Juniper Networks Virtual Chassis Fabric and Nutanix Acropolis Hypervisor

Reference Architecture

June 2016
Table of Contents

1 Executive Summary ................................................................................................................................. 1

2 Introduction .................................................................................................................................................. 1
   2.1 Audience ....................................................................................................................................................... 1
   2.2 Purpose .......................................................................................................................................................... 1

3 Nutanix Overview ....................................................................................................................................... 2
   3.1 What Is the Nutanix Architecture? .................................................................................................................. 2
   3.2 Acropolis Hypervisor ...................................................................................................................................... 5
      3.2.1 Acropolis Architecture .................................................................................................................................. 6
      3.2.2 VM Networking ............................................................................................................................................. 7
      3.2.3 IP Address Management ............................................................................................................................ 8
      3.2.4 Acropolis Clones .......................................................................................................................................... 9

4 Juniper Networks Virtual Chassis Fabric ................................................................................................. 10
   4.1 Platforms ..................................................................................................................................................... 11
      4.1.1 Juniper Networks QFX5100-24Q ................................................................................................................... 11
      4.1.2 Juniper Networks QFX5100-48S ................................................................................................................... 12
   4.2 Virtual Chassis Fabric Topology ................................................................................................................... 12
   4.3 Performance and Scale .................................................................................................................................. 13
   4.4 High Availability .......................................................................................................................................... 15
   4.5 Adaptive Flowlet Splicing ............................................................................................................................ 16
   4.6 Implementation ............................................................................................................................................ 16
      4.6.1 Configure Virtual Management Ethernet Interface ............................................................................................ 17
      4.6.2 Auto-Provisioned Mode ................................................................................................................................ 18
      4.6.3 Installing the First Spine Switch .................................................................................................................... 18
      4.6.4 Installing the First Leaf Switch .................................................................................................................... 20
      4.6.5 Installing the Remaining Leaf Switches ........................................................................................................ 20
      4.6.6 Installing the Last Spine Switch .................................................................................................................. 21
      4.6.7 Configure High Availability ....................................................................................................................... 23
   4.7 Using Virtual Chassis Fabric ..................................................................................................................... 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7.1</td>
<td>Adding VLANs</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td><strong>Application Overview</strong></td>
<td>28</td>
</tr>
<tr>
<td>5.1</td>
<td>What Is Citrix XenDesktop?</td>
<td>28</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Deployment Scenario: Scripted, Full Clones</td>
<td>29</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Power Management</td>
<td>30</td>
</tr>
<tr>
<td>5.2</td>
<td>Citrix XenDesktop the Nutanix Way</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td><strong>Solution Design</strong></td>
<td>32</td>
</tr>
<tr>
<td>6.1</td>
<td>Desktop Optimizations</td>
<td>36</td>
</tr>
<tr>
<td>6.2</td>
<td>XenDesktop on AHV</td>
<td>37</td>
</tr>
<tr>
<td>6.2.1</td>
<td>AHV Pod Design</td>
<td>38</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Hosted Virtual Desktop I/O Path</td>
<td>38</td>
</tr>
<tr>
<td>6.3</td>
<td>Nutanix: Compute and Storage</td>
<td>39</td>
</tr>
<tr>
<td>6.4</td>
<td>Network</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td><strong>Solution Application</strong></td>
<td>41</td>
</tr>
<tr>
<td>7.1</td>
<td>Port Configuration</td>
<td>41</td>
</tr>
<tr>
<td>7.2</td>
<td>Scenario: 480 Desktops</td>
<td>42</td>
</tr>
<tr>
<td>7.3</td>
<td>Scenario: 960 Desktops</td>
<td>43</td>
</tr>
<tr>
<td>7.4</td>
<td>Scenario: 1,920 Desktops</td>
<td>44</td>
</tr>
<tr>
<td>7.5</td>
<td>Scenario: 3,840 Desktops</td>
<td>45</td>
</tr>
<tr>
<td>7.6</td>
<td>Scenario: 7,680 Desktops</td>
<td>46</td>
</tr>
<tr>
<td>7.7</td>
<td>Scenario: 15,360 Desktops</td>
<td>47</td>
</tr>
<tr>
<td>7.8</td>
<td>Scenario: 30,720 Desktops</td>
<td>48</td>
</tr>
<tr>
<td>8</td>
<td><strong>Validation and Benchmarking</strong></td>
<td>49</td>
</tr>
<tr>
<td>8.1</td>
<td>Environment Overview</td>
<td>49</td>
</tr>
<tr>
<td>8.2</td>
<td>Test Environment Configuration</td>
<td>49</td>
</tr>
<tr>
<td>8.3</td>
<td>XenDesktop Configuration</td>
<td>51</td>
</tr>
<tr>
<td>8.4</td>
<td>Test Image Preparation</td>
<td>51</td>
</tr>
<tr>
<td>8.5</td>
<td>Test Execution</td>
<td>52</td>
</tr>
<tr>
<td>8.6</td>
<td>Login VSI Benchmark</td>
<td>52</td>
</tr>
<tr>
<td>8.7</td>
<td>How to Interpret the Results</td>
<td>53</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Results</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>9.1 360 Knowledge Worker Desktops</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>Appendix: Configuration</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>References</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>12.1 Table of Figures</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>12.2 Table of Tables</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>About the Authors</td>
<td>62</td>
<td></td>
</tr>
</tbody>
</table>
1 Executive Summary

Citrix XenDesktop on Nutanix with a Juniper Networks Virtual Chassis Fabric network is a powerful solution that offers unrivaled user experience, simple administration, and web-scale flexibility and economics. In this reference architecture, we make recommendations for the design, optimization, and scaling of XenDesktop deployments on Nutanix Acropolis Hypervisor. We used Login VSI and an intelligent scripting framework on Nutanix to simulate real-world workloads in a XenDesktop environment. The Nutanix Xtreme Computing Platform offers high IOPS and low latency; our testing demonstrates that host CPU resources drive desktop user density, rather than concerns about I/O or resource bottlenecks—you can easily deploy 120 Knowledge Worker desktops per node, with four nodes per 2U appliance. In addition to desktop and application performance reliability, when deploying XenDesktop on Nutanix you get unlimited scalability, data locality, Acropolis Hypervisor clones, and a single data store. Nutanix also takes the Citrix commitment to simplicity to another level, with streamlined management, reduced rollout time, and lower operating expenses. The Juniper Networks Virtual Chassis Fabric network allows the operator to easily add additional network capacity with a single-point of management. The network fabric allows for all-active network access for L2, L3, and multicast traffic ensuring high-availability, low latency, and non-blocking bandwidth. Juniper Networks Virtual Chassis Fabric has Adaptive Flowlet Splicing technology to detect application elephant flows and break them down into small flowlets, ensuring that no uplinks become congested. Full application traffic visibility and network management is provided by Juniper Networks Junos Space Network Director.

2 Introduction

2.1 Audience

This reference architecture document is part of the Nutanix Solutions Library and is intended for architecting, designing, managing, and supporting Nutanix infrastructures. Consumers of this document should be familiar with Nutanix Acropolis, Prism and Acropolis Hypervisor (AHV), and Citrix XenDesktop.

We have organized this document to address key items for each role that focuses on enabling a successful design, implementation, and transition to operation.

2.2 Purpose

This document covers the following subject areas:

- Overview of the Nutanix solution.
- Overview of Citrix XenDesktop and its use cases.
- Overview of the Juniper Networks Virtual Chassis Fabric and Network Director solution.
- The benefits of Citrix XenDesktop on Nutanix.
- Architecting a complete Citrix XenDesktop solution on the Nutanix platform.
- Design and configuration considerations when architecting a Citrix XenDesktop solution on Nutanix.
3 Nutanix Overview

3.1 What Is the Nutanix Architecture?
Nutanix delivers a hyperconverged infrastructure solution purpose-built for virtualization and cloud environments. This solution brings the performance and economic benefits of web-scale architecture to the enterprise through the Xtreme Computing Platform (XCP), composed of two product families—Nutanix Acropolis and Nutanix Prism.

Attributes of this solution include:

- Storage and compute resources hyperconverged on x86 servers.
- System intelligence located in software.
- Data, metadata, and operations fully distributed across entire cluster of x86 servers.
- Self-healing to tolerate and adjust to component failures.
- API-based automation and rich analytics.

XCP does not rely on traditional SAN/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 is based on industry-standard Intel processor technology and delivers a unified, scale-out, shared-nothing architecture with no single points of failure.

The Nutanix solution has no LUNs to manage, no RAID groups to configure, and no complicated storage multipathing to set up. All storage management is VM-centric and the Acropolis Distributed Storage Fabric (DSF) storage optimizes I/O at the VM virtual disk level. There is one shared pool of storage that includes flash-based SSDs for high performance and low latency, and high-capacity HDDs for affordable capacity. The file system automatically tiers data across different types of storage devices using Nutanix’s intelligent data placement algorithms. These algorithms make sure the most frequently used data is available in memory or in flash for the fastest possible performance.
With the Distributed Storage Fabric (DSF), a Controller VM (CVM) writes data to local flash memory for fast acknowledgment; the CVM also handles read operations locally for reduced latency and fast data delivery.

Figure 2 shows an overview of the Nutanix architecture, including the Acropolis hypervisor, user VMs, the Nutanix Controller VM, and its local disk devices. Each Controller VM is directly connected to the local storage controller and its associated disks. By using local storage controllers on each host, access to data through DSF is localized, thereby reducing storage I/O latency. DSF ensures that writes are replicated synchronously to at least one other node in the system, distributing data throughout the cluster for resiliency and availability. Having a local storage controller on each host ensures that storage performance as well as storage capacity increase when adding nodes.
Local storage for each node in the architecture appears to the hypervisor as one large pool of shared storage. This allows DSF to support all key virtualization features. Data localization also allows local flash to service I/O operations, which minimizes the effect of noisy VMs. This functionality allows large, mixed-workload clusters that are more efficient and more resilient to failure when compared to traditional architectures with stand-alone, shared, and dual-controller storage arrays.

When VMs move from one hypervisor node to another, such as during high availability, the newly migrated VM’s data is served by the now local CVM. When reading old data (stored on the now remote node CVM), the local CVM forwards the I/O request to the remote CVM. All write I/O occurs locally. All read I/O occurs locally as well; when DSF detects that I/O is occurring from a different node, it migrates the data locally in the background. The data migration takes place on read only to minimize network utilization. Figure 3 shows how data follows the VM as it moves between hypervisor nodes.

Another key feature of DSF is the Nutanix Elastic Deduplication Engine, a software-driven, massively scalable, and intelligent data reduction technology. By eliminating duplicate data, it increases the effective
capacity in the disk tier, as well as the system’s RAM and flash cache tiers. This substantially increases storage efficiency, while also improving performance due to larger effective cache capacity in RAM and flash. Each node in the cluster performs deduplication individually, allowing for efficient and uniform deduplication at scale. This technology is increasingly effective with full persistent clones or P2V migrations.

Figure 4: Elastic Deduplication Engine

3.2 Acropolis Hypervisor

The Acropolis Hypervisor integrates tightly with DSF and provides cost-effective virtualization with consumer-grade management. Acropolis includes a distributed VM management service responsible for storing VM configuration, making scheduling decisions, and exposing a management interface. The Acropolis Hypervisor is a full type-1 virtualization solution. The Prism interface, a robust REST API, and an interactive command line interface called aCLI (Acropolis CLI) combine to eliminate the complex management associated with open source hypervisors. The Acropolis Hypervisor provides the following capabilities:

- **Virtual machine storage.** Storage devices for the virtual machine, such as SCSI and IDE devices.
- **Crash-consistent snapshots.** Includes VM configuration and disk contents.
- **Virtual networks (L2).** Layer-2 network communication between virtual machines and to the external network, with support for multiple vSwitches and VLANs.
- **Managed networks (L3).** IP Address Management (IPAM) to provide Layer-3 addresses for virtual machines.
- **App Mobility Fabric.** Enables unprecedented workload flexibility. The App Mobility Fabric also delivers a range of virtualization capabilities, such as High Availability (HA) and disaster recovery.
3.2.1 Acropolis Architecture

The Acropolis Hypervisor extends the open source hypervisor’s base functionality. True to web-scale design principles, Acropolis is available on all nodes with a shared-nothing architecture. Command and control is maintained through a master-subordinate mechanism in which the cluster nodes elect the master and replace it with a new election in the event of a failure. On a single Acropolis node, the Controller Virtual Machine accesses disks directly, as shown Figure 6.
3.2.2 VM Networking

Acropolis uses Open vSwitch (OvS) for all VM networking, as shown in Figure 7. Open vSwitch manages the bonded interfaces (bond0) that connect the hypervisor to the physical network. All VMs connect into one or more virtual switches for inter-VM communication. VMs can also communicate externally if the vSwitch has external interfaces. Each VM NIC connects to an Open vSwitch tap interface (vnet0, tap1). The Acropolis host connects to the CVM through a local Linux bridge (vnet1 to Local) for iSCSI storage traffic.
**Note:** Acropolis virtualizes CPUs using an HVM (Hardware Virtual Machine) device type. Other devices, such as network adapters, use paravirtualized devices that may require VirtIO drivers in your guest OS.

### 3.2.3 IP Address Management

An Acropolis network can be either managed or unmanaged. When unmanaged, the VM traffic is simply passed to the external network with proper VLAN tags. When managed, the Acropolis IP address management (IPAM) solution leverages VXLAN and OpenFlow rules to intercept DHCP requests and respond with addresses from a configured pool. Figure 8 shows a sample DHCP request using the Nutanix IPAM solution.
3.2.4 Acropolis Clones

When cloning a VM or vDisk, the information about the blocks owned by that vDisk, its block map, is locked. The changes to the clone involve metadata only; no physical I/O takes place. The same method applies for clones of clones. The previously cloned VM acts as the “Base vDisk” and, upon cloning, that vDisk’s data block map is locked and two “clones” are created: one for the VM being cloned and another for the new clone. Both clones inherit the prior block map and any new writes or updates take place on the individual VM’s block maps.
Juniper Networks Virtual Chassis Fabric, shown in Figure 10, is a plug and play Ethernet fabric technology that allows you to combine a set of switches into a single logical switch. Virtual Chassis Fabric has the following benefits:

- Single point of management
- Supports 100MB, 1GE, 10GE, and 40GE interfaces
- Full Layer 2, Layer 3, and Multicast support
- Equal Cost Multi-pathing (ECMP)
- Supports Fiber Channel over Ethernet (FCoE) transit
- Scales up to 20 switches
Juniper Networks Virtual Chassis Fabric allows you to easily manage your data center environment. Everything is, at most, only a single hop away, regardless of where it is located in the data center. Virtual Chassis Fabric offers line-rate performance for all types of workloads.

### 4.1 Platforms

The Virtual Chassis Fabric is constructed using a set of switches. Virtual Chassis Fabric offers the best performance and features with the Juniper Networks QFX5100 series, which comes in two models, QFX5100-24Q and QFX5100-48S, which are explained in detail in the following sections.

#### 4.1.1 Juniper Networks QFX5100-24Q

The Juniper Networks QFX5100-24Q, shown in Figure 11, offers 24 built-in QSFP ports and two modules that can support 4 ports of QSFP, bringing the total number of 40GE ports to 32. One of the great things about the QSFP ports is that you can break a single QSFP into 4x10GE ports.
4.1.2 Juniper Networks QFX5100-48S

The Juniper Networks QFX5100-48S, shown in Figure 12, offers 48x10GE ports and 6x40GE ports.

Virtual Chassis Fabric is most often deployed using the Juniper Networks QFX5100-24Q and QFX5100-48S; these switches complement each other when building a simple 3-stage topology.

4.2 Virtual Chassis Fabric Topology

Juniper Networks Virtual Chassis Fabric topology is a spine and leaf architecture that offers the best latency, performance, and scale. As shown in Figure 13, the Juniper Networks QFX5100-24Q is most commonly used as a spine switch and the Juniper Networks QFX5100-48S is most commonly used as a leaf switch.
One of the advantages of using Virtual Chassis Fabric is that there are no physical restrictions on where you can connect devices; you can use both the spine and leaf switches to connect servers, routers, or any other device. When creating a small-to-medium sized data center, port flexibility creates a distinct advantage because a single fabric can connect all servers, storage, firewalls, and even the Internet and WAN.

4.3 Performance and Scale

Juniper Networks Virtual Chassis Fabric is a high-speed Ethernet fabric for every device in the data center. The very nature of the spine and leaf topology enables deterministic traffic patterns and latency, which means that applications are lightning fast. Some of the performance characteristics of Virtual Chassis Fabric include:

- End-to-end latency of 2.5 microseconds
- Line-rate performance of 10GE and 40GE Ethernet
- 1.28Tbps of forwarding capacity per switch
- 25.6Tbps of forwarding capacity for the entire fabric
There is enough scale and performance in a Virtual Chassis Fabric no matter what type of servers and applications you are using. Virtual Chassis Fabric uses a new technology called the Unified Forwarding Table to give you the power to adjust the logical scale of the Ethernet fabric. Virtual Chassis Fabric uses next generation flexible tables, as shown in Figure 14.

![Figure 14: Virtual Chassis Fabric Unified Forwarding Tables](image)

There are five profiles that you may choose from; each profile incrementally increases the amount of Layer 3 scale.

<table>
<thead>
<tr>
<th>Profile</th>
<th>MAC Addresses</th>
<th>L3 Hosts</th>
<th>Long Prefix Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>l2-profile-one</td>
<td>288,000</td>
<td>16,000</td>
<td>16,000</td>
</tr>
<tr>
<td>l2-profile-two</td>
<td>224,000</td>
<td>56,000</td>
<td>16,000</td>
</tr>
<tr>
<td>l3-profile-three</td>
<td>160,000</td>
<td>88,000</td>
<td>16,000</td>
</tr>
<tr>
<td>l3-profile</td>
<td>96,000</td>
<td>120,000</td>
<td>16,000</td>
</tr>
<tr>
<td>lpm-profile</td>
<td>32,000</td>
<td>16,000</td>
<td>128,000</td>
</tr>
</tbody>
</table>
For example, the **l2-profile-one** starts with 288K MAC addresses; each profile incrementally increases the amount of Layer 3 scale you are using. Virtual Chassis Fabric uses a new technology called the Unified Forwarding Table to load of WAN addresses into the data center while at the same time supporting a large number of virtual machines and servers.

### 4.4 High Availability

Virtual Chassis Fabric leverages the functionality from carrier-class routers, such as the Juniper Networks MX and T Series, to provide high availability in the Ethernet fabric. This functionality includes:

- **Graceful Routing Engine Switchover** (GRES) - keeps the operating system state synchronized between the master and backup Routing Engines (RE)
- **Non-Stop Routing** (NSR) - keeps the routing protocol state synchronized between the master and backup REs
- **Non-Stop Bridging** (NSB) - keeps the Layer 2 protocol state synchronized between the master and backup REs

The spine switches act as the master and backup REs in a Virtual Chassis Fabric, as shown in Figure 15:

![Figure 15: Master and Backup Routing Engines in a Virtual Chassis Fabric](image)

The other leaf switches in the Virtual Chassis Fabric act as simple line cards. If the master RE were to experience a failure, the backup RE would immediately become the new master RE. Traffic will not be interrupted because GRES, NSR, and NSB keep the two REs continuously synchronized.

Juniper Networks Virtual Chassis Fabric is an easy to manage, high performance, and highly available Ethernet fabric that allows you to build a best-of-class data center network.
4.5 Adaptive Flowlet Splicing

In the data center, there is a story of elephant and mice. The idea is that there are long-lived flows in the data center network that consume a lot of bandwidth; these types of flows are referred to as elephant flows. Some examples of elephant flows are backups and copying data. Other types of flows are short-lived and consume little bandwidth; these are referred to as mice flows. Examples of mice flows include DNS and NTP.

The problem is that each flow inside of a network switch is subject to ECMP and is pinned to a particular next hop or uplink interface. If a couple of elephant flows get pinned to the same interface and consume all of the bandwidth, they will start to over-run other, smaller mice flows on the same egress uplink. Due to the nature of standard flow hashing, mice flows have a tendency to be trampled by elephant flows, which has a negative impact on application performance.

Juniper Networks Virtual Chassis Fabric has a very unique solution to the elephant and mice problem. If you take a closer look at TCP flows, you will notice something called flowlets. These are the blocks of data being transmitted in between the TCP acknowledgement from the receiver. Depending on the bandwidth, TCP window size, and other variables, flowlets exist in different sizes and frequencies, as illustrated in Figure 16.

![Figure 16: Flowlets in a Virtual Chassis Fabric](image)

One method to solve the elephant and mice problem is to hash the flowlets to different next hops. For example, if Flow 1 and Flow 3 in the above figure were elephant flows, each of the flowlets could be hashed to a different uplink, as opposed to the entire flow being stuck on a single uplink. Virtual Chassis Fabric uses the flowlet hashing functionality to solve the elephant and mice problem. This feature is called Juniper Networks Adaptive Flowlet Splicing (AFS).

4.6 Implementation

Configuring Virtual Chassis Fabric is very straightforward and easy. We can look at all three provisioning modes to understand the configuration differences. We will also take a look at how to add and remove spine and leaf switches. Each provisioning mode is a little different in the configuration and process of expanding the fabric.

Before configuring the switches, there are a few steps that are required before configuring the Virtual Chassis Fabric.
1. Software Version - Make sure that all switches have the same version of Junos OS installed. Use Junos OS 13.2X51-D20 or newer.

2. Un-cable Everything - Before you start to configure Virtual Chassis Fabric, make sure that all of the switches are un-cabled. This is because if you want to use the plug-and-play feature of the auto-provisioned mode, you want to explicitly control the creation of the spine switches, then simply add other switches. For pre-provisioned and non-provisioned, you can cable up the switches.

3. Identify Serial Numbers - Identify the serial numbers for each switch. For auto-provisioned mode, you only need the serial numbers for the spine switches. For pre-provisioned and non-provisioned modes, you will need all of the spine and leaf switch serial numbers.

4. Check for LLDP - LLDP should be turned on by factory default, but always check and make sure it is enabled. Use the command `set protocols lldp interface all` to enable LLDP. Virtual Chassis Fabric uses LLDP to enable the plug-and-play functionality in auto-provisioned mode.

Once you have upgraded the software, un-cabled everything, and identified all of the serial numbers, we can begin to build the Virtual Chassis Fabric.

4.6.1 Configure Virtual Management Ethernet Interface

Follow these steps to configure the virtual management Ethernet interface:

1. Use the following command to configure a management IP address for this switch.

   ```
   {master:0}[edit]
   root# set interfaces vme.0 family inet address 192.0.2.2/24
   ```

2. Set a root password for the switch. The system will prompt you to enter a password, then again for verification.

   ```
   {master:0}[edit]
   root# set system root-authentication plain-text-password
   New password: 
   Retype new password: 
   ```

3. Enable SSH so that we can login and copy files to and from the switch.

   ```
   {master:0}[edit]
   root# set system services ssh root-login allow
   ```

4. Commit the changes for the configuration to take effect:

   ```
   {master:0}[edit]
   root# commit and-quit
   configuration check succeeds
   commit complete
   Exiting configuration mode
   root>
   ```
The switch should now be reachable on the C0 management interface on the rear of the switch. Use the ping command to double-check.

```
host:~ user1$ ping 192.0.2.2
PING 192.0.2.2 (192.0.2.2): 56 data bytes
64 bytes from 192.0.2.2: icmp_seq=0 ttl=55 time=21.695 ms
64 bytes from 192.0.2.2: icmp_seq=1 ttl=55 time=20.222 ms
^C
--- 192.0.2.2 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 20.222/20.959/21.695/0.737 ms
```

Everything looks great. The switch is now upgraded and able to be managed using IP instead of the serial console.

### 4.6.2 Auto-Provisioned Mode

The auto-provisioned mode only requires that you define the spine switches and the serial numbers. Let's take a look at a simple topology, as shown in Figure 17, and what the configuration would be.

![Figure 17: Simple Virtual Chassis Fabric Topology](image)

The topology in Figure 17 has two spines and four leaf switches. In this example, both spine switches need to be configured. The spine switch serial numbers have been identified as in the following table:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Serial Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFX5100-24Q-01</td>
<td>TB3714070330</td>
</tr>
<tr>
<td>QFX5100-24Q-02</td>
<td>TB3714070064</td>
</tr>
</tbody>
</table>

### 4.6.3 Installing the First Spine Switch

Perform the following steps to install the first spine switch.

1. Make sure that the spine switches are in fabric mode, by issuing the following command on both QFX5100-24Q switches:
root> request virtual-chassis mode fabric reboot

The switches will reboot and now be in fabric mode.

2. Begin configuring Virtual Chassis Fabric on the first spine switch by putting the QFX5100-24Q-01 into auto-provisioned mode. We will also support upgrading the software of other switches connected into the fabric with the `auto-sw-upgrade` statement.

**Note:** Don’t worry about the second spine switch QFX5100-24Q-02 for right now. We’ll focus on QFX5100-24Q-01 and move on to the leaf switches. Adding the final spine switch will be the last step when bringing up Virtual Chassis Fabric.

Starting on QFX5100-24Q-01, issue the following commands to begin to configure Virtual Chassis Fabric:

```
[edit]
root# set virtual-chassis auto-provisioned
[edit]
root# set virtual-chassis auto-sw-upgrade
```

3. The next step is to configure the role and serial numbers of all of the spine switches from the previous table.

```
[edit]
root# set virtual-chassis member 0 role routing-engine serial-number TB3714070330
[edit]
root# set virtual-chassis member 1 role routing-engine serial-number TB3714070064
```

4. Next, verify the configuration before committing it:

```
[edit]
root# show virtual-chassis
auto-provisioned;
member 0 {
  role routing-engine;
  serial-number TB3714070330;
}
member 1 {
  role routing-engine;
  serial-number TB3714070064;
}
```

5. Commit the configuration.

```
[edit]
root# commit and-quit
configuration check succeeds
commit complete
Exiting configuration mode
```

The Juniper Networks QFX5100-24Q is now in Virtual Chassis Fabric mode. You can verify this with the `show virtual-chassis` command:

```
{master:0}
root> show virtual-chassis
4.6.4 Installing the First Leaf Switch

Perform the following steps to install the first leaf switch.

1. Login to the first QFX5100-48S-01 and reset the switch to a factory default state:

   root> request system zeroize
   warning: System will be rebooted and may not boot without configuration
   Erase all data, including configuration and log files? [yes,no] (no) yes

   warning: ipsec-key-management subsystem not running - not needed by configuration.
   root> Terminated

   **Note:** If the leaf switches already have Junos 13.2X51-D20 installed and are in a factory default state, you can skip the `request system zeroize` step. You can simply cable the leaf switch into the spine switch.

2. Once the switch reboots and comes back up, simply connect a 40G cable from the Juniper Networks QFX5100-24Q-01 to QFX5100-48S-01, as shown in Figure 18.

   Figure 18: Cabling the First Leaf Switch

   Once the cable is connected, the master QFX5100-24Q-01 will automatically add the new QFX5100-48S-01 into the Virtual Chassis Fabric.

4.6.5 Installing the Remaining Leaf Switches

Repeat the procedure given in the previous section for each QFX5100-48S switches in the Virtual Chassis Fabric as shown below.
Once all of the Juniper Networks QFX5100-48S leaves have been reset to factory default and cabled up, the Juniper Networks QFX5100-24Q-01 switch will bring all of the switches into the Virtual Chassis Fabric. You can verify this with the `show virtual-chassis` command:

```
{master:0}
root> show virtual-chassis

Auto-provisioned Virtual Chassis Fabric
Fabric ID: 742a.6f8b.6de6
Fabric Mode: Enabled

<table>
<thead>
<tr>
<th>Member ID</th>
<th>Status</th>
<th>Serial No</th>
<th>Model</th>
<th>Mstr</th>
<th>Role</th>
<th>Mode</th>
<th>Mode ID</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (FPC 0)</td>
<td>Prsnt</td>
<td>TB3714070330</td>
<td>qfx5100-24q-2p</td>
<td>129</td>
<td>Master*</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td>1 (FPC 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 (FPC 2)</td>
<td>Prsnt</td>
<td>TA3713840228</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/48</td>
</tr>
<tr>
<td>3 (FPC 3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (FPC 4)</td>
<td>Prsnt</td>
<td>TA3713840106</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/48</td>
</tr>
<tr>
<td>5 (FPC 5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The newly added switches appear in italics in the code output above; for reference, they are member ID 2, 3, 4, and 5.

### 4.6.6 Installing the Last Spine Switch

The last step is to add the second QFX5100-24Q-02 spine into the Virtual Chassis Fabric. Repeat the same steps and zeroize the second QFX5100-24Q-02 switch. After the switch reboots, connect the remaining cables into a full mesh as shown in Figure 20:
Wait a couple of minutes and check the status of the Virtual Chassis Fabric again; you should see the missing member 1 as **Prsnt** with a role of Backup:

```bash
user@> show virtual-chassis
```

Auto-provisioned Virtual Chassis Fabric
Fabric ID: 742a.6f8b.6de6
Fabric Mode: Enabled

<table>
<thead>
<tr>
<th>Member ID</th>
<th>Status</th>
<th>Serial No</th>
<th>Model</th>
<th>prio</th>
<th>Role</th>
<th>Mode</th>
<th>Mode ID</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (FPC 0)</td>
<td>Prsnt</td>
<td>TB3714070330</td>
<td>qfx5100-24q-2p</td>
<td>129</td>
<td>Master*</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/4</td>
</tr>
<tr>
<td>1 (FPC 1)</td>
<td>Prsnt</td>
<td>TB3714070064</td>
<td>qfx5100-24q-2p</td>
<td>129</td>
<td>Backup</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/4</td>
</tr>
<tr>
<td>2 (FPC 2)</td>
<td>Prsnt</td>
<td>TA3713480228</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/4</td>
</tr>
<tr>
<td>3 (FPC 3)</td>
<td>Prsnt</td>
<td>TA3713480106</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/4</td>
</tr>
<tr>
<td>4 (FPC 4)</td>
<td>Prsnt</td>
<td>TA3713470455</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vcp-255/0/4</td>
</tr>
<tr>
<td>5 (FPC 5)</td>
<td>Prsnt</td>
<td>TA3713480037</td>
<td>qfx5100-48s-6q</td>
<td>0</td>
<td>Linecard</td>
<td>N</td>
<td>F</td>
<td>vcp-255/0/0</td>
</tr>
</tbody>
</table>

Use the **show interface** command to verify that the new Virtual Chassis Fabric management interface is up:

```bash
{master:0}
root> show interfaces terse vme
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Admin</th>
<th>Link</th>
<th>Proto</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>vme</td>
<td>up</td>
<td>up</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vme.0</td>
<td>up</td>
<td>up</td>
<td>inet</td>
<td>192.0.2.2/24</td>
<td></td>
</tr>
</tbody>
</table>

The astute reader will recognize that this is the same **vme** interface that we originally configured on the Juniper Networks QFX5100-24Q-01 switch when it was in standalone mode. The **vme** configuration has persisted through the placing of the device into Virtual Chassis Fabric. Because the Juniper Networks QFX5100-24Q-01 switch is the master Routing Engine, it also owns the **vme** interface. We can also check the reachability from our laptop:
host:~ user$ ping 192.0.2.2
PING 192.0.2.2 (192.0.2.2): 56 data bytes
64 bytes from 192.0.2.2: icmp_seq=0 ttl=55 time=21.695 ms
64 bytes from 192.0.2.2: icmp_seq=1 ttl=55 time=20.222 ms
^C
--- 192.0.2.2 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 20.222/20.959/21.695/0.737 ms
host:~ user$

As shown, we can now reach the Virtual Chassis Fabric using the built-in management port. We can now move on to the next step.

4.6.7 Configure High Availability

To ensure that the Virtual Chassis Fabric recovers quickly from failures, there are three key features that we need to enable:

- **Graceful Routing Engine Switchover (GRES)**—Synchronize kernel state between the master and backup routing engines
- **Non-Stop Routing (NSR)**—Synchronize routing protocol state between the master and backup routing engines
- **Non-Stop Bridging (NSB)**—Synchronize Layer 2 protocol state between the master and backup Routing Engines

Perform the following steps to configure high availability.

1. The first step is to configure GRES.

   {master:0}[edit]
   user@VCF# set chassis redundancy graceful-switchover
   {master:0}[edit]
   user@VCF# set system commit synchronize

2. Configure NSR and NSB.

   {master:0}[edit]
   user@VCF# set routing-options nonstop-routing
   {master:0}[edit]
   user@VCF# set protocols layer2-control nonstop-bridging
   {master:0}[edit]
   user@VCF# commit and-quit
   configuration check succeeds
   commit complete
   Exiting configuration mode

3. Verify that the master Routing Engine is sending data to the backup Routing Engine through the GRES protocol:

   {master:0}
   user@VCF> show task replication
4. Verify that NSR and NSB are synchronizing state. To do this, login to the backup Routing Engine with the request session command:

```
(master:0)
user@VCF> request session member 1
--- JUNOS 13.2-X51D20
user@VCF:BK:1% clear
user@VCF:BK:1% cli
warning: This chassis is operating in a non-master role as part of a virtual-chassis fabric (VCF) system.
warning: Use of interactive commands should be limited to debugging and VC Port operations.
warning: Full CLI access is provided by the Virtual Chassis Fabric Master (VCF-M) chassis.
warning: The VCF-M can be identified through the show fabric status command executed at this console.
warning: Please logout and log into the VCF-M to use CLI.
```

5. Verify that NSR and NSB are configured properly:

```
(backup:1)
user@VCF> show system switchover
fpc1:

Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady State
(backup:1)
user@VCF> show l2cpd task replication
  Stateful Replication: Enabled
  RE mode: Backup
```

At this point, Virtual Chassis Fabric is configured and ready to be used.

### 4.7 Using Virtual Chassis Fabric

Now that Virtual Chassis Fabric is configured, we will take a look at some of the most common day-to-day tasks and how they work in Virtual Chassis Fabric.

- Adding new VLANs and assigning them to switch ports
- Assigning routed VLAN interfaces so that the fabric can route between VLANs
- Adding access control lists
- Mirroring traffic
- Setting up SNMP to enable monitoring of the fabric

Remember that Virtual Chassis Fabric is just a single, logical switch with many physical components. All configuration is done through a single command-line interface. The Virtual Chassis Fabric also appears as a single, large switch from the perspective of SNMP.
We’ll make the assumption that our Virtual Chassis Fabric has the following topology as shown:

![Virtual Chassis Fabric Topology](image)

**Figure 21: Virtual Chassis Fabric Topology**

### 4.7.1 Adding VLANs

The most basic task is adding new VLANs to the network in order to segment servers, as follows:

1. The first step is to drop into configuration mode and define the VLAN:

   ```
   {master:0}[edit]
   root@VCF# set vlans Engineering description "Broadcast domain for Engineering group"
   {master:0}[edit]
   root@VCF# set vlans Engineering vlan-id 100
   {master:0}[edit]
   root@VCF# set vlans Engineering l3-interface irb.100
   ```

2. Next, create a Layer 3 interface for the new Engineering VLAN so that servers have a default gateway. Use the same `l3-interface` that was referenced in creating the Engineering VLAN.

   ```
   {master:0}[edit]
   root@VCF# set interfaces irb.100 family inet address 198.51.100.10/24
   ```

3. Now that the VLAN and its associated Layer 3 interface is ready to go, the next step is to add servers into the VLAN. We will assume that the first QFX5100-48S switch is in the first rack.

   When working with Virtual Chassis Fabric, each switch is identified by its FPC number. To determine a switch’s FPC number, use the `show chassis hardware` command. You can identify switches by the serial number. It is important to note that, because we used the auto-provision feature in Virtual Chassis Fabric, the FPC numbers are assigned chronologically as new switches are added.

   ```
   {master:0}
   root@VCF> show chassis hardware | match FPC
   FPC 0   REV 11  650-049942  TB3714070330  QFX5100-24Q-2P
   FPC 1   REV 11  650-049942  TB3714070064  QFX5100-24Q-2P
   FPC 2   REV 09  650-049937  TA3713480228  QFX5100-48S-6Q
   FPC 3   REV 09  650-049937  TA3713480106  QFX5100-48S-6Q
   FPC 4   REV 09  650-049937  TA3713470455  QFX5100-48S-6Q
   FPC 5   REV 09  650-049937  TA3713480037  QFX5100-48S-6Q
   ```

   In our example, the FPC numbers are sequential starting from 0 and ending in 5, as shown below.
4. Now that we know that the first switch is FPC2, we can begin to assign the new Engineering VLAN to this switch. The easiest method is to create an alias for all of the 10GE interfaces on this switch; we’ll call this alias rack-01.

```
{master:0}[edit]
root@VCF# set interfaces interface-range rack-01 member-range xe-2/0/0 to xe-2/0/47
{master:0}[edit]
root@VCF# set interfaces interface-range rack-01 description "Alias for all 10GE interfaces on FPC2/rack-02"
{master:0}[edit]
root@VCF# set interfaces interface-range rack-01 unit 0 family ethernet-switching vlan members Engineering
{master:0}[edit]
root@VCF# commit and-quit
configuration check succeeds
commit complete
Exiting configuration mode
```

The new interface alias called **rack-01** is now configured to include all 10GE interfaces from **xe-0/0/0** to **xe-0/0/47** on the front panel.

5. The next step was to assign the Engineering VLAN, which was done using the **vlan members** statement.

6. Verify the configuration with the **show vlans** command:

```
{master:0}
root@VCF> show vlans
Routing instance  VLAN name        Tag  Interfaces
default-switch    Engineering      100  xe-2/0/0.0
                  xe-2/0/1.0
                  xe-2/0/12.0
                  xe-2/0/13.0
                  xe-2/0/2.0
                  xe-2/0/3.0
                  xe-2/0/4.0
                  xe-2/0/5.0
                  xe-2/0/6.0
                  xe-2/0/7.0
```

![Figure 22: Virtual Chassis Fabric](image)
We can now see that all of the interfaces that have optics in rack-01 are now assigned to the Engineering VLAN.

7. Let’s add another VLAN on a different switch for System Test.

[master:0][edit]
root@VCF# set vlans Systest description "Broadcast domain for System Test"
[master:0][edit]
root@VCF# set vlans Systest vlan-id 200
[master:0][edit]
root@VCF# set vlans Systest 13-interface irb.200
[master:0][edit]
root@VCF# set interfaces irb.200 family inet address 203.0.113.18/24
[master:0][edit]
root@VCF# set interfaces interface-range rack-02 member-range xe-3/0/0 to xe-3/0/47
[master:0][edit]
root@VCF# set interfaces interface-range rack-02 description "Alias for all 10GE interfaces on FPC3/rack-03"
[master:0][edit]
root@VCF# set interfaces interface-range rack-02 unit 0 family ethernet-switching vlan members Systest
[master:0][edit]
root@VCF# commit and-quit
configuration check succeeds
commit complete
Exiting configuration mode

8. We can verify that the new System Test VLAN is up and working with the following show commands:

[master:0]
root@VCF> show vlans

<table>
<thead>
<tr>
<th>Routing instance</th>
<th>VLAN name</th>
<th>Tag</th>
<th>Interfaces</th>
</tr>
</thead>
</table>
| default-switch   | Engineering | 100 | xe-2/0/0.0
|                  |           |     | xe-2/0/1.0
|                  |           |     | xe-2/0/12.0
|                  |           |     | xe-2/0/13.0
|                  |           |     | xe-2/0/2.0
|                  |           |     | xe-2/0/3.0
|                  |           |     | xe-2/0/4.0
|                  |           |     | xe-2/0/5.0
|                  |           |     | xe-2/0/6.0
|                  |           |     | xe-2/0/7.0
| default-switch   | Systest   | 200 | xe-3/0/0.0
|                  |           |     | xe-3/0/1.0
|                  |           |     | xe-3/0/12.0
|                  |           |     | xe-3/0/13.0
|                  |           |     | xe-3/0/2.0
|                  |           |     | xe-3/0/3.0

© 2016 Juniper Networks, Inc
5 Application Overview

5.1 What Is Citrix XenDesktop?

Citrix XenDesktop is a desktop virtualization solution that transforms desktops and applications into a secure, on-demand service available to any user, anywhere, on any device. With XenDesktop, you can deliver individual Windows, web, and SaaS applications, or full virtual desktops, to PCs, Macs, tablets, smartphones, laptops, and thin clients with a high-definition user experience.

Citrix XenDesktop provides a complete virtual desktop delivery system by integrating several distributed components with advanced configuration tools that simplify the creation and real-time management of the virtual desktop infrastructure.

The core components of XenDesktop are:

- **Desktop Delivery Controller.** Installed on servers in the data center, the controller authenticates users, manages the assembly of users' virtual desktop environments, and brokers connections between users and their virtual desktops. It controls the state of the desktops, starting and stopping them based on demand, and administrative configuration. The Citrix license needed to run XenDesktop also includes Profile management, in some editions, to manage user personalization settings in virtualized or physical Windows environments.

- **Studio.** Citrix Studio is the management console that will allow you to configure and manage your Citrix XenDesktop environment. It will provide different wizard-based deployment or configuration scenarios to publish resources using desktops or applications.

- **Virtual Desktop Provisioning powered by Citrix Machine Creation Services.** Machine Creation Services (MCS) is the building mechanism of the Citrix Desktop Delivery controller that automates and orchestrates the deployment of desktops using a single image. MCS communicates with the orchestration layer of your hypervisor, providing a robust and flexible method of image management.

- **Virtual Desktop Provisioning powered by Citrix Provisioning Services.** Provisioning Services (PVS) creates and provisions virtual desktops from a single desktop image on demand, optimizing storage utilization and providing a pristine virtual desktop to each user every time they log on.
Desktop provisioning also simplifies desktop images, provides optimal flexibility, and offers fewer points of desktop management for both applications and desktops.

- **Virtual Desktop Agent.** Installed on virtual desktops, the agent enables direct FMA (Flexcast Management Architecture) connections between the virtual desktop and user devices.

- **Citrix Receiver.** Installed on user devices, the Citrix Desktop Receiver enables direct ICA connections from user devices to virtual desktops.

- **Citrix FlexCast.** Citrix XenDesktop with FlexCast delivery technology lets you deliver virtual desktops and applications tailored to meet the diverse performance, security, and flexibility requirements of every worker in your organization through a single solution. Centralized, single-instance management helps you deploy, manage, and secure user desktops more easily and efficiently.

### 5.1.1 Deployment Scenario: Scripted, Full Clones

Nutanix offers scripting frameworks to get started with XenDesktop on AHV. Read about the process of deploying virtual desktops using XenDesktop on Nutanix Acropolis Hypervisor on the Nutanix Next Community ([http://next.nutanix.com/t5/Virtual-Desktop-Infrastructure/XenDesktop-on-Nutanix-Acropolis-Hypervisor-8-Steps/m-p/6633#U6633](http://next.nutanix.com/t5/Virtual-Desktop-Infrastructure/XenDesktop-on-Nutanix-Acropolis-Hypervisor-8-Steps/m-p/6633#U6633)). Here are blogs that cover the scripting framework:


- **Using PowerShell Commandlets:** [http://blog.myvirtualvision.com/2015/10/15/mass-sysprep-your-windows-vms-for-nutanix-ahv/](http://blog.myvirtualvision.com/2015/10/15/mass-sysprep-your-windows-vms-for-nutanix-ahv/)

These scripts create multiple instances cloned from your master image. Nutanix is focusing on persistent desktops for the moment. Planned Machine Creation Service integration will automate this process for non-persistent desktops as well.
5.1.2 Power Management

Nutanix offers a XenDesktop power-management plug-in that allows you to automatically manage desktop power state from within XenDesktop tools. The plug-in is available at https://portal.nutanix.com/#/page/static/supportTools.

5.2 Citrix XenDesktop the Nutanix Way

Nutanix’s modular web-scale architecture lets you start small and then expand to meet increasing demand—a node, a block, or multiple blocks at a time—with no impact on performance. This design removes the hurdle of a large initial infrastructure purchase, generating faster time-to-value for your XenDesktop implementation.

The figure below presents a typical XenDesktop/XenApp deployment on Nutanix:
Running Citrix XenDesktop on Nutanix enables you to run multiple workloads all on the same, scalable converged infrastructure while achieving these benefits:

- **Modular incremental scale:** With the Nutanix solution you can start small and scale up. A single Nutanix block provides 10’s of TB storage and over 400 virtual desktops in a compact 2U footprint. Given the modularity of the solution, you can granularly scale by node; by block; or with multiple blocks, giving you the ability to accurately match supply with demand and minimize the upfront CapEx.

- **High performance:** By using system memory caching for read I/O and flash storage for read and write I/O, you can deliver high performance throughput in a compact form factor.
• **Change management**: Maintain environmental control and separation between development, test, staging, and production environments. Snapshots and fast clones can help in sharing production data with non-production jobs, without requiring full copies and unnecessary data duplication.

• **Business continuity and data protection**: User data and desktops are mission critical and need enterprise-grade data management features, including backup and DR.

• **Data efficiency**: Acropolis Distributed Storage Fabric offers both compression and deduplication to help reduce the storage footprint. The compression functionality is truly VM-centric. Unlike traditional solutions that perform compression mainly at the LUN level, the Nutanix solution provides all of these capabilities at the VM and file level, greatly increasing efficiency and simplicity. These capabilities ensure the highest possible compression and decompression performance on a sub-block level. By allowing for both inline and post-process compression capabilities, the Nutanix solution breaks the bounds set by traditional compression solutions.

• **Enterprise-grade cluster management**: A simplified and intuitive Apple-like approach to managing large clusters, including a converged GUI that serves as a central point for servers and storage, alert notifications, and the bonjour mechanism to auto-detect new nodes in the cluster. As a result, you can spend more time enhancing your environment rather than maintaining it.

• **High-density architecture**: Nutanix uses an advanced server architecture that, when using the NX-3000 series as an example, can house eight Intel CPUs (up to 96 cores) and up to 3 TB of memory in a single 2U appliance. Coupled with data archiving and compression, Nutanix can reduce desktop hardware footprints by up to 5x.

6 **Solution Design**

In the following section, we cover the design decisions and rationale for the XenDesktop deployments on the Nutanix Complete Cluster.

<table>
<thead>
<tr>
<th>Item</th>
<th>Detail</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Version</td>
<td>Citrix XenDesktop 7.6 FP3 (7.6.200)</td>
<td></td>
</tr>
<tr>
<td>Software Version</td>
<td>Acropolis Base Software 4.5.1</td>
<td></td>
</tr>
<tr>
<td>Minimum Size</td>
<td>4 Nutanix XCP hosts running AHV (NX-3000 – 1 block)</td>
<td>Minimum size requirement</td>
</tr>
<tr>
<td>Scale Approach</td>
<td>Incremental modular scale</td>
<td>Allow for growth from PoC (hundreds of desktops) to massive scale (thousands of desktops)</td>
</tr>
<tr>
<td>Item</td>
<td>Detail</td>
<td>Rationale</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Scale Unit</td>
<td>Node(s), Block(s), or Pod(s)</td>
<td>Granular scale to precisely meet the capacity demands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scale in n x node increments</td>
</tr>
<tr>
<td><strong>Nutanix AHV</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Size</td>
<td>As many as 12-32 hosts (starting with a minimum of 3 hosts)</td>
<td>Isolated fault domains best practice</td>
</tr>
<tr>
<td>Datastore(s)</td>
<td>1 x Nutanix DSF datastore per pod (XenDesktop Server VMs, VM clones, etc.)</td>
<td>Nutanix handles I/O distribution/localization n-Controller model</td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>Small deployments: Shared cluster</td>
<td>Dedicated infrastructure cluster for larger deployments (best practice)</td>
</tr>
<tr>
<td></td>
<td>Large deployments: Dedicated cluster</td>
<td></td>
</tr>
<tr>
<td><strong>Nutanix XCP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Size</td>
<td>As many as 32 nodes</td>
<td>Isolated fault domains best practices</td>
</tr>
<tr>
<td>Storage Pool(s)</td>
<td>1 x Storage Pool (SSD, SATA SSD, SATA HDD)</td>
<td>Standard practice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intelligent tiering handles data locality</td>
</tr>
<tr>
<td>Container(s)</td>
<td>1 x Container for VMs</td>
<td>Standard practice</td>
</tr>
<tr>
<td></td>
<td>1 x Container for Data (not used here)</td>
<td></td>
</tr>
<tr>
<td>Features/Enhancements</td>
<td>Increase CVM Memory to 32 GB</td>
<td>Deduplication needs 32 GB of RAM to be enabled.</td>
</tr>
<tr>
<td></td>
<td>Turn on deduplication</td>
<td></td>
</tr>
<tr>
<td><strong>Citrix XenDesktop</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XenDesktop Controllers</td>
<td>Min: 2 (n+1)</td>
<td>HA for XenDesktop Controllers</td>
</tr>
<tr>
<td></td>
<td>Scale: 1 per additional pod</td>
<td></td>
</tr>
<tr>
<td>Users per Controller</td>
<td>Up to 5,000 users</td>
<td>XenDesktop best practice</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>Citrix NetScaler (including NetScaler VPX – AHV is Citrix Ready)</td>
<td>Ensures availability of controllers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Balances load between controllers and pods</td>
</tr>
</tbody>
</table>
## Desktop Power Management

**Detail**
Nutanix plug-in for controlling desktop power

**Rationale**
Allows clean reboot and power down

## Citrix Storefront

### Storefront Servers

**Detail**
Min: 2 (n+1)

**Rationale**
HA for Storefront servers

## Load Balancing

**Detail**
Citrix NetScaler (including NetScaler VPX – AHV is Citrix Ready)

**Rationale**
Ensures availability of StoreFront servers
Balances load between StoreFront servers

## Citrix NetScaler including NetScaler VPX (If used)

### NetScaler Servers

**Detail**
Min: 2

**Rationale**
HA for NetScaler (active/passive)

### Users per NetScaler Server

**Detail**
See product data sheet

**Rationale**
Varies per model

### Load Balancing

**Detail**
NetScaler HA

**Rationale**
Ensures availability of NetScaler servers
Balances load between NetScaler servers and pods

---

Table 2 shows a high-level summary of the pod design for Citrix XenDesktop on Nutanix XCP.

### Table 2: Pod Highlights

<table>
<thead>
<tr>
<th>Item</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control Pod</strong></td>
<td></td>
</tr>
<tr>
<td># of XenDesktop Controller(s)</td>
<td>2</td>
</tr>
<tr>
<td># of XenDesktop StoreFront Server(s)</td>
<td>2</td>
</tr>
<tr>
<td><strong>Services Pod</strong></td>
<td></td>
</tr>
<tr>
<td># of AHV Hosts</td>
<td>Up to 32</td>
</tr>
<tr>
<td># of Nutanix Cluster(s)</td>
<td>1</td>
</tr>
<tr>
<td># of Datastore(s)</td>
<td>1</td>
</tr>
</tbody>
</table>

---

© 2016 Juniper Networks, Inc.
The section below describes desktop sizing considerations for hosted virtual and streamed desktops.

The following are examples of typical scenarios for desktop deployment and use (drawn from Login VSI definitions).

### Table 3: Desktop Scenarios Definition

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Workers</td>
<td>Task workers and administrative workers perform repetitive tasks within a small set of applications, usually at a stationary computer. The applications are usually not as CPU- and memory-intensive as the applications used by knowledge workers. Task workers who work specific shifts might all login to their virtual desktops at the same time. Task workers include call center analysts, retail employees, and warehouse workers.</td>
</tr>
<tr>
<td>Knowledge Workers</td>
<td>Knowledge workers’ daily tasks include accessing the Internet, using email, and creating complex documents, presentations, and spreadsheets. Knowledge</td>
</tr>
</tbody>
</table>
workers include accountants, sales managers, and marketing research analysts.

Power Users

Power users include application developers and people who use graphics-intensive applications.

Table 4 proposes initial sizing recommendations for a Windows 7 desktop for these typical scenarios. **Note**: These are recommendations and should be modified after a current state analysis.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>vCPU</th>
<th>Memory</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Workers</td>
<td>1</td>
<td>1.5 GB</td>
<td>35 GB (OS)</td>
</tr>
<tr>
<td>Knowledge Workers</td>
<td>2</td>
<td>2 GB</td>
<td>35 GB (OS)</td>
</tr>
<tr>
<td>Power Users</td>
<td>2</td>
<td>4 GB</td>
<td>35 GB+ (OS)</td>
</tr>
</tbody>
</table>

### 6.1 Desktop Optimizations

We generated our design with the following high-level desktop guidelines in mind:

- Size desktops appropriately for each particular use case.
- Use a mix of applications installed in gold images and application virtualization, depending on the scenario.
- Disable unnecessary OS services and applications.

6.2 XenDesktop on AHV

Using a base, or “Master VM,” the script framework creates full clone VMs that duplicate the base image, which provides both data locality and persistent desktops.

Figure 26 shows the main architectural components of a deployment on Nutanix and the communication path between services.

![Figure 26: Communication Path]
6.2.1 AHV Pod Design

Table 5 contains highlights from a high-level snapshot of the Citrix XenDesktop on Nutanix Hosted Virtual Desktop Pod.

Table 5: AHV Pod Detail

<table>
<thead>
<tr>
<th>Item</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Pod</td>
<td></td>
</tr>
<tr>
<td># of XenDesktop Controller(s)</td>
<td>2</td>
</tr>
<tr>
<td># of XenDesktop StoreFront Server(s)</td>
<td>2</td>
</tr>
<tr>
<td>Services Pod</td>
<td></td>
</tr>
<tr>
<td># of AHV Hosts</td>
<td>Up to 32</td>
</tr>
<tr>
<td># of Nutanix Cluster(s)</td>
<td>1</td>
</tr>
<tr>
<td># of Datastore(s)</td>
<td>1</td>
</tr>
<tr>
<td># of Desktops</td>
<td>Up to 3,840</td>
</tr>
</tbody>
</table>

6.2.2 Hosted Virtual Desktop I/O Path

The following figure describes the high-level I/O path for an AHV-based desktop on Nutanix. As shown, the Nutanix Distributed Storage Fabric (DSF) handles I/O operations, all of which occur on the local node to provide the highest possible I/O performance.
The following figure describes the detailed I/O path for an AHV-based desktop on Nutanix. All write I/O occurs locally on the local node’s SSD tier to provide optimal performance. The reads are served from the high-performance read cache (if cached) or from the SSD tier. Each node also caches frequently accessed data in the read cache for any local data. Nutanix ILM places data in the appropriate tier by continually monitoring data and I/O patterns.

6.3 Nutanix: Compute and Storage

The Nutanix Xtreme Computing Platform provides an ideal combination of both high-performance compute with localized storage to meet any demand. True to this capability, this reference architecture contains no reconfiguration or customization to the Nutanix product to optimize for this use case.

The following figure shows a high-level example of the relationship between a Nutanix block, node, storage pool, and container.
Table 6 includes the Nutanix storage pool and container configuration.

Table 6: Nutanix Storage Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Role</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP01</td>
<td>Main storage pool for all data</td>
<td>SSD and HDD</td>
</tr>
<tr>
<td>CTR-RF2-VM-01</td>
<td>Container for all VMs</td>
<td>AHV – Datastore</td>
</tr>
<tr>
<td>CTR-RF2-DATA-01</td>
<td>Container for all data (not used here)</td>
<td>AHV – Datastore</td>
</tr>
</tbody>
</table>

### 6.4 Network

Designed for true linear scaling, Juniper Networks and Nutanix uses a plug-and-play Ethernet Fabric network architecture called Virtual Chassis Fabric. The Juniper Networks Virtual Chassis Fabric architecture is based on a spine and leaf topology which provided non-blocking performance and predictable low latency; all Nutanix clusters are only a single hop away from any other cluster in the network. Juniper Networks Virtual Chassis Fabric allows you to manage the entire network as a single logical device. Adding additional network capacity simply requires cabling up a new leaf switch and powering it up. All software and configuration is automatically installed on the new switch so it seamlessly joins the Ethernet Fabric. Another benefit of Juniper Networks Virtual Chassis Fabric is that leaf switches do not require an inter-chassis link (ICL) and these ports can be used for additional Nutanix clusters instead.

Figure 29 shows how Juniper Networks Virtual Chassis Fabric provides 1,344x10GbE interfaces for Nutanix clusters. Each Nutanix appliance is multi-homed to a pair of leaf switches using Layer 2 which operates as all-active and non-blocking. Each leaf switch has an all-active 4-way connection into the spine for both Layer 2, Layer 3, and multicast network access.

Juniper Networks Junos Space Network Director is a powerful graphical user interface that allows you to manage all of your Juniper Networks switches, including Juniper Networks Virtual Chassis Fabric. You may add new ports, assign VLAN membership, configure routing protocols, and much more. Network Director also collects and graphs network activity such as traffic bandwidth, switch capacity, and link congestion.
7 Solution Application

This section applies the recommended pod-based reference architecture to real-world scenarios and outlines the sizing metrics and components.

**Note**: We provide detailed hardware configuration and product models in the appendix.

7.1 Port Configuration

When you multi-home the Nutanix 10GbE ports into the Juniper Networks Virtual Chassis Fabric, each of the network ports should be configured as a standard untagged or tagged Layer 2 port, depending how many VLANs you want to carry on the Nutanix appliance. The AHV will be setup with balance-slb into the Juniper Networks Virtual Chassis Fabric, and load balance traffic evenly.
7.2 Scenario: 480 Desktops

Table 7: Detailed Component Breakdown: 480 Desktops using NX-3060-G4

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>1 (partial)</td>
<td># of AHV Hosts</td>
<td>4</td>
</tr>
<tr>
<td># of Nutanix chassis (blocks)</td>
<td>1</td>
<td># of Datastore(s)</td>
<td>1</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 31: Rack Layout: 480 Desktops
### 7.3 Scenario: 960 Desktops

Table 8: Detailed Component Breakdown: 960 Desktops using NX-3060-G4

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>1 (partial)</td>
<td># of AHV Hosts</td>
<td>8</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>2</td>
<td># of Datastore(s)</td>
<td>1</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 32: Rack Layout: 960 Desktops
### 7.4 Scenario: 1,920 Desktops

#### Table 9: Detailed Component Breakdown: 1,920 Desktops using NX-3060-G4

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>1</td>
<td># of AHV Hosts</td>
<td>16</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>4</td>
<td># of Datastore(s)</td>
<td>1</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 33: Rack Layout: 1,920 Desktops*
### 7.5 Scenario: 3,840 Desktops

**Table 10: Detailed Component Breakdown: 3,840 Desktops using NX-3060-G4**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>2</td>
<td># of AHV Hosts</td>
<td>32</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>8</td>
<td># of Datastore(s)</td>
<td>2</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 34: Rack Layout: 3,840 Desktops*
### 7.6 Scenario: 7,680 Desktops

Table 11: Detailed Component Breakdown: 7,680 Desktops using NX-3060-G4

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
<td></td>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>4</td>
<td># of AHV Hosts</td>
<td>64</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>16</td>
<td># of Datastore(s)</td>
<td>4</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 35: Rack Layout: 7,680 Desktops
### 7.7 Scenario: 15,360 Desktops

**Table 12: Detailed Component Breakdown: 15,360 Desktops using NX-3060-G4**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
<td></td>
<td>Infrastructure</td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>8</td>
<td># of AHV Hosts</td>
<td>128</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>32</td>
<td># of Datastore(s)</td>
<td>8</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 36: Rack Layout: 15,360 Desktops*
7.8 Scenario: 30,720 Desktops

Table 13: Detailed Component Breakdown: 30,720 Desktops using NX-3060-G4

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
<td></td>
<td>Infrastructure</td>
<td></td>
</tr>
<tr>
<td># of Nutanix Desktop Pods</td>
<td>16</td>
<td># of AHV Hosts</td>
<td>256</td>
</tr>
<tr>
<td># of Nutanix Blocks</td>
<td>64</td>
<td># of datastore(s)</td>
<td>16</td>
</tr>
<tr>
<td># of RU (Nutanix)</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 10 GbE Ports</td>
<td>512</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 100/1000 Ports (IPMI)</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Leaf Switches</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Juniper Spine Switches</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 37: Rack Layout: 30,720 Desktops
8 Validation and Benchmarking

We conducted the solution and testing in this document with Citrix XenDesktop 7.6 FP3 deployed on the Nutanix Acropolis Hypervisor and the Nutanix Xtreme Computing Platform. Login VSI Office Worker and Knowledge Worker benchmarks supplied the model for the Knowledge Worker user on the Nutanix appliance.

8.1 Environment Overview

We used one node of an existing Nutanix NX-3460-G4 to host all infrastructure and XenDesktop services, as well as the Login VSI test harness. The three remaining nodes in the Nutanix NX-3460-G4 functioned as the target environment and provided all desktop hosting. The Nutanix block was connected to a Juniper QFX5100-48S top-of-rack switch using 10 GbE.

8.2 Test Environment Configuration

Assumptions:

- Knowledge worker use case
- Per-desktop IOPS (Knowledge Worker): 10 sustained/70 peak (startup)
- Using full clones

Hardware:
• Storage and Compute: 1 Nutanix NX-3460-G4
• Network: Juniper QFX5100-24Q spine switches and QFX5100-48S leaf switches

Desktop Configuration:
• OS: Windows 7 SP1 x64
• 2 vCPU and 2 GB memory
• 1 x 35 GB OS Disk
• Applications:
  o Microsoft Office 2013
  o Adobe Acrobat Reader XI
  o Internet Explorer
  o Flash video

Login VSI:
• Login VSI 4.1 Professional
8.3 XenDesktop Configuration

Table 14 shows the XenDesktop configuration used in the test environment.

<table>
<thead>
<tr>
<th>VM</th>
<th>Qty</th>
<th>vCPU</th>
<th>Memory</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>XenDesktop Controller(s)</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>1 x 40 GB (OS)</td>
</tr>
<tr>
<td>StoreFront Server(s)</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1 x 40 GB (OS)</td>
</tr>
</tbody>
</table>

8.4 Test Image Preparation

1. Create base VM.
2. Install Windows 7.
3. Install standard software.
5. Add machine to domain.
6. Install Citrix VDA.
7. Install Login VSI components.
8. Create snapshot.
8.5 Test Execution

1. Restart/turn on desktops
2. Restart/start Login VSI Launcher(s)
3. Log in to VSI Management Console
4. Set test parameters and number of sessions
5. Start test
6. Wait for test execution to finish
7. Analyze results (Login VSI)

8.6 Login VSI Benchmark

Login Virtual Session Indexer (Login VSI) is the de facto industry standard benchmarking tool for testing the performance and scalability of centralized Windows desktop environments such as Server-Based Computing (SBC) and Virtual Desktop Infrastructures (VDI).

Login VSI is 100 percent vendor independent and is used to test virtual desktop environments like Citrix XenDesktop and XenApp, Microsoft VDI and Remote Desktop Services, VMware View, or any other Windows-based SBC or VDI solution. It works with standardized user workloads and thus conclusions based on Login VSI test data are objective, verifiable, and replicable.

For more information about Login VSI visit [http://www.loginvsi.com/](http://www.loginvsi.com/).

The following table includes all four workloads available on Login VSI 4.1.

<table>
<thead>
<tr>
<th>Task Worker</th>
<th>Office Worker</th>
<th>Knowledge Worker</th>
<th>Power User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>Medium</td>
<td>Medium</td>
<td>Heavy</td>
</tr>
<tr>
<td>1 vCPU</td>
<td>1 vCPU</td>
<td>2 vCPUs</td>
<td>2-4 vCPUs</td>
</tr>
</tbody>
</table>
2-3 Apps | 4-6 Apps | 4-7 Apps | 5-9 Apps
---|---|---|---
No video | 240p video | 360p video | 720p video

Login VSI Workflows

The Login VSI Workflow base layout is captured in the *Login VSI 4.1 Workloads* document, which also documents the changes from previous versions of Login VSI to version 4.1 in great detail.

<table>
<thead>
<tr>
<th>Workload Name</th>
<th>Light</th>
<th>Medium</th>
<th>Heavy</th>
<th>Task Worker</th>
<th>Office Worker</th>
<th>Knowledge Worker</th>
<th>Power User</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSI Version</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4.1</td>
<td>4.1</td>
<td>4.1</td>
<td>4.1</td>
</tr>
<tr>
<td>Apps Open</td>
<td>2</td>
<td>5-7</td>
<td>8-10</td>
<td>2-7</td>
<td>5-8</td>
<td>5-9</td>
<td>8-12</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>66%</td>
<td>99%</td>
<td>124%</td>
<td>70%</td>
<td>82%</td>
<td>100%</td>
<td>119%</td>
</tr>
<tr>
<td>Disk Reads</td>
<td>52%</td>
<td>93%</td>
<td>89%</td>
<td>79%</td>
<td>90%</td>
<td>100%</td>
<td>133%</td>
</tr>
<tr>
<td>Disk Writes</td>
<td>65%</td>
<td>97%</td>
<td>94%</td>
<td>77%</td>
<td>101%</td>
<td>100%</td>
<td>123%</td>
</tr>
<tr>
<td>IOPS</td>
<td>5.2</td>
<td>7.4</td>
<td>7</td>
<td>6</td>
<td>8.1</td>
<td>8.5</td>
<td>10.8</td>
</tr>
<tr>
<td>Memory</td>
<td>1 GB</td>
<td>1 GB</td>
<td>1 GB</td>
<td>1 GB</td>
<td>1.5 GB</td>
<td>1.5 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td>vCPU</td>
<td>1vCPU</td>
<td>2vCPU</td>
<td>2vCPU</td>
<td>1vCPU</td>
<td>1vCPU</td>
<td>2vCPU</td>
<td>2vCPU+</td>
</tr>
</tbody>
</table>

8.7 How to Interpret the Results

Login VSI

Login VSI is a test benchmark used to simulate real-world user workload on a desktop. These values represent the time it takes for an application or task to complete (launching Outlook, for example), and is not in addition to traditional desktop response times. These do not refer to the round trip time (RTT) for network I/O; rather, they refer to the total time to perform an action on the desktop.

During the test, all VMs are turned on and the workload is started on a new desktop every 30 seconds, until all sessions and workload are active.

We quantified the evaluation using the following metrics:

- Minimum Response: The minimum application response time.
- Average Response: The average application response time.
• Maximum Response: The maximum application response time.
• VSI Baseline: Average application response time of the first 15 sessions.
• VSI Index Average: The VSI index average is the average response time, dropping the highest and lowest two percent.
• VSI$max$: If reached, the maximum value of sessions launched before the VSI index average gets above the VSI baseline × 125 percent + 3,000 ms.

Based on user experience and industry standards, we recommend keeping ranges for these values below those stated in the table below.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value (ms)</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Response</td>
<td>&lt;1,000</td>
<td>Acceptable ideal response time</td>
</tr>
<tr>
<td>Average Response</td>
<td>&lt;2,000</td>
<td>Acceptable average response time</td>
</tr>
<tr>
<td>Maximum Response</td>
<td>&lt;3,000</td>
<td>Acceptable peak response time</td>
</tr>
<tr>
<td>VSI Baseline</td>
<td>&lt;1,000</td>
<td>Acceptable ideal response time</td>
</tr>
<tr>
<td>VSI Index Average</td>
<td>&lt;2,000</td>
<td>Acceptable average response time</td>
</tr>
</tbody>
</table>

**Login VSI Graphs**

The Login VSI graphs show the values defined in Table 17 during the launching of each desktop session. The following figure shows an example graph of the test data. The y-axis is the response time in milliseconds and the x-axis is the number of active sessions.
9 Results

9.1 360 Knowledge Worker Desktops

Login VSI Knowledge Worker Results

During the testing with 360 desktops, VSI\textsuperscript{max} was not reached, with a baseline of 1,762 ms and an average VSI\textsuperscript{index} of 4,079 ms.
Cluster Metrics

User sessions over time measured:

Average logon duration over time during the test, measured with Login VSI. The scale is from 0 to 11 seconds:

At the peak of the test execution, CPU utilization for the AHV hosts peaked at 96.52 percent and memory utilization peaked at approximately 84.63 percent.
Nutanix Metrics

IOPS peaked at approximately 18,788 during the high-volume startup period to refresh the desktops; peak during the test was 4,824.
Command latency peaked at ~4.7 ms during the tests:

![Figure 43: Command Latency Peak](image)

## 10 Conclusion

Our extensive testing of Citrix XenDesktop on Nutanix with Juniper Networks Virtual Chassis Fabric network revealed light I/O footprints on the Xtreme Computing Platform. Aggregate IOPS peaked at approximately 18,000, during the high-volume startup periods, and sustained IOPS ranged from 4,000-5,000. I/O latencies averaged less than 2 ms for read and less than 5 ms for write during peak load. Nutanix offers the performance and low latency necessary for uncompromised end-user experience, even during boot storms, patch and update operations, and application and OS upgrades.

The Citrix XenDesktop-on-Nutanix solution provides a single, high-density platform for desktop and application delivery. Running XenDesktop on Nutanix also means that you can

- Get started quickly, without onerous upfront expenditures, and then scale as you grow.
- Deliver excellent user density, supporting over 120 knowledge workers per AHV host.
- Simplify management and administration by eliminating LUNs.
- Significantly lower acquisition and ongoing costs by running VDI alongside other workloads.

The Juniper Networks Virtual Chassis Fabric network allowed Nutanix and Citrix to easily add additional VDI capacity without having to change the network architecture and advanced network features such as:

- Single point of network management for both command-line and GUI.
- Application traffic visibility to get end-to-end solution health.
- Adaptive Flowlet Splicing to break down large applications flows into smaller flowlets to ensure perfect hashing across all four spines in an all-active architecture.
- Plug-and-play for adding additional network switches to Juniper Networks Virtual Chassis Fabric.
11 Appendix: Configuration

Hardware

- Storage and Compute
  - Nutanix NX-3460-G4
- Per node specs (4 nodes per 2U block):
  - CPU: 2x Intel Xeon E5-2620v3
  - Memory: 512 GB Memory
- Network
  - Juniper Networks QFX5100-24Q – Juniper Networks Virtual Chassis Fabric Spine
  - Juniper Networks QFX5100-48S – Juniper Networks Virtual Chassis Fabric Leaf

Software

- Nutanix
  - Acropolis 4.5.0.2
- XenDesktop
  - 7.6 FP3
- Desktop
  - Windows 7 SP1 x64
- Infrastructure
  - AHV

VM

- Desktop

CPU: 2 vCPU

- Memory: 2 GB
- Storage: 1 x 35 GB OS Disk on CTR-RF2-VM-01 NDFS-backed DSF datastore
# 12 References

## 12.1 Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Information Lifecycle Management</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Overview of the Nutanix Architecture</td>
<td>4</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Data Locality and VM Migration</td>
<td>4</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Elastic Deduplication Engine</td>
<td>5</td>
</tr>
<tr>
<td>Figure 5</td>
<td>The Nutanix Architecture with Acropolis</td>
<td>6</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Acropolis and PCI Device Passthrough for Storage</td>
<td>7</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Acropolis and VM Networking</td>
<td>8</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Acropolis IPAM</td>
<td>9</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Multiclone Block Maps</td>
<td>10</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Juniper Networks Virtual Chassis Fabric Architecture</td>
<td>11</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Juniper Networks QFX5100-24Q Switch</td>
<td>12</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Juniper Networks QFX5100-48S Switch</td>
<td>12</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Virtual Chassis Fabric Spine and Leaf Architecture</td>
<td>13</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Virtual Chassis Fabric Unified Forwarding Tables</td>
<td>14</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Master and Backup Routing Engines in a Virtual Chassis Fabric</td>
<td>15</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Flowlets in a Virtual Chassis Fabric</td>
<td>16</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Simple Virtual Chassis Fabric Topology</td>
<td>18</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Cabling the First Leaf Switch</td>
<td>20</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Installing the Remaining Leaf Switch</td>
<td>21</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Installing the Last Spine Switch and Completing the Cabling</td>
<td>22</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Virtual Chassis Fabric Topology</td>
<td>25</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Virtual Chassis Fabric</td>
<td>26</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Full Clones on AHV</td>
<td>30</td>
</tr>
<tr>
<td>Figure 24</td>
<td>XenDesktop on Nutanix: Conceptual Architecture</td>
<td>31</td>
</tr>
<tr>
<td>Figure 25</td>
<td>XenDesktop Pod Overview</td>
<td>35</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Communication Path</td>
<td>37</td>
</tr>
</tbody>
</table>
Figure 27: AHV I/O Overview........................................................................................................................................................... 39
Figure 28: AHV I/O Detail ................................................................................................................................................................... 39
Figure 29: Nutanix Component Architecture........................................................................................................................................ 40
Figure 30: Leaf-Spine Network Architecture........................................................................................................................................ 41
Figure 31: Rack Layout: 480 Desktops ............................................................................................................................................... 42
Figure 32: Rack Layout: 960 Desktops ............................................................................................................................................... 43
Figure 33: Rack Layout: 1,920 Desktops ........................................................................................................................................ 44
Figure 34: Rack Layout: 3,840 Desktops ........................................................................................................................................ 45
Figure 35: Rack Layout: 7,680 Desktops ........................................................................................................................................ 46
Figure 36: Rack Layout: 15,360 Desktops ......................................................................................................................................... 47
Figure 37: Rack Layout: 30,720 Desktops ....................................................................................................................................... 48
Figure 38: Test Environment Overview ......................................................................................................................................... 49
Figure 39: Example Graph of Login VSI Test....................................................................................................................................... 55
Figure 40: Login VSI Knowledge Worker Results .................................................................................................................................. 56
Figure 41: Logon Duration over Time ........................................................................................................................................ 56
Figure 42: Peak CPU Utilization: ..................................................................................................................................................... 57
Figure 43: Command Latency Peak ................................................................................................................................................... 58

12.2 Table of Tables
Table 1: Solution Design Decisions ................................................................................................................................................ 32
Table 2: Pod Highlights ....................................................................................................................................................................... 34
Table 3: Desktop Scenarios Definition ........................................................................................................................................ 35
Table 4: Desktop Scenario Sizing .................................................................................................................................................... 36
Table 5: AHV Pod Detail ...................................................................................................................................................................... 38
Table 6: Nutanix Storage Configuration ........................................................................................................................................ 40
Table 7: Detailed Component Breakdown: 480 Desktops using NX-3060-G4...................................................................................... 42
Table 8: Detailed Component Breakdown: 960 Desktops using NX-3060-G4 .................................................................................... 43
Table 9: Detailed Component Breakdown: 1,920 Desktops using NX-3060-G4 ................................................................................ 44
Table 10: Detailed Component Breakdown: 3,840 Desktops using NX-3060-G4................................................................. 45
Table 11: Detailed Component Breakdown: 7,680 Desktops using NX-3060-G4................................................................. 46
Table 12: Detailed Component Breakdown: 15,360 Desktops using NX-3060-G4............................................................... 47
Table 13: Detailed Component Breakdown: 30,720 Desktops using NX-3060-G4............................................................... 48
Table 15: XenDesktop Configuration ............................................................................................................................................ 51
Table 16: Login VSI 4.1 Workloads ................................................................................................................................................ 52
Table 17: Login VSI Workload Definitions .................................................................................................................................. 53
Table 18: Login VSI Metric Values .................................................................................................................................................. 54

13 About the Authors

Kees Baggerman is a senior solution architect for End User Computing at Nutanix Inc. In his role, Kees develops methods for successfully implementing applications on the Nutanix platform. In addition, he delivers customer projects, including defining architectural, business, and technical requirements, creating designs, and implementing the Nutanix solution.

Before working with Nutanix, Kees’ main areas of work were migrations and implementations of Microsoft and Citrix infrastructures, writing functional/technical designs for Microsoft infrastructures, Microsoft Terminal Server or Citrix (Presentation Server/ XenApp, XenDesktop and NetScaler) in combination with RES Workspace Manager and RES Automation Manager.

Kees is a Citrix Certified Integration Architect, Microsoft Certified IT Professional, RES Certified Professional and RES Certified Trainer. RES Software also named him RES RSVP six consecutive years and Kees was honored as the RES Software Most Valuable Professional of 2011. As a demonstration of his passion for virtualization technology, Kees earned the title of VMware vExpert in 2013, 2014, and 2015. Citrix also named him a Citrix Technology Professional in 2015.

Follow Kees on Twitter at @kbaggerman.

Douglas Richard Hanks Jr. is a Director of Product Management with Juniper Networks and focuses on next-generation hardware and software solutions and corporate strategy. He works in the Juniper Development and Innovation (JDI) organization that is responsible for all Juniper Networks hardware, software, and solutions.

Douglas is certified with Juniper Networks as JNCIE-ENT #213 and JNCIE-SP #875. Douglas’ interests are network engineering and architecture for enterprise and service provider technologies. He is the author of the Juniper QFX1000 Series, Juniper QFX5100 and Juniper MX Series published by O’Reilly Media and several Day One books published by Juniper Networks Books.
About Nutanix

Nutanix delivers invisible infrastructure for next-generation enterprise computing, elevating IT to focus on the applications and services that power their business. The company’s software-driven Xtreme Computing Platform natively converges compute, virtualization and storage into a single solution to drive simplicity in the datacenter.

Get involved in the Nutanix Next community at http://next.nutanix.com and follow the Nutanix blogs at http://www.nutanix.com/blog/. Also follow Nutanix on Twitter at @Nutanix.

About Juniper Networks

At Juniper Networks, we believe that the network is the single greatest vehicle for knowledge, understanding, and human advancement that the world has ever known. The great task of delivering a new network for the next decade hinges on the creativity and commitment of our people. To achieve it requires a committed practice that we call the Juniper Way.