IBM Redbooks

Implementing and Managing a High-performance Enterprise Infrastructure with Nutanix on IBM Power Systems

January 2020
Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (January 2020)

This edition applies to Version

RHEL 7.7
Nutanix v20170331.78
MongoDB v4.2
IBM AIX v7.2 SP3
Ubuntu Cloud Bionic 18.04 LTS
Ubuntu Cloud Xenial 16.04 LTS

© Copyright International Business Machines Corporation 2020. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Notices ........................................................................................................................................ vii
Trademarks ................................................................................................................................... viii

Preface ........................................................................................................................................... ix
Authors .......................................................................................................................................... ix
Now you can become a published author, too! ................................................................................ xi
Comments welcome .................................................................................................................... xi
Stay connected to IBM Redbooks ................................................................................................ xi

Chapter 1. Introduction to Nutanix and Hyperconverged computing for cloud native applications and cognitive workloads .......................................................... 1
  1.1 Hyperconverged infrastructure .............................................................................................. 2
  1.2 What is a Hyperconverged infrastructure? .......................................................................... 3
  1.3 Nutanix on IBM Power Systems .......................................................................................... 3
  1.4 IBM Power Systems Architecture for Nutanix ................................................................... 5

Chapter 2. Why a Web-scale Hyperconverged Infrastructure with Nutanix on IBM Power Systems? ....................................................................................... 7
  2.1 Why the combination of IBM Power Systems and Nutanix can be interesting? ............... 8
    2.1.1 Why IBM? ....................................................................................................................... 8
    2.1.2 Why Nutanix? ................................................................................................................ 8
  2.2 Simplify your data center with IBM Hyperconverged Power Systems with Nutanix ....... 8
    2.2.1 Focus on the applications, not the infrastructure ......................................................... 9
    2.2.2 Eliminate bottlenecks ..................................................................................................... 9
    2.2.3 Ease of DevOps ............................................................................................................. 10
    2.2.4 Increase security without adding silos. ........................................................................ 10

Chapter 3. System configuration, reference architecture, systems management for Nutanix on IBM Power Systems ................................................................. 11
  3.1 Starting Nutanix on IBM Power Systems ........................................................................... 12
    3.1.1 Available configurations ............................................................................................... 12
  3.2 Build Blocks - Nutanix on IBM Power Systems ................................................................ 12
  3.3 Nutanix web console - Prism Central .................................................................................. 14
  3.4 Verifying and preparing the environment for deployment ................................................. 17
  3.5 Deploying a virtual machine (VM) ..................................................................................... 25

Chapter 4. Configuration and management recommended practices ..................................... 37
  4.1 Nutanix disaster recovery .................................................................................................... 38
    4.1.1 Implementing disaster recovery ..................................................................................... 38
  4.2 Configuring a protection domain (async DR) ...................................................................... 43
    4.2.1 General guidelines ........................................................................................................ 44

Chapter 5. Nutanix use cases ...................................................................................................... 49
  5.1 Installing packages in Nutanix ............................................................................................. 50
  5.2 Introduction to MongoDB .................................................................................................... 50
    5.2.1 The document-based query language ........................................................................... 50
  5.3 Installation and configuration of a MongoDB Shard in a Nutanix Cloud ......................... 50
    5.3.1 MongoDB and Nutanix ................................................................................................. 51
    5.3.2 Reference architecture ................................................................................................. 52
    5.3.3 Installing MongoDB on Nutanix ................................................................................... 52
5.3.4 Configuring a MongoDB Replica Set ........................................ 54
5.3.5 Recommended practices and conclusions ................................ 57
5.4 The Nutanix Command Line Interface (CLI) ............................. 59
5.5 The Nutanix REST API .......................................................... 62

Chapter 6. Deployment scenarios .................................................... 67
6.1 Overview ................................................................................. 68
6.2 How do you deploy AIX using a Cloud Ready Image by way of Prism ......................................................... 68
6.3 Router VM ............................................................................. 74
6.4 OpenVPN .............................................................................. 79
   6.4.1 Configure an OpenVPN VM ............................................. 79

Related publications ........................................................................ 87
IBM Redbooks ............................................................................. 87
Online resources ........................................................................... 87
Help from IBM ............................................................................ 87
Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

- AIX®
- Cognitive Era®
- DB2®
- IBM®
- IBM Cloud™
- IBM Services™
- Passport Advantage®
- POWER®
- POWER8®
- POWER9™
- PowerVM®
- Redbooks®
- Redbooks (logo) ®
- WebSphere®
- XIV®
- Evolution, are trademarks or registered trademarks of Kenexa, an IBM Company.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Ansible, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.
Preface

This IBM® Redbooks® publication describes how to implement and manage a hyperconverged private cloud solution using theoretical knowledge, hands-on exercises, and documenting the findings by way of sample scenarios. Moreover this book is a guide on how to implement and manage a high-performance enterprise infrastructure and private cloud platform for big data, artificial intelligence, transactional and analytics workloads on Power Systems.

This book utilizes available documentation, hardware, and software resources to:

- Document the web-scale architecture demonstrating simplicity and agility public clouds.
- Showcase the hyperconverged infrastructure to help cloud native applications mine cognitive analytics workloads.
- Conduct and document implementation case studies.
- Document guidelines to help provide an optimal system configuration, implementation, and management.

This publication addresses topics for developers, IT architects, IT specialists, sellers and anyone looking to implement and manage a high-performance enterprise infrastructure and private cloud platform on IBM Power Systems. Moreover, this book provides documentation to transfer the how-to-skills to the technical teams, and solution guidance to the sales team. This book compliments any documentation available at the IBM Knowledge Center, and also aligns with the educational materials provided by the IBM Systems Software Education (SSE).

Authors

This book was produced by a team of specialists from around the world working at IBM Redbooks, Poughkeepsie Center.

**Dino Quintero** is an IT Management Consultant and an IBM Level 3 Senior Certified IT Specialist with IBM Redbooks in Poughkeepsie, New York. He has 24 years of experience with IBM Power Systems technologies and solutions. Dino shares his technical computing passion and expertise by leading teams developing technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing, cloud computing, artificial intelligence including machine and deep learning, and cognitive solutions. He also is a Certified Open Group Distinguished IT Specialist. Dino is formerly from the province of Chiriqui in Panama. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

**Ricardo Dobelin Barros** is an IBM Certified IT Specialist and IT Architect in Brazil. Ricardo joined IBM in 2002 and has a total of 20 years of experience in the IT industry. He works in the Sales Solutions group for IBM Services™, as pre-sales Technical Solution Architect. Ricardo has expertise in availability management, service-levels management, sizing, and performance tuning in operating systems. He also has experience in capacity planning, modelling, measurement, automation, virtualization, multicloud and infrastructure management for many IBM clients in Brazil. Ricardo holds a bachelor degree in systems
analysis, post-graduate in Teaching Degree in Higher Education. He also has Master of Business Administration in Strategic Management and Business Planning.

**Miguel Gomez Gonzalez** is an IBM Power Systems Integration engineer based in Mexico with over 11 years experience in Linux and IBM Power Systems technologies. He has extensive experience in implementing several IBM Linux and AIX® clustering solutions. His areas of expertise are Power Systems performance boost, virtualization, high availability, system administration, and test design. Miguel holds a Master in Information Technology Management from ITESM.

**Luis Daniel Gonzalez Diaz** is a System Administrator and an Automation Engineer on Power Systems in IBM Guadalajara, México. He holds a B.Sc. degree in Cyber Security and Networking Engineering. He joined IBM in 2018 as an automation engineer helping out automating Power Systems installation and configurations. He has experiences configuring Linux systems and services for 5 years.

**Slav Martinski** works in IBM since 2015 as a Remote Technical Support agent in the hardware support division for Australia, New Zealand, South Africa and the EMEA markets. In 2019 Slav joined the HMC project and works as an SME on cases raised by customers with HMC and FSP related issues. Slav obtained his first degree in Computer Science from Delaware County Community College followed by a Bachelor of Science in Finance from the University of Tennessee in Knoxville, USA and a Master's degree in Economics from the University of Sofia in Bulgaria.

**Gabriel Padilla Jimenez** is a Test Architect for Hardware on Power System. He is involved in early faces of most of the hardware general availability working closely with development to achieve best quality products. Gabriel previously worked on manufacturing lines at IBM Mexico as Test Lead for XIV® Storage and Power Systems. Gabriel has a Masters degree in Information Technology and a Bachelors of Science in Electronic Engineering.

**Ismael Solis Moreno** is a data scientist and performance analyst for IBM at Mexico Software Lab. He got his Masters from the National Center of Research and Technological Development in Mexico and his PhD from the University of Leeds in the UK. Ismael has participated as technical leader in different projects related to big data analytics and machine learning within the IBM and other companies. These include, the University of Leeds, The UK Datacenter Alliance and Apollo MIS researching predictive algorithms for Google, Alibaba and the British Premier League. Ismael has about 15 international publications in prestigious computing science journals. He has participated as speaker in over 25 international conferences related to data science and machine learning, and co-authored patents to improve datacenter energy efficiency by exploiting big data. Ismael has collaborated as researcher in the field of data science at the University of Leeds in the UK, The National University of Defense and Technology and The University of Aeronautics and Astronautics in China. In Mexico, he has collaborated with The National Institute of Electricity and Clean Energy and The National Center of Advance Technology creating solutions for reducing energy consumption in smart cities. Ismael’s current work at IBM is focused on developing machine learning mechanisms to improve the performance of large distributed storage systems by analyzing big data.

**Alan Verdugo Munoz** is a software engineer who currently works as a Python developer for the Sales and Marketing group of the CIO in IBM Mexico. He has over 10 years of experience writing code professionally and 15 years of experience playing with GNU/Linux systems. He is a passionate advocate for the Free and Open Source movements. He currently holds CompTIA Linux+ and LPIC-1 certifications.

Thanks to the following people for their contributions to this project:
Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online Contact us review Redbooks form found at: ibm.com/redbooks
- Send your comments in an email to: redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook: http://www.facebook.com/IBMRedbooks
- Follow us on Twitter: http://twitter.com/ibmredbooks
► Look for us on LinkedIn:
   http://www.linkedin.com/groups?home=&gid=2130806

► Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

► Stay current on recent Redbooks publications with RSS Feeds:
   http://www.redbooks.ibm.com/rss.html
Introduction to Nutanix and Hyperconverged computing for cloud native applications and cognitive workloads

This chapter contains a brief history about hyperconverged infrastructures, their reference architecture, and how the solution has evolved to provide a Cloud native environment for applications.

This chapter includes the following sections:

- Hyperconverged infrastructure
- What is a Hyperconverged infrastructure?
- Nutanix on IBM Power Systems
- IBM Power Systems Architecture for Nutanix
1.1 Hyperconverged infrastructure

How does this infrastructure started? Technology evolves rapidly by nature, hence at times one misses developments, and cannot catch immediately the rapidly evolution of technology. This section provides a description of how technology has evolved incorporating traditional technologies to bring to you what hyperconverged is today. Notice that some of these traditional technologies are still in use, changing and reinventing themselves, but these are still the basic components of hyperconverged. Figure 1-1 shows a general idea of the evolution of data center technologies, and how these have become what you know today as a Cloud environment.

Figure 1-1  Evolution® of data center technologies

This section provides some of the characteristics for each of the data center technologies shown in Figure 1-1.

Mainframe characteristics:
- Natively converged CPU, main memory, and storage
- Engineered internal redundancy

Stand-alone servers characteristics:
- CPU, main memory, and direct-attached storage (DAS)
- More flexibility than the mainframe
- Low cost than the mainframe
- Accessed over the network

Centralized storage characteristics:
- Pooled storage resources led to better storage utilization
- Centralized data protection, RAID eliminated the chance that server loss caused data loss
- Storage were performed over the network

Virtualized environment characteristics:
- Abstracting the operating system from hardware (VM)
- Workload consolidation
- Clustering led to pooled compute resources
- Ability to dynamically migrate workloads between compute nodes (DRS and vMotion)
1.2 What is a Hyperconverged infrastructure?

Hyperconverged infrastructure is a category of scale-out software-integrated infrastructure that applies a modular approach to compute, network and storage on standard hardware, leveraging distributed, and horizontal building blocks under unified management.¹

These are the main characteristics of a hyperconverged infrastructure:
- Environment scale-out
- Network and storage on standard hardware
- Software-defined-environment (storage, network, and server CPU and memory)

¹ https://www.gartner.com/reviews/market/hyperconverged-infrastructure

1.3 Nutanix on IBM Power Systems

The agility and simplicity of public cloud capabilities can be delivered in your data center to help you run your mission critical workloads. This is the goal and the role of IBM Hyperconverged Power Systems with Nutanix from the start. These 1U and 2U IBM POWER®-based appliances (CS821 and CS822 models) are currently the only hyperconverged solution to combine Power Systems performance with simplicity of the Nutanix Enterprise Cloud Platform software solution. Nutanix has been a prominent solution in the hyperconverged private cloud space.

Hyperconvergence is a type of infrastructure system with a software-centric architecture that tightly integrates compute, storage, storage networking and virtualization resources and other

Note: The vast majority of the characteristics cited in this section are also found in other platforms. Although this section does not show a time line, it shows how the technologies matured to form the hyperconverged solution today. Moreover all these technical characteristics are also found today in hybrid cloud environments.

Note: For more details about cloud, multicloud, hybrid cloud, containers, Kubernetes and other related topics on IBM Power Systems, refer to Red Hat OpenShift and IBM Cloud™ Paks on IBM Power Systems: Volume 1, SG24-8459.
Implementing and Managing a High-performance Enterprise Infrastructure with Nutanix on IBM Power Systems

Technologies pre-integrated in a scale-out server. Individual servers (called nodes) are clustered together in a scale-out topology managed by the Nutanix software stack. Internal physical storage resources from each node are pooled into one large, virtualized distributed file system. Additional servers can easily be added to increase compute or storage capacity dynamically with simplicity.

Scale-out Linux workloads such as IBM WebSphere® Application Server (WAS), NGINX, and NoSQL or open source databases (EDB Postgres and MongoDB) are a good fit for IBM Hyperconverged Power Systems with Nutanix as shown in Chapter 5, “Nutanix use cases” on page 49. This solution delivers the flexibility to run a variety of virtualized Linux applications using Nutanix’s built-in Nutanix Acropolis Hypervisor (AHV).

Also running AIX 7.2 on IBM Hyperconverged Power Systems with Nutanix enables the system to run AIX applications such as IBM WebSphere Application Server and IBM DB2® on a hyperconverged private cloud. Refer to section 6.2, “How do you deploy AIX using a Cloud Ready Image by way of Prism” on page 68 which shows how you can deploy an AIX image using Nutanix. With Nutanix providing a virtualized scale-out cloud environment, clients can experience the simplicity of a public cloud, although with the security and control of an on-premises solution.

IBM Hyperconverged Power Systems with Nutanix can deliver simplified operations and significantly reduced costs as you can consolidate homegrown applications, development and test workloads, and complex middleware running on AIX or Linux on this hyperconverged private cloud fabric. Moreover, you can run AIX workloads side by side, not only with Linux on Power Systems, but even with x86 workloads running on other clusters, managed all from the single pane Nutanix Prism Central console. This eases the administrative burden on IT infrastructure staff, and simplifies the data center environment, although reducing the cost of operations by 60 percent compared to traditional infrastructure environments.

Prism simplifies infrastructure management with one-click operations. With its consumer-grade interface, Prism provides an end-to-end management solution for virtualized data center environments. It streamlines and automates common workflows and eliminates the need for multiple management solutions across data center operations. Powered by advanced machine learning technology, Prism analyzes system data to generate actionable insights, optimizing virtualization and infrastructure management. Chapter 3, “System configuration, reference architecture, systems management for Nutanix on IBM Power Systems” on page 11 and Chapter 4, “Configuration and management recommended practices” on page 37 presents more details about Prism.

Table 1-1 shows supported hardware configurations.

<table>
<thead>
<tr>
<th>Table 1-1 Supported hardware configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://www.afcea.org/signal/resources/content/IDC-TCO-Report-August2017.pdf" alt="" /></td>
</tr>
</tbody>
</table>
### 1.4 IBM Power Systems Architecture for Nutanix

The solution is deployed as a cluster of nodes. The solution distributes data and metadata across all the nodes in the cluster enabling to grow the cluster by adding nodes when needed. You can achieve linear predictable performance and capacity increases as the cluster grows.

Figure 1-2 on page 6 shows the hyperconverged system reference architecture which:

- Virtualizes and moves the controllers to the host
- Provides core services and logic through software
- Distributes (shards) data across all nodes in the system
- Moves the storage local to the compute

A Nutanix Controller VM runs on each node, hence creating the pooling of local storage from all nodes in the cluster.

<table>
<thead>
<tr>
<th></th>
<th><strong>Compute optimized</strong></th>
<th><strong>Storage optimized</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base platform</strong></td>
<td>S821LC</td>
<td>S822LC</td>
</tr>
<tr>
<td><strong>Server compute</strong></td>
<td>Two 10- core 2.09 GHz IBM POWER8® processors with eight hardware threads per core</td>
<td>Two 11- core 2.89 GHz POWER8® processors with eight hardware threads per core</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>128 GB or 256 GB per system using 16 x DDR4 DIMMs with total of 64 MB L4 cache</td>
<td>256 GB or 512 GB per system using 16 x DDR4 DIMMs with total of 64 MB L4 cache</td>
</tr>
<tr>
<td><strong>Network connection</strong></td>
<td>4 port 10 G BaseT Ethernet Add on: PCIe3 2- port 10 Gbe SFP+ Adapter, based on Intel X710 Chipset and Driver</td>
<td>8</td>
</tr>
<tr>
<td><strong>Boot device</strong></td>
<td>1x64 GB SATADOM</td>
<td></td>
</tr>
<tr>
<td><strong>SAS Controller</strong></td>
<td>PCIe Gen3 SAS Controller, based on LSI 3008L</td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>Samsung SM863a [480 GB, 960 GB or 1.92 TB]—all flash configuration</td>
<td>Samsung SM863a [480 GB, 960 GB, 1.92 TB or 3.8 TB]— all flash configuration</td>
</tr>
<tr>
<td><strong>Drive type</strong></td>
<td>4 x 2.5&quot; SSD</td>
<td>8 x 2.5&quot; SSD, 12 x 2.5&quot; SSD</td>
</tr>
<tr>
<td><strong>Hypervisor</strong></td>
<td>Nutanix AHV</td>
<td></td>
</tr>
<tr>
<td><strong>Software license options</strong></td>
<td>Acropolis Pro or Ultimate License Prism Starter or Pro License (includes Prism Central)</td>
<td></td>
</tr>
</tbody>
</table>

a. These models are still available to buy. IBM intends to continue supporting this solution on IBM POWER9™ processors.
The IBM Hyperconverged Power Systems with Nutanix solution is a converged storage and compute solution which leverages local components and creates a distributed platform for running workloads.

Each node runs an Acropolis Hypervisor Virtualization and the Nutanix Controller VM (CVM). The Nutanix CVM is what runs the Nutanix software and serves all of the I/O operations for the hypervisor and all VMs running on that host.

Figure 1-3 shows what a typical node logically looks like.
Why a Web-scale Hyperconverged Infrastructure with Nutanix on IBM Power Systems?

This chapter provides information to understand the reasons to implement Nutanix on a platform such as Power Systems. Although there are many options to deploy the Nutanix solution, the combination of Nutanix running on Power Systems is a flexible and scalable alternative to consider to create a private cloud. The implementation provides many benefits you find throughout this chapter.

This chapter contains the following sections:

- Why the combination of IBM Power Systems and Nutanix can be interesting?
- Simplify your data center with IBM Hyperconverged Power Systems with Nutanix
2.1 Why the combination of IBM Power Systems and Nutanix can be interesting?

IBM Power Systems are servers designed for mission-critical applications and emerging cognitive era workloads including artificial intelligence, machine learning, deep learning, advanced analytics, high performance computing, data lakes and operational datastores. Designed to deliver efficiency whether deployed in a private, public and hybrid cloud, Power Systems benefit from a wide range of open technologies, many deriving from the collaboration with OpenPOWER Foundation members. Customers today can enjoy benchmark setting performance for a wide variety of data-intensive workloads.

2.1.1 Why IBM?

IBM Hyperconverged Power Systems with Nutanix combines Nutanix Enterprise Cloud Platform software with Power Systems. The integrated solution targets your most demanding transactional and cognitive analytics workloads with an infrastructure that is easy to manage and simple to scale. IBM Power Servers\(^1\) deliver 80 percent more performance per dollar on average than x86-based\(^2\) servers. You can replace your existing infrastructure to transform your business, by putting your most data intensive workloads on these servers.

2.1.2 Why Nutanix?

Clients are adopting hyperconverged infrastructure solutions because these help organizations to simplify operations in the data center and to bring capital reductions in most use cases. Hyperconverged infrastructure increases IT agility and help organizations to adopt a public cloud approach to deploying and provisioning resources. Rather than building and managing storage, servers, and software components separately, everything is managed by a centralized management tool called Prism. This also eliminates the need for specialized IT skills to build and operate cloud-driven infrastructures. There is more information about Prism in Chapter 3, “System configuration, reference architecture, systems management for Nutanix on IBM Power Systems” on page 11.

2.2 Simplify your data center with IBM Hyperconverged Power Systems with Nutanix

Nutanix Enterprise Cloud Platform can transform any data center from an expensive and overcomplicated IT infrastructure, to an efficient, cost-effective virtualization environment, enabling organizations to successfully meet their business goals. With the combination of IBM Power Systems and Nutanix software, customers get a high-performance enterprise-class hardware platform and a top-rated on-premises cloud operating system to easily manage the infrastructure.

A trademark of the Nutanix offering is choice in supported processor architectures, hardware generations, operating systems, hypervisors and public cloud integration. By embracing these

---

\(^1\) CS821 and CS822 are based on Power S821LC and Power S822LC respectively. Actual performance and price-performance of the CS821 and CS822 can vary.

\(^2\) 80% price-performance advantage is based on the average of IBM internal measurements of Power System S822LC for Big Data relative to comparable x86 E5-2600v4 (Broadwell) 2-socket offerings across multiple open source databases including MongoDB, EnterpriseDB, and MariaDB. Comparisons utilize current pricing as of August 24, 2016. More details can be found at: [http://ibm.com/developerworks/linux/perfcol/index.html](http://ibm.com/developerworks/linux/perfcol/index.html)
wide array of choices, Nutanix helps customers’ minimize technical challenges, although keeping the management experience simple.

With IBM and Nutanix partnering together, customers gain the ability to manage both POWER-based clusters and x86-based clusters from a single point through the Nutanix Management Console Prism Central. In other words, one instance of the Nutanix Prism Central Management Console can manage an endless number of POWER-based and x86-based clusters in one single view, with one tool and without complexity. This eases the administration burden on IT infrastructure staff, which simplifies the data center infrastructure.

2.2.1 Focus on the applications, not the infrastructure

In general, the primary directive for IT is to provide application services that help the organization to fulfill its mission. Thus the more that the infrastructure can serve as a seamless on-demand resource that appears invisible, then the better for the business. The IBM Hyperconverged Power System with Nutanix provides this capability. For example, the IBM POWER architecture delivers performance and when combined with Nutanix, this enables data scientists to extract insights from the data.

The Power Systems solution provides:

- Better performance

More processor threads, larger cache and lower latency design mean faster throughput on transactions and queries.

- Remarkable price/performance

Run workloads such as EDB and MongoDB with lower total cost of ownership because you get the same work done with half the number of systems, floor space, power and cooling. Chapter 5, “Nutanix use cases” on page 49 shows more information about MongoDB running in Nutanix with IBM Power Systems.

- Unlimited scalability

Applications can run at any scale of total data, size of active data set or compute needed.

- Higher availability

Built-in self-healing, backup and disaster recovery capabilities provide better uptime for applications than traditional infrastructures.

2.2.2 Eliminate bottlenecks

Deployments can expand quickly as new users or workloads are added. By using IBM Hyperconverged Power Systems, you start small and scale out without the bottlenecks that occur with traditional architectures.

Administrators can scale existing IBM Hyperconverged clusters or deploy new clusters in minutes with less concern for compute, storage and network bottlenecks. Each additional node delivers predictable performance and, because of its distributed architecture, IBM Hyperconverged Power Systems with Nutanix clusters prevent one workload from starving another, allowing the infrastructure to be shared, if desired.

In concert with POWER performance, the Nutanix cloud operating system takes full advantage of server virtualization without the limitations of other solutions.
2.2.3 Ease of DevOps

IBM Hyperconverged Power Systems with Nutanix bridge the efficiency and performance gaps between enterprise data centers and public clouds. The solution helps to drive faster time-to-market for important IT initiatives, delivering benefits that were formerly only possible with public clouds. These benefits are exploited by implementing:

- Data locality

Nutanix continuously monitors data access patterns and places data in the most appropriate location.

- Next-generation virtualization

Designed for the era of unstructured data, Nutanix Acropolis Hypervisor (AHV) is a hypervisor that accelerates deployment and eases management. It is included at no extra cost with the solution eliminating virtualization licensing costs.

**Note:** The Acropolis operating system (Nutanix Acropolis Hypervisor) delivers software-defined enterprise-class storage, compute and native virtualization services to run nearly any application. It provides feature-rich data services in a turnkey hyperconverged infrastructure solution that can be deployed ready for use in 60 minutes or less. Its distributed file services eliminate the need for stand-alone SAN or NAS-based storage separate from compute servers, hence reducing the complexity of infrastructure management.

- Self-healing infrastructure

A Nutanix enterprise cloud is resilient by design. If a drive or node fails, workloads are automatically restarted and full resiliency is restored quickly without operator intervention, protecting applications from unplanned downtime.

- Built-in availability

Data protection, disaster recovery, and high availability are integral to the Nutanix environment, delivering higher application availability with less time and effort.

- One-click management

With Nutanix Prism, systems administrators easily monitor and manage all infrastructure, gaining full visibility of storage, CPU, and memory resources across IBM Power Systems and Intel x86-based servers from a single pane. One-click software, hypervisor, and firmware upgrades and one-click problem remediation take the pain out of day-to-day operations.

2.2.4 Increase security without adding silos

To ensure the security of sensitive data, system administrators find they have no choice but to deploy dedicated infrastructure for each application. However, applications can be deployed securely on an IBM Hyperconverged Power Systems with Nutanix with other workloads, avoiding the need for a separate silo of infrastructure.

Furthermore, Nutanix combines features such as two-factor authentication and data-at-rest encryption with a security development lifecycle. Nutanix systems are certified across a broad set of evaluation programs to ensure compliance with the strictest standards.
System configuration, reference architecture, systems management for Nutanix on IBM Power Systems

This chapter describes the general requirements for Nutanix running on IBM Power Systems servers. It includes Nutanix features and how these features can help to address the deployment of virtual machines.

This chapter includes the following sections:
- Starting Nutanix on IBM Power Systems
- Build Blocks - Nutanix on IBM Power Systems
- Nutanix web console - Prism Central
- Verifying and preparing the environment for deployment
- Deploying a virtual machine (VM)
3.1 Starting Nutanix on IBM Power Systems

IT teams recognize the need, and benefits of embracing the next generation of data center infrastructure technology. IBM with Nutanix are designed to give enterprise customers a scalable, resilient, high-performance hyperconverged infrastructure solution, benefiting from the data and compute capabilities of the POWER architecture, and the one-click simplicity of the Nutanix Enterprise Cloud Platform.

3.1.1 Available configurations

This solution is currently offered in two models as shown in the Table 3-1.

Table 3-1. IBM CS-series models

<table>
<thead>
<tr>
<th>Model</th>
<th>CS821 (1U)</th>
<th>CS822 (2U)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Compute</td>
<td>Two 10-core 2.09 GHz POWER8a</td>
<td>Two 11-core 2.89 GHz POWER8a</td>
</tr>
<tr>
<td>Threads</td>
<td>Up to 160 threads</td>
<td>Up to 176 threads</td>
</tr>
<tr>
<td>Memory</td>
<td>Up to 256 GB</td>
<td>Up to 512 GB memory</td>
</tr>
<tr>
<td>Disk</td>
<td>Up to 7.68 TB flash</td>
<td>Up to 46.08 TB flash</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>Nutanix AHV hypervisor</td>
<td></td>
</tr>
<tr>
<td>Supported operating system</td>
<td>Linux &amp; AIX</td>
<td></td>
</tr>
</tbody>
</table>

a. These models are still available for ordering. IBM intends to support this solution on POWER9 processors.

For more details about the hardware specs see 1.3, “Nutanix on IBM Power Systems” on page 3, and Table 1-1 on page 4.

For information about the products, see the IBM and Nutanix Launch Hyperconverged Initiative to bring Enterprises into the Cognitive Era® - IBM POWER and Nutanix Software target high performance workloads delivered by way of one-click private clouds announcement letters at the following websites:

https://ibm.co/2YmxYm0
https://ibm.co/2OZSar3

3.2 Build Blocks - Nutanix on IBM Power Systems

In this environment each node is called a block. The following are recommendations for production and non-production environments.
Figure 3-1 shows a reference for productive and non-productive environments.

CS821 or CS822 nodes or blocks recommended initial deployment characteristics:

1. Use 3 nodes minimum for non production environments (Dev/QA workloads)
2. Use 4 nodes minimum for production environments (critical workloads)

To scale the environment, you need to add new nodes and grow linearly:

- No SAN bottlenecks
- Pay-as-you grow (storage + compute)
- Simplified management

Figure 3-2 shows workload examples to have an idea on how to scale the environment.

Another configuration possibility for the environment is shown in Figure 3-3 on page 14 where you can see a multicloud solution with numerous scenarios and platforms.
3.3 Nutanix web console - Prism Central

Nutanix provides an option to monitor and manage multiple clusters through a single web console. This multi-cluster view web console, known as Prism Central, is a centralized management tool that runs as a separate instance and consists of either a single VM or a set of VMs.

From the web console, the Nutanix administrators can access the clusters and nodes using a web browser to visualize and manage the content of their projects. The web console runs as an application on the CVM.
Figure 3-4 shows the Prism Central login window.

![Prism Central login window](image1)

**Figure 3-4  Prism Central - Login pane**

The home option in the Prism dashboard is the main menu where most of the tasks start including for example, the creation of a VM. The Prism Central home dashboard is shown in Figure 3-5.

![Prism Central home dashboard](image2)

**Figure 3-5  Prism Central home dashboard pane**
The test lab environment is a cluster with three CS822 nodes. You can check the cluster and node configurations in Prism Central by way of **Prism Central → Main Menu → Hardware → Diagram**.

Figure 3-6 shows the diagram of cluster with the three nodes or blocks.

![Diagram of the cluster](https://bit.ly/2OqTaEj)

For more detail about the hardware diagram view, see the following link: [https://bit.ly/2OqTaEj](https://bit.ly/2OqTaEj)

**Prism Central main menu**

The main menu, shown at the top of every window, provides access to the features of Prism Central. Figure 3-7 shows the menu bar and the main options in Prism.

![Menu bar](https://bit.ly/2RDF0Qs)

1. Cluster Name
2. Main Menu
3. Health Indicator
4. Alerts
5. Tasks
6. Search field
7. Help Menu
8. Settings
9. User options

**Note:** It is not the purpose of this book to provide a detailed summary of Prism Central’s features. However, this section presents the main steps to help you understand the environment including performing a VM deployment and basic cluster management activities. There are several links throughout this section to help you find information about performing management and maintenance of the environment.

Table 3-2 shows the description for each of the options in the main menu.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Cluster name</td>
<td>Use to see the cluster name and change some data such as cluster name, Cluster Virtual IP Address, and others. <a href="https://bit.ly/2RDF0Qs">https://bit.ly/2RDF0Qs</a></td>
</tr>
</tbody>
</table>
3.4 Verifying and preparing the environment for deployment

This section explains a few verification checks to follow before deploying new VMs (virtual machines).

Cluster details
Each node has a CVM and Prism installed including its own IP address. You can also create a cluster virtual IP address to access a management console available even if one of the cluster nodes has failed.

You can add a cluster name or virtual facing IP address using Login in Console → Prism Central → Cluster name.
Figure 3-8 shows where to add or change a cluster name or virtual facing IP address.

![Cluster Details](image)

**Figure 3-8  Cluster details**

For more information see 3.3, “Nutanix web console - Prism Central” on page 14.

**Image configuration**

The goal is to deploy VMs and to do that you need operating system images.

You can import and configure operating system ISO or disk image files through the web console. Click **Login in Console → Prism Central → Main Menu → Settings → Image Configuration → Upload Image**.
Figure 3-9 shows the image configuration settings pane.

To create an image, you must enter a name, choose the type of image (ISO or disk), storage container and image source (this can be a URL or a file you have already downloaded).
Figure 3-10 presents the field you need to provide to create an image.

![Create Image Pane](image)

*Figure 3-10  Create image pane*
Figure 3-11 provides the pane for the images update.

![Update Image pane](image)

For information about image configuration options, refer to 6.2, “How do you deploy AIX using a Cloud Ready Image by way of Prism” on page 68. For additional details, see the Nutanix Portal at:

https://bit.ly/2Os2wQe

**Network configuration**

The second item you need is a network. After creating an image which you use in later steps, you need to create your network.

For clusters with Nutanix virtualization running AHV, you can configure network connections through the web console. Each VM network interface is bound to a virtual network, and each virtual network is bound to a single VLAN. To create the network:

Login in Console → Prism Central → Main Menu → Settings → Network Configuration → +Create Network
Figure 3-12 shows the window where you find the network settings.

![Network configuration](image)

**Figure 3-12 Network configuration**

Figure 3-12 shows the following fields and the details to provide:

- **Name**: Enter a name of the network.
- **VLAN ID**: Give a number our VLAN.
- **Enable IP Address Management**: Mark the check box.
- **Network IP Address/Prefix Length**: Type the number of IP address of the gateway for the network and prefix with the network prefix (for example, 10.8.0.0/24).
- **Gateway IP Address**: Inform VLAN default gateway IP address.

**Note**: If you do not enable the IP address management option when creating a network, you cannot enable it.
Figure 3-13 shows the basic settings to create a vlan.

![Figure 3-13  Create network (vlan) pane](image)

Figure 3-13 shows the following fields and the details to provide:

- **Configure Domain Settings**: Mark the check box.
- **Domain Name Servers**: Inform a DNS Server.
- **Domain Search**: Inform a domain.
- **Domain Name**: Enter with VLAN domain name.
Figure 3-14 shows basic information to configure the domain settings.

![Figure 3-14 Configure domain settings pane]

**Configure Domain Settings**

**Domain Name Servers (Comma Separated)**

10.31.200

**Domain Search (Comma Separated)**

redbooks.ibm.com

**Domain Name**

redbooks.ibm.com

**TFTP Server Name**

-

**Boot File Name**

-

**Note:** If you need it, you can also create a range of addresses for automatic assignment of virtual NICs. This option called Create Pool.

Figure 3-15 on page 25 shows the following fields options to consider and configure as needed:

- **Override DHCP Server:** Mark the check box if you already have a DHCP server.
- **DHCP Server IP Address:** Enter an IP address in the DHCP server IP address → Click Save.
Figure 3-15  Override DHCP server pane

Now you are ready to deploy a virtual machine using the downloaded operating system images.

For more information about network configuration for VM interfaces, see the Nutanix Portal documentation at the following website:


### 3.5 Deploying a virtual machine (VM)

This section presents the necessary steps for creating a VM. If you have not yet created an Image or network, review sections: “Image configuration” on page 18, “Network configuration” on page 21, and section 3.4, “Verifying and preparing the environment for deployment” on page 17.
Figure 3-16 shows the VM main menu dashboard.

The dashboard (Figure 3-16) has information about the virtual machines already created. To know more details about the VM dashboard, refer to the following link:

https://bit.ly/2qxfDXa

To perform the creation of a virtual machine, Login in Console → Prism Central → Main Menu → VM → +Create VM.
Figure 3-17 displays information about the running VMs.

![Create VM - Table pane](image)

Figure 3-17 shows information about the virtual machines already created in a table format. To learn more details about each field or function, refer to the following link:

https://bit.ly/2LqC0og

The following are a few fields to know about:

- **Name**: Enter a name of the virtual machine
- **Description**: You can leave a description that helps to identify the virtual machine. Example: Application Type, Server Role, Operating System, and others.
- **vCPU**: Number of virtual CPUs to allocate to this VM.
- **Number of Cores per vCPU**: Number of cores assigned to each vCPU.

**Note**: Do not click the save button until you have completed all the steps.
Figure 3-18 shows the general fields required to create a VM.

![Create VM pane - General Configuration fields](image)

*Figure 3-18  Create VM pane - General Configuration fields*
Figure 3-19 shows the second part in the process to create a VM. In this pane, you set up the amount of allocated memory (in MBs) to this VM.

![Create VM pane](image)

<table>
<thead>
<tr>
<th>Create VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
</tr>
<tr>
<td><img src="image" alt="Memory" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Disks" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Volume Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Volume Groups" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Adapters (NIC)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Network Adapters" /></td>
</tr>
</tbody>
</table>

Then **Click + Add New Disk**.

The following are important fields to note as shown in Figure 3-20 on page 30:

- **Type**: CD-ROM.
- **Operation**: Clone from image Service.
- **Image**: Selects the operating system image that you can deploy.
Figure 3-20 shows an example of adding an image to install an operating system, in this case Red Hat Linux.

![Add Disk Form](image)

**Figure 3-20  Create VM (Add Disk pane) - Image**

Figure 3-20 shows the following configuration fields:

- **Type**: Disk.
- **Operation**: Allocate on Storage Container.
- **Storage Container**: Choose the Storage Container that is available. This option is presented only when DISK is selected in the type field.
- **Size**: Disk size in GBs.

Then Click **Add**.

Figure 3-21 on page 31 shows an example to create a disk. This disk is used to receive the operating system installation.
Figure 3-21  Create VM - Add Disk - Storage Container

Then Click **Add**.
Figure 3-22 shows the last step you need to perform before creating a virtual machine. Click +Add New NIC.

![Create VM - Add New NIC](image)

### Volume Groups

Please create a VM before you can add a volume group.

- Add Volume Group

### Network Adapters (NIC)

You haven’t added any NICs yet.

- Add New NIC

**Figure 3-22  Create VM - Add New NIC**

Figure 3-23 on page 33 shows an example to create a NIC. The following fields are presented:

- **VLAN Name**: Choose a VLAN.
- **Connected**: Mark this option.
- **IP Address**: Choose a valid IP address for the server as the VLAN mask.

**Note**: The IP address entry is optional. If left blank, the next available IP address is picked and used automatically.
Then Click **Add → Save**.

When you click **Save** a task is created. You can track the status of your tasks, and if something has not gone as planned, you can check and start the creation process again. From the dashboard menu as shown in Figure 3-7 on page 16 (item numbered 5), click **Tasks → View All Tasks**.

Figure 3-24 shows the panel of all recent tasks that have been executed or are still running.

![Create VM - Tasks pane](image-url)
You now see the virtual machine created in table view. At this point, the state of the server is stopped as shown in Figure 3-25.

Now you can start the VM by Selecting the VM Created → Power ON → Launch Console → VNC.

Figure 3-26 presents options that can be executed for the VMs created.
From this point, you can install an application, set up a database, and perform any action similar to a physical server.

For more information about all fields and options for creating a VM, see the following link: https://bit.ly/2PjDcLd

Figure 3-27 shows the monitoring of a Red Hat Linux installation on a VM using the VNC Launch Console.

![Figure 3-27](image)

Figure 3-27   VM Created pane - Installing the operating system

For more information about management of VMs and other related functions with Nutanix’s Prism Central, refer to the following link:

https://bit.ly/37ML0xN
Configuration and management recommended practices

This chapter describes a few management recommended practices from the experiences gathered during the project.

This chapter contains the following:

- Nutanix disaster recovery
- Configuring a protection domain (async DR)
4.1 Nutanix disaster recovery

This section presents a topic important for system administrators. This section delivers steps on how to configure disaster recovery in a Nutanix cluster.

4.1.1 Implementing disaster recovery

Nutanix allows native backup and disaster recovery (DR) giving the users the ability to backup and restore objects running on local or cloud environments (Xi).

To start the data protection follow the steps using the Nutanix’s Web Console as shown in Figure 4-1.

![Nutanix Web Console - Data Protection](image_url)

*Figure 4-1 Nutanix Web Console - Data Protection*
In the Data Protection dashboard click Protection Domain and select Async DR from the drop-down list. The Protection Domain window appears as shown in Figure 4-2.

![Figure 4-2 Nutanix Web Console Protection Domain pane](image1)

Figure 4-3 shows where the administrator enters the details of the remote site either using a proxy and selecting a backup or the disaster recovery option.

![Figure 4-3 Nutanix Web Console - Remote Site configuration pane](image2)
Figure 4-4 shows the details for the remote site, the running replications and a snap shot of the disaster recovery such as metrics, alerts and events.
Figure 4-5 shows the name and the current setting of the disaster recovery configuration.
Figure 4-6 shows the table, overview and current settings of the disaster recovery restore configuration.

Figure 4-6  Nutanix Web Console - Restoring the VM
Figure 4-7 shows the restored snapshot, its name, and the type of the protection domain.

![Figure 4-7 Nutanix Web Console - Restore Snapshot pane](image)

Nutanix also supports several types of protection strategies, some of which are one-to-one or one-to-many replications. The replication strategies can be implemented as a data protection by configuring the protection domains and remote sites by using the Nutanix Web Console.

A protection domain is a defined set of virtual machines and volume groups or storage containers.

Replication is an important component of the enterprise data protection solution, and it ensures that critical data and applications can be effectively replicated to a site or another environment.

Some of the replication options are *Per-VM Backup* which provides the ability to designate certain VMs for backup to a different site, and the *Selective Bi-directional Replication* which is in addition to replicating selected VMs. This provides a flexible replication solution by accommodating a variety of enterprise topologies.

### 4.2 Configuring a protection domain (async DR)

This procedure shows how to create a protection domain that supports backup snapshots for selected VMs and volume groups using asynchronous data replication.
Before starting, you need to ensure that you meet the protection domain guidelines for configuring the Async DR before proceeding. These are the guidelines provided by Nutanix and are available on their website:


### 4.2.1 General guidelines

For successful replication, all local VMs controllers must be able to communicate with all remote controller VMs.

Nutanix recommends that any ESXi clusters implementing a disaster recovery configuration be registered with vCenter Server. This also applies to a cross-hypervisor disaster recovery (CHDR) configuration.

For VM migration as part of data replication to succeed in an ESXi hypervisor environment, check that you have configured forward (DNS A) and reverse (DNS PTR) DNS entries for each ESXi management host on the DNS servers used by the Nutanix cluster.

**Note:** The hardware page on the web console shows the host name and hypervisor IP address for each management host. The nCLI command `ncli host ls` also lists each hypervisor's IP address.

When configuring encryption of data replication, set up an encrypted site-to-site tunnel and specify the tunnel IP address when you create the remote site (specify the tunnel IP address in the addresses parameter).

For cases when bandwidth between sites is limited, set a limit on the bandwidth that replication uses by specifying the Maximum Bandwidth parameter.

**Note:** A consistency group is a subset of the entities in a protection domain. Consistency group typically must not exceed more than 20 entities.

One-time snapshots have infinite expiry time, and hence it is recommended to specify retention time when you are creating one-time snapshots.

- Do not include the source and the destination cluster under the same data center because VMs can be deleted from both the source and destination clusters post migration process.
- Do not make any changes in the VM or its configuration on the source cluster after the VM gets powered off by the system during the migration of the protection domain. Otherwise, the changes that you have made will be lost.

**Note:** In case you are using the same vCenter Server to manage both the primary and remote sites, do not have the storage containers with the same name on both the sites.

- Do not have VMs with the same name on the primary and the secondary clusters. Otherwise, it can affect the recovery procedures.

The snapshot operation fails after six retries of the protection domain that has the VM on which some other ongoing tasks are currently in progress. The snapshot operation succeeds only if within these six retries the ongoing tasks on the VM are completed.
If you use Prism to take one or more storage snapshots of a protection domain that includes a VM which also has VMware snapshots, there is a risk that the VM snapshot might become corrupted under certain circumstances. To avoid this issue, apply the following best practices:

1. Ensure that a VMware snapshot (through vCenter) does not exist when a storage snapshot (Prism) is taken.

2. Perform storage snapshots during times when VM snapshots are less likely to happen.

3. Restore VMs only from a storage snapshot that was taken when no VMware snapshots existed for the VMs or the VMs were powered off when the storage snapshot was taken.

4. Schedule Nutanix storage snapshots to have limited or no overlap with manual or backup initiated VM snapshots.

5. Do not use the VMware level encryption that is used to encrypt the existing virtual machine or virtual disk along with Async DR configuration as this is not supported.

To protect VMs created by VMware View Composer or Citrix XenDesktop, Nutanix recommends adding files associated with the VM gold image to a protection domain. Use the nCLI command to create the protection domain.

For example, to protect the replica-ABC.vmdk file in a protection domain named vmware1-pd and a consistency group named vmware1-cg run the following command:

```
ncli> protection-domain protect \
    files=/container1/view-gold-image/replica-ABC.vmdk name=vmware1-pd \
    cg-name=vmware1-cg
```

**Note:** You must disable the VMware View pool from the View Composer. For more information about disabling View Pool from the View Composer, see VMware View documentation at the following website: https://bit.ly/36o9nQa

If the local and remote sites are running ESXi and are registered with vCenter Server, MAC address retention after a failover works as follows:

- If you configure a static MAC address for a virtual network adapter on the local site, the adapter retains its MAC address after the protection domain is activated at the remote site and the VM is powered on.

- If you configure automatic MAC address assignment for the virtual network card on the local site, and both sites (local and remote) are registered with the same vCenter Server, the MAC address changes as soon as the VM is registered at the remote site.

- If you configure automatic MAC address assignment for the virtual network card on the local site, and the local and remote sites are registered with separate vCenter Servers, the MAC address changes only after the VM is powered on at the remote site.

At a given site (primary or remote), all the nodes in the Nutanix cluster must be part of the same ESXi host cluster and the network must be available on all the nodes in the Nutanix cluster.
In the event you are deploying an intrusion prevention system (IPS) appliance or software, consider whether any configured filters or other network monitoring aides can block packets transferred during replication operations. You can add the IP address of any appliances or systems running the software to the whitelist as described in Configuring a Filesystem Whitelist in the Web Console Guide at the following website (https://bit.ly/2NUZDGG):

- (Hyper-V) - It is recommended to create a VM in their unique folders instead of using a default folder. If a default folder is used to create the VMs, you will not be able to protect these VMs.
- (Hyper-V) - Path-prefix must not be reused for the same VMs in a protection domain.

When a VM running NGT is restored from a hypervisor-based snapshot created before NGT was installed on the VM, the VM is restored without NGT. Consequently, any native snapshots of the VM created after restoration are based on stale NGT information. To avoid this issue, after restoration, either reinstall or disable NGT on the VM.

There are also some limitations which need consideration. General limitations for Async DR are as follows:

- Protection domains can have no more than 200 entities (VMs or volume groups). It is recommended that each application which constitutes set of entities is protected by a unique protection domain.
- Because restoring a VM does not allow for VMX editing, VM characteristics such as MAC addresses can be in conflict with other VMs in the cluster.
- To be in a protection domain, a VM must be entirely on Nutanix datastore (no external storage).
- Data replication between sites relies on the connection for encryption.
- It is not possible to make snapshots of entire file systems or storage containers.
- The shortest possible snapshot frequency is one per hour.
- Consistency groups cannot define boot ordering.

Inactivating a protection domain deletes the entities from the cluster. Deleting a protection domain removes all the snapshots associated with the protection domain from the cluster. Hence, do not inactivate and then delete a protection domain that contains VMs. Either delete the protection domain without inactivating it, or remove the VMs from the protection domain before deleting it.

**Attention:** If you inactivate and then delete a protection domain that contains VMs, the VMs in the protection domain gets deleted.

Some VMs might not appear in the web console or in the nCLI command results during rolling upgrades, planned Controller VM maintenance, or when hosts are down or unavailable. Some protection domain operations like snapshots or protection might also fail in this case.

The following limitations apply to the inclusion of related entities in a protection domain:

1. If the number of entities in a consistency group exceeds ten, protection of related entities fails. However, if you want to include more than ten entities in a protection domain, protect the entities in separate consistency groups within the same protection domain. Even if two related entities are in separate consistency groups, as long as they are in the same protection domain, their attachment configuration is included in the snapshots and restored during recovery.
2. Snapshot creation and recovery of attachments are not supported if you have configured volume groups with the following:
   - Challenge-Handshake Authentication Protocol (CHAP). The iSCSI target secret is cleared after a volume group is restored from a snapshot.
   - IP addresses. If you attach volume groups to VMs by whitelisting the IP addresses of the VMs, entities are recovered, but their attachment configuration is not recovered. You must manually reattach the entities after recovery.

Supported operating systems:
- Red Hat Enterprise Linux 6.7 and 6.8.
- Oracle Linux 6.7 and 7.2.

Limitations specific to vSphere environment are as follows.
- Nutanix native snapshots cannot be used to protect VMs on which VMware fault tolerance is enabled.
- Virtual machines connected to a vSphere Distributed Switch (dvSwitch) are not connected to their port group after failover. After failover, you must manually connect such VMs to their port group.

Limitations specific to Hyper-V environment are as follows:
1. A disaster replication snapshot fails and raises an alert if:
   - Any VM files (for example, configuration, snapshots, virtual disks, and ISOs) are residing on non-Nutanix storage containers.
   - All virtual disks associated with a VM are located in different directories or folders. That is, all virtual disks associated with a VM must be located in a single directory or folder.
   - A VM's folder and its snapshot folder are located in different directory or folder paths. That is, a snapshot folder is typically located in a snapshot folder under the VM's folder. The snapshot folder must be under the VM folder or the replication fails.
2. Run-as account must be a domain account and must have local administrator privileges on the Nutanix hosts. This can be a domain administrator account. When the Nutanix hosts are joined to the domain, the domain administrator accounts automatically takes administrator privileges on the host. If the domain account used as the run-as account in SCVMM is not a domain administrator account, you need to manually add it to the list of local administrators on each host by running `sconfig`.
3. Nutanix does not support Hyper-V replica VM in the Async DR protection domain.
4. The name of the Hyper-V virtual switches between primary and remote site must be the same, otherwise the restore fails.

If the base VM and the differencing disk is in the same protection domain, after migrating to the protection domain to the secondary site in-place restore fails.

Any VMs created during hypervisor upgrade, including VMs created through disaster recovery operations, experience downtime if they are not configured for high availability and the corresponding node is yet to be upgraded. It is recommended that you not create any VMs, either manually or through disaster recovery operations, during hypervisor upgrade.

After a VM is migrated, its entry is not removed from Failover Cluster Manager on the source cluster.
If a highly available VM is protected, clones of the VM do not inherit the high availability configuration.

A highly available VM is no longer highly available after it is migrated to a remote site unless an entry with the same VMID already exists in Failover Cluster Manager.

During an in-place restore or failback operation, the HA property of a VM is honored only if the entry is not cleaned up in Failover Cluster Manager.

The state of the HA property of a VM is not captured when a snapshot is created. If a VM's VMID entry is present in Failover Cluster Manager when it is being restored from its snapshot, then the VM is HA protected.

The owner node of a VM can change when a disaster recovery operation, such as in-place restoration or migration, is performed on the VM.

For more information, Nutanix provides a practices guide to setup DR at the following website:

https://portal.nutanix.com/#/page/solutions
Nutanix use cases

This chapter describes Nutanix use cases created and documented during the project to illustrate common scenarios and providing how-to technical configuration details.

This chapter contains the following:

- Installing packages in Nutanix
- Introduction to MongoDB
- Installation and configuration of a MongoDB Shard in a Nutanix Cloud
- The Nutanix Command Line Interface (CLI)
- The Nutanix REST API
5.1 Installing packages in Nutanix

Nutanix has a wide array of use cases, including virtualization, databases, machine learning, file storage, analytics, and backups.

The current landscape of the industry requires high availability services that remain operational even when some failure occurs. This represents a considerable challenge for infrastructure teams, who now, more than ever, have the responsibility to deliver a better service in reduced timeframes.

Nutanix provides the toolbox that can help a team to focus on maintaining, improving and upgrading those services and not on worrying about the underlying platform configuration.

5.2 Introduction to MongoDB

MongoDB is a document-oriented NoSQL database which has gained tremendous popularity in recent years. It is developed by MongoDB Inc., and is currently published with the Server Side Public License (https://www.mongodb.com/licensing/server-side-public-license). MongoDB Inc. offers Community and Enterprise versions for MongoDB and the fully-managed Atlas service.

MongoDB’s JSON storage methodology and the smooth manner in which it is installed, configured and scaled has attracted users looking for a solution that can start small and grow with their projects.

5.2.1 The document-based query language

In MongoDB, the JSON files are referred to as documents, and each document can be seen as the equivalent of a table in SQL databases.

JSON (which stands for JavaScript Object Notation) is a widely-used open file format. Due to being language-independent and its human-readable way of storing information and representing relations in data, JSON is often used for REST APIs and configuration files. Data in JSON can be easily parsed and many libraries have been written to facilitate even more the interaction with it.

In order to retrieve data, MongoDB offers the Mongo Shell and connectors for several programming languages, including Python, Java, Go, Ruby, Scala, and others.

As opposed to traditional relational databases, MongoDB does not use the SQL syntax to interact with the stored data. MongoDB uses JSON syntax both for the query and the output of the results.

For instance, the MongoDB equivalent of a SELECT * FROM table SQL query is:

db.inventory.find( {} )

5.3 Installation and configuration of a MongoDB Shard in a Nutanix Cloud

This sections describes the installation and configuration of MongoDB on Nutanix.
5.3.1 MongoDB and Nutanix

MongoDB is the next-generation database that helps businesses transform their industries by utilizing the power of data. The world’s most savvy organizations, from startups to the largest companies, use MongoDB to create applications at a fraction of the cost of older generation databases. MongoDB is the fastest-growing database ecosystem, with over 10 million downloads, thousands of customers, and over 1,000 technology and service partners.

On the other hand, 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. Using Nutanix, customers benefit from predictable performance, linear scalability and cloud-like infrastructure consumption.

Nutanix enables MongoDB to run elastic workloads on a highly scalable converged infrastructure as follows:

- **Modular incremental scale** - With Nutanix you can start small and scale both vertically and horizontally. A single Nutanix block provides up to 20 TB of storage and 80 cores in a compact 2RU footprint.

- **Vertical scale** - You can increase RAM and CPU in a rolling fashion as workload increases; the SSD-backed hot tier provides improved I/O response for all workload eventualities; and you can easily upgrade to new hardware by adding nodes to a running cluster and then hot migrating VMs.

- **Horizontal scale** - The ability to add a single block holding a replica set (or shard) creates several advantages: Nutanix supports MongoDB at elastic scale; you can grow the database in a continuously highly available fashion; operations can add capacity without developer involvement or knowledge of application dependencies.

- **Effective transparent data tiering** - Nutanix incorporates heat-optimized or activity-based tiering, which uses multiple storage tiers and places data on the tier that provides the best performance.

- **New compute and storage** - The Prism management interface makes it simple to increase capacity, and the Prism Central tool allows you to easily manage multiple different cluster systems, eliminating the need for multiple sessions.

- **Enterprise-grade cluster management** - Through Prism, Nutanix provides a simplified and intuitive approach to managing large clusters that includes alert notifications, a converged GUI that serves as a single pane of glass for servers and storage, and a bonjour mechanism to automatically detect new nodes in the cluster.

- **High-density architecture** - The Nutanix advanced server architecture integrates eight Intel CPUs (potentially more than 80 cores) and up to 4 TB of memory into a single 2RU appliance.

- **Business continuity and data protection** - MongoDB and its associated infrastructure VMs are mission-critical and need enterprise-grade data management features, including backup and disaster recovery. Nutanix provides these features ready to use, and you manage them alongside the rest of your virtual environment.

- **Data efficiency** - Nutanix compression policies are truly VM-centric. Unlike traditional solutions that perform compression mainly at the LUN level, Nutanix provides all of these capabilities at the VM and file levels, greatly increasing efficiency and simplicity.
5.3.2 Reference architecture

The architecture for this case scenario is integrated on three nodes: One primary and two secondary nodes as shown in Figure 5-1. This is the basic unit of *sharding*.

![Reference architecture](image)

The primary instance of a MongoDB database consists of a single *mongod* daemon. This daemon handles all reads and writes by default. Both read and write operations are memory based. Data is written to memory first and then to a journal file on disk every `journalCommitInterval` milliseconds (default value of 100). Then the OS flushes the journal file contents, at 60-second intervals, to the permanent data files residing on disk.

To increase redundancy and data availability, MongoDB employs replica sets. A replica set consists of three or more *mongod* instances, located on one primary and two or more secondary nodes. The primary node receives all writes to ensure strict consistency. By default, all read operations also go to the primary node, but the secondary nodes can be configured to receive read operations if required. Note that, because of what is called replication lag, the secondary nodes cannot always contain the most current data.

5.3.3 Installing MongoDB on Nutanix

This section describes the steps for installing MongoDB on a Nutanix VM.

First deploy a Linux ppc64 VM according to supported platforms and desired Enterprise MongoDB versions as shown in Table 5-1.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Enterprise 4.2</th>
<th>Enterprise 4.0</th>
<th>Enterprise 3.6</th>
<th>Enterprise 3.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL or Centos 7</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Ubuntu16.04</td>
<td>Not supported</td>
<td>Supported</td>
<td>Removed starting in 3.6.13</td>
<td>Removed starting in 3.4.21</td>
</tr>
</tbody>
</table>
This case uses a VM with RHEL 7.7 and install it using `yum`. Other methods can be used for installing the operating system. For example downloading .rpm or .tgz files.

1. Create a repository file for mongoDB.
   
   ```bash
   $ touch /etc/yum.repos.d/mongodb-enterprise.repo
   ```

2. Edit the file created by adding the desire mongoDB version repository. This scenario uses the latest release 4.2.
   ```
   [mongodb-enterprise]
   name=MongoDB Enterprise Repository
   baseurl=https://repo.mongodb.com/yum/redhat/$releasever/mongodb-enterprise/4.2/
   $basearch/
   gpgcheck=1
   enabled=1
   gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
   ```

   **Note:** If other version of mongoDB is required, you can find the repository locations at the following link:
   https://docs.mongodb.com/v4.0/tutorial/install-mongodb-enterprise-on-red-hat/

3. Refresh the `yum` repositories.
   ```bash
   $ yum cache search mongodb-enterprise
   ```

4. Install mongoDB by executing the `yum` command.
   ```bash
   $ yum install -y mongodb-enterprise
   ```

5. This installation creates default directories and the owner and group is set to `mongod`.
   ```
   # data directory
   /var/lib/mongodb
   # log directory
   /var/log/mongodb
   ```

6. Check for SELinux to be running in enforcing mode to enable access to the files.
   ```bash
   $ getenforce
   Enforcing
   ```

7. Create a custom policy file.
   ```bash
   $ cat > mongodb_cgroup_memory.te <<EOF
   module mongodb_cgroup_memory 1.0;
   require {
     type cgroup_t;
     type mongod_t;
     class dir search;
     class file { getattr open read };
   }

   ================ mongod_t ================
   allow mongod_t cgroup_t:dir search;
   allow mongod_t cgroup_t:file { getattr open read };
   EOF
   ```

8. Compile and load the custom policy.
   ```bash
   $ yum install policycoreutils-python
   $ checkmodule -M -m -o mongodb_cgroup_memory.mod mongodb_cgroup_memory.te
   ```
$ semodule_package -o mongodb_cgroup_memory.pp -m mongodb_cgroup_memory.mod
$ sudo semodule -i mongodb_cgroup_memory.pp

9. Start the mongoDB process.
   $ service mongod start
   Redirecting to /bin/systemctl start mongod.service

10. For a process to start after a system reboot:
    $ chkconfig mongod on
    Note: Forwarding request to 'systemctl enable mongod.service'.

11. Make sure it is running or search for any errors by looking at the log file
    /var/log/mongodb/mongod.log. When running, the following line is in the file: the port
default is 27017.
    [initandlisten] waiting for connections on port <port>

12. To stop or restart MongoDB use the following commands:
    $ service mongod stop

    $ service mongod restart

5.3.4 Configuring a MongoDB Replica Set

A replica set enables MongoDB with redundancy and increased data availability by replicating
the database to multiple servers. The primary member of the replica set can be used for read
and write operations whereas the secondary members are available for read-only operations.

A replica set is integrated of an odd number of MongoDB servers. If a single server fails, the
replica set members must be able to elect a new primary server.

The following steps explain how to configure a replica set with one PRIMARY and two
SECONDARY nodes. Check the hosts file entries by using the following command:

$ cat /etc/hosts
10.53.168.110 mongodb01 mongodb01.ibm.redbook.com
10.53.168.111 mongodb02 mongodb02.ibm.redbook.com
10.53.168.112 mongodb03 mongodb03.ibm.redbook.com

The following steps are just for the PRIMARY node:

1. MongoDB must successfully be running on the primary server instance. Start mongo with
   the following command:
   $ service mongod start

2. Create the necessary mongodb users. This case creates an admin and a test user:
   #Start the mongo shell
   $mongo

   #create the admin user
   > use admin
   > db.createUser ({user: 'admin',pwd: 'Passw0rd',roles: [{role: 'root',
     db:'admin'}]})
   > db.auth("admin","Passw0rd")

   #create the test user for the test database
   > use test
> db.createUser ({user: 'usrtest',pwd: 'Passw0rd',roles: [{role: 'readWrite',
db:'test'}]})
> db.auth("usrtest","Passw0rd")

3. Modify the `/etc/mongod.conf` file to enable the replica set as follows (just the bold text has been modified):

```
$vi /etc/mongod.conf
# mongod.conf
# for documentation of all options, see:
# http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
    destination: file
    logAppend: true
    path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
    dbPath: /var/lib/mongo
    journal:
        enabled: true
# engine:
# wiredTiger:

# how the process runs
processManagement:
    fork: true  # fork and run in background
    pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
    timeZoneInfo: /usr/share/zoneinfo

# network interfaces
net:
    port: 27017
    bindIp: 0.0.0.0 #mongodb01 #127.0.0.1  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.

security:
    keyFile: /etc/mongodb-keyfile

#operationProfiling:

replication:
    replSetName: "rs0"

#sharding:

## Enterprise-Only Options

#auditLog:

#snmp:
```
4. Open port 27017 if not already opened with the following commands:
   $ firewall-cmd --zone=public --add-port=27017/tcp --permanent
   $ firewall-cmd --reload
   $ firewall-cmd --list-all

5. Create a mongodb key file:
   $ echo mypassword > /etc/mongodb-keyfile
   $ chmod 400 /etc/mongodb-keyfile
   $ chown mongod:mongod /etc/mongodb-keyfile

6. Start the mongo shell again and initiate the replica set:
   $mongo
   > use admin
   > db.auth("admin","Passw0rd")
   > rs.initiate()

   All the SECONDARY nodes must have mongodb properly installed and running. The following steps need to be executed over each SECONDARY node:

7. Open port 27017 if not already opened:
   $ firewall-cmd --zone=public --add-port=27017/tcp --permanent
   $ firewall-cmd --reload
   $ firewall-cmd --list-all

8. Create a mongodb key file with the same password stored in the file created for the PRIMARY node:
   $ echo mypassword > /etc/mongodb-keyfile
   $ chmod 400 /etc/mongodb-keyfile
   $ chown mongod:mongod /etc/mongodb-keyfile

9. Modify the /etc/mongod.conf file to enable the replica set as follows (just the bold text has been modified):
   $vi /etc/mongod.conf
   # mongod.conf
   # for documentation of all options, see:
   # http://docs.mongodb.org/manual/reference/configuration-options/

   # where to write logging data.
   systemLog:
   destination: file
   logAppend: true
   path: /var/log/mongodb/mongod.log

   # Where and how to store data.
   storage:
   dbPath: /var/lib/mongo
   journal:
   enabled: true
   # engine:
   # wiredTiger:

   # how the process runs
   processManagement:
   fork: true  # fork and run in background
   pidfilePath: /var/run/mongodb/mongod.pid  # location of pidfile
5.3.5 Recommended practices and conclusions

This section shows a few recommended practices, and provides some concluding remarks.

1. Use LVM to create a 6 column striped data volume. All Nutanix vDisks are redundant (RF=2). To create a RAID10 data volume just stripe the vDisks, and then create two further linear volumes. First create the underlying physical volumes as follows:

   # lsscsi | awk '{print $6}' | grep /dev/sd | grep -v sda | xargs pvcreate
   Physical volume "/dev/sdb" successfully created
   Physical volume "/dev/sdc" successfully created
   Physical volume "/dev/sdd" successfully created
   Physical volume "/dev/sde" successfully created
   Physical volume "/dev/sdf" successfully created
   Physical volume "/dev/sdg" successfully created
   Physical volume "/dev/sdh" successfully created
   Physical volume "/dev/sdi" successfully created
   Physical volume "/dev/sdj" successfully created

2. Then create both the volume groups and the required volumes:

   vgcreate mongodata /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg
   vgcreate mongojournal /dev/sdh
   vgcreate mongolog /dev/sdi
   # lvcreate -i 6 -l 100%VG -n mongodata mongodata
   # lvcreate -l 100%VG -n mongojournal mongojournal
Implementing and Managing a High-performance Enterprise Infrastructure with Nutanix on IBM Power

3. Create an XFS file system on each volume:

```
+ lvcreate -l 100%VG -n mongolog mongolog
```

```
+ mkfs.xfs /dev/mapper/mongodata-mongodata
+ mkfs.xfs /dev/mapper/mongojournal-mongojournal
+ mkfs.xfs /dev/mapper/mongolog-mongolog
```

4. Create the required mountpoints:

```
+ mkdir -p /mongodb/data mongodb/journal /mongodb/log
```

5. Mount the file systems and set the `noatime` option on the data volume:

```
+ /dev/mapper/mongodata-mongodata /mongodb/data xfs defaults,auto,noatime,noexec 0 0
+ /dev/mapper/mongojournal-mongojournal /mongodb/journal xfs defaults,auto,noexec 0 0
+ /dev/mapper/mongolog-mongolog /mongodb/log xfs defaults,auto,noexec 0 0
```

6. Set up a softlink to re-direct the journal I/O to a separate volume:

```
+ # ln -s /mongodb/journal /mongodb/data/journal
+ lrwxrwxrwx. 1 root root 21 Nov 21 14:13 journal -> /mongodb/journal
```

7. At this point set the file system ownership to the MongoDB user:

```
+ # chown -R mongod:mongod /mongodb/data mongodb/journal mongodb/log
```

Prior to starting MongoDB, there are a few known best practices to adhere to:

1. First, reduce the read ahead on the data volume to avoid filling RAM with unwanted pages of data. MongoDB documents are quite small and a large `readahead` figure can fill the RAM with additional pages of data that have to then be evicted to make room for other required pages. Filling virtual memory with this superfluous data can have an adverse effect on performance. The usual recommendation is to start with a setting of 16 K (32 * 512 M sectors) and then adjust upwards from there:

```
+ # blockdev --setra 32 /dev/dm-3
+ # blockdev --getra /dev/dm-3
+ 32
```

2. MongoDB recommends that you disable transparent huge pages. Edit your startup files as follows:

```
+ #disable THP at boot time
+ if test -f /sys/kernel/mm/redhat_transparent_hugepage/enabled; then
+ echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
+ fi
+ if test -f /sys/kernel/mm/redhat_transparent_hugepage/defrag; then
+ echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
+ fi
```

3. Set `swappiness=1` - MongoDB is a memory-based database; if the nodes are sized correctly, then there is no need to swap. However, setting `swappiness=0` can cause unexpected invocations of the out of memory (OOM) killer in certain Linux distros.

```
+ $ sudo sysctl vm.swappiness=1 (for current runtime)
+ $ sudo echo 'vm.swappiness=1' >> /etc/sysctl.conf (make permanent)
```

4. Disable NUMA, either in the VM BIOS or invoke `mongod` with NUMA disabled. All supported versions of MongoDB ship with an init script that automates this as follows:

```
+ numactl --interleave=all /usr/bin/mongod -f /etc/mongod.conf
```
5. Also check the `zone_reclaim_mode` is 0:

```bash
$ sudo cat /proc/sys/vm/zone_reclaim_mode
0
```

6. Finally, after you have configured the `/etc/mongod.conf` file (as root), you can start the `mongod` service. See output from `grep -v /etc/mongod.conf` as follows:

```bash
logpath=/mongodb/log/mongod.log
logappend=true
fork=true
dbpath=/mongodb/data
pidfilepath=/var/run/mongodb/mongod.pid
bind_ip=127.0.0.1,10.68.64.110
```

After the database has started, then you can connect by way of the mongo shell, and verify the database is up and running:

```bash
$ mongo
MongoDB shell version: 3.0.3
connecting to: test
>
```

Now that you have the `mongodb` instance installed, you can use it as a template to clone additional MongoDB hosts on demand. Now you need to get some data loaded, perform a few CRUD (create, read, update and delete) operations, and perform some additional testing.

### 5.4 The Nutanix Command Line Interface (CLI)

The Nutanix CLI is a set of commands that help to control Nutanix clusters without the need to interact with Prism. This has several advantages like enabling the scheduling and automation of tasks that are performed continuously.

Interaction with other tools is also possible. For example, the `cron` scheduler can be used to automatically perform repetitive tasks in regular intervals.
The CLI can be installed by clicking the upper right corner of the Prism dashboard, then selecting the *Download nCLI* option as shown in Figure 5-2. This results in the download of a compressed file, which is included in the Nutanix CLI executables.

![Figure 5-2   Downloading the Nutanix CLI package](image)

After the downloaded file is uncompressed, a terminal can be used to access the Nutanix CLI. Example 5-1 displays the version of the executable.

**Example 5-1   Displays the Nutanix version**

```
./ncli -v
ncli Version : euphrates-5.10.0.7-stable
Changeset ID : 0492fb
Changeset Date : 2018-12-19 01:23:07 -0800
Is LTS : false
```

To login into a Nutanix cluster using the nCLI, the following options are specified:

- `-u` (username)
- `-p` (password, leave black for prompt)
- `-s` (hostname or IP address)

Example 5-2 shows starting a session with a cluster.

**Example 5-2   Starting a session in the cluster using the Nutanix CLI**

```
./ncli -s 10.53.168.60 -u redbook -p
User Name: redbook
Enter Password:
```
Welcome, Sridhar
You’re now connected to 000571ed-8964-3268-6f7c-0cc47aeb8cfc (nx2) at 10.53.168.60
<ncli>

The <ncli> prompt indicates that an active session is ongoing. From here, many other options are available. A shortened example is the help command which shows a list of possible interactions within the Nutanix CLI. Example 5-3 shows a shortened output of the help command.

Example 5-3  Nutanix CLI help command

<ncli> help

Command Formats:
  <ncli> <entity> <action> [name=<value>] ...
  <ncli> help [detailed=true|false]
  <ncli> <entity> help [detailed=true|false]
  <ncli> <entity> <action> help [detailed=true|false]

Entities and their corresponding Actions:

  software
    list | ls
    download
    upload
    remove | rm | delete
    pause
    automatic-download

  snmp
    get-status
    set-status
    list-users | ls-users
    add-user
    edit-user | update-user
    remove-user | delete-user
    list-traps | ls-traps
    add-trap
    edit-trap | update-trap
    remove-trap | delete-trap
    list-transports | ls-transports
    add-transport
    remove-transport | delete-transport

  alerts | alert
    list | ls
    history
    acknowledge | ack
    resolve
    get-alert-config
    edit-alert-config | update-alert-config

  virtualmachine | vm
    list | ls
Reading the output of the `help` command provides a useful tool to understand how the Nutanix CLI works. For example, to list the virtual machines (and their specifications) in the current cluster, the following command can be issued as shown in Example 5-4.

**Example 5-4  List virtual machines with the Nutanix CLI**

```
<ncli> virtualmachine list
```

```
Id                        : 000571ed-8964-3268-6f7c-0cc47aeb8cfc::135668bf-e7d3-449f-9eab-cf9ca59336f9
Uuid                      : 135668bf-e7d3-449f-9eab-cf9ca59336f9
Name                      : luke-ubuntu
VM IP Addresses           : 9.53.170.6
Hypervisor Host Id        : 000571ed-8964-3268-6f7c-0cc47aeb8cfc::5
Hypervisor Host Uuid      : b68fa0a7-3b5e-46d5-b908-cad0b8c3bce6
Hypervisor Host Name      : nx22
Memory                    : 8 GiB (8,589,934,592 bytes)
Virtual CPUs              : 8
VDisk Count               : 1
VDisks                    : 000571ed-8964-3268-6f7c-0cc47aeb8cfc::NFS:1:0:6028
Protection Domain         :
Consistency Group         :

Id                        : 000571ed-8964-3268-6f7c-0cc47aeb8cfc::16fb850c-fc97-461b-b338-be4f47de9135
Uuid                      : ee91290b-6ca0-4dc0-9f13-35da68bb3970
Name                      : NTNX-nx23-CVM
VM IP Addresses           : 9.53.168.59, 192.168.5.254, 192.168.5.2
Hypervisor Host Id        : 000571ed-8964-3268-6f7c-0cc47aeb8cfc::5
Hypervisor Host Uuid      : cc59fa43-1e5e-4d5d-bd89-a0887cb08ed4
Hypervisor Host Name      : nx23
Memory                    : 32 GiB (34,359,738,368 bytes)
Virtual CPUs              : 24
VDisk Count               : 0
Protection Domain         :
Consistency Group         :
```

The Nutanix CLI is a powerful tool that helps to perform complex administrative tasks. A comprehensive review of its capabilities is omitted in this publication.

### 5.5 The Nutanix REST API

Prism, the Nutanix graphical user interface, already offers a quick and easy way to interact with Nutanix clusters. However Nutanix also offers a REST API to facilitate the interaction with
its services. This REST API allows the creation of automation scripts that can save even more time and effort.

In a similar way to the Nutanix CLI, the API offers an easy way to automate repetitive tasks, to extend the functionality of specific use cases, and integrate other tools. An example of this automation is the periodic generation of status reports about the performance of the Nutanix cluster.

The REST API Explorer can be accessed from the Prism GUI to review information about the endpoints and how the API requests can be executed. Requests can also be executed directly in the explorer so they can be tested before adding them to scripts.

In order to access the REST API explorer, click the Prism account name, located on the upper right corner and select REST API Explorer.

API endpoints (Figure 5-3) include clusters, nodes, images, storage pools, users, VMs, network security rules and many more options, so there is a wide range of possibilities for automation. In fact, anything that can be done using Prism, can be duplicated using Nutanix API. Due to this extended potential, developers can create custom dashboards, implement new tools or integrate configuration tools like Chef, Puppet, Ansible, and others.

Example 5-5 shows examples of retrieving the NTP servers from the cluster.

Example 5-5 Python code to retrieve NTP server data

```python
#!/usr/bin/python
"""NTP Python example."""
```
# Requests is an elegant and simple HTTP library for Python, built for human beings.
import requests

# Request headers.
HEADERS = {'accept': 'application/json'}

# Nutanix API's endpoint for NTP servers.
CLUSTER_URL = "https://9.53.168.60:9440/PrismGateway/services/rest/v1/cluster/ntp_servers"

# Execute the get request.
RESPONSE = requests.get(CLUSTER_URL,
    headers=HEADERS,
    auth=('IBMUsername', 'IBMP4ssw0rd'),
    verify=False)

# Print the response to stdout.
print(RESPONSE.json())

Example 5-6 shows the output from the execution of the Python code.

Example 5-6   Output from Python code execution
['0.pool.ntp.org', '1.pool.ntp.org']

Example 5-7 shows an example of retrieving the same data, but using the curl command.

Example 5-7   Curl command to retrieve NTP server data
curl --user IBMUsername:IBMP4ssw0rd --insecure -H "Content-Type: application/json"
-H "Accept: application/json"
https://9.53.168.60:9440/PrismGateway/services/rest/v1/cluster/ntp_servers | python -m json.tool

["0.pool.ntp.org",
 "1.pool.ntp.org"
]

Example 5-8 shows another Python example which retrieves the descriptions of the virtual machines in the cluster.

Example 5-8   Retrieving the descriptions of virtual machines
#!/usr/bin/python
"""Virtual machines Python example."""

# JSON handling module.
import json

# Requests is an elegant and simple HTTP library for Python, built for human beings.
import requests

# Request headers.
HEADERS = {'accept': 'application/json'}

# Nutanix API's endpoint for virtual machines.
CLUSTER_URL = "https://9.53.168.60:9440/PrismGateway/services/rest/v1/vms/"

# Execute the get request.
RESPONSE = requests.get(CLUSTER_URL,
                        headers=HEADERS,
                        auth=('IBMUsername', 'IBMP4ssw0rd'),
                        verify=False)

# Print the response to stdout.
for entity in RESPONSE.json()['entities']:
    print(json.dumps(entity['description'], indent=4, sort_keys=True))

Example 5-9 shows the output of the executing command in the project's cluster.

**Example 5-9 Output from previous example**

"redbook mongodb 2"
"S0 RHEL 7.2"
"redbook router"
"redbook openvpn ubuntu server"
"redbook RHEL 7.1"
"Created by HYCU_3.1.3-343-g575ff65 from host hycu (1917618e-f345-4f07-a92a-d88642d8d159) on date 2019-06-24 13:41:07 by job acff3749-f8ff-43c3-bf8c-4f26d1b8ec3c."
"redbook vm for openvpn test"
"redbook mongodb 2"
"Created by HYCU_3.1.3-343-g575ff65 from host hycu (1917618e-f345-4f07-a92a-d88642d8d159) on date 2019-06-24 13:42:32 by job 3b9faea3-c512-49bb-b467-88075e49d643."

Based on the experiments in this section, there are many possibilities to gather information about the cluster, including capacity, statistics, performance and the current status of the components.

For more information about the Nutanix APIs, a reference guide can be read at the following website:

Deployment scenarios

This chapter presents documented deployment scenarios utilizing common solutions, configured and implemented on IBM Hyperconverged Systems with Nutanix.

This chapter shows how to deploy the AIX operating system on IBM Hyperconverged Systems. This chapter demonstrates how easily is to deploy the AIX operating system.

This chapter contains the following sections:

- Overview
- How do you deploy AIX using a Cloud Ready Image by way of Prism
- Router VM
- OpenVPN
6.1 Overview

This chapter covers some examples of most command tools to deploy in Nutanix on IBM Power Systems.

6.2 How do you deploy AIX using a Cloud Ready Image by way of Prism

IBM AIX UNIX based Enterprise Operating System is secure, highly available, adapted to business needs, and can be installed on IBM Hyperconverged Systems.

AIX runs on top of the Nutanix Acropolis Hypervisor, fully virtualized, including networking and storage. The require version is AIX 7.2 T 2 plus a service pack, older versions are not supported by Nutanix.

AIX on IBM Hyperconverge Systems powered by Nutanix supports installation method from one of the following:

- AIX cloud images
- DVD ISO media
- Network Installation Management (NIM)

The best method, and the recommended one, is deploying it using cloud ready AIX disk images.

The binary compatibility is the same as IBM PowerVM® models (this is not a PowerVM solution), and therefore runs unchanged applications without recompilation from AIX 5 or later. You can run AIX workloads on the same system such as Oracle, IBM DB2, WebSphere, other custom applications.

There are a few features currently not available on the Nutanix solution:

- Dynamic Systems Optimizer
- Active Memory Expansion
- 64K page support
- Hardware accelerated SSL
- RDMA
- Concurrent LVM
- AIX live kernel update
- Transactional Memory
- DR Event Notification

The default AIX base media install configuration includes:

- Cloud init and its dependencies
- Pre install yum for piking up open source packages
- Enables dhcp when no adapter has been defined
- ISO image for AIX 7.2 level
- AIX version 7.2 with the 7200-02 Technology Level with Service Pack 7200-02-02-1810 and APARs IJ05283 and IJ06373 or later
- Recommended using AIX 7.2 TL 3 (after installing all the recommended APARs)
- Use of ghostdev requires APAR IJ10523 (call IBM to obtain the fix)

For more information, refer to the following website:
AIX cloud ready image, best approach to deploying, can be obtained from IBM Entitled System Support website or IBM Passport Advantage®. AIX can also be order with CS821 or CS822 servers using the AIX monthly subscription model provided through IBM Passport Advantage (PA: D1S5GLL) at the following website:


1. Download the AIX cloud ready image from Passport Advantage.
2. Uncompress the image using the `gunzip` command.
3. Upload the uncompressed image to the cluster. Click the Open Prism Central settings (Figure 6-1) then click Image Configuration under the Settings pane as shown in Figure 6-2.
4. From the Image Configuration pane, you see available images listed. In this case, select Upload Image as shown in Figure 6-3.

![Image Configuration pane - Managing the images and uploading an image](image)

**Figure 6-3**  Image Configuration pane - Managing the images and uploading an image

5. Upload the image file.
   a. Fill the Create Image required fields, Name and Annotation.
   b. Select Image Type: ISO.
   c. Select Storage Container: default-container (only one default container is listed).
   d. Select from Image Source to Upload a file → Click Choose File button to browse the location of the uncompressed image as shown in Figure 6-4 on page 71.
6. Create a VM with a disk cloned from the image uploaded.
   a. Click VM from the drop-down Menu.
   b. Click Create VM as shown in Figure 6-5.
   c. Fill out Create VM menu.
      i. Enter the general configuration as shown in Figure 6-6 on page 72.
ii. Enter the compute configuration details for the VM as shown in Figure 6-7.

iii. In the Disks pane → Click +Add New Disk as shown in Figure 6-8.
Figure 6-9 displays the pane to enter the disk configuration details.

![Add Disk pane - Enter the disk configuration details](image)

iv. The Volume Group does not change as it is not applicable to Power Systems.

v. Network adapters (NIC) if required can be added from a previously configured network.

7. Start the VM created.
   a. Click the VM as shown in Figure 6-10 (Table view).

![VM created and ready to start](image)

   b. Click Power on as shown in Figure 6-11.

![Power on the VM](image)

8. Login through the console using the COM1 connection for the VM.
   a. Select COM1.
   b. Click Launch Console as shown in Figure 6-12 on page 74.
9. You can now login to AIX as shown in Figure 6-13.

```
AIX Version 7
Console login: root
******************************************************************************

* * Welcome to AIX Version 7.2!
* *
* * Please see the README file in /usr/lpp/box for information pertinent to
* * this release of the AIX Operating System.
* *
******************************************************************************
```

Figure 6-13  AIX login pane

### 6.3 Router VM

You can deploy a VM and configure it as a router for managing two networks. This helps to access others VMs through a private network. For example, you can have a node on a 9.x.x.x public network, and access VMs that are on a 10.x.x.x private network. You want to do this due to the lack of public IPs or for security reasons in which case you want to control access only through one route to the private network.

This scenario configures an Ubuntu VM with two network interface cards (NIC) to work as a router. One of the NIC has access to the public network, and the other one has a private network which others VMs can connect to.

There are a few steps to configure this scenario:

- Configure the two networks with the desire parameters in Prism. The steps are described at the following website:
  
Then perform the following steps for each of the two network IP address range:

1. From Prism Click Settings → Click *Network Configuration* as shown in Figure 6-14.

![Network Configuration pane](image)

2. Click +Create Network as shown in Figure 6-15.

![Network Interfaces pane - Select +Create Network](image)
3. To continue with the creation of the private network, fill out the requested fields including selecting the *Enabled IP address management* box as shown in Figure 6-16.

![Create Network pane - Private network creation example](image)

When deploying a VM, remember to add the two networks created. Then select your preferred method to install or deploy the cloud version of your operating system, for example, 6.2, “How do you deploy AIX using a Cloud Ready Image by way of Prism” on page 68.
This section shows an example for adding two networks when creating a VM router as shown in Figure 6-17. There is a vlan.0 which is the public address (IP from public network is DHCP), and there is also a vlan.1 which is the private network (IP is set to a static).

To configure router in a VM or partition, use the following steps:

1. Check the partition lists both interfaces and has the correct IP settings running the `ifconfig` command.
2. Edit the file `/etc/ufw/sysctl.conf` and check if the following line is uncommented (#). If not, then comment it by adding the #:
   ```
   # Uncomment this to allow this host to route packets between interfaces
   net/ipv4/ip_forward=1
   ```
3. Set up the iptables by modifying or creating the `/etc/rc.local` file, and adding the following details:

   **Note:** This case assumes `enp0s3` is the interface name for the public network and `enp0s8` is the name for the internal private network. This ensures network traffic flow between private and external public networks.
#!/bin/bash
# /etc/rc.local
# Policy to drop all incoming packets
iptables -P INPUT DROP
iptables -P FORWARD DROP
# Accept incoming packets from local system and the internal network we setup
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i enp0s8 -j ACCEPT
# Accept incoming packets from external network
iptables -A INPUT -i enp0s3 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# Forward internal network packets to external network
iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
# Forward external network packets to internal network if the request was initiated by internal network
iptables -A FORWARD -i enp0s3 -o enp0s8 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# Network address translation through external network from internal network
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
exit 0
4. Add the correct permissions to the file using the following command:
$ sudo chmod 755 /etc/rc.local

You can now deploy VMs using only one NIC corresponding to the private network, for example vlan.1. After deploying the VM, you can access it through the router VM (Figure 6-18) using ssh as follows:

ssh to router VM
ssh to private network VM

![Figure 6-18 Accessing the private network using the router VM](image)
After accessing the VM, you can reach the Internet without any additional setup. For example, to update packages or to download any extra tools or scripts. Use the following command to verify access and update packages (in Ubuntu):

```
$ sudo apt-get update
```

### 6.4 OpenVPN

When creating deployment VMs with private networks in Nutanix, these can be accessed through a router VM. You can also use an OpenVPN setup which offers an open source Secure Sockets Layer (SSL) VPN. This means that OpenVPN uses certificates to encrypt traffic between a server or notebook and clients.

OpenVPN allows to setup access from your computer to the VMs deployed only with a private network. This setup allows traffic from your computer through a VPN server into the clients (VMs). This adds security through restricted access because only the computers, servers or notebooks that have certificate files can access the clients (VMs).

For setting this up, you need to deploy an image with two networks: private and public. Then generate the certificates. You can create as many certificates as you need.

Distribute the client certificate files to clients wanting to access the VPN. For Mac clients, you can use Tunnelblick software. On Windows or Linux, there is an equivalent OpenVPN client software available. These software use client certificates and are able to connect to your private network. You can play with TCP or UDP protocols according to your requirements. If one of them is slow, switch over to the other.

The following case scenario uses Ubuntu cloud 18 image deployed in Prism with two networks, one private and one public. This VM also issues the certificates for connecting securely to clients.

There are a few prerequisites before starting the OpenVPN VM configuration:

- Deploy a Ubuntu 18.04 cloud image to host OpenVPN
- The user must have `sudo` privileges
- Setup the firewall with the following:
  ```
  #allow SSH connections
  $ sudo ufw allow OpenSSH
  #enable firewall
  $ sudo ufw enable
  ```
- Check both network interfaces are active (up) and are listed:
  ```
  #list networks
  $ ifconfig
  ```

#### 6.4.1 Configure an OpenVPN VM

Follow these steps to configure OpenVPN in a VM:

1. Install OpenVPN and EasyRSA.
   a. Check your partition is up-to-date
      ```
      $ sudo apt-get update
      ```
b. Install `openvpn` which is available through Ubuntu’s default repositories.
   
   ```bash
   $ sudo apt install openvpn
   ```

c. Download EasyRSA and extract the tarball.
   
   ```bash
   $ wget -P ~/. /github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz
   $ tar xvf EasyRSA-3.0.4.tgz
   ```

2. Configure variables for EasyRSA and build the certificate authority (CA).

a. Go to your EasyRSA path directory, and make a copy of the variables file example. Use this file as the base to modify OpenVPN.
   
   ```bash
   $ cd ~/.EasyRSA-3.0.4/
   $ cp vars.example vars
   ```

b. Open the file from your choice of editor, in our case `vim`, and uncomment the lines with the information you require. Do not leave them blank.
   
   ```bash
   $ vim vars
   ...
   set_var EASYRSA_REQ_COUNTRY      "Mexico"
   set_var EASYRSA_REQ_PROVINCE     "Jalisco"
   set_var EASYRSA_REQ_CITY         "Guadalajara"
   set_var EASYRSA_REQ_ORG          "IBM"
   set_var EASYRSA_REQ_EMAIL        "me@ibm.com"
   set_var EASYRSA_REQ_OU           "RedBooks"
   ...
   ```

c. Initiate the public key infrastructure.
   
   ```bash
   $ ./easyrsa init-pki
   ```

   **Note:** Using Easy-RSA configuration from: ./vars.

   init-pki complete; you can now create a CA or requests.

   Your newly created PKI dir is: /home/ubuntu/EasyRSA-3.0.4/pki

d. Create `ca.crt` (public certificate) and `ca.key` (private certificate) using the `easyrsa` script. The `ca.crt` must be on all your servers and clients to inform one another that they are part of the same trusted web.
   
   ```bash
   $ ./easyrsa build-ca nopass
   ```

   Part of the output

   ```
   ... If you enter '.', the field will be left blank.
   ----- Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
   CA creation complete and you may now import and sign cert requests.
   Your new CA certificate file for publishing is at:
   /home/ubuntu/EasyRSA-3.0.4/pki/ca.crt
   ```

e. Create a private key for the server and a certificate request file. Run the `easyrsa` script and select name for the machine. For this example, and general purpose use `server`.
   
   ```bash
   $ ./easyrsa gen-req server nopass
   ```

   Part of the output

   ```
   ... If you enter '.', the field will be left blank.
   ```
-----

Common Name (eg: your user, host, or server name) [server]:

Keypair and certificate request completed. Your files are:
  req: /home/ubuntu/EasyRSA-3.0.4/pki/reqs/server.req
  key: /home/ubuntu/EasyRSA-3.0.4/pki/private/server.key

f. Copy the files generated to the openvpn directory

```bash
$ sudo cp ~/EasyRSA-3.0.4/pki/private/server.key /etc/openvpn/
```

g. Run the `easyrsa` script to sign the request, with request type `server` followed by the machine name used.

```bash
$ ./easyrsa sign-req server server
```

Part of the output

```
Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/ubuntu/EasyRSA-3.0.4/pki/issued/server.crt
```

h. Copy ca.crt and server.crt previously generated to the openvpn directory.

```bash
$ sudo cp /home/ubuntu/EasyRSA-3.0.4/pki/issued/server.crt /etc/openvpn/
$ sudo cp /home/ubuntu/EasyRSA-3.0.4/pki/ca.crt /etc/openvpn/
```

i. Create a strong diffie-hellman key, public-key protocol for sharing keys over the public channel. The script takes a few seconds to run.

```bash
$ ./easyrsa gen-dh
```

j. Generate a HMAC signature to strengthen the TLS integrity verification for the server.

```bash
$ openvpn --genkey --secret ta.key
```

k. Copy the following files created from previous command to the openvpn directory.

```bash
$ sudo cp ~/EasyRSA-3.0.4/ta.key /etc/openvpn/
$ sudo cp ~/EasyRSA-3.0.4/pki/dh.pem /etc/openvpn/
```

3. Create the certificates and keys for the client.

These steps generate a certificate for one client. Keys can be generated as needed.

a. Create the directory where the client keys will be stored,

```bash
$ mkdir -p ~/client-configs/keys
```

b. Give appropriate permissions as a security measure to lock down the directory.

```bash
$ chmod -R 700 ~/client-configs
```

c. Generate the key from the EasyRSA directory, using the `easyrsa` script, with the name of desired client. For this case, client1 is the VM in the private network.

```bash
$ ./easyrsa gen-req client1 nopass
```

Part of the output

```
If you enter '.', the field will be left blank.

-----

Common Name (eg: your user, host, or server name) [client1]:
```

Keypair and certificate request completed. Your files are:
  req: /home/ubuntu/EasyRSA-3.0.4/pki/reqs/client1.req
key: /home/ubuntu/EasyRSA-3.0.4/pki/private/client1.key

d. Copy the client1.key file to the client config directory previously created.
   $ cp pki/private/client1.key ~/client-configs/keys/

e. Sign the certificate. Be sure to specify that is a client request.
   $ ./easyrsa sign-req client client1
      Part of the output
      ...
      Write out database with 1 new entries
      Database Updated

      Certificate created at: /home/ubuntu/EasyRSA-3.0.4/pki/issued/client1.crt

f. Copy files to the keys directory for clients previously created. Client certificates *.crt, ta.key, and ca.crt.
   $ cp /home/ubuntu/EasyRSA-3.0.4/pki/issued/client1.crt
      ~/client-configs/keys/
   $ cp ~/EasyRSA-3.0.4/ta.key ~/client-configs/keys/
   sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/

4. Configuring the OpenVPN service.

Start configuring the service with certificates and keys created for both client and OpenVPN server.

a. Make a copy of the sample configuration file, and copy it to the OpenVPN configuration directory.
   $ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
      /etc/openvpn/
   $ sudo gzip -d /etc/openvpn/server.conf.gz

b. Use your text editor to uncomment (remove ";"), modify or add the following lines:
   $ sudo vim /etc/openvpn/server.conf
      ...
      tls-auth ta.key 0 # This file is secret
cipher AES-256-CBC
#auth directive HMAC message digest algorithm
auth SHA256
dh dh.pem
user nobody
group nogroup

5. Perform the network configuration for the OpenVPN VM.

a. Modify the file sysctl.conf to allow IP forwarding. This determines where the IP traffic must be routed.
   $ sudo vim /etc/sysctl.conf
      ...
      net.ipv4.ip_forward=1
      ...

b. Set the firewall rules by adding the following lines to the file before.rules, making sure that the interface used is for you public network, in this case enp0s1.
   $ sudo vim /etc/ufw/before.rules
      ...
      # ufw-before-forward
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to wlp11s0 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/24 -o enp0s1 -j MASQUERADE
COMMIT
# END OPENVPN RULES

…
c. Allow ufw to forward packets. Modify the ufw file.
$ sudo vim /etc/default/ufw
…
DEFAULT_FORWARD_POLICY="ACCEPT"
…
d. Run the following commands to allow OpenVPN traffic:
$ sudo ufw allow 1194/udp
$ sudo ufw allow OpenSSH
$ sudo ufw disable
$ sudo ufw enable

6. Start and enable the OpenVPN Service.
a. Use the system utility to start the OpenVPN service. Our configuration file (/etc/openvpn/server.conf) name is server. Hence @server is added at the end of command.
$ sudo systemctl start openvpn@server
b. Check the status.
$ sudo systemctl status openvpn@server
Part of the output
…
? openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-11-26 15:57:49 UTC; 2min 6s ago
      Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
…
c. Enable the service to start at boot.
$ sudo systemctl enable openvpn@server

7. Client configuration infrastructure.

Each client must have its own config. This part of the procedure outlines how to build the client configuration infrastructure so it can easily be replicated to all clients.
a. Create a directory to store the configuration files.
$ mkdir -p ~/client-configs/files
b. Copy the sample configuration file to use as a base.
   $ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
   ~/client-configs/base.conf

c. Edit the file by adding, uncommenting or modifying the base.conf file with the following entries. Change your_server_ip to the public network OpenVPN VM setup.
   $ sudo vim ~/client-configs/base.conf

   …
   # to load balance between the servers.
   remote your_server_ip 1194

   proto udp

   user nobody
   group nogroup

   #ca ca.crt
   #cert client.crt
   #key client.key

   #tls-auth ta.key 1

   cipher AES-256-CBC
   auth SHA256

   key-direction 1
   # script-security 2
   # up /etc/openvpn/update-resolv-conf
   # down /etc/openvpn/update-resolv-conf
   …

d. Make a script to compile your configuration based on the files created and generate a configuration.
   $ vim ~/client-configs/make_config.sh

   …

   #!/bin/bash

   # First argument: Client identifier

   KEY_DIR=~/client-configs/keys
   OUTPUT_DIR=~/client-configs/files
   BASE_CONFIG=~/client-configs/base.conf

   cat $BASE_CONFIG
   <(echo -e '<ca>')
   <(echo -e '${KEY_DIR}/ca.crt')
   <(echo -e '</ca>
   <(echo -e '<cert>')
   <(echo -e '${KEY_DIR}client.crt')
   <(echo -e '</cert>
   <(echo -e '<key>')
   <(echo -e '${KEY_DIR}client.key')
   <(echo -e '</key>
   <(echo -e '<tls-auth>')
   <(echo -e '${KEY_DIR}/ta.key')
   <(echo -e '</tls-auth>')
   > ${OUTPUT_DIR}/${1}.ovpn
e. Give permission to the script to be executable.

$ chmod 700 ~/client-configs/make_config.sh

8. Generate the client configuration.

Now generate a config file from client1.crt and client1.key from previous steps, using the script `make_config.sh`.

a. Move to the directory of the script location and run with parameter client1.

$ cd ~/client-configs
$ sudo ./make_config.sh client1

b. File generated *.ovpn can be moved to the server or computer that you want to use to login to the private network.

$ ls /home/ubuntu/client-configs/files
Part of the output
client1.ovpn
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- Red Hat OpenShift and IBM Cloud Paks on IBM Power Systems: Volume 1, SG24-8459

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Nutanix Platform Overview
- Nutanix Virtual Machine Management
  https://bit.ly/37ML0xN

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services
To determine the spine width of a book, you divide the number of pages in the book. In this case, you would use the formula:  

\[ \text{Spine width} = \frac{\text{Number of Pages}}{\text{Paper PPI}} \] 

For example, a 250-page book using 50# smooth paper with a PPI of 526 would have a spine width of:  

\[ \frac{250}{526} = 0.4752" \]

In this case, you would use the .5" spine. Now select the spine width for the book and hide the others:

**Special>Conditional Text>Show/Hide>SpineSize (-->Hide:)**

Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and:

**File>Import>Formats**

Conditional Text Settings (ONLY!) to the book files.

Implementing and Managing a High-performance Enterprise Infrastructure

**High-performance infrastructure**

8461spine.fm
To determine the spine width of a book, you divide the number of pages in the book, in this case 526, by the paper PPI or 1052, which results in a spine width of .4752". In this case, you would use the .5" spine. Now select the spine width for the book and hide the other spine widths. Move the changed Conditional Text Settings (ONLY!) to the book files. Import the Conditional Text Settings (ONLY!) to the book files.

To implement and manage a high-performance enterprise infrastructure with IBM, implementing and managing a high-performance enterprise infrastructure with IBM.