from **IP Allocation**, such as **DHCP** or **Static-Manual**.

Deploying Virtual Networks with vNV (cont'd)

When configuring a **Static IP**, enter the specific network values as shown in the following example.

For **DHCP** configuration, leave the fields empty.



Deploy OVF Template

Customize template

Customize the deployment properties of this software solution.

All properties have valid values

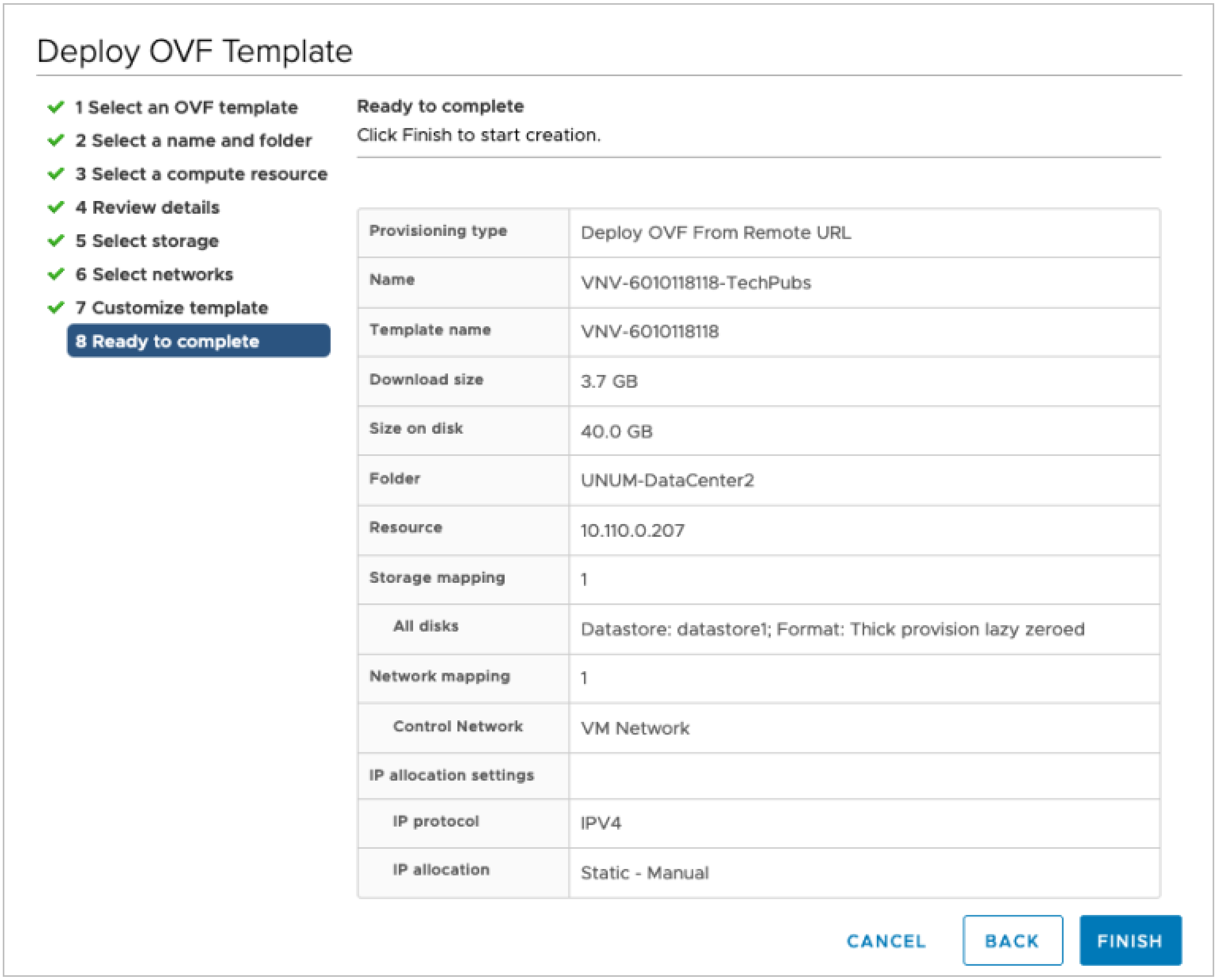
Networking	5 settings
DNS	10.20.4.1
IP Address	10.110.1.62
Domain	pluribusnetworks.com
Network Mask	255.255.252.0
Gateway	10.110.0.1

Static IP Network Configuration

After configuring the network configuration, click **Next**.

Deploying Virtual Networks with vNV (cont'd)

Step 8 Verify the configuration is correct in the **Ready to Complete** screen and click **Finish**.



Deploy OVF Template

- ✓ 1 Select an OVF template
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Review details
- ✓ 5 Select storage
- ✓ 6 Select networks
- ✓ 7 Customize template
- 8 Ready to complete

Ready to complete

Click Finish to start creation.

Provisioning type	Deploy OVF From Remote URL
Name	VNV-6010118118-TechPubs
Template name	VNV-6010118118
Download size	3.7 GB
Size on disk	40.0 GB
Folder	UNUM-DataCenter2
Resource	10.110.0.207
Storage mapping	1
All disks	Datastore: datastore1; Format: Thick provision lazy zeroed
Network mapping	1
Control Network	VM Network
IP allocation settings	
IP protocol	IPV4
IP allocation	Static - Manual

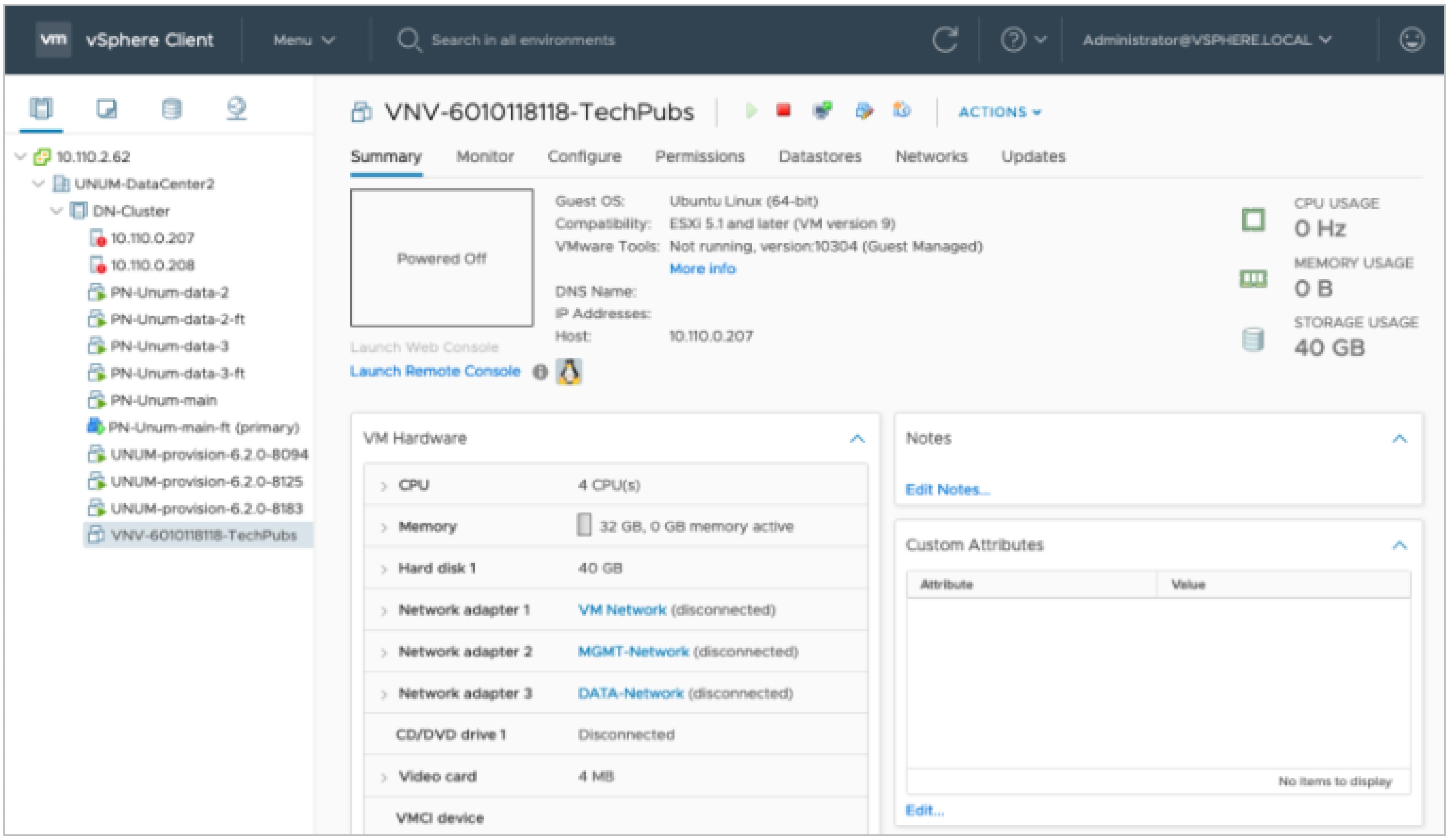
Ready to Complete and Review Settings

Deploying Virtual Networks with vNV (cont'd)

You have now completed the deployment of your vNV.

Check the progress of vNV VM installation in the tasks section.

Once the installation successfully completes, ensure the vNV VM is powered **OFF** and proceed to the next step.



VNV-6010118118-TechPubs

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 5.1 and later (VM version 9)
VMware Tools: Not running, version:10304 (Guest Managed)
More info
DNS Name:
IP Addresses:
Host: 10.110.0.207

CPU USAGE 0 Hz
MEMORY USAGE 0 B
STORAGE USAGE 40 GB

VM Hardware

CPU	4 CPU(s)
Memory	32 GB, 0 GB memory active
Hard disk 1	40 GB
Network adapter 1	VM Network (disconnected)
Network adapter 2	MGMT-Network (disconnected)
Network adapter 3	DATA-Network (disconnected)
CD/DVD drive 1	Disconnected
Video card	4 MB
VMCI device	

Notes

Edit Notes...

Custom Attributes

Attribute	Value

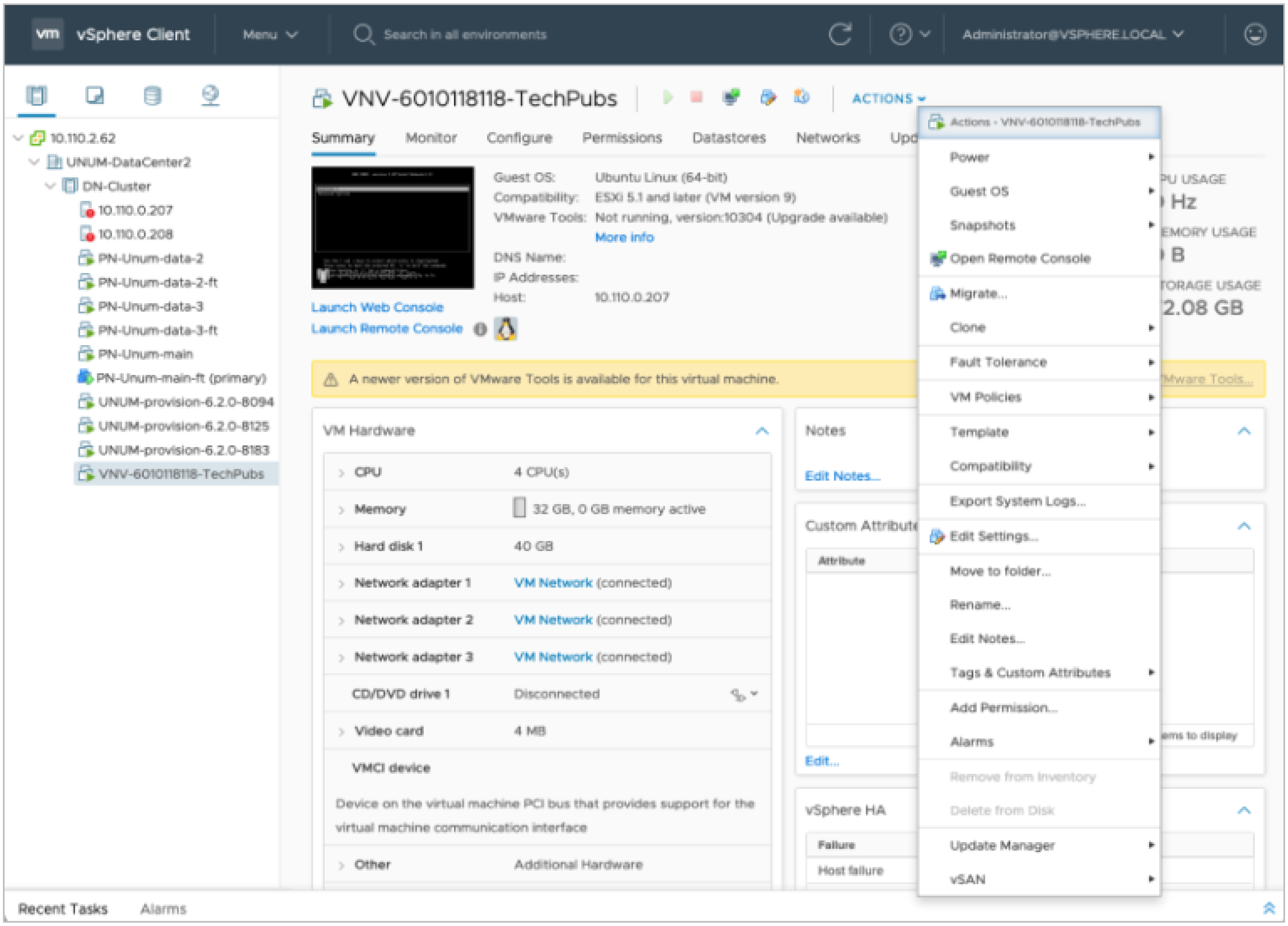
No items to display

Edit...

vNV Instance Powered Off

Deploying Virtual Networks with vNV (cont'd)

Step 9 Right click on **vNV** and select “**Edit Settings**” to modify network settings on your **vNV**. Disable “**Network Adapter 1**” (un-check **Connect at Power On**), select **Management Network** for “**Network Adapter 2**” and **Inband/Data Network** for “**Network Adapter 3**” and click OK."



Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 5.1 and later (VM version 9)
VMware Tools: Not running, version:10304 (Upgrade available)
More info
DNS Name:
IP Addresses:
Host: 10.110.0.207

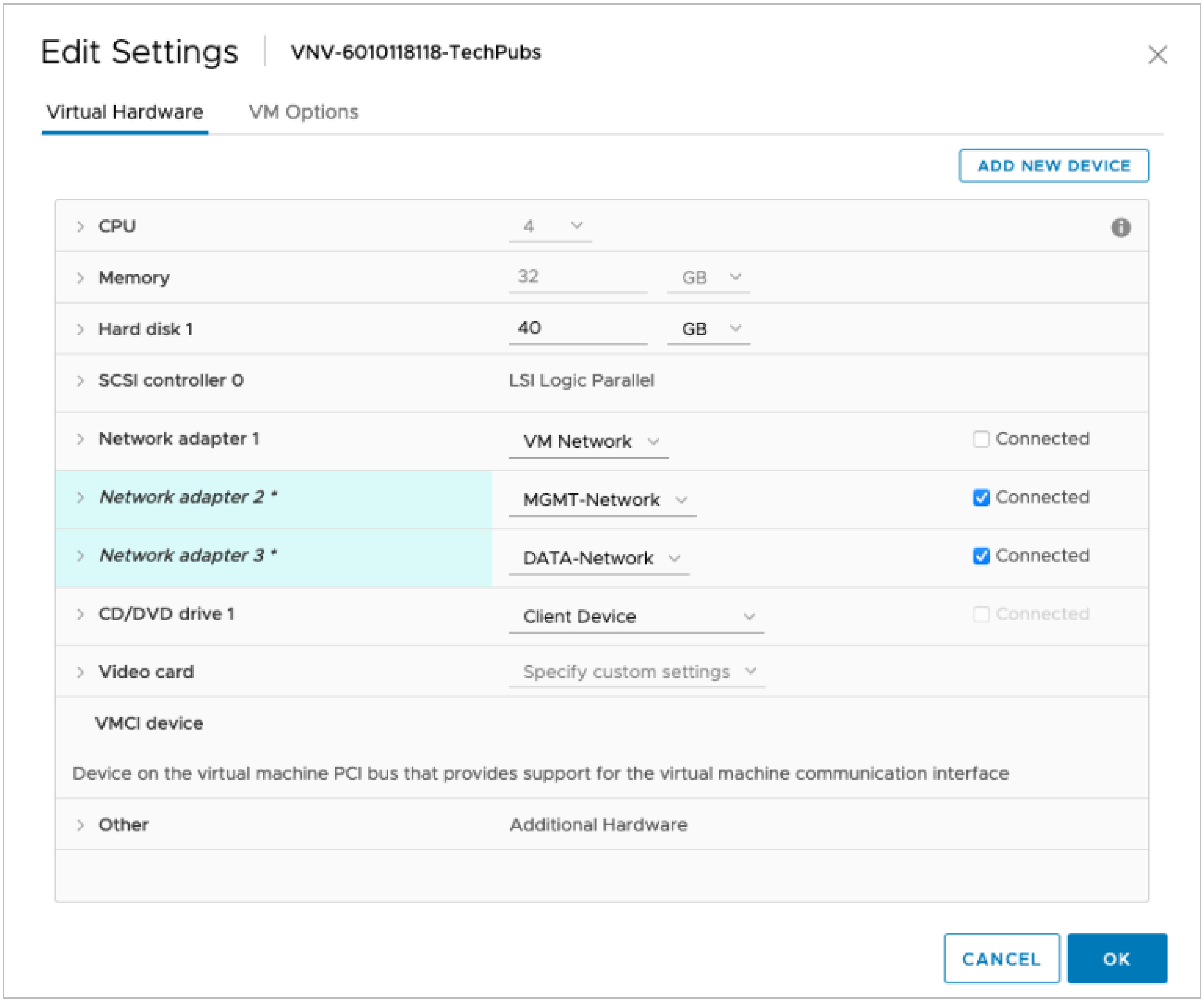
A newer version of VMware Tools is available for this virtual machine. VMware Tools...

VM Hardware

CPU	4 CPU(s)
Memory	32 GB, 0 GB memory active
Hard disk 1	40 GB
Network adapter 1	VM Network (connected)
Network adapter 2	VM Network (connected)
Network adapter 3	VM Network (connected)
CD/DVD drive 1	Disconnected
Video card	4 MB
VMCI device	
Device on the virtual machine PCI bus that provides support for the virtual machine communication interface	
Other	Additional Hardware

Edit Settings

Deploying Virtual Networks with vNV (cont'd)



Edit Settings | VNV-6010118118-TechPubs

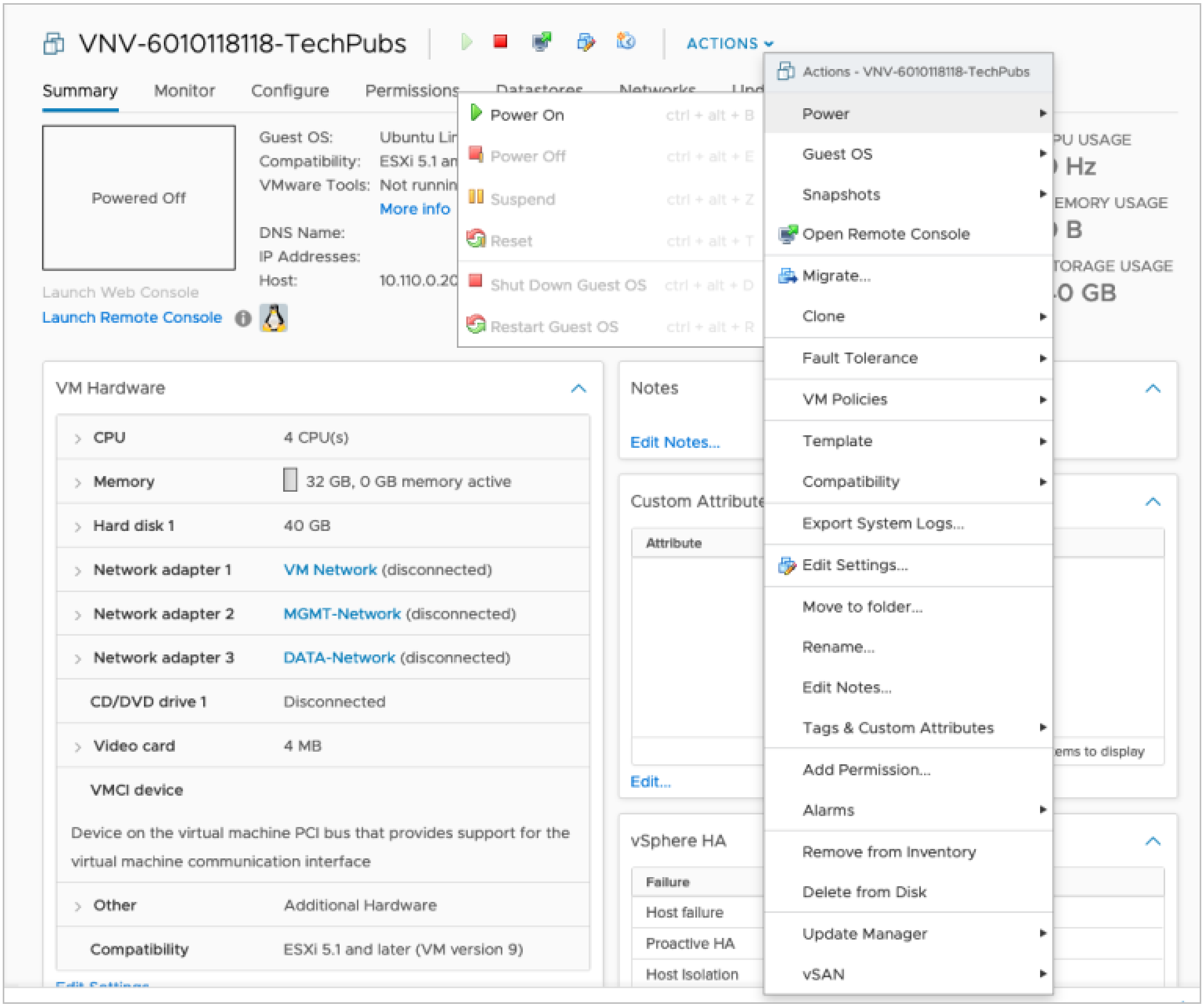
CPU	4		
Memory	32	GB	
Hard disk 1	40	GB	
SCSI controller 0	LSI Logic Parallel		
Network adapter 1	VM Network		□ Connected
*Network adapter 2 **	MGMT-Network		☑ Connected
*Network adapter 3 **	DATA-Network		☑ Connected
CD/DVD drive 1	Client Device		□ Connected
Video card	Specify custom settings		
VMCI device			
Other	Additional Hardware		

Device on the virtual machine PCI bus that provides support for the virtual machine communication interface

Network Settings

Deploying Virtual Networks with vNV (cont'd)

Step 10 **Power On** your **vNV VM**.



Guest OS: Ubuntu Lin
Compatibility: ESXi 5.1 an
VMware Tools: Not runnin
More info
DNS Name:
IP Addresses:
Host: 10.110.0.20

VM Hardware

CPU	4 CPU(s)
Memory	32 GB, 0 GB memory active
Hard disk 1	40 GB
Network adapter 1	VM Network (disconnected)
Network adapter 2	MGMT-Network (disconnected)
Network adapter 3	DATA-Network (disconnected)
CD/DVD drive 1	Disconnected
Video card	4 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 5.1 and later (VM version 9)

Power on VM

Deploying Virtual Networks with vNV (cont'd)

After a period of time, the VM is booted and Netvisor is ready to run and form the fabric.

Step 11 Ensure fabric nodes are ready before joining the fabric with **vNV** using:

```
CLI (network-admin@switch) > fabric-node-show
```

An example of an output is shown below:

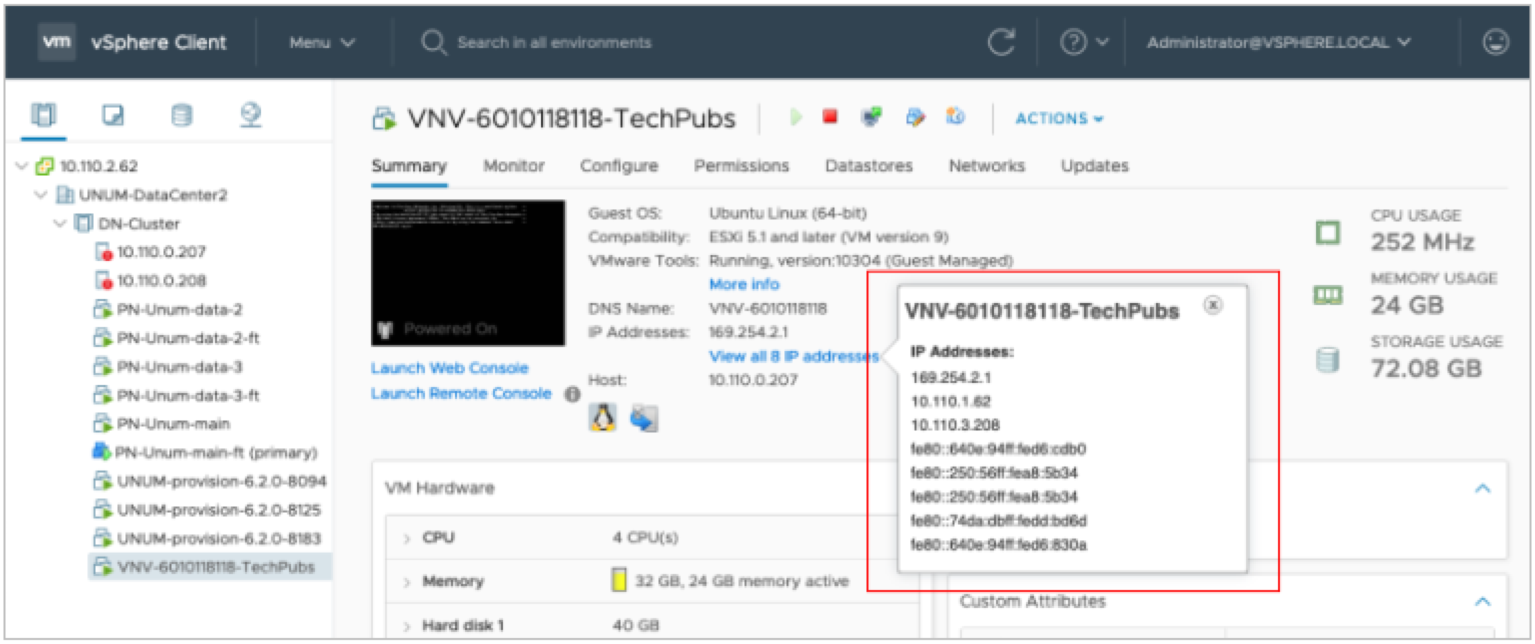
```
CLI (network-admin@sw11) > fabric-node-show format name,fab-name,mgmt-ip,in-band-ip,fab-
tid,version,state,device-state

name fab-name   mgmt-ip        in-band-ip   fab-tid version                            state
           device-state
---- ---------- -------------- ------------ ------- ---------------------------------
---------------- ------------
sw11 Hybrid-ACF 10.36.10.11/24 10.0.11.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
online           ok
sw12 Hybrid-ACF 10.36.10.12/24 10.0.12.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online ok
sw13 Hybrid-ACF 10.36.10.13/24 10.0.13.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online ok
sw14 Hybrid-ACF 10.36.10.14/24 10.0.14.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online ok
sw15 Hybrid-ACF 10.36.10.15/24 10.0.15.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online ok
sw16 Hybrid-ACF 10.36.10.16/24 10.0.16.1/30 6        5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online ok
CLI (network-admin@sw11) >
CLI (network-admin@sw11) > fabric-info
name:                         Hybrid-ACF
id:                           b000b27:5aa75367
vlan:                         1
fabric-network:               mgmt
control-network:              mgmt
tid:                          6
fabric-advertisement-network:  in-band-mgmt
CLI (network-admin@sw11) >
```

This fabric is a six node fabric using management network.

Step 12 Login to your **vNV VM**:

- Using VMware console.
- Using **SSH** if you selected **DHCP** in the management network.

> **Note:** You can find the IP address of your vNV VM by simply looking at the IP Addresses of the VM in your VMware vSphere Web Client. See an example below:



VNV-6010118118-TechPubs

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 5.1 and later (VM version 9)
VMware Tools: Running, version:10304 (Guest Managed)
DNS Name: VNV-6010118118
IP Addresses: 169.254.2.1
Host: 10.110.0.207

CPU USAGE 252 MHz
MEMORY USAGE 24 GB
STORAGE USAGE 72.08 GB

CPU 4 CPU(s)
Memory 32 GB, 24 GB memory active
Hard disk 1 40 GB

VNV-6010118118-TechPubs

IP Addresses:
169.254.2.1
10.110.1.62
10.110.3.208
fe80::640e:94ff:fed6:cdb0
fe80::250:56ff:fea8:5b34
fe80::250:56ff:fea8:5b34
fe80::74da:dbff:fedd:bd6d
fe80::640e:94ff:fed6:830a

Review IP Addresses

> **Note:** The default credentials to connect to your **vNV** are: **network-admin/admin**

Deploying Virtual Networks with vNV (cont'd)

Step 13 Run the initial setup on **vNV**:

```
johndoe-pc:~ johndoe$ ssh network-admin@10.36.10.169
The authenticity of host '10.36.10.169 (10.36.10.169)' can't be established.
ECDSA key fingerprint is SHA256:5+RNHHFaWYJda15+0qJGB4VGMLmsqOo04h0GHeVTLGo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.36.10.169' (ECDSA) to the list of known hosts.
Last login: Fri Mar 16 22:32:34 2018 from thomas-pc.pluribusnetworks.com
Netvisor OS Command Line Interface 5.1
      By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU HAVE READ THE TERMS OF THE
PLURIBUS NETWORKS END USER LICENSE AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?:
YES
Switch setup required:
      Switch Name (vnv1): vnv1
      network-admin Password:
      Re-enter Password:
      Mgmt IP/Netmask (dhcp):
      Mgmt IPv6/Netmask:
      In-band IP/Netmask (10.0.167.1/30): 10.0.169.1/30
      In-band IPv6/Netmask:
      Gateway IP (10.36.10.254):
      Gateway IPv6:
      Primary DNS IP (10.34.30.1):
      Secondary DNS IP (10.34.30.2):
      Domain name (tme.pluribusnetworks.lab):
      Automatically Upload Diagnostics (yes):
      Enable host ports by default (yes):
Switch Setup:
      Switch Name          : vnv1
      Switch Mgmt IP       : 10.36.10.169/24
      Switch Mgmt IPv6     : fe80::640e:94ff:fe68:a274/64
      Switch In-band IP    : 10.0.169.1/30
      Switch In-band IPv6  : fe80::640e:94ff:fe68:92ee/64
      Switch Gateway       : 10.36.10.254
      Switch IPv6 Gateway  : ::
      Switch DNS Server    : 10.34.30.1
      Switch DNS2 Server   : 10.34.30.2
      Switch Domain Name   : tme.pluribusnetworks.lab
      Switch NTP Server    :
      Switch Timezone      : Etc/UTC
      Switch Date          : 2018-03-16,22:40:41
      Upload Diagnostics   : yes
      Enable host ports    : yes
      Analytics Store      : default
Fabric required. Please use fabric-create/join/show
Connected to Switch vnv1; nvOS Identifier:0x32be0968; Ver: 5.1.0-5010014980
CLI (network-admin@vnv1) >
```

Step 14 Check for the existing fabrics:

```
CLI (network-admin@vnv1) > fabric-show
name         id               vlan fabric-network control-network tid fabric-advertisement-
network
------------ ---------------- ---- -------------- --------------- ---
----------------------------
Hybrid-ACF   b000b27:5aa75367 1    mgmt                           6   in-band-mgmt
fabric-show: Fabric required. Please use fabric-create/join/show
CLI (network-admin@vnv1) >
```

Step 15 Join the fabric:

```
CLI (network-admin@vnv1) > fabric-join name Hybrid-ACF
Joined fabric Hybrid-ACF. Restarting nvOS...
Connected to Switch vnv1; nvOS Identifier:0x32be0968; Ver: 5.1.0-5010014980
CLI (network-admin@vnv1) >
```

Deploying Virtual Networks with vNV (cont'd)

Step 16 Validate **vNV-1** is now part of the management fabric:

```
CLI (network-admin@vnv1) > fabric-node-show format name,fab-name,mgmt-ip,in-band-ip,fab-
tid,version,state,device-state

name fab-name   mgmt-ip         in-band-ip    fab-tid version
state             device-state
---- ---------- --------------- ------------- ------- ---------------------------------
---------------- ------------
vnv1 Hybrid-ACF 10.36.10.169/24 10.0.169.1/30 7       5.1.0-5010014980,#50~14.04.1-Ubuntu
online            ok
sw13 Hybrid-ACF 10.36.10.13/24  10.0.13.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
sw11 Hybrid-ACF 10.36.10.11/24  10.0.11.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
sw12 Hybrid-ACF 10.36.10.12/24  10.0.12.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
sw14 Hybrid-ACF 10.36.10.14/24  10.0.14.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
sw15 Hybrid-ACF 10.36.10.15/24  10.0.15.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
sw16 Hybrid-ACF 10.36.10.16/24  10.0.16.1/30  7       5.1.0-5010014980,#50~14.04.1-Ubuntu
mgmt-only-online  ok
CLI (network-admin@vnv1) >
```

Note: Repeat all these steps to deploy the second vNV to provide HA to the vNET Managers deployed on vNV.

Deploying Virtual Networks with vNV (cont'd)

Deploying vNETs

Step 1 Create a Public-to-Private VLAN Mapping.

To allow independent VLAN IDs per tenant/vNET, the fabric administrator must configure a range of usable VLANs in the default 4k range of VLANs available (Public VLANs) to map to the private VLANs configured in each vNET:

```
CLI (network-admin@vnv1) > switch leaves vnet-public-vlans-modify vlans 1000-3999
```

In this example, the range 1000-3999 is used, which means every Private VLAN created is associated to a Public VLAN in that range.

The following table is an example of the mapping between Public and Private VLANs with the default vNET and two private vNETs created:

vNET	Private VLAN ID	Public VLAN ID
global	10	10
	20	20
mgmt-vnet	10	1000
	20	1001
cust1-vnet	10	1002

Public and Private VLANs

The association between a range of Public VLANs are either reserved per vNET (static assignment) or shared across the different vNETs created.

Deploying Virtual Networks with vNV (cont'd)

Step 2 Create a Private vNET with a vNET Manager on vNV-1.

To create a Private vNET, use the following command:

```
CLI (network-admin@vnv1) > switch vnv1 vnet-create name mgmt-vnet scope fabric vlan-type
private num-private-vlans 100 vxlans 1010001-1014094 vnet-mgr-name mgmt-vnet-mgr1
```

Parameters:

- scope: defines the scope of this vNET. If this vNET must NOT be seen by other switches, don’t create it with a fabric scope, restrict this to a cluster scope or even local.
- vlan-type: defines the type of VLAN used in this vNET, consequently defines the type of vNET (private or public).
- num-private-vlans: defines the number of Private VLANs configured in this vNET. Here it is limited to 100 VLANs.
- vxlans: defines the range of VNIs used by the vNET administrator. In this example, the VNI convention is XXXYYYY:
    - XXX is the tenant ID (101 for mgmt-vnet)
    - YYYY is the VLAN ID (from VLAN 1 to VLAN 4094)
- vnet-mgr-name: defines the name of the vNET Manager created. For vNV-1, as a convention, we use mgmt- vnet-mgr1.

This creates a vNET with a vNET Manager container located on vNV-1.

Step 3 Configure vNET Administrator Credentials.

```
CLI (network-admin@vnv1) > user-modify name mgmt-vnet-admin password password:
```

Once the vNET Manager is created, to customize credentials for the vNET admin, use the following command:

Deploying Virtual Networks with vNV (cont'd)

Step 4 Create a vNET Manager on vNV-2.

```
CLI (network-admin@vnv2) > switch vnv2 vnet-manager-create name mgmt-vnet-mgr2 vnet mgmt-
vnet enable
```

Once the vNET Manager is created, to customize credentials for the vNET admin, use the following command:

This creates a second vNET Manager container, located on vNV-2.

The output below shows two vNET Manager created on 2 different vNVs:

```
CLI (network-admin@vnv1) > vnet-manager-show
name           type     scope  vnet      is-global vnet-service state
-------------- -------- ------ --------- --------- ------------ -------
mgmt-vnet-mgr1 vnet-mgr fabric mgmt-vnet false     shared       enabled
mgmt-vnet-mgr2 vnet-mgr fabric mgmt-vnet false     shared       enabled
CLI (network-admin@vnv1) >
```

Step 5 Configure the vNET Manager Interface on both vNVs.

Option 5a On the Management vNIC.

To provide an IP address to the vNET admin to manage his own vNET through the Management vNIC, use the following command:

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.36.10.201/24 if mgmt vlan 0 vlan-type public
```

Parameters:

- ip: IP address for the vNET Manager container.
- if: interface on the switch to be used to associate this IP with. This is either mgmt or data (front-facing ports). In this example, we'll use the dedicated mgmt port.
- vlan: ID of the VLAN to be used on the port for this vNET Manager interface.
- vlan-type: type of VLAN to be used when a VLAN is used. If private is used, then the Private/Public VLAN mapping defines which VLAN is used in the Private vNET and the Public VLAN space (global vNET). As the mgmt port is used, no VLAN is used ("0") and the VLAN type is public.

> **Note:** It is important to either issue this command on vnv1 directly or specifying switch vnv1 if issued from another switch, otherwise this command outputs an error (as show below).

```
CLI (network-admin@sw11) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.36.10.201/24 if mgmt vlan 0 vlan-type public vnet-manager-interface-add: vlan 0 not found
CLI (network-admin@sw11) >
```

We perform the same operation on the second vNV:

```
CLI (network-admin@vnv2) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr2 ip
10.36.10.202/24 if mgmt vlan 0 vlan-type public
```

The output below shows the two vNET Manager interfaces configured on both vNVs:

```
CLI (network-admin@vnv1) > vnet-manager-interface-show format all layout vertical
vnet-manager-name: mgmt-vnet-mgr1
nic: eth3
ip: 10.36.10.201/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 66:0e:94:ec:38:92
vlan-type: public
if: mgmt
exclusive: no
nic-config: enable
nic-state: up
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal

vnet-manager-name: mgmt-vnet-mgr2
nic: eth4
ip: 10.36.10.202/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 66:0e:94:d8:a2:f5
vlan-type: public
if: mgmt
exclusive: no
nic-config: enable
nic-state: up
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal

CLI (network-admin@vnv1) >
```

Option 5b On the In-band vNIC.

Alternatively, to provide an IP address to the vNET admin to manage his own vNET through the In-band vNIC, use the following command:

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.10.10.1/24 if data vlan 100 vlan-type public
```

Deploying Virtual Networks with vNV (cont'd)

Step 6 Configure a Default Gateway on vNET Managers.

To provide connectivity to the vNET admin, a default gateway is configured in the vNET Manager container:

```
CLI (network-admin@vnv1) > vnet-manager-modify name mgmt-vnet-mgr1 gateway 10.36.10.254
```

Same operation on the second vNV:

```
CLI (network-admin@vnv2) > vnet-manager-modify name mgmt-vnet-mgr2 gateway 10.36.10.254
```

The output below shows the two vNET Managers with a gateway configured on both:

```
CLI (network-admin@vnv1) > vnet-manager-show
name           type     scope  vnet      is-global vnet-service state   gateway
-------------- -------- ------ --------- --------- ------------ ------- ------------
mgmt-vnet-mgr1 vnet-mgr fabric mgmt-vnet false     shared       enabled 10.36.10.254
mgmt-vnet-mgr2 vnet-mgr fabric mgmt-vnet false     shared       enabled
CLI (network-admin@vnv1) >
```

```
CLI (network-admin@vnv2) > vnet-manager-show
name           type     scope  vnet      is-global vnet-service state   gateway
-------------- -------- ------ --------- --------- ------------ ------- ------------
mgmt-vnet-mgr1 vnet-mgr fabric mgmt-vnet false     shared       enabled
mgmt-vnet-mgr2 vnet-mgr fabric mgmt-vnet false     shared       enabled 10.36.10.254
CLI (network-admin@vnv2) >
```

Note: In this example, the same network (physical and subnet) is used for both between the default vNET and the mgmt- vnet for the vNET Manager Interface. Another VLAN (part of the mgmt-vnet) could have defined on the Management vNIC of the vNV to have a complete isolation of the management subnets. An example for such scenario is shown below.

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.36.10.102/24 if mgmt vlan 100 vnet mgmt-vnet vlan-type private
```

Deploying Virtual Networks with vNV (cont'd)

An example is shown below for the in-band vNIC:

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.10.10.1/24 if data vlan 200 vnet mgmt-vnet vlan-type private
```

To configure an IP address on the second vNET Manager located on vNV-2, use the following command:

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr2 ip
10.36.10.103/24 if mgmt vlan 0 vlan-type public
```

Step 7 Configure VRRP.

To configure VRRP on a vNET Manager interface, use the following command:

```
CLI (network-admin@vnv1) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr1 ip
10.36.10.200/24 if mgmt vlan 0 vlan-type public vrrp-id 111 vrrp-primary eth3 vrrp-priority
250
```

We perform the same operation on the second vNV:

```
CLI (network-admin@vnv2) > vnet-manager-interface-add vnet-manager-name mgmt-vnet-mgr2 ip
10.36.10.200/24 if mgmt vlan 0 vlan-type public vrrp-id 111 vrrp-primary eth4 vrrp-priority
240
```

Note: This configuration only applies to the case where the 2 vNVs are in the same bridge/broadcast domain. The output below shows the two vNET Manager interfaces configured on both vNVs:

Deploying Virtual Networks with vNV (cont'd)

```
CLI (network-admin@vnv1) > vnet-manager-interface-show format all layout vertical
vnet-manager-name: mgmt-vnet-mgr1
nic: eth3
ip: 10.36.10.201/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 66:0e:94:ec:38:92
vlan-type: public
if: mgmt exclusive: no
nic-config: enable
nic-state: up
is-primary: true
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal

vnet-manager-name: mgmt-vnet-mgr1
nic: eth8
ip: 10.36.10.200/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 00:00:5e:00:01:6f
vlan-type: public
if: mgmt exclusive: no
nic-config: enable
nic-state: up
is-vip: true
vrrp-id: 111
vrrp-primary: eth3
vrrp-priority: 250
vrrp-adv-int(ms): 1000
vrrp-state: master
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal
```

Deploying Virtual Networks with vNV (cont'd)

```
vnet-manager-name: mgmt-vnet-mgr2
nic: eth4
ip: 10.36.10.202/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 66:0e:94:d8:a2:f5
vlan-type: public
if: mgmt exclusive: no
nic-config: enable
nic-state: up
is-primary: true
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal

vnet-manager-name: mgmt-vnet-mgr2
nic: eth7
ip: 10.36.10.200/24
assignment: static
assignment2: none
assignment-linklocal: none
mac: 00:00:5e:00:01:6f
vlan-type: public
if: mgmt exclusive: no
nic-config: enable
nic-state: down
is-vip: true
vrrp-id: 111
vrrp-primary: eth4
vrrp-priority: 240
vrrp-adv-int(ms): 1000
vrrp-state: slave
mtu: 1500
sriov-vf: false
mirror-traffic: false
if-nat-realm: internal
CLI (network-admin@vnv1) >
```

Deploying Virtual Networks with vNV (cont'd)

With this configuration, the vNET Admin can login on his own slice of the fabric by using SSH on the IP address 10.36.10.200:

```
mgmt-pc:~ admin$ ssh mgmt-vnet-admin@10.36.10.200
mgmt-vnet-admin@10.36.10.200's password:
Last login: Wed Jul 25 09:09:16 2018 from 10.10.10.10
Netvisor OS Command Line Interface 5.1
Connected to Switch vnv1; nvOS Identifier:0x680783ec; Ver: 5.1.0-5010014980
CLI (mgmt-vnet-admin@vnv1) >
CLI (mgmt-vnet-admin@vnv1) > show vlan
CLI (mgmt-vnet-admin@vnv1) >
CLI (mgmt-vnet-admin@vnv1) > vnet-show
switch name      scope  vlan-type vlans public-vlans num-private-vlans vxlans
managed-ports shared-ports admin
------ --------- ------ --------- ----- ------------ ----------------- ---------------
------------- ------------ ---------------
sw11   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw12   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw13   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw14   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw15   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw16   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
sw18   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
vnv1   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
vnv2   mgmt-vnet fabric private   none  none         100               1010001-1014094 none
         none          mgmt-vnet-admin
CLI (mgmt-vnet-admin@vnv1) >
```

In this output, we can see when we SSH to the VIP, we end up connected to the master VRRP, in this example, vnv1.

Configure vNV in a Fabric over L3 Scenario

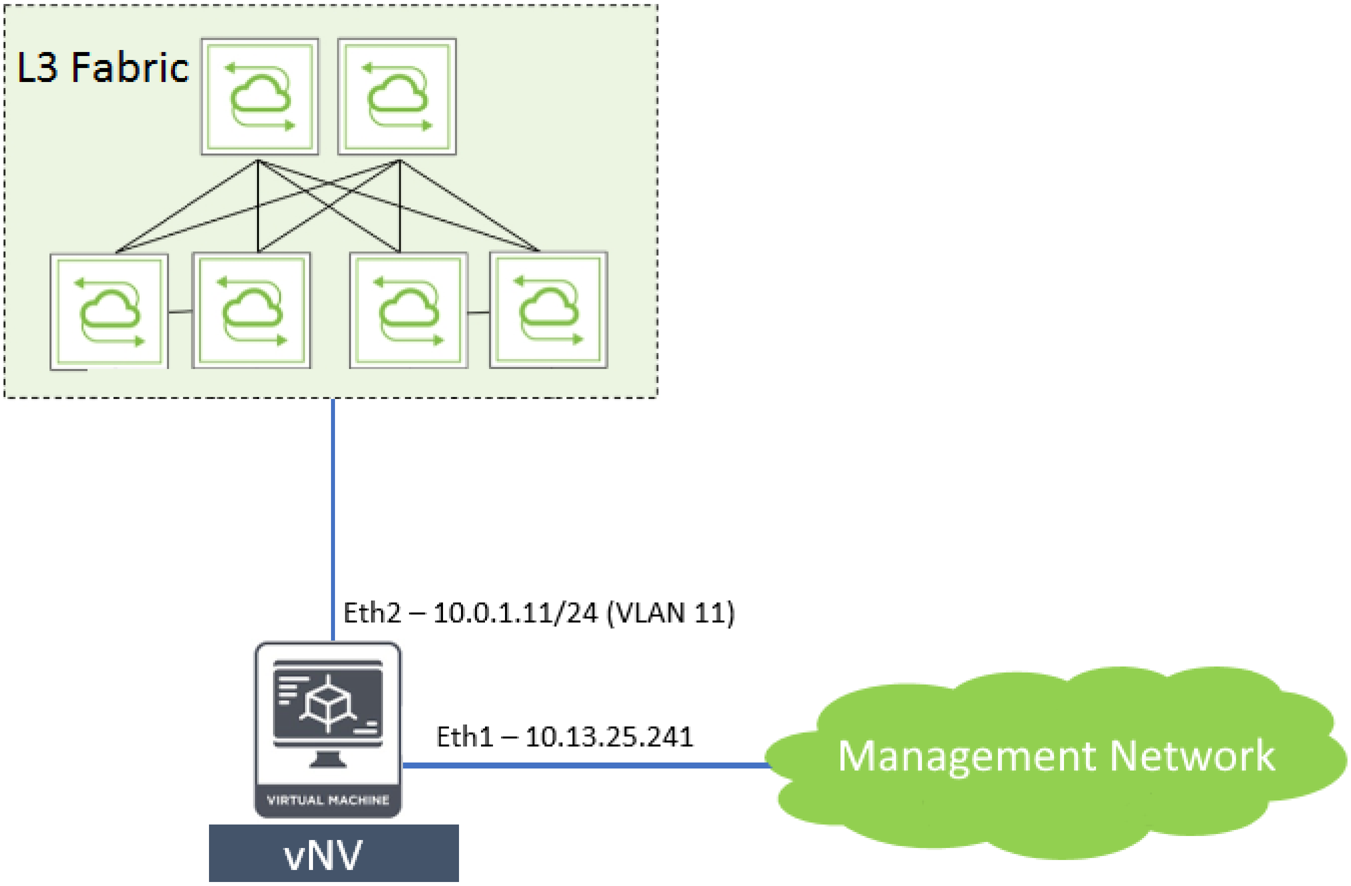
Configure vNV - L3 Scenario

Virtual Netvisor (vNV) can be used in conjunction with UNUM to create a seed switch.

Install vNV on the same ESXi server as UNUM, and it reduces the impact of UNUM polling on physical switches.

This section details how to configure vNV in a fabric over an L3 scenario.

Topology



Topology Layer 3

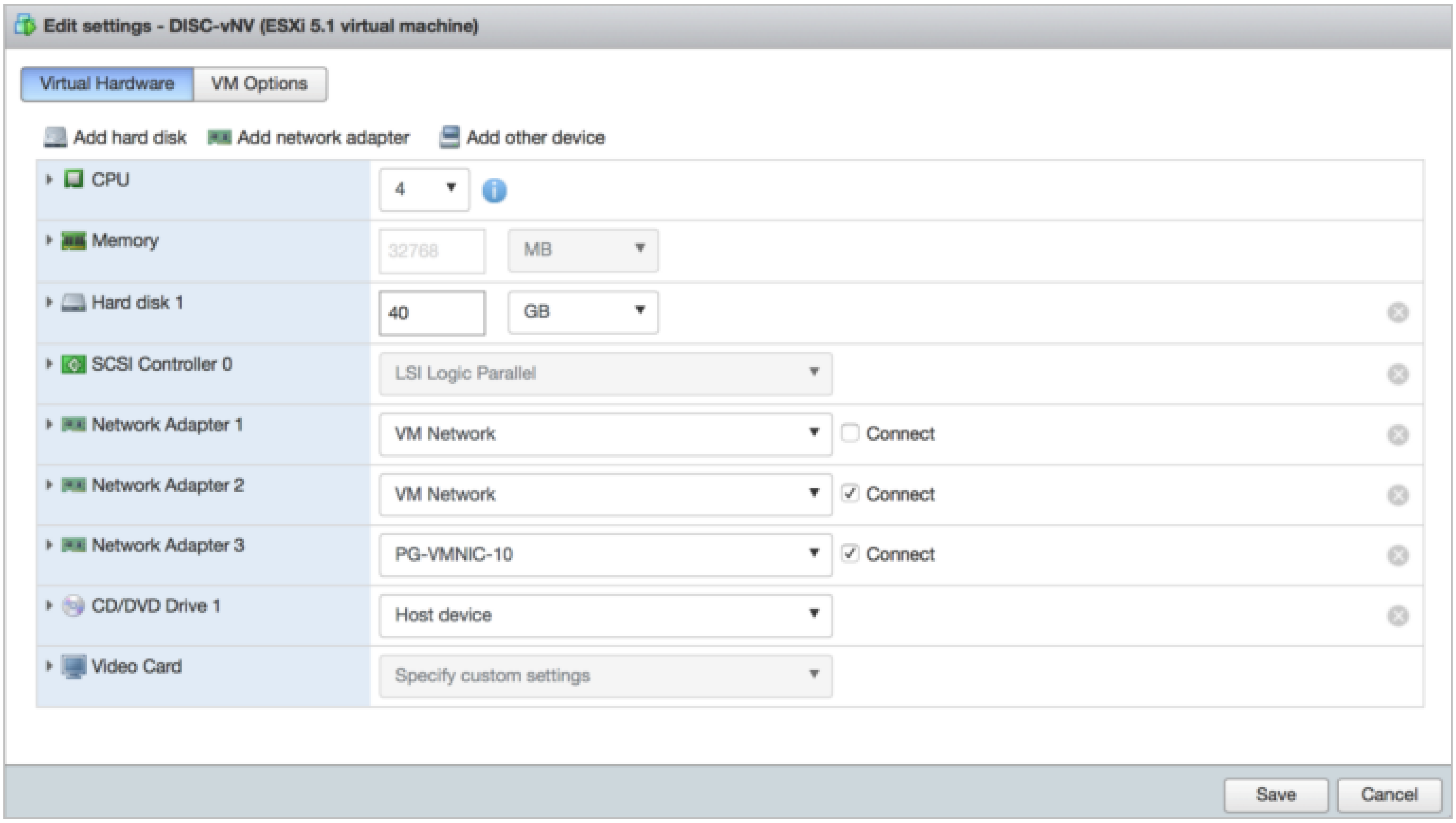
Use the above topology for this demonstration. Here, switches use VLAN 4000 for fabric communication (by default, we use VLAN 1 for fabric communication).

VLAN 11 connected via eth2 (Network Adapter 3) is used to communicate with the rest of the fabric.

VM Properties

A vNV has three interfaces, Eth0, Eth1, and Eth2. In the VM, these interfaces are Network Adapter 1, Network Adapter 2, and Network Adapter 3, respectively.

Eth0 (Network Adapter 1) is not connected anywhere in the VM configuration. Eth1 (Network Adapter 2) and Eth2 (Network Adapter 3) are for management and in-band communication, as illustrated in the image below.



Edit settings - DISC-vNV (ESXi 5.1 virtual machine)

CPU	4	
Memory	32768 MB	
Hard disk 1	40 GB	
SCSI Controller 0	LSI Logic Parallel	
Network Adapter 1	VM Network	□ Connect
Network Adapter 2	VM Network	☑ Connect
Network Adapter 3	PG-VMNIC-10	☑ Connect
CD/DVD Drive 1	Host device	
Video Card	Specify custom settings	

VMware NIC Settings

Configuring vNV in a Fabric over L3 Scenario (Cont'd)

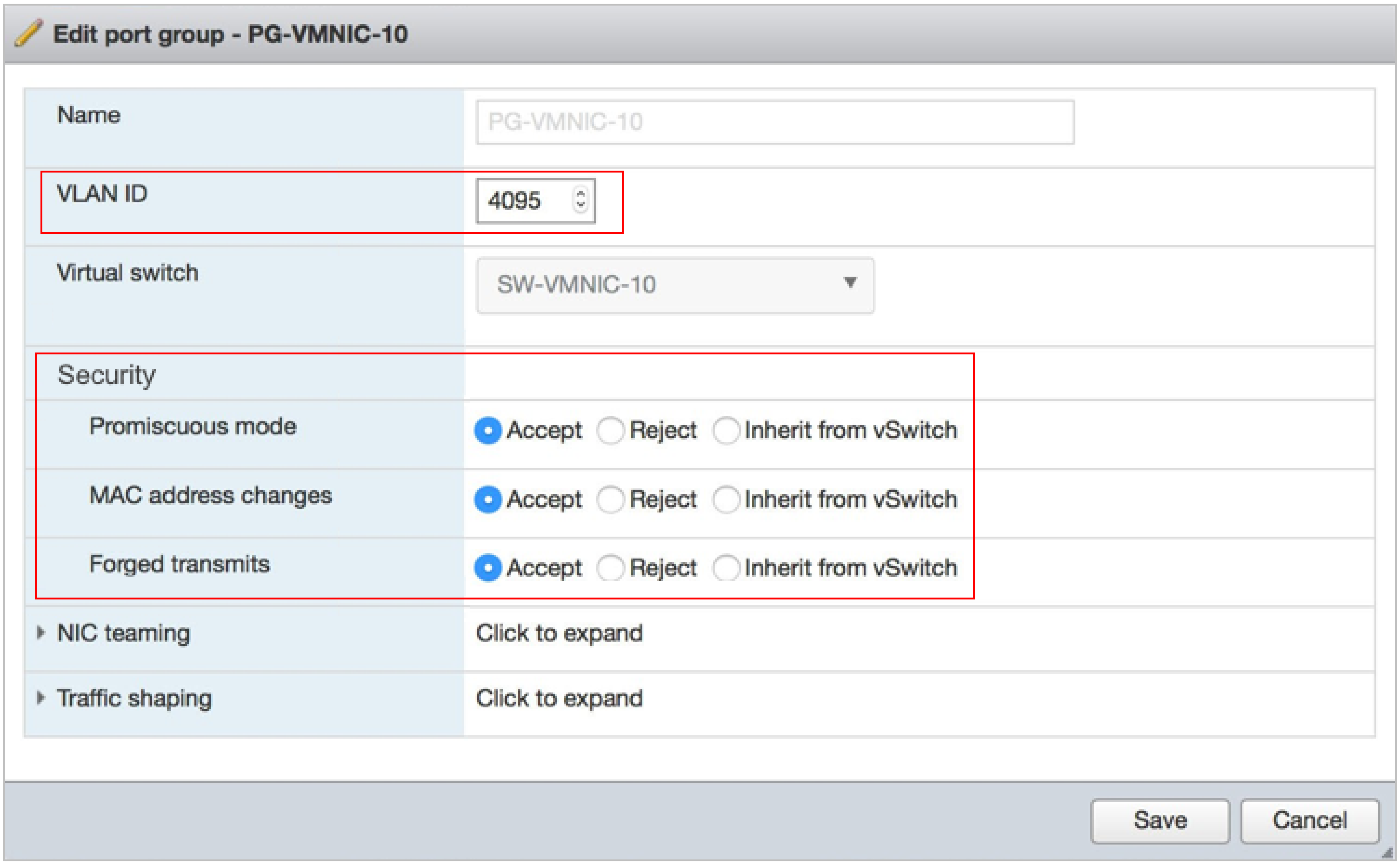
Port Group Configuration

Following Port Groups (PG) properties are selected on PGs that are associated to VM interfaces Network Adapter 2 and Network Adapter 3.

VLAN ID - 4095

Security → Promiscuous mode – Accept
Security → MAC address Changes – Accept
Security → Forged Transmits - Accept

The figure below demonstrates the same.



Edit port group - PG-VMNIC-10

Name	PG-VMNIC-10
VLAN ID	4095
Virtual switch	SW-VMNIC-10
Security	
Promiscuous mode	● Accept ○ Reject ○ Inherit from vSwitch
MAC address changes	● Accept ○ Reject ○ Inherit from vSwitch
Forged transmits	● Accept ○ Reject ○ Inherit from vSwitch
NIC teaming	Click to expand
Traffic shaping	Click to expand

VMware Port Group Security Settings

vNV Configuration

You need to configure vNV to communicate with other switches in the fabric and to ensure vNV's in-band network is reachable from the rest of the fabric.

Management/in-band Shell Configuration

The following is the vmgmt0 (mgmt) and vdata0 (in-band) interface configuration of vNV.

From below snippet, `10.20.11.11/24` is associated with `vdata0` (in-band) and `10.13.25.241/23` is associated with vmgmt0 (mgmt).

```
vdata0
Link encap: Ethernet
HWaddr  66:0e:94:f9:53:ba
inet addr: 10.20.11.11
Bcast:  10.20.11.255
Mask:  255.255.255.0

vmgmt0
Link encap: Ethernet
HWaddr  66:0e:94:f9:ac:0a
inet addr: 10.13.25.241
Bcast:  10.13.25.255
Mask:  255.255.254.0
```

vNV CLI Configuration

VLAN Configuration:

```
CLI (network-admin@DISC-vNV*) > running-config-show | grep vlan-create
CLI (network-admin@DISC-vNV*) > vlan-create id 11 replicators none scope local description
vlan-11 ports 1-63,65-271
CLI (network-admin@DISC-vNV*) > vlan-create id 4000 replicators none scope local description
vlan-4000 ports 1-271
```

Here VLAN 11 is used to connect to the rest of the fabric through eth2 (Network Adapter 3) and VLAN 4000 is used to patch in-band interface to vRouter zone.

vRouter Configuration

```
CLI (network-admin@DISC-vNV*) > running-config-show | grep vrouter-create vrouter-create
name DISC-vNV location DISC-vNV fabric-comm proto-multi none bgp-bfd-all-if enable
```

Here vRouter type is "fabric-comm" and is needed in case of fabric over L3 configuration.

Configuring vNV in a Fabric over L3 Scenario (Cont'd)

vRouter Interface Configuration

```
CLI (network-admin@DISC-vNV*) > running-config-show | grep vrouter-interface-add
CLI (network-admin@DISC-vNV*) > vrouter-interface-add vrouter-name DISC-vNV nic eth1.11 ip
10.0.1.11/24 assignment2 none vlan 11
CLI (network-admin@DISC-vNV*) > vlan-type public if data if-nat-realm internal
CLI (network-admin@DISC-vNV*) > vrouter-interface-add vrouter-name DISC-vNV nic eth0.4000 ip
10.20.11.100/24 assignment2 none vlan 4000 vlan-type public if data fabric-nic if-nat-realm
internal
```

The VLAN-11 interface communicates with the fabric, and the VLAN-4000 interface to reach the switch in-band IP.

The VLAN-4000 interface IP is in the same subnet as switch in-band IP and that interface is “fabric-nic.”

Switch Route Configuration

Switch routes are configured to reach in-band IPs of other switches from the global zone.

```
CLI (network-admin@DISC-vNV*) > running-config-show | grep switch-route-create
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.0.1.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.1.0/30 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.2.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.3.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.4.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.5.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.6.0/24 gateway-ip
10.20.11.100
CLI (network-admin@DISC-vNV*) > switch-route-create network 10.20.7.0/24 gateway-ip
10.20.11.100
```

Networks defined here are the in-band networks of other switches in the fabric and `10.20.11.100` is the VLAN-4000 IP configured on the vNV vRouter.

Configuring vNV in a Fabric over L3 Scenario (Cont'd)

vRouter Static Route Configuration

```
CLI (network-admin@DISC-vNV*) > running-config-show | grep vrouter-static
CLI (network-admin@DISC-vNV*) > vrouter-static-route-add vrouter-name DISC-vNV network
0.0.0.0/0 gateway-ip 10.0.1.1
```

The default route is now `10.0.1.1`, which is on the other end of the VLAN-11 vrouter interface reachable through eth2 (Network Adapter 3).

With this, you are able to reach in-band IPs of switches in the fabric and vNV is able to join the fabric.

Use Cases

Use Cases

The following Use Case section and its examples may be helpful in assisting you with configuring Virtual Netvisor.

Virtual NetVisor Deployment Example

Inband Virtual Netvisor (vNV) Deployment

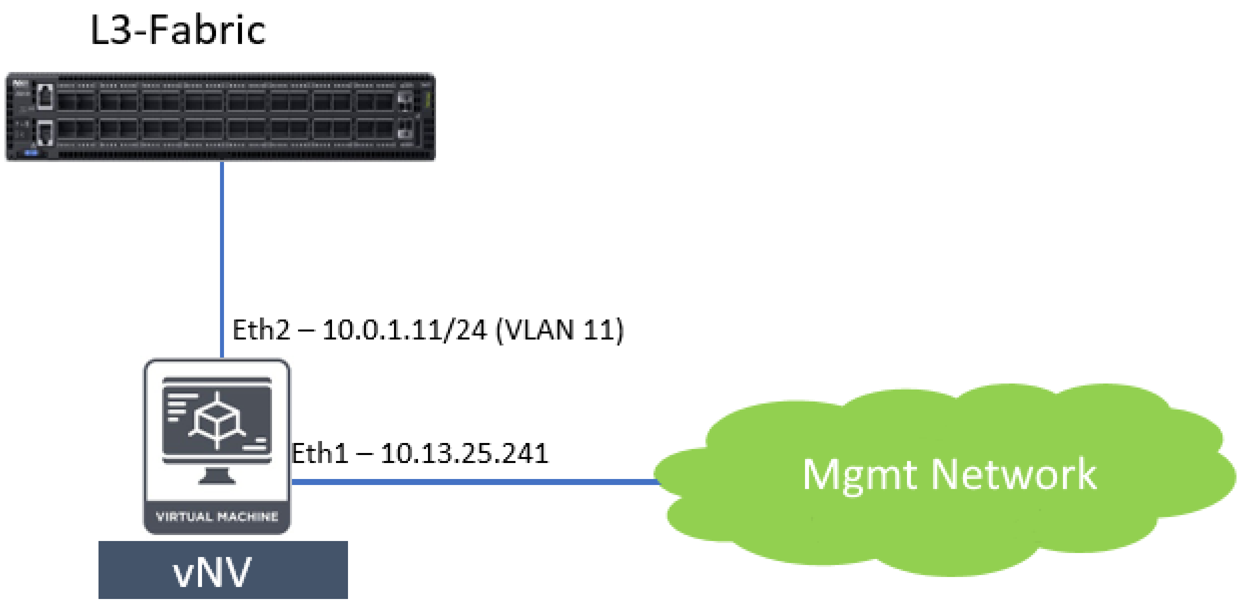
Overview

Typically used in conjunction with UNUM, **Virtual Netvisor (vNV)** deploys as a seed switch.

vNV is used to reduce the impact of UNUM polling on physical switches installs on the same UNUM ESXi server.

This Use Case example demonstrates how to connect vNV through Inband communication.

Topology



As illustrated in the topology example shown above, there are two ways to connect vNV to the fabric through inband.

- vNV uses VLAN 1 to communicate to the Fabric inband network.
- vNV uses a separate Network and uses routing to communicate with the Fabric inband network.

Virtual Netvisor Deployment Use Case (Cont'd)

Virtual Machine (VM) Properties

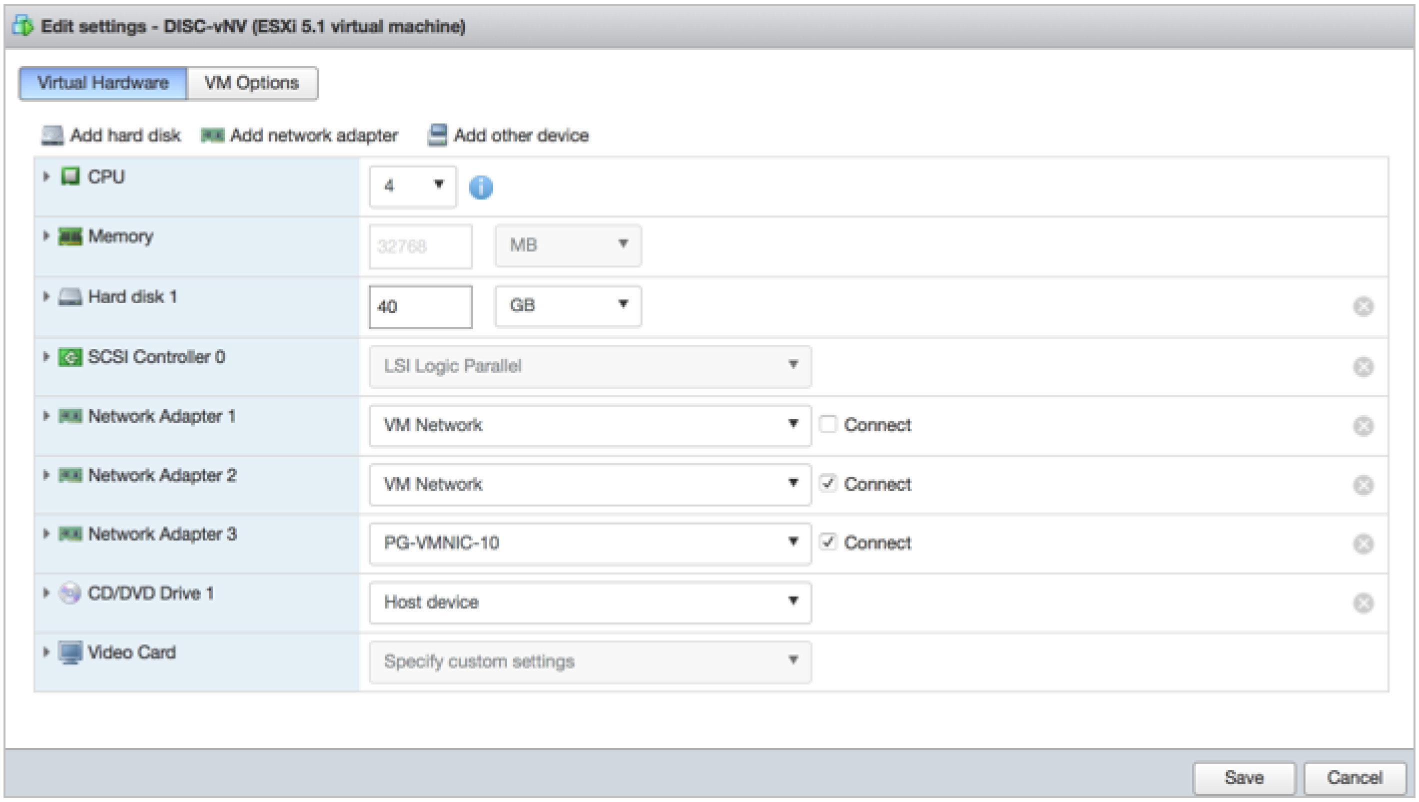
A vNV has three interfaces, **eth0**, **eth1**, and **eth2**.

In the VM, these are identified as “**Network Adapter 1**”, “**Network Adapter 2**”, and “**Network Adapter 3**” respectively

Eth0 (Network Adapter 1) is **not** connected in the VM configuration.

Eth1 (Network Adapter 2) and **eth2** (Network Adapter 3) are for management and inband communication.

The following ESXi Virtual Hardware settings example illustrates the network adapter configuration.



Edit settings - DISC-vNV (ESXi 5.1 virtual machine)

CPU	4	
Memory	32768 MB	
Hard disk 1	40 GB	
SCSI Controller 0	LSI Logic Parallel	
Network Adapter 1	VM Network	☐ Connect
Network Adapter 2	VM Network	☑ Connect
Network Adapter 3	PG-VMNIC-10	☑ Connect
CD/DVD Drive 1	Host device	
Video Card	Specify custom settings	

Virtual Netvisor Deployment Use Case (Cont'd)

Port Group Configuration

Set the following Port Groups (PG) properties associated with the VM interfaces “Network Adapter 2 and “Network Adapter 3.”

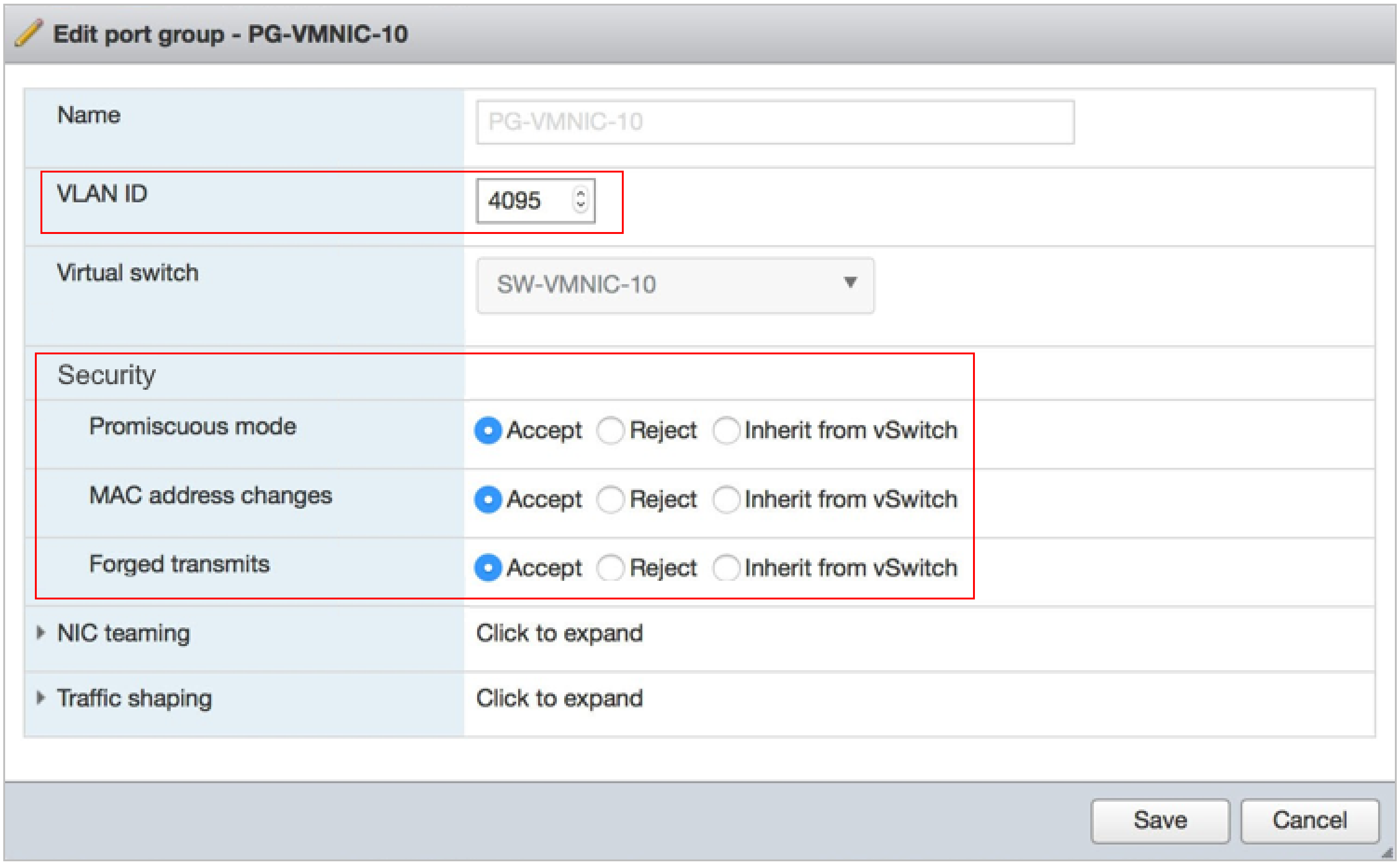
VLAN ID to - 4095

Security Promiscuous mode – Accept

Security MAC address Changes – Accept

Security Forged Transmits - Accept

The following ESXi Edit Port Group settings example illustrates the PG security configuration.



Edit port group - PG-VMNIC-10

Name	PG-VMNIC-10
VLAN ID	4095
Virtual switch	SW-VMNIC-10 ▼
Security	
Promiscuous mode	● Accept ○ Reject ○ Inherit from vSwitch
MAC address changes	● Accept ○ Reject ○ Inherit from vSwitch
Forged transmits	● Accept ○ Reject ○ Inherit from vSwitch
▸ NIC teaming	Click to expand
▸ Traffic shaping	Click to expand

vNV Configuration

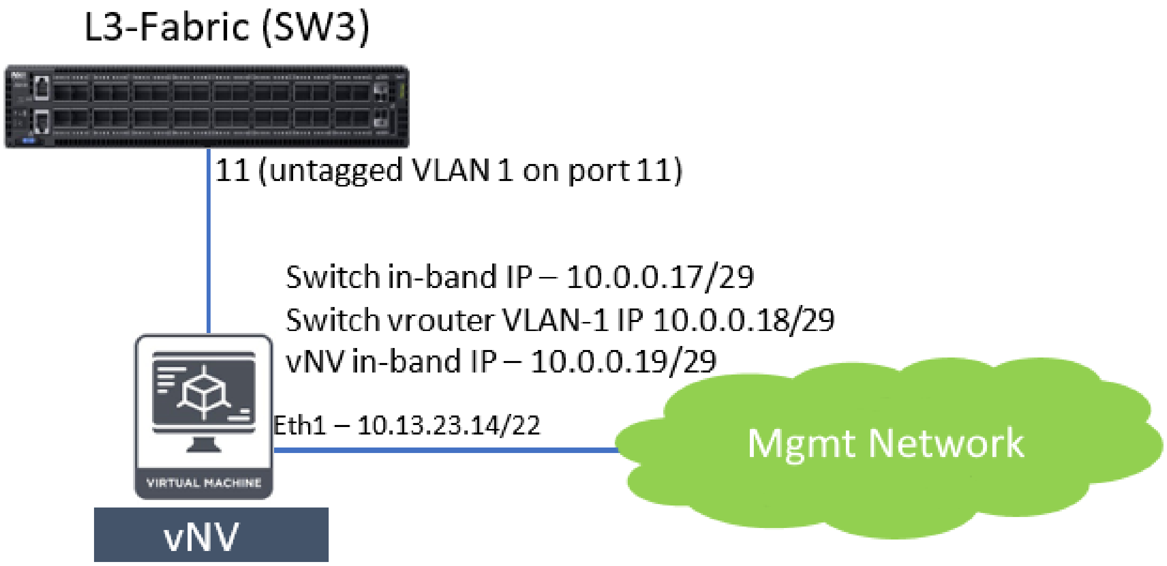
vNV needs to be configured such that it can communicate to inband IPs of other switches in the Fabric and also vNV's in-band network is reachable from the rest of the Fabric. There are two ways to connect vNV to the Fabric through inband:

- vNV uses VLAN 1 to communicate to Fabric inband network
- vNV uses a separate network and routing to communicate with the Fabric inband network.

vNV uses VLAN 1 to communicate with the Fabric Inband Network.

Refer to the following topology used in this scenario. In this example, vNV is connected directly to one of the switches in the Fabric (SW3).

This switch has an in-band IP of `10.0.0.17/29` and a vrouter interface IP of `10.0.0.18/29` in VLAN-1.


L3-Fabric (SW3)
11 (untagged VLAN 1 on port 11)
Switch in-band IP – 10.0.0.17/29
Switch vrouter VLAN-1 IP 10.0.0.18/29
vNV in-band IP – 10.0.0.19/29
Eth1 – 10.13.23.14/22
Mgmt Network
VIRTUAL MACHINE
vNV

1) SW3 is configured with an in-band IP of: `10.0.0.17/29`
2) SW3 is configured with a vrouter interface IP of `10.0.0.18/29` for VLAN-1: `vrouter-interface-add vrouter-name vrouter-SW3 ip 10.0.0.18/29 vlan 1`
3) SW3 has following switch-route created for inband communication: `switch-route-create network 10.0.0.0/24 gateway-ip 10.0.0.18`

Virtual Netvisor Deployment Use Case (Cont'd)

4) VLAN 1 is now configured as untagged on Port 11 of SW3. Run the command: `port-vlan-show ports 11`

```
CLI (network-admin@SW3*) > port-vlan-show ports 11

port           vlans          untagged-vlan   description    active-vlans
----           -----          -------------   -----------    ------------
11             1              1                              1
```

5) vNV running on VM has an in-band IP of `10.0.0.19/29`

6) vNV has following switch-route created for inband communication: `switch-route-create network 10.0.0.0/24 gateway-ip 10.0.0.18`

7) With this configuration vNV should be able to ping ping in-band IPs of other switches in the fabric.

```
CLI (network-admin@vNV-UNUM) > ping 10.0.0.17 PING 10.0.0.17 (10.0.0.17) 56(84)
bytes of data.
64 bytes from 10.0.0.17: icmp_seq=1 ttl=64 time=5.63 ms
64 bytes from 10.0.0.17: icmp_seq=3 ttl=64 time=1.10 ms
^C
--- 10.0.0.17 ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5019ms rtt
min/avg/max/mdev = 1.103/2.038/5.639/1.800 ms
ping: Fabric required. Please use fabric-create/join/show CLI (network-admin@vNV-
UNUM) >
```

Virtual Netvisor Deployment Use Case (Cont'd)

8) Now vNV should be able to join the fabric (in this example: **`test_case`**):

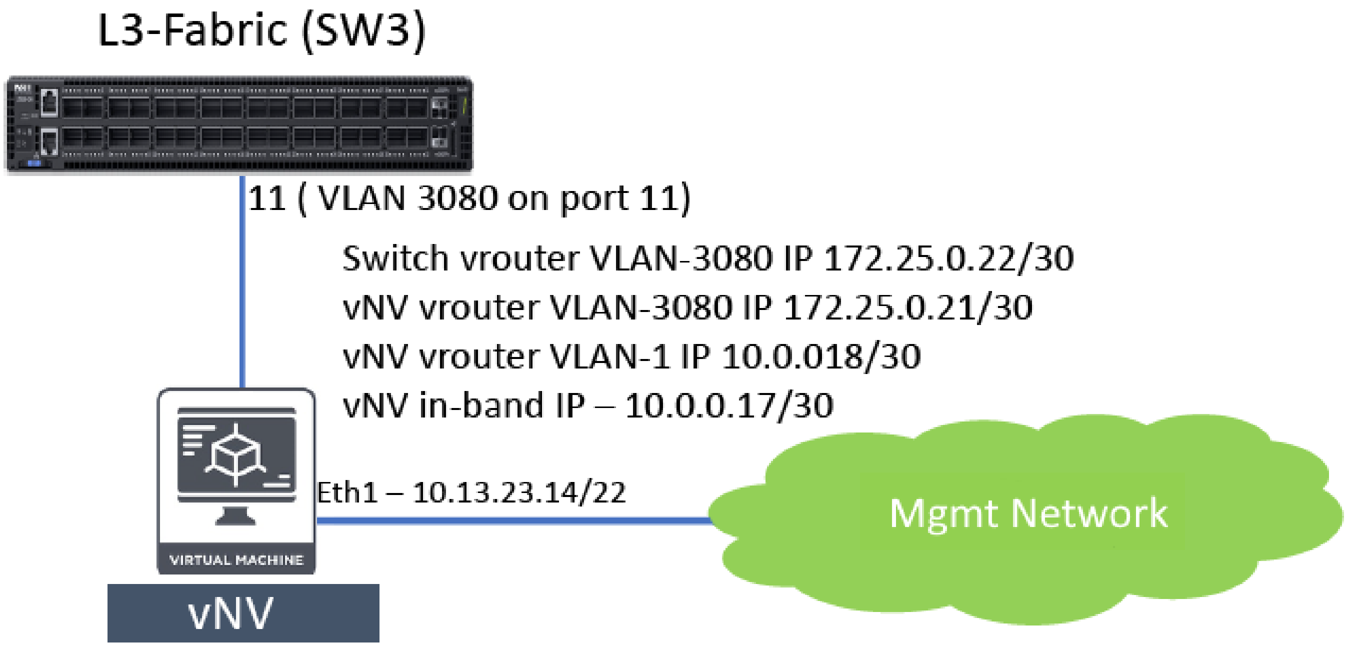
```
CLI (network-admin@vNV-UNUM) > fabric-join switch-ip 10.0.0.17 Joined fabric
test_case. Restarting nvOS...
Connected to Switch vNV-UNUM; nvOS Identifier:0x4db3f218; Ver: 5.2.0- 5020015650
CLI (network-admin@vNV-UNUM) >
CLI (network-admin@vNV-UNUM) > fabric-node-show format name,fab-name,mgmt- ip,in-
band-ip,state,

 name            fab-name        mgmt-ip         in-band-ip      state
 --------        ---------       --------------  ------------    ------
 vNV-UNUM        test_case       10.13.23.14/23  10.0.0.19/29    online
 SW3             test_case       10.13.22.221/23 10.0.0.17/29    online
 SW1             test_case       10.13.22.220/23 10.0.0.1/30     online
 SW4             test_case       10.13.20.221/23 10.0.0.13/30    online
 SW2             test_case       10.13.20.220/23 10.0.0.5/30     online

CLI (network-admin@vNV-UNUM) >
```

vNV uses a Separate Network and Routing to Communicate to the Fabric Inband Network

Refer to the following topology used in this scenario. In this example, use **VLAN 3080** to link vNV to the fabric.

> **Note:** The **vNV Inband IP** is in a different, /30, subnet.

Virtual Netvisor Deployment Use Case (Cont'd)


L3-Fabric (SW3)
11 (VLAN 3080 on port 11)
Switch vrouter VLAN-3080 IP 172.25.0.22/30
vNV vrouter VLAN-3080 IP 172.25.0.21/30
vNV vrouter VLAN-1 IP 10.0.018/30
vNV in-band IP – 10.0.0.17/30
Eth1 – 10.13.23.14/22
VIRTUAL MACHINE
vNV
Mgmt Network

The following steps are needed to accomplish this integration:

1) Configure VLAN 3080 on port 11: `vlan-create id 3080 scope local ports 11`

```
CLI (network-admin@SW3*) > port-vlan-show ports 11 port vlans description active-
vlans

   port                vlans               description         active-vlans
   ----                -----               -----------         ------------
   11                  3080                                    none
```

Virtual Netvisor Deployment Use Case (Cont'd)

2) Create a vrouter interface for VLAN 3080 on Switch SW3: `vrouter-interface-add vrouter-name vrouter-SW3 ip 172.25.0.22/30 vlan 3080`
3) Make sure correct in-band IP, `10.0.0.17/30`, is configured on vNV during initial setup.
4) Create a vrouter on vNV: `vrouter-create name vNV-UNUM fabric-comm`
5) Create VLAN 3080 on vNV: `vlan-create id 3080 scope local ports all`
6) Create a vrouter interface in VLAN 1 for in-band network to reach rest of the fabric on vNV: `vrouter-interface-add vrouter-name vNV-UNUM ip 10.0.0.18/30 vlan 1`
7) Create a vrouter interface in VLAN 3080 for vNV to communicate to SW3: `vrouter-interface-add vrouter-name vNV-UNUM ip 172.25.0.21/30 vlan 3080`

   Now vNV should be able to VLAN 3080 interface of SW3.

```
CLI (network-admin@vNV-UNUM) > vrouter-ping vrouter-name vNV-UNUM host-ip
172.25.0.22
PING 172.25.0.22 (172.25.0.22) 56(84) bytes of data.
64 bytes from 172.25.0.22: icmp_seq=1 ttl=64 time=3.91 ms
64 bytes from 172.25.0.22: icmp_seq=2 ttl=64 time=1.18 ms
^C
--- 172.25.0.22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt
min/avg/max/mdev = 1.185/2.548/3.912/1.364 ms
vrouter-ping: Fabric required. Please use fabric-create/join/show CLI (network-
admin@vNV-UNUM) >
```

8) Create a switch-route on vNV to reach in-band network of other witches in the fabric: `switch-route-create network 10.0.0.0/24 gateway-ip 10.0.0.18`
9) Vrouter default route on vNV pointing SW3 VLAN 3080 vrouter interface: `vrouter-static-route-add vrouter-name vNV-UNUM network 0.0.0.0 netmask 0.0.0.0 gateway-ip 172.25.0.22`

Virtual Netvisor Deployment Use Case (Cont'd)

10) Vrouter static route on SW3 to reach vNV in-band networking pointing to vNV VLAN 3080 vrouter interface: `vrouter-static-route-add vrouter-name vrouter-SW3 network 10.0.0.16/30 gateway-ip 172.25.0.21`

> **Note:** Configure the SW3 vrouter-ospf to redistribute static.

11) At this point vNV should be able to ping in-band IPs of switches in the fabric.

```
CLI (network-admin@vNV-UNUM) > ping 10.0.0.9 PING 10.0.0.9 (10.0.0.9) 56(84)
bytes of data.
64 bytes from 10.0.0.9: icmp_seq=1 ttl=62 time=2.33 ms
64 bytes from 10.0.0.9: icmp_seq=2 ttl=62 time=1.23 ms
^C
--- 10.0.0.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms

rtt min/avg/max/mdev = 1.237/1.786/2.335/0.549 ms
ping: Fabric required. Please use fabric-create/join/show CLI (network-admin@vNV-
UNUM) >
```

12) vNV should be to join the fabric now (in this example: **`test_case`**).

```
CLI (network-admin@vNV-UNUM) > fabric-join switch-ip 10.0.0.9 Joined fabric
test_case. Restarting nvOS...
Connected to Switch vNV-UNUM; nvOS Identifier:0x4db3f218; Ver: 5.2.0- 5020015650

CLI (network-admin@vNV-UNUM) > fabric-node-show format name,fab-name,mgmt- ip,in-
band-ip,state,

name            fab-name        mgmt-ip         in-band-ip      state
--------        ---------       --------------  ------------    ------
vNV-UNUM        test_case       10.13.23.14/23  10.0.0.17/30    online
SW3             test_case       10.13.22.221/23 10.0.0.9/30     online
SW1             test_case       10.13.22.220/23 10.0.0.1/30     online
SW4             test_case       10.13.20.221/23 10.0.0.13/30    online
SW2             test_case       10.13.20.220/23 10.0.0.5/30     online
```

Switch Configuration Tweaks to Connect to UNUM

1) Disable web admin service:

   ```
   admin-service-modify if mgmt no-web
   admin-service-modify if data no-web
   ```

2) Tweak cos3 buffer limit to 500

   Redirect analytic traffic to an unused cos queue (in this example `cos3`) and set the rate limit to prevent it from overrunning CPU: `port-cos-rate-setting-modify port control-port cos3-rate 500`

3) Enable analytics, if disabled and redirect to cos queue: `connection-stats-settings-modify enable`

   ```
   debug-nvOS set-level flow
   vflow-system-modify name System-S flow-class class3 vflow-system-modify name
   System-F flow-class class3 vflow-system-modify name System-R flow-class class3
   debug-nvOS unset-level flow
   ```

4) **Repeat** above steps on **all** switches in the fabric.
5) Enable web admin service on vNV: `admin-service-modify if mgmt web`
6) Disable analytics on vNV: `connection-stats-settings-modify disable`

At this point, vNV is ready to act as a seed switch for UNUM to collect analytics.

Notes and Observations

Notes and Observations

Please use this area for your notes and observations about your use of Arista Networks Virtual Networks with Virtual NetVisor.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	