# Pluribus Networks

# Virtual Networks with Virtual Netvisor

# Deployment Guide

Version 7.0.1

July 2022

# Table of Contents

## Legal Notice

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR PLURIBUS NETWORKS REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE ARE PROVIDED “AS IS” WITH ALL FAULTS. PLURIBUS NETWORKS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL PLURIBUS NETWORKS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA, ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF PLURIBUS NETWORKS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

# EULA

## End-User License Agreement ("EULA") Pluribus Networks, Inc.

IMPORTANT: PLEASE READ THIS END USER LICENSE AGREEMENT CAREFULLY. IT IS VERY IMPORTANT THAT YOU CONFIRM THAT YOU ARE PURCHASING PLURIBUS SOFTWARE OR EQUIPMENT FROM AN APPROVED SOURCE AND THAT YOU, OR THE ENTITY YOU REPRESENT (COLLECTIVELY, THE "CUSTOMER") HAVE BEEN REGISTERED AS THE END USER FOR THE PURPOSES OF THIS PLURIBUS END USER LICENSE AGREEMENT. IF YOU ARE NOT REGISTERED AS THE END USER YOU HAVE NO LICENSE TO USE THE SOFTWARE AND THE LIMITED WARRANTY IN THIS END USER LICENSE AGREEMENT DOES NOT APPLY TO YOU.

1. **Introduction**
   - 1.1. Your use of the Pluribus Networks software ("SOFTWARE") and related documentation ("DOCUMENTATION") is subject to this legal agreement between you and PLURIBUS. "PLURIBUS" means Pluribus Networks, Inc., whose principal place of business is at 6001 America Center Drive, Suite 450, San Jose, CA 95002. This document explains how the agreement is made up, and sets out the terms of the agreement.
   - 1.2. SOFTWARE means software provided to you by PLURIBUS.
   - 1.3. DOCUMENTATION means written information (whether contained in user or technical manuals, training materials, specifications or other such materials provided to you by PLURIBUS) related to the SOFTWARE.
   - 1.4. Unless otherwise agreed to in writing with PLURIBUS, your agreement with PLURIBUS at it relates to the SOFTWARE and DOCUMENTATION will always include, at a minimum, the terms and conditions set out in this EULA.
   - 1.5. This agreement forms a legally binding agreement between you and PLURIBUS in relation to your use of the SOFTWARE and DOCUMENTATION. It is important that you take the time to read this EULA carefully. Collectively, this legal agreement is referred to below as the "EULA."

2. **Accepting this EULA**
   - 2.1. In order to use the SOFTWARE and DOCUMENTATION, you must first agree to this EULA. You may not use the SOFTWARE and DOCUMENTATION if you do not accept this EULA.
   - 2.2. You can accept this EULA by: (A) clicking to accept or agree to this EULA, where this option is made available to you by PLURIBUS; or (B) by actually downloading, installing or using the SOFTWARE, in which case, you understand and agree that PLURIBUS will treat your use of the SOFTWARE as acceptance of this EULA from that point onward.
   - 2.3. If you are accepting this EULA on behalf of another legal entity, you represent that you have the authority to bind such legal entity.
   - 2.4. You may not use the SOFTWARE and DOCUMENTATION and may not accept this EULA if (A) you are not of legal age to form a binding contract with PLURIBUS, or (B) you are a person barred from receiving the SOFTWARE under the laws of the United States or other countries including the country in which you are resident or from which you use the SOFTWARE.
   - 2.5. PLURIBUS is willing to license this software to you only upon the condition that you purchased the software from an Approved Source and that you accept all of the terms contained in this EULA plus any additional limitations on the license set forth in a supplemental license agreement if any accompanying the product or available at the time of your order (collectively the "agreement"). To the extent of any conflict between the terms of this EULA and any supplemental license agreement, the supplemental license agreement shall apply.
   - 2.6. You may access a copy of this EULA at www.pluribusnetworks.com/EULA

## End User License Agreement (cont'd)

3. **Provision of the SOFTWARE AND DOCUMENTATION by PLURIBUS**
   - 3.1. You acknowledge and agree that the form and nature of the SOFTWARE which PLURIBUS provides may change from time to time without prior notice toyou.
   - 3.2. You acknowledge and agree that PLURIBUS assumes no responsibility for any inaccuracies in the DOCUMENTATION and that PLURIBUS reserves the right to change, modify, or otherwise revise the DOCUMENTATION withoutnotice.
   - 3.3. As part of this continuing innovation, you acknowledge and agree that PLURIBUS may stop (permanently or temporarily) providing the SOFTWARE (or any features within the SOFTWARE) to you or to customers generally, at PLURIBUS' sole discretion, with 30 (thirty) days prior notice to you.

4. **Use of the SOFTWARE by you:**
   - 4.1. You agree to use the SOFTWARE in accordance with this EULA.
   - 4.2. You agree not to access (or attempt to access) the SOFTWARE by any means other than through the command line interface that is provided by PLURIBUS, unless you have been specifically allowed to do so in a separate agreement with PLURIBUS.
   - 4.3. By downloading, installing, or using the software, you are representing that you purchased the software from an Approved Source and binding yourself to the agreement. If you do not agree to all of the terms of the agreement, then PLURIBUS is unwilling to license the software to you and (a) you may not download, install or use the software, and (b) you may return the software (including any unopened CD package and any written materials) for a full refund, or, if the software and written materials are supplied as part of another product, you may return the entire product for a full refund. Your right to return and refund expires 30 days after receipt of such product from an approved source, and applies only if you are the original and registered end user purchaser. For the purposes of this EULA, an "Approved Source" means (a) PLURIBUS; or (b) a distributor or systems integrator authorized by PLURIBUS to distribute/ sell PLURIBUS equipment, software and services within your territory to end users; or (c) a reseller authorized by any such distributor or systems integrator in accordance with the terms of the distributor's agreement with PLURIBUS to distribute/sell the PLURIBUS equipment, software and services within your territory to end users.
   - 4.4. You agree that you will not engage in any activity that interferes with or disrupts the operation of the SOFTWARE.
   - 4.5. Unless you have been specifically permitted to do so in a separate agreement with PLURIBUS, you agree that you will not reproduce, duplicate, copy, sell, trade or resell the SOFTWARE (or any portion thereof) for any purpose.
   - 4.6. PLURIBUS may from time to time send updates over the Internet to you in order to update the SOFTWARE. You acknowledge and agree that PLURIBUS may make available new version of SOFTWARE without notice to you and that prior version of the SOFTWARE may be temporarily unavailable while PLURIBUS is updating the SOFTWARE. Pursuant to section 11 of this EULA, PLURIBUS is not liable for any disruptions in your use of the SOFTWARE, including while PLURIBUS is updating the SOFTWARE.

## End User License Agreement (cont'd)

5. **Privacy and your Usage Information**
   - 5.1. Should you enable sending product updates to the PluribusCloud, PLURIBUS may collect information ("USAGE INFORMATION") related to how you are using the SOFTWARE in accordance with your Pluribus Networks Purchase Agreement ("PURCHASE AGREEMENT"). The USAGE INFORMATION is collected and maintained in accordance with the PLURIBUS PRIVACY POLICY, as updated, located at www.pluribusnetworks.com/privacy
   - 5.2. You agree to allow PLURIBUS to collect USAGE INFORMATION and give PLURIBUS a perpetual, irrevocable, worldwide, royalty-free, and non-exclusive license to use (including, but not limited to, the rights to reproduce, adapt, and modify) the USAGE INFORMATION internally.
   - 5.3. You give PLURIBUS a perpetual, irrevocable, worldwide, royalty-free, and non-exclusive license to: (a) generate aggregated, non-personal information, where aggregated, non-personal information is USAGE INFORMATION that is collected into groups so that it no longer reflects or references an individually identifiable person or legal entity, and (b), to the extent necessary, reproduce, adapt, modify, translate, publish, publicly perform, publicly display and distribute any generated aggregated, non-personal information.
   - 5.4. You confirm and warrant to PLURIBUS that you have all the rights, power and authority necessary to grant PLURIBUS permission to collect USAGE INFORMATION and to use the USAGE INFORMATION in the manner specified in this section 5.
   - 5.5. PLURIBUS agrees not to disclose the USAGE INFORMATION to any third party except in accordance with Section 5.3 or the PLURIBUS PRIVACY POLICY. In the event of a conflict between this EULA and the PLURIBUS PRIVACY POLICY, the PLURIBUS PRIVACY POLICY shall govern.

6. **Proprietary Rights**
   - 6.1. You acknowledge and agree that PLURIBUS owns all legal right, title and interest in and to the SOFTWARE and DOCUMENTATION, including any intellectual property rights which subsist in the SOFTWARE and DOCUMENTATION (whether those rights happen to be registered or not, and wherever in the world those rights may exist). You further acknowledge that the SOFTWARE and DOCUMENTATION may contain information which is designated confidential by PLURIBUS and that you shall not disclose such information without PLURIBUS' prior written consent.
   - 6.2. You may use PLURIBUS' trademarks provided that such use is strictly limited and in accordance with the trademark guidelines located at www.pluribusnetworks.com/legal as adjusted from time to time.
   - 6.3. Unless you have agreed otherwise to in writing with PLURIBUS, nothing in this EULA gives you a right to use any of PLURIBUS' domain names and other distinctive brand features (separate and apart from PLURIBUS' trademarks).
   - 6.4. If you have been given an explicit right to use any of PLURIBUS' domain names and other distinctive brand features in a separate written agreement with PLURIBUS, then you agree that your use of such features shall be in compliance with that agreement, and any applicable provisions of this EULA.
   - 6.5. You agree that you shall not remove, obscure, or alter any proprietary rights or notices (including copyright and trademark notices) which may be affixed to or contained within the SOFTWARE and DOCUMENTATION.

## End User License Agreement (cont'd)

7. **License from PLURIBUS**

7.1. PLURIBUS gives you a personal, worldwide, non-assignable, non-transferable, and non-exclusive license to use the SOFTWARE and DOCUMENTATION provided to you by PLURIBUS. Conditioned upon compliance with the terms and conditions of the Agreement, PLURIBUS grants to you a nonexclusive and nontransferable license to use for your internal business purposes the Software and the DOCUMENTATION for which you have paid the required license fees to an Approved Source. “DOCUMENTATION” means written information (whether contained in user or technical manuals, training materials, specifications or otherwise) pertaining to the SOFTWARE and made available by an Approved Source with the SOFTWARE in any manner (including on CD-Rom, or on-line). In order to use the SOFTWARE, you may be required to input a registration number or product authorization key and register your copy of the SOFTWARE online at PLURIBUS ' website to obtain the necessary license key or license file.

7.2. Your license to use the SOFTWARE shall be limited to, and you shall not use the SOFTWARE in excess of, a single hardware chassis or card or such other limitations as are set forth in the applicable Supplemental License Agreement or in the applicable Purchase Agreement or purchase order which has been accepted by an Approved Source and for which you have paid to an Approved Source the required license fee (the “Purchase Order”).

7.3. Unless otherwise expressly provided in the DOCUMENTATION or any applicable Supplemental License Agreement, you shall use the SOFTWARE solely as embedded in, for execution on, or (where the applicable DOCUMENTATION permits installation on non-Pluribus equipment) for communication with PLURIBUS equipment owned or leased by you from an Approved Source and used for your internal business purposes. No other licenses are granted by implication, estoppel or otherwise.

7.4. You may not (and you may not permit anyone else to) copy, modify, create a derivative work of any DOCUMENTATION, unless this is expressly permitted or required by law, or unless you have been specifically told that you may do so by PLURIBUS, in writing.

7.5. This is a license, not a transfer of title, to the Software and DOCUMENTATION, and PLURIBUS retains ownership of all copies of the SOFTWARE and DOCUMENTATION. Customer acknowledges that the SOFTWARE and DOCUMENTATION contain trade secrets of PLURIBUS or its suppliers or licensors, including but not limited to the specific internal design and structure of individual programs and associated interface information. Except as otherwise expressly provided under this EULA, you shall have no right, and you specifically agree not to:

i) transfer, assign or sublicense its license rights to any other person or entity (other than in compliance with any PLURIBUS relicensing/transfer policy then in force), or use the SOFTWARE on PLURIBUS equipment not purchased by you from an Approved Source or on secondhand PLURIBUS equipment, and you acknowledge that any attempted transfer, assignment, sublicense or use shall be void;

ii) make error corrections to or otherwise modify or adapt the SOFTWARE or create derivative works based upon the SOFTWARE, or permit third parties to do the same;

iii) reverse engineer or decompile, decrypt, disassemble or otherwise reduce the SOFTWARE to human-readable form, except to the extent otherwise expressly permitted under applicable law notwithstanding this restriction or except to the extent that PLURIBUS is legally required to permit such specific activity pursuant to any applicable open source license;

iv) publish any results of benchmark tests run on the SOFTWARE;

v) use or permit the SOFTWARE to be used on a service bureau or time sharing basis as relates to direct shared use of such SOFTWARE (and not to applications or services running upon or utilizing such SOFTWARE), without the express written authorization of PLURIBUS;or

vi) disclose, provide, or otherwise make available trade secrets contained within the SOFTWARE and DOCUMENTATION in any form to any third party without the prior written consent of PLURIBUS. You shall implement reasonable security measures to protect such tradesecrets

7.6. To the extent required by applicable law, and at your written request, PLURIBUS shall provide you with the interface information needed to achieve interoperability between the SOFTWARE and another independently created program, on payment of PLURIBUS' applicable fee, if any. You shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which PLURIBUS makes such information available.

8. **Ending your relationship with PLURIBUS**

8.1. This EULA will continue to apply and will not come to an end until terminated by either you or PLURIBUS as set out below.

8.2. PLURIBUS shall terminate its legal agreement with you if: (a) you have breached any provision of this EULA (or have acted in manner which clearly shows that you do not intend to, or are unable to comply with the provisions of this EULA), automatically and without requiring any further action by PLURIBUS; or (b) PLURIBUS is required to do so by law (for example, where the provision of the SOFTWARE or DOCUMENTATION to you is, or becomes,unlawful).

8.3. When this EULA comes to an end, all of the legal rights, obligations and liabilities that you and PLURIBUS have benefited from, been subject to (or which have accrued over time whilst this EULA has been in force) or which are expressed to continue perpetually, shall be unaffected by this cessation. Upon termination, you shall destroy all copies of SOFTWARE and DOCUMENTATION in your possession or control.

## End User License Agreement (cont'd)

9. **WARRANTY**

9.1. Except as specifically warranted in the PURCHASE AGREEMENT, PLURIBUS disclaims all warranties as set forth in Section 10 of this agreement. Subject to the limitations and conditions set forth herein, PLURIBUS warrants that commencing from the date of shipment to you (but in case of resale by an Approved Source other than PLURIBUS, commencing not more than ninety (90) days after original shipment by PLURIBUS), and continuing for a period of the longer of (a) ninety (90) days or (b) the warranty period (if any) expressly set forth as applicable specifically to SOFTWARE in the warranty card accompanying the product of which the SOFTWARE is a part (the "Product") (if any): (a) the media on which the SOFTWARE is furnished will be free of defects in materials and workmanship under normal use; and (b) the SOFTWARE substantially conforms to the DOCUMENTATION.

9.2. The date of shipment of a Product by PLURIBUS is set forth on the packaging material in which the Product is shipped. Except for the foregoing, the SOFTWARE is provided "AS IS". This limited warranty extends only to the SOFTWARE purchased from an Approved Source by a user who is the first registered end user. Your sole and exclusive remedy and the entire liability of PLURIBUS and its suppliers under this limited warranty will be (i) replacement of defective media and/or (ii) at PLURIBUS ' option, repair, replacement, or refund of the purchase price of the SOFTWARE, in both cases subject to the condition that any error or defect constituting a breach of this limited warranty is reported to the Approved Source supplying the Software to you, within the warranty period. PLURIBUS or the Approved Source supplying the SOFTWARE to you may, at its option, require return of the SOFTWARE and/or DOCUMENTATION as a condition to the remedy.

9.3. In no event does PLURIBUS warrant that the SOFTWARE is error free or that you will be able to operate the SOFTWARE without problems or interruptions. In addition, due to the continual development of new techniques for intruding upon and attacking networks, PLURIBUS does not warrant that the SOFTWARE or any equipment, system or network on which the SOFTWARE is used will be free of vulnerability to intrusion or attack.

9.4. This warranty does NOT apply if the SOFTWARE, Product or any other equipment upon which the SOFTWARE is authorized to be used (a) has been altered, except by PLURIBUS or its authorized representative, (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by PLURIBUS, (c) has been subjected to abnormal physical or electrical stress, abnormal environmental conditions, misuse, negligence, or accident; or (d) is licensed for beta, evaluation, testing or demonstration purposes. The SOFTWARE warranty also does not apply to (e) any temporary SOFTWARE modules; (f) any SOFTWARE not posted on the software update or support site on Pluribus' web page: (g) any SOFTWARE that PLURIBUS expressly provides on an "AS IS" basis on PLURIBUS software update or support site on Pluribus' web page: (h) any SOFTWARE for which PLURIBUS or an Approved Source does not receive a license fee; and (i) SOFTWARE supplied by any third party which is not an Approved Source.

## End User License Agreement (cont'd)

10. **EXCLUSION OF WARRANTIES**

10.1. NOTHING IN EULA, INCLUDING SECTIONS 10 AND 11, SHALL EXCLUDE OR LIMIT PLURIBUS' WARRANTY OR LIABILITY FOR LOSSES WHICH MAY NOT BE LAWFULLY EXCLUDED OR LIMITED BY APPLICABLE LAW. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF CERTAIN WARRANTIES OR CONDITIONS OR THE LIMITATION OR EXCLUSION OF LIABILITY FOR LOSS OR DAMAGE CAUSED BY NEGLIGENCE, BREACH OF CONTRACT OR BREACH OF IMPLIED TERMS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES. ACCORDINGLY, ONLY THE LIMITATIONS WHICH ARE LAWFUL IN YOUR JURISDICTION WILL APPLY TO YOU AND OUR LIABILITY IS LIMITED IN ALL CASES TO THE MAXIMUM EXTENT PERMITTED BY LAW.

10.2. YOU EXPRESSLY UNDERSTAND AND AGREE THAT YOUR USE OF THE SOFTWARE AND DOCUMENTATION IS AT YOUR SOLE RISK AND THAT THE SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS" AND "AS AVAILABLE."

10.3. IN PARTICULAR, PLURIBUS DOES NOT REPRESENT OR WARRANT TO YOU THAT:

a) YOUR USE OF THE SOFTWARE AND DOCUMENTATION WILL MEET YOUR REQUIREMENTS,

b) YOUR USE OF THE SOFTWARE OR ACCESS TO THE DOCUMENTATION WILL BE UNINTERRUPTED, TIMELY, SECURE OR FREE FROM ERROR,

c) ANY INFORMATION OBTAINED BY YOU AS A RESULT OF YOUR USE OF THE SOFTWARE AND DOCUMENTATION WILL BE ACCURATE OR RELIABLE, AND

d) THAT DEFECTS IN THE OPERATION OR FUNCTIONALITY OF ANY SOFTWARE PROVIDED TO YOU WILL BE CORRECTED.

10.4. ANY MATERIAL DOWNLOADED OR OTHERWISE OBTAINED THROUGH THE USE OF THE SOFTWARE IS DONE AT YOUR OWN DISCRETION AND RISK AND YOU, AND NOT PLURIBUS, WILL BE RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR OTHER DEVICES OR LOSS OF DATA THAT RESULTS FROM THE DOWNLOAD OF ANY SUCH MATERIAL.

10.5. NO ADVICE OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED BY YOU FROM PLURIBUS SHALL CREATE ANY WARRANTY NOT EXPRESSLY STATED IN THIS EULA.

10.6. PLURIBUS FURTHER EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

## End User License Agreement (cont'd)

11. LIMITATION OF LIABILITY

11.1. SUBJECT TO THE OVERALL PROVISION IN PARAGRAPH 10.1 ABOVE, YOU EXPRESSLY UNDERSTAND AND AGREE THAT PLURIBUS, SHALL NOT BE LIABLE TO YOU FOR:

a) ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL CONSEQUENTIAL OR EXEMPLARY DAMAGES WHICH MAY BE INCURRED BY YOU, HOWEVER CAUSED AND UNDER ANY THEORY OF LIABILITY. THIS SHALL INCLUDE, BUT NOT BE LIMITED TO, ANY LOSS OF PROFIT (WHETHER INCURRED DIRECTLY OR INDIRECTLY), ANY LOSS OF GOODWILL OR BUSINESS REPUTATION, ANY LOSS OF DATA SUFFERED, COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR OTHER INTANGIBLE LOSS;

b) ANY LOSS OR DAMAGE WHICH MAY BE INCURRED BY YOU, INCLUDING BUT NOT LIMITED TO LOSS OR DAMAGE AS A RESULT OF:

1. ANY CHANGES WHICH PLURIBUS MAY MAKE TO THE SOFTWARE OR DOCUMENTATION, OR FOR ANY PERMANENT OR TEMPORARY CESSATION IN THE PROVISION OF THE SOFTWARE (OR ANY FEATURES WITHIN THE SOFTWARE); OR
2. THE DELETION OF, CORRUPTION OF, OR FAILURE TO STORE, ANY CONTENT AND OTHER COMMUNICATIONS DATA MAINTAINED OR TRANSMITTED BY OR THROUGH YOUR USE OF THE SOFTWARE.

11.2. THE LIMITATIONS ON PLURIBUS'S LIABILITY TO YOU IN PARAGRAPH 11.1 ABOVE SHALL APPLY WHETHER OR NOT PLURIBUS HAS BEEN ADVISED OF OR SHOULD HAVE BEEN AWARE OF THE POSSIBILITY OF ANY SUCH LOSSES ARISING.

11.3. IN NO EVENT SHALL THE AGGREGATE LIABILITY OF PLURIBUS EXCEED $5,000. OR EXCEED THE PRICE PAID BY CUSTOMER TO ANY APPROVED SOURCE FOR THE SOFTWARE THAT GAVE RISE TO THE CLAIM OR IF THE SOFTWARE IS PART OF ANOTHER PRODUCT, THE PRICE PAID FOR SUCH OTHER PRODUCT. THIS LIMITATION OF LIABILITY FOR SOFTWARE IS CUMULATIVE AND NOT PER INCIDENT (I.E. THE EXISTENCE OF TWO OR MORE CLAIMS WILL NOT ENLARGE THIS LIMIT).

11.4. You acknowledge and agree that PLURIBUS has set its prices and entered into this EULA and any Purchase Agreement or Purchase Order with you in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the parties.

## End User License Agreement (cont'd)

12. The **Software** may contain or be delivered with one or more components, which may include third-party components, identified by Pluribus in the Documentation, readme.txt file, third-party click-accept or elsewhere (e.g. on www.pluribusnetworks.com) (the "Identified Component(s)") as being subject to different license agreement terms, disclaimers of warranties, limited warranties or other terms and conditions (collectively, "Additional Terms") than those set forth herein. You agree to the applicable Additional Terms for any such Identified Component(s).
    12.1. Notwithstanding other statements in this EULA, third party software including free, Copyleft and open source software components (collectively referred to as "Third Party Software") are distributed in compliance with the particular licensing terms and conditions attributable to the Third Party Software. PLURIBUS provides the Third Party Software to You "AS IS" without any warranties or indemnities of any kind.
    12.2. Copyright notices and licensing terms and conditions applicable to the Third Party Software will be available for review on the Pluribus' web site, and are included on the media on which you received the SOFTWARE within a "ATTRIBUTIONS" file (e.g., attributions.pdf or attributions.txt) included within the downloaded files, and/or reproduced within the materials or DOCUMENTATION accompanying the SOFTWARE.

13. **Audit**
    13.1. PLURIBUS reserves the right to take steps PLURIBUS believes are reasonably necessary or appropriate to enforce and/or verify compliance with any part of this EULA. You agree that PLURIBUS has the right, without liability to you, to disclose any USAGE INFORMATION to law enforcement authorities, government officials, and/or a third party, as PLURIBUS believes is reasonably necessary or appropriate to enforce and/or verify compliance with any part of this EULA (including but not limited to PLURIBUS' right to cooperate with any legal process relating to your use of the SOFTWARE, and/or a third-party claim that your use of the SOFTWARE is unlawful and/or infringes such third party rights).

## End User License Agreement (cont'd)

14. **General legal terms**

14.1. This EULA constitutes the whole legal agreement between you and PLURIBUS and governs your use of the SOFTWARE and DOCUMENTATION and completely replaces any prior agreements between you and PLURIBUS in relation to the SOFTWARE and DOCUMENTATION (but excluding any SOFTWARE and DOCUMENTATION which PLURIBUS may provide to you under a separate written agreement).

14.2. You agree that PLURIBUS may provide you with notices, including those regarding changes to this EULA, by email, regular mail, or via the user interface implemented by the SOFTWARE.

14.3. You agree that if PLURIBUS does not exercise or enforce any legal right or remedy which is contained in this EULA (or which PLURIBUS has the benefit of under any applicable law), this will not be taken to be a formal waiver of PLURIBUS' rights and that those rights or remedies will still be available to PLURIBUS.

14.4. If any court of law or arbitration panel, having the jurisdiction to decide on this matter, rules that any provision of this EULA is invalid, then that provision will be removed from this EULA without affecting the rest of this EULA. The remaining provisions of this EULA will continue to be valid and enforceable.

14.5. You and PLURIBUS agree that this EULA, and your relationship with PLURIBUS under this EULA, shall be governed by the laws of the State of California without regard to its conflict of laws provisions.

14.6. The SOFTWARE and DOCUMENTATION is deemed to include "commercial computer software" and "commercial computer software documentation," respectively, pursuant to DFAR Section 227.7202 and FAR Section 12.212, as applicable. Any use, modification, reproduction, release, performance, display or disclosure of the SOFTWARE and DOCUMENTATION by the United States Government shall be governed solely by this EULA.

14.7. In the event that the Uniform Computer Information Transaction Act, any version thereof or a substantially similar law (collectively "UCITA") is enacted and/or interpreted as to be applicable to the performance of PLURIBUS under this Agreement, the statute shall not govern any aspect of this Purchase Agreement, any license granted hereunder, nor any of the rights and obligations of the parties pursuant to this EULA.

14.8. You agree that the SOFTWARE and DOCUMENTATION will not be shipped, transferred, or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other exports laws, restrictions, or regulations. All rights to use the SOFTWARE and DOCUMENTATION are granted on condition that such rights are forfeited if you fail to comply with this EULA.

## End User License Agreement (cont'd)

14.9. ANY CLAIM, DISPUTE, OR CONTROVERSY (WHETHER IN CONTRACT, TORT, OR OTHERWISE, WHETHER PREEXISTING, PRESENT OR FUTURE, AND INCLUDING STATUTORY, CONSUMER PROTECTION, COMMON LAW, INTENTIONAL TORT AND EQUITABLE CLAIMS) BETWEEN YOU AND PLURIBUS, its agents, employees, principals, successors, assigns, affiliates (collectively for purposes of this paragraph, “PLURIBUS”) arising from or relating to this EULA, its interpretation, or the breach, termination or validity thereof, the relationships which result from this EULA (including, to the full extent permitted by applicable law, relationships with third parties who are not signatories to this Agreement) SHALL BE EXCLUSIVELY AND FINALLY SETTLED BY ARBITRATION. THE ARBITRATION SHALL BE HELD IN SANTA CLARA, CALIFORNIA AND CONDUCTED IN ACCORDANCE WITH THE COMMERCIAL ARBITRATION RULES OF THE AMERICAN ARBITRATION ASSOCIATION. THE ARBITRATION SHALL BE CONDUCTED BEFORE THREE ARBITRATORS, ONE SELECTED BY EACH OF THE PARTIES, AND THE THIRD SELECTED BY THE FIRST TWO ARBITRATORS. JUDGMENT UPON THE AWARD RENDERED MAY BE ENTERED IN ANY COURT HAVING JURISDICTION, OR APPLICATION MAY BE MADE TO SUCH COURT FOR JUDICIAL ACCEPTANCE OF THE AWARD AND IN ORDER OF ENFORCEMENT AS THE CASE MAY BE.

## Summary

The Pluribus 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. Pluribus 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 Pluribus 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, Pluribus 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.

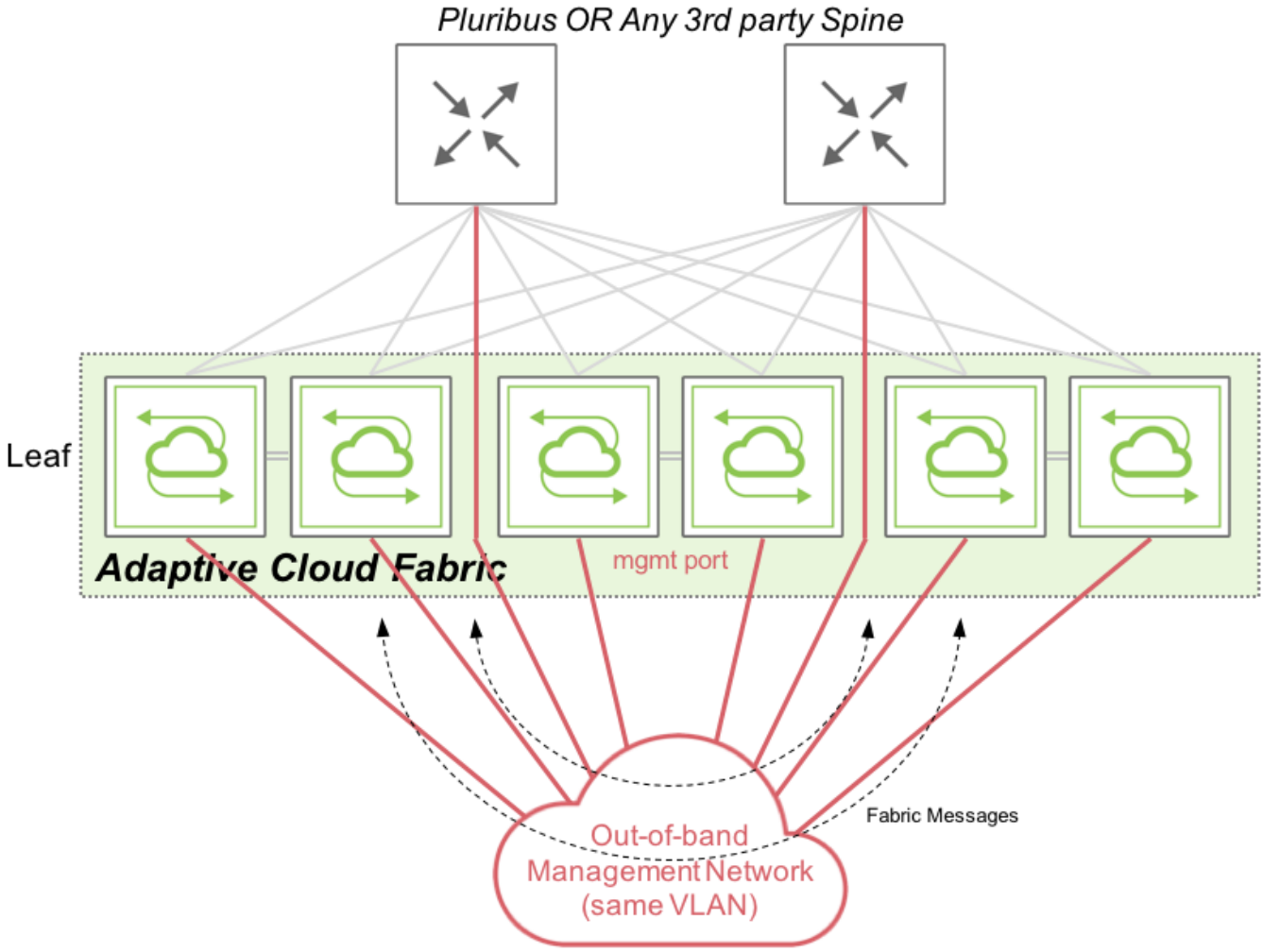
## Pluribus Networks Unified Cloud Fabric Overview

Pluribus 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 Pluribus 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 Pluribus 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).

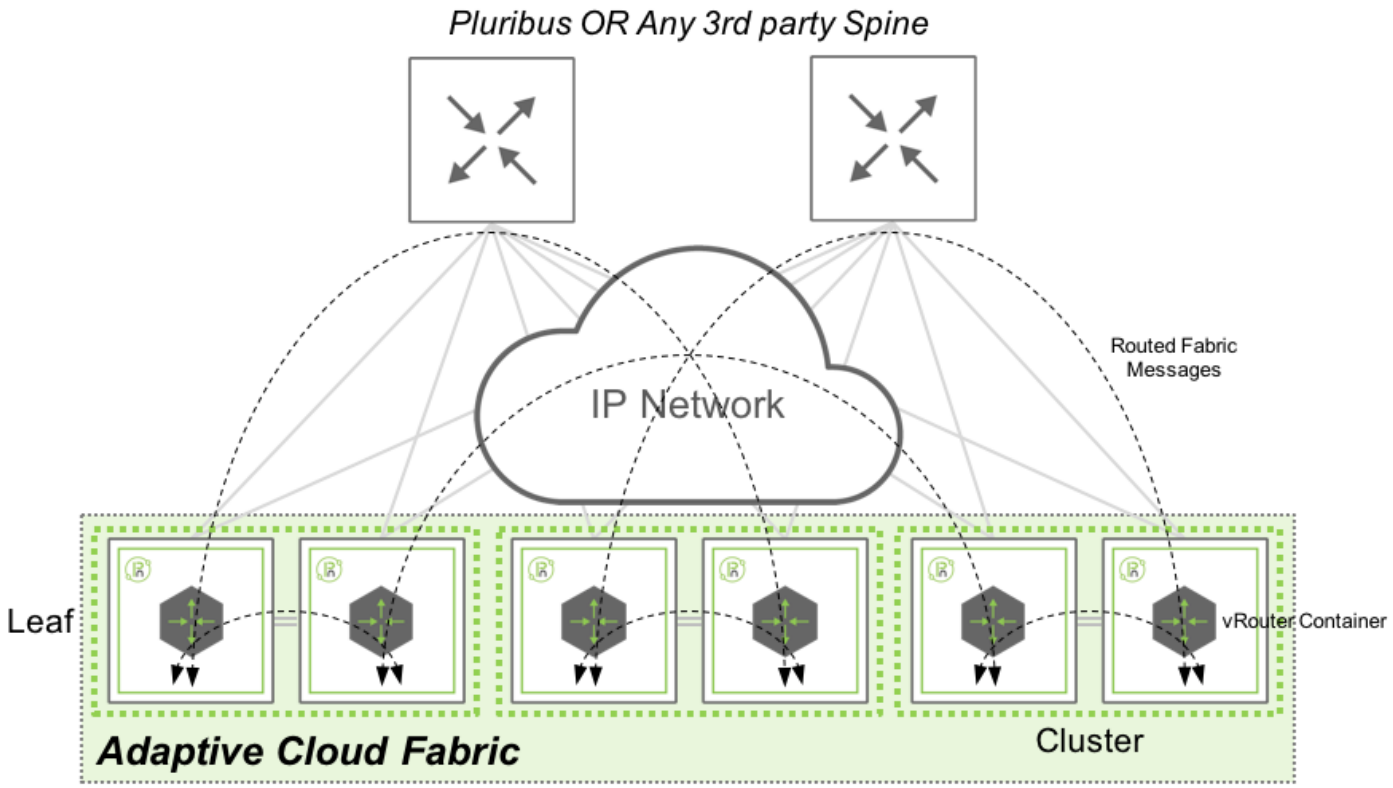
- The underlay network is a Layer 3 network (using IGP and ECMP for redundancy):

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 Pluribus 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, Pluribus 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 Pluribus 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 Pluribus 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 Pluribus Networks Netvisor on top to bring a data center class network feature set.

## Another Fabric Node

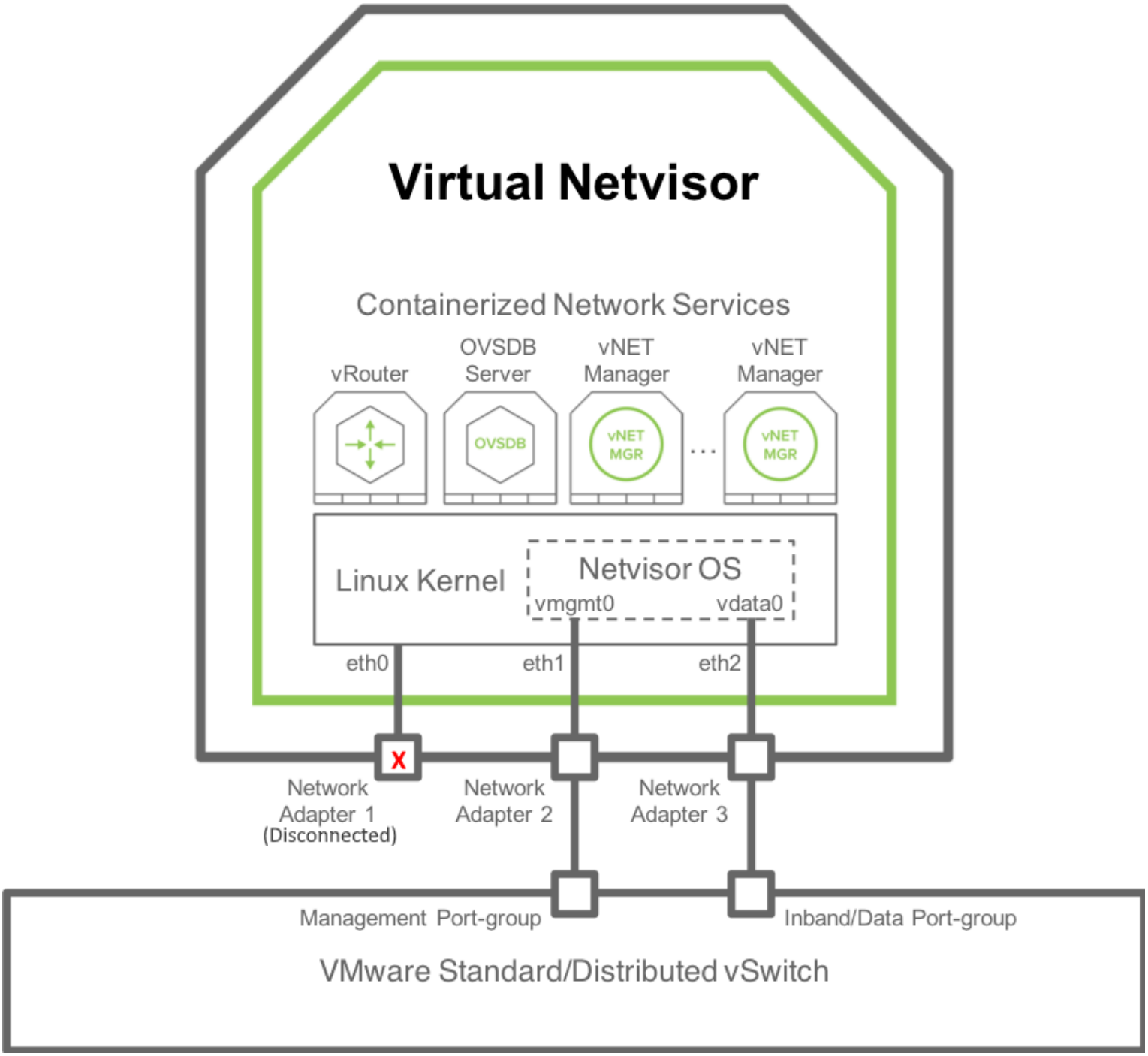
Virtual Netvisor is running Netvisor like a regular node, consequently, it is another fabric node participating in the distributed management and control plane. Despite a limited set of functionalities available in Virtual Netvisor (no L2 functions for instance), we use this node as part of an existing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

## 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.

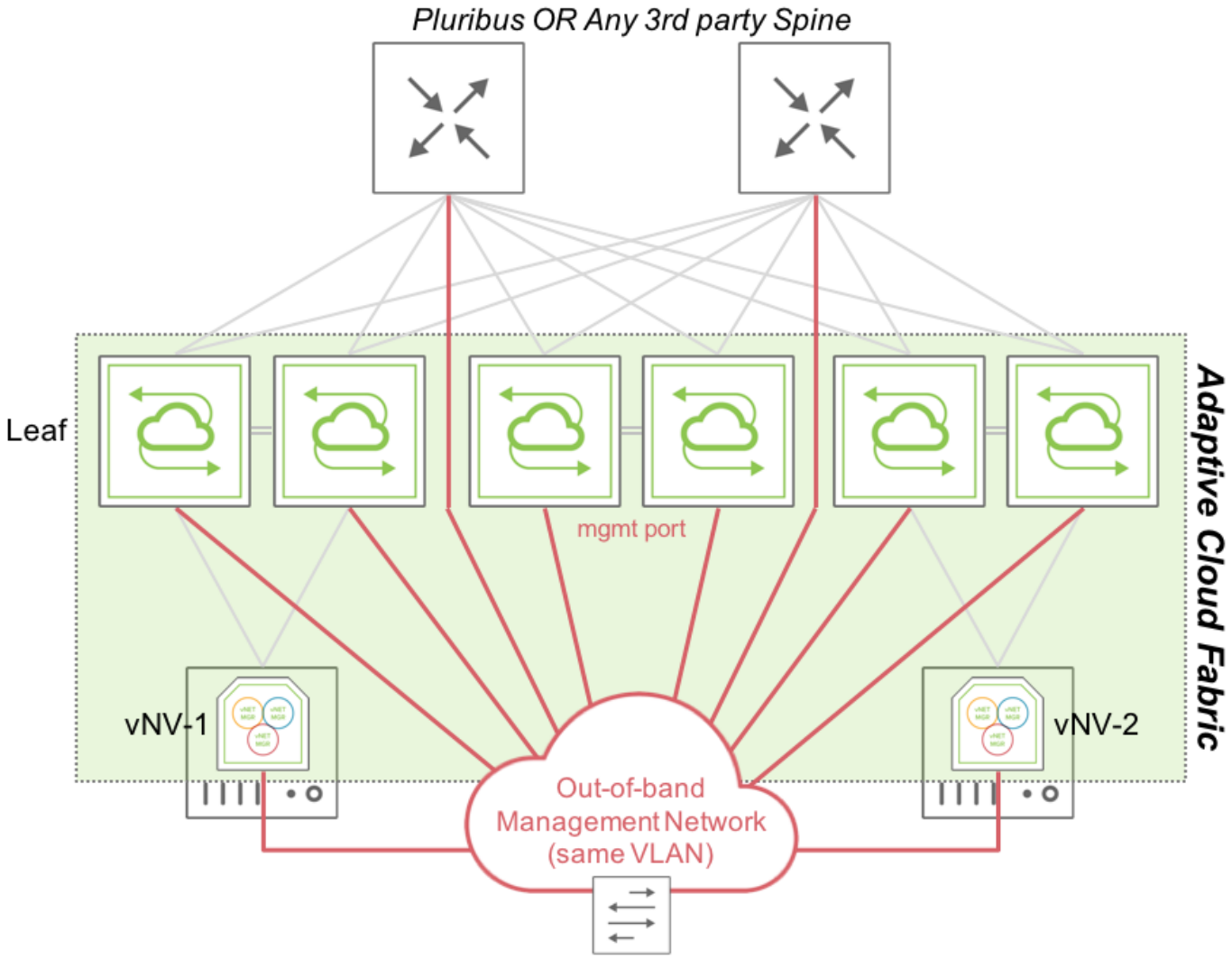
As vNV is a Virtual Machine (VM), the way you can connect it to the network is slightly different from a physical switch.

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:




*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 Pluribus 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

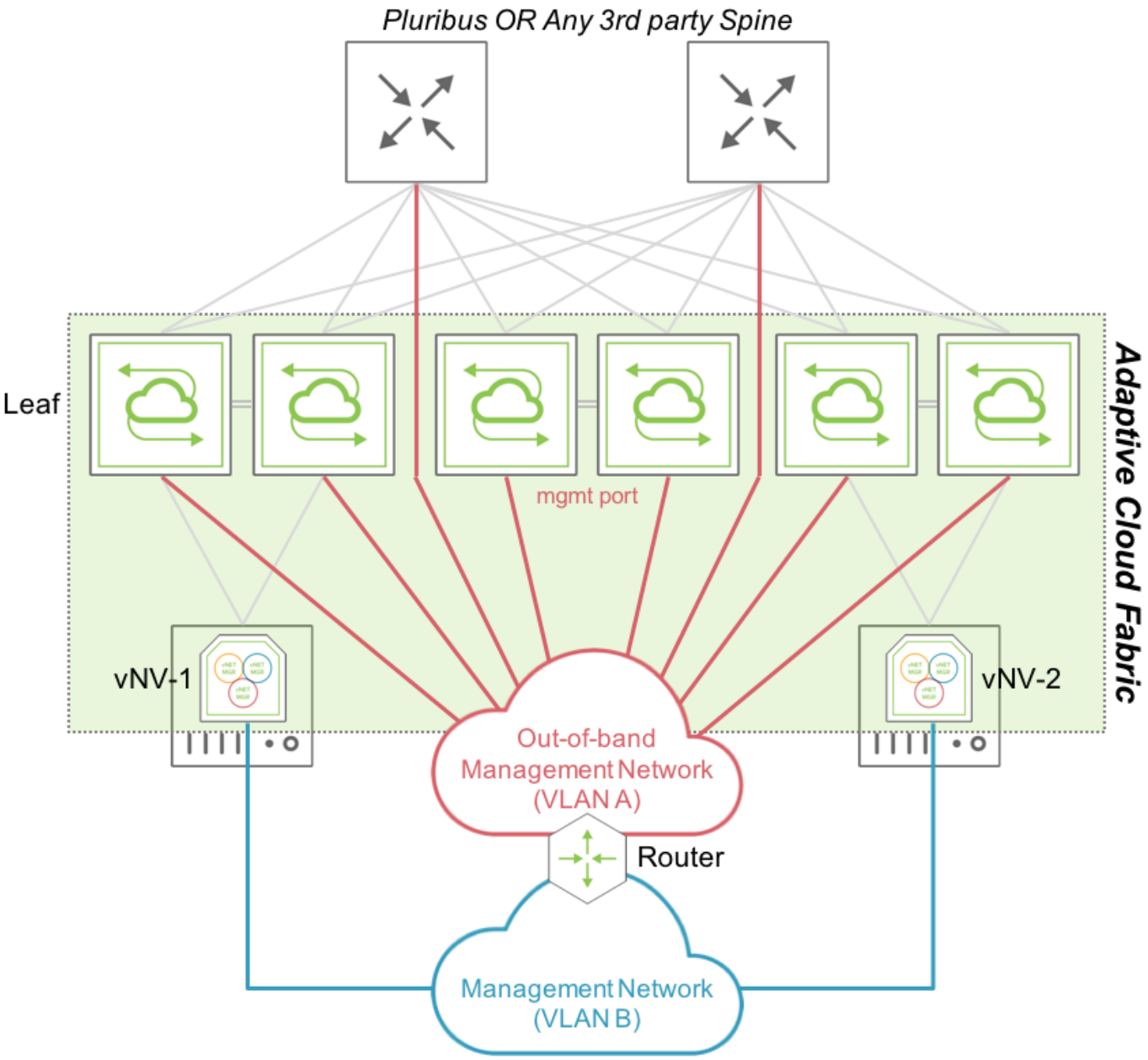
In certain cases, it might be complicated to connect ESXi hypervisors in the out-of-band network directly (in the same VLAN or bridge domain):

- 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.

## Overview (cont'd)

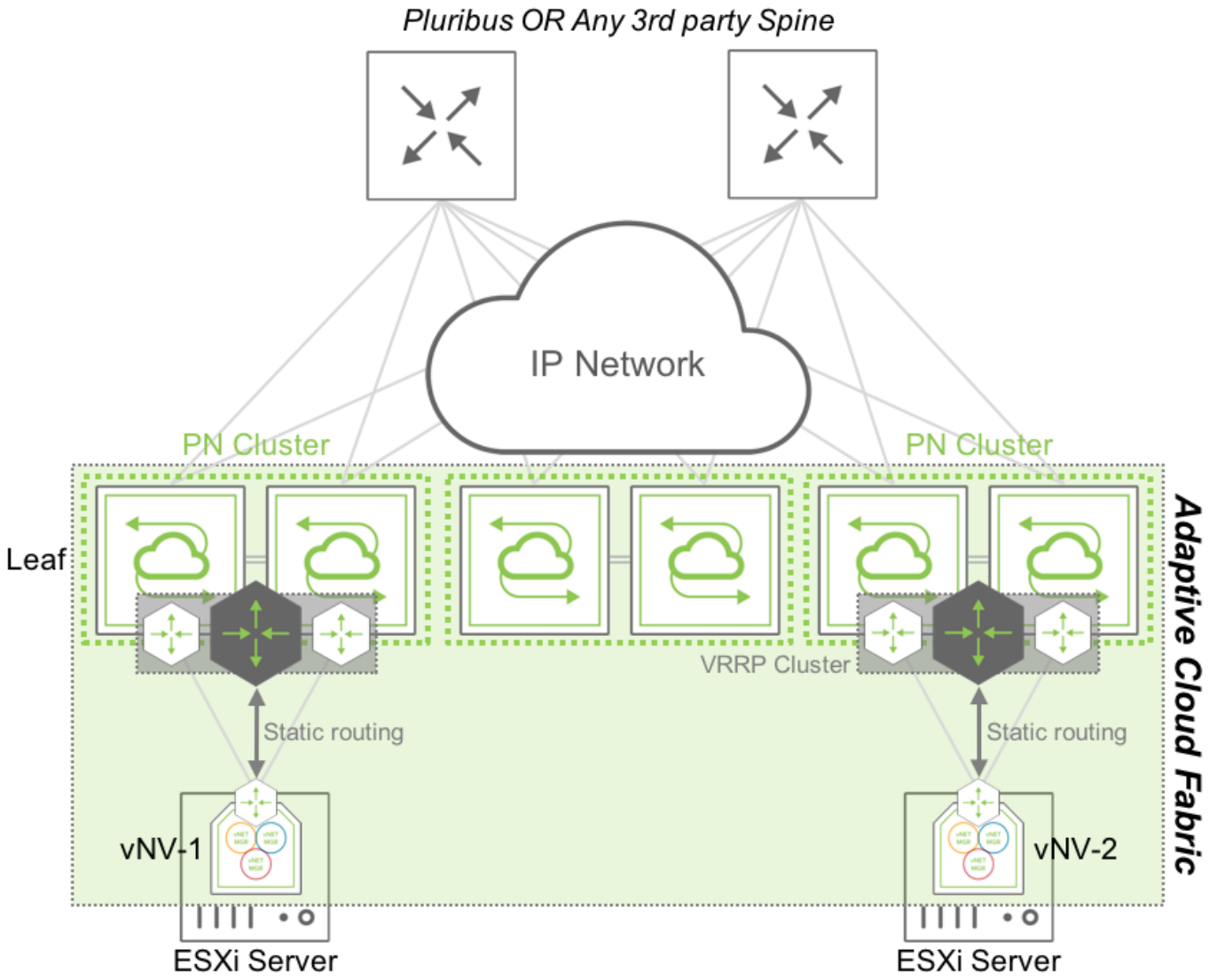



*Fabric Nodes in a Routed Management Network*

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 Pluribus Networks switches or to the same cluster.

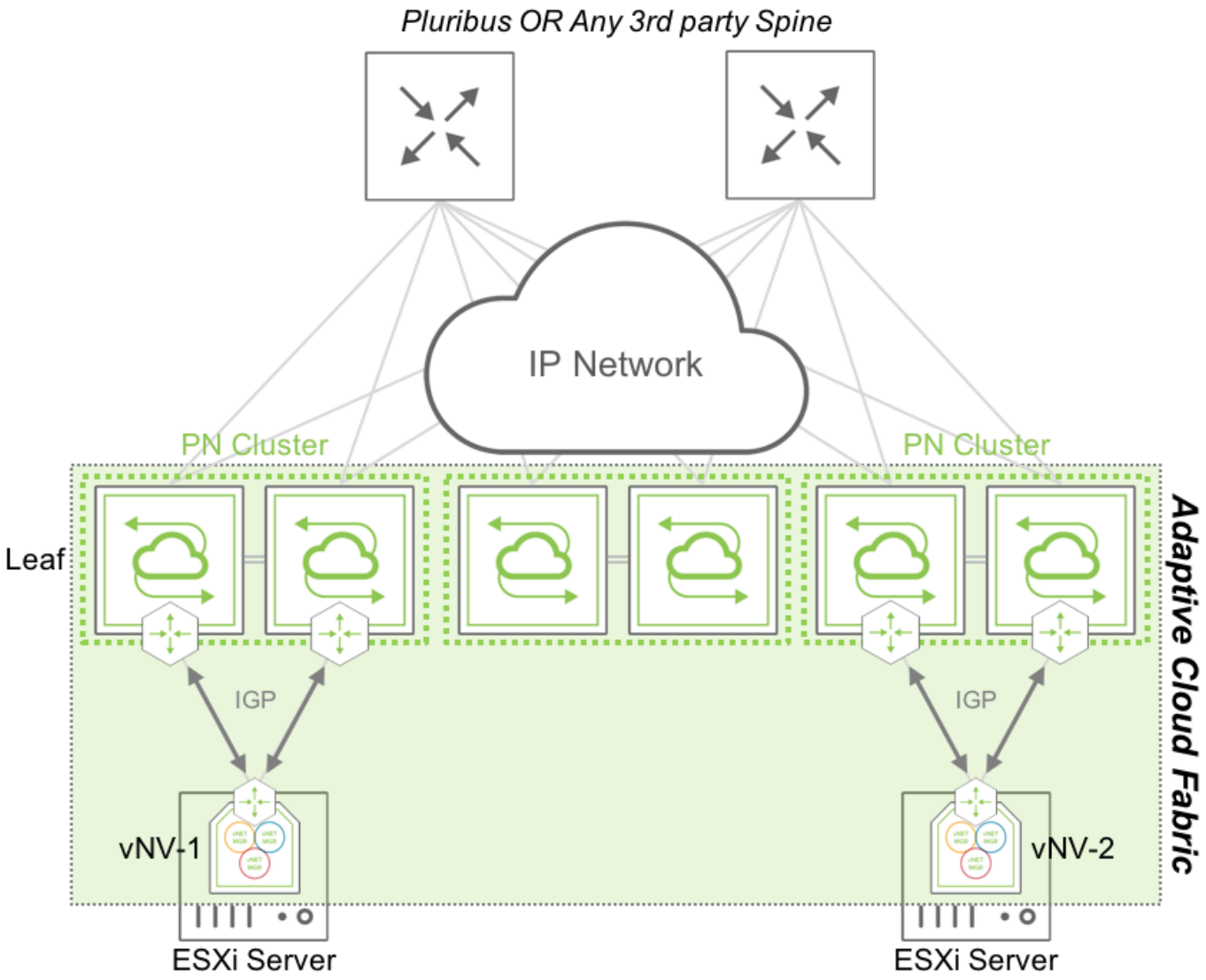
vNV-1 and vNV-2 use their in-band interface to communicate with the rest of the fabric and use the top of the rack (ToR) switches to communicate with other nodes. The two ToR switches are configured in a VRRP cluster and redistribute this local subnet in the IGP to advertise it to other fabric nodes. 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, Pluribus 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, Pluribus 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 Pluribus 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 Pluribus 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.

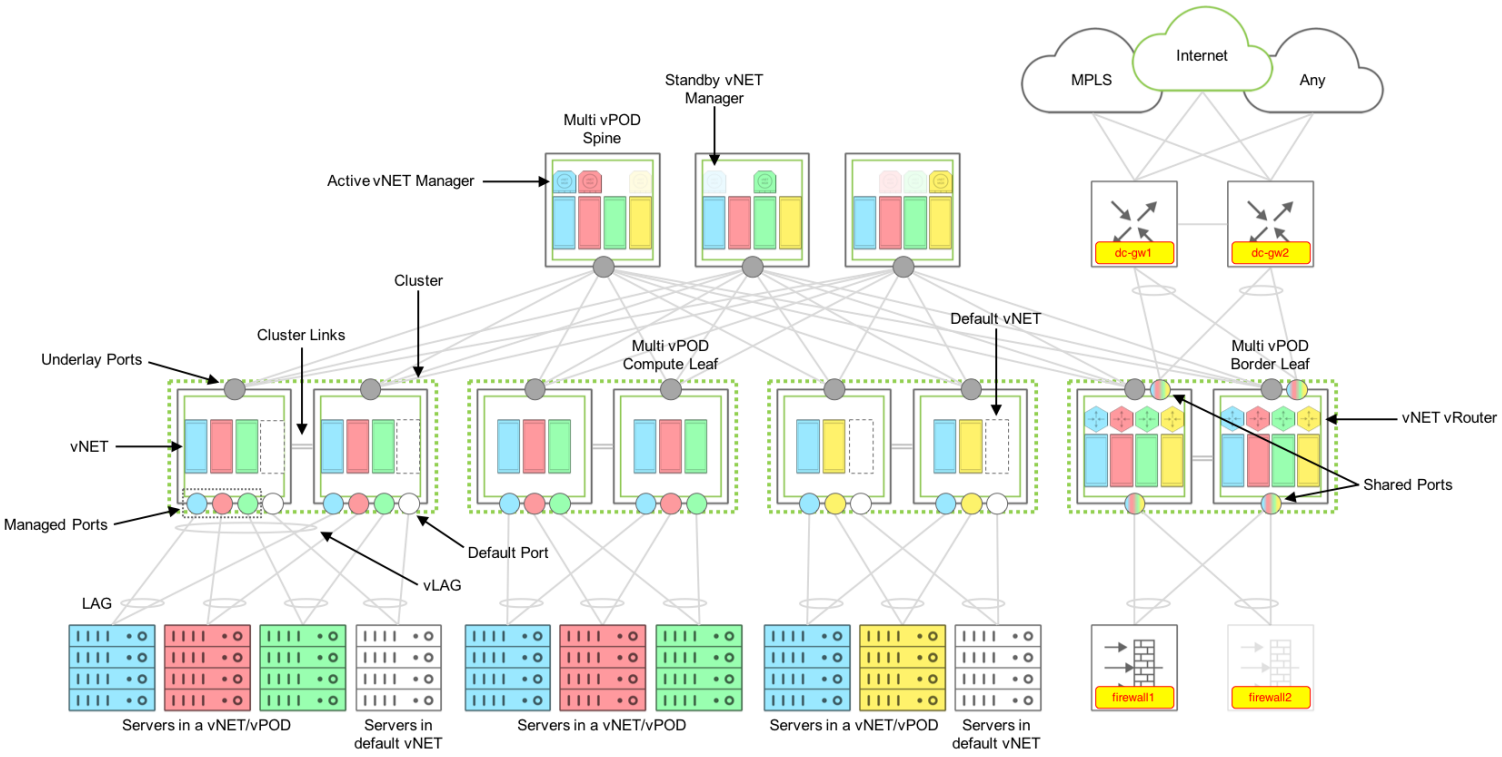
Virtual Networks (vNETs) is Pluribus Networks' advanced solution to dealing with multi-tenancy requirements in a secure and versatile manner that goes beyond basic VLAN and VRF standards. In basic terms vNETs are separate resource management spaces and operate in the data plane as well as in the control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

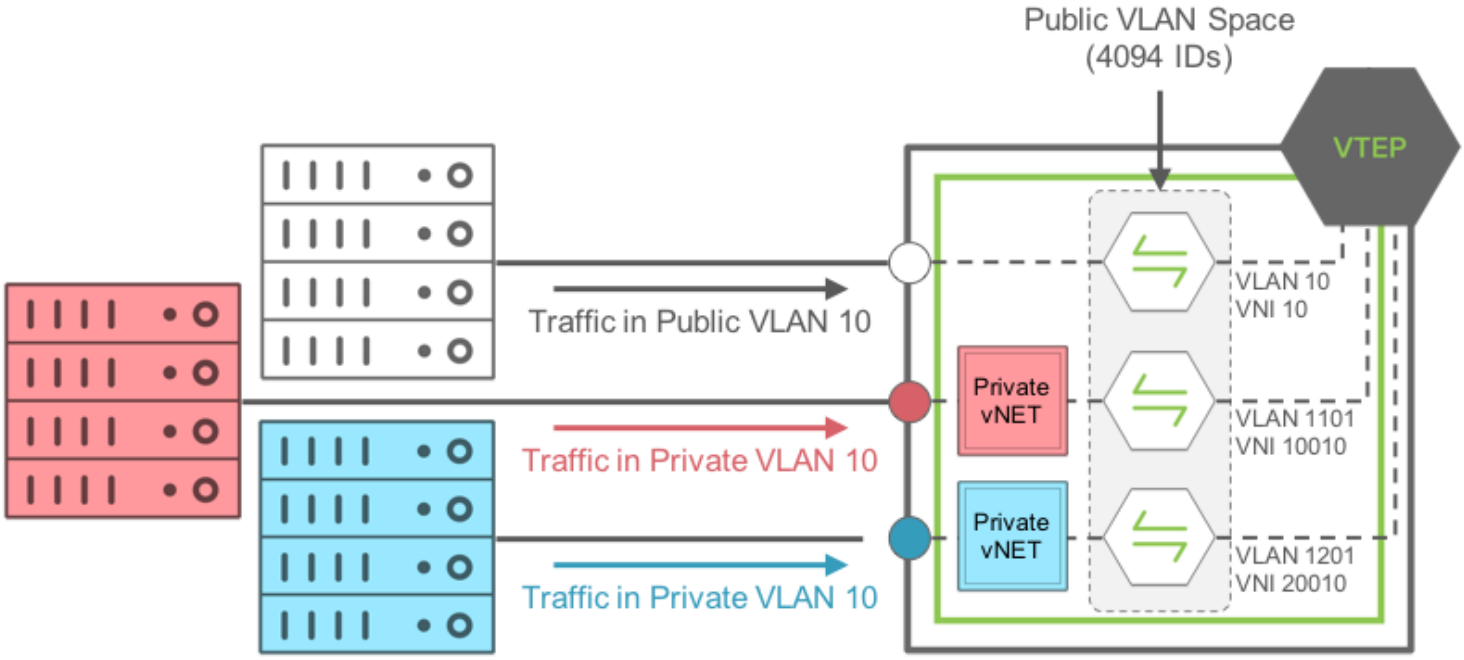
## Public VLAN

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 Pluribus 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 Pluribus 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, Pluribus 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.

## Overview (cont'd)

### 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.

## Overview (cont'd)

### 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.

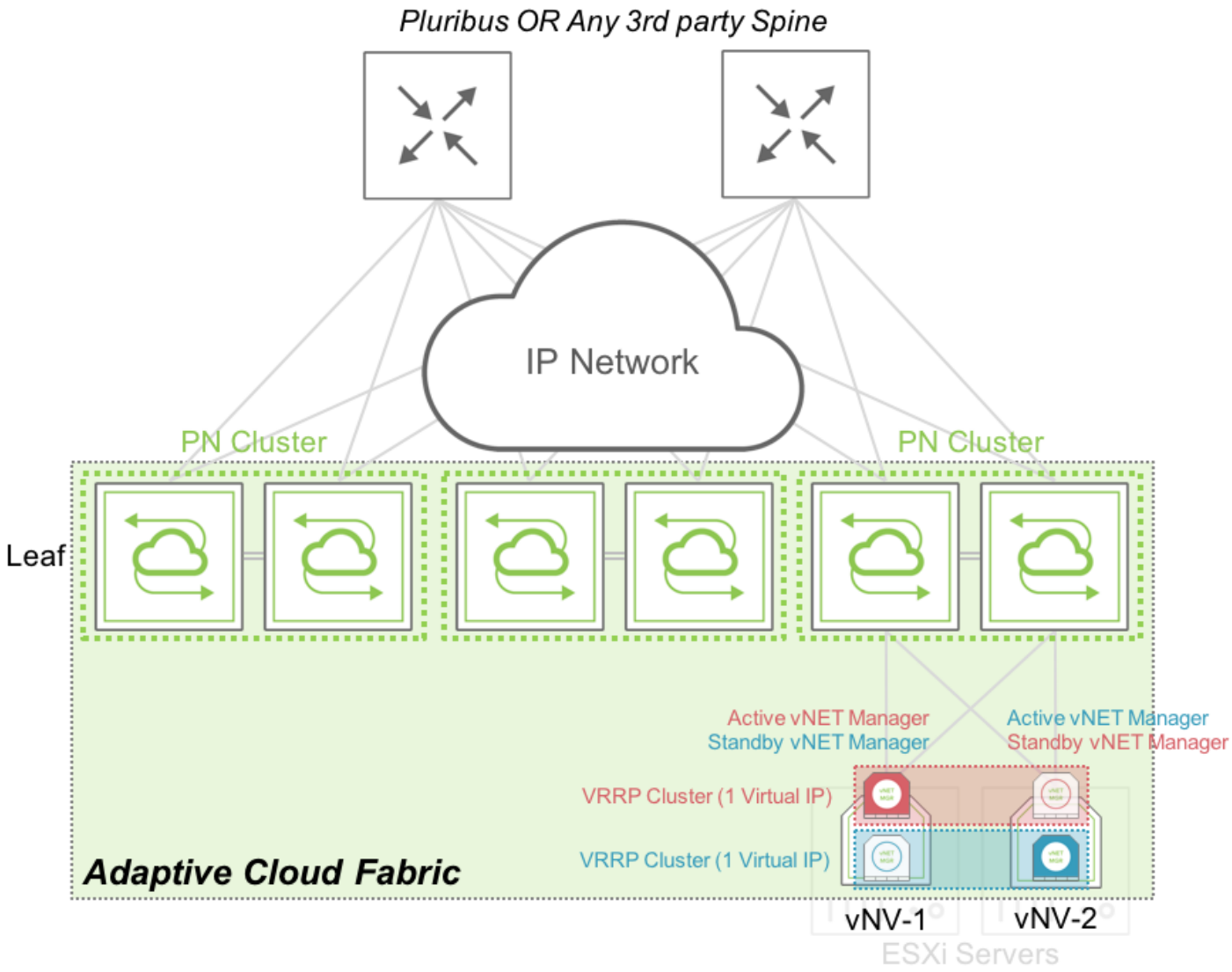
In this configuration, you can easily create 2 vNVs sharing the same broadcast domain and run VRRP protocol on their vNET Manager Interface.

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:

## Overview (cont'd)

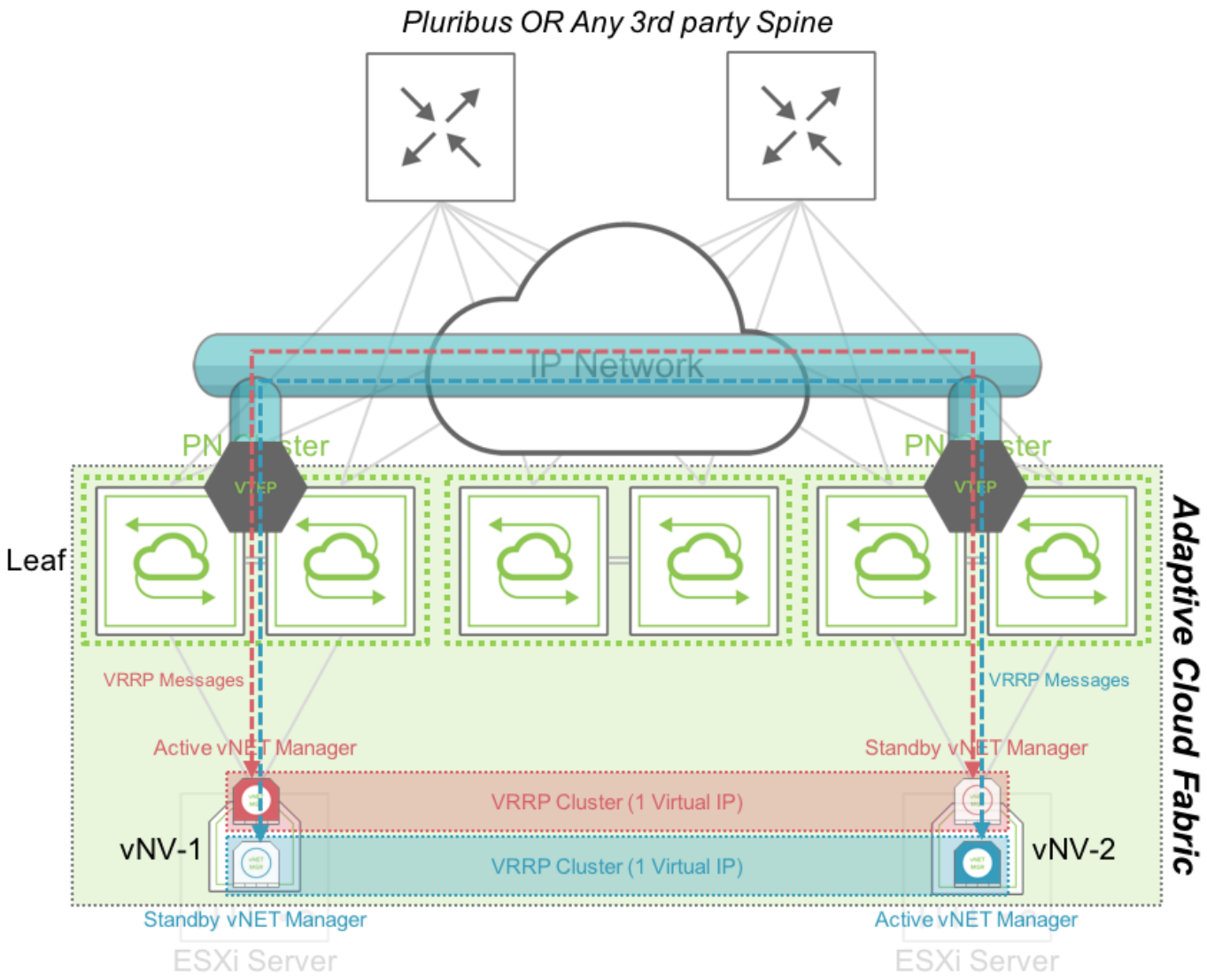



*vNET Manager Redundancy – Active/Standby in a Rack*

## 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. Pluribus 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).

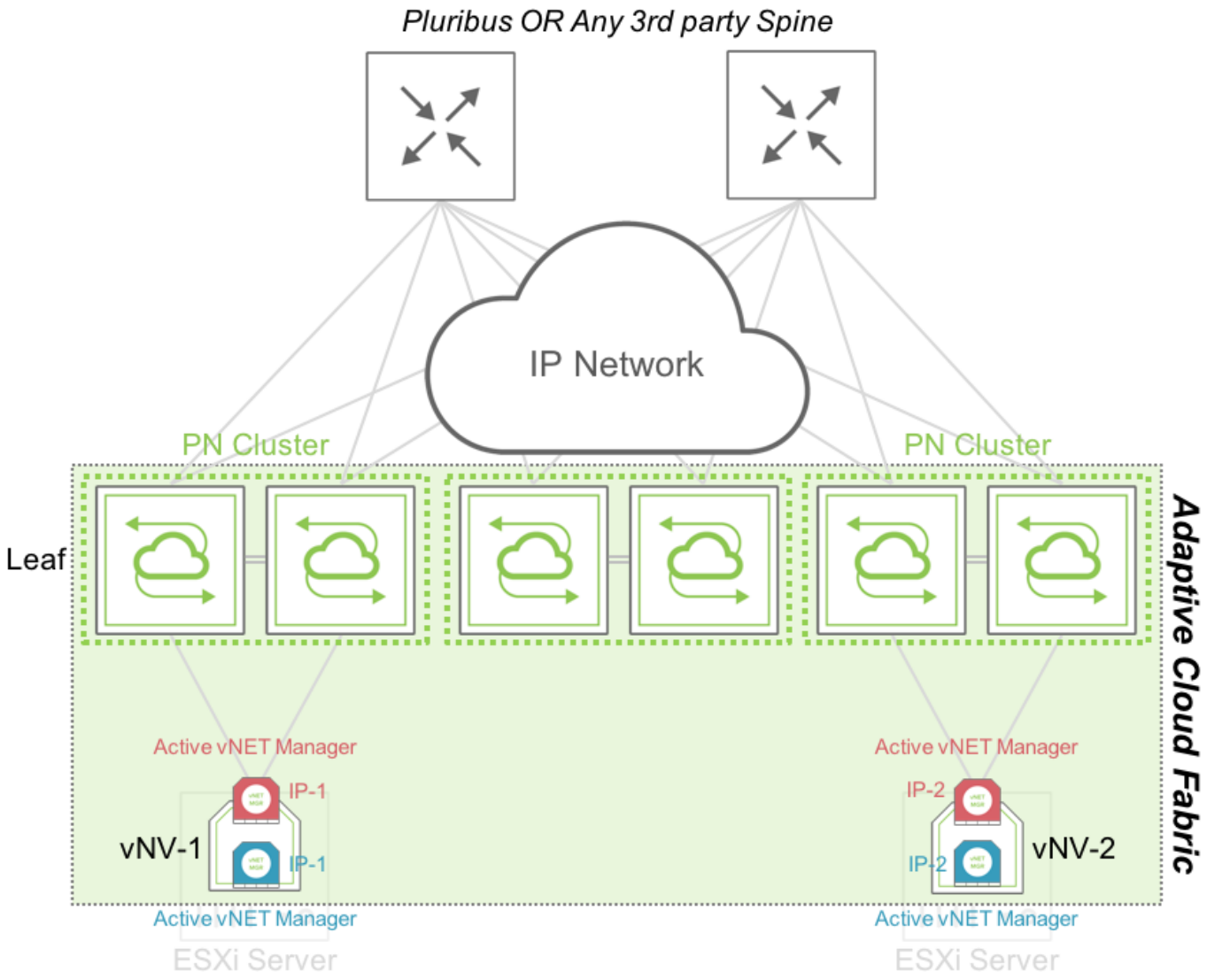
In this configuration, you can easily create 2 vNVs sharing only the knowledge of the vNET but won't use any control plane messages to offer a single IP address to manage the vNET, both are active at the same time.

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 UNUM and Netvisor ONE® Terms

To review the **Glossary of UNUM and Netvisor ONE® Terms**, please refer to to the online document.

## vNV Deployment Scenarios and UNUM

As the Pluribus 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.

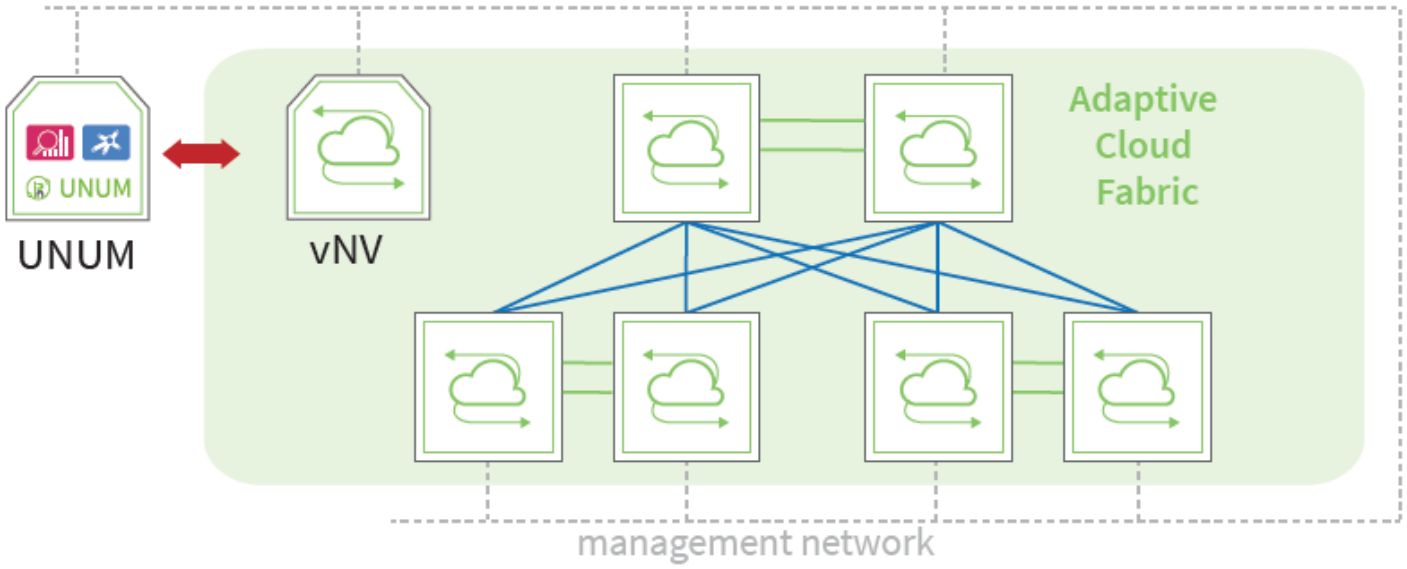
Enter Virtual Netvisor or vNV, a virtual machine running Pluribus Networks' Netvisor® ONE operating system. Once deployed and added to a fabric, vNV enables administrators to offload multiple vNET managers and other services from physical network nodes, freeing up switch resources.

**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.

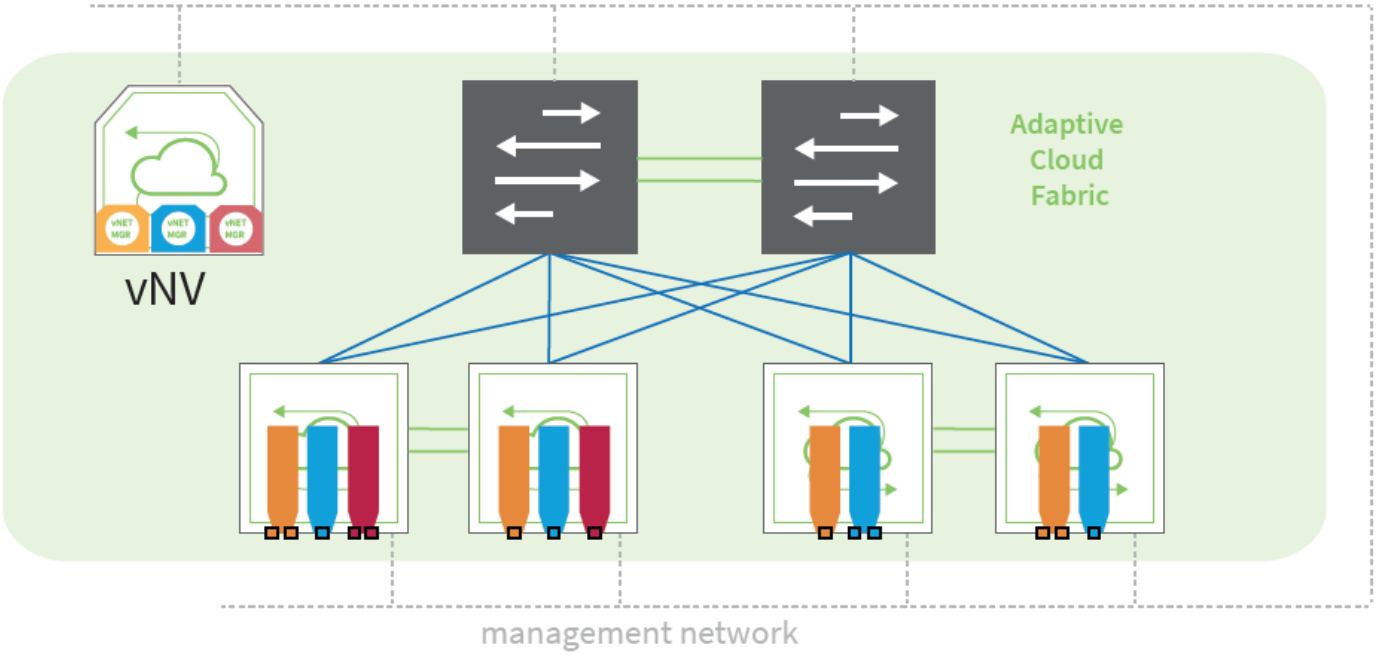
Virtual Netvisor typically deploys with UNUM™, Pluribus Networks' fabric management, analytics, and automation platform, to move resource consumption from a fabric node known as a seed switch onto the more powerful vNV virtual machine. vNV is especially helpful in environments with high numbers of traffic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

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




*Virtual Netvisor - Deployment Scenario Example 1*

**Virtual Netvisor - Pluribus 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 **Pluribus 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

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

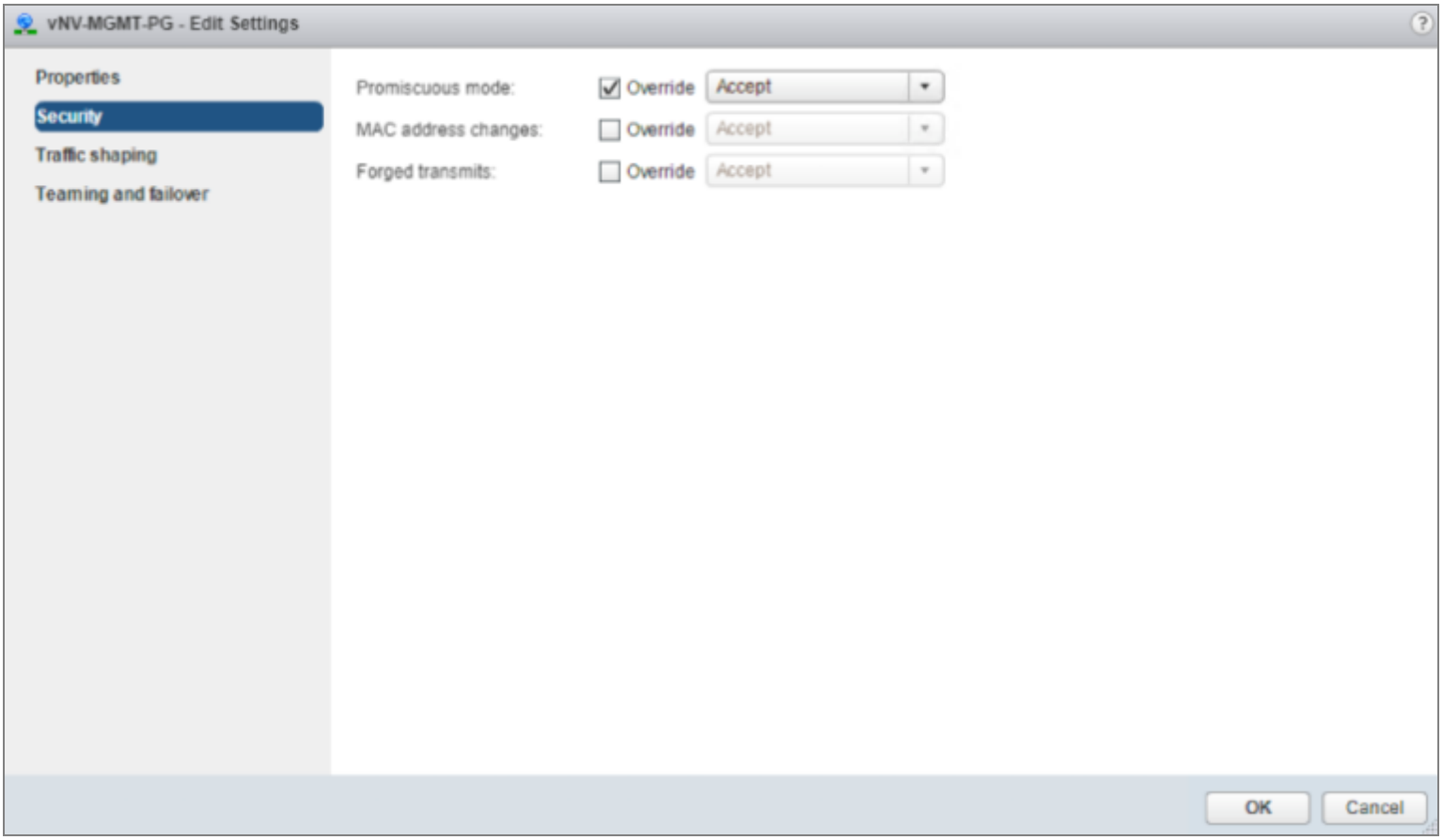
- 6.5.x
- 6.7.x
- 7.0.x

### ESXi Host Prerequisites for vNV

VMware ESXi hosts have the following prerequisites:

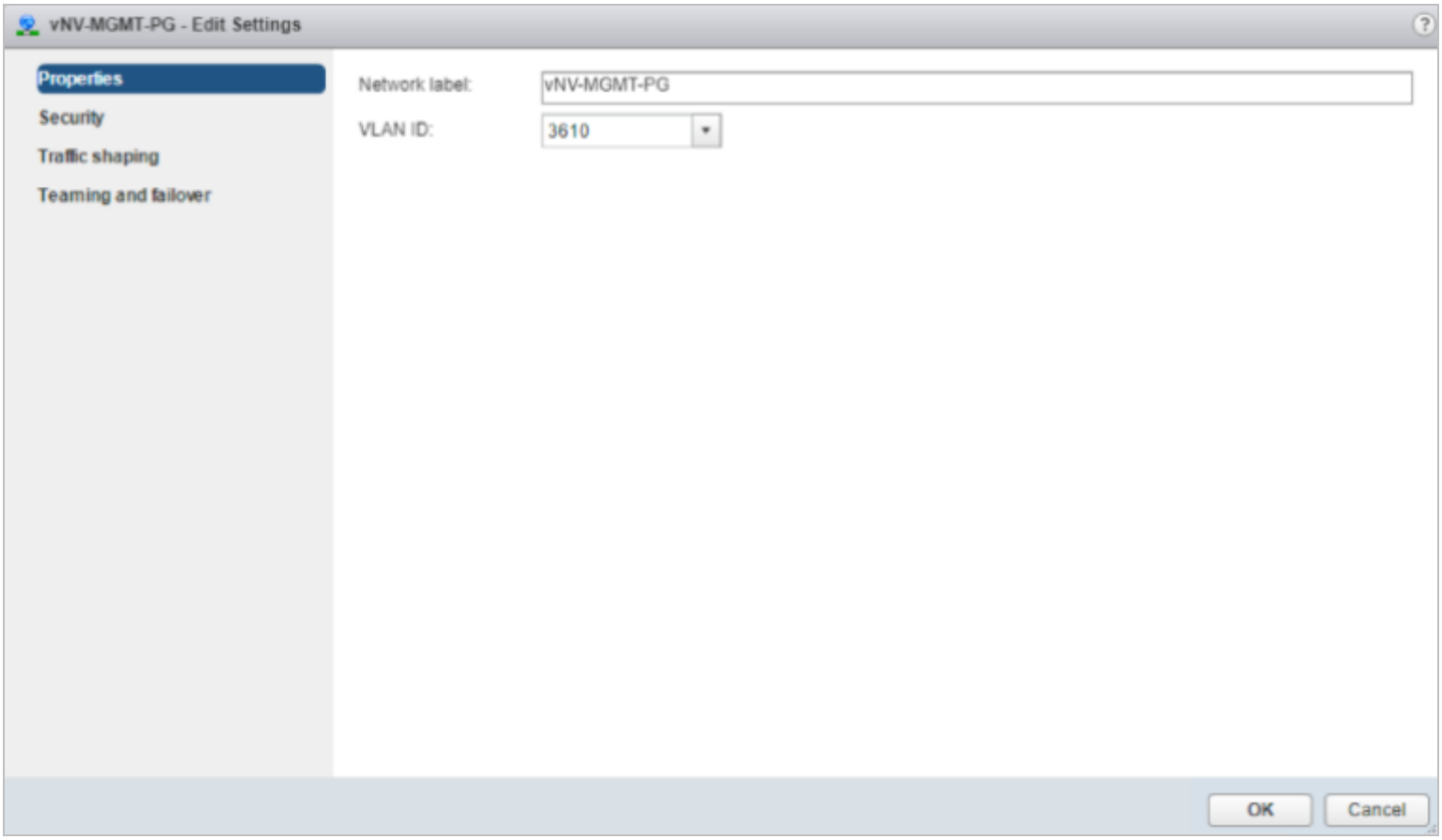
- You have installed and prepared vCenter Server for host management using the instructions from VMware.
- You have VMware vSphere Client installed or vCenter integration Plug-in in your browser.
- A specific license is not required for the vCenter Server to deploy vNV OVA images.
- You must configure 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)




*Management Port-group Security Configuration on a Standard vSwitch*

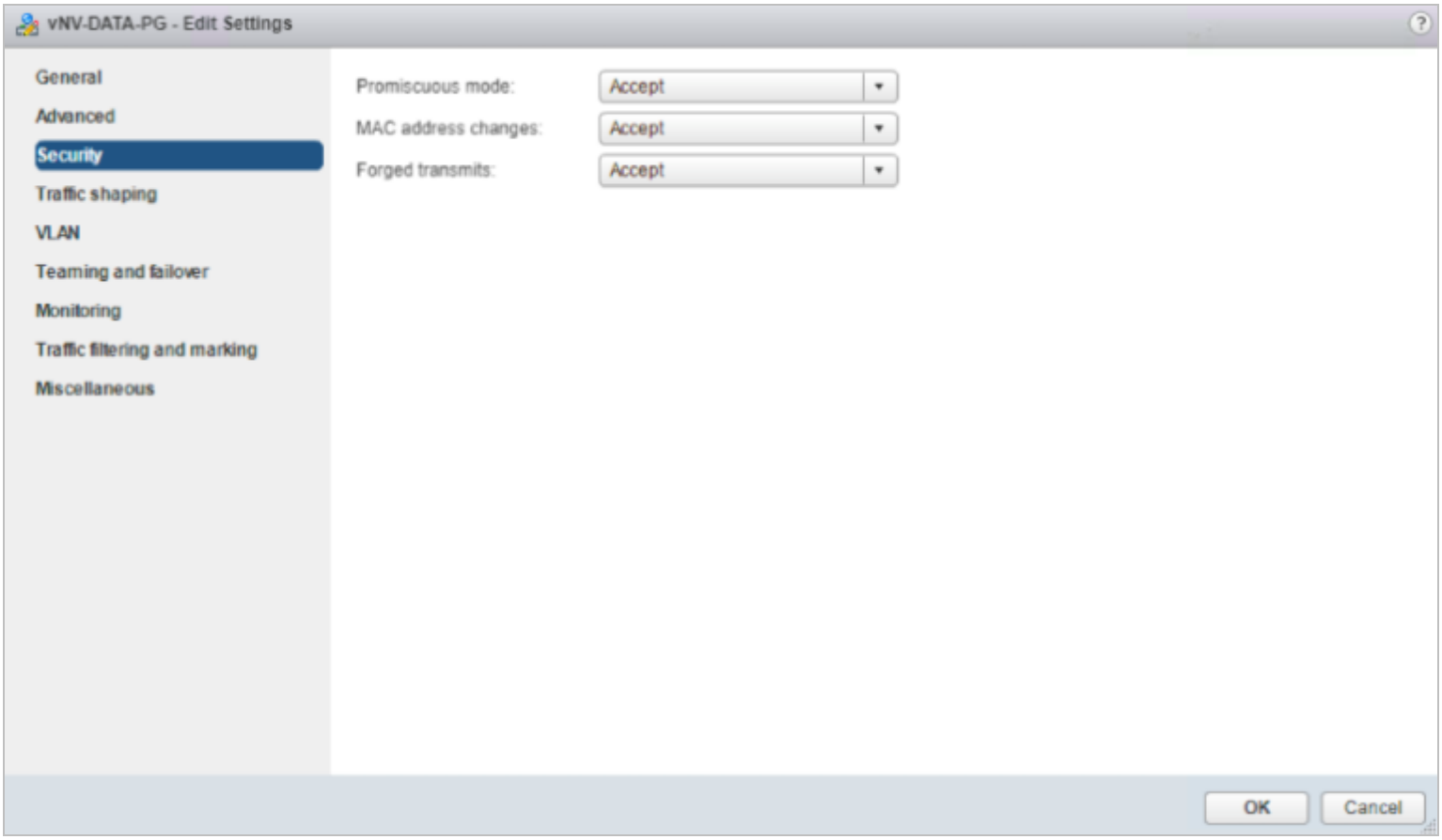
# Deploying Virtual Networks with vNV (cont'd)




*Management Port-group VLAN Configuration on a Standard vSwitch*

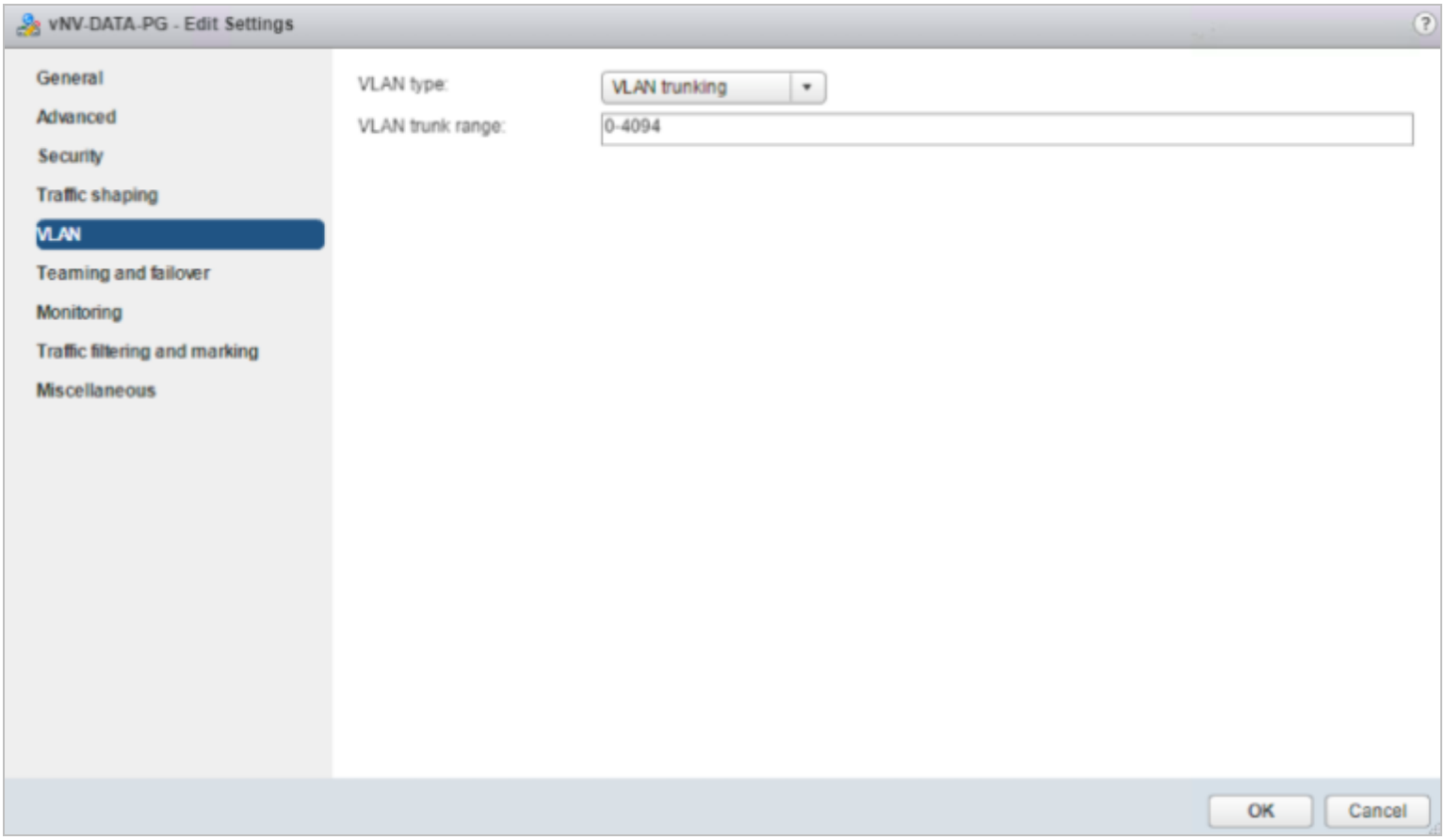
## Deploying Virtual Networks with vNV (cont'd)

- 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.




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

# Deploying Virtual Networks with vNV (cont'd)




*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:

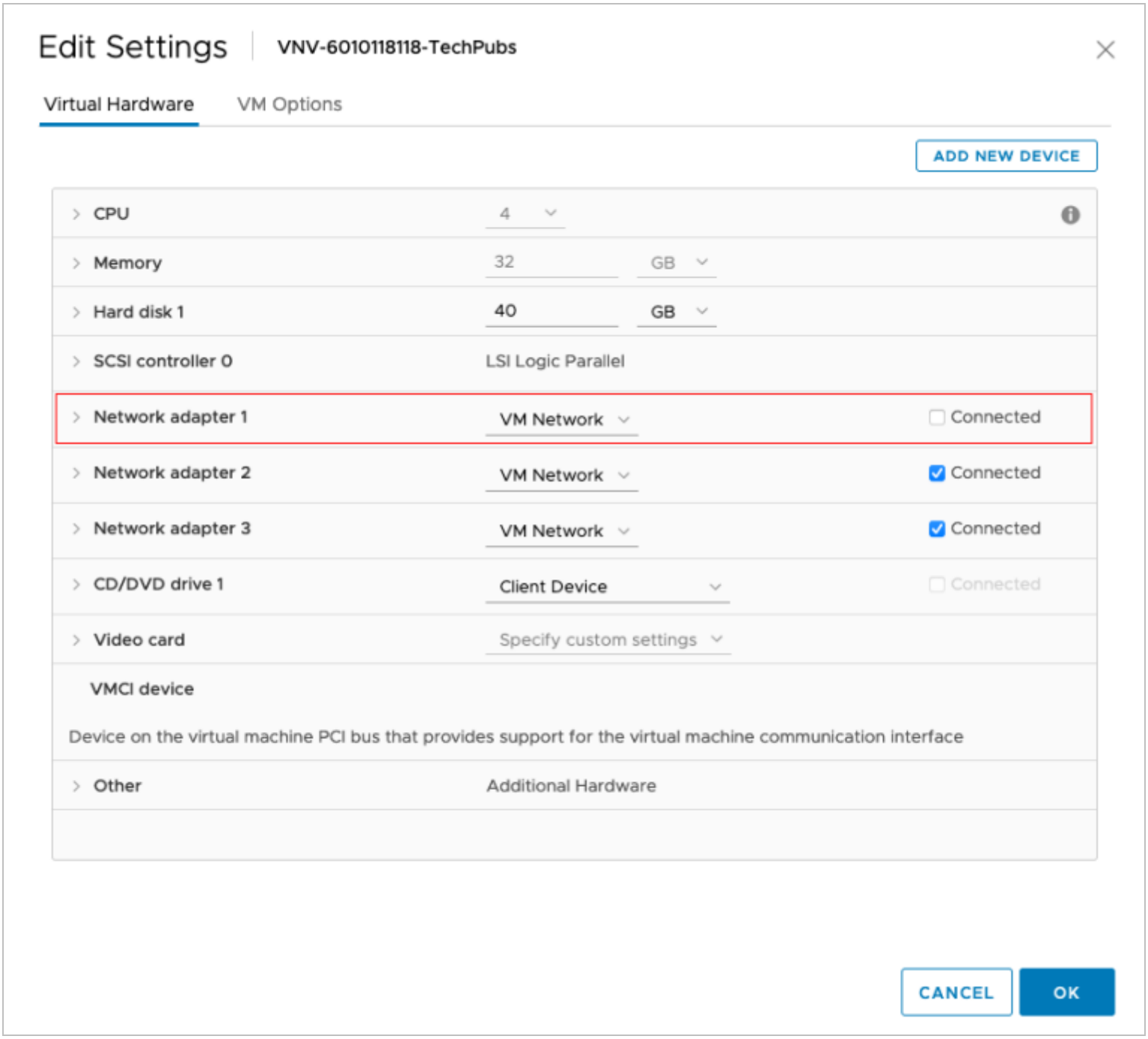
| Function | Component | Minimum Requirement |
|---|---|---|
| **vNV** | Hardware Version | 9 and higher |
| | Platform | 64-bit |
| | Type | Other 64-bit Linux |
| | Processor | 4 vCPU (2 cores hyper threaded) |
| | Memory | 32 GB |
| | Disk | 40 GB in Thin Provisioning |
| | CPU Speed | > 2 GHz |

*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




*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

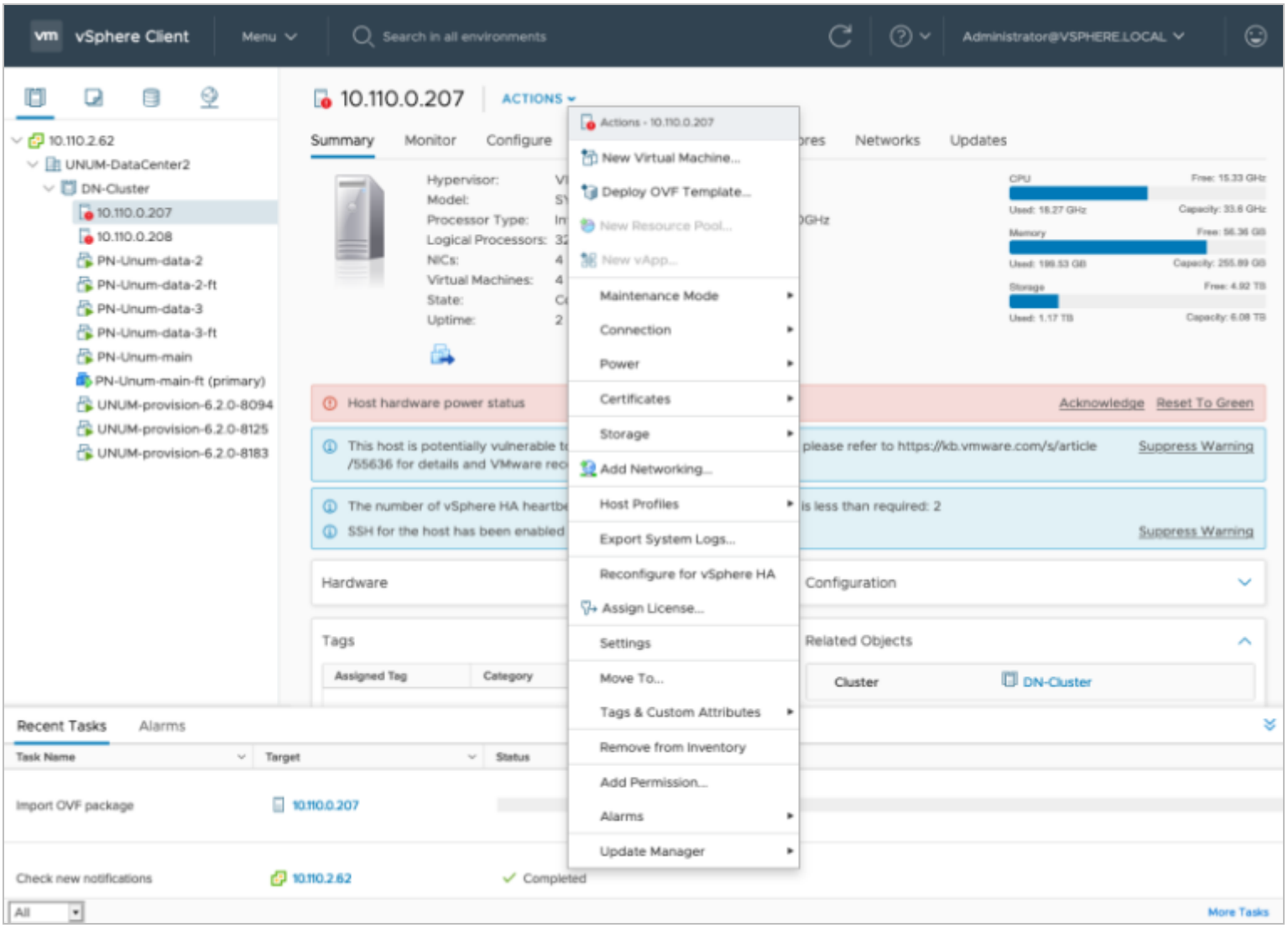
Before you begin this procedure, you must do the following:

- 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:**

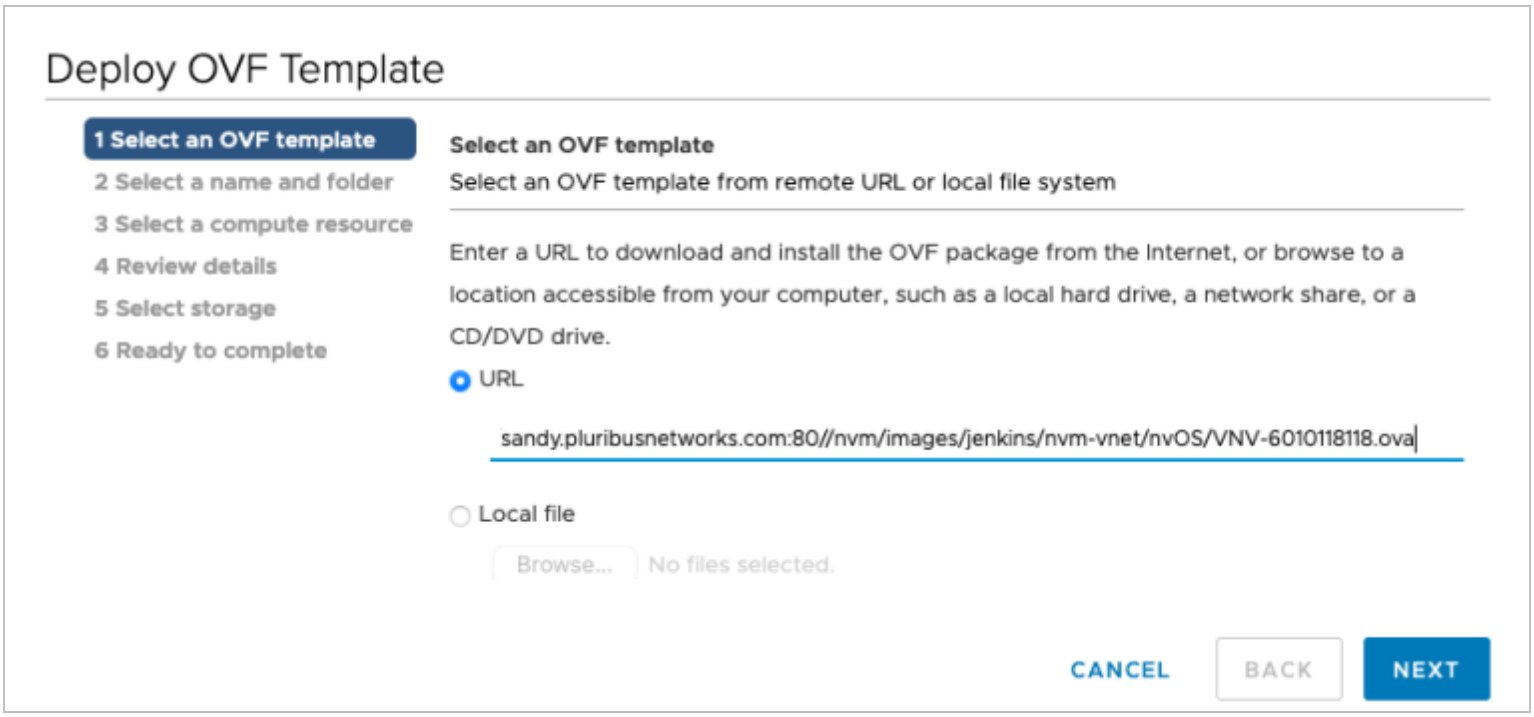
**Step 1** From the vSphere Web Client, find the host, cluster and/or resource pool where you want to deploy and right click > **Deploy OVF Template...**




*Deploy OVF Template*

## Deploying Virtual Networks with vNV (cont'd)

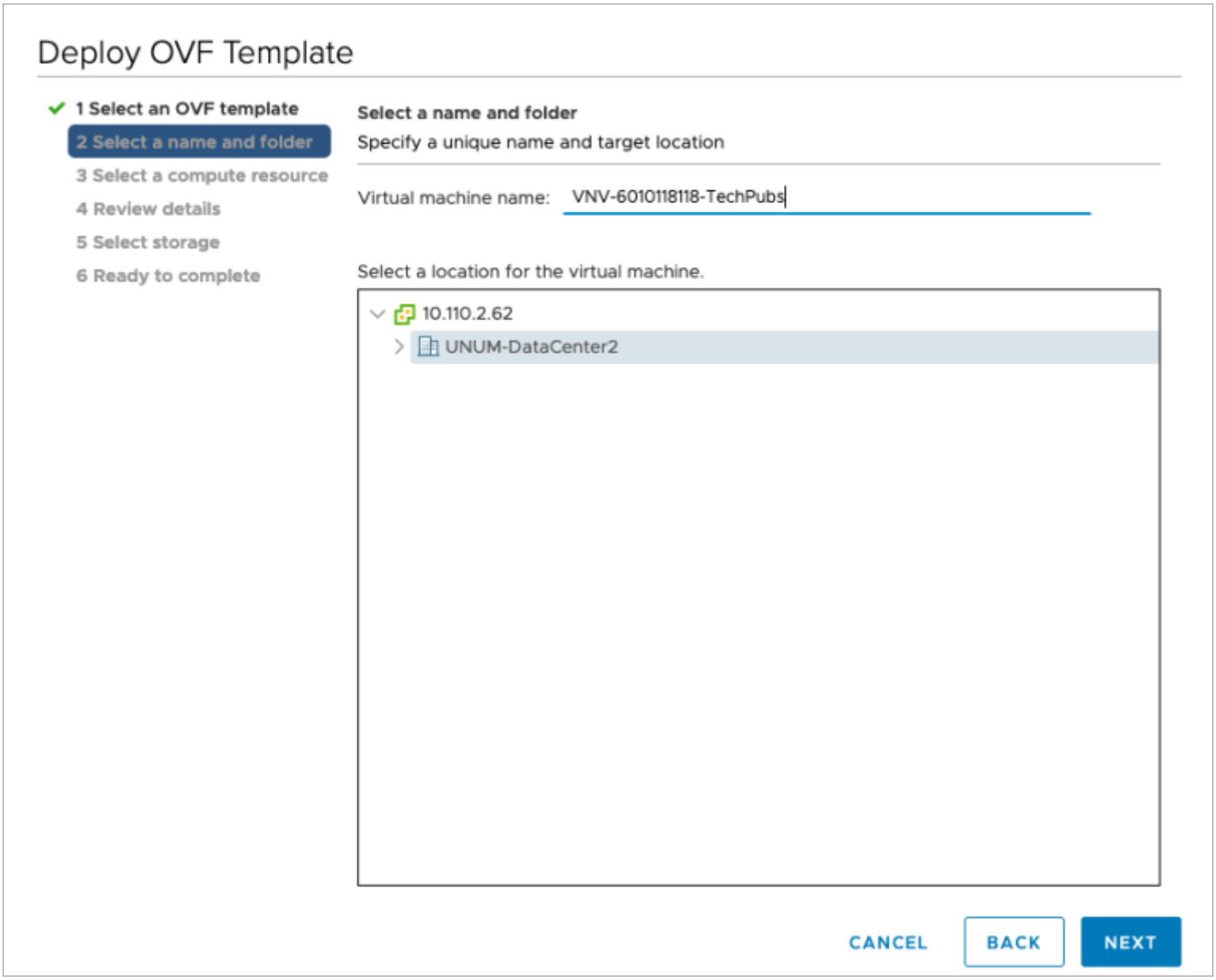
**Step 2** In the Template screen, specify the location of the **OVA** file as a **URL** or **Local File** and click **Next**.




*Select OVF Template*

## Deploying Virtual Networks with vNV (cont'd)

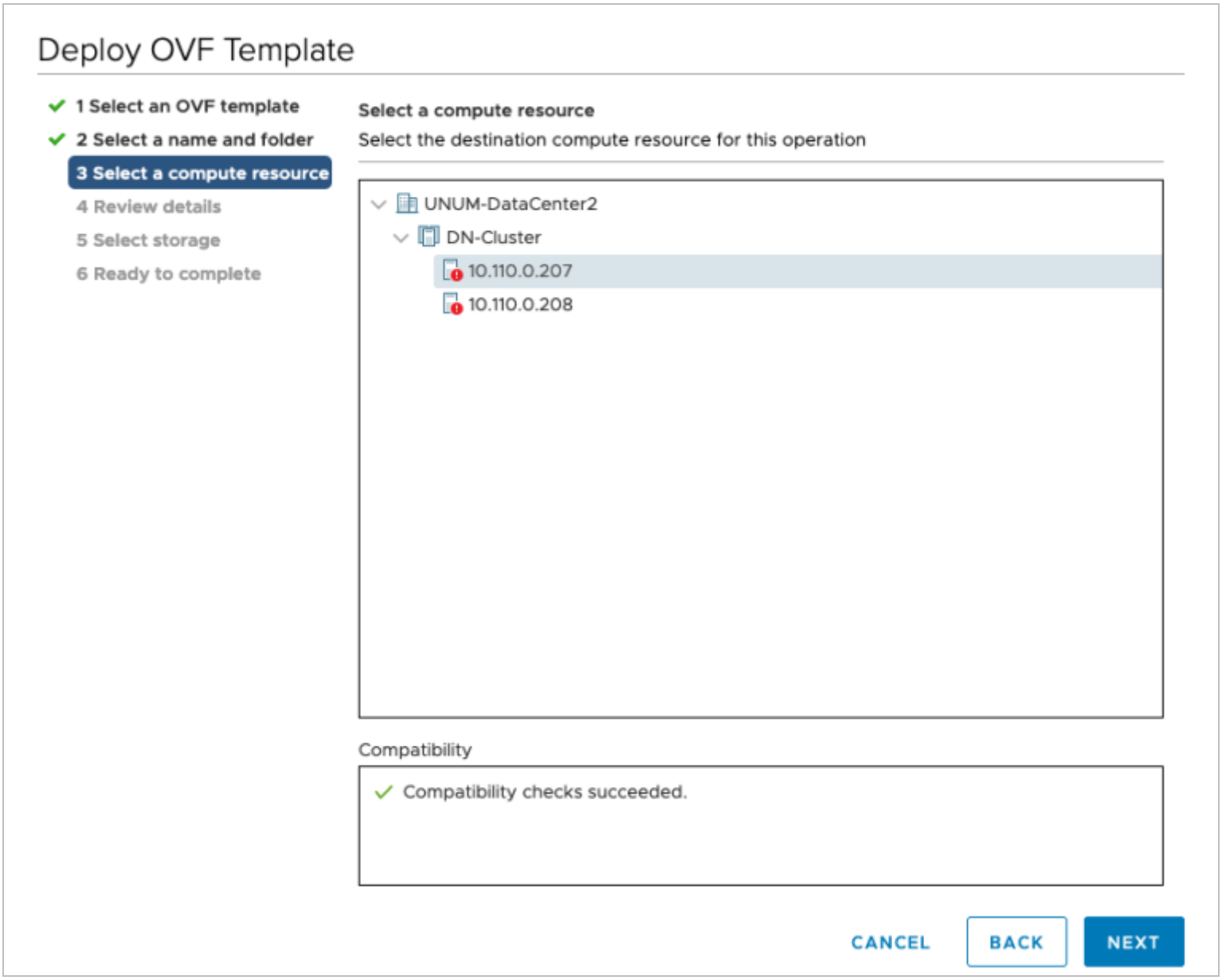
**Step 3** In the **Name** and **Location** screen, specify the **Name** of your **vNV VM** and select datacenter or folder in the **Inventory** and click **Next**.




*Select Name and Location*

## Deploying Virtual Networks with vNV (cont'd)

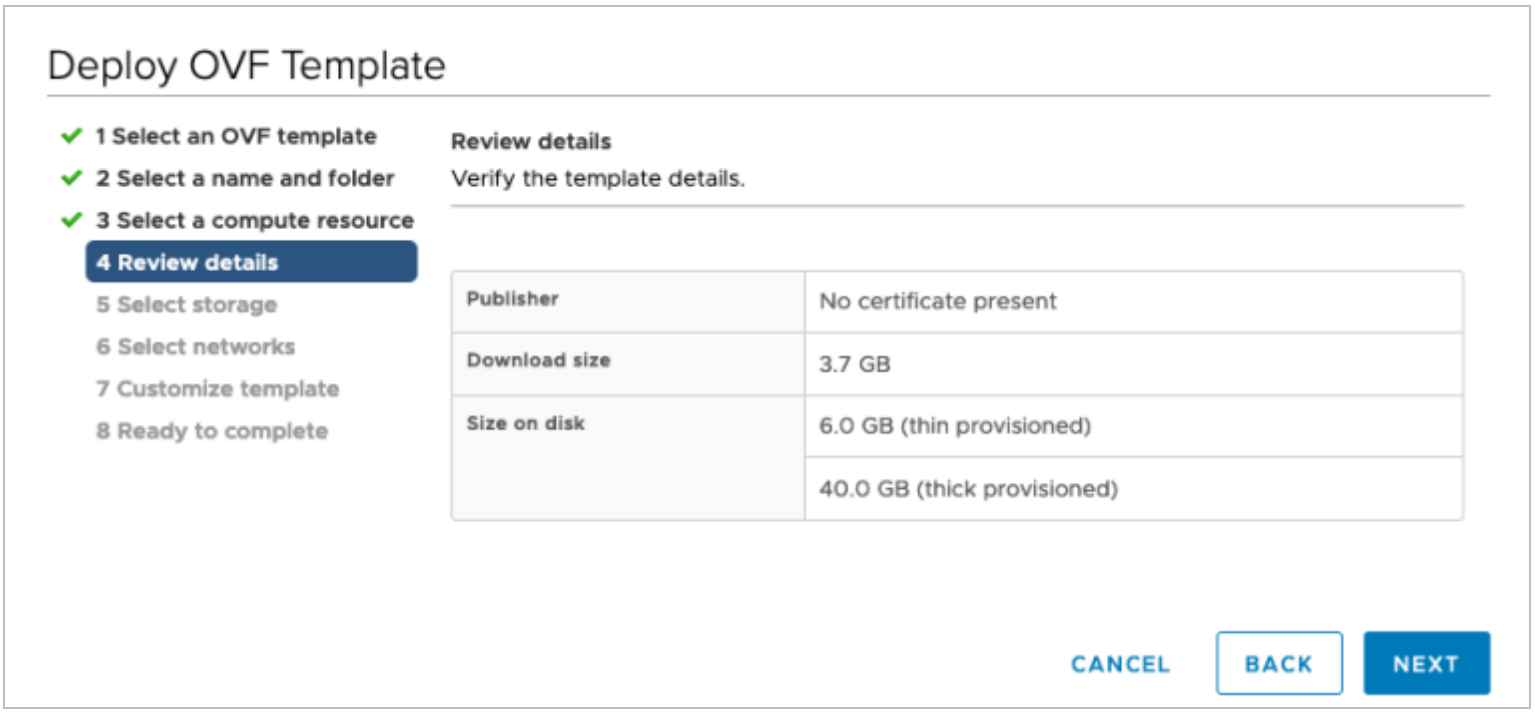
**Step 4** In the **Select a Compute Resource** screen, select host or cluster or resource pool and click **Next**.




*Select Host or Cluster*

## Deploying Virtual Networks with vNV (cont'd)

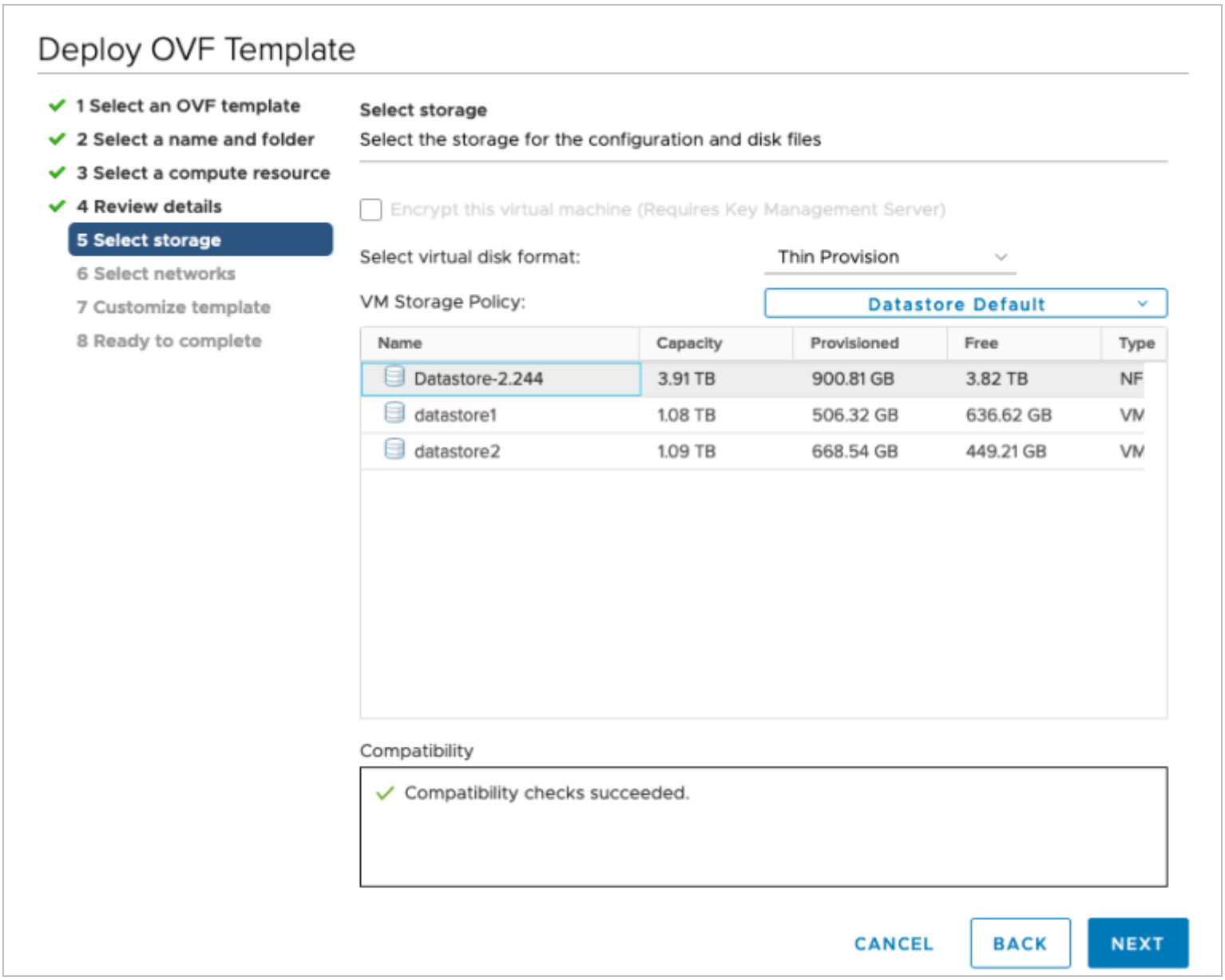
**Step 5** In the **Review Details** screen, verify details and click **Next**.




*Review Details*

## Deploying Virtual Networks with vNV (cont'd)

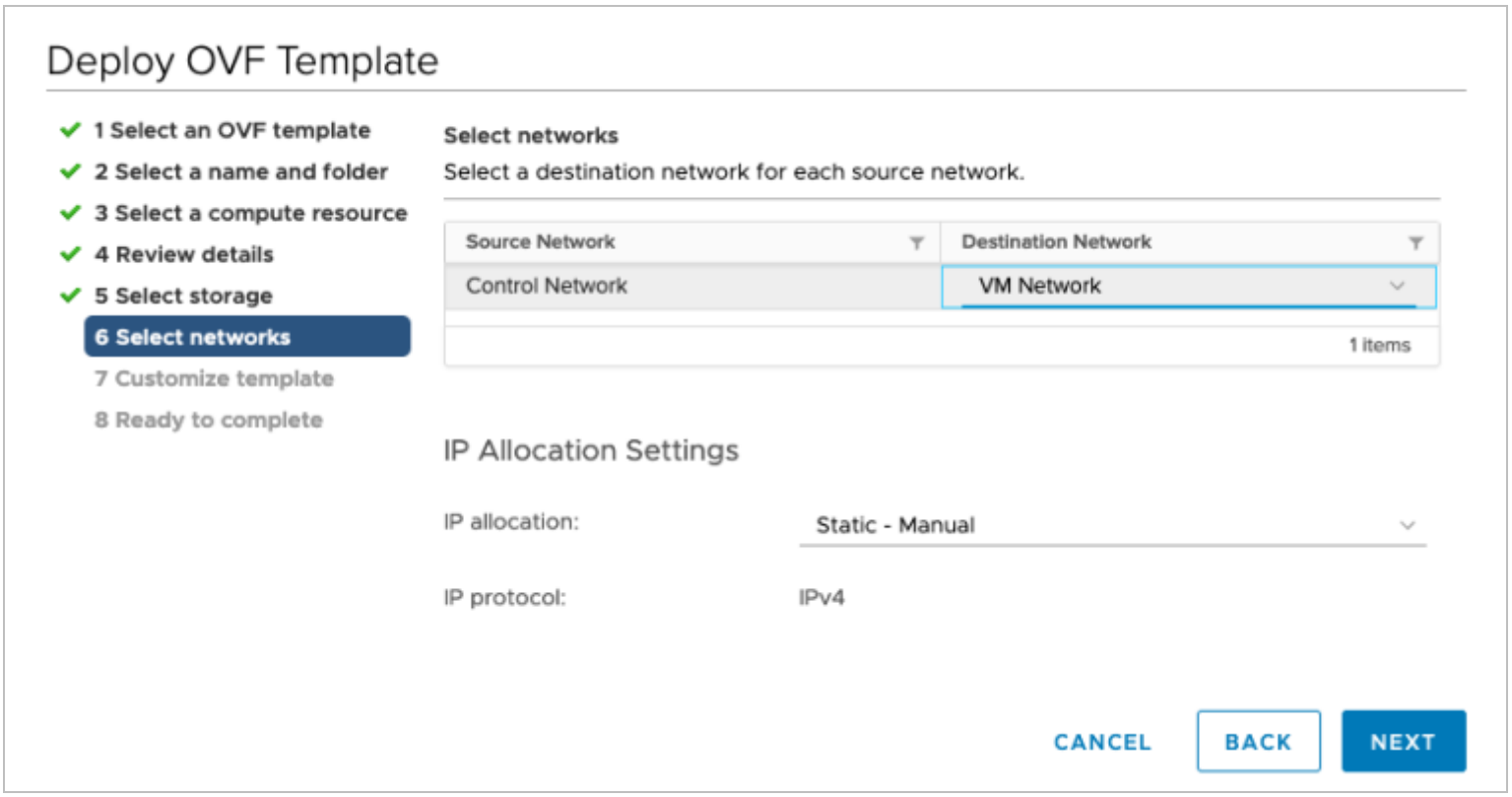
**Step 6** In the **Select Storage** screen, select **Thin Provision** from Select virtual disk format list, select the desired **Datastore** and click **Next**.




*Select Storage*

## Deploying Virtual Networks with vNV (cont'd)

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



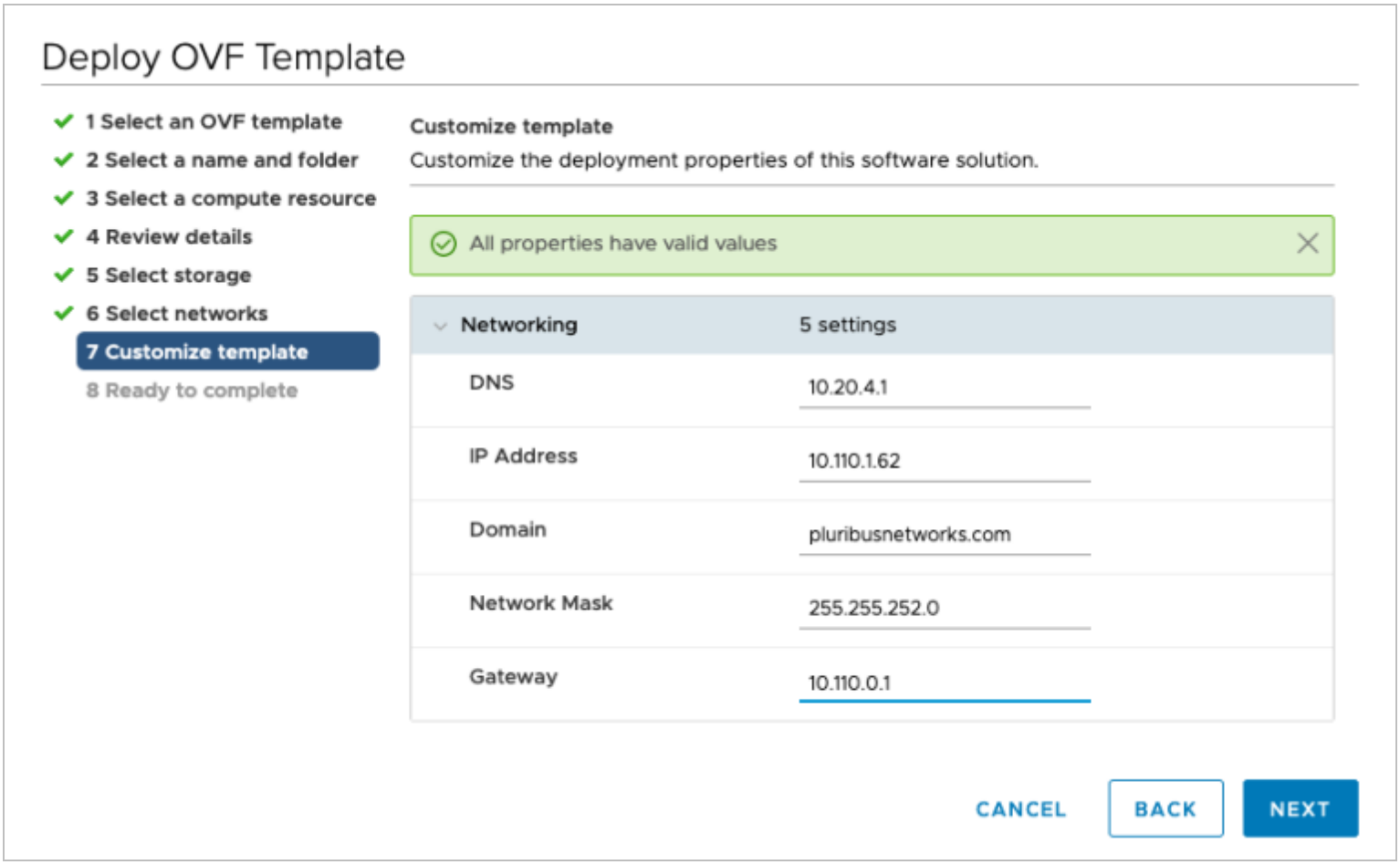

*Select Networks*

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.

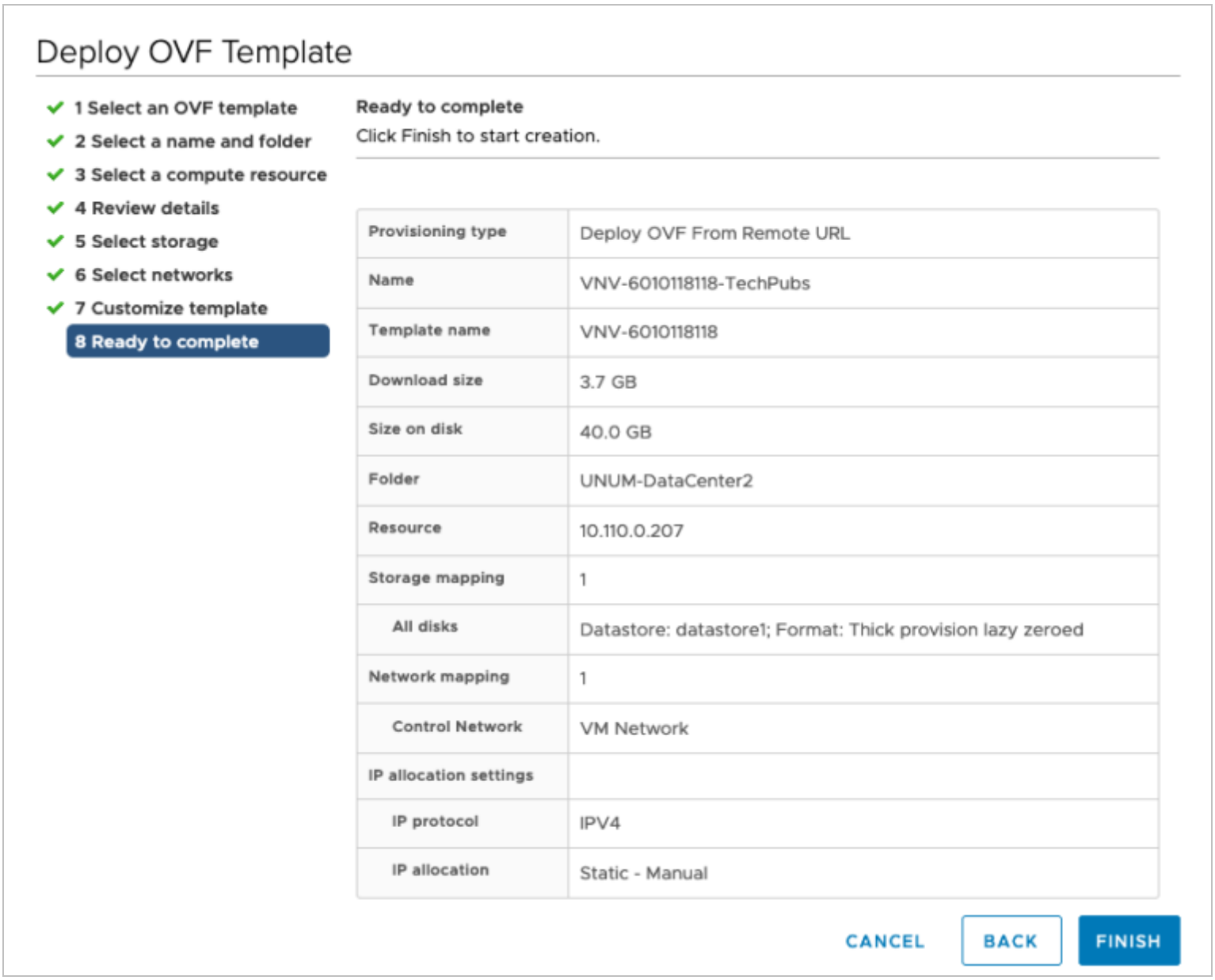



*Static IP Network Configuration*

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**.



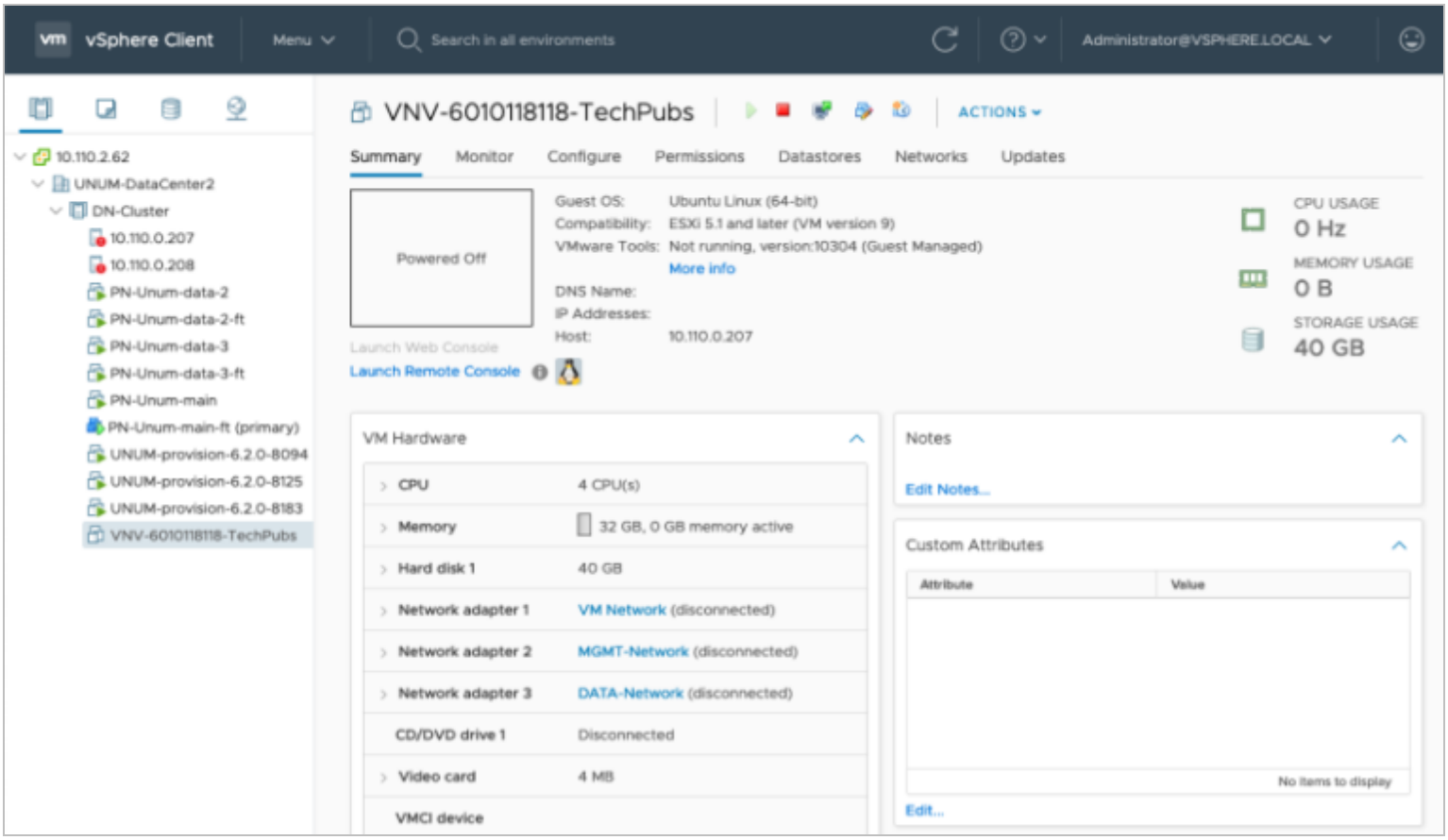

*Ready to Complete and Review Settings*

## 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.

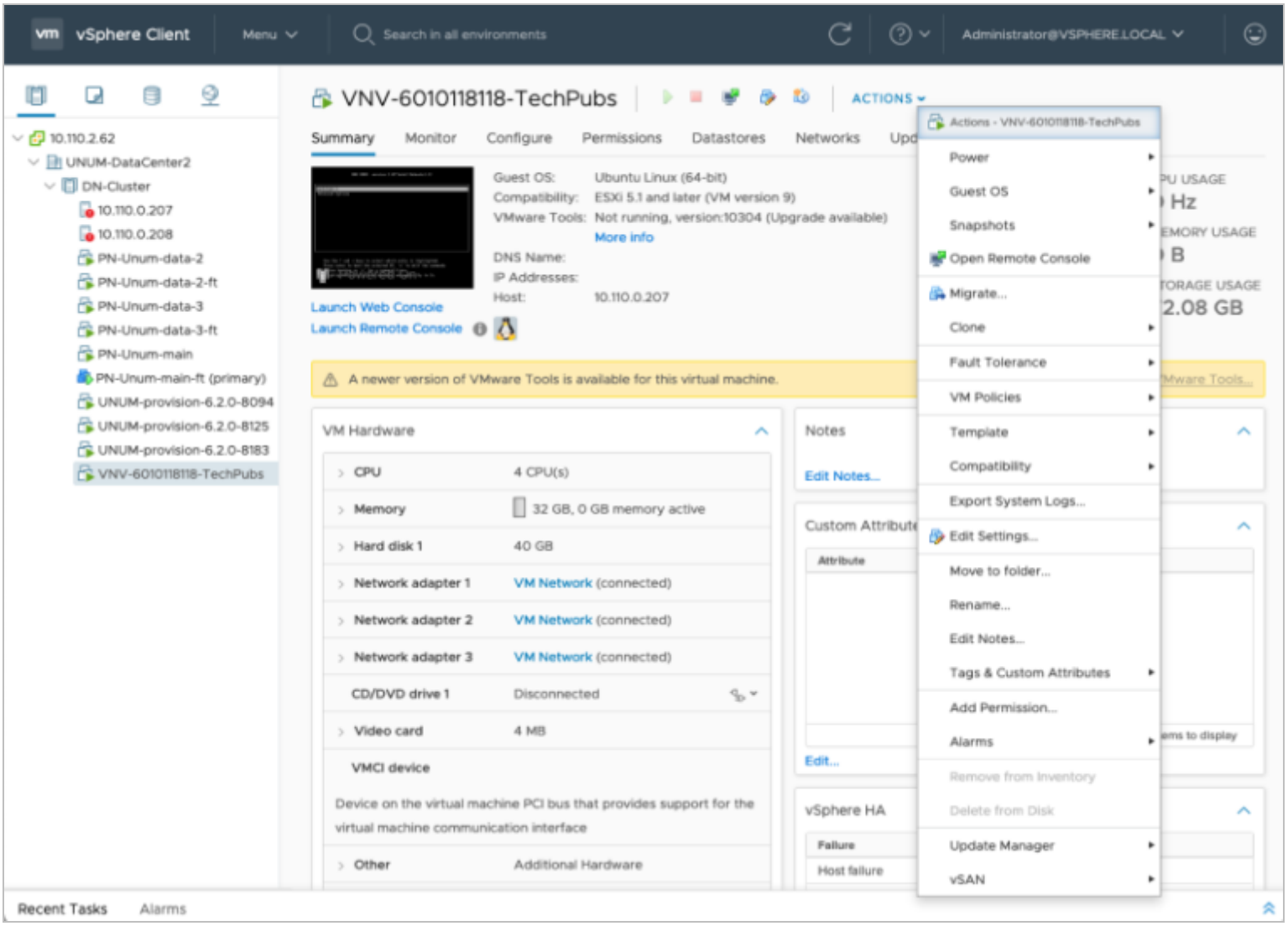
Once the installation successfully completes, ensure the vNV VM is powered **OFF** and proceed to the next step.




*vNV Instance Powered Off*

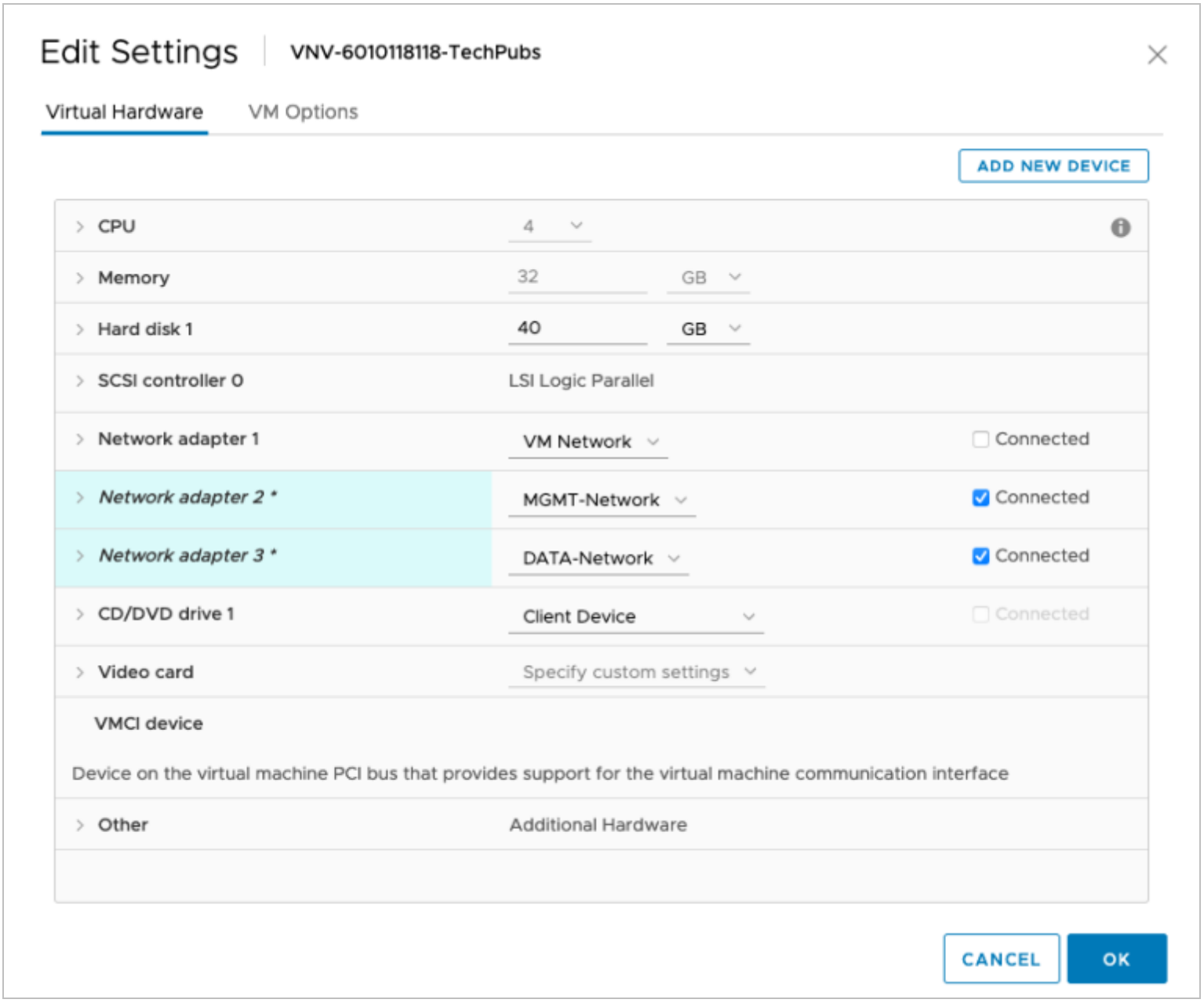
## Deploying Virtual Networks with vNV (cont'd)

**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."




*Edit Settings*

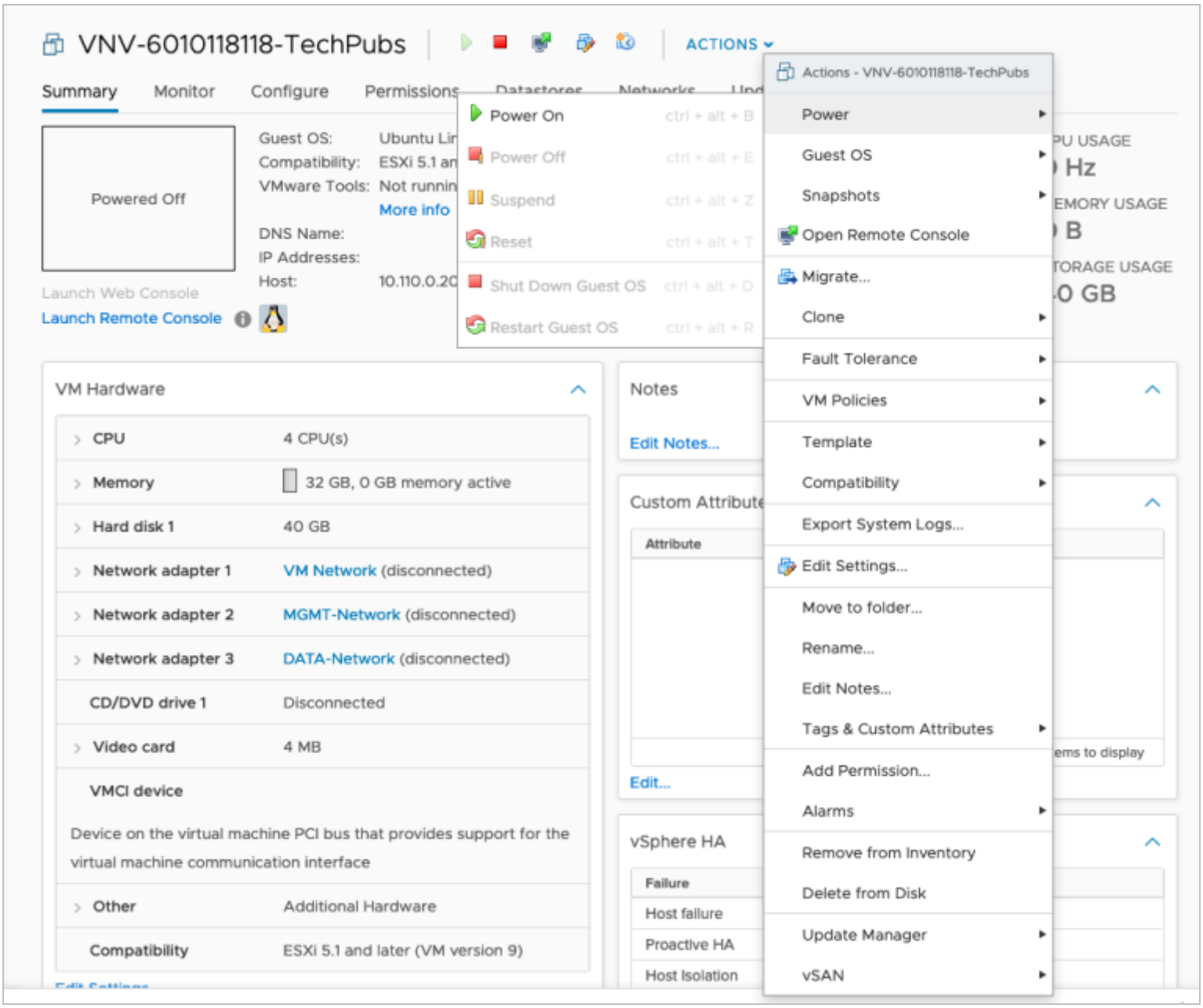
# Deploying Virtual Networks with vNV (cont'd)




*Network Settings*

# Deploying Virtual Networks with vNV (cont'd)

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




*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) >
```

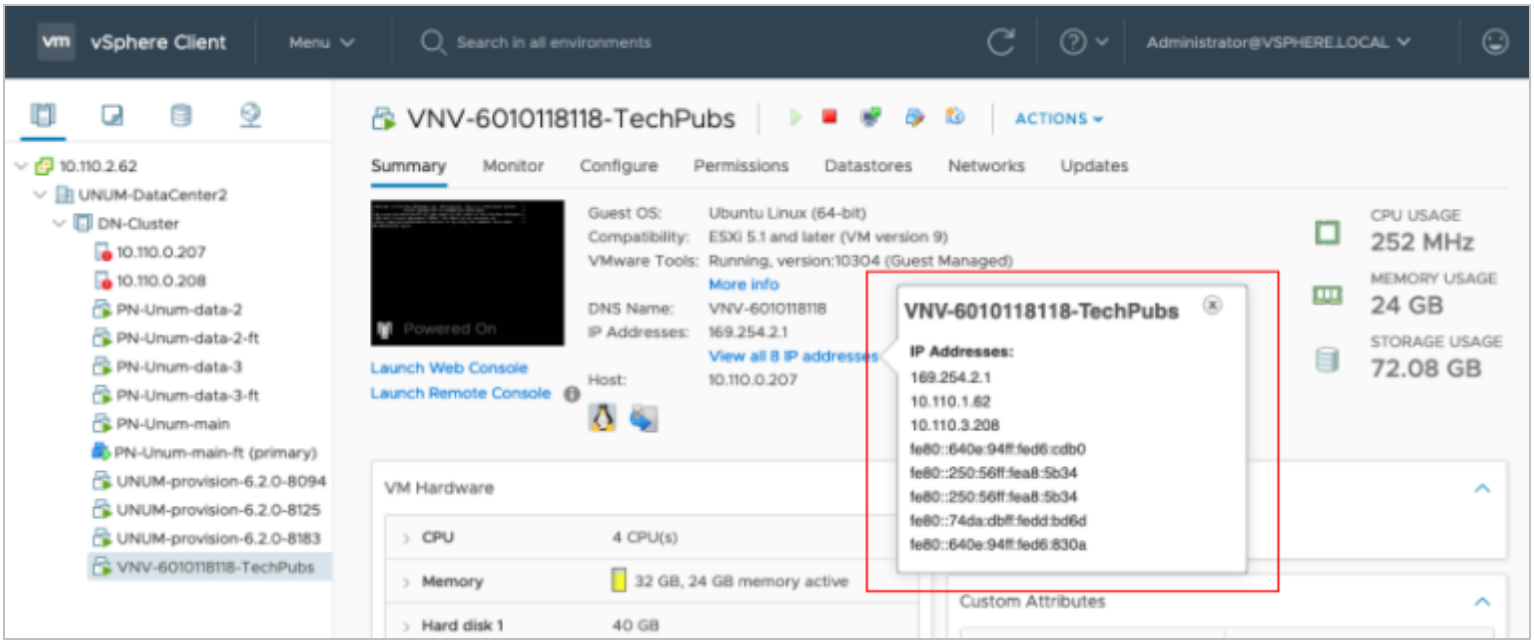
This fabric is a six node fabric using management network.

## Deploying Virtual Networks with vNV (cont'd)

**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:




*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) >
```

## Deploying Virtual Networks with vNV (cont'd)

**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
```

## Deploying Virtual Networks with vNV (cont'd)

**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
```

## Deploying Virtual Networks with vNV (cont'd)

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

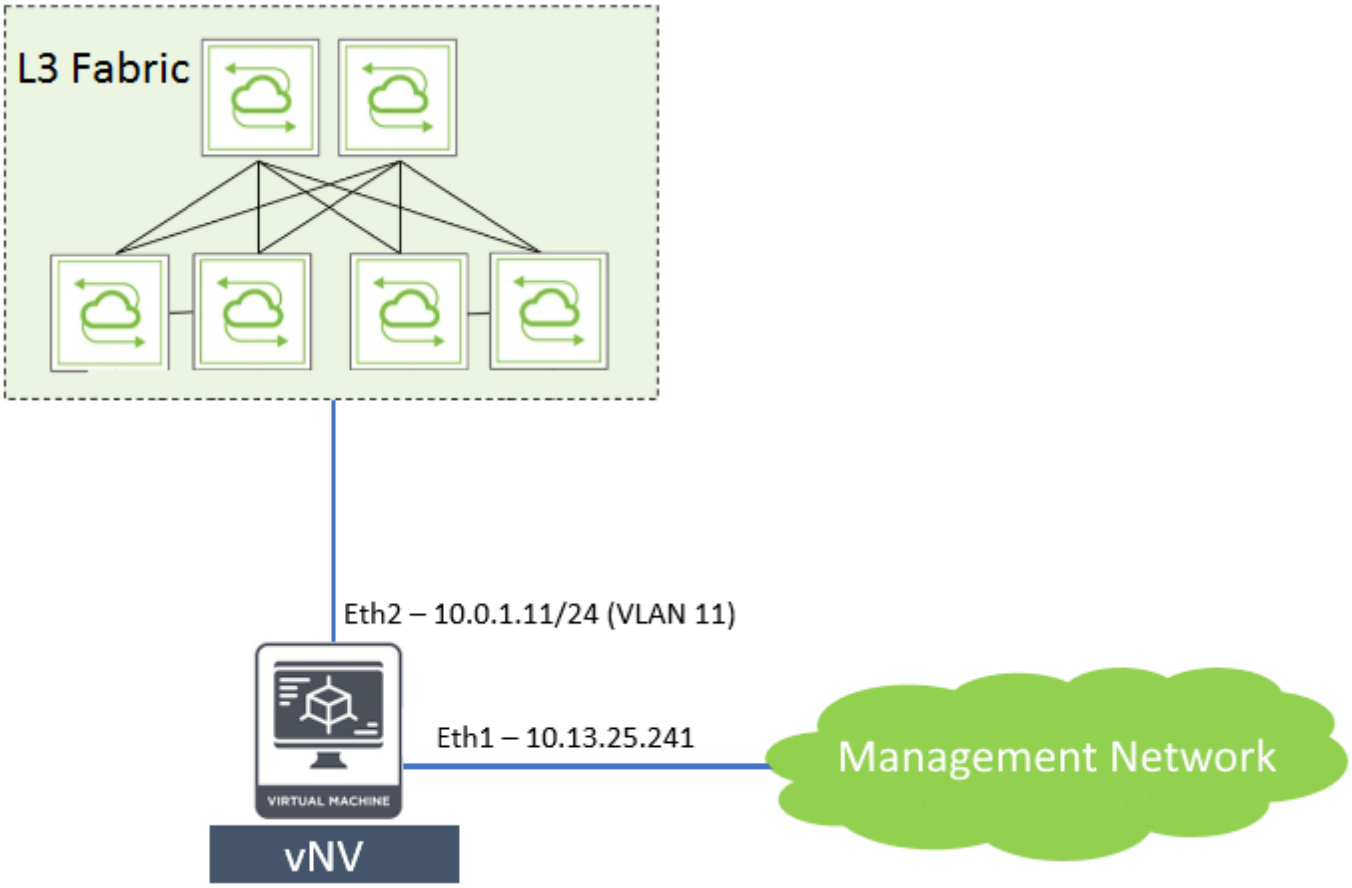
## Configure vNV - L3 Scenario

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).

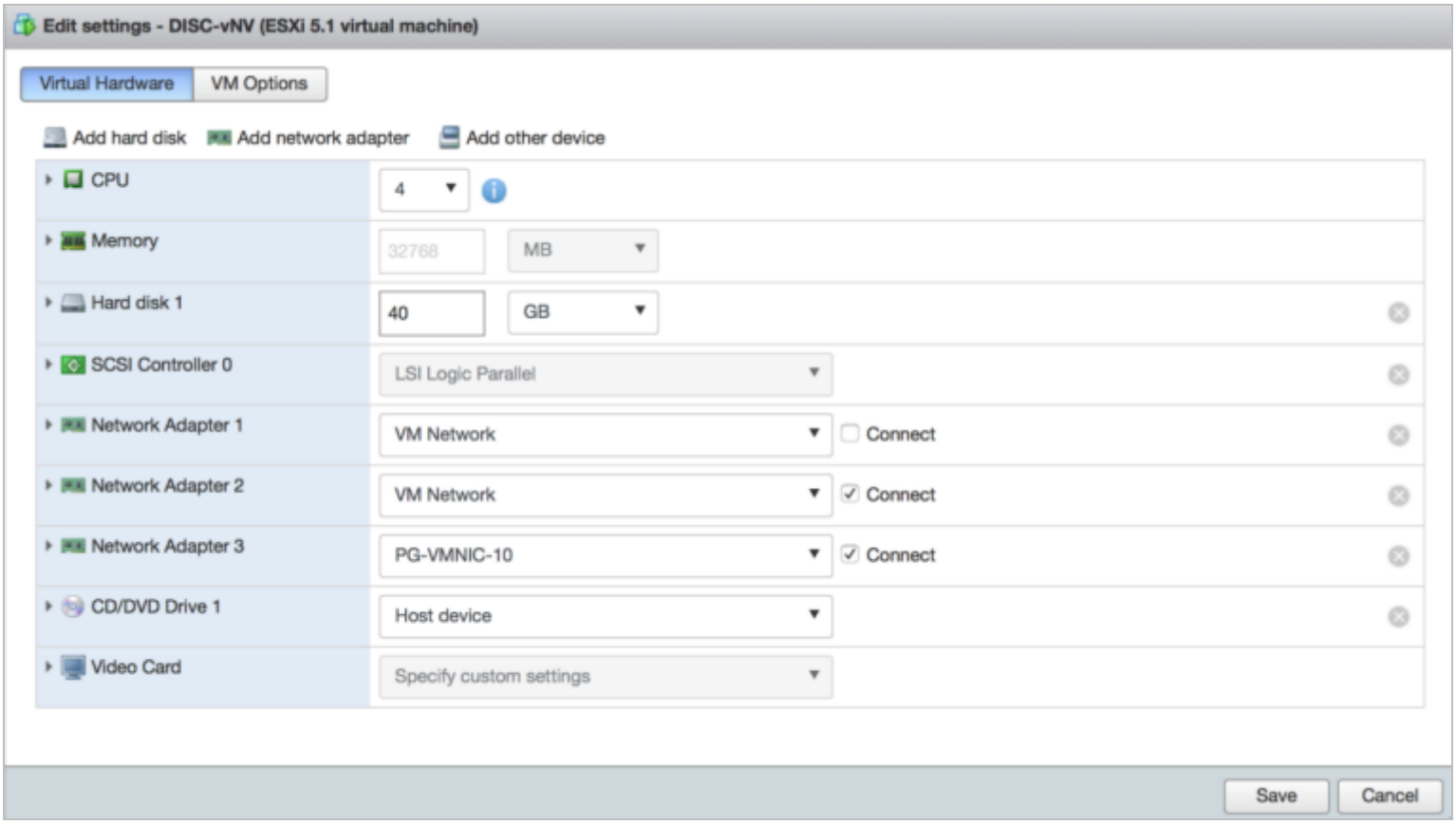
VLAN 11 connected via eth2 (Network Adapter 3) is used to communicate with the rest of the fabric.

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

## 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.




*VMware NIC Settings*

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

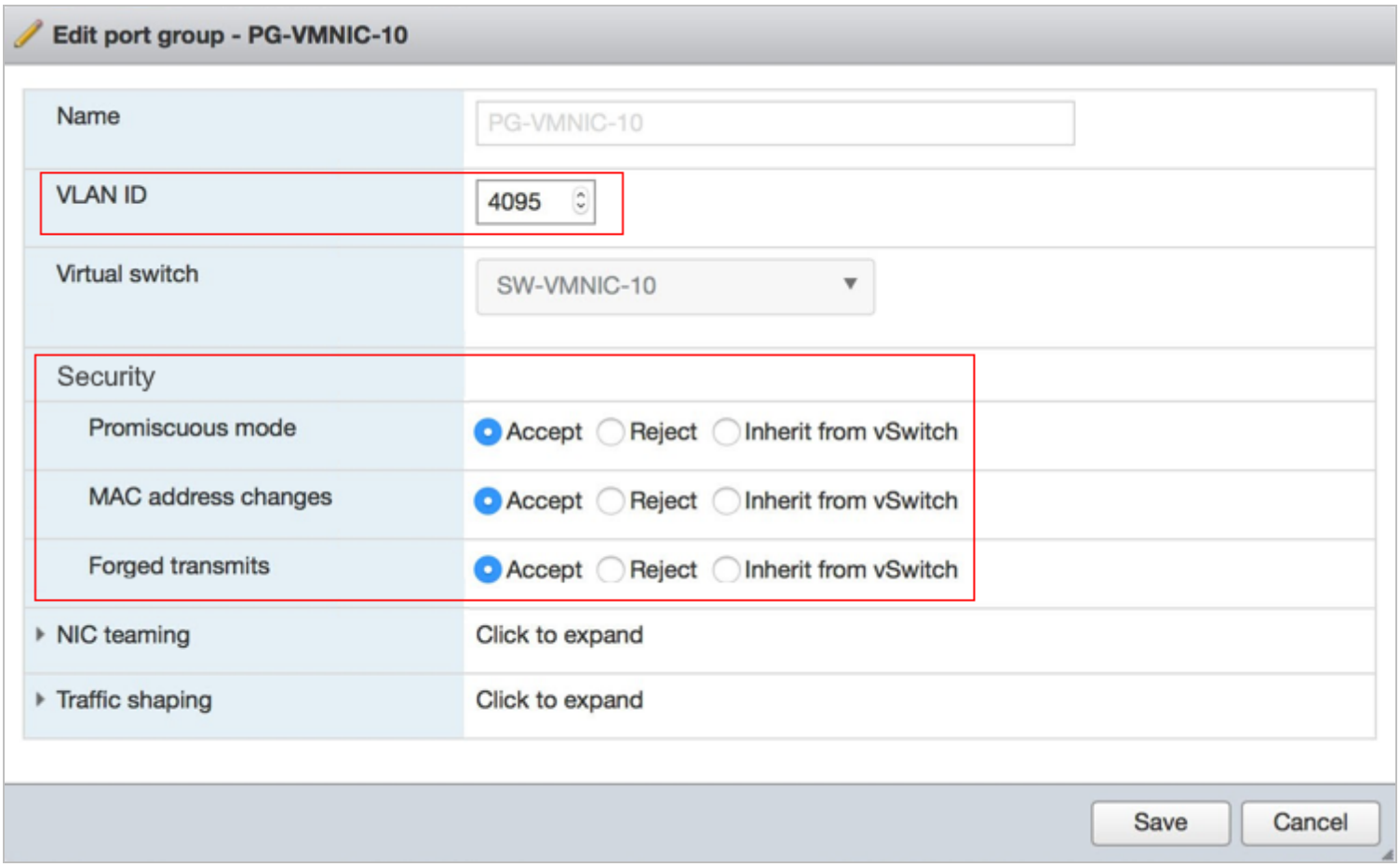
## Port Group Configuration

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.




*VMware Port Group Security Settings*

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

## 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 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 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

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

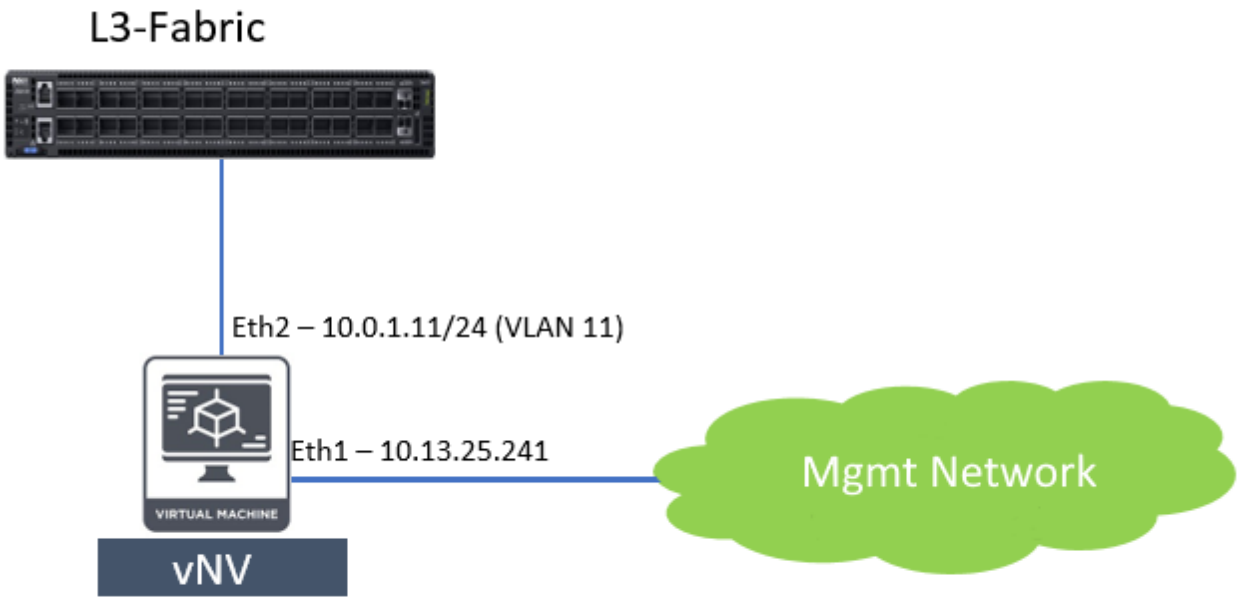
### Overview

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 Machine (VM) Properties

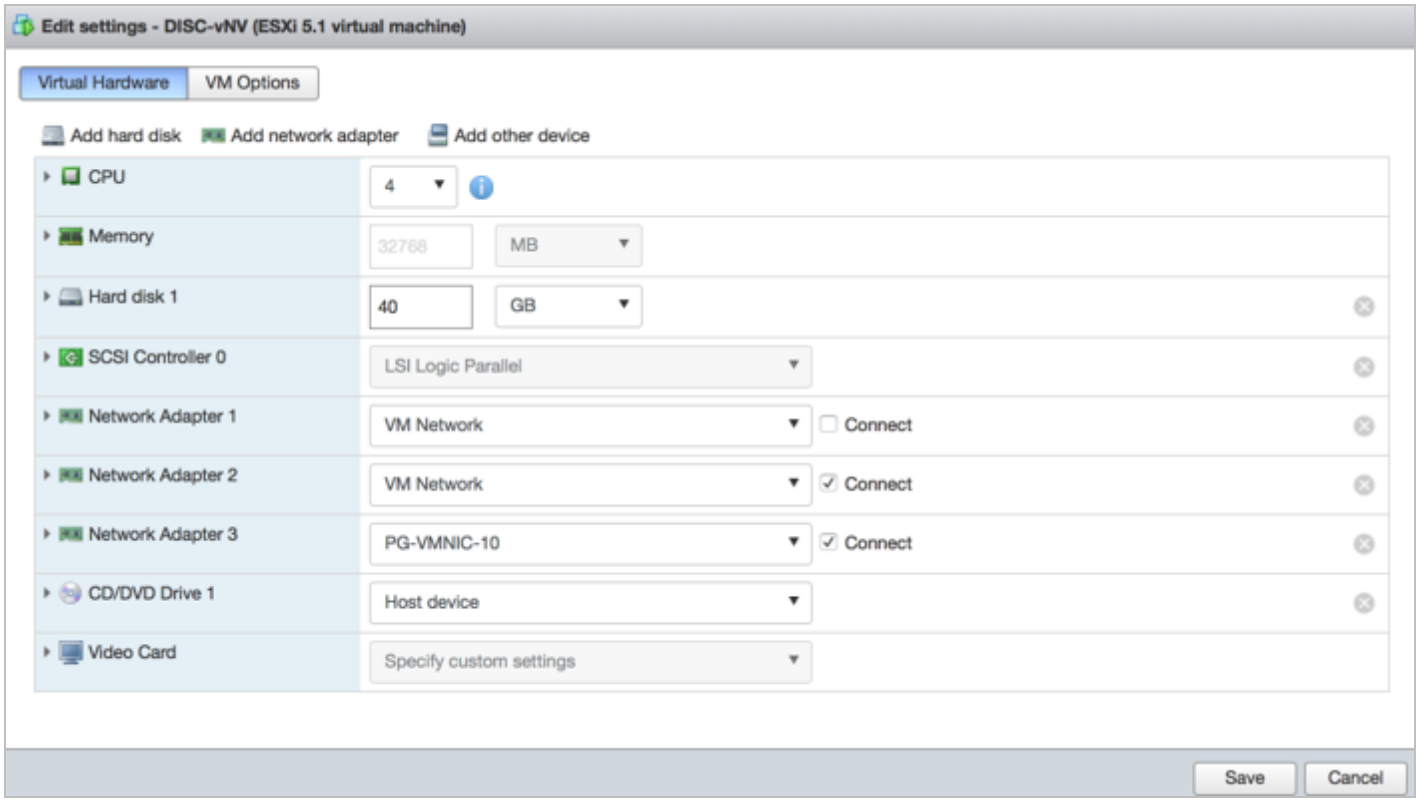
A vNV has three interfaces, **eth0**, **eth1**, and **eth2**.

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.



## Port Group Configuration

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

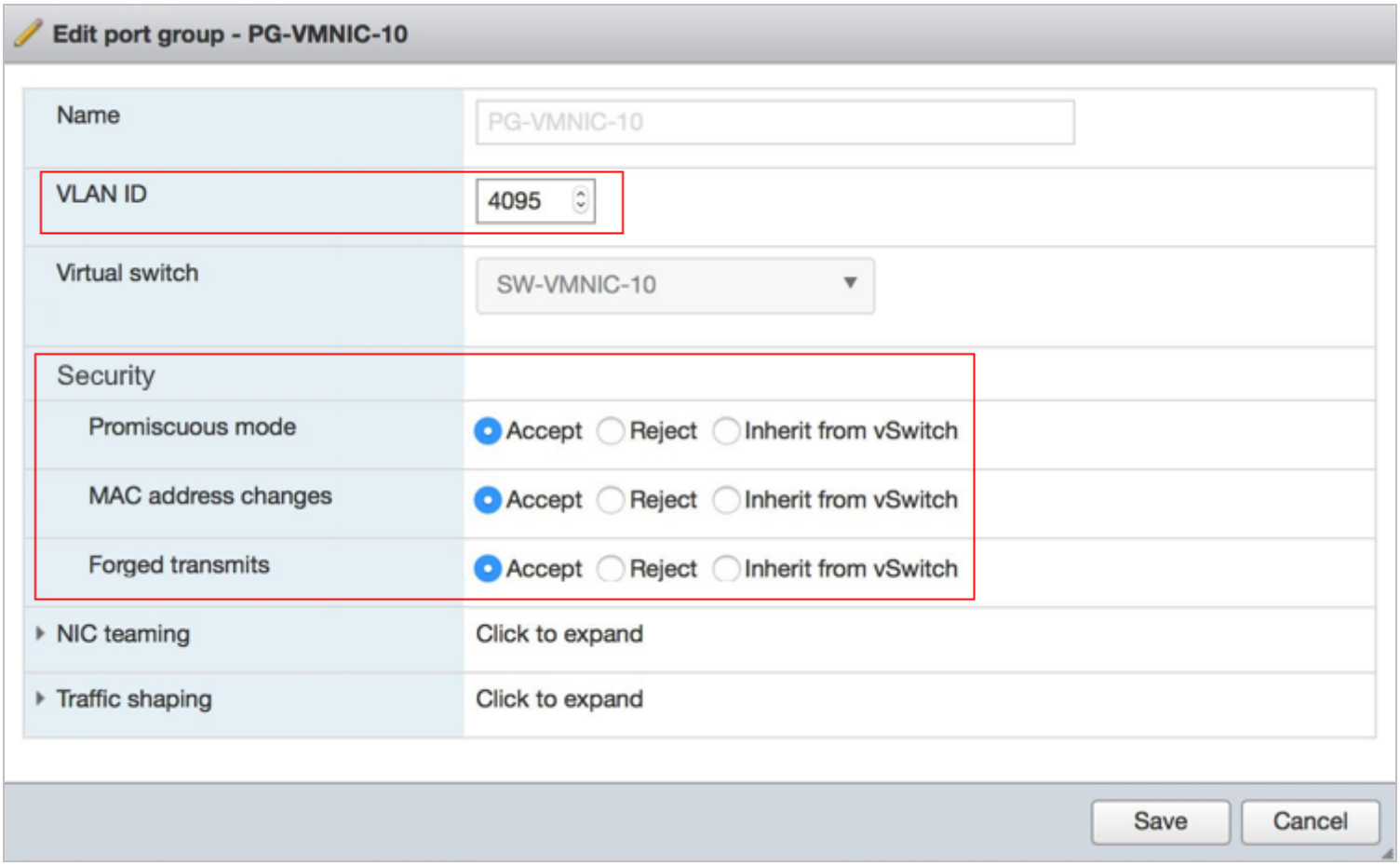
VLAN ID to - 4095

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.



## vNV Configuration

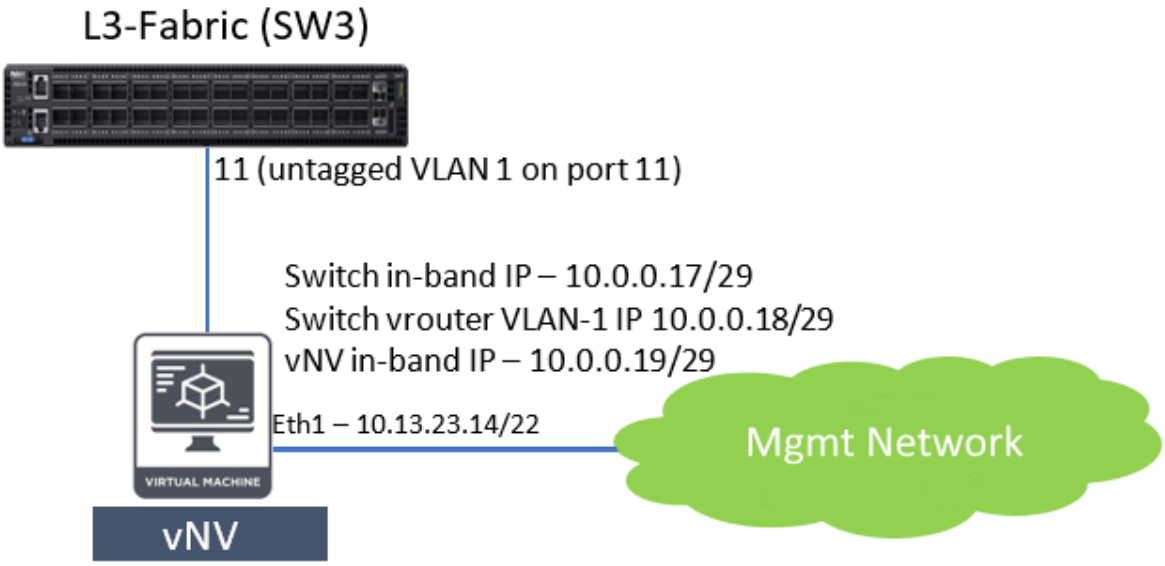
vNV needs to be configured such that it can communicate to inband IPs of other switches in the Fabric and also vNV's in-band network is reachable from the rest of the Fabric. There are two ways to connect vNV to the Fabric through inband:

- 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.




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**):

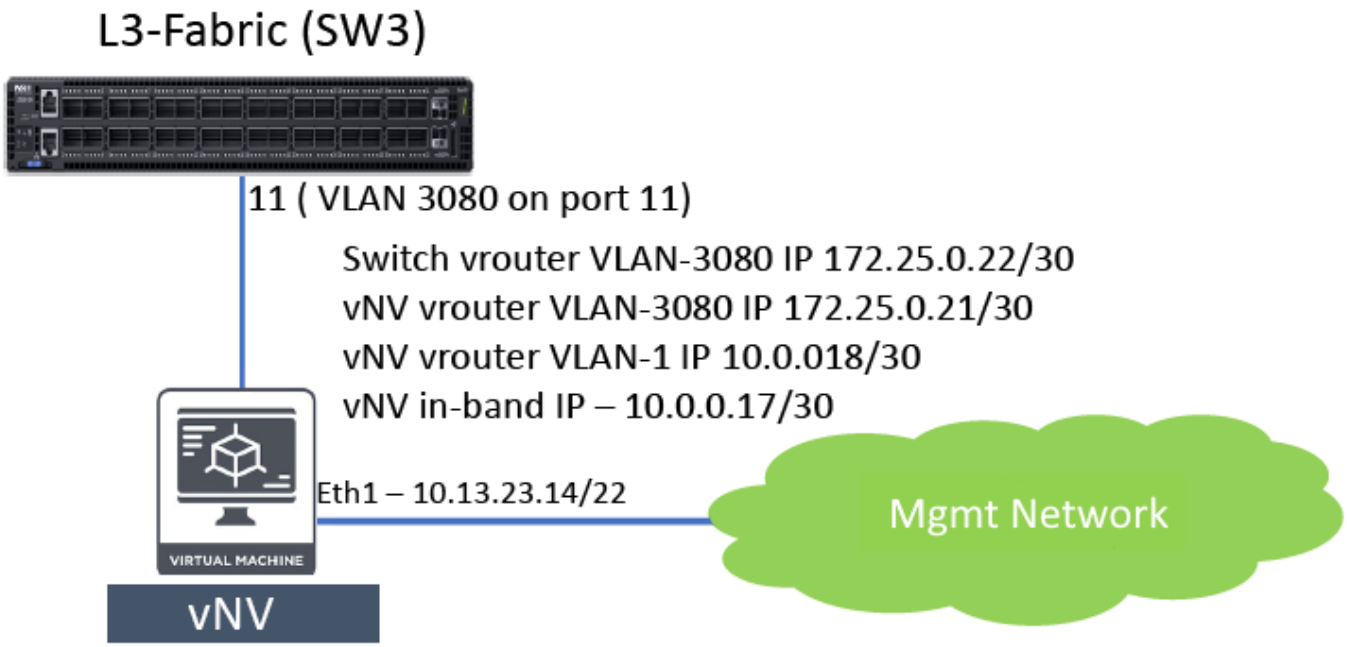
```
CLI (network-admin@vNV-UNUM) > fabric-join switch-ip 10.0.0.17 Joined fabric
test_case. Restarting nvOS...
Connected to Switch vNV-UNUM; nvOS Identifier:0x4db3f218; Ver: 5.2.0- 5020015650
CLI (network-admin@vNV-UNUM) >
CLI (network-admin@vNV-UNUM) > fabric-node-show format name,fab-name,mgmt- ip,in-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




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.

## Virtual Netvisor Deployment Use Case (Cont'd)

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 Pluribus 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 | |

# About Pluribus Networks

## About Pluribus Networks

Pluribus Networks delivers an open, controllerless software-defined network fabric for modern data centers, multi-site data centers and distributed cloud edge environments.

The Linux-based Netvisor® ONE operating system and the Unified Cloud Fabric™ have been purpose-built to deliver radically simplified networking and comprehensive visibility along with white box economics by leveraging hardware from our partners Dell EMC, Edgecore, Celestica and Champion ONE, as well as Pluribus' own Freedom™ Series of switches.

The Unified Cloud Fabric provides a fully automated underlay and virtualized overlay with comprehensive visibility and brownfield interoperability and is optimized to deliver rich and highly secure per-tenant services across data center sites with simple operations having no single point of failure.

Further simplifying network operations is Pluribus UNUM™, an agile, multi-functional web management portal that provides a rich graphical user interface to manage the Unified Cloud Fabric. UNUM has two key modules - UNUM Fabric Manager for provisioning and management of the fabric and UNUM Insight Analytics to quickly examine billions of flows traversing the fabric to ensure quality and performance.

Pluribus is deployed in more than 275 customers worldwide, including the 4G and 5G mobile cores of more than 75 Tier 1 service providers delivering mission-critical traffic across the data center for hundreds of millions of connected devices. Pluribus is networking, simplified.

For additional information contact Pluribus Networks at info@pluribusnetworks.com, or visit www.pluribusnetworks.com

Follow us on Twitter @pluribusnet. or on LinkedIn at https://www.linkedin.com/company/pluribus-networks/
.

**Corporate Headquarters**
Pluribus Networks, Inc.
5201 Great America Parkway, Suite 422
Santa Clara, CA 95054

1-855-438-8638 / +1-650-289-4717

**India Office**
Pluribus Networks India Private Limited
Indiqube Brigade Square, 4th Floor
21, Cambridge Road
Bangalore 560008

Document Version - July 2022