vmware® PRESS

VMware NSX® for Disaster Recovery

Day 1

Brad Christian, VCDX#217, VMware
Sean Howard, VCDX#130, VMware
William de Marigny, VCIX, VMware

Foreword by **Paul Byrne**, VP of SDDC Systems Engineering, VMware

VMware NSX® for Disaster Recovery

Day 1

Brad Christian, VCDX#217, VMware
Sean Howard, VCDX#130, VMware
William de Marigny, VCIX, VMware

Foreword by **Paul Byrne**, VP of SDDC Systems Engineering, VMware

vmware® PRESS

VMWARE PRESS

Program Managers

Katie Holms
Shinie Shaw

Technical Writer

Rob Greanias

Graphics Manager

Sappington

Production Manager

Sappington

Warning & Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors, VMware Press, VMware, and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

The opinions expressed in this book belong to the author and are not necessarily those of VMware.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA
Tel 877-486-9273 Fax 650-427-5001 www.vmware.com.

Table of Contents

List of Figures

List of Tables

About the Authors

Brad Christian, VCDX#217 is a senior systems engineer in the Networking and Security Business Unit at VMware.

Brad has worked as a systems engineer at organizations ranging from the Fortune 500 to boutique development shops to big SLEDs. He has been using VMware technology in production since the 3.x days. In 2011, Brad took over the Dallas-Fort Worth VMware User Group and grew it to one of the largest world-wide. He has earned the distinction of vExpert 6 times and became one of the first NSX vExperts in 2017. He often presents at VMUGs and other events and loves mentoring SEs.

Brad has earned the VMware Certified Design Expert designation and is VCDX #217.

You can sometimes find him on twitter at @ vhipster or (rarely) blogging at vhipster.com, but he is usually reading Sci-Fi/Fantasy novels or building Minecraft servers for his kids. Brad lives in Frisco, TX.

Sean Howard, VCDX#130 is a 15 year veteran of the IT field. During this time he has specialized in several areas including software development, storage, networking, and virtualization.

In addition to the VCDX-DCV, he holds a Bachelor of Science in Information Systems from Excelsior College. Now a systems engineer manager for the Network and Security Business Unit at VMware, he runs a team that helps customers solve some of their most challenging infrastructure problems.

William de Marigny, VCIX is a 13-year veteran of the IT industry. During this time, he has specialized in service provider scale deployments in several areas including storage, network, virtualization, and managed backup.

William is a senior NSX technical account specialist based in San Antonio Texas and holds many VMware certifications including the VCP5-Cloud, VCP-NV(5,6), VCP-DCV(4,5,6) VCAP5-DCA/DCD,VCAP6-DCA, VCIX6-NV, VCIX6-DCV.

He is a three time vExpert and two time NSX vExpert.

Reviewers

A big thanks to the reviewers, who helped fact-check and keep us honest:

Fouad Sethna

Jaden Narbut

Suds Pathi

Acknowledgements

To the lonely computer widows, our wives!

Soon-to-be Mrs. Angelica Howard

Kristin Christian

Catherine de Marigny

Foreword

I have been in the IT business for thirty years. Disaster recovery (DR) has always been a complex topic, with outdated run books and a re-ip addressing nightmares. At the end of the day, it is about the cost incurred by having applications down. The economics can run into the millions or even billions of dollars.

Enter network virtualization with NSX. What if you could simplify the network topology to avoid changing IP addresses by stretching the network to a DR site? What if you could reduce the recovery time objective and test the DR plan during the day with no impact to production?

The authors show us with amazing clarity the what, why, and how of how to use network virtualization to greatly simplify disaster recovery. Designing the proper solution for your environment is paramount. The authors will show you all the things you need to consider in a clear and concise manner. This is a must read for all networking professionals who need to automate and simplify their DR plans.

Paul Byrne
Vice President, Worldwide Systems Engineering
Software Defined Data Center
VMware, Inc.

Abstract

VMware NSX for Disaster Recovery - Day 1 brings together the knowledge and guidance for planning, designing, and implementing a disaster recovery architecture for the software-defined data center that meets the needs of your business. VMware NSX simplifies the DR planning and testing that goes into a resilient infrastructure and drastically reduces the time it takes to recover from an event. It enables true workload portability between data centers, private clouds or public clouds. NSX has helped enterprises recover from natural disasters and outages as well as simplifying the mergers and acquisitions of organizations and their networks. *VMware NSX for Disaster Recovery - Day 1* is your roadmap to create a robust network infrastructure within software-defined data centers running NSX. You will find insights and recommendations proven in the field for moving your organization to a resilient, highly available architecture based on VMware NSX.

Introduction

It is a sad fact that many enterprises that experience a disaster go out of business. All too often, disaster recovery is seen as a cost center and not given the priority it deserves, particularly over time. As recently as 2014, the average cost to businesses for a downtime disaster event was over $418,000. Something as simple as an email or web outage can run between $11,000 to $47,000 per event. Even if the value of disaster recovery planning is understood by the business, it can be very difficult to map business requirements to technical solutions.

VMware NSX for Disaster Recovery - Day 1 will explain how to overcome these hurdles. It will show how VMware NSX® simplifies the DR planning and testing for a resilient infrastructure and how it drastically reduces the time it takes to recover from an event. Doing nothing is the easy and inexpensive solution at first glance. There are no equipment, personnel, or software costs incurred; however, doing nothing only works until the moment a disaster actually happens. And it is guaranteed to happen; it is only a matter of time.

Blue Gulf Logistics

It is common for an IT department to ask the business how much downtime can be tolerated. Inevitably the answer is, “None!” – at least until the cost of a zero-downtime solution is discussed. To illustrate how to get past this vital first step, the example of a fictitious company called **Blue Gulf Logistics** will be used to demonstrate how an architect can discover the requirements for a true DR solution using NSX. The Blue Gulf Logistics example design decisions will provide insight into the journey to protect the enterprise.

Designing a DR Solution

Spot solutions abound in IT. VARs and solution providers can and will sell software or hardware to solve portions of any DR problem that exists, but these only solve part of the problem. Instead of discovering the requirements, constraints, and risks that meet the needs of a business, engineers are often left guessing on what truly must be addressed. This chapter will cover the importance of input from the business and define the architectural principles that go into a complete DR solution.

Overview of NSX and Virtual Components

Much like VMware vSphere®, VMware NSX has evolved into a platform used by many products. It has countless use cases; the primary ones – security, automation, and disaster recovery – will be discussed.

The basic building blocks of an NSX design – logical routing and switching – will be examined in a later chapter. Other products that dovetail with NSX (e.g., SRM) will be explained in greater detail.

NSX and Virtual Components Resiliency

With the introduction of more complex virtualized components, questions of resiliency and recovery become more important. This book examines and identifies the protection and recovery mechanisms of the components necessary for an NSX disaster recovery solution. Understanding these mechanisms will provide the knowledge required to properly design a highly resilient solution.

Physical Network Considerations

Software-defined networks still need a robust physical network fabric, also referred to as a network underlay. NSX cannot replace a well-defined and maintained physical network; without the physical layer, NSX cannot properly function. This chapter will show how a proper underlay can support NSX, and how NSX can stretch layer 2 over layer 3. NSX introduces the concept of universal constructs, which will be described in detail in later chapters. Additionally, the architectural differences between active/passive and active/active will be reviewed.

On-Premise DR Automation Solutions with NSX

As the marketplace and business environment have become faster, more complex, and more integrated, maintaining existing legacy solutions for disaster recovery is proving to be expensive, time consuming, and skill intensive. Businesses can no longer afford to be down for weeks, days, or even hours; the permissible time window for recovery continues to contract. As time windows shrink, automated solutions are the only viable answer; only they can respond fast enough and be agile enough to allow for rapid recovery.

VMware Site Recovery Manager™ provides the automation tools needed to allow businesses to capture the steps necessary to recover from a disaster. These steps can be outlined as discrete recovery plans that can and should be tested on a regular basis.

Security Design

Security compliance is an important part of any IT design and must be maintained even during disaster recovery scenarios. Tools such as VMware NSX can provide this security compliance with minimal effort during these events.

Backup Planning

No technical book is complete without a section on how to back up the solution. SRM and NSX contain multiple components and constructs that must be protected. This section will discuss methodologies utilized to protect a disaster recovery environment and considerations to ensure a robust solution.

Designing a DR Solution

Enterprise Architecture

For those who have been handed the task of designing a DR solution for a virtualized environment, congratulations! Businesses look to those serving as architects to be experts in protecting them from operational damage and financial loss.

An architect must have the ability to step back and design a solution that best supports the company's business goals for the infrastructure. A CEO will not invest in servers, storage, networking, licensing, or structured cabling just for fun. Down in the trenches, it is easy to get caught up in the how and lose track of the why. Customers may refuse to provide the details, but all too often they are not even asked.

Enterprise architecture as a verb means "using some kind of rational methodology to ensure the decisions made on a project are the ones that best meet the needs of the actual consumers of this service". There is not one perfect process used by all enterprise architects. The important thing is to follow some kind of system that prevents unavoidable biases, laziness, and preconceptions from delivering a design that addresses the wrong goals.

A formal EA process must be followed for the same reasons that medical researchers perform double-blind studies before declaring a drug safe and effective – it is not that doctors are dumb or careless, there simply must be controls in place to prevent human fallibility from corrupting the end result.

About 20 years ago, EA was developed to address two main problems: system complexity and poor business alignment. Organizations were spending more and more money building IT systems that were increasingly interconnected and growing in complexity. At the same time, they were finding it more and more difficult to keep those increasingly expensive IT systems aligned with business need. This situation was further compounded by the rate of business and technological change.

Currently there are four main EA frameworks in use by enterprise architects when designing an IT solution. Each has its own strengths and weaknesses, but they all seek to put order into what can be a chaotic and confusing process.

- **The Zachman Framework for Enterprise Architectures –** First developed by John Zachman in the 1980's while at IBM, this framework creates a formal structure to answer the classic questions of "why, how, what, who, where, and when" when working through the layered framework of any IT organization.

- **The Open Group Architectural Framework (TOGAF) –** The TOGAF is a framework for enterprise architecture that provides an approach for designing, planning, implementing, and governing of complex enterprise IT architecture. The TOGAF is typically a high-level design allowing for the modeling of IT architecture at four levels: business, application, data, and technology. Rather than new cutting edge techniques and technology, the framework relies heavily on existing proven technology and products.

- **The Federal Enterprise Architecture –** The FEAF is a reference enterprise architecture that can be used by federal government agencies. It provides a common approach allowing for the integration of strategic, business, and technology management when creating a design. Common in the US federal government, this architecture was first published in 1999.

- **The Gartner Methodology –** The Gartner methodology is a well-defined practice allowing for the conduction of enterprise analysis, design, planning, and implementation using a comprehensive approach ensuring a successful development and execution.

Any of these frameworks will suffice for developing a DR solution; however, there is an informal, hybrid framework that both VMware professional services and the VMware Certified Design Expert programs use. It is based largely on Zachman, but incorporates a number of VMware specific elements and is designed specifically for businesses developing a solution based on VMware products.

This chapter covers some of the most important elements for consideration when planning a DR design. Additional resources on the topic include:

1. *IT Architect: Foundation in the Art of Infrastructure Design* by John Arrasjid, Mark Gabryjelski, and Chris McCain, 2014

2. *VCDX Boot Camp: Preparing for the VCDX Panel Defense* by Ben Lin, John Arrasjid, and Mostafa Khalil, May 31, 2013

Business Impact Analysis

Discovery of Protected Applications

Before doing any design work – before looking at any products or making any decisions – it is important to establish what requires protection. Focus on the applications from start to finish; applications should govern every decision.

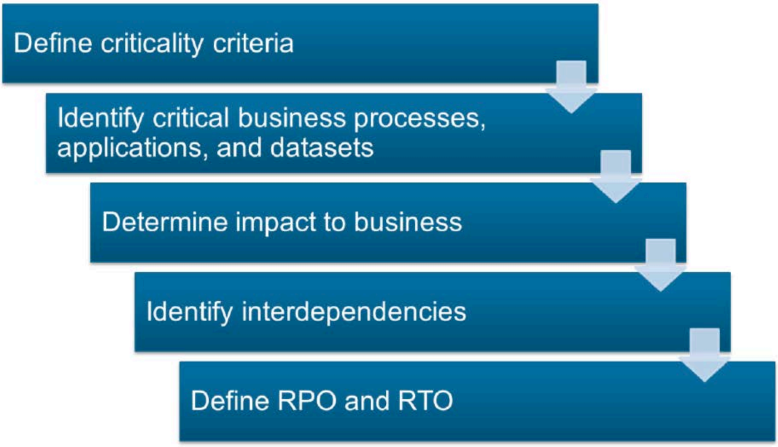
Start by doing some spelunking with the app owners, making a basic list of the apps and which lines of business they belong to.

Table 2.1 Examples of LOB applications

LOB	App Name	App Description	App Owner
Corp	SAP	Companywide ERP system	Cecilia Tucci
Corp	Exchange	Corporate and LOB Mail service	Denis Wallace
Corp	SharePoint	Company wide information sharing	Denis Wallace
Marine	Business Platform	L4 IT Data center Functions	Lamont Lockett
Marine	Ops Platform	L1-L3 Functions at customer sites	Mathew Mikus
Marine	External Systems	Core real time functions	Dwight Deasy
Land	Overland Web	Web access for customers	Lee Blanchette
Land	Overland SCADA	Supervisory Control & Data Acqstn.	Lee Blanchette
Land	Oracle Analytics	Big Data Analytics Platform	Don Durazo

Business Impact Analysis

When analyzing an outage's business impact, it is important to use a formal method for determining the central factors that will govern DR design. The following figure illustrates the stages of the process:


Define criticality criteria
Identify critical business processes, applications, and datasets
Determine impact to business
Identify interdependencies
Define RPO and RTO

This diagram shows an iterative process whereby the following are determined for each application:

Business Impact: How much does every hour of downtime cost for this application? While this seems like a simple question with a straightforward answer, in most organizations this is not the case. This is quite unfortunate because related items like RTO can only be guessed at without this information.

Business stakeholders will often declare it impossible to know such things. In the absence of a defined cost for downtime, create a placeholder figure/estimate to begin negotiations and form an understanding of what value the solution actually delivers. Fortunately, this process is straightforward; take control of the discussion and lead the horse to water, so to speak.

A simple approach would be to walk through the following formula on a whiteboard and evaluate how it aligns with their environment:

$$\left(\frac{\textit{Annual business revenue}}{\textit{Number of critical applications}}/8784\right) = \textit{cost per hour per application per year}$$

This will not be correct out of the gate, as the formula is makes many assumptions. Examples include:

1. A 1:00AM outage has the same impact as a 1:00PM outage.

2. All lines of business and applications contribute equally to annual revenue.

3. The business is only impacted by an outage inasmuch as revenue cannot be earned while the apps are not running, and that there are no soft costs

4. There is no recovery hangover period where the apps are not technically down, but people still have not logged back in and resumed their work.

Dig a little deeper to understand the specifics, then develop a more sophisticated formula from the available information. These details could be summarized as follows:

$$\left(\left(\frac{r*s}{n}/b\right)*h*d\right) = cost\;per\;hr\;per\;app\;\left[for\;this\;LOB\right]$$

Where:

r = annual business revenue for the entire business

s = percent revenue this LOB represents

b = business hours per year for this LOB

n = number of critical applications within this LOB

h = hangover factor – how long after app restoration before business truly resumes

d = day factor – what day of the outage is it?
The first, second, third, etc.

1. In this example, the company has two lines of business that contribute to the $125mm annual revenue

    a. One supporting offshore oil rigs (about 70% of revenue)

    b. One supporting oil pipelines that go overland (about 30% of revenue).

2. The team has decided to stick with the 24/7 model for the Marine LOB, and go with a 8-5 model for the Land LOB.

3. They also have an LOB called Corp that both Marine and Land share. This LOB generates zero revenue. The stakeholders have agreed to split the cost burden of Corp 50/50, thereby neutralizing its effect on these calculations.

4. They have agreed it takes about twice as long as an outage lasts to truly return to full revenue generation.

5. Finally, they also agree that if an outage is prolonged, it becomes more and more impactful, ultimately destroying the business entirely if it went on for weeks. Thus they have agreed to a linear day factor.

Thus, the formulas to calculate per hour per app for both LOBs on day 1 of an outage are as follows:

$$\left(\left(\frac{\$125{,}000,\,000*70\%}{3}/8784\right)*2*1\right) = \$6640\,per\,hour\;\;[Marine\,LOB]$$

$$\left(\left(\frac{\$125{,}000,\,000*30\%}{3}/2080\right)*2*1\right) = \$12019\,per\,hour\;\;[Land\,LOB]$$

Note that the Land LOB only incurs those losses from 8AM-5PM. So for short outages of one or two hours, the Land LOBs apps being down are actually more impactful to the business *if they happen during business hours*. If the Land LOB apps go down at night, the financial impact is zero.

Looking at longer outages, calculation for the first 24 hour day shows:

$$\$6{,}640*24 = \$159{,}360\,per\,day\,on\,the\,first\,day\,[Marine\,LOB]$$

$$\$12{,}019*8 = \$96{,}153\,per\,day\,on\,the\,first\,day\,\left[Land\,LOB\right]$$

Outside of short outages that happen during peak business hours, losing the Marine LOB's apps has twice the impact. In addition to knowing the cost per hour to justify capital expenditures to prevent the outages, there is also an understanding of relative priorities to help in building run books. It may make sense to have two separate run books – one for the day that prioritizes getting the Land LOB's applications up first, and one for the night that prioritizes the Marine LOB's apps.

Table 2.2 examines the impact of a prolonged outage lasting up to four days. In this formula, the number of outage days adds an ever-increasing impact to the business of being down for longer and longer periods.

Table 2.2 Impact of a prolonged outage

Day	Marine	Marine (cumulative)	Land	Land (cumulative)	Overall (cumulative)
1	159,817.35	159,817.35	96,153.85	96,153.85	255,971.20
2	319,634.70	479,452.05	192,307.69	288,461.54	767,913.59
3	479,452.05	958,904.11	288,461.54	576,923.08	1,535,827.19
4	639,269.41	1,598,173.52	384,615.38	961,538.46	2,559,711.98

Notice how quickly the impact starts to pile up; after four days, the business has lost $2.5 million dollars. The value of the DR solution can be easily summarized:

- A **one-hour** outage costs the business **$18,660**
- A **one-day** outage costs the business **$255,971**
- A **one-week** outage costs the business **$7,167,193**

These figures represent lost revenue; they do not account for soft costs like brand impact, lost customers, or contract penalties, which should also be accounted where possible. The formulas presented here are simply examples to help build the proper modeling mindset.

If the business cannot accommodate a meeting to come up with at least a ballpark figure, educated guesses can still be useful. Seek out contacts in other parts of the company or research public filing information; it is the job of the solution architect to determine these figures if the business will not provide them.

This is essential, as the quality of the solution depends greatly on establishing information like this; it cannot be based on gut feeling or tribal knowledge. In the event of a lawsuit, a court will not look favorably on excuses about internal barriers and communication challenges.

Run Book

A run book is the plan that will be executed when a failover event occurs. Ideally this would be an automated plan carried out by a solution such as Site Recovery Manager; however, it could be as simple as a binder full of notes detailing how to bring everything up the right way when a failover event or test occurs. This section is all about determining interdependencies.

These may be dependencies that are unique to a given application. For instance, a front-end application server VM needs its backend SQL database VM to be online first or it will not work. There could also be dependencies that are shared across applications. Classic examples include Active Directory, DNS, or a shared enterprise message bus. This type of dependency must be available before any app can be brought online. Techniques for authoring the run book should become clear while walking through the process of identifying these dependencies on a per application basis.

Recovery Time Objective

The formal definition of recovery time objective (RTO) is as follows:

> The **recovery time objective (RTO)** is the targeted duration of time and a service level within which a business process must be restored after a disaster (or disruption) in order to avoid unacceptable consequences associated with a break in business continuity.

A more succinct way to put it is, “how quickly does this application need to be functioning at minimum levels?” This is governed by the business impact quantified out earlier in the process.

Business stakeholders will often throw out requirements like “5 nines of availability”. It is important to explain the realities of the costs associated with delivering this level of service.

RTO will always be a tradeoff between instant DR site availability and the overall outage impact to the business.

Recovery Point Objective

The recovery point objective (RPO) is defined as:

> Recovery Point Objective (RPO) describes the interval of time that might pass during a disruption before the quantity of data lost during that period exceeds the Business Continuity Plan’s maximum allowable threshold or “tolerance.”

The RPO is essentially an evaluation of how much data can be lost when a disaster strikes. The level of retention required help define requirements for solution components such as storage replication technologies. A financial firm that cannot lose even a single transaction will have to use synchronous replication between site storage arrays – a very expensive proposition that usually comes with performance impact.

Customer Requirements - Design Parameters

Definition of the term "requirements"

In the context of determining design parameters, the word requirement has a specific meaning. This may differ from the way this word is commonly used in the IT profession.

Requirement: Describe in business terms the necessary properties, qualities, or characteristics of a solution. Even if they are technical elements, they relate directly to business needs.

Because these requirements are what the business is hoping to achieve with the solution, it is best that they are phrased in a way that will make sense to a business stakeholder having little familiarity with the specific technologies involved.

Examples of Bad Requirements

Table 2.3 Bad requirement that does not meet the needs of the business

ID	Requirement
REQ-01	Solution should use SRM

The requirement detailed in Table 2.3 is bad for a couple of reasons. Requirements should be structured assuming the business does not care about the underlying technology in use; they should only care that business needs are met.

In practice, business stakeholders may very well care about such specifics. They may have already purchased SRM licenses and do not want them to go to waste. Items like this fall into the category of constraints.

Table 2.4 Bad requirement that is a constraint

ID	Requirement
REQ-02	Solution must be available as much as possible

While references to availability requirements are legitimate, the one shown in Table 2.4 is non-specific and cannot be delivered in a realistic fashion. It is important to insist on specific, quantifiable parameters, otherwise the success of the deployed design can only be judged based on individual opinion. DR is serious business; consider the legal realities of defending delivery of a solution which is to be available "as much as possible".

Example of a Good Requirement

Table 2.5 Good requirement

ID	Requirement
REQ-02	Performance of applications protected by the DR solution should be tolerances specified in the relevant performance SLAs, regardless of the infrastructure recovery state. [see attachment 1 – Application performance SLAs]

References to well-defined SLAs are ideal for several reasons. SLAs directly inform important design qualities such as performance or availability. Providing specific numbers is the only practical way a solution can be designed in advance to meet expectations.

In the particular case of a DR solution, it is important to be clear how things will to work during failovers. For this example, the solution will need a comparable hardware footprint for the VMs on standby in the recovery site that perform no other function. Compare this to a business requirement that accepts performance degradation when VMs are running in the secondary site. The latter case would allow a much smaller capacity footprint to support failover scenarios, dramatically impacting the constraints of the design.

Extracting requirements from business stakeholders

Business stakeholders seldom present requirements in a way that leads to a successful design. It is incumbent upon the architect to coach and negotiate toward a set of properly phrased, well-understood requirements that all parties can agree on.

With regard to a virtualized DR solution, there are a number of questions that should be answered. These should form the basis of the requirements. The next section offers examples of the type of questions to ask; every engagement and solution is different, so this list is not exhaustive.

Primary Impact	Question
Design Qualities (Recoverability)	What are the specific applications being protected by this DR solution?

The entire design should begin and end with a focus on the applications being protected. This question is the start of the business impact analysis process, which is described in more detail later in this chapter.

Primary Impact	Question
Design Qualities (Manageability)	What automation solutions are available in the environment?

When virtual network constructs are created or modified in the primary site (e.g., DLRs, VXLANs, ESGs), it is best that matching constructs on the secondary site are updated automatically. Solutions often used to accomplish this include VMware Realize® Orchestrator™, VMware vRealize® Automation™, PowerCLI, and Python scripts.

Primary Impact	Question
Design Qualities (Manageability)	Is there already a monitoring system in place that can accurately measure RTO? Upon what metric should RTO be based?

RTO does not measure itself; DR solutions are often deployed with no explicit method for articulating downtime. A requirement may be that a given VM be pingable to stop the RTO clock. Alternatively, it might be that the RTO does not stop counting until the app is fully live and taking transactions. In this case, a monitoring system such as VMware vRealize® Operations Manager™ would be able to test solution status with end-to-end synthetic transactions.

Primary Impact	Question
Design Qualities (Recoverability)	What is the frequency for to conducting “test bubble” failovers, and how often should “true” data center failover testing occur?

It is extremely important to test failover as often as possible when using NSX and SRM. This environment catches configuration drift and makes a successful failover in a true disaster much more likely.

However, a balance must be struck as even “test bubble” failovers entail some risk. VMware Site Recovery Manager mitigates most of it, but it is possible that the run book is modified in such a way that the “test bubble” is not truly isolated from the production network. This would be bad in most environments, creating the potential for duplicate hostnames, IPs, Active Directory clients, etc. It is imperative that change control tightly manages these kinds of changes and that some form of automation is periodically validating run book configuration against the policies laid out in the design.

Primary Impact	Question
Design Qualities (Recoverability)	What is the minimum RTO (Recovery Time Objective)?

This is usually not a simple question, and there is rarely a single answer for every VM or application in a given environment. More typically, applications and their dependencies are grouped into classes with different priorities. A later section of this chapter comprehensively discusses this.

Primary Impact	**Question**
Engineering Specs (Physical Network)	How far apart are the primary and secondary sites?

The physical distance will be the primary factor governing the latency between the primary and secondary data centers due to speed of light limitations. As an example, a data center in Los Angeles and a data center in New York City might experience as much as 60ms of latency.

Primary Impact	**Question**
Engineering Specs (Physical Network)	What type of network will be used between primary and secondary sites?

The type of network between sites is a secondary factor governing latency; however, it more directly affects the reliability of the link between data centers. Where critical database transactions cross the link between sites, a relatively expensive MPLS circuit may be required to ensure reliability comparable to that within the local data center. Less important transactions might be trusted to a VPN traversing the general Internet.

Primary Impact	**Question**
Engineering Specs (Virtual Network)	Would a traffic tromboning scenario be acceptable to the protected applications?

Traffic tromboning occurs when a given L2 domain (e.g., VLAN, VXLAN) is stretched across sites while the default gateway lives in only one. Traffic for VMs at the other site must send all traffic across the WAN to reach that gateway.

Primary Impact	**Question**
Design Qualities (Security)	Are there regulatory / compliance issues?

DR projects are often initiated due to regulatory or contractual obligations. Be aware of these drivers, along with compliance requirements that would drive special security considerations such as PCI.

Primary Impact	Question
Design Qualities (Security)	Who should be allowed access to critical components of the solution? For instance, who should be allowed to initiate failover events?

Any DR solution that protects critical apps is vulnerable to various form of attack, with most of them coming from within the company. Think about how powerful someone is when they can create a snapshot of a critical VM and work on cracking it sight unseen in the DR site.

Primary Impact	Question
Design Parameters (Constraints)	What organizational policies or political factors might affect the design?

A good example of this is siloed teams. It is common to find that the owners of the physical network report through a different command structure than the people who run vSphere. These two teams have very different incentives and goals with the only communication often being through ticketing and automated systems.

It is also typical that the two teams cannot agree who should own NSX, further imperiling any project that is caught in-between. Endeavour to uncover this sort of thing early on.

Primary Impact	Question
Engineering Specs (Automation)	How automatic should the failover process to be? Is it acceptable if human intervention is required? For instance, to cut over an active/passive ESG upstream of a UDLR?

This is an example of an in-the-weeds technical question that will uncover a lot of information germane to both logical and conceptual designs. Some organizations are fine with certain failover activities requiring manual intervention; a few may even prefer this as they consider it risk mitigation against a “script gone wild” scenario.

However, most businesses want this to happen automatically as part of an SRM run book. A VRO workflow or script can easily be added to the run book to automate this process where appropriate.

Primary Impact	Question
Engineering Specs (Project Plan)	What is the project schedule like? Can the delivery be broken into phases with milestones based on something like applications protected?

Constraints

Constraints limit the features or implementation of a design. Common examples include choice of hardware vendor, budget, or project timeline considerations. They can also be organizational limitations such as a tight change control process prevents vMotion without an approved RFC.

Table 2.6 Example Constraints

ID	Constraint
CON-01	vCenter Server and ESXi hosts cannot be upgraded past 6.0.0 U1 due to a compatibility limitation introduced by an existing backup solution that uses VADP. Therefore, the maximum version of NSX that may be deployed is 6.3.5 rather than the newer release 6.4.0.
CON-02	MTU cannot be raised past 1500 on the MPLS circuit; use L2VPN – which only supports 2Gb/s – as a workaround.

Assumptions

Assumptions are expectations made when going into the design phase that cannot be initially confirmed.

For instance, a primary data center may be a new greenfield build, and there is no information that team that who deployed basic shared infrastructure did so correctly. This could yield assumptions such as:

Table 2.7 Example Assumptions

ID	Assumption
ASM-01	vCenter 6.5 instances will be fully operational and properly configured for enhanced linked mode in both primary and secondary data centers. Additionally, this will be available in time to support an on-schedule deployment of NSX Manager.
ASM-02	DNS infrastructure with properly configured PTR records will be functional in time for proper deployment of vCenter 6.5.

Risks

Risks are items that may negatively impact the reliability of a design. These can be related to people, process, or technology. Examples include: the existence of a single point of failure within the network that could impact SLA adherence; or a new technology introduced to the team, where insufficient training could hinder day to day operations.

Design choices at all levels will inevitably introduce risks. As a design is fleshed out, log all risks as an ongoing exercise.

Table 2.8 Example Risks

ID	Risks
RSK-01	NSX Manager fails and while it is being restored, an admin initiates vMotion jobs that move VMs to a different cluster that does not already have the DFW rules pushed down to it due to use of Apply To.
RSK-02	The number of DFW rules a given host receives from NSX Manager exceeds the maximum of 100,000 and overflows the heap – thus causing unpredictable forwarding behavior on that host.

Mitigations

Mitigations are not meant to eliminate risks altogether. If this were possible, that would occur in the solution architecture and there would be nothing to mitigate. Rather, they are meant to minimize the impact of specific risks called out in the design. Sometimes it is possible to reduce the impact by 90%, other times by only 10%. This is why it is important to have all business stakeholders sign off on both risks and mitigations.

Table 2.9 Example Assumptions

ID	Assumption
MIT-01	To minimize the impact of **[RSK-01]**, the NOC's run book will be updated to include a procedure for alerting all admins that NSX Manager restoration activities are underway and to freeze even simple changes like vMotion events.
MIT-02	**[RSK-02]** can be mitigated through training admins authoring firewall rules in the use of features like Apply To in DFW. Additionally, the monitoring system will have a function added that periodically initiates an SSH session to each ESXi host and screen scrapes the current rule count, alerting when it is over 8,000.

Conflicts

Once all of the requirements, constraints, assumptions, and risks are agreed upon, it is inevitable that conflicts will become apparent. It is important to raise these items to business stakeholders and track the decisions that are made. This will save tremendous heartache in the long run.

The following is an example of how to log and track such activities:

In this scenario, there were a large number of phone calls and ad-hoc meetings with the listed stakeholders throughout the development of this design, particularly during the discovery phase. Most of these meetings were not tracked or documented. There were, however, four organized meetings where stakeholders went on the record as either approving or disapproving various elements of the design.

Table 2.10 details the results of these meetings, the acknowledgement and approval of various risk conditions. Because most of these risks are introduced as a result of constraints imposed upon IT by the stakeholders, it was critical to formally validate each constraint and risk as unavoidable, and each mitigation as sufficient given the circumstances.

Finally, there were instances where requirements came into conflict, and priorities had to be agreed upon. These are also noted in the table.

Table 2.10 Example Conflicts

ID	Date	Primary Meeting Agenda Items	Stakeholders Present [STK-]
MTG-001 [1]	3/10/2017	Executive Sponsorship, Budget, Governance	001, 002, 003, 004
MTG-002 [2] [3]	4/2/2017	Service Consumer Issues	004, 005, 007, 010, 011, 012
MTG-003	6/1/2017	Business Continuity and Disaster Recovery	003, 004, 005, 006, 007, 008, 009
MTG-004	6/2/2017	Monitoring, Run Book, Change Control	004, 006, 008

[1] Conflicting requirements [REQ-008], [REQ-010], and [REQ-011] were identified and addressed in this meeting. It was clear that in many cases, it is not possible to make efficient use of physical resources and cover multiple worst case scenarios while and also keeping the configuration in line with what current SE skillsets can handle.

Therefore, it was agreed that the design should prioritize covering multiple worst case scenarios first [REQ-008], SE skillsets second [REQ-010], and efficient use of physical resources last [REQ-011].

[2] Object level RTO/RPOs were negotiated and approved with Media Business Unit stakeholders in this meeting. [REQ-007]

[3] Conflicting requirements [REQ-002] and [REQ-013] were identified and addressed in this meeting. While it would be ideal for Service Consumers to experience fully deterministic IO response times [REQ-002], the only way that can really be achieved in vSphere is by dedicating a LUN to each VM. This would raise the CAPEX for the solution above $1.2M [REQ-013]. Service consumers present in this meeting were educated on the specific mechanisms in this solution (such as SIOC) that would help with this, but only partially. Agreement was obtained from all present to prioritize [REQ-013] over [REQ-002].

Solution Architecture - Design Qualities and Decisions

Every technical decision made in a design affects one of the following design qualities; even little settings left at the default constitute design decisions.

Availability

Indicates the effect of a choice on the ability of a technology and the related infrastructure to achieve highly available operation and to sustain operation during system failures.

Table 2.11 Example decisions that impact availability

Sec.	Parameter	Decision
5.3.1	IP-HASH vs LBT	IP-HASH and Cisco VPC will be used on the VM traffic DVS
5.4.3	Admission Control Policy	Percentage of cluster resources reserved as failover capacity (13%)

Performance

Reflects whether the option has a positive or negative impact on the overall infrastructure performance.

Table 2.12 Example decisions that primarily impact performance

Sec.	Parameter	Decision
2.1.5	NIOC	NIOC will be used on the VM traffic DVS.
5.3.3	Reserved Cap. per cluster	N+1 Overhead = 70.4 GHz, 256GB RAM, 2048GB Disk Space, and 14801 IOPS

Scalability

Depicts the potential for the solution to be augmented to achieve higher sustained performance within the infrastructure.

Table 2.13 Example decisions that primarily impact scalability

Sec.	Parameter	Decision
1.2.7	UDLRs	UDLRs will be used to optimize E/W throughput, and provide a foundation for cross-data center DR
5.2.4	VM Capacity Model Params	VRAM=100%, CPU=40%, Disk=70%, IOPS=100%, 1:1 VCPU-to-logical core count

Security

Reflects whether the option has a positive or negative impact on overall infrastructure security. Can also indicate whether an option impacts ability of a business to demonstrate or achieve compliance with certain regulatory policies.

Table 2.14 Example decisions that primarily impact security

Sec.	Parameter	Decision
1.1.9	RBAC for management constructs	RBAC will be used to ensure that any given user of NSX Manager, vCenter, SRM or similar have the fewest privileges necessary
7.1.1	vCenter Hardening	vSphere 6.0 Hardening Guide Profile 3 recs where possible

Manageability

Relates the effect of a choice on overall infrastructure manageability.

Table 2.15 Example decisions that primarily impact manageability

Sec.	Parameter	Decision
8.1.4	Automated Deployments	All VMs and virtual network constructs will be deployed via VRO workflows to ensure compliance with intended design
2.1.5	DVS Traffic Shaping	DVS Traffic Shaping will not be used as it introduces unnecessary complexity given that we are using NIOC

Recoverability

Indicates the effect of a choice on the ability to recover from a catastrophic event.

Table 2.16 Example decisions that primarily impact recoverability

Sec.	Parameter	Decision
8.2.5	Failover Testing	Process and automation will be used monthly to validate DR
1.1.6	NSX Manager Backups	NSX Manager will be set to automatically back up to the central SFTP server located in the AUS2 colocation facility

Engineering Specifications – Design Documents

Architecture Design

The architecture design document is typically done in Microsoft Word with diagrams created by either PowerPoint or Visio. It is a comprehensive explanation of the proposed solution. At a minimum, it should include the following:

1. Current state analysis – breakdown of applications/tiers and other discovery data

2. Requirements – RTOs, RPOs, and other solution requirements

3. Constraints, assumptions, risks, mitigations, conflicts

4. Decisions made in the design and a justification for each

5. Diagrams and explanation of the conceptual, logical, and physical design layers

6. Capacity model – the initial capacity this solution will support and a description of how it will scale

The following figures and tables offer examples of additional content that should be included in the architecture design document.

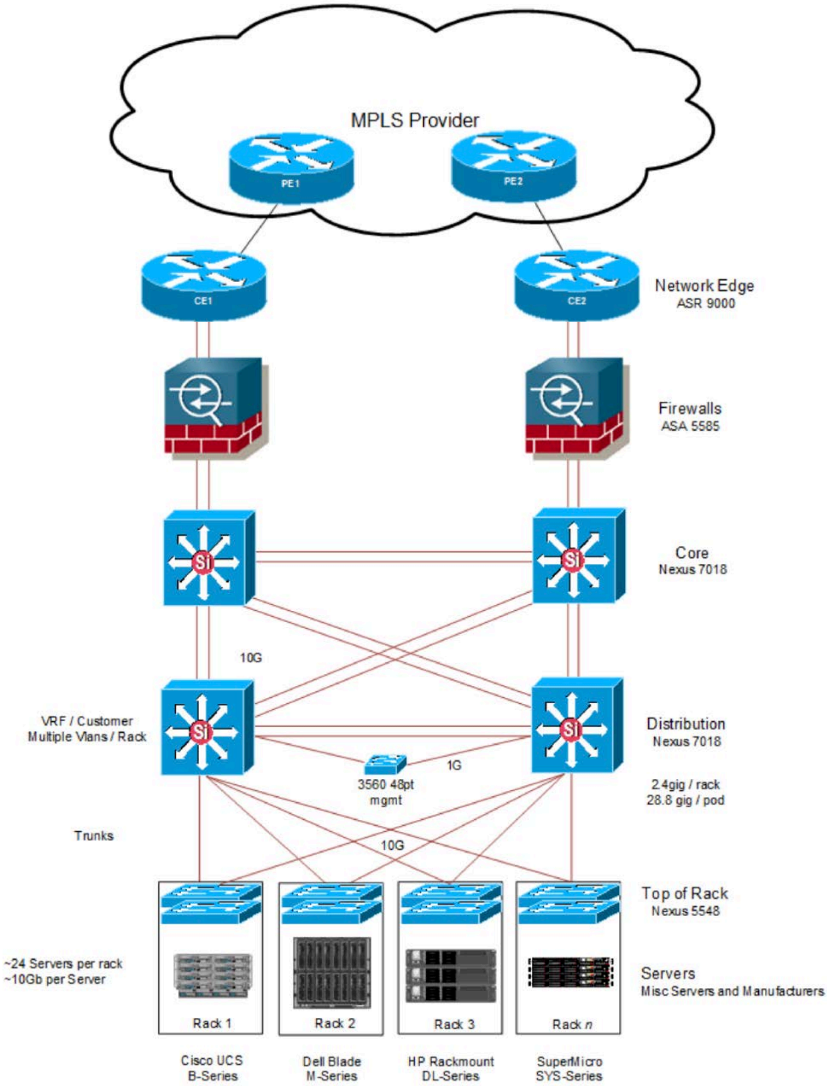


Figure 2.1 Physical Network Visio Diagram Example

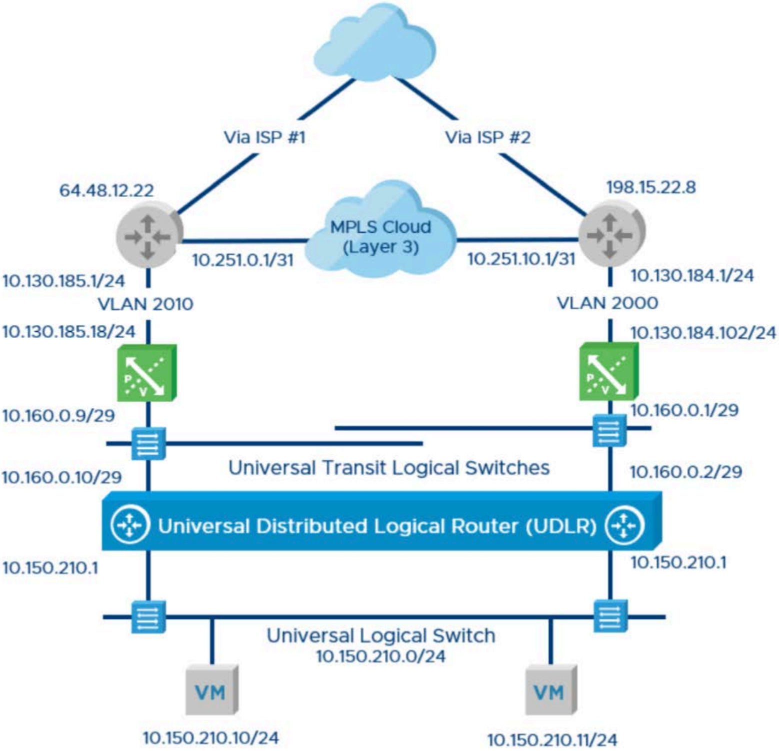


Figure 2.2 Logical Network Diagram Example

Table 2.17 Examples of LOB applications

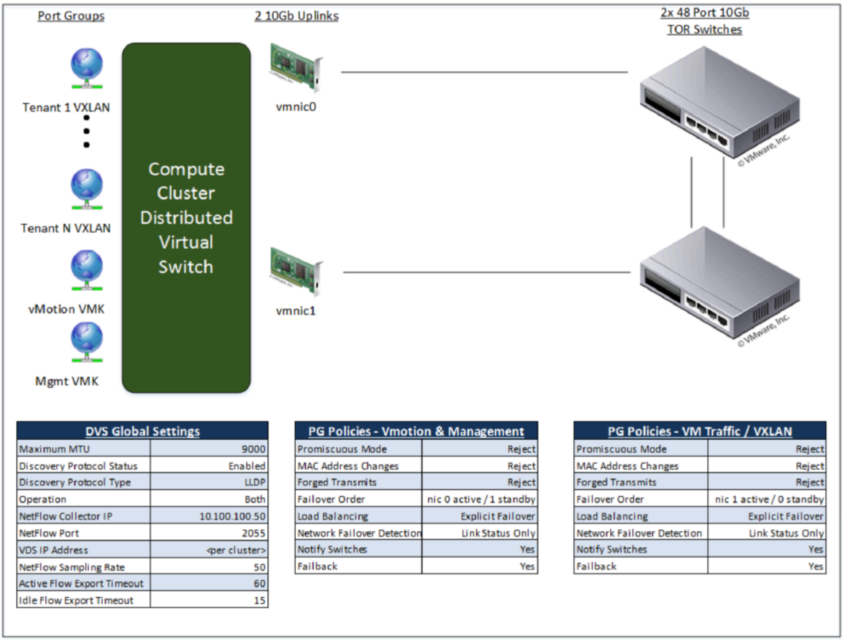
LOB Name	App Name	Tier Description	Log.Switch	# VMs
Backoffice	SAP	Web	BO-BS-PRES	10
		Application	BO-BS-LOG	4
		Database	BO-BS-DATA	2
		Enterprise Message Bus	BO-BS-MSG	3
		Maintenance task / admin	BO-BS-UTIL	2
	Exchange	Client Access	BO-EX-CAS	8
		Database Availability Group	BO-EX-DAG	8
		Mail routing	BO-EX-ROUT	2
		External facing SMTP	BO-EX-SMTP	2
	SharePoint	Entire SharePoint application	BO-SP-MAIN	2
Marine Ops	Business Platform	Web	MO-BP-WEB	17
		Application	MO-BP-APP	6
		Database	MO-BP-DB	4
	Ops Platform	Type 1 customer app interface	MO-OP-T1	8
		Type 2 customer app interface	MO-OP-T2	6
		Type 3 customer app interface	MO-OP-T3	6
	External Systems	Real time service bus	MO-ES-BUS	12
		Bus consumer #1(DCS)	MO-ES-DCS	4
		Bus consumer #2 (DCN)	MO-ES-DCN	7
		Bus consumer #3 (Machine)	MO-ES-MAC	3
		Bus consumer #4 (Safety)	MO-ES-SAF	6
		Bus consumer #5 (PLC)	MO-ES-PLC	12
Land Ops	Overland Web	Web	OL-OW-WEB	6
		Application	OL-OW-APP	2
		Database	OL-OW-DB	2
	Overland SCADA	Distributed Control System	OL-OS-DCS	8
		IOT / Device Control	OL-OS-IOT	4
		Programmable Logic Controllers	OL-OS-PLC	2
	Oracle Analytics	Reverse proxy to Forms	OL-OA-RP	8
		Web (Tomcat)	OL-OA-WEB	8
		Application (OC4J)	OL-OA-OC4J	4
		Application (CPS)	OL-OA-CPS	2
		Database	OL-OA-DB	6
		Management (OPMN)	OL-OA-OPM	1

Table 2.18 Object level RTO example

Object	Log.Switch	Externalities	Under It is control
Individual VM	30 minutes	N/A	30 minutes
All VMs in one Application	4 hours	2 hours	2 hours
All protected VMs	8 hours	2 hours	6 hours

Table 2.19 Line of Business to Logical Switch Mapping Example

App	Log.Switch	# VMs	95th% Gbps / VM	95th% Gbps all VMs
SAP	BO-BS-PRES	10	0.68	6.78
	BO-BS-LOG	4	0.50	2.00
	BO-BS-DATA	2	2.80	5.60
	BO-BS-MSG	3	0.90	2.70
	BO-BS-UTIL	2	0.56	1.11
Exchange	BO-EX-CAS	8	0.72	2.90
	BO-EX-DAG	8	0.86	6.89
	BO-EX-ROUT	2	2.27	4.53
	BO-EX-SMTP	2	0.12	0.24
SharePoint	BO-SP-MAIN	2	0.88	1.75
Business Platform	MO-BP-WEB	17	0.80	13.60
	MO-BP-APP	6	0.40	2.40
	MO-BP-DB	4	0.60	2.40
Ops Platform	MO-OP-T1	8	0.50	4.00
	MO-OP-T2	6	0.30	1.80
	MO-OP-T3	6	0.80	4.80
External Systems	MO-ES-BUS	12	0.40	4.80
	MO-ES-DCS	4	0.40	1.60
	MO-ES-DCN	7	0.30	2.10
	MO-ES-MAC	3	0.30	0.90
	MO-ES-SAF	6	0.70	4.20
	MO-ES-PLC	12	0.80	9.60
Overland Web	OL-OW-WEB	6	0.50	3.00
	OL-OW-APP	2	0.20	0.40
	OL-OW-DB	2	0.70	1.40
Overland SCADA	OL-OS-DCS	8	0.50	4.00
	OL-OS-IOT	4	0.70	2.80
	OL-OS-PLC	2	0.80	1.60
Oracle Analytics	OL-OA-RP	8	0.80	6.40
	OL-OA-WEB	8	0.40	3.20
	OL-OA-OC4J	4	0.60	2.40
	OL-OA-CPS	2	0.70	1.40
	OL-OA-DB	6	0.60	3.60
	OL-OA-OPM	1	0.60	0.60

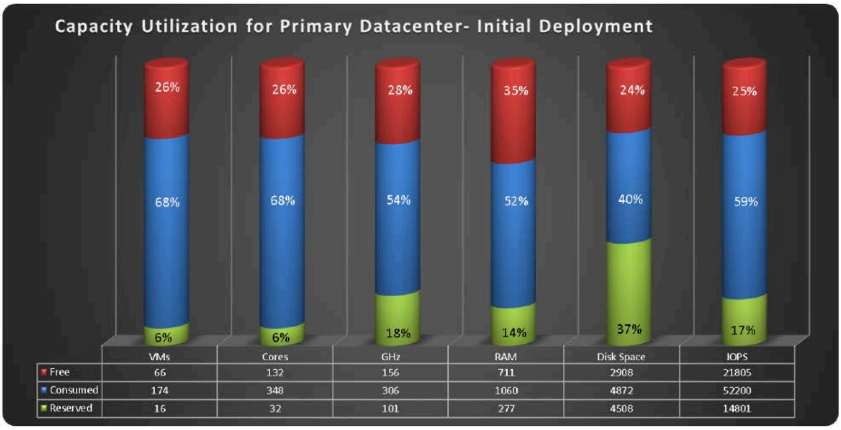


DVS Global Settings	
Maximum MTU	9000
Discovery Protocol Status	Enabled
Discovery Protocol Type	LLDP
Operation	Both
NetFlow Collector IP	10.100.100.50
NetFlow Port	2055
VDS IP Address	<per cluster>
NetFlow Sampling Rate	50
Active Flow Export Timeout	60
Idle Flow Export Timeout	15

PG Policies - Vmotion & Management	
Promiscuous Mode	Reject
MAC Address Changes	Reject
Forged Transmits	Reject
Failover Order	nic 0 active / 1 standby
Load Balancing	Explicit Failover
Network Failover Detection	Link Status Only
Notify Switches	Yes
Failback	Yes

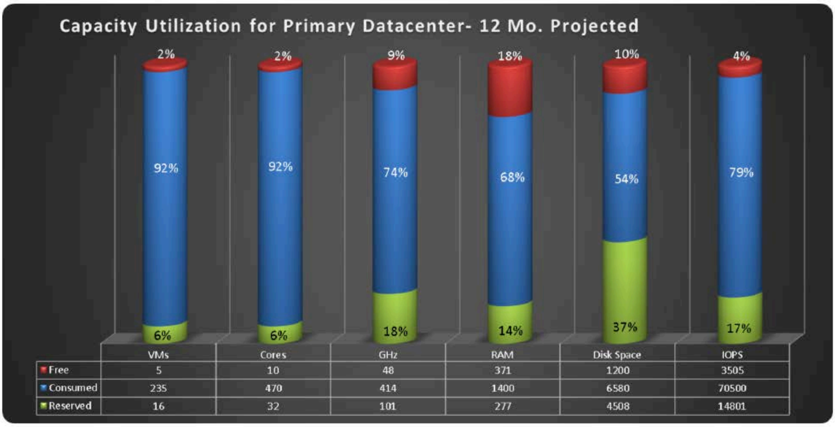
PG Policies - VM Traffic / VXLAN	
Promiscuous Mode	Reject
MAC Address Changes	Reject
Forged Transmits	Reject
Failover Order	nic 1 active / 0 standby
Load Balancing	Explicit Failover
Network Failover Detection	Link Status Only
Notify Switches	Yes
Failback	Yes

Figure 2.1 Distributed Virtual Switch Design Example



	VMs	Cores	GHz	RAM	Disk Space	IOPS
🟥 Free	66	132	156	711	2908	21805
🟦 Consumed	174	348	306	1060	4872	52200
🟩 Reserved	16	32	101	277	4508	14801

Figure 2.2 Day 1 Capacity Model Example



	VMs	Cores	GHz	RAM	Disk Space	IOPS
🟥 Free	5	10	48	371	1200	3505
🟦 Consumed	235	470	414	1400	6580	70500
🟩 Reserved	16	32	101	277	4508	14801

Figure 2.3 Day 365 Capacity Model Projection Example

Overview of NSX and Virtual Components

What is NSX

VMware NSX is a software and networking platform that delivers the same operational model to networks as VMware vSphere does for virtual machines. NSX virtual networks fully reproduce layers 2-7 of the OSI network model in software, allowing complex network topologies to be created programmatically, on demand, and independent of physical networking platforms.

In addition to replicating the OSI networking model, NSX enables security virtualization by enforcing stateful firewalling at the virtual port of virtual machines, regardless of their location within the VMware environment.

How does NSX work

NSX includes a library of logical networking services – logical switches, logical routers, logical firewalls, logical load balancers, logical VPN, and distributed security. Administrators can create custom combinations of these services in isolated software-based virtual networks, supporting existing applications without modification or delivering unique requirements for new application workloads. Virtual networks are programmatically provisioned and managed independent of networking hardware. This decoupling from hardware introduces agility, speed, and operational efficiency that can transform data center operations.

NSX Networking Capabilities

NSX provides a full library of logical components that replicate the physical network. These can be combined to create complex topologies in software, on demand, without requiring changes in configuration to the underlying physical network. The decoupling from the underlying physical network enables greater flexibility in deployment speed, operational efficiency, and programmatic deployment of networking.

This is important to disaster recovery; it allows administrators to perfectly replicate a given network on dissimilar hardware, with a dissimilar configuration, at a DR location. This replication is seamless to the VM and application layer, greatly enhancing the portability of workloads between sites.

NSX in Disaster Recovery

With the introduction of NSX into the disaster recovery model, administrators are freed from complex physical networking configurations across one or more sites. Instead, using either logical or universal components of NSX, networks from an on premises location can be replicated to a remote site programmatically – without changes to the physical network design, vendor, or code version.

Testing of DR scenarios is also simplified because the need to set up bubble networks or re-IP virtual machines is no longer required. Individual VMs can be failed over and continue to participate in an overall applications operation without the need for networking changes.

VMware NSX

The VMware NSX components are building blocks that, when combined, allow for the perfect replication of a physical network for consumption by virtual machines and applications. These components can be arranged into complex networking topologies that span data centers. This can enable disaster recovery in ways that were previously very complex, requiring extensive setup and configuration.

Logical Switches

The NSX logical switch is a construct that replicates the L2 broadcast domain of a physical network entirely within the NSX software. Virtual machines connect to these logical switches via their virtual ports and virtual NICs, perfectly replicating the characteristics of a physical network's broadcast domain (e.g., VLAN) across entire data centers. Through NSX, this is accomplished without the drawbacks of L2 sprawl or spanning tree.

A logical switch is distributed and can span clusters within one or more VMware vCenters. This allows for VM mobility within the virtual infrastructure without the limitations of standard physical L2 networking. Using vMotion, VMs may move between hosts, clusters, and vCenters while maintaining L2 networking without changes to the underlying physical network configuration.

Logical Routers

NSX logical routers provide an important enhancement to traditional routing between L2 broadcast domains. By utilizing dynamic routing at the host level, forwarding information between different L2 broadcast domains allows for direct VM-to-VM communication without the costly time needed to traverse a centralized gateway. This removal of required north/south traffic hair-pinning allows for workloads residing on the same host to instead traverse east/west across a host's system bus. This greatly reduces latency, the number of hops, and the need to enforce ACLs on a centralized gateway.

At the same time, NSX also provides north/south connectivity, thereby enabling workloads to access public networks.

Logical Firewall

The NSX logical firewall provides security mechanisms to protect network traffic using two distinct mechanisms.

The first is the distributed firewall which runs on the hosts of each NSX-enabled cluster in a virtual center. The distributed firewall allows segmentation of virtual data center objects (e.g., VMs, virtual appliances) from other virtual data center objects as well as traditional networking components (e.g., VLANs, IP addresses, MAC addresses). The enforcement of the distributed firewall occurs on the vNIC of every VM. This allows for traffic to be filtered before it hits the wire and lets rules follow the workload as it moves across the virtual environment.

The second mechanism is the Edge firewall, which acts like a traditional physical firewall appliance within the virtual data center. The Edge firewall provides load balancing and routing services while facilitating creation of DMZs based on multiple technologies:

- IP/VLAN constructs
- Tenant-to-tenant isolation in multi-tenant virtual data centers
- Network Address Translation (NAT)
- Partner (extranet) VPNs
- User based SSL VPNs

Logical Virtual Private Networks

NSX offers VPN services through three different distinct offerings. These offerings allow external users or networks to access restricted networks normally hidden behind a logical firewall.

NSX SSL VPN-Plus allows remote users to access private corporate networks. The most common use case is to allow end users access to networks behind a firewall using some sort of personal computing device (e.g., laptop, tablet, and phone).

IPSec VPN offers site-to-site connectivity between an NSX Edge instance and remote sites. The remote site could be another NSX Edge instance or a physical device.

L2 VPN allows NSX to connect two L2 broadcast domains sharing the same IP space through two NSX Edges. This allows network connectivity to extend across geographical boundaries while preserving the network configuration for workloads to consume.

Logical Load Balancer

NSX offers load balancing services via the NSX Edge. Two methods exist for NSX to load balance incoming network traffic and services. Load balancing helps achieve optimal resource utilization, maximum throughput, and minimal response time by distributing incoming connections over a pool of servers.

An in-line load balancer works by distributing incoming service requests among servers residing behind the load balancer in such a way that the load is transparent to users.

A one-armed load balancer works similarly to in-line, but it resides on a network not in-line with the incoming traffic connections. This allows the servers to be aware they are being load balanced and directly respond to client requests.

Both methods are supported with NSX load balancers and provide services up to layer 7.

Service Composer

Service Composer is an automation framework tool within NSX that allows for the assignment of networking and security services to objects and security groups within vCenter. These services are mapped to security groups and applied to the objects that reside within the group.

An additional component of Service Composer is Data Security, which provides visibility into sensitive data stored within an organization's virtualized and cloud environments. Acting on the violations reported by NSX Data Security can ensure that sensitive data is adequately protected and is in compliance with regulations around the world.

NSX Extensibility

VMware partners can integrate their solutions with the NSX platform, enabling customers to have an integrated experience across VMware products and partner solutions. Data center operators can provision complex, multi-tier virtual networks in seconds, independent of the underlying network topology or components. Administrators are able to perform complex networking, security, and data inspection operations closer to the actual workload on the hypervisor, rather than routing traffic to a centralized vendor appliance.

NSX Universal Objects

NSX 6.2 introduced the concept of universal objects to NSX to further enhance management and operations of NSX. Previously, the library of NSX components (e.g., logical switch, logical router) could only exist and operate within a single vCenter. This prevented a single L2 broadcast domain from spanning across vCenters.

6.2 introduced universal objects to address this shortcoming. It allowed objects such as logical switches to span vCenters, providing seamless network connectivity between different vCenters.

Universal components generally act like their localized counterparts; a detailed discussion of their differences is outside the scope of this book. For further information on universal objects, please refer to proper VMware documentation.

NSX Control Components

Data Plane

The NSX data plane consists of the NSX vSwitch – which is based on the vSphere Distributed Switch (VDS) – with additional components to enable services. NSX kernel modules, user space agents, configuration files, and install scripts are packaged into VIBs and run within the hypervisor kernel. These provide services including distributed routing, logical firewall, VXLAN bridging.

Control Plane

The NSX control plane runs in the NSX Controller cluster. The NSX Controller is an advanced distributed state management system that provides control plane functions for NSX logical switching and routing functions. It is the central control point for all logical switches within a network and maintains information about all hosts, logical switches (i.e., VXLANs), and distributed logical routers.

Management Plane

The NSX management plane is built by the NSX Manager, the centralized network management component of NSX. It provides the single point of configuration and REST API entry points.

Consumption Platform

NSX consumption can be driven directly through the NSX Manager user interface, available in the vSphere web client. To facilitate connection to a cloud management platform, NSX supports rich integration through REST APIs. Out-of-the-box integration is also available through VMware vCloud Automation Center, VMware vCloud Director®, and OpenStack with the Neutron plug-in for NSX.

VMware Site Recovery Manager

VMware Site Recovery Manager is a business continuity and disaster recovery solution that helps plan, test, and run the recovery of VMs between a protected vCenter Server site and a recovery vCenter Server site.

SRM orchestrates failover by associating VMware vCenter objects, allowing for workloads to migrate between network, storage, and compute segments.

SRM can be used to implement two types of recovery from a protected site to a recovery site. While the two types differ, they both use the same methodology for recovering VMs at a protected site.

Disaster Recovery Test

SRM provides for testing of recovery plans in a logical manner, allowing validation of DR plans without disrupting production workloads. During a disaster recovery test, the target components are moved to an isolated environment at the recovery site. Upon completion or failure of the test, the workloads and networks are automatically cleaned up and removed from the recovery site.

During the test, storage, RAM, and CPU are consumed at the recovery site to ensure actual resource availability. SRM testing requires both sites be up and accessible; if either site is unavailable, the SRM test will fail.

Disaster Recovery

Similar to disaster recovery test, except that disaster recovery does not require that both sites be up and running (e.g., if the protected site goes offline unexpectedly). During a disaster recovery operation, failure of operations on the protected site is reported but is otherwise ignored.

SRM orchestrates the recovery process with the replication mechanisms to minimize data loss and system down time.

For a clean failover, SRM cleanly shuts down virtual machines at the protected site and synchronizes storage. In the event of a hard failover, SRM does not shut down VMs or synchronize storage. In both cases, it powers on the replicated virtual machines at the recovery site according to a recovery plan. Where the protected site is unavailable, SRM will use the recovery plan at the DR site to bring up the VMs from the replicated storage.

A recovery plan specifies the order in which virtual machines start up on the recovery site. A recovery plan specifies network parameters, such as IP addresses, vSphere networks, datastores, and VM boot order. It can contain user-specified scripts that SRM can run to perform custom recovery actions on VMs.

Site Recovery Manager allows testing of recovery plans using a temporary copy of replicated data in a way that does not disrupt ongoing operations at either site.

SRM Components

Similar to disaster recovery test, except that disaster recovery does not require that both sites be up and running (e.g., if the protected site goes offline unexpectedly). During a disaster recovery operation, failure of operations on the protected site is reported but is otherwise ignored.

Inventory Mappings

Inventory mappings provide a convenient way to specify how Site Recovery Manager maps VM resources at the protected site to resources at the recovery site. SRM applies these mappings to all members of a protection group when the group is created. These mappings can be reapplied whenever necessary (e.g., addition of new members to a group).

Site Recovery Manager does not enforce an inventory mapping requirement. If a protection group is created without defining inventory mappings, each protected virtual machine must be created individually or through use the Configure All option. SRM cannot protect a virtual machine unless it has valid inventory mappings for key virtual machine resources, including networks, folders, compute resources, and placeholder datastores.

When SRM creates a placeholder virtual machine, it derives the folder and compute resource assignments from inventory mappings established at the protected site. A vCenter administrator at the recovery site can modify folder and compute resource assignments as necessary.

Virtual Machine Replication

Replication of virtual machines can be done in two ways within vSphere and SRM. Each has its own merits and drawbacks; these are not covered in this book, but each will accomplish the same overall task.

Datastore Replication

Virtual machines residing on a datastore are protected using a third-party disk replication mechanism that configures and executes array-based replication. During recovery, array-based replication presents replicated datastores on the recovery vCenter to recover VM workloads.

Individual Virtual Machine Replication

Individual VMs can be protected using Site Recovery Manager and VMware vSphere® Replication™. vSphere Replication replicates the individual files that compose a VM from one datastore to another, within a vCenter instance or between vCenters.

Storage Policies

Storage Policy Based Management (SPBM) is a method of assigning components such as encryption, replication, or backup to a storage policy. When a virtual machine is created, the storage policy is assigned to the VM, thus applying the individual policy components to that virtual machine.

SRM can take advantage of this by automatically protecting workloads assigned a given SPBM policy without requiring individual setup and configuration of virtual machines.

NSX Resiliency

The movement of IT-based solutions from one based solely on hardware to one based largely on software-defined constructs is many times difficult for IT professionals to accept. Those who have worked in IT anytime in the last 20 years know that for something to be resilient or fault tolerant often requires redundant hardware – be it servers, switches, firewalls, leased lines or power. The idea of resiliency via redundancy is the foundation of highly available IT services, and NSX continues this legacy via software instead of hardware.

Many will likely have doubts in place; doesn't NSX itself run on x86 servers thus providing a redundant platform? It absolutely does, but rather than duplicate pieces of hardware – such as two switches or firewalls acting in an active/active or active/passive configuration – NSX creates highly available software constructs that mimic the functionality of physical entities.

Earlier chapters discussed how NSX creates a logical library of networking constructs that replicate a physical network and its services in software on x86 servers. Many of these constructs have built-in capabilities that provide HA or resiliency mirroring the effect of duplicate physical pieces of equipment. This chapter will discuss some of these constructs and how they provide resiliency and redundancy in the software defined landscape.

How Resilient is NSX Really?

A common question is, "just how resilient is NSX and its constructs?" The answer is that for most IT networking needs, NSX provides the functionality expected for any enterprise class system. As NSX works with software defined constructs rather than dedicated special built hardware, there will to be limitations and caveats. For some situations, hardware-based solutions are still going to be the only option, but the good news is that NSX is not an either-or solution. It works with any physical network and any physical network device due to its network agnostic design.

VMware NSX Edge

The VMware NSX Edge is easily the most visible networking construct because it is the construct that ultimately will connect the physical network to the virtual network. The Edge provides a library of services including DHCP, logical firewall, load balancer, routing, and DNS among others, but its most important feature is connecting and routing virtual network traffic workloads to a physical VLAN backed network.

Being the on-ramp and off-ramp so to speak, it is vitally important to maintain the Edge in its expected state. There are three mechanisms within NSX to do so:

1. VMware HA
2. NSX Edge HA
3. NSX Edge ECMP

VMware HA

As the VMware NSX Edge is a virtual machine running within a VMware environment, it can take advantage of the protection mechanisms afforded any virtual machine in vCenter. VMware HA will restart a failed virtual machine on hosts within a cluster should a physical host, storage, or network failure be detected on an ESXi hypervisor.

One thing to make clear – VMware HA simply restarts virtual machines. This means there will be downtime for a VM until it restarts. In the case of the Edge, any services or connectivity provided via the Edge in question will be unavailable until such time that the Edge reboots on another host. Depending on the environment this downtime could be a few seconds to a few minutes.

NSX Edge HA

If the prospect of network services downtime for several minutes is unacceptable, then the Edge provides an additional HA mechanism in the form of a passive Edge in standby mode should the primary fail.

With HA enabled on a NSX Edge, a duplicate NSX Edge is deployed with a mirrored configuration of the primary active Edge. State is synchronized between the primary and secondary Edge. Should a failure be detected, the secondary Edge assumes the role of primary and resumes network services. It is important to note that the configuration is exactly replicated between Edges, negating any changes of IP, MAC, or even firewall state during transition.

The Edges utilize a heartbeat with adjustable times allowing for standby Edges to assume the primary role in as few as 15 seconds. While a better option than relying on VMware HA, the 15 seconds of downtime nonetheless are more than many customers are willing to accept.

NSX Edge ECMP

The final mechanism to provide resiliency moves the NSX Edge away from a singular deployment and towards a cluster deployment. Equal Cost Multi Path (ECMP) is a networking mechanism that allows for dynamic routing of network traffic across multiple ingress and egress points. The ultimate route is determined by internal mechanisms within the OSPF and BGB routing protocols.

By utilizing ECMP, traffic can be rerouted around failed or non-responsive Edges with nearly no disruption to network traffic, but this does have some downsides. By moving away from a centralized deployment, stateful services are disabled and will not work.

This means that firewall, DHCP, VPN, and load balancer services will not work on an Edge participating in a ECMP cluster. If the desire is to provide those services, then an additional Edge must be deployed downstream from the ECMP cluster to provide the stateful services.

VMware NSX Logical Switch

The NSX logical switch is the most basic component of NSX. In many ways, it is the easiest to understand. The logical switch is visible within the VMware environment as a distributed port group residing on a distributed virtual switch. In reality, the logical switch is simply an L2 broadcast domain present in memory on every hypervisor. Virtual machines located on a host can communicate over this broadcast domain, and even between broadcast domains, when a distributed logical router is present and configured.

Looking at the user interface, the distributed port group which represents each logical switch will be present on each host within a prepared cluster. Under the hood however, the logical switches are not present on every prepared host at the same time. Only when a connected VM moves to a host on a logical switch is it actually presented to the host. This presentation of connected logical switches is controlled by the NSX Controller cluster. The logical switches have no redundancy as would normally be expected, but instead are simply port groups on a distributed switch present on each host. Should network connectivity be lost to a VM on a host, VMware HA can be configured to restart the VM on another host that is also prepared for NSX.

VMware Distributed Logical Router (Logical Distributed Router)

The VMware DLR (or LDR) is one of the trickiest components to understand. It has a presence on each hypervisor within a prepared cluster as well as externally via a control Edge VM – or 2 if HA is present.

When a DLR is deployed within a prepared cluster (multiple DLRs can be deployed) a kernel module is loaded within each prepared hypervisor that handles routing between logical switches on each hypervisor. In a typical environment, network traffic between different broadcast domains, typically isolated using VLANs, must leave the hypervisor via its uplink and be routed upstream at either a switch or router. The DLR enables routing to be done within the hypervisor itself, saving the north/south traffic flows when VMs on the same host must communicate.

The most visible portion of a DLR is the control VM which is deployed when configuring a DLR. The control VM is simply an NSX Edge that handles the dynamic routing peering between physical and virtual networking components. For example, when setting up OSPF routing between a DLR and an upstream NSX perimeter Edge, the NSX Edge will peer with the DLR control VM. Upon learning routes the DLR Edge will pass the routes down to each DLR instance which resides on every hypervisor in a prepared cluster.

The DLR Edge is a standard NSX Edge, so it falls under the same protection mechanisms as outlined above.

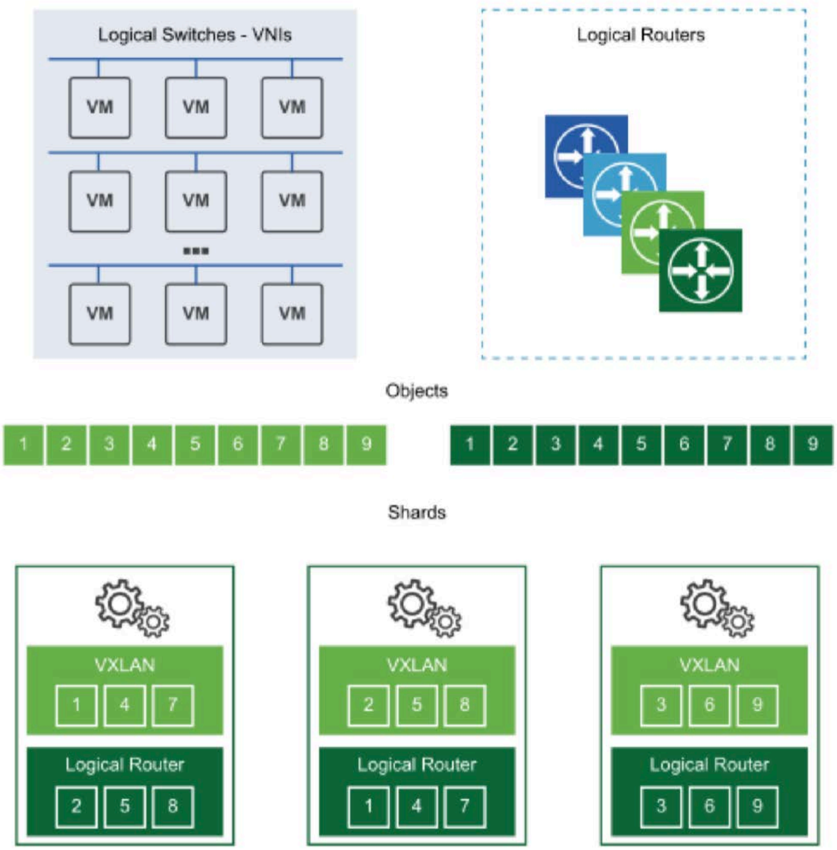
The DLR component within each hypervisor is controlled and managed by the kernel of the hypervisor itself. Should the DLR become unresponsive or crash, the kernel will attempt to restart the service but should the module have a catastrophic failure the host likely will purple screen. In this instance VM's and workloads on the host will be restarted on remaining hosts within a cluster using VMware HA.

NSX Controllers and Controller Cluster

The NSX Controller and Controller cluster is the most complex component within NSX, thus the one that needs the most explanation. The NSX Controller is the brain of NSX that knows how to route packets between hosts. The Controller is responsible for knowing the location of virtual machines, prepared hosts, and the routing between them.

The NSX Controller cluster is the deployment of three NSX Controllers working together to provide necessary information to VM workloads and hosts to enable software defined networking. The Controller cluster is set up as a scaled-out distributed system with each Controller assigned a set of roles and responsibilities. This is known as sharding, and is common when implementing scaled designs.

Sharding is used to distribute workloads across NSX Controller cluster nodes. Sharding is the action of dividing NSX Controller workloads into different shards so that each NSX Controller instance has an equal portion of the work.



The above picture demonstrates how each Controller node acts as a master for a given role such as logical switching, logical routing, or other NSX services. Once a master NSX Controller instance is chosen for a role, that NSX Controller divides the different logical switches and routers among all available NSX Controller instances in a cluster.

The numbered boxes above on the shard represent the divided workloads on each NSX Controller. The logical switch master divides the logical switches into shards and assigns these shards to different NSX Controllers. This same operation occurs for the logical router master and remaining Controller services.

The master for a role decides which NSX Controller is assigned to which shard. If a request comes in on router shard 3, the shard is told to connect to the third NSX Controller instance. If a request comes in on logical switch shard 2, that request is processed by the second NSX Controller instance.

When one of the NSX Controller instances in a cluster fails, the masters for the roles redistribute the shards to the remaining available clusters. One of the Controller nodes is elected as a master for each role. The master is responsible for allocating shards to individual Controller nodes, determining when a node has failed, and reallocating the shards to the other nodes. The master also informs the ESXi hosts about the failure of the cluster node.

The election of the master for each role requires a majority vote of all active and inactive nodes in the cluster. This is the primary reason why a Controller cluster must always be deployed with an odd number of nodes.

ZooKeeper

ZooKeeper is a client server architecture that is responsible for NSX Controller cluster mechanism. The Controller cluster is discovered and created using Zookeeper. When a cluster is coming up, it literally means ZooKeeper is coming up between all the nodes. ZooKeeper nodes go through an election process to form the control cluster. There must be a ZooKeeper master node in the cluster. This master node is chosen via inter-node election.

When a new Controller node is created, NSX Manager propagates the node information to the current cluster with node IP and ID. As such, each node knows the total number of nodes available for clustering. During ZooKeeper master election, each node casts one vote to elect a master node. The election is triggered again until one node has a majority of the votes. For example, in a three-node cluster, the master must have received at least two of the votes.

- When the first Controller is deployed, it is a special case and the first Controller becomes master. As such, when deploying controllers, the first node must complete deployment before any other nodes are added.

- When adding the second Controller, it is also a special case, because the number of nodes at this time is even.

- When the third node is added, the cluster reaches a supported stable state.

ZooKeeper can sustain only one failure at a time. This means that if one Controller node goes down, it must be recovered before any other failures. Otherwise, there can be problems with the cluster breaking.

Central Control Plane (CCP) Domain Manager

This is the layer above ZooKeeper which provides configuration for ZooKeeper on all nodes to start. Domain manager updates the configuration between all nodes in the cluster, then makes a remote procedure call for the ZooKeeper process to start.

Domain manager is responsible to start all domains. To join the cluster, CCP domain talks to CCP domain on other machines. The component of CCP domain that helps with cluster initialization is *zk-cluster-bootstrap*.

Controller Relation with Other Components

The Controller cluster is responsible for maintaining and providing information about logical switches, logical routers, and VTEPs to the ESXi hosts.

When a logical switch is created, the Controller nodes within the cluster determines which node will be master, or owner, for that logical switch. The same applies when a logical router is added.

Once ownership is established for a logical switch or logical router, the node sends that ownership to the ESXi hosts that belong to that switch's or router's transport zone. The entire election of ownership and propagation of the ownership information to the hosts is called 'sharding'. Note that ownership means that node is responsible for all NSX-related operations for that logical switch or logical router. The other nodes will not perform any operation for that logical switch.

Only one owner may be the source of truth for a logical switch and logical router. Any time the Controller cluster breaks in such a way that two or more nodes are elected as owner for a logical switch or logical router, each host in the network may have a different information regarding the source of truth for that logical switch or logical router. If this happens, there will be a network outage because network control and data plane operations can only have one source of truth.

If a Controller node goes down, the remaining nodes in the cluster will rerun sharding to determine ownership of the logical switch and logical routing.

Physical Network Considerations

Purpose

When deploying NSX, it is often said that the physical network does not matter. For the most part, that is true. VXLAN encapsulation removes the need for two vSphere hosts to have the same VLAN extended from the physical network. It enables deployment of new physical network topologies – like layer 3 leaf/spine – while allowing focus to remain only the logical network topology.

It is also a reason for interest in this book –the ability to move VMs from one site to another without changing their IP addresses. NSX makes this easy.

Even with this abstraction, the one spot where the physical network configuration does matter is at the edge of the virtual network. Regardless of its popularity, there will always be a place in the logical topology where NSX stops and the physical network begins.

This chapter addresses those considerations and offers insight into DR scenarios from existing NSX customers.

Data Center Interconnects

Multiprotocol Label Switching (MPLS)

MPLS is a highly reliable, very scalable, protocol-independent option for data center interconnect (DCI). MPLS adds labels to packets when they enter the MPLS network. It uses these labels to make forwarding decisions without the need to examine the packet itself. This allows a provider to create an MPLS cloud that may have several different L1 or L2 long distance transport mechanisms in play (e.g., SONET, ATM, Frame Relay), which are simply presented as an MPLS port in both the primary and secondary data centers.

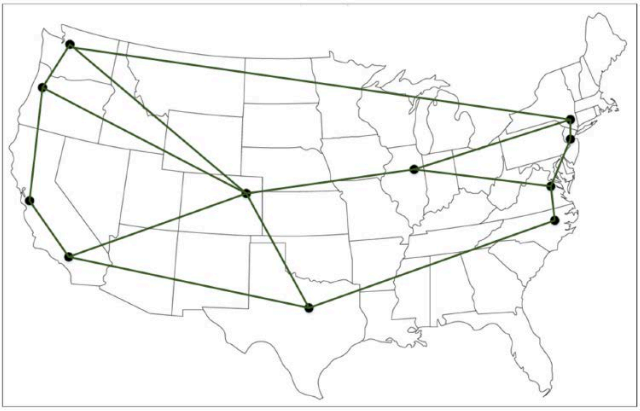


Figure 4.1 A nation-wide MPLS network

MPLS is what 90% of customers use who have successfully deployed a DR solution as a DCI. The reason is reliability. MPLS has a feature called MPLS Local Protection; this ensures ultra-fast cutover when a redundant link in the network dies. OSPF or BGP routing convergence can take multiple seconds, and spanning-tree convergence longer still. If a path in the MPLS cloud goes down, the switchover happens in milliseconds. This is fast enough to trust to important applications like business critical OLTP databases. Carriers often promise four nines (i.e., 99.99%) SLAs across their fabrics. This is likely better than what enterprises see in their own data centers – unless they have built their own private MPLS network.

The glaring downside of this approach is cost. Where a 1Gbps business class internet circuit may run $500 a month, a 1Gbps MPLS circuit in that same area may cost $50,000. Most customers who do DR have MPLS circuits on the order of 250mbps to support long distance vMotion or 100mbps for straight failover with vSphere replication.

Layer 2 MPLS (i.e., VPLS)

As the MPLS cloud is shared with of other customers, there will be a provider-managed VPN in place. Therefore, traversing the MPLS cloud almost always involves one or more L3 routing hops, also preventing use of the same subnet on both sides.

VPLS is a permutation of MPLS offered by some carriers. It is a layer 2 VPN that allow the same subnet to exist on both sides of the tunnel. The problem is that it does not scale. VPLS limits the number of MAC addresses than can participate in the VPLS, usually on the order of one or two dozen, which is not enough for most customers.

Layer 3 MPLS

Most customers use a standard layer 3 version of MPLS service, but still want to be able to easily stretch a subnet between sites. Using virtual network constructs like universal logical switches and routers, NSX allows not only subnets to span both data centers, but the virtual routers used by those subnets as well. These capabilities are possible regardless of any routing hops in the MPLS network in-between data centers.

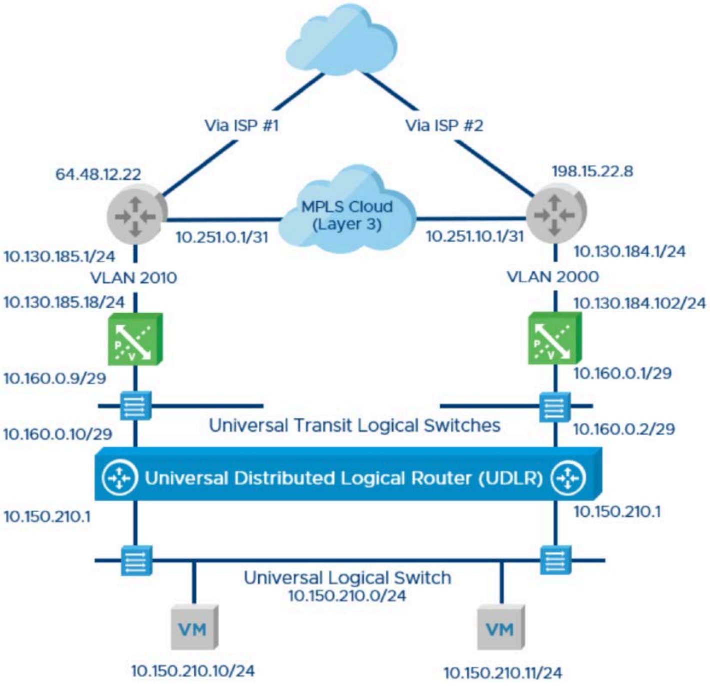


Figure 4.2 Layer 3 MPLS + NSX Implementation Example

Layer 3 VPN over the open internet

Some customers opt to use IPSEC over the open Internet for their DCI. This is far cheaper than MPLS, though also far less reliable. This approach relies on random entities on the Internet to move traffic, so there will be periods of outage, congestion, and confusion that are out of corporate control. In even the best case, expect this approach to DCI to provide no better than 99% (i.e., two nines) of availability.

Such availability numbers may be acceptable based on the overall architecture. A straight active/passive DR configuration with vSphere replication and an RPO of 15 minutes may be designed to tolerate five minute outages while relying on simply adding bandwidth to address possible congestion issues. To provide perspective, a 1Gbps Internet connection at both data centers is a drop in the bucket compared to 100Mbps of MPLS.

While the NSX Edge Services Gateway can terminate IPSEC connections, in this context, the L3 VPN should be terminated by physical network equipment, as VXLAN needs a physical L3 network to ride on top of.

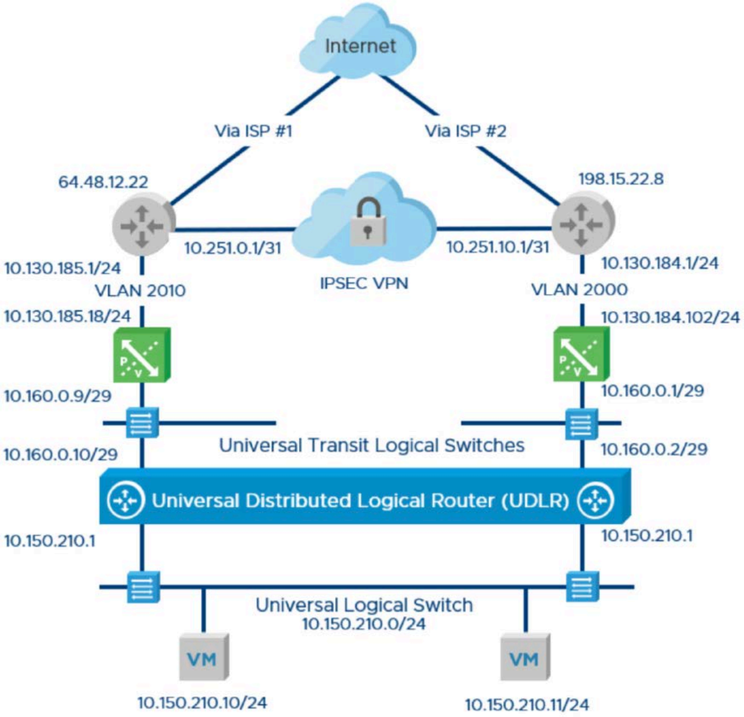


Figure 4.3 Layer 3 VPN + NSX implementation example

Software Defined Wide Area Networks (SD-WAN)

SD-WAN is an emerging technology with significant market interest. It is usually implemented as an appliance – either physical or virtual – that sits on both ends of a DCI link.

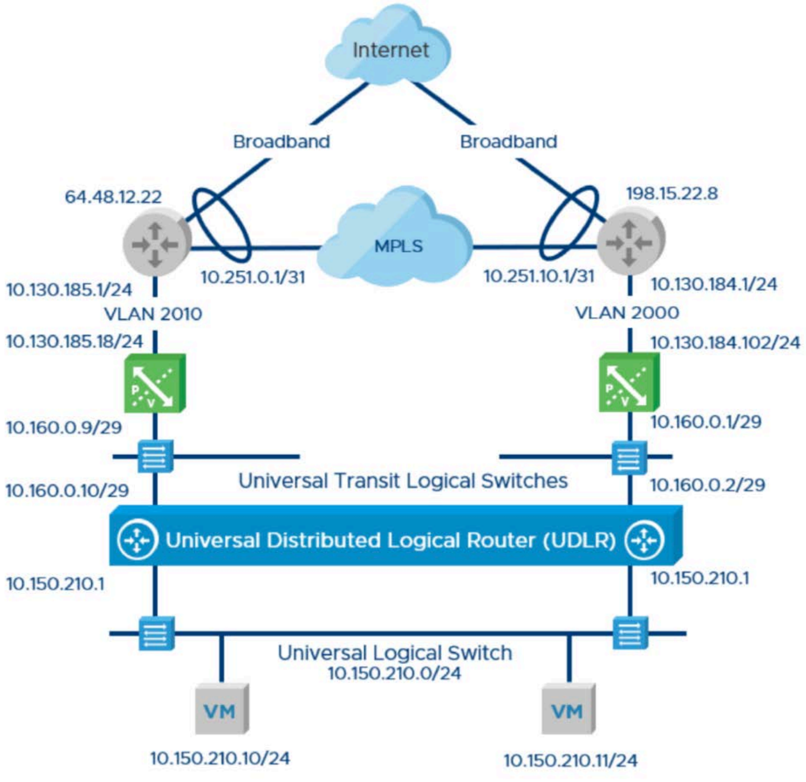


Figure 4.4 SD-WAN + NSX implementation example

Of the many features offered by SD-WAN, the one relevant to this discussion is the ability to bond multiple different DCI link types into one virtual circuit. The SD-WAN appliance then intelligently manages traffic across the links. It restricts use of MPLS to traffic with strict performance requirements while using IPSEC over the Internet for flows that are less critical. Some solutions, such as NSX SD-WAN by Velocloud, combine inexpensive broadband links from multiple providers into a single virtual circuit, approaching the reliability of MPLS with a much lower price point.

MTU Considerations

Most MPLS providers and devices used to terminate IPSEC are capable of raising the MTU to 1600. This is required to accommodate the extra header that VXLAN wraps around packets when it spans a subnet between data centers.

However, there are times where the MPLS provider is intransigent or the IPSEC equipment cannot do this. In those cases, it is possible to work around this issue by employing a WAN optimization appliance (e.g., Silver Peak, Riverbed) that is capable of fragmenting and reassembling the packets as they traverse the MPLS cloud or Internet.

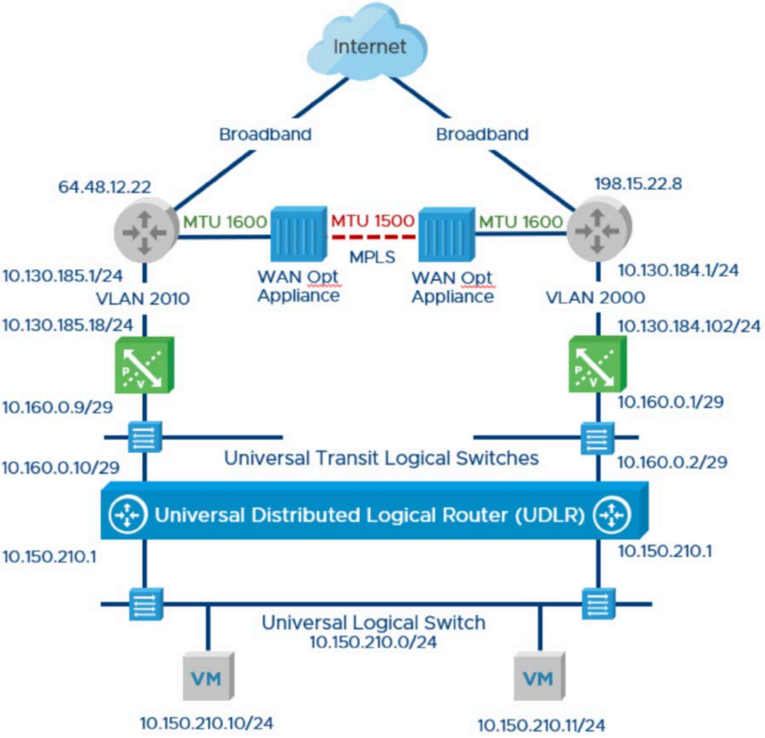


Figure 4.5 MTU 1600 achieved with Wan Optimization Appliances

Dynamic routing protocols

Why use a dynamic routing protocol?

While it is possible to use only static routes with NSX, VMware advises against this as it goes against one of the main benefits of network virtualization – the ability automatically to create network constructs on the fly without having to touch the physical network. When using static routes, any time new logical switch with a new subnet is added, configuration changes must be made on physical routers. A new route must be added so that physical devices know how to reach the newly created subnet.

When using of a standard dynamic routing protocol such as OSPF or BGP to create a new subnet inside the virtual fabric, routes to it will automatically appear in the physical routers' route tables.

Open Shortest Path First (OSPF)

OSPF is a standards-based, non-proprietary routing protocol. It is widely used by enterprises for internal network routing due to its ease of implementation and management. One downside is its lack of scalability beyond a few dozen devices.

Border Gateway Protocol (BGP)

BGP is another a standards-based, non-proprietary routing protocol. It is preferred by organizations with significant scalability concerns within their data centers (e.g., Amazon, Google, Microsoft, Rackspace). BGP is also the foundational protocol for public Internet routing. A full feed of BGP routes contains approximately 750,000 prefixes spread across 60,000 autonomous systems. The full database of this routing information can consume 2GB of RAM on each router running the protocol.

A secondary benefit of BGP is its configurability; compared to OSPF, it offers a much larger set of advanced settings that can be used to fine tune behavior.

Inbound access scenarios

Assumptions

This section builds on the assumption that the majority of applications deployed are commercial off-the-shelf software. If a majority of applications in question are written in-house, specific concepts discussed (e.g., multi-master topologies) may not be applicable.

Traffic from Trusted Networks

A trusted network can be defined as any network behind a perimeter firewall that needs to access applications at the secondary site when a failover occurs. Trusted networks generally consist of physical desktops or other devices that are located inside one or more corporate offices.

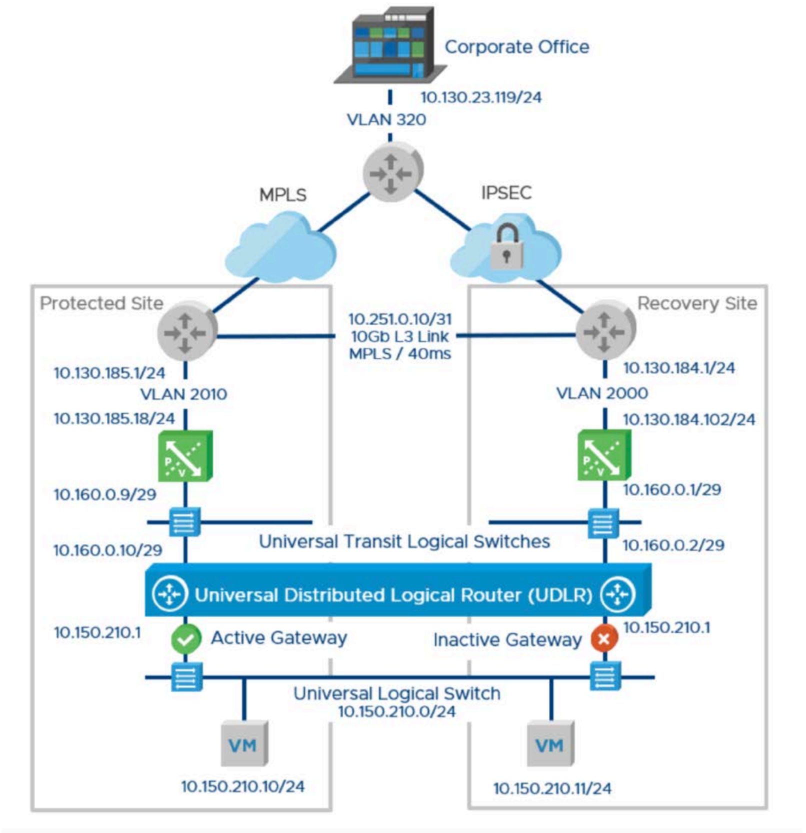


Figure 4.6 Inbound Access from Trusted Networks Example

A common connectivity question asks how clients in a corporate office know how to reach remote VMs that are capable of moving between data centers. The most common solution is use of an active/passive default gateway model which spans a subnet/logical switch across the sites using a universal distributed logical switch (i.e., UVXLAN) that sits below a universal distributed logical router (UDLR).

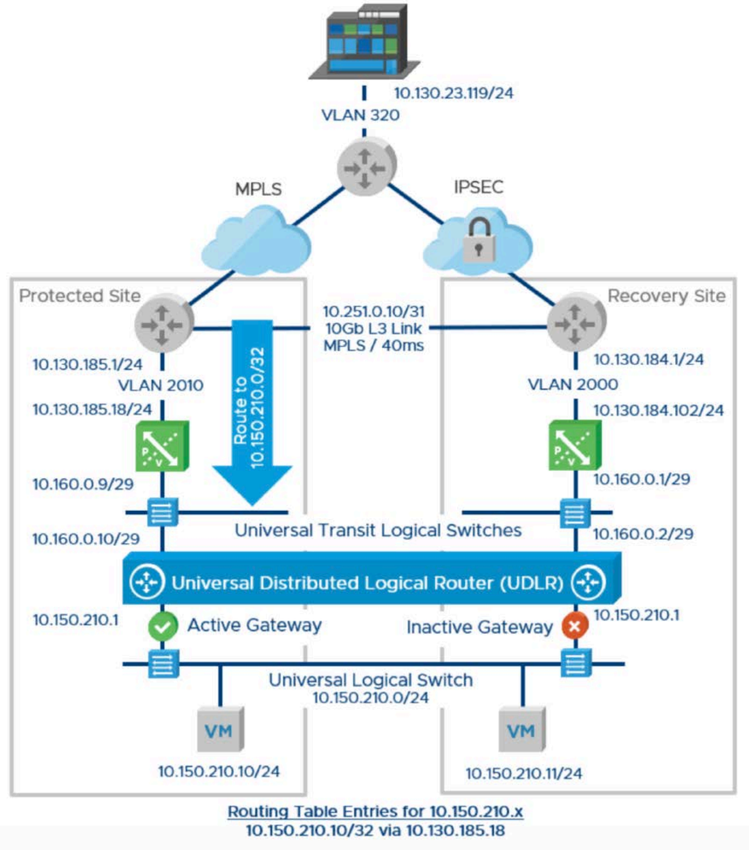


Figure 4.7 Active/Passive Under Normal Conditions

In Figure 4.7, while both sides of the UDLR host the default gateway for the 10.150.210.0/24 subnet, only the instance in the primary data center is active (i.e., forwards traffic) under normal conditions. The two VMs currently at the secondary site – IP addresses ending in .10 and .11 – must send their traffic across the DCI link to reach their default gateway. All inbound traffic for the subnet routes through the ESG with address 10.130.185.18. This situation is known as traffic tromboning. In this case, VMs at the secondary site experience an additional 40ms of latency to reach their default gateway compared to those located at the primary site.

Figure 4.8 updates the environment to reflect state after a full failover event.

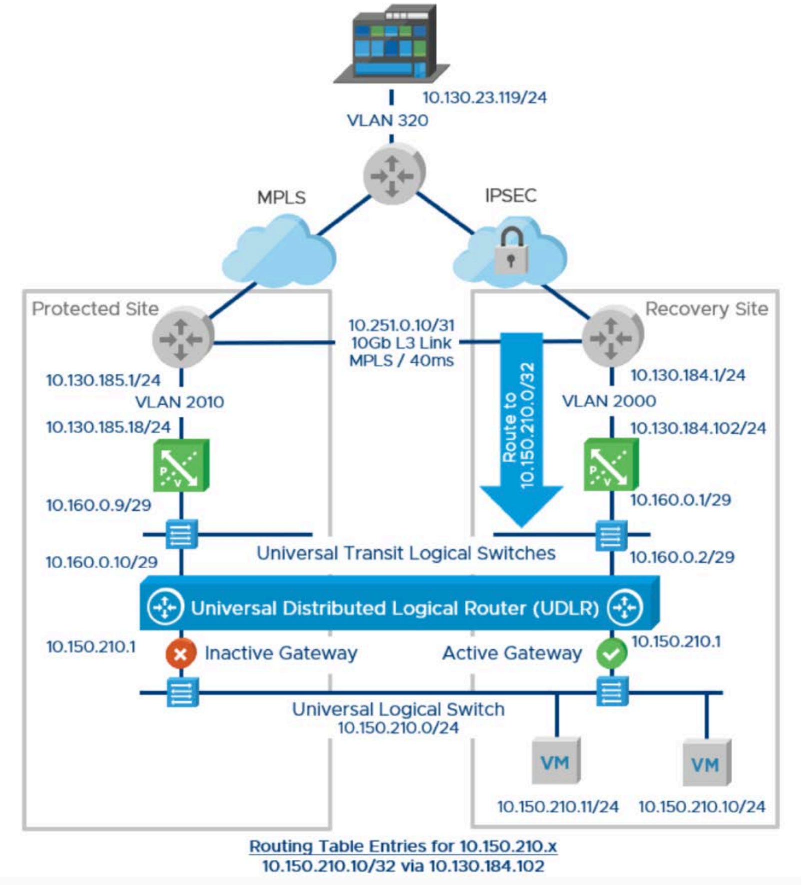


Figure 4.8 Active/Passive under failover conditions

When a failover occurs, all traffic associated with the 10.150.210.0/24 subnet, both inbound and outbound, goes through the ESG in the secondary data center, 10.130.184.102.

A Note About Traffic Tromboning

NSX offers a feature on UDLRs known as local egress. This functionality activates the default gateway for 10.130.184.0/24 at both sites simultaneously, eliminating the additional 40ms latency incurred by crossing the DCI.

Even with local egress enabled, latency is a factor of physical distance; it is driven by the speed of light. Unless users physically move closer to the secondary data center when their VMs are located there, they will always experience that 40ms of latency no matter how the packets route.

This is rarely practical, and because active/active designs are considerably more complex, VMware recommends the active/passive design for most DR solutions with NSX.

Traffic from Untrusted Networks

For purposes of this discussion, the term untrusted networks may refer to remote workers, customers on mobile devices, business partners on desktop PCs inside a corporate office, or any connection to systems outside of the perimeter firewall.

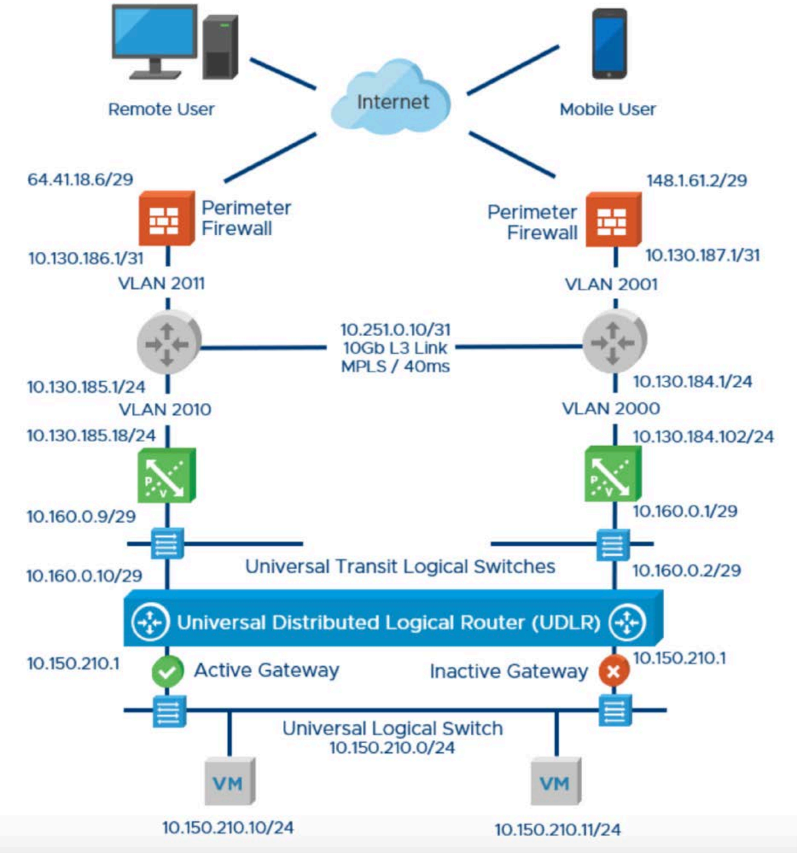


Figure 4.9 Active/Passive under failover conditions

In Figure 4.9, clients on the Internet normally access services on IP 64.41.18.6, a host known in DNS as apps.bluegulflogistics.com. What is the process during a failover event to redirect clients to go through the firewall at the secondary site; how do they know to automatically and quickly connect to IP 148.1.61.2?

DNS TTL manipulation

One solution is setting a very low time-to-live (TTL) value in the DNS address (A) record for apps.bluegulflogistics.com. When a failover occurs, some form of automation (e.g., SRM, custom monitoring script) updates the A record for apps.bluegulflogistics.com to 148.1.61.2. This will work for new queries; however, the old information may still be cached for some users. Clients or DNS servers who have already resolved apps.bluegulflogistics.com to 64.41.18.6 may not honor the low TTL, overriding the desired behavior. These users will need to flush their DNS caches to get the new IP address. This may or may not be a significant problem, depending on the specific use case.

This most significant benefits of this method are its cost – it is free – and simplicity of understanding, implementation, and management.

BGP tricks

The second method involves configuring the same IP address on both of the links, as shown in Figure 4.10.

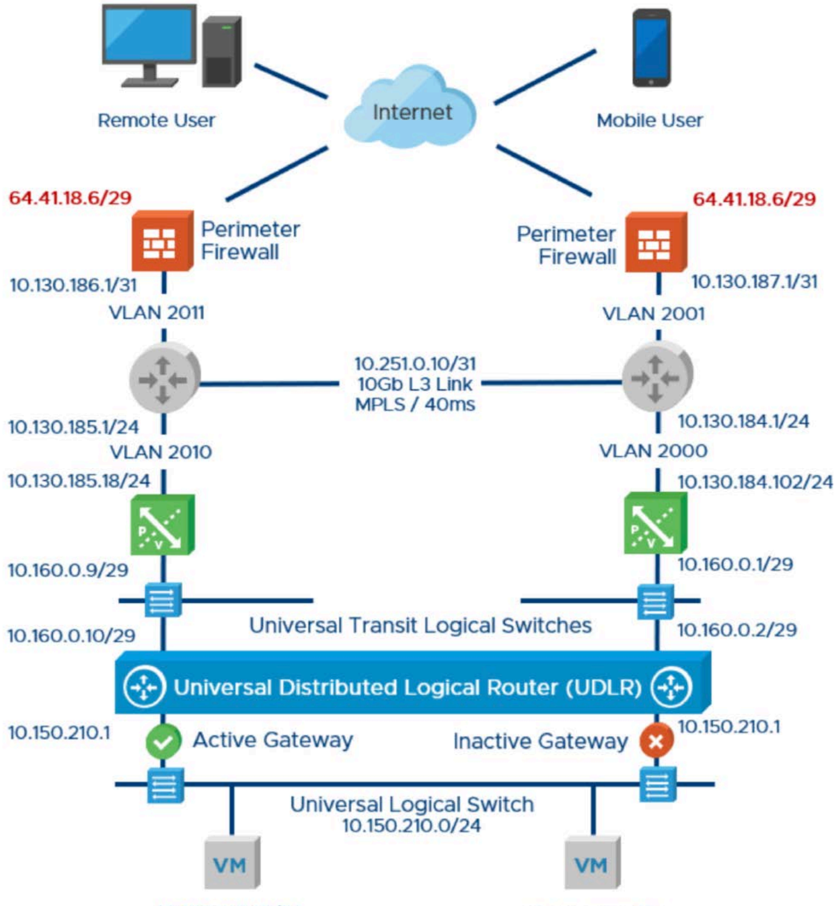


Figure 4.10 Using BGP to have the same IP on both sides

One way to accomplish this is to obtain an autonomous system number (ASN) and provider-independent IP range from ARIN, then pull a full BGP feed from each service provider. Set the weight so that the address path from the primary data center has priority over the path from the secondary data center. Specific details of this implementation are beyond the scope of this book and should be pursued further with local networking experts.

Another option is to have the service provider for the secondary site advertise the host's network via their network, again setting a lower priority weight. Most providers in the US are willing to manage such a configuration for a small monthly fee. This is considerably less work and an easier management solution than the first option.

In either case, when the primary site goes down, BGP will converge and all traffic from the Internet will soon route via the secondary site.

On-Premise DR Automation Solutions with NSX

With an understanding of the building blocks that go into a NSX solution, it is possible to discuss how they come together to form an architectural design for DR. This chapter will cover the high-level design choices that should be considered.

There are several challenges that must be addressed by a DR design:

1. Changing IP addresses is huge challenge within the guest operating system of VMs. It remains common for MAC or IP addresses to be hard-coded within legacy applications, and many software vendors still use IP addresses and other network characteristics to license their products.

2. Keeping an up-to-date DR configuration for a VM's networks, load balancing, firewall rules, and application dependencies can be a manageability challenge.

3. Continual changes to the production site further add to the problem. There is not always time to update the recovery site when configurations are bound to hardware.

When looking at examples of and solutions to these issues, please note that the architectures and diagrams in this chapter assume an active/passive DR scenario with active/standby ESGs.

Traditional Approach

Before discussing how to architect NSX for DR, it is helpful to understand the approach used prior to network virtualization. There have been several hardware-bound approaches to solving the problem of stretching a network between two data centers, including OTV (Overlay Transport Virtualization, a Cisco-proprietary protocol) and virtual private LAN services (VPLS, an industry standard defined in RFCs 4761 and 4762).

Products such as VMware Site Recovery(r) were used with or without OTV. SRM was used to orchestrate the change of IP addresses if OTV was not present. This may have been due to cost, lack of specific hardware, desire to avoid vendor lock-in, or insufficient staff training. In these cases, SRM could customize the IP properties of multiple virtual machines, and an administrator could map VLAN-backed port groups between the two vCenters.

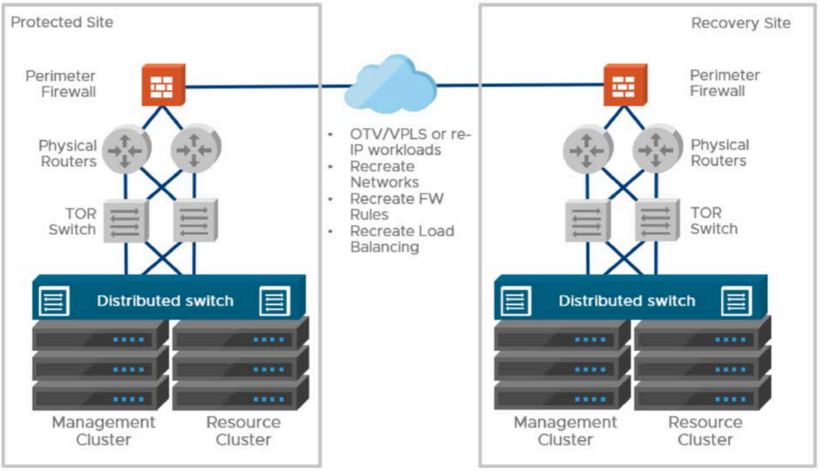


Figure 5.1 Traditional DR approach before network virtualization.

If OTV or another mechanism existed for stretching L2 over L3, the networks would exist in both data centers, but an administrator would still need to explicitly map VLAN-backed port groups between vCenters using SRM.

OTV and SRM perform well at stretching networks and executing recovery plans; however, stretching a network between two sites only solves the problem of changing IP addresses. The re-creation of networks, load-balancing rules, and firewall rules still poses an administrative burden.

DR Scenario Types

Before discussing how NSX has changed DR, the types of failure and recovery conditions should be defined:

- **Partial Application Failover:** Only a part of the application fails over from the protected site to the recovery site. The application components at each site continue to function and communicate as before.

- **Full Application Failover:** The entire application fails over from the protected site to the recovery site.

- **Site Failure** – The entire protected site has failed, including all NSX components. The application and NSX components are restored at the recovery site.

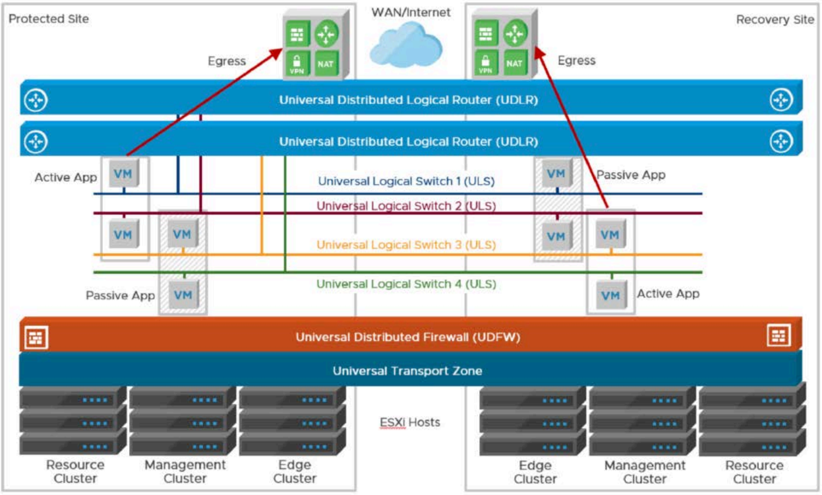
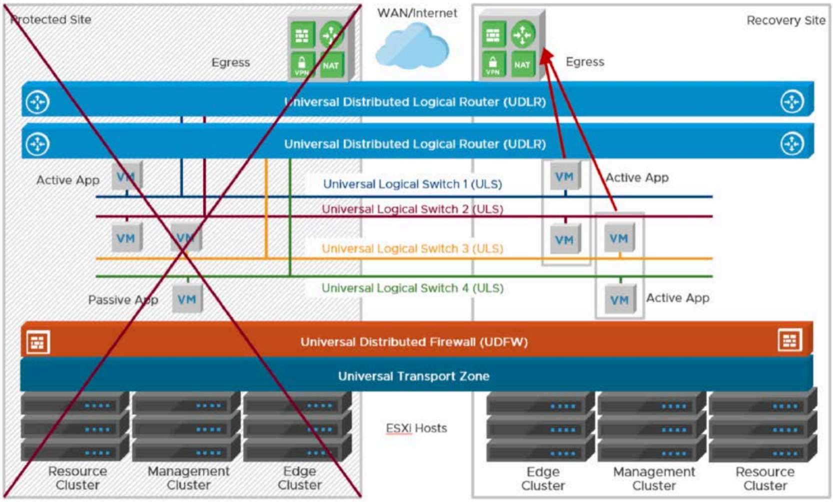

Protected Site
WAN/Internet
Recovery Site
Egress
VPN
NAT
Egress
Universal Distributed Logical Router (UDLR)
Universal Distributed Logical Router (UDLR)
Active App
VM
Universal Logical Switch 1 (ULS)
Passive App
Universal Logical Switch 2 (ULS)
Universal Logical Switch 3 (ULS)
Universal Logical Switch 4 (ULS)
Passive App
Active App
Universal Distributed Firewall (UDFW)
Universal Transport Zone
ESXi Hosts
Resource Cluster
Management Cluster
Edge Cluster
Edge Cluster
Management Cluster
Resource Cluster

Figure 5.2 Bi-Directional Active/Passive Deployment.

Each of these scenarios is active/passive in nature, and failure may be bi-directional (i.e., an application at either site can be failed over to the other).



Protected Site
WAN/Internet
Recovery Site
Egress
Egress
VPN
NAT
Universal Distributed Logical Router (UDLR)
Universal Distributed Logical Router (UDLR)
Active App
Active App
VM
Universal Logical Switch 1 (ULS)
Universal Logical Switch 2 (ULS)
Universal Logical Switch 3 (ULS)
Universal Logical Switch 4 (ULS)
Passive App
Active App
Universal Distributed Firewall (UDFW)
Universal Transport Zone
ESXi Hosts
Resource Cluster
Management Cluster
Edge Cluster
Edge Cluster
Management Cluster
Resource Cluster

Figure 5.3 Full Site Failure

For a full description of NSX's active/active multi-site capabilities, refer to the book *VMware NSX Multi-site Solutions and Cross-vCenter NSX Design Day 1*.

Protecting Appropriate Workloads

Not all workloads are appropriate to protect at a DR site. As an example, Active Directory Domain Controllers are site specific and have their own replication methodology. Where appropriate, always refer to the vendor's support documents and use the appropriate method to protect the application.

For workloads that are site specific, consider keeping those VMs connected to standard logical switches rather than universal logical switches. There is no need to stretch L2 over L3 for a network that is site specific.

vSphere Cluster Design Considerations

At three in the morning when the call comes in about a disaster, the last thing an NSX administrator wants to deal with is circular dependencies. An appropriate cluster design is vital.

The smallest recommended size of a vSphere cluster in any deployment is three hosts. While one host is in maintenance mode, the other two hosts can still provide high availability. An additional best practice for most vSphere deployments is to separate VMs into a management cluster and resource clusters.

A management cluster should host the virtual machines used to support vSphere, NSX, and SRM in addition to other components such as vRealize Automation or VMware Horizon® View™. This is also the appropriate location to place logging and management workloads like vRealize Operations Management. Domain controllers and other infrastructure workloads should be on the management cluster as well. A good rule of thumb is to keep any virtual machine that is site specific and not used to host an end-user application on the management cluster.

One or more resource clusters should be created for workloads that are directly used to support applications for end-users and require replication to the DR site.

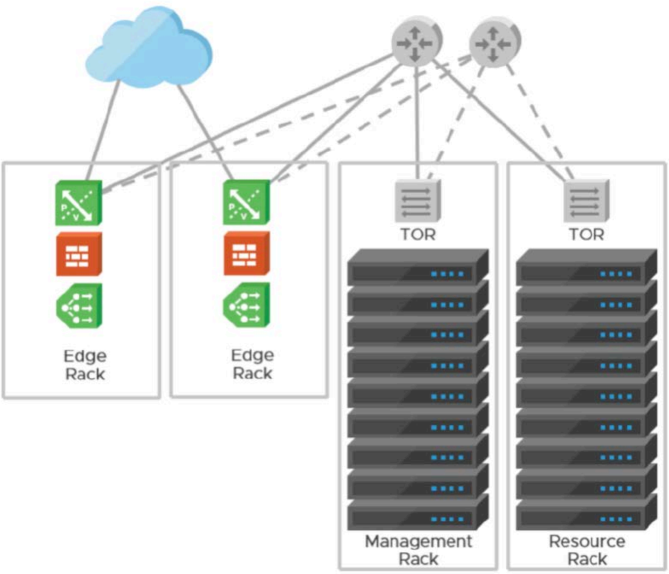


Figure 5.4 Management, Edge and Resource Clusters

Consider the use of an Edge cluster in larger environments. This cluster hosts the Edge Service Gateways, which may require special consideration due to different resource consumption profiles; ESGs are much more CPU-centric with consistent memory requirements. If an L2 bridge is needed to connect a logical switch to a VLAN, the DLR control VM should be in the Edge cluster instead of the resource cluster. It is recommended to use “route based on source ID” as the NIC teaming policy.

Table 5.1 Types of Workloads by Cluster

Management Cluster	Edge Cluster	Resource Cluster
• vCenter • Platform Service Controller (if external) • SRM • NSX Manager • NSX Controller Cluster • Domain Controllers • vCenter • Logging • Management	• Edge Service Gateways • DLR Control VM (if bridging)	• Applications • DLR control VM (if not bridging)

Edge Service Gateway Active/Passive vs. ECMP

Edge Service Gateways provide high availability through configuration as either an active/passive pair or ECMP cluster of up to eight nodes. ECMP mode is recommended for the following situations:

1. ECMP can be used if no stateful services (e.g., firewall, load balancer, NAT) will reside on the ESGs. Note that additional ESGs could be deployed for stateful services near the application tier, south of the nodes being used for connectivity to the physical network.

2. When rapid convergence is required, use ECMP mode; an active/passive deployment may require 1-2 minutes for failover.

3. Where throughput greater than 10Gbps is required, use ECMP.

In summation, use active/passive if a flatter, simpler model is desirable. Use ECMP if extra ESGs, which also means extra hops, are acceptable.

Dynamic Routing Protocols

Avoidance of static routes is strongly recommended; select either BGP or OSPF. BGP is deterministic while OSPF is simpler; pick OSPF unless the flexibility and power of BGP is required.

Traffic Redirection Using Locale-ID vs. Dynamic Routing

Review the NSX documentation discussing the proper steps for creating universal transport zones, universal distributed logical routers, and universal logical switches. As part of this process, a design decision needs to be made – whether to use the NSX Locale ID feature (i.e., a site ID based on the UUID of the NSX Manager) to control north/south routed traffic or to offload control to the dynamic routing protocol.

Two transit universal logical switches are required when using Locale ID. Each ULS will connect to the local ESG and serve as the UDLR uplink logical interface (LIF). A UDLR control appliance must be deployed at each site with routing established to the local ESG. Configuration of route redistribution prefixes is also required to provide failover routing.

There are several advantages to this method; the control appliance does not need to be re-deployed and granular network failovers are possible.

When opting not to use Locale IDs, the burden of controlling site ingress and egress falls to OSPF link cost or BGP weight to ensure the protected site is always the preferred route. In this approach, only one UDLR appliance – located at the protected site – is required, but in the event of a site failure it will need to be re-deployed and routing reestablished. There is no need to orchestrate the setting of a Locale ID, and only one ULS is needed to uplink the UDLRs. In the case of a failure, all networks will swing to the recovery site and traffic will be routed automatically to the recovery site. This can be desirable or alarming, depending on use cases.

When choosing between the two methods, weigh the benefits of manually setting costs/weights and having traffic re-routed automatically. Is changing the Locale ID and setting up routing redistribution a management burden? During an emergency, is reestablishing routing between the UDLR and the ESG acceptable?

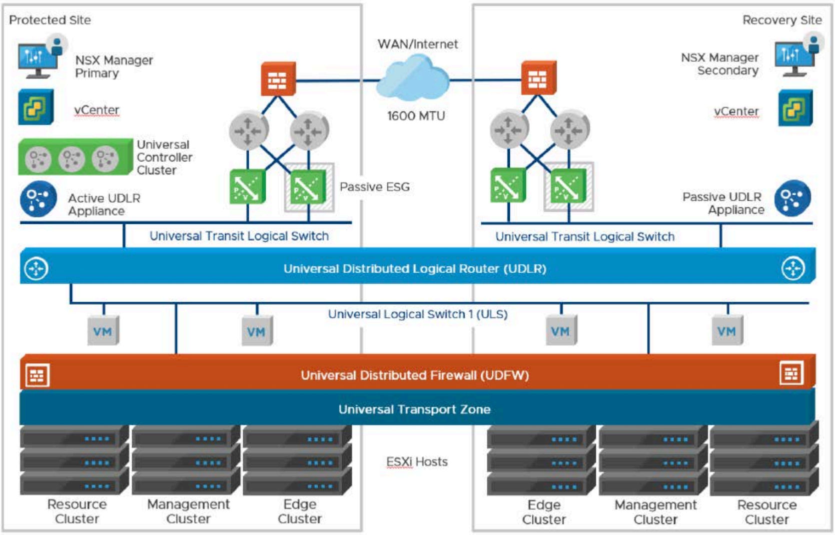


Figure 5.5 NSX Active/Passive High-Level Design using Locale ID for Traffic Redirection

Many real world deployments will manually disconnect the UDLR from the Edge Service Gateways by disconnecting the southbound vNIC on the ESG for that LIF. Some choose to disable dynamic routing between the ESG and the UDLR. This is to prevent any tests or activity from leaving the DR site and impacting production workloads; this is a belt-and-suspenders approach for protecting production.

Post-Recovery NSX Management

Once it is decided that the protected site is indeed down for the duration of an event, components will need to be brought online at the recovery site. The NSX Manager will need to be promoted to primary and a universal Controller cluster deployed.

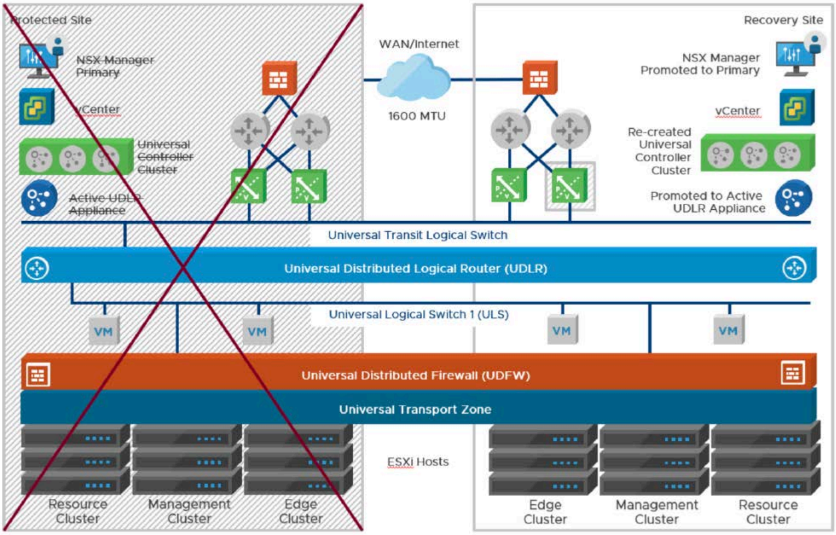

Protected Site
~~NSX Manager Primary~~
vCenter
~~Universal Controller Cluster~~
~~Active UDLR Appliance~~
WAN/Internet
1600 MTU
Recovery Site
NSX Manager Promoted to Primary
vCenter
Re-created Universal Controller Cluster
Promoted to Active UDLR Appliance
Universal Transit Logical Switch
Universal Distributed Logical Router (UDLR)
Universal Logical Switch 1 (ULS)
VM
VM
VM
VM
Universal Distributed Firewall (UDFW)
Universal Transport Zone
ESXi Hosts
Resource Cluster
Management Cluster
Edge Cluster
Edge Cluster
Management Cluster
Resource Cluster

Figure 5.6 Failed Protected Site using Dynamic Routing for Traffic Redirection

There are several ways to automate these steps, such as a vRealize Orchestrator workflows or a PowerShell script called by SRM.

At a high level, a workflow will:

- Promote the secondary NSX Manager to primary
- Redeploy the universal Controller cluster
- Configure the Locale ID to the secondary Manager ID so that traffic can egress the recovery site or re-deploy a Controller appliance if Locale ID is not used
- Reconnect the ESG vNIC or enable dynamic routing if needed

Site Recovery Manager Design Considerations

The major operational benefits of using SRM with NSX are the simplified mapping of networks and the preservation of security policies. Even if constraints exist from another product that stretches L2 over L3, the ability to seamlessly automate network failovers is invaluable. Automating failovers will reduce failover recovery time by decreasing the number of manual configurations. This also makes DR testing faster. Both of these add up to reduce OPEX and free up IT staff to work on projects that support the business.

SRM does not protect NSX components, so ensure NSX is properly deployed and backed up to address specific use cases. Additional information is available in Chapter 7 - Backup Planning.

Storage Policy Protection Groups vs. Regular Virtual Machine Protection Groups

One of the tasks any DR automation product must perform is a mapping of the vCenter inventory. That inventory is then mapped to the recovery site vCenter when a protection group is created. When a protected VM is brought up for DR or testing, these mappings are used to provide the required VM resources, including networking.

Storage policy protection groups (SPPGs) automatically protect VMs in a storage policy. They provide network inventory auto-mapping with no user intervention required.

SPPGs do come with a few caveats. An SPPG is a device-based mapping as opposed to a vCenter inventory mapping; consequently, it requires array-based replication. SPPGs only provide auto-mapping for universal logical switches. SPPGs do not support raw device mappings (RDMs), which should be rarely used.

If array-based replication is not used and a solution like vSphere Replication is deployed, regular virtual machine port groups must be used. This configuration will require user intervention perform the inventory mapping.

DR testing

For testing recovery plans, dedicated test UDLRs can be created. Testing can be simplified by using isolated logical switches which can have overlapping IP addresses. East/west connectivity with applications can be tested without impinging on the production network. It is also possible to allow north/south connectivity on a separate DLR for stateless services.

If stateful services (e.g., access to an Active Directory Domain Controller) are required, carefully plan access for those services. It is much easier to allow access to stateless services like NTP that will not impact production. Accidentally changing AD could be a nightmare. Make sure to refer to the vendor's documentation for testing DR when a change could be made outside of the NSX test networks.

During testing, make sure to create recovery plans and order them by priority. Some applications can be simple – bring up the database, then the app tier, then the web servers. Other applications can be much more difficult.

If micro-segmenting the environment, strongly consider using vRealize Network Insight for mapping. In addition to being a tool for micro-segmentation planning and visibility, it can be useful for identifying application dependency mappings. No tool is perfect for discovering dependencies listed in imperfect documentation, but vRNI can considerably shorten the process.

Security Design

Security is one of the most complex and least understood subjects when it comes to disaster planning and IT organizations. Many still view security to be physical security and access to equipment, not realizing that the real damage and threat is not physical theft or damage but the digital destruction or altering of records.

While the physical security of IT infrastructure is vitally important, it is not the only type of security organizations must keep in mind. This chapter reviews key concepts of digital security and looks at aspects that must be taken into account in a DR scenario.

Role-Based Access Control (RBAC)

As the IT world shrinks and systems become more interconnected, system administrators have access to more systems and thus more critical information. One of the simplest forms of security that organizations can enforce is role-based access control, or RBAC.

RBAC is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file.

In vSphere and the related DR environment, this means controlling access to the means of moving workloads. vSphere contains a rich permission mechanism for granting or restricting access to nearly every portion of the vSphere infrastructure. Granular permissions can be set on individual vSphere objects or entire clusters. Overlapping permissions can exist across a user with multiple accounts enforcing both everyday and admin-level user account access.

VMware NSX follows much the same vein as vSphere, but its controls are far less granular. Permissions within NSX are assigned within NSX Manager itself, not within vSphere. For example, a vSphere administrator may only have auditor rights in NSX; conversely, an NSX enterprise admin may not have rights to vSphere. Table 6.1 details four roles and permissions that can be created for users.

Table 6.1 RBAC Roles

Enterprise Administrator	NSX operations and security.
NSX Administrator	NSX operations only (e.g., install virtual appliances, configure port groups)
Security Administrator	NSX security (e.g., define data security policies, create port groups, create reports for NSX modules)
Auditor	Read only

The role of enterprise administrator should be tightly controlled and only assigned to users that absolutely require it. Operations personal should be assigned with the NSX administrator role if they will interact with NSX on a daily basis. Security admin and auditor should be reserved for those that only rarely interact with NSX or those who should never be able to make changes.

A common mistake organizations make is restricting access to DR infrastructure and applications too tightly. The principal of least access is a sound principal, but DR involves the movement of an entire IT infrastructure stack and will require heightened access. One recommendation is to create different users for everyday operations and users for DR operations. Allow the DR users to have full admin rights, using logging and auditing to ensure those accounts are only used during DR testing or actual DR events.

Logging and Auditing

Every organization should have centralized logging for auditing, historical tracking, and real time research capabilities. Having a central location for logging for the entire IT organization allows for the quick access to data to answer the age-old question of, "who, what, when, and why?"

VMware NSX provides one such tool in VMware vRealize® Log Insight™, which is licensed for free customer use with the purchase of NSX. vRealize Log Insight is, at its core, a syslog server with an advanced data analytics and search engine that allows for easy consumption of logging within an IT organization.

RBAC, logging, and auditing are the three corners of the security triangle and must always be in place in any modern IT organization. Without adequate logging, proper enforcement of RBAC is impossible for any organization performing audits. Without determination of a user's actions within an environment, it is impossible to enforce any sort of security in today's modern IT infrastructure.

Therefore, it is vital that organizations utilize logging to enforce RBAC within the vSphere environment. Imagine the scenario where a malicious actor failed over an entire organizations IT environment during the middle of the day resulting in 24 hours of downtime. Without logging, determining how this occurred and who needs to be held responsible would be impossible.

Security Posture in Disaster Recovery

An important discussion for organizations is the security compliance stance during an actual DR event. The initial assumption is that whatever is in production should be replicated in the DR location, but there may be valid reasons for this not to be the first choice. Three options for discussion include:

1. Mirror compliance between primary and secondary
    a. This would imply that the security posture, RBAC, firewall, logging, and compliance standards are the same regardless of the location of the workloads.
    b. This will necessitate either tools that can replicate security postures between sites or some form of automation to copy a security stance from a primary to secondary site.
2. Primary and secondary have independent security postures
    a. This would imply that the security postures at both sites organization are maintained independently, with changes at one not directly impacting the other.
    b. A scenario where a DR resides in another country or state may require that different security postures be enforced.
3. Security policies are only maintained on the primary site
    a. Organizations may place more importance on uptime and will layer in security after a DR event has occurred.
    b. While the easiest, this is also the riskiest stance an organization can take. The temptation to leave what is working alone can set in, and the addition of security after the fact may never occur.

VMware NSX Universal Object Constructs

Where the above sections examined an organization's overall security stance, the discussion now moves into how VMware NSX allows organizations to enforce security during a DR event.

VMware NSX Localized Objects

With the introduction of NSX, VMware produced a library of software defined networking constructs that allows administrators to perfectly replicate physical network topologies in software. This library implements security constructs including logical routers & switches, distributed firewalls, security tags, and groups. Orchestration of these constructs can be driven through the NSX Service Composer, creating security policies and enforcing them on vSphere objects.

Historically, these constructs and the associated policies were bound to the vCenter where objects resided, so a security tag or firewall rule created in one vCenter could not exist within another vCenter.

Design of a disaster recovery solution necessitated a mechanism to replicate these constructs between different vSphere implementations. VMware professional services designed a mechanism that works via vRealize Orchestrator to copy NSX constructs between vCenters. This allows for a firewall rule, security tag, or group to exist in two different locations.

With this in place, when a VM moves from one location to another for a DR event, its security policy would follow the VM. With the open NSX API, vRealize Orchestrator is not the only option for administrators; any tool that can query the NSX API can be used to replicate networking constructs between two sites.

VMware NSX and Universal Objects

For larger-scale organizations, the manual nature of copying localized constructs is insufficient, leaving too many gaps in a security posture spread over a large distributed environment. With this in mind, VMware introduced the concept of universal objects to VMware NSX.

Unlike localized constructs, universal objects allow for a primary NSX Manager to replicate an object in up to seven secondary NSX Managers. This allows an administrator to create a logical switch with a consistent UUID across all vCenters, in effect stretching a single L2 broadcast domain across potentially an entire group of virtual centers, clusters, or hosts.

Initially limited to IP and MAC sets, VMware now includes a rich library of universal constructs. This gives administrators the tools necessary to create security policy across an entire IT organization, including primary and secondary sites, for DR purposes. Supported constructs include:

- Universal Security Tag
- Universal Security Group
- Universal Logical Switch
- Universal IPSet
- Universal MACSet
- Universal Distributed Logical Router

Limitations do occur within the current implementation of NSX with regard to the Service Composer tool. Service Composer is the policy engine that allows creation and application of security policies to security groups. These groups contain objects tagged with either static or dynamic security tags.

Administrators have two options for creating security policies with NSX when they have more than one vCenter:

1. Manual creation of universal firewall rules using universal objects
    a. When an administrator uses the universal distributed firewall and universal objects, the rules are replicated to each vCenter in that NSX environment. This allows for centralized and consistent rule creation across multiple vCenters, such as primary and secondary data centers in a DR scenario.

    b. Administrators will not be able to use Service Composer; they must manually create the rules in the distributed firewall section.

2. Automatic creation of localized firewall rules using universal objects
    a. When an administrator uses Service Composer, the option exists to use universal constructs. Policy can be applied to universal security tags, groups, IPSets, and MACSets.

    b. This requires Service Composer to setup identical policies in each vCenter and NSX instance.

An often overlooked issue is the assignment of universal security tags to objects not in the primary vCenter. This can occur with an active/active setup or use the DR site for deployment of VMs.

When looking in the UI, there will be no option to tag a VM unless it resides within the primary NSX/vCenter instance. To address this issue, use a REST API call to tag the entity with the universal security tag.

Summary

Security within the DR environment is more complex because of the need to outline the security policy and stance between the primary and secondary sites. Once that policy is in place, ensure that policy and enforcement is replicated at both sites as outlined above. Each option has benefits and drawbacks, so spend the necessary time to properly evaluate the best approach.

Chapter 7

Backup Planning

Why Backups and Disaster Recovery are not the same

It is often asked why disaster recovery is needed if backups are in place, and vice versa. The answer is that the two are very different, each having its own set of operational and business requirements. This chapter outlines the differences between the two, detailing why both are important and required in a modern data center.

Disaster Recovery

Disaster recovery consists of policies, tools, and procedures that enable the recovery and continuation of vital business infrastructure. It focuses on the IT and technology systems critical to supporting business functions. DR is not the same as business continuity, which involves protecting and keeping the essential aspects of a business functioning through major disruptive events. Disaster recovery is a subset of the larger umbrella that business continuity encompasses.

Some examples that make the distinction clearer include:

1. A business has a primary and a secondary data center located in geographically different areas of the country. A disaster occurs in the primary data center and the business is forced to recovery operations at the secondary location. Business continues now that the IT infrastructure has been recovered at the DR site.

2. An office building is cut off from access due to flooding. The staff is unable to come into work and there is no external access to IT systems. IT systems have been recovered at a secondary site, but employees are unable to reach this office, so they are unable to access the IT systems. Business continuity would be the process, procedure, and tools required to enable the workers to access systems, keeping the business functional even through a flood.

As illustrated above, disaster recovery is a subset of business continuity. Both must be in place for a company or institution to properly recovery from a disaster.

Backups

Data protection, or backup, is the process of making copies of data in case the original is lost or damaged. There are three primary backup methodologies used by vendors:

1. **Full:** Full copies are made of the data being protected. Each subsequent backup is an additional full copy of the source data.

2. **Incremental:** Only the files or blocks that have been changed since the last backup are protected. The changes can be compared to either a full or an additional incremental backup.

3. **Differential:** Only the files or blocks that have changed since the last full backup are protected. The changes are only compared to the most recent full backup.

Backups are typically scheduled to run at specific times during a day and have a defined retention period that dictates how many copies or versions of a piece of data are kept.

Differences between DR and Backups

Data Retention Requirements

Backups are most commonly performed on a daily basis for the sole purpose of copying data. Backups can occur more than once per day, but the process is the same – at a specific time and date, a copy of a specified bit of data is made.

Disaster recovery requires that a business outline a recovery time object (RTO). The RTO is defined as the maximum time the business can be without critical IT infrastructure. RTO requirements typically dictate that duplicate or substantial IT infrastructure resides at a secondary location to allow for replication of protected primary site data. Without the secondary infrastructure positioned to receive the replicated data, a primary site cannot be said to have a DR plan in place.

Recovery Ability

Disaster recovery is the process of failing over a primary environment to an alternate environment that is capable of delivering business continuity.

Backups are useful for immediate access in the event of the need to restore a document, but they do not facilitate the failover of the total environment should the infrastructure become compromised. They also do not include the physical resources required to bring them online.

Additional Resources Required

A backup is a copy of data that is intended to be protected. It can be restored either locally or remotely, but it must have physical resources onto which to be restored.

DR requires an additional environment where production data can reside. The current configuration and specifications of the primary site should be replicated at the recovery site. These include physical, software, connectivity, access, and security resources.

Planning process

Recovery point objective (RPO) goals and data retention requirements determine backup planning requirements.

A DR scenario requires: planning, process, and procedure to determine which systems are critical to business operations; creation of the recovery process for those systems; and development a strategy to test system recovery.

The primary benefits of a DR plan are risk mitigation and the avoidance of unplanned downtime. Additionally, DR planning may be required for auditing and compliance purposes.

Outlining the general principle of backups in a DR environment, regardless of what software or platform is selected, data at the primary site should be protected using a combination of onsite and offsite backups.

1. Onsite backups consist of local copies of data kept for a time period determined by the retention period of the backup. A two-week retention period would dictate keeping two weeks of backups for a block of data onsite local to the original source.

2. Offsite backups are also copies of the original data that also follow the retention policy, but are stored remotely from the source data.

Determining the retention period of a backup is a business, compliance, and financial decision specific to every organization. The more frequently backups are made and the longer they are kept, the larger the overall cost and complexity to an organization.

Once a retention period has been determined, decide how often data should be protected. The most common model is a single backup once a day, but business needs may dictate additional coverage.

The type of backup must also be determined, as this directly impacts recovery time. Each of the three defined backup types have benefits and drawbacks on both the time to backup and restore.

Once the backup retention, frequency, and type are determined, administrators should work with their software vendor of choice to create backup schedules that meet the requirements. Ensure that backups are kept both onsite and offsite, but they do not have to be an equal pairing. Many companies will keep daily backups onsite while performing off-site backups only once a week save on costs. Individual business and compliance requirements will dictate specific operational details.

A true DR solution must be able to accommodate IT infrastructure moving from the protected to recovery site for a significant period of time. Backup and restore infrastructure must also be present in the recovery site to provide data protection on workloads once they are recovered at the DR site.

Many software companies have created technologies that offer global catalogs of backups that can be accessed remotely. This may sufficiently address business needs, but ensure that data is kept in redundant locations so that a disaster does not wipe out all data important to the organization.

Backing up SRM and NSX and vSphere Environments

NSX

Backup and restoration of NSX Manager data includes system configuration, events, and audit log tables. Backups are saved to a remote location that must be accessible by the NSX Manager. The backup size will be small because only the configuration data of the NSX Manager is protected, not the actual VM.

When an NSX Manager is restored, a fresh deployment of NSX is instantiated. Users are only required to make the appliance network accessible; no further configuration is necessary. The restoration process will set up the necessary configuration to match settings of the protected NSX Manager.

NSX Edges, distributed logical routers, logical switches, and distributed firewall rules are protected as part of the NSX Manager backup; they do not need to be protected through other means. If the NSX Manager is lost, the state of NSX can be restored back to the point when the backup was taken. Any changes made to the environment post backup are still in effect, but the environment will be out of sync with the state of the NSX Manager. In case of a total environment loss, NSX Manager will allow redeployment of NSX Edges, firewall rules, and distributed logical routers.

Distributed virtual switches and distributed firewall rules can be exported and imported between vCenters for backup, DR, or migration purposes. In the case of a total loss, the distributed virtual switch can be restored to the state of the most recent export. The distributed firewall rules are covered by the NSX Manager backup, but can also be individually exported as a secondary protection mechanism.

vCenter

The VMware vCenter Server® Appliance™ supports a file-based backup and restore mechanism that helps to recover the environment after failures.

In vSphere 6.5, the vCenter Server Appliance management interface allows creation of a file-based backup of the vCenter Server Appliance and Platform Services Controller™ appliance. After creation of the backup, it can be restored by using the GUI installer of the appliance.

The management interface also supports file-based backup of the vCenter Server core configuration, inventory, and historical data. The backed-up data is streamed over FTP, FTPS, HTTP, HTTPS, or SCP to a remote system; the backup is not stored on the vCenter Server Appliance.

File-based restore is only supported for a vCenter Server Appliance that has previously been backed up through the vCenter Server Appliance management interface. This restoration is performed through the GUI installer of the vCenter Server Appliance. The process consists of deploying a new vCenter Server Appliance and copying the data from the file-based backup to the new appliance.

A restore operation can also be performed by deploying a new vCenter Server Appliance, then using the vCenter Server Appliance management interface to copy the data from the file-based backup to the new appliance.

Image-based backups are supported of the vCenter Server Appliance, but only as a full backup copy. Additional work is required to restore the appliance; details can be found in the VMware knowledge base.

SRM

The SRM built-in vPostgres database can be backed up and restored to protect the data within an SRM installation. The procedure requires direct access to the appliance to use Postgres database commands. It is fully documented in the VMware knowledge base.

Snapshotting SRM VMs with a standard image-based backup solution is the most common restoration strategy.

Conclusion

Message from the authors

Putting a functional disaster recovery solution in place is hard. This is why DR projects are perpetually in a state of “yeah we know we need that, and we’re going to do something at some point.” No product will eliminate the non-technical work that must be done to ensure a successful deployment - articulating the “what we want”. However, VMware NSX provides options not available before. It will greatly ease the realization of business requirements for disaster recovery.

It is possible this book has shed some light on things not previously considered, possibly making the task of executing on DR seem more daunting than originally thought. That said, keep in mind, assistance is available. Both VMware Professional Services and members of the extensive partner community can help with the end-to-end design and deployment process. Even if an organization does not conduct the full architecture and deployment of its DR solution in house, the concepts in this book will inform engagements with the selected professional services team.

Other Resources

This book focuses solely on the architecture specifically of the DR functions that NSX performs. Addressing challenges with deployment of a disaster recovery solution is just one of many facets of the value VMware NSX brings to the table.

Due to the fact that NSX can be so transformational to architecture and operations, the authors of this book strongly suggest approaching deployment of each use case individually when building toward a network virtualization environment that realizes the full potential of the investment in NSX. Additional topics such as micro-segmentation, automation of IaaS, regulatory compliance, hybrid cloud, and many others are covered in additional VMware Press books:

- Lees, Kevin. *Operationalizing VMware NSX*. VMware Press, 2017
- Holmes, Wade. *VMware NSX Micro-segmentation: Day 1 Guide*. VMware Press, 2017
- Wilmington, Geoff. *VMware NSX Micro-segmentation: Day 2 Guide*. VMware Press, 2017
- Ali, Shahzad. *Building VMware NSX Powered Clouds and Data Centers for Small and Medium Businesses*. VMware Press, 2017
- Burke, Anthony. *Automating NSX® for vSphere with PowerNSX*. VMware Press, 2017

The authors also strongly encourage participation in a local VMware User's Group chapter (http://vmug.com). VMUG operates independently of VMware and is one of the largest and most active user communities in the world. VMUG presents the opportunity to meet people from other organizations that have successfully deployed DR solutions with NSX, as well as highly knowledgeable community members that can directly answer questions or at least point in the right direction.

Index

VMware NSX for Disaster Recovery - Day 1 brings together the knowledge and guidance for planning, designing, and implementing a disaster recovery architecture for the software-defined data center that meets the needs of your business. VMware NSX simplifies the DR planning and testing that goes into a resilient infrastructure and drastically reduces the time it takes to recover from an event. It enables true workload portability between data centers, private clouds or public clouds. NSX has helped enterprises recover from natural disasters and outages as well as simplifying the mergers and acquisitions of organizations and their networks. *VMware NSX for Disaster Recovery - Day 1* is your roadmap to create a robust network infrastructure within software-defined data centers running NSX. You will find insights and recommendations proven in the field for moving your organization to a resilient, highly available architecture based on VMware NSX.

About the Authors

Brad Christian, VCDX#217 is a Sr. Systems Engineer in the Networking and Security Business Unit at VMware. Brad has worked as a Systems Engineer at organizations ranging from the Fortune 500 to boutique development shops and big SLEDs. He has been using VMware technology in production since the 3.x days. In 2011, Brad took over the Dallas-Fort Worth VMware User Group and grew it to one of the largest world-wide. He has earned the distinction of vExpert 6 times and was one of the first NSX vExperts starting in 2017. He often presents at VMUGs and other events and loves mentoring SEs. Brad has earned the VMware Certified Design Expert designation and is VCDX #217.

Sean Howard, VCDX#130 is a 15 year veteran of the IT field. During this time he has specialized in several areas including Software Development, Storage, Networking and Virtualization. In addition to the VCDX-DCV, he holds a Bachelor of Science in Information Systems from Excelsior College. Now a Systems Engineer Manager for the Network and Security Business Unit at VMware, he runs a team that helps customers solve some of their most challenging infrastructure problems.

William de Marigny, VCIX is a 13-year veteran of the IT industry. During this time, he has specialized in Service Provider scale deployments in several areas including Storage, Network and Virtualization and Managed Backup William is a Senior NSX Technical Account Specialist based in San Antonio Texas and holds many VMware certifications including the VCP5-Cloud, VCP-NV(5,6), VCP-DCV(4,5,6) VCAP5-DCA/DCD,VCAP6-DCA, VCIX6-NV, VCIX6-DCV. He is a three time vExpert and two time NSX vExpert.

vmware PRESS

ISBN-10: 0-9986104-8-8
ISBN-13: 978-0-9986104-8-1

Cover design: VMware

Cover photo: Dreamstime / Dtfoxfoto

www.vmware.com/go/run-nsx

ISBN 9780998610481
51299 >
9 780998 610481

$12.99