

AHV Networking

Nutanix Best Practices

Version 1.3 • December 2018 • BP-2071

Copyright

Copyright 2018 Nutanix, Inc.

Nutanix, Inc.

1740 Technology Drive, Suite 150

San Jose, CA 95110

All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws.

Nutanix is a trademark of Nutanix, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Contents

- 1. Executive Summary..... 5**
- 2. Introduction..... 6**
 - 2.1. Audience..... 6
 - 2.2. Purpose..... 6
- 3. Nutanix Enterprise Cloud Overview..... 7**
 - 3.1. Nutanix Acropolis Architecture..... 8
- 4. AHV Networking Overview..... 9**
 - 4.1. Open vSwitch..... 9
 - 4.2. Bridges..... 9
 - 4.3. Ports..... 9
 - 4.4. Bonds..... 10
 - 4.5. Virtual Local Area Networks (VLANs)..... 11
 - 4.6. IP Address Management (IPAM)..... 12
 - 4.7. View Network Status..... 13
- 5. AHV Networking Best Practices..... 18**
 - 5.1. Configuring Without Connectivity: Staging and Recovery..... 18
 - 5.2. Open vSwitch Bridge and Bond Recommendations..... 19
 - 5.3. Load Balancing within Bond Interfaces..... 26
 - 5.4. VLANs for AHV Hosts and CVMs..... 33
 - 5.5. CVM Network Segmentation..... 36
 - 5.6. Jumbo Frames..... 37
 - 5.7. Virtual NIC Trunking..... 38
- 6. Conclusion..... 39**
- Appendix..... 40**
 - AHV Networking Terminology..... 40

AHV Networking Best Practices Checklist.....	40
AHV Command Line Tutorial.....	43
AHV Networking Command Examples.....	46
References.....	48
About the Author.....	48
About Nutanix.....	48
List of Figures.....	49
List of Tables.....	50

1. Executive Summary

The default networking that we describe in the [AHV Best Practices Guide](#) covers a wide range of scenarios that Nutanix administrators encounter. However, for those situations with unique VM and host networking requirements that are not covered elsewhere, use this advanced networking guide.

The default AHV networking configuration provides a highly available network for guest VMs and the Nutanix Controller VM (CVM). This structure includes simple control and segmentation of guest VM traffic using VLANs, as well as IP address management. The network visualization for AHV available in Prism also provides a view of the guest and host network configuration for troubleshooting and verification.

This advanced guide is useful when the defaults don't match customer requirements. Configuration options include host networking high availability and load balancing mechanisms beyond the default active-backup, tagged VLAN segmentation for host and CVM traffic, and detailed command line configuration techniques for situations where a GUI may not be sufficient. The tools we present here enable you to configure AHV to meet the most demanding network requirements.

2. Introduction

2.1. Audience

This best practices guide is part of the Nutanix Solutions Library. It is intended for AHV administrators configuring advanced host and VM networking. Readers of this document should already be familiar with the [AHV Best Practices Guide](#), which covers basic networking.

2.2. Purpose

In this document, we cover the following topics:

- Command line overview and tips.
- Open vSwitch in AHV.
- VLANs for hosts, CVMs, and guest VMs.
- IP address management (IPAM).
- Network adapter teaming within bonds.
- Network adapter load balancing.

Table 1: Document Version History

Version Number	Published	Notes
1.0	February 2017	Original publication.
1.1	February 2018	Added jumbo frame configuration, bond name recommendation, and considerations for staging installation with flat switch.
1.2	October 2018	Updated product naming and recommendations regarding the balance-slb bond mode.
1.3	December 2018	Updated the Open vSwitch Bridge and Bond Recommendations section.

3. Nutanix Enterprise Cloud Overview

Nutanix delivers a web-scale, hyperconverged infrastructure solution purpose-built for virtualization and cloud environments. This solution brings the scale, **resilience**, and economic benefits of web-scale architecture to the enterprise through the Nutanix Enterprise Cloud Platform, which combines three product families—Nutanix Acropolis, Nutanix Prism, and Nutanix Calm.

Attributes of this Enterprise Cloud OS include:

- Optimized for storage and compute resources.
- Machine learning to plan for and adapt to changing conditions automatically.
- Self-healing to tolerate and adjust to component failures.
- API-based automation and rich analytics.
- Simplified one-click upgrade.
- Native file services for user and application data.
- Native backup and disaster recovery solutions.
- Powerful and feature-rich virtualization.
- Flexible software-defined networking for visualization, automation, and security.
- Cloud automation and life cycle management.

Nutanix Acropolis provides data services and can be broken down into three foundational components: the Distributed Storage Fabric (DSF), the App Mobility Fabric (AMF), and AHV. Prism furnishes one-click infrastructure management for virtual environments running on Acropolis. Acropolis is hypervisor agnostic, supporting three third-party hypervisors—ESXi, Hyper-V, and XenServer—in addition to the native Nutanix hypervisor, AHV.

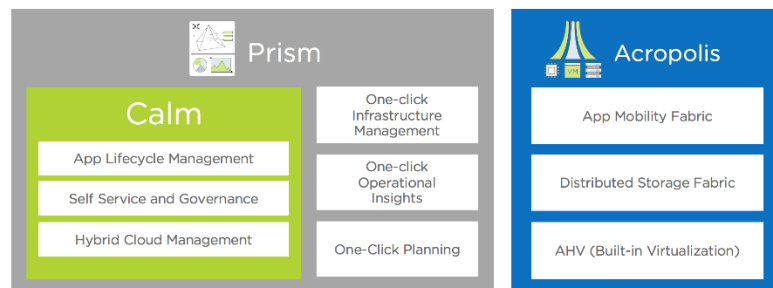


Figure 1: Nutanix Enterprise Cloud

3.1. Nutanix Acropolis Architecture

Acropolis does not rely on traditional SAN or NAS storage or expensive storage network interconnects. It combines highly dense storage and server compute (CPU and RAM) into a single platform building block. Each building block delivers a unified, scale-out, shared-nothing architecture with no single points of failure.

The Nutanix solution requires no SAN constructs, such as LUNs, RAID groups, or expensive storage switches. All storage management is VM-centric, and I/O is optimized at the VM virtual disk level. The software solution runs on nodes from a variety of manufacturers that are either all-flash for optimal performance, or a hybrid combination of SSD and HDD that provides a combination of performance and additional capacity. The DSF automatically tiers data across the cluster to different classes of storage devices using intelligent data placement algorithms. For best performance, algorithms make sure the most frequently used data is available in memory or in flash on the node local to the VM.

To learn more about the Nutanix Enterprise Cloud, please visit [the Nutanix Bible](#) and [Nutanix.com](#).

4. AHV Networking Overview

AHV uses Open vSwitch (OVS) to connect the CVM, the hypervisor, and guest VMs to each other and to the physical network on each node.

4.1. Open vSwitch

Open vSwitch (OVS) is an open source software switch implemented in the Linux kernel and designed to work in a multiserver virtualization environment. By default, OVS behaves like a layer-2 learning switch that maintains a MAC address table. The hypervisor host and VMs connect to virtual ports on the switch.

OVS supports many popular switch features, such as VLAN tagging, load balancing, and link aggregation control protocol (LACP). Each AHV server maintains an OVS instance, and all OVS instances combine to form a single logical switch. Constructs called bridges manage the switch instances residing on the AHV hosts.

4.2. Bridges

Bridges act as virtual switches to manage traffic between physical and virtual network interfaces. The default AHV configuration includes an OVS bridge called br0 and a native Linux bridge called virbr0. The virbr0 Linux bridge carries management and storage communication between the CVM and AHV host. All other storage, host, and VM network traffic flows through the br0 OVS bridge. The AHV host, VMs, and physical interfaces use "ports" for connectivity to the bridge.

4.3. Ports

Ports are logical constructs created in a bridge that represent connectivity to the virtual switch. Nutanix uses several port types, including internal, tap, VXLAN, and bond.

- An internal port—with the same name as the default bridge (br0)—provides access for the AHV host.
- Tap ports connect virtual NICs presented to VMs.
- VXLAN ports are used for the IP address management functionality provided by Acropolis.
- Bonded ports provide NIC teaming for the physical interfaces of the AHV host.

4.4. Bonds

Bonded ports aggregate the physical interfaces on the AHV host. By default, the system creates a bond named br0-up in bridge br0 containing all physical interfaces. Changes to the default bond (br0-up) using `manage_ovs` commands can rename it to `bond0` when using older examples, so keep in mind that your system may be named differently than the diagram below. Nutanix recommends using the name `br0-up` to quickly identify this interface as the bridge `br0` uplink. Using this naming scheme, you can also easily distinguish uplinks for additional bridges from each other.

OVS bonds allow for several load-balancing modes, including `active-backup`, `balance-slb`, and `balance-tcp`. Administrators can also activate LACP for a bond to negotiate link aggregation with a physical switch. Because the `bond_mode` setting is not specified during installation, it defaults to `active-backup`, which is the configuration we recommend for ease of use.

The following diagram illustrates the networking configuration of a single host immediately after imaging. The best practice is to use only the 10 Gb NICs and to disconnect the 1 Gb NICs if you do not need them. For additional information on bonds, please refer to the Best Practices section below.



Note: Only utilize NICs of the same speed within the same bond.

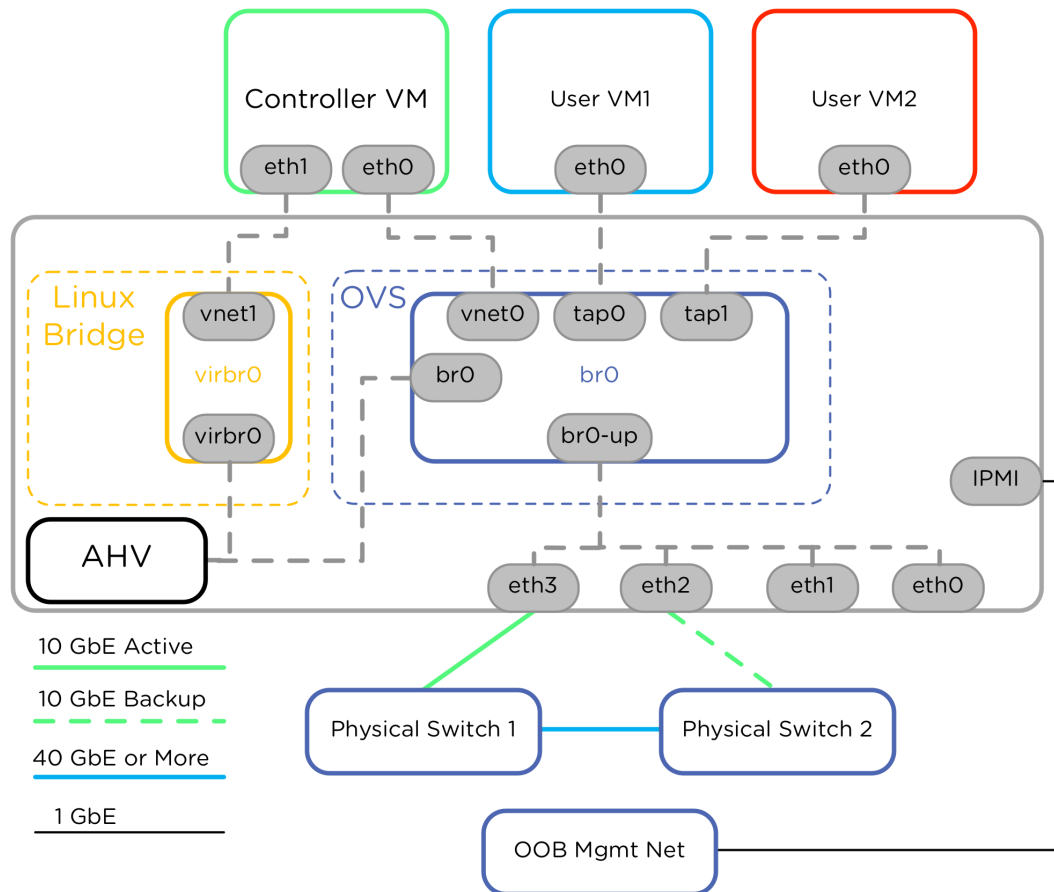


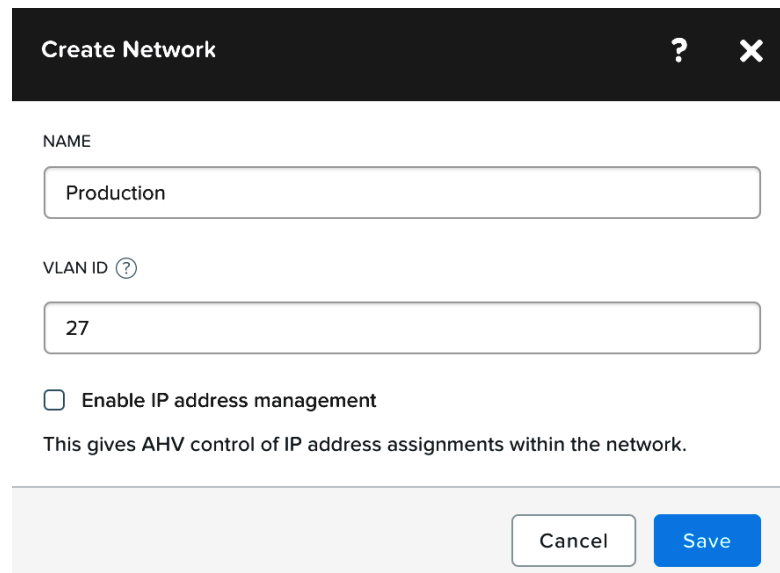
Figure 2: Post-Imaging Network State

Connections from the server to the physical switch use 10 GbE networking. You can establish connections between the switches with 40 GbE or faster direct links, or through a leaf-spine network topology (not shown). The IPMI management interface of the Nutanix node also connects to the out-of-band management network, which may connect to the production network. Each node always has a single connection to the management network, but we have omitted this element from further images in this document for clarity and simplicity.

4.5. Virtual Local Area Networks (VLANs)

AHV supports the use of VLANs for the CVM, AHV host, and user VMs. We discuss the steps for assigning VLANs to the AHV host and CVM in the Best Practices section below. You can easily create and manage a virtual NIC's networks for user VMs in the Prism GUI, the Acropolis CLI (aCLI), or using REST without any additional AHV host configuration.

Each virtual network in AHV maps to a single VLAN and bridge. All VLANs allowed on the physical switch port to the AHV host are available for VM network creation. In the following figure, we're using Prism to assign a friendly network name of Production and VLAN ID 27 for a user VM network on the default bridge, br0.



Create Network ? X

NAME

Production

VLAN ID ?

27

Enable IP address management

This gives AHV control of IP address assignments within the network.

Cancel Save

Figure 3: Prism UI Network Creation

By default, all virtual NICs are created in “access” mode, which permits only one VLAN per virtual network. However, you can choose to configure a virtual NIC in “trunked” mode using the aCLI instead, allowing multiple VLANs on a single VM NIC for network-aware user VMs. For more information on virtual NIC modes, please refer to the Best Practices section below.

4.6. IP Address Management (IPAM)

In addition to network creation and VLAN management, AHV also supports IP address management (IPAM), as shown in the figure below. IPAM allows AHV to assign IP addresses automatically to VMs using DHCP. Administrators can configure each virtual network with a specific IP subnet, associated domain settings, and group of IP address pools available for assignment.

Create Network
?
✕

Enable IP address management

This gives AHV control of IP address assignments within the network.

NETWORK IP ADDRESS / PREFIX LENGTH

192.168.101.0/24

GATEWAY IP ADDRESS

192.168.1101.1

Configure Domain Settings

IP ADDRESS POOLS ?

+ Create Pool

START ADDRESS	END ADDRESS	
192.168.101.20	192.168.101.200	✎ ✕

Cancel

Save

Figure 4: IPAM

Administrators can use AHV with IPAM to deliver a complete virtualization deployment, including network management, from the unified Prism interface. To avoid address overlap, be sure to work with your network team to reserve a range of addresses for VMs before enabling the IPAM feature.

AHV assigns an IP address from the address pool when creating a managed VM NIC; the address releases back to the pool when the VM NIC or VM is deleted. With a managed network, AHV intercepts DHCP requests and bypasses traditional network-based DHCP servers. AHV uses the last network IP address for the DHCP server unless you select **Override DHCP server** when creating the network.

4.7. View Network Status

The following sections illustrate the most common methods used to view network configuration for VMs and AHV hosts. Some information is visible in both Prism and the CLI, and we show

both outputs when available. Refer to the AHV Command Line section in the appendix for more information on CLI usage.

Viewing Network Configuration for VMs in Prism

Select **Network Configuration**, then **User VM Interfaces** to view VM virtual networks from the VM page, as shown in the figure below.

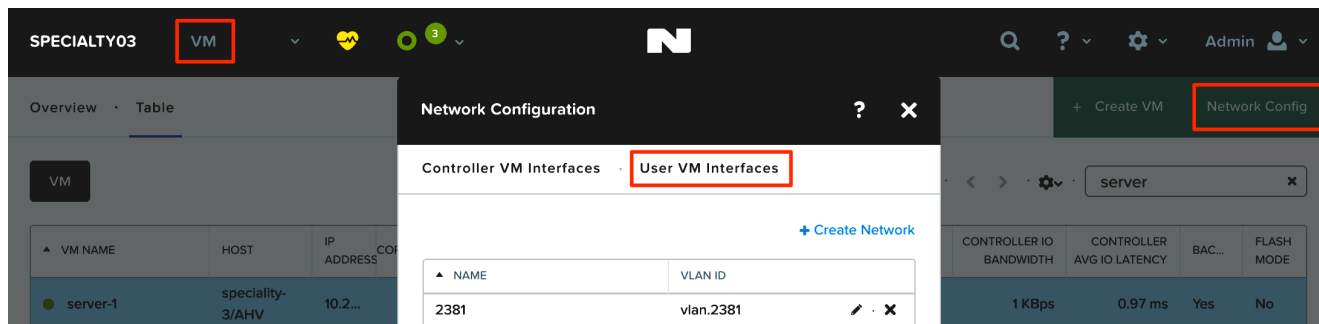


Figure 5: Prism UI Network List

You can see individual VM network details under the Table view on the VM page by selecting the desired VM and choosing **Update**, as shown in the figure below.

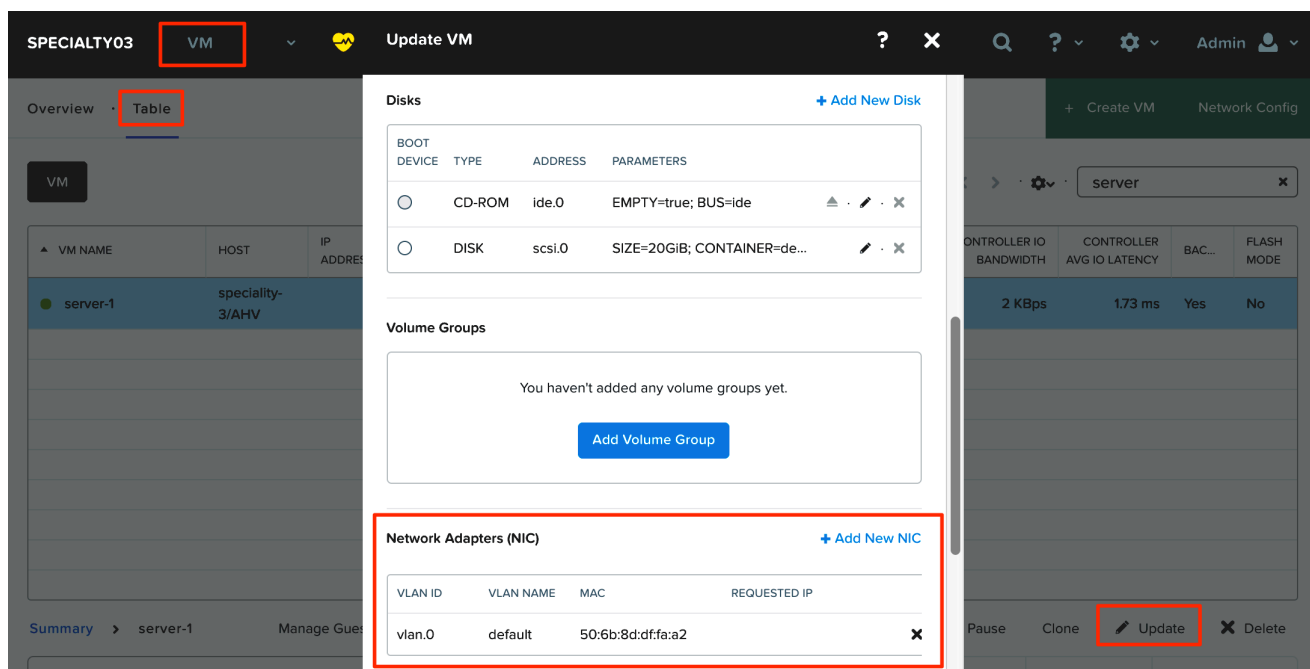


Figure 6: Prism UI VM Network Details

Viewing AHV Host Network Configuration in Prism

Select the Network page to view VM- and host-specific networking details. When you select a specific AHV host, Prism displays the network configuration, as shown in the figure below. For more information on the new Network Visualization feature, refer to the [Prism Web Console Guide](#).

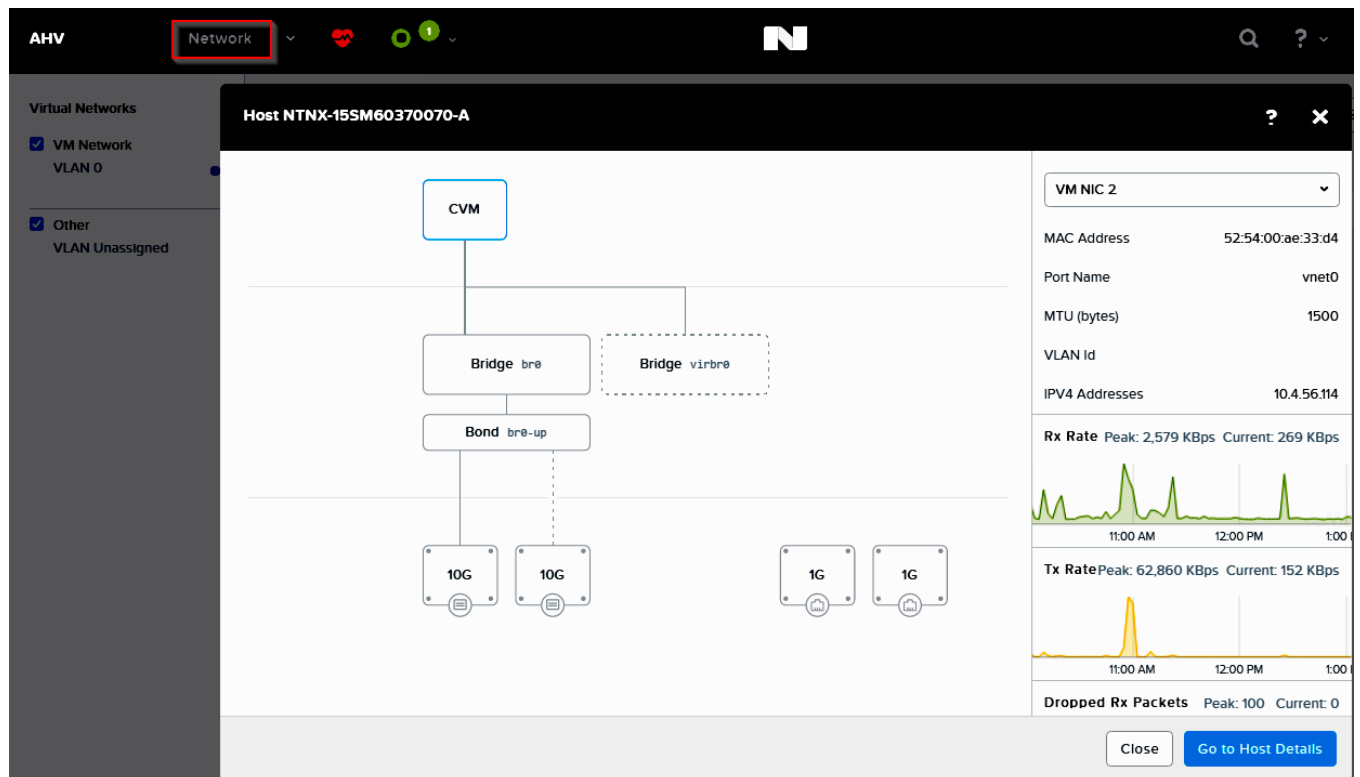


Figure 7: AHV Host Network Visualization

View AHV Host Network Configuration in the CLI

You can view Nutanix AHV network configuration in detail using aCLI, native AHV, and OVS commands as shown in the appendix. The following sections outline basic administration tasks and the commands needed to review and validate a configuration.

Administrators can perform all management operations through the Prism web interface and APIs or through SSH access to the Controller VM.



Tip: For better security and a single point of management, avoid connecting directly to the AHV hosts. All AHV host operations can be performed from the CVM by connecting to 192.168.5.1, the internal management address of the AHV host.

View Physical NIC Status from the CVM

```
nutanix@CVM$ manage_ovs --bridge_name br0 show_uplinks
Uplink ports: br0-up
Uplink ifaces: eth3 eth2
nutanix@CVM$ manage_ovs show_interfaces
name  mode link speed
eth0  1000 True  1000
eth1  1000 True  1000
eth2  10000 True 10000
eth3  10000 True 10000
```

View OVS Bond Status from the CVM

Use the special 192.168.5.1 address from the CVM to connect to the AHV host and view bond details. Replace br0-up with the desired bond name, or omit the name to show all bonds.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up"
---- br0-up ----
bond_mode: active-backup
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: off
slave eth2: enabled
    may_enable: true
slave eth3: enabled
    active slave
    may_enable: true
```

View OVS Bridge Configuration from the CVM

List all bridge details in the AHV host, or include a specific bridge name to view an individual bridge.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl show"
```


View VM Network Configuration from a CVM Using the aCLI

Connect to any CVM in the Nutanix cluster to launch the aCLI and view cluster-wide VM network details.

```
nutanix@CVM$ acli
<acropolis> net.list
Network name      Network UUID                               Type  Identifier
Production        ea8468ec-c1ca-4220-bc51-714483c6a266      VLAN  27
vlan.0            a1850d8a-a4e0-4dc9-b247-1849ec97b1ba      VLAN  0
<acropolis> net.list_vms vlan.0
VM UUID                               VM name   MAC address
7956152a-ce08-468f-89a7-e377040d5310  VM1       52:54:00:db:2d:11
47c3a7a2-a7be-43e4-8ebf-c52c3b26c738  VM2       52:54:00:be:ad:bc
501188a6-faa7-4be0-9735-0e38a419a115  VM3       52:54:00:0c:15:35
```

5. AHV Networking Best Practices

The main best practice for AHV networking is to keep things simple. The recommended networking configuration, with two 10 Gb adapters using active-backup, provides a highly available environment that performs well. Nutanix CVMs and AHV hosts communicate in the untagged VLAN, and tagged VLANs serve guest VM traffic. Use this basic configuration unless there is a compelling business requirement for advanced configuration.

5.1. Configuring Without Connectivity: Staging and Recovery

All instructions in this document assume that the Nutanix nodes are connected to end-state, production network infrastructure. This network infrastructure must be preconfigured with the correct VLANs and port settings for load balancing. However, sometimes you aren't on a production network—for example, when configuring a cluster at a staging site before delivery, or when using a standalone switch to configure a cluster for the first time before connecting it to the production network. You can also find yourself disconnected from the production network when the connected switches are not configured as expected, at which point you must be able to recover.

When performing any of the following commands in a staging environment, with a staging switch, or during recovery, we recommend following the steps below:

- Perform a **cluster stop** operation from any CVM to ensure that the cluster is stopped. Many of the commands in this document can break cluster connectivity unless the upstream switch is properly configured as well. Stopping the cluster prevents any problems related to unexpected connectivity loss on multiple nodes.

```
nutanix@CVM$ cluster stop
```

- When altering the configuration during staging or recovery, connect to the IPMI management console to run the commands that follow. This connection ensures that even if the upstream switch is not configured (or is misconfigured), you can still apply the desired configuration to the hosts and CVMs.

Use the IPMI management console's remote KVM to get access to the AHV root account. From the AHV root shell, connect to the CVM using the dedicated 192.168.5.254 address.

```
root@AHV# ssh nutanix@192.168.5.254
```

- When the Nutanix nodes are finally connected to the production network, ensure connectivity on all configured hosts and CVMs before performing **cluster start**. Pinging the default gateway with `allssh` and `hostssh` is a useful shortcut to verify connectivity.

```
nutanix@CVM$ allssh "ping -c 2 <default gateway IP>"
nutanix@CVM$ hostssh "ping -c 2 <default gateway IP>"
nutanix@CVM$ cluster start
```

5.2. Open vSwitch Bridge and Bond Recommendations

This section addresses advanced bridge and bond configuration scenarios using a combination of CLI commands that we describe in more detail in the AHV Command Line section in the appendix. Identify the scenario that best matches the desired use case and follow those instructions.


 **Tip:** Nutanix recommends that each bond have at least two interfaces.

Table 2: Bridge and Bond Use Cases

Bridge and Bond Scenario	Use Case
2x 10 Gb (no 1 Gb)	Use when you can send all CVM and user VM traffic over the same pair of 10 Gb adapters. Compatible with any load balancing algorithm. Recommended configuration for easy setup.
2x 10 Gb and 2x 1 Gb separated	Use when you need an additional, separate pair of physical 1 Gb adapters for VM traffic that must be isolated to another adapter or physical switch. Keep the CVM traffic on the 10 Gb network. You can either allow user VM traffic on the 10 Gb network or force it onto the 1 Gb network. Compatible with any load balancing algorithm.
4x 10 Gb (2 + 2) and 2x 1 Gb separated	Use to physically separate CVM traffic such as storage and Nutanix Volumes from user VM traffic while still providing 10 Gb connectivity for both traffic types. The four 10 Gb adapters are divided into two separate pairs. Compatible with any load balancing algorithm.

Bridge and Bond Scenario	Use Case
4x 10 Gb combined and 2x 1 Gb separated	Use to provide additional bandwidth and failover capacity to the CVM and user VMs sharing four 10 Gb adapters in the same bond. We recommend using LACP with balance-tcp to take advantage of all adapters. This case is not illustrated in the diagrams below.

Administrators can view and change OVS configuration from the CVM command line with the **manage_ovs** command. To execute a single command on every Nutanix CVM in a cluster, use the **allssh** shortcut described in the AHV Command Line appendix.



Note: The order in which flags and actions pass to **manage_ovs** is critical. Flags must come first. Any flag passed after an action is not parsed.

```
nutanix@CVM$ manage_ovs --help
USAGE: manage_ovs [flags] <action>
```

To list all physical interfaces on all nodes, use the **show_interfaces** action. The **show_uplinks** action returns the details of a bonded adapter for a single bridge.



Note: If you do not enter a **bridge_name**, the action runs on the default bridge, br0.

```
nutanix@CVM$ allssh "manage_ovs show_interfaces"
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> show_uplinks"
```

The **update_uplinks** action takes a comma-separated list of interfaces and configures these into a single uplink bond in the specified bridge. If the bond does not have at least one member physically connected, the **manage_ovs** command produces a warning and exits without configuring the bond. To avoid this error and provision members of the bond even if they are not connected, use the **require_link=false** flag.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> --interfaces <interfaces>
update_uplinks"
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> --interfaces <interfaces> --
require_link=false update_uplinks"
```

5.2. Scenario 1: 2x 10 Gb

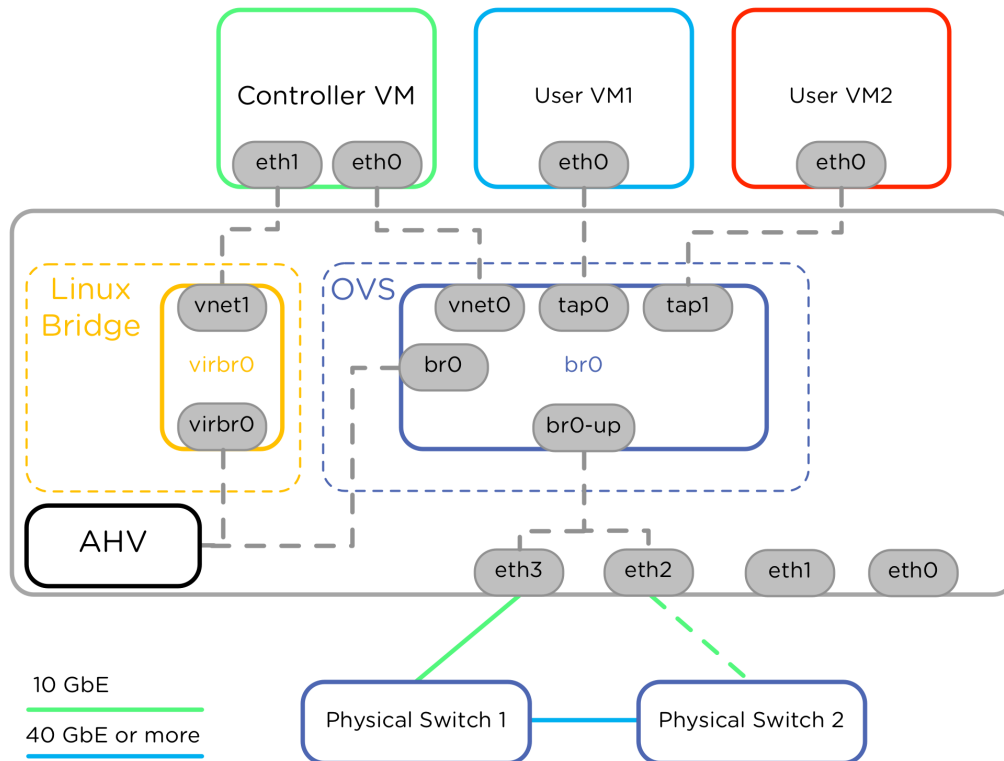


Figure 8: Network Connections for 2x 10 Gb NICs

The most common network configuration is to utilize the 10 Gb interfaces within the default bond for all networking traffic. The CVM and all user VMs use the 10 Gb interfaces. In this configuration, we don't use the 1 Gb interfaces. Note that this is different from the factory configuration, because we have removed the 1 Gb interfaces from the OVS bond. For simplicity, we have not included the IPMI connection in these diagrams.

This scenario uses two physical upstream switches, and each 10 Gb interface within the bond plugs into a separate physical switch for high availability. Within the bond, only one physical interface is active when using the default active-backup load balancing mode. Nutanix recommends using active-backup because it is easy to configure, works immediately after install, and requires no upstream switch configuration. See the Load Balancing within Bond Interfaces section below for more information and alternate configurations.

Remove NICs that are not in use from the default bond, especially when they are of different speeds. To do so, perform the following `manage_ovs` action for each Nutanix node in the cluster, or use with the command `allssh`:

- From the CVM, remove eth0 and eth1 from the default bridge br0 on all CVMs by specifying that only eth2 and eth3 remain in the bridge. The **10g** shortcut lets you include all 10 Gb

interfaces without having to specify the interfaces explicitly by name. Shortcuts also exist for 25 Gb (**25g**) and 40 Gb (**40g**) interfaces. Some Nutanix models have different ethX names for 1 Gb and 10 Gb links, so this shortcut is helpful:

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 --bond_name br0-up --interfaces 10g update_uplinks"
```

Note: Previous versions of this guide used the bond name bond0 instead of br0-up. We recommend using br0-up because it identifies the associated bridge and the uplink function of this bond.

If you're performing the network configuration without all 10 Gb adapters plugged in, be sure to use the require_link=false flag:

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 --bond_name br0-up --interfaces 10g --require_link=false update_uplinks"
```

This setting is useful when you are performing the installation in a staging environment before connecting to production.

5.2. Scenario 2: 2x 10 Gb and 2x 1 Gb Separated

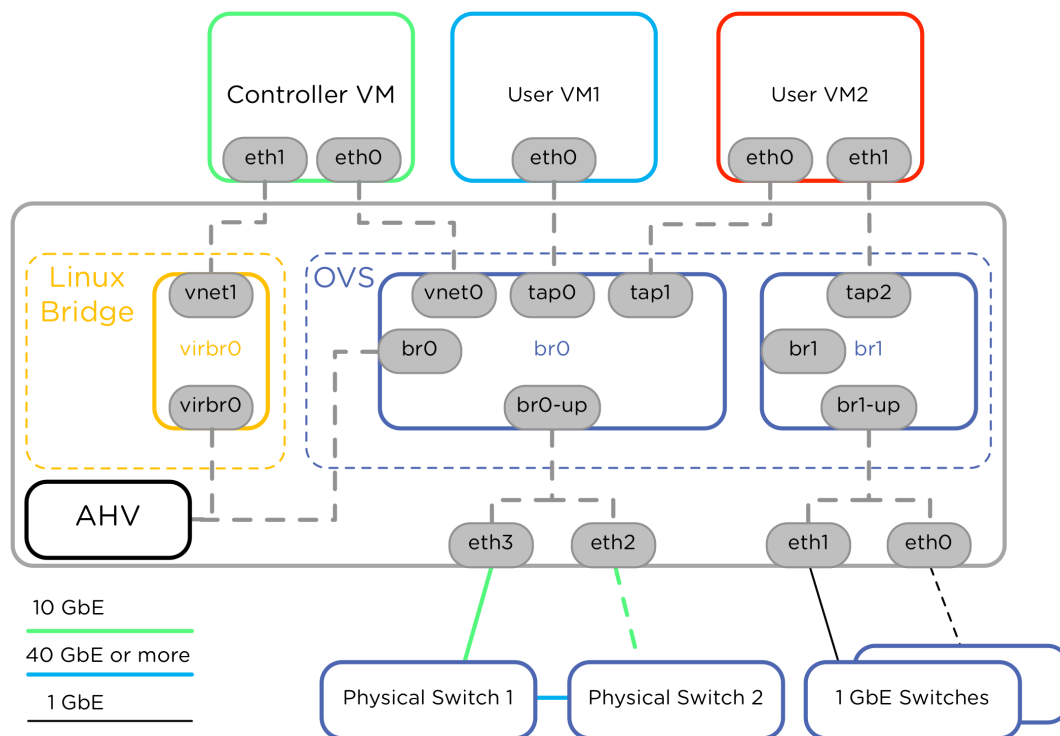


Figure 9: Network Connections for 2x 10 Gb and 2x 1 Gb NICs

If you need the 1 Gb physical interfaces, separate the 10 Gb and 1 Gb interfaces into different bridges and bonds to ensure that CVM and user VM traffic always traverses the fastest possible link.



Note: Links of different speeds cannot be mixed in the same bond.

Here, we've grouped the 10 Gb interfaces (eth2 and eth3) into br0-up and dedicated them to the CVM and User VM1. We've grouped the 1 Gb interfaces into br1-up; only a second link on User VM2 uses them. Bonds br0-up and br1-up are added into br0 and br1, respectively.

In this configuration, the CVM and user VMs use the 10 Gb interfaces. Bridge br1 is available for VMs that require physical network separation from the CVM and VMs on br0. Devices eth0 and eth1 could alternatively plug into a different pair of upstream switches for further physical traffic separation as shown.

Perform the following actions for each Nutanix node in the cluster to achieve the configuration shown above:

- In AOS 5.5 or later, **manage_ovs** handles bridge creation. On each CVM, add bridge br1. You can use **allssh** to execute a command on all CVMs. Bridge names must not exceed six characters. We suggest using the name br1.



Note: When adding a bridge, ensure that the bridge is created on every host in the cluster. Failure to add bridges to all hosts can lead to VM migration errors.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 create_single_bridge"
```

- With AOS versions prior to 5.5, use **ovs-vsctl** on the AHV host to add the bridge. Use **hostssh** to execute the command on all Nutanix AHV hosts in the cluster.



Note: In AOS version 5.5 or later, always use `manage_ovs` to add bridges. When using `ovs-vsctl add-br`, you must restart the Acropolis master process for the change to take effect. However, when you use `manage_ovs` to add bridges, the restart is not required. Once all bridges are added, see the restart commands at the end of this section.

```
nutanix@CVM$ hostssh "ovs-vsctl add-br br1"
```

- From the CVM, remove eth0 and eth1 from the default bridge br0 on all CVMs, as described in the first scenario. You can use the `allssh` shortcut here, but first run the appropriate `show` commands to make sure that all interfaces are in a good state before executing the update.

```
nutanix@CVM$ allssh "manage_ovs show_interfaces"
```

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 show_uplinks"
```

The output from the show commands above should clarify that the 10 Gb interfaces have connectivity to the upstream switches—just look for the columns labeled link and speed. Once you've confirmed connectivity, update the bond to include only 10 Gb interfaces.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 --bond_name br0-up --interfaces 10g
update_uplinks"
```

- Add the eth0 and eth1 uplinks to br1 in the CVM using the **1g** interface shortcut.



Note: Use the `--require_link=false` flag to create the bond even if not all 1g adapters are connected.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 --bond_name br1-up --interfaces 1g --
require_link=false update_uplinks"
```

- Now that a bridge, br1, exists just for the 1 Gb interfaces, you can create networks for "User VM2" with the following global aCLI command. Putting the bridge name in the network name is helpful when viewing the network in the Prism GUI. In this example, Prism shows a network named "br1_vlan99" to indicate that this network sends VM traffic over VLAN 99 on bridge br1.

```
nutanix@CVM$ acli net.create br1_vlan99 vswitch_name=br1 vlan=99
```

If **ovs-vsctl** was used to add bridges, restart the Acropolis master process to detect the new bridges. Once the new bridges have been added to all nodes in the cluster, find the IP address of the Acropolis master by running the following on any CVM.

```
nutanix@CVM$ links http:0:2030 | grep Master
Acropolis Master: this node
```

If an address other than this node is returned, connect to that address with SSH as the Nutanix user and run the following commands. If this node was returned, just run the following commands to stop and start the process.

```
nutanix@CVM$ genesis stop acropolis
nutanix@CVM$ cluster start
```


5.2. Scenario 3: 4x 10 Gb and 2x 1 Gb Separated

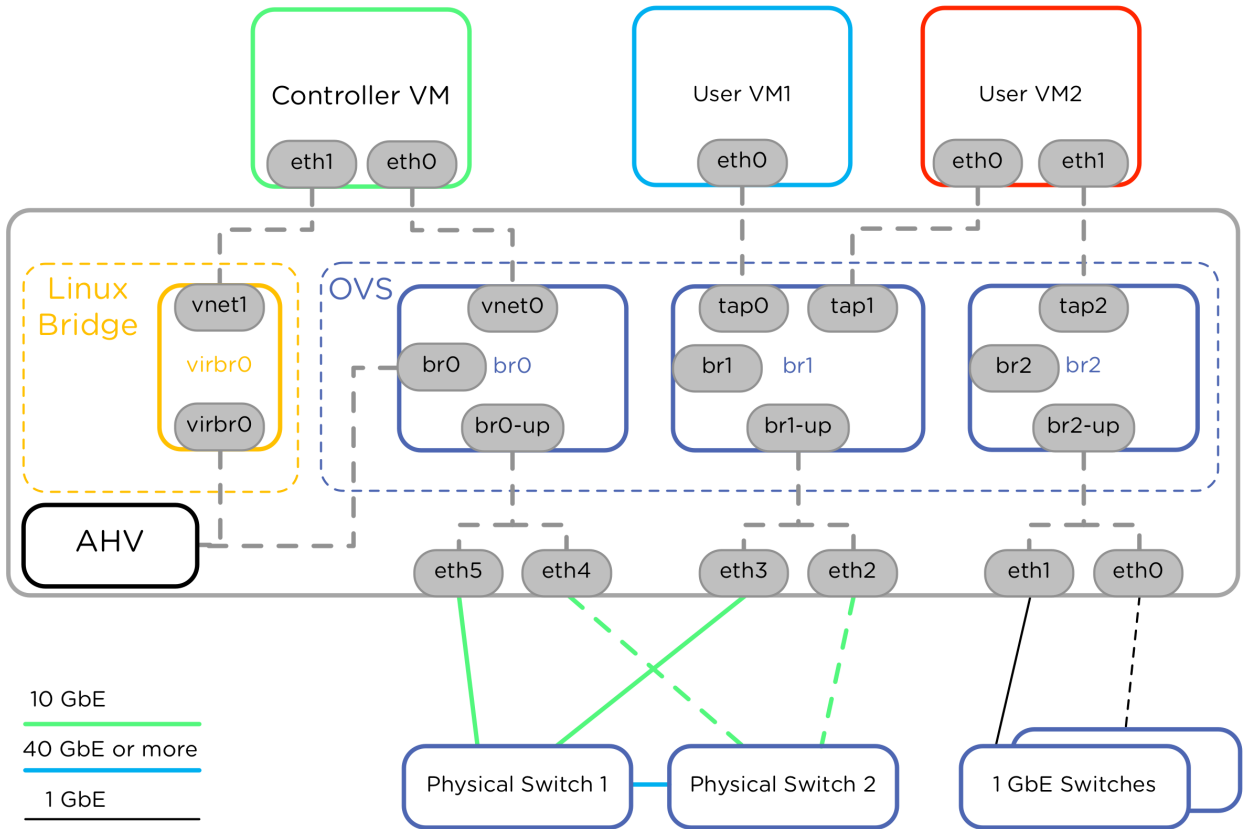


Figure 10: Network Connections for 4x 10 Gb and 2x 1 Gb NICs

Nutanix servers are also available with four 10 Gb adapters. In this configuration Nutanix recommends dedicating two 10 Gb adapters to CVM traffic and two 10 Gb adapters for user VM traffic, thus separating user and CVM network traffic. Alternatively, for increased data throughput, you can combine all 10 Gb adapters into a single bridge and bond with advanced load balancing. Split the 1 Gb adapters into a dedicated bond for user VMs if required; otherwise, these interfaces can remain unused. In the diagram above we show the 10 GbE interfaces connected to the same pair of switches, but you could use separate switches for all three bonds to provide further traffic segmentation.

To achieve this configuration, perform the following actions for each Nutanix node in the cluster:

- On each AHV host, add bridge br1 and br2 from the CVM. Bridge names must not exceed six characters. Use allssh to execute the command against all CVMs in the cluster. Refer to the previous scenario for the commands to create bridges in AOS versions prior to 5.5.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 create_single_bridge"
nutanix@CVM$ allssh "manage_ovs --bridge_name br2 create_single_bridge"
```

- On the CVM, remove eth0, eth1, eth2, and eth3 from the default bridge br0 on all CVMs by specifying that only eth4 and eth5 remain in the bridge. We can't use the 10 Gb interface shortcut here because only two of the four 10 Gb adapters are needed in the bond. If all Nutanix nodes in the cluster share identical network configuration, use the allssh shortcut. Again, confirm the network link using show commands before executing these changes.

```
nutanix@CVM$ allssh "manage_ovs show_interfaces"
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 show_uplinks"
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 --bond_name br0-up --interfaces eth4,eth5
update_uplinks"
```

- Add the eth2 and eth3 uplinks to br1 in the CVM and eth0 and eth1 to br2.



Note: Use the `--require_link=false` flag to create the bond even if not all adapters are connected.

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 --bond_name br1-up --interfaces eth2,eth3
update_uplinks"
nutanix@CVM$ allssh "manage_ovs --bridge_name br2 --bond_name br2-up --interfaces eth0,eth1
update_uplinks"
```

You can now create networks in these new bridges using the same syntax as in the previous scenarios in the aCLI. Once you've created the networks in the aCLI, you can view them in the Prism GUI, so it's helpful to include the bridge name in the network name.

```
nutanix@CVM$ acli net.create <net_name> vswitch_name=<br_name> vlan=<vlan_num>
```

For example:

```
nutanix@CVM$ acli net.create br1_production vswitch_name=br1 vlan=1001
nutanix@CVM$ acli net.create br2_production vswitch_name=br2 vlan=2001
```

5.3. Load Balancing within Bond Interfaces

OVS connects the AHV host to the network via bond interfaces. Each of these bonds contains multiple physical interfaces that can connect to one or more physical switches. To build a fault-tolerant network connection between the AHV host and the rest of the network, connect the physical interfaces in a bond to separate physical switches, as shown in the diagrams above.

A bond distributes its traffic between multiple physical interfaces according to the bond mode.

Table 3: Load Balancing Use Cases

Bond Mode	Use Case	Maximum VM NIC Throughput*	Maximum Host Throughput*
active-backup	Recommended. Default configuration, which transmits all traffic over a single active adapter.	10 Gb	10 Gb
balance-slb	Has caveats for multicast traffic. Increases host bandwidth utilization beyond a single 10 Gb adapter. Places each VM NIC on a single adapter at a time. Do not use with link aggregation such as LACP.	10 Gb	20 Gb
LACP and balance-tcp	Link aggregation required. Increases host and VM bandwidth utilization beyond a single 10 Gb adapter by balancing each VM NIC TCP session on a different adapter. Also used when network switches require LACP negotiation.	20 Gb	20 Gb

* Assuming 2x 10 Gb adapters. Simplex speed.

Active-Backup

The recommended and default bond mode is active-backup, where one interface in the bond carries traffic and other interfaces in the bond are used only when the active link fails. Active-backup is the simplest bond mode, easily allowing connections to multiple upstream switches without any additional switch configuration. The downside is that traffic from all VMs uses only the single active link within the bond at one time. All backup links remain unused until the active link fails. In a system with dual 10 Gb adapters, the maximum throughput of all VMs running on a Nutanix node is limited to 10 Gbps, or the speed of a single link.

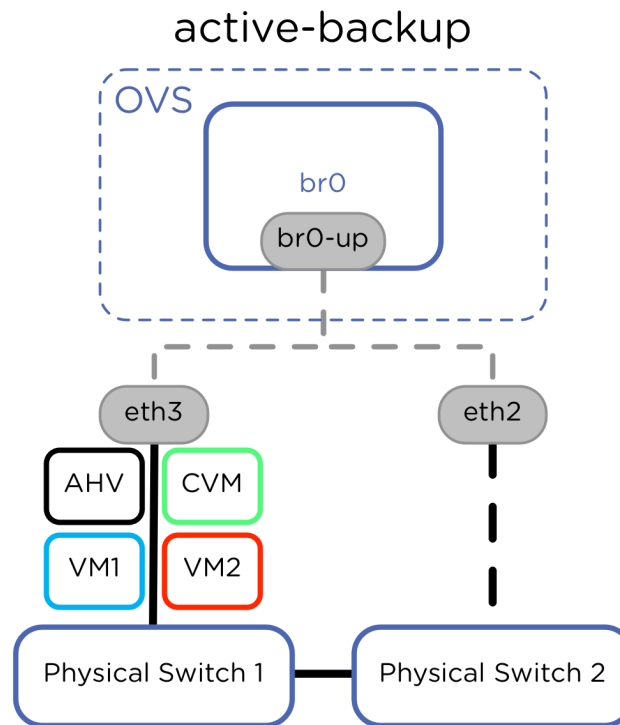


Figure 11: Active-Backup Fault Tolerance

Active-backup mode is enabled by default, but you can also configure it with the following AHV command:

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=active-backup"
```

View the bond mode and current active interface with the following AHV command:

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show"
```

In the active-backup configuration, this command's output would be similar to the following, where eth2 is shown as the active and eth3 is the backup interface. The value `may_enable: true` indicates that both eth2 and eth3 have connectivity and can be used as the active adapter:

```
---- br0-up ----
bond_mode: active-backup
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: off
slave eth2: enabled
    active slave
    may_enable: true
slave eth3: enabled
    may_enable: true
```

Balance-SLB

Nutanix does not recommend balance-slb due to the multicast traffic caveats noted below. To utilize the bandwidth of multiple links, consider using link aggregation with LACP and `balance-tcp` instead of `balance-slb`. The `balance-slb` bond mode in OVS takes advantage of all links in a bond and uses measured traffic load to rebalance VM traffic from highly used to less used interfaces. When the configurable `bond-rebalance` interval expires, OVS uses the measured load for each interface and the load for each source MAC hash to spread traffic evenly among links in the bond. Traffic from some source MAC hashes may move to a less active link to more evenly balance bond member utilization. Perfectly even balancing may not always be possible, depending on the number of source MAC hashes and their stream sizes.

Each individual VM NIC uses only a single bond member interface at a time, but a hashing algorithm distributes multiple VM NICs (multiple source MAC addresses) across bond member interfaces. As a result, it is possible for a Nutanix AHV node with two 10 Gb interfaces to use up to 20 Gbps of network throughput, while individual VMs have a maximum throughput of 10 Gbps, the speed of a single physical interface.

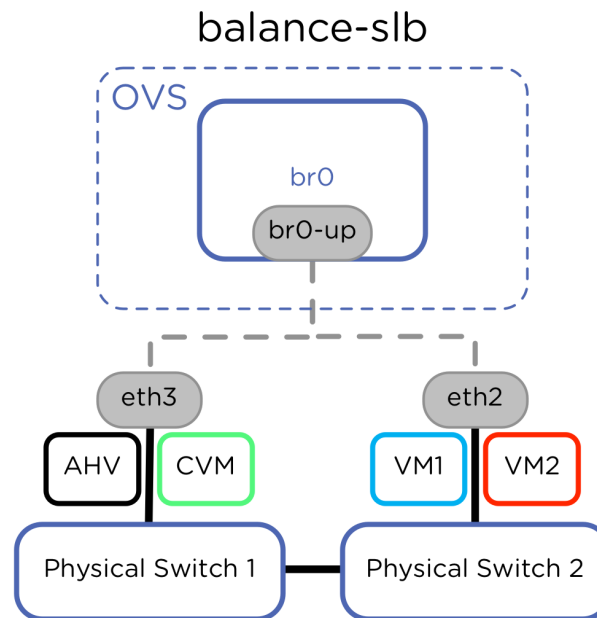


Figure 12: Balance-SLB Load Balancing

The default rebalance interval is 10 seconds, but Nutanix recommends setting this interval to 30 seconds to avoid excessive movement of source MAC address hashes between upstream switches. Nutanix has tested this configuration using two separate upstream switches with AHV. No additional configuration (such as link aggregation) is required on the switch side, as long as the upstream switches are interconnected physically or virtually and both uplinks allow the same VLANs.



Note: Do not use link aggregation technologies such as LACP with balance-slb. The balance-slb algorithm assumes that upstream switch links are independent layer-2 interfaces and handles broadcast, unknown, and multicast (BUM) traffic accordingly, selectively listening for this traffic on only a single active adapter in the bond.



Note: Do not use IGMP snooping on physical switches connected to Nutanix servers using balance-slb. Balance-slb forwards inbound multicast traffic on only a single active adapter and discards multicast traffic from other adapters. Switches with IGMP snooping may discard traffic to the active adapter and only send it to the backup adapters. This mismatch leads to unpredictable multicast traffic behavior. Disable IGMP snooping or configure static IGMP groups for all switch ports connected to Nutanix servers using balance-slb. IGMP snooping is often enabled by default on physical switches.

Configure the balance-slb algorithm for each bond on all AHV nodes in the Nutanix cluster with the following commands:

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=balance-slb"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:bond-rebalance-interval=30000"
```

Repeat this configuration on all CVMs in the cluster.

Verify the proper bond mode on each CVM with the following commands:

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up "
```

```
---- br0-up ----
bond_mode: balance-slb
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
next rebalance: 29108 ms
lacp_status: off
slave eth2: enabled
    may_enable: true
    hash 120: 138065 kB load
    hash 182: 20 kB load
slave eth3: enabled
    active slave
    may_enable: true
    hash 27: 0 kB load
    hash 31: 20 kB load
    hash 104: 1802 kB load
    hash 206: 20 kB load
```

LACP and Link Aggregation

Taking full advantage of the bandwidth provided by multiple links to upstream switches from a single VM requires link aggregation in OVS negotiated dynamically with LACP and load balancing using balance-tcp. Nutanix recommends dynamic link aggregation with LACP instead of static link aggregation due to improved failure detection and recovery.



Note: Ensure that you've appropriately configured the upstream switches before enabling LACP. On the switch, link aggregation is commonly referred to as port

channel or LAG, depending on the switch vendor. Using multiple upstream switches may require additional configuration such as MLAG or vPC. Configure switches to fall back to active-backup mode in case LACP negotiation fails (sometimes called fallback or no suspend-individual). This setting assists with node imaging and initial configuration where LACP may not yet be available.

With link aggregation negotiated by LACP, multiple links to separate physical switches appear as a single layer-2 link. A traffic-hashing algorithm such as balance-tcp can split traffic between multiple links in an active-active fashion. Because the uplinks appear as a single L2 link, the algorithm can balance traffic among bond members without any regard for switch MAC address tables. Nutanix recommends using balance-tcp when LACP and link aggregation are configured, because each TCP stream from a single VM can potentially use a different uplink in this configuration. With link aggregation, LACP, and balance-tcp, a single user VM with multiple TCP streams could use up to 20 Gbps of bandwidth in an AHV node with two 10 Gb adapters.

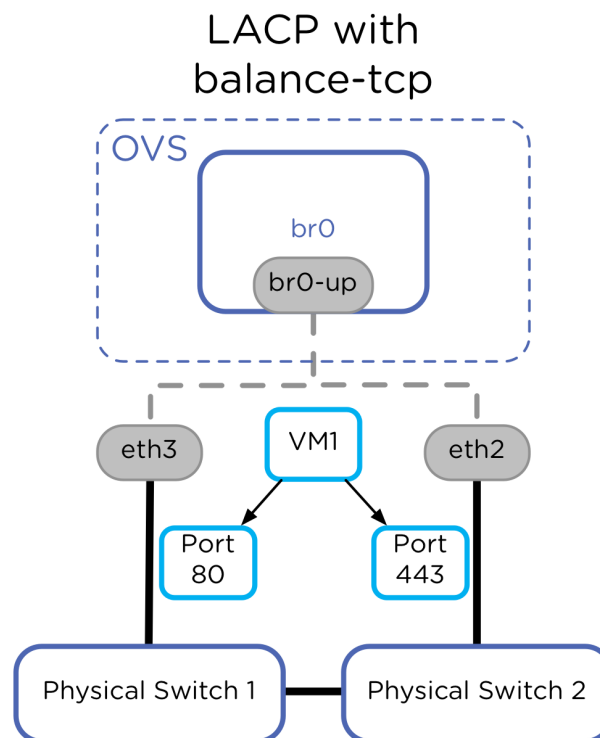


Figure 13: LACP and Balance-TCP Load Balancing

Configure link aggregation with LACP and balance-tcp using the commands below on all Nutanix CVMs in the cluster.



Note: You must configure upstream switches for link aggregation with LACP before configuring the AHV host from the CVM. Upstream LACP settings such as timers should match the AHV host settings for configuration consistency.

If upstream LACP negotiation fails, the default AHV host configuration disables the bond, thus blocking all traffic. The following command allows fallback to active-backup bond mode in the AHV host in the event of LACP negotiation failure:

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:lacp-fallback-ab=true"
```

In the AHV host and on most switches, the default OVS LACP timer configuration is slow, or 30 seconds. This value—which is independent of the switch timer setting—determines how frequently the AHV host requests LACPDUs from the connected physical switch. The fast setting (1 second) requests LACPDUs from the connected physical switch every second, thereby helping to detect interface failures more quickly. Failure to receive three LACPDUs—in other words, after 3 seconds with the fast setting—shuts down the link within the bond. Nutanix recommends setting lacp-time to fast to decrease link failure detection time from 90 seconds to 3 seconds. Only use the slow lacp-time setting if the physical switch requires it for interoperability.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:lacp-time=fast"
```

Next, enable LACP negotiation and set the hash algorithm to balance-tcp.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up lacp=active"
```

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=balance-tcp"
```

Confirm the LACP negotiation with the upstream switch or switches using ovs-appctl, looking for the word "negotiated" in the status lines.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up"
```

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl lacp/show br0-up"
```

Storage Traffic between CVMs

Using active-backup or any other OVS load balancing method, it is not possible to select the active adapter for the CVM in a way that is persistent between host reboots. When multiple uplinks from the AHV host connect to multiple switches, ensure that adequate bandwidth exists between these switches to support Nutanix CVM replication traffic between nodes. Nutanix recommends redundant 40 Gbps or faster connections between switches. A leaf-spine configuration or direct inter-switch link can satisfy this recommendation.

5.4. VLANs for AHV Hosts and CVMs

The recommended VLAN configuration is to place the CVM and AHV host in the untagged VLAN (sometimes called the native VLAN) as shown in the figure below. Neither the CVM nor the AHV host requires special configuration with this option. Configure the switch to allow tagged VLANs

for guest VM networks to the AHV host using standard 802.1Q VLAN tags. Also, configure the switch to send and receive untagged traffic for the CVM and AHV host's VLAN. Choose any VLAN on the switch other than 1 as the native untagged VLAN on ports facing AHV hosts.

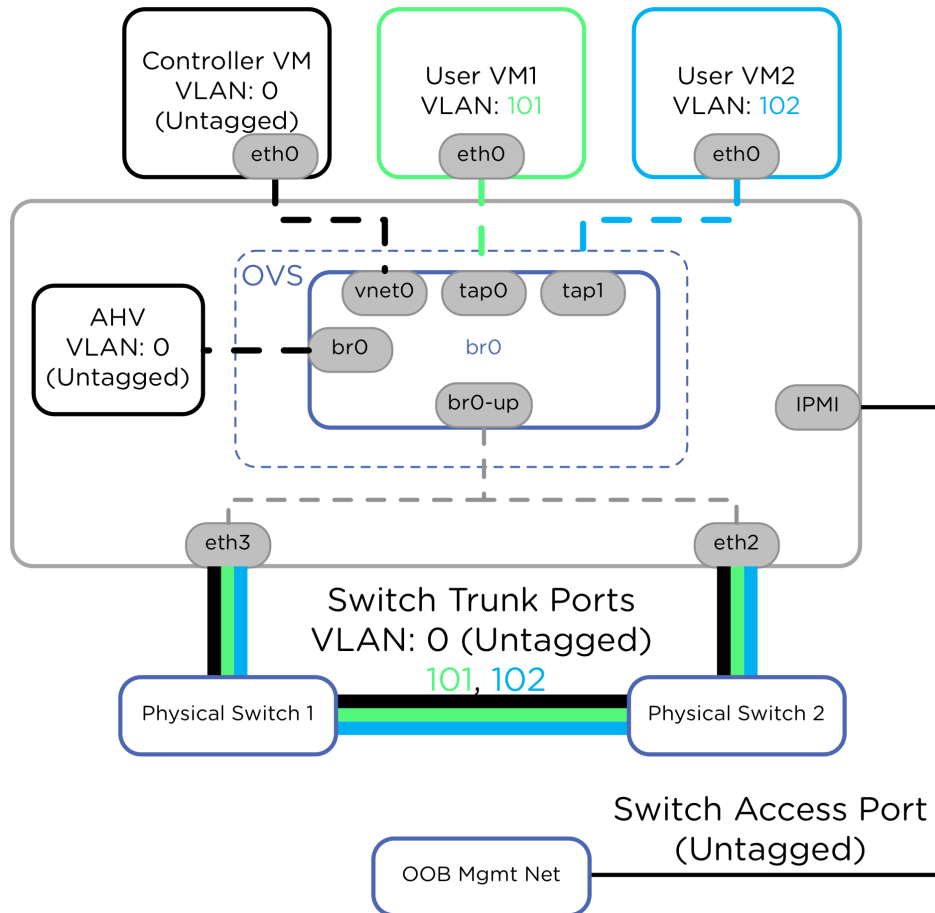


Figure 14: Default Untagged VLAN for CVM and AHV Host

The setup above works well for situations where the switch administrator can set the CVM and AHV VLAN to untagged. However, if you do not want to send untagged traffic to the AHV host and CVM, or if security policy doesn't allow this configuration, you can add a VLAN tag to the host and the CVM with the following procedure.

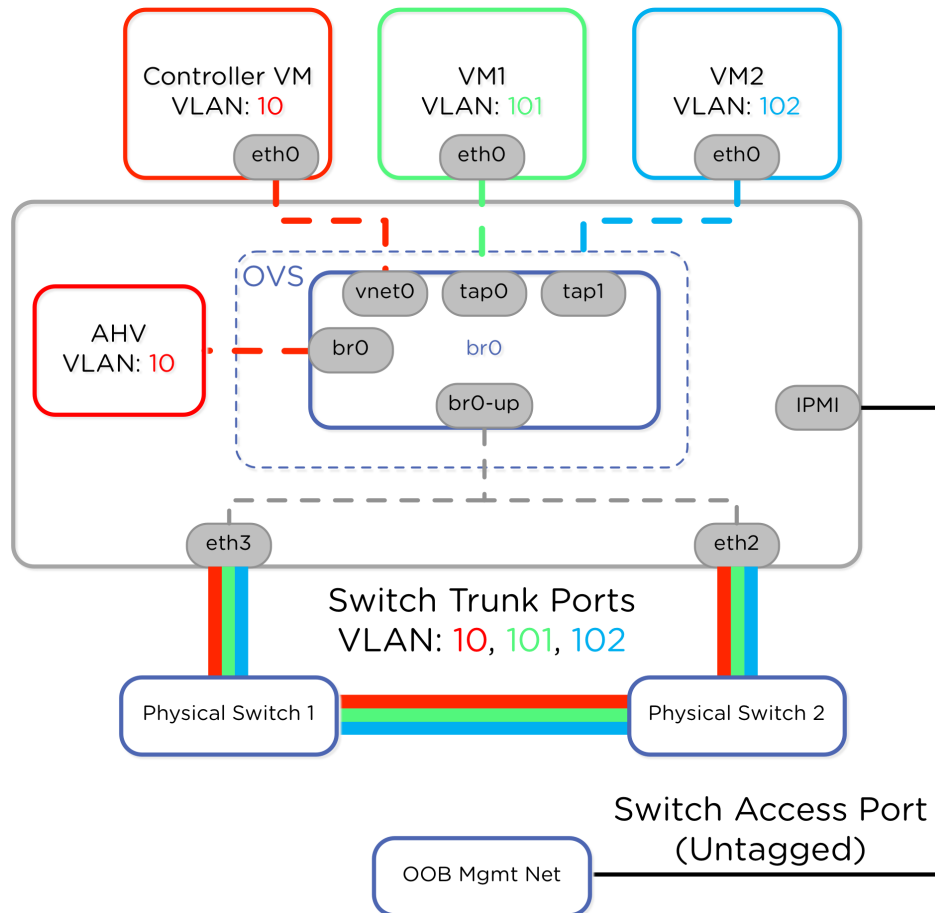


Figure 15: Tagged VLAN for CVM and AHV Host



Note: Ensure that IPMI console access is available for recovery before starting this configuration. Refer to the Configuring Without Connectivity: Staging and Recovery section above for more information.

- Configure VLAN tags on br0 on every AHV host in the cluster.
- Configure VLAN tags for the CVM on every CVM in the Nutanix cluster.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0 tag=10"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl list port br0"
```

```
nutanix@CVM$ change_cvm_vlan 10
```



Tip: Perform the commands above individually on every AHV host and CVM. Nutanix does not recommend using the allssh or hostssh shortcuts to set VLAN tags,

because if the switch is misconfigured, using a shortcut could disconnect all Nutanix nodes in a cluster.

- To remove VLAN tags and revert to the default settings, perform the following on every CVM.

```
nutanix@CVM$ change_cvm_vlan 0
```

- Perform the following on every AHV host to remove VLAN settings.

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0 tag=0"
```

- If you need network segmentation between storage backplane and management traffic within the CVM to meet security requirements, please see the [CVM Network Segmentation feature documentation](#).

5.5. CVM Network Segmentation

Use the **Controller VM Interfaces** menu to create a private CVM backplane network if required. The optional backplane LAN creates a dedicated interface in a separate VLAN on all CVMs in the cluster for exchanging storage replication traffic. The backplane network shares the same physical adapters on bridge br0, but uses a different nonroutable VLAN. Use the backplane network only if there is a requirement to separate CVM management traffic such as Prism from storage replication traffic. The official [Network Segmentation documentation](#) includes diagrams and configuration instructions.

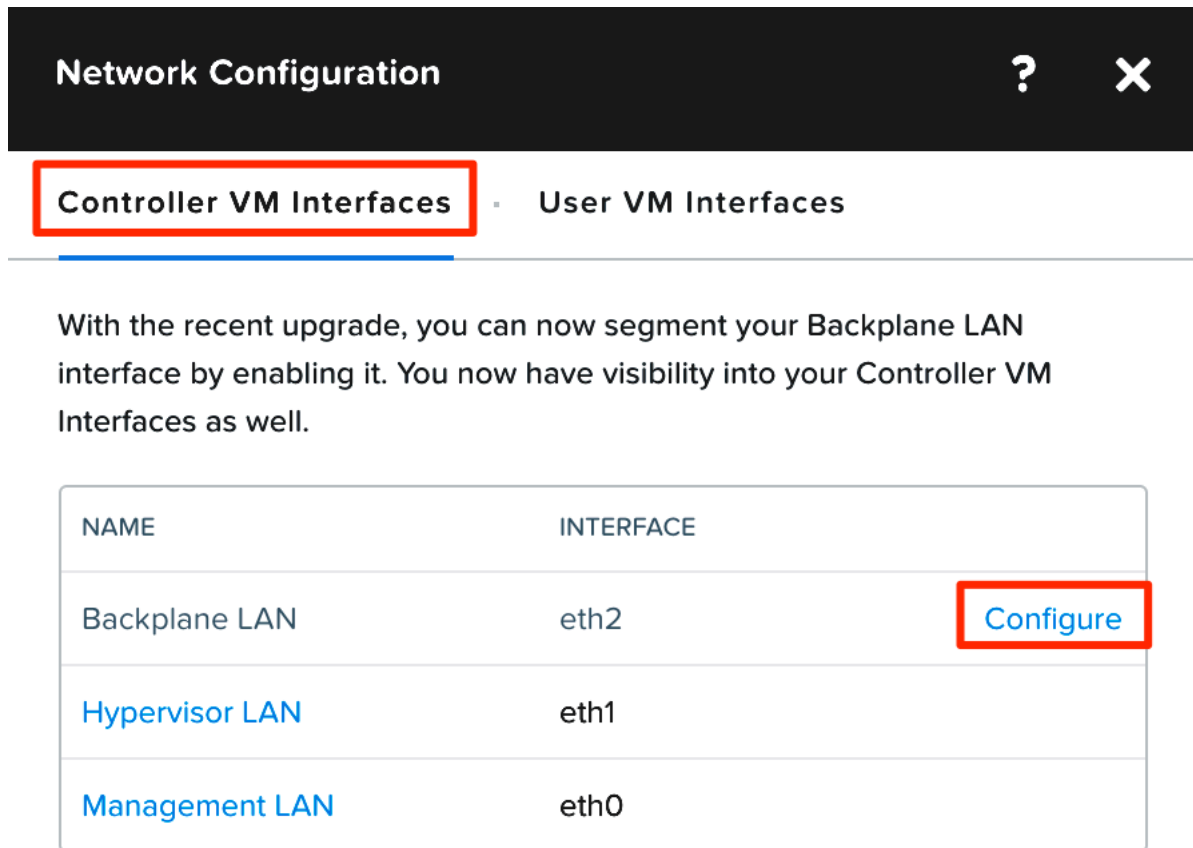


Figure 16: Prism UI CVM Network Interfaces

5.6. Jumbo Frames

Ethernet frames with a payload larger than 1,500 bytes are called jumbo frames and can reduce network overhead and increase throughput due to the reduced number of frames on the network, each carrying a larger payload. The conventional jumbo frame size is 9,000 bytes of Ethernet payload.

Nutanix AHV uses the standard Ethernet MTU (Max Transmission Unit) of 1,500 bytes for all external communication by default. Jumbo frames are optional, because the standard 1,500 byte MTU still delivers excellent performance, but increasing the MTU can be beneficial in some scenarios. For example, you can configure jumbo frames in your network when using Nutanix Volumes to connect high-performance external servers to Nutanix volume groups over iSCSI. This configuration increases efficiency for applications such as databases connecting to Volumes.



Note: Nutanix currently does not recommend jumbo frames when using AHV unless specifically required by high-performance Nutanix Volumes iSCSI workloads or guest VMs. When switching from 1,500 byte frames to 9,000 byte frames, performance improvements are generally not significant in most cases.

Be sure to plan carefully before enabling jumbo frames. The entire network infrastructure, Nutanix cluster, and components that connect to Nutanix must all be configured properly to avoid packet fragmentation, poor performance, or disrupted connectivity. Configuration details for increasing the network MTU to enable jumbo frames on AHV hosts can be found in [KB 3529](#) and [KB 1807](#) for CVMs. Work with Nutanix support before enabling jumbo frames.

5.7. Virtual NIC Trunking

When using the aCLI, you can create a virtual NIC in trunked mode, which allows multiple tagged VLANs on a single virtual NIC. With the aCLI, you can also change an existing virtual NIC from access mode (with only a single untagged VLAN) to trunked mode and modify an existing trunked virtual NIC's VLANs.

To create a new virtual NIC in network "vlan100" with access to VLANs 200 and 300 for VM1:

```
nutanix@CVM$ acli
<acropolis> vm.nic_create VM1 network=vlan100 vlan_mode=kTrunked trunked_networks=200,300
NicCreate: complete
<acropolis> vm.get VM1
(truncated output)
nic_list {
    mac_addr: "52:54:00:76:8b:2c"
    network_name: "vlan100"
    network_uuid: "f0d549f7-0717-4712-9a7d-ac3b513b74e3"
    trunked_networks: 200
    trunked_networks: 300
    vlan_mode: "kTrunked"
}
```

To modify an existing virtual NIC:

```
nutanix@CVM$ acli
<acropolis> vm.nic_update VM1 52:54:00:76:8b:2c update_vlan_trunk_info=true
vlan_mode=kTrunked trunked_networks=200,300,400
NicUpdate: completed
```

6. Conclusion

Nutanix recommends using the default AHV networking settings, configured via the Prism GUI, for most Nutanix deployments. But when your requirements demand specific configuration outside the defaults, this advanced networking guide provides detailed CLI configuration examples that can help.

Administrators can use the Nutanix CLI to configure advanced networking features on a single host or, conveniently, on all hosts at once. VLAN trunking for guest VMs allows a single VM NIC to pass traffic on multiple VLANs for network-intensive applications. You can apply VLAN tags to the AHV host and CVM in situations that require all traffic to be tagged. Grouping host adapters in different ways can provide physical traffic isolation or allow advanced load balancing and link aggregation to provide maximum throughput and redundancy for VMs and hosts.

With these advanced networking techniques, administrators can configure a Nutanix system with AHV to meet the demanding requirements of any VM or application.

For feedback or questions, please contact us using the [Nutanix NEXT Community forums](#).

Appendix

AHV Networking Terminology

Table 4: Networking Terminology Matrix

AHV Term	VMware Term	Microsoft Hyper-V or SCVMM Term
Bridge	vSwitch, Distributed Virtual Switch	Virtual switch, logical switch
Bond	NIC team	Team or uplink port profile
Port or tap	Port	N/A
Network	Port group	VLAN tag or logical network
Uplink	pNIC or vmnic	Physical NIC or pNIC
VM NIC	vNIC	VM NIC
Internal port	VMkernel port	Virtual NIC
Active-backup	Active-standby	Active-standby
Balance-slb	Route based on source MAC hash combined with route based on physical NIC load	Switch independent / dynamic
LACP with balance-tcp	LACP and route based on IP hash	Switch dependent (LACP) / address hash

AHV Networking Best Practices Checklist

- Command line
 - # Use the Prism GUI network visualization feature before using the command line.
 - # Use the allssh and hostssh shortcuts primarily with view and show commands. Use extreme caution with commands that make configuration changes, as the shortcuts execute them on every CVM or AHV host. Executing a disruptive command on all hosts risks disconnecting all hosts.

- # Ensure that IPMI console connectivity is available before making any host networking changes.
- # Connect to a CVM instead of to the AHV hosts when using SSH. Use the `hostssh` or `192.168.5.1` shortcut for any AHV host operation.
- # For high availability, connect to the cluster Virtual IP (VIP) for cluster-wide commands entered in the aCLI rather than to a single CVM.
- Open vSwitch
 - # Do not modify the OpenFlow tables associated with the default OVS bridge `br0`; the AHV host, CVM, and IPAM rely on this bridge.
 - # While it is possible to set QoS policies and other network configuration on the VM tap interfaces manually (using the `ovs-vsctl` command), we do not recommend it. Policies are ephemeral and do not persist across VM power cycles or migrations between hosts.
 - # Do not delete or rename OVS bridge `br0`.
 - # Do not modify the native Linux bridge `virbr0`.
- OVS bonds
 - # Include at least two physical interfaces in every bond.
 - # Aggregate the 10 Gb or faster interfaces on the physical host to an OVS bond named `br0-up` on the default OVS bridge `br0` and trunk VLANs to these interfaces on the physical switch.
 - # Use active-backup load balancing unless you have a specific need for LACP with `balance-tcp`.
 - # Create a separate bond and bridge for the connected 1 Gb interfaces, or remove them from the primary bond `br0-up`.
 - # Do not include 1 Gb interfaces in the same bond or bridge as the 10 Gb or faster interfaces.
 - # If required, connect the 1 Gb interfaces to different physical switches than those connecting to the 10 Gb or faster to provide physical network separation for user VMs.
 - # Use LACP with `balance-tcp` only if guest VMs require link aggregation. Ensure that you have completed LACP configuration on the physical switches first.
 - # Do not use the `balance-tcp` algorithm without upstream switch link aggregation such as LACP.
 - # Do not use the `balance-slb` algorithm if the physical switches use IGMP snooping and pruning.
 - # Do not use the `balance-slb` algorithm with link aggregation such as LACP.

- Physical network layout
 - # Use redundant top-of-rack switches in a leaf-spine architecture. This simple, flat network design is well suited for a highly distributed, shared-nothing compute and storage architecture.
 - # Add all the nodes that belong to a given cluster to the same layer-2 network segment.
 - # If you need more east-west traffic capacity, add spine switches.
 - # Use redundant 40 Gbps (or faster) connections to ensure adequate bandwidth between upstream switches.
- Upstream physical switch specifications
 - # Connect the 10 Gb or faster uplink ports on the AHV node to switch ports that are nonblocking datacenter-class switches that provide line-rate traffic throughput.
 - # Use an Ethernet switch that has a low-latency, cut-through design, and that provides predictable, consistent traffic latency regardless of packet size, traffic pattern, or the features enabled on the 10 Gb interfaces. Port-to-port latency should be no higher than two microseconds.
 - # Use fast-convergence technologies (such as Cisco PortFast) on switch ports that are connected to the AHV host.
 - # To prevent packet loss from oversubscription, avoid switches that use a shared port-buffer architecture.
- Switch and host VLANs
 - # Keep the CVM and AHV in the same VLAN. By default, the CVM and the hypervisor are assigned to VLAN 0, which effectively places them on the native untagged VLAN configured on the upstream physical switch.
 - # Configure switch ports connected to AHV as VLAN trunk ports.
 - # Configure any dedicated native untagged VLAN other than 1 on switch ports facing AHV hosts to carry only CVM and AHV traffic.
- Guest VM VLANs
 - # Configure guest VM network VLANs on br0 using the Prism GUI.
 - # Use VLANs other than the dedicated CVM and AHV VLAN.
 - # Use the aCLI to add guest VM network VLANs for additional bridges, and include the bridge name in the network name for easy bridge identification.
 - # Use VM NIC VLAN trunking only in cases where guest VMs require multiple VLANs on the same NIC. In all other cases, add a new VM NIC with a single VLAN in access mode to bring new VLANs to guest VMs.

- CVM network configuration
 - # Do not remove the CVM from either the OVS bridge br0 or the native Linux bridge virbr0.
 - # If required for security, add a dedicated CVM backplane VLAN with a nonroutable subnet to separate CVM storage backplane traffic from CVM management traffic.
- Jumbo frames
 - # Nutanix currently does not recommend jumbo frames when using AHV unless required for high-performance Nutanix Volumes iSCSI workloads. Performance improvements are generally not significant when switching from 1,500 byte frames to 9,000 byte frames. See [KB 3529](#) and [KB 1807](#) and contact Nutanix support for jumbo frame configuration.
- IP address management
 - # Coordinate the configuration of IP address pools to avoid address overlap with existing network DHCP pools.
 - # Confirm IP address availability with the network administrator before configuring an IPAM address pool in AHV.
- IPMI ports
 - # Do not allow multiple VLANs on switch ports that connect to the IPMI interface. For management simplicity, only configure the IPMI switch ports as access ports in a single VLAN.

AHV Command Line Tutorial

Nutanix systems have a number of command line utilities that make it easy to inspect the status of network parameters and adjust advanced attributes that may not be available in the Prism GUI. In this section, we address the three primary locations where you can enter CLI commands.

The first such location is in the CVM BASH shell. A command entered here takes effect locally on a single CVM. Administrators can also enter CLI commands in the CVM aCLI shell. Commands entered in aCLI operate on the level of an entire Nutanix cluster, even though you're accessing the CLI from one CVM. Finally, administrators can enter CLI commands in an AHV host's BASH shell. Commands entered here take effect only on that AHV host. The diagram below illustrates the basic CLI locations.

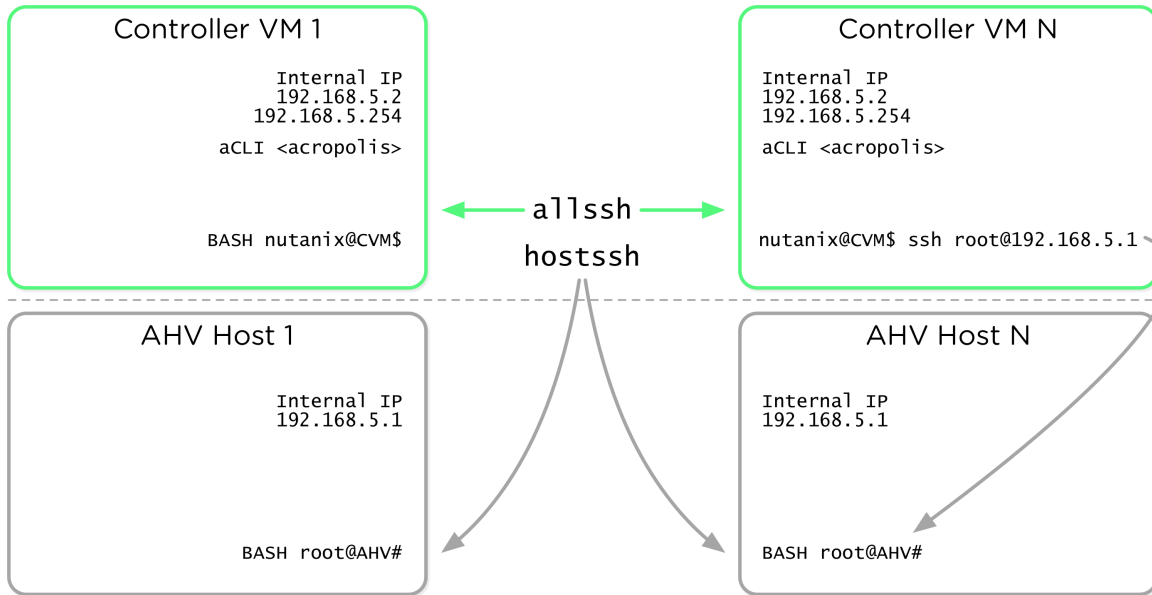


Figure 17: Command Line Operation Overview

CLI shortcuts exist to make cluster management a bit easier. Often, you need to execute a command on all CVMs, or on all AHV hosts, rather than on just a single host. It would be tedious to log on to every system and enter the same command on each of them, especially in a large cluster. That's where the `allssh` and `hostssh` shortcuts come in. `allssh` takes a given command entered on the CVM BASH CLI and executes that command on every CVM in the cluster. `hostssh` works similarly, taking a command entered on the CVM BASH CLI and executing that command on every AHV host in the cluster, as shown in the figure above.

To streamline the management of CVMs and AHV hosts, the SSH shortcut connects a single CVM directly to the local AHV host. From any single CVM, you can use SSH to connect to the AHV host's local address at IP address 192.168.5.1. Similarly, any AHV host can SSH to the local CVM using the static IP address 192.168.5.254. Because the address 192.168.5.2 on a CVM is used for dynamic high availability purposes in the AHV host, it may not always direct to the local CVM. This SSH connection uses the internal Linux bridge `virbr0`.

Let's take a look at a few examples to demonstrate the usefulness of these commands.

Example 1: `allssh`

Imagine that we need to determine which network interfaces are plugged in on all nodes in the cluster, and the link speed of each interface. We could use `manage_ovs show_interfaces` at each CVM, but instead let's use the `allssh` shortcut. First, SSH into any CVM in the cluster as

the **nutanix** user, then execute the command `allssh "manage_ovs show_interfaces"` at the CVM BASH shell:

```
nutanix@NTNX-A-CVM:10.0.0.25:~$ allssh "manage_ovs show_interfaces"
```

In the sample output below, we've truncated the results after the second node to save space.

```
Executing manage_ovs show_interfaces on the cluster
```

```
===== 10.0.0.25 =====
```

```
name  mode link speed
```

```
eth0  1000 True  1000
```

```
eth1  1000 True  1000
```

```
eth2 10000 True 10000
```

```
eth3 10000 True 10000
```

```
Connection to 10.0.0.25 closed.
```

```
===== 10.0.0.26 =====
```

```
name  mode link speed
```

```
eth0  1000 True  1000
```

```
eth1  1000 True  1000
```

```
eth2 10000 True 10000
```

```
eth3 10000 True 10000
```

```
Connection to 10.0.0.26 closed.
```

Example 2: hostssh

If we wanted to view the MAC address of the eth0 interface on every AHV host, we could connect to each AHV host individually and use `ifconfig eth0`. To make things faster, let's use the `hostssh` shortcut instead. In this example, we still use SSH to connect to the CVM BASH shell, then prefix our desired command with `hostssh`.

```
nutanix@NTNX-A-CVM:10.0.0.25:~$ hostssh "ifconfig eth0 | grep HWaddr"
```

```
===== 10.0.0.23 =====
```

```
eth0      Link encap:Ethernet  HWaddr 0C:C4:7A:46:B1:FE
```

```
===== 10.0.0.22 =====
```

```
eth0      Link encap:Ethernet  HWaddr 0C:C4:7A:46:B2:4E
```

Example 3: aCLI

Administrators can use the aCLI shell to view Nutanix cluster information that might not be easily available in the Prism GUI. For example, let's list all of the VMs in a given network. First, connect to any CVM using SSH, then enter the aCLI.

```
nutanix@NTNX-A-CVM:10.0.0.25:~$ acli
<acropolis> net.list_vms 1GBNet
VM UUID                                VM name  MAC address
0d6afd4a-954d-4fe9-a184-4a9a51c9e2c1  VM2      50:6b:8d:cb:1b:f9
```

Example 4: ssh root@192.168.5.1

The shortcut between the CVM and AHV host can be helpful when we're connected directly to a CVM but need to view some information or execute a command against the local AHV host instead. In this example, we're verifying the localhost line of the /etc/hosts file on the AHV host while we're already connected to the CVM.

```
nutanix@NTNX-14SM36510031-A-CVM:10.0.0.25:~$ ssh root@192.168.5.1 "cat /etc/hosts | grep 127"
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

With these command line utilities, we can manage a large number of Nutanix nodes at once. Centralized management helps administrators apply configuration consistently and verify configuration across a number of servers.

AHV Networking Command Examples

- Network view commands

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br0 show_uplinks"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl show"
nutanix@CVM$ acli
<acropolis> net.list
<acropolis> net.list_vms vlan.0
nutanix@CVM$ manage_ovs --help
nutanix@CVM$ allssh "manage_ovs show_interfaces"
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> show_uplinks"
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> --interfaces <interfaces>
update_uplinks"
nutanix@CVM$ allssh "manage_ovs --bridge_name <bridge> --interfaces <interfaces> --
require_link=false update_uplinks"
```

- Bond configuration for 2x 10 Gb

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 create_single_bridge"
nutanix@CVM$ manage_ovs --bridge_name br0 --bond_name br0-up --interfaces 10g update_uplinks
nutanix@CVM$ manage_ovs --bridge_name br1 --bond_name br1-up --interfaces 1g update_uplinks
nutanix@cvm$ acli net.create br1_vlan99 vswitch_name=br1 vlan=99
```

- Bond configuration for 4x 10 Gb

```
nutanix@CVM$ allssh "manage_ovs --bridge_name br1 create_single_bridge"
nutanix@CVM$ allssh "manage_ovs --bridge_name br2 create_single_bridge"
nutanix@CVM$ manage_ovs --bridge_name br0 --bond_name br0-up --interfaces eth4,eth5
update_uplinks
nutanix@CVM$ manage_ovs --bridge_name br1 --bond_name br1-up --interfaces eth2,eth3
update_uplinks
nutanix@CVM$ manage_ovs --bridge_name br2 --bond_name br2-up --interfaces eth0,eth1
update_uplinks
nutanix@cvm$ acli net.create br1_vlan99 vswitch_name=br1 vlan=99
nutanix@cvm$ acli net.create br2_vlan100 vswitch_name=br2 vlan=100
```

- Load balance view command

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show"
```

- Load balance active-backup configuration

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=active-backup"
```

- Load balance balance-slb configuration

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=balance-slb"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:bond-rebalance-
interval=30000"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up "
```

- Load balance balance-tcp and LACP configuration

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:lacp-fallback-
ab=true"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:lacp-time=fast"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up lacp=active"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=balance-tcp"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl bond/show br0-up "
nutanix@CVM$ ssh root@192.168.5.1 "ovs-appctl lacp/show br0-up "
```

- CVM and AHV host tagged VLAN configuration

```
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0 tag=10"
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl list port br0"
nutanix@CVM$ change_cvm_vlan 10
nutanix@CVM$ change_cvm_vlan 0
nutanix@CVM$ ssh root@192.168.5.1 "ovs-vsctl set port br0 tag=0"
```

References

1. [AHV Best Practices Guide](#)
2. [AHV Administration Guide: Host Network Management](#)
3. [AHV Administration Guide: VM Network Management](#)
4. [Prism Web Console Guide: Securing Traffic Through Network Segmentation](#)
5. [Open vSwitch Documentation](#)
6. [Prism Web Console Guide: Network Visualization](#)
7. [KB-3529 Changing Host MTU for Jumbo Frames on AHV](#)
8. [KB-1807 Changing CVM MTU for Jumbo Frames on AHV](#)

About the Author

Jason Burns is an NPP-certified Technical Marketing Engineer at Nutanix, Inc. and CCIE Collaboration #20707. He designs, tests, and documents virtual workloads on the Nutanix platform, creating solutions that solve critical business problems.

Jason has designed and supported Unified Communications infrastructure in the enterprise for the past decade, deploying UC to connect hundreds of thousands of end-users. Outside of his day job, he has an unusual passion for certificates, security, and motorcycles.

Follow Jason on Twitter [@bbbburns](#).

About Nutanix

Nutanix makes infrastructure invisible, elevating IT to focus on the applications and services that power their business. The Nutanix Enterprise Cloud OS leverages web-scale engineering and consumer-grade design to natively converge compute, virtualization, and storage into a resilient, software-defined solution with rich machine intelligence. The result is predictable performance, cloud-like infrastructure consumption, robust security, and seamless application mobility for a broad range of enterprise applications. Learn more at www.nutanix.com or follow us on Twitter [@nutanix](#).

List of Figures

Figure 1: Nutanix Enterprise Cloud.....	7
Figure 2: Post-Imaging Network State.....	11
Figure 3: Prism UI Network Creation.....	12
Figure 4: IPAM.....	13
Figure 5: Prism UI Network List.....	14
Figure 6: Prism UI VM Network Details.....	14
Figure 7: AHV Host Network Visualization.....	15
Figure 8: Network Connections for 2x 10 Gb NICs.....	21
Figure 9: Network Connections for 2x 10 Gb and 2x 1 Gb NICs.....	22
Figure 10: Network Connections for 4x 10 Gb and 2x 1 Gb NICs.....	25
Figure 11: Active-Backup Fault Tolerance.....	28
Figure 12: Balance-SLB Load Balancing.....	30
Figure 13: LACP and Balance-TCP Load Balancing.....	32
Figure 14: Default Untagged VLAN for CVM and AHV Host.....	34
Figure 15: Tagged VLAN for CVM and AHV Host.....	35
Figure 16: Prism UI CVM Network Interfaces.....	37
Figure 17: Command Line Operation Overview.....	44

List of Tables

Table 1: Document Version History.....	6
Table 2: Bridge and Bond Use Cases.....	19
Table 3: Load Balancing Use Cases.....	27
Table 4: Networking Terminology Matrix.....	40