HPE Reference Architecture on VMware Cloud Foundation on HPE Synergy

Contents

Executive summary

In today's world, businesses need to turn ideas into services faster, respond quickly to new customer demands, and innovate by building new services with technology to stay competitive. To meet these business demands IT is increasingly adopting new cloud technologies, moving away from expensive purpose built hardware to a software-defined model, and running VMs in the public cloud, while keeping the majority of their workloads on premises. Enterprises need an ideal Hybrid IT model that supports both traditional and new breed of cloud native applications. Therefore, business are embracing a journey to digital transformation and software-defined data center (SDDC) to balance the needs of the business and the needs of IT to support this change.

To address this challenge, Hewlett Packard Enterprise and VMware have collaborated to help customers accelerate the journey to hybrid cloud, bringing the promise of the software-defined data center to life. The combination of HPE's composable infrastructure with VMware's SDDC solution dramatically improves both the value and business outcomes our customers experience. HPE Synergy combined with VMware Cloud Foundation, delivers a simplified and secure private cloud-based on the VMware Software-Defined Data Center stack on composable infrastructure that is flexible, easy to deploy, seamless to manage, and simple to operate. For enterprise customers looking to accelerate their journey to hybrid cloud, HPE Synergy combined with VMware Cloud Foundation is the right solution to deliver a simplified and secure private cloud to run all your enterprise apps—both traditional and containerized—in cloud environments.

This Reference Architecture provides architectural guidance for deploying, and managing VMware Cloud Foundation (VCF) on HPE Synergy for traditional application and a modern containerized based application, both automated by vRealize Automation and secured by NSX within a single VCF Workload domain. The traditional application showcased in this Reference Architecture is WordPress and the containerized application is Yelb. There are three use cases demonstrated in this paper:

- Use Case 1: Demonstrate a traditional multi-tier application and a containerized application deployment
- Use Case 2: Auto Provisioning of Network uplink and downlink ports using Ansible scripts
- Use case 3: Demonstrate ease of monitoring and reporting of VCF infrastructure using HPE OneView for vRealize Operations

This Reference Architecture demonstrates the following benefits:

- The value of combining HPE Synergy with VMware VCF from a deployment and lifecycle perspective in a cost-effective and simplified management for faster time to value.
- An easy-to-operate VI traditional data center and containerized applications in a single VCF workload domain.
- A tested and tuned solution architecture that offers optimal performance and security to deploy VMware VCF 3.0 running multi-tier apps and containerized apps. Both applications are delivered through vRealize Automation blueprints which include security tied in with NSX including micro-segmentation to provide dynamic isolation between different tiers of the application and edge-services such as load balancing of the web tier.
- Automation of Arista end of row switch operations, by adding uplinks trunk group ports for bringing up VMware VCF deployment with Ansible Playbooks.
- Ability to efficiently scale storage and compute independently, to deploy SDDC environments. Grow-as-you-go and support both traditional virtual machines and cloud-native applications.
- To expand and contract physical and virtual infrastructure on-demand, to quickly meet changing business requirements with HPE OneView.
- An efficient monitoring for VCF environment using HPE OneView for VMware vRealize Operation's custom dashboards.

Target audience: This document is intended for IT decision makers as well as architects and implementation personnel who want to understand enterprise ready private cloud solutions using the HPE Composable Infrastructure capabilities offered by the HPE Synergy platform and VMware Cloud Foundation. The reader should have a solid understanding and familiarity with VMware Cloud Foundation, VMware vRealize suite and HPE Synergy.

Document purpose: The purpose of this document is to demonstrate enterprise ready private cloud solutions by combining the value of VMware Cloud Foundation (VCF) on HPE Composable Infrastructure that is flexible, easy to deploy, and cloud management capability for self-service, auto scales and ease operation efficient. The document provides blue prints deployment design to run all your enterprise apps—both traditional and containerized—in virtualization cloud environments.

This Reference Architecture describes solution testing performed in January 2019.

Solution overview

This Reference Architecture demonstrates design best practices for customers building a private cloud solution in an enterprise data center and deploying business critical applications in a fully secure and automated manner. The solution design is based on VMware Cloud Foundation on HPE Synergy. VMware Cloud Foundation (VCF) provides a unified software-defined data center (SDDC) platform integrating VMware vSphere platform, VMware Virtual SAN Storage and VMware NSX networking and fully automating and providing lifecycle management with SDDC Manager. In addition, VCF also delivers VMware vRealize suite which includes vRealize Automation, vRealize Orchestration and vRealize Operations to provide cloud management and automation capabilities.

The VMware Cloud Foundation (VCF) software is deployed over HPE Synergy composable, software-defined infrastructure platform which provides a fluid pools of physical and virtual compute, storage and fabric resources, ready for any configuration for any application.

The solution provides architectural details for deploying a traditional multi-tier application and a modern VMware vSphere Integrated Container (VIC) based application, both automated by vRealize Automation and secured by NSX within a single VCF Workload domain. The traditional multi-tier application showcased in this Reference Architecture is WordPress and the containerized application shown is Yelb. Both these applications are delivered through vRealize Automation blueprints which include security tied in with NSX including micro-segmentation to provide dynamic isolation between different tiers of the application and edge-services such as load balancing of the web tier.

The solution involves two (2) Synergy 12000 frames each equipped with four (4) HPE Synergy 480 Gen10 Servers, and an HPE Synergy D3940 Module. Each HPE Synergy 12000 frame uses HPE VC SE 40Gb F8 Module as a fabric module to provide uplink to the data center network The eight (8) HPE Synergy 480 Gen10 servers are used to configure one management workload domain and one compute workload domain for VI. The compute workload domain is used to host the WordPress application which is Virtual Machine based and the Yelb application which is vSphere Integrated Container (VIC) based. Using HPE D3940 direct attached storage module, VMware vSAN provides software based distributed storage for the entire solution.

The following image, Figure 1, shows the high-level design diagram showcasing the solution components involved.

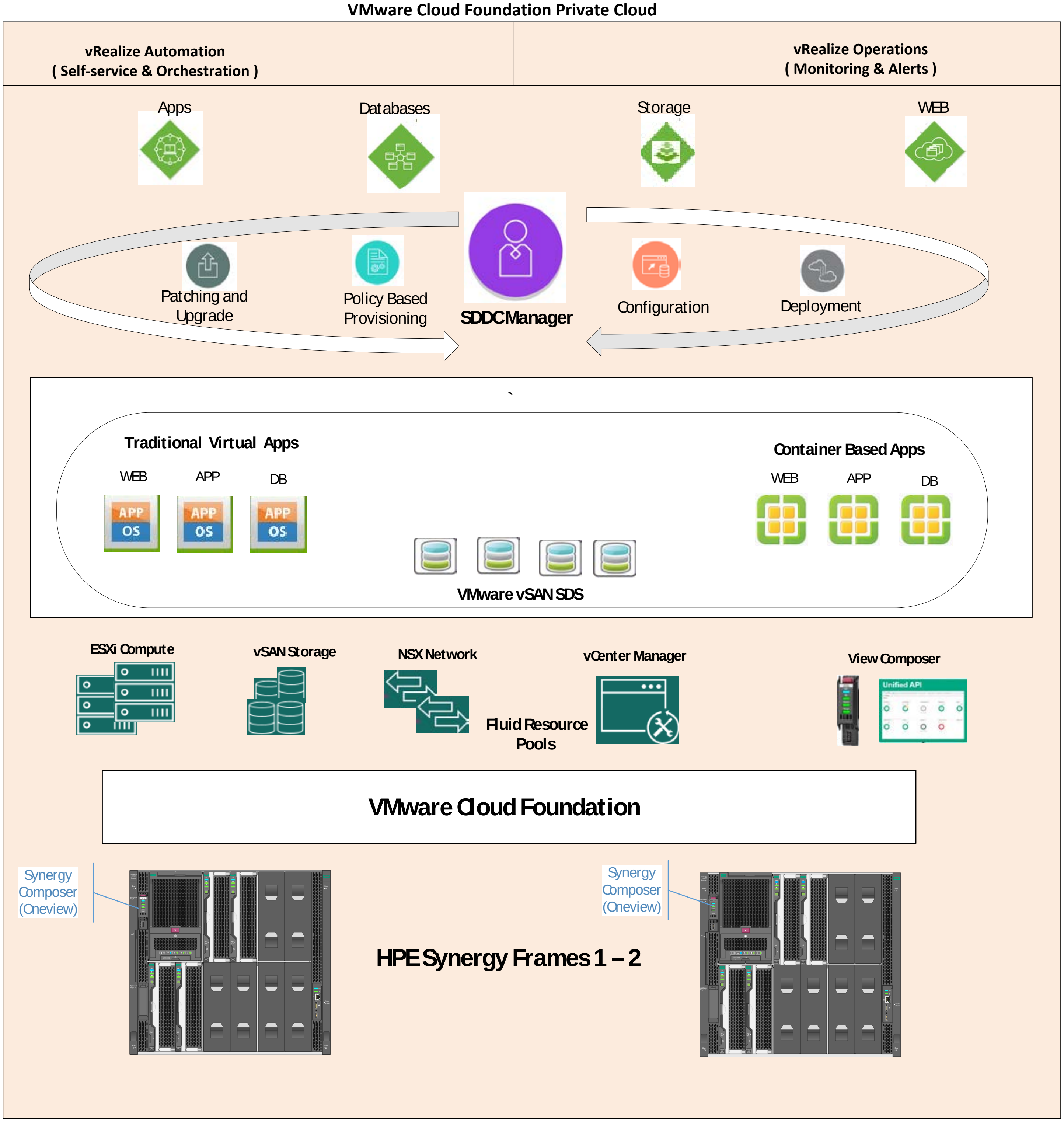


Figure 1. High-level design diagram

Solution components

Solution Hardware components

HPE Synergy

HPE Synergy is a composable infrastructure platform that empowers IT to create and deploy resources instantly and continuously, gain control of IT resources efficiently, and simplifies IT operation using a single software-defined infrastructure for physical, virtual and containerized workload. Developers and ISVs can programmatically control a composable infrastructure through a single, open API that is native in HPE Synergy powered by HPE OneView. This Reference Architecture is built upon the following composability concepts and capabilities of the HPE Synergy platform.

Fluid resource pools

HPE Synergy allows the transformation of traditionally rigid physical systems into flexible virtual resource pools. HPE Synergy creates resource pools of "stateless" compute, storage, and fabric capacity that can be configured almost instantly to rapidly provision infrastructure for a broad range of applications.

Software-defined intelligence

The software-defined intelligence in HPE Synergy reduces operational complexity and enables IT organizations to make needed programmatic changes quickly and confidently, with minimal human intervention. HPE Synergy abstracts operational details and replaces them with high-level, automated operations. HPE Synergy uses templates to automatically implement change operations such as updating firmware, adding additional storage to a service, or modifying a network.

Unified API

HPE Synergy delivers automation through a unified API that provides a single interface to discover, inventory, configure, provision, update, and diagnose the Composable Infrastructure in a heterogeneous environment. This fully programmable interface integrates with dozens of popular management tools such as Microsoft System Centre, VMware vCenter and open source automation and DevOps tools such as Chef, Docker, and OpenStack.

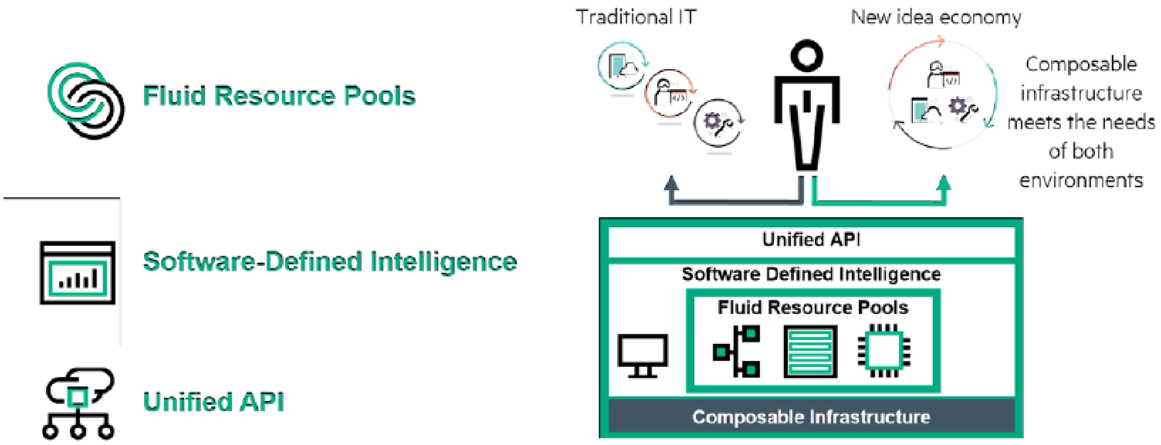

Fluid Resource Pools
Software-Defined Intelligence
Unified API
Traditional IT
New idea economy
Composable infrastructure meets the needs of both environments
Unified API
Software Defined Intelligence
Fluid Resource Pools
Composable Infrastructure

Figure 2: The three architectural principles of HPE Synergy Composable Infrastructure

HPE Synergy Composer

HPE Synergy Composer provides enterprise-level management to compose and deploy system resources, for all of your application needs. This management appliance uses software-defined intelligence with embedded HPE OneView to aggregate compute, storage and fabric resources in a manner that scales to your application needs, instead of being restricted to the fixed ratios of traditional resource offerings. HPE OneView server profiles and profile templates capture the entire server configuration in one place, enabling administrators to replicate new server profiles and to modify them as needed to reflect changes in the data center. With HPE OneView Rest API and automation tools, the entire process of server personality definition and configuration can be automated. For this Reference Architecture the HPE OneView REST API and PowerShell library were used to automate the server profile application to "stateless" servers.

HPE Synergy 12000 Frame

The HPE Synergy 12000 Frame is a base infrastructure that ties together compute, storage, network fabric, and power into a scalable solution that easily addresses and scales with various customer workloads and infrastructures. The Synergy 12000 reduces complexity in the IT infrastructure by unifying all these resources into a common bus, and with the myriad of available network and storage interconnects, allows the frame to interoperate with any other IT environment. At a high level the Synergy frame supports the following:

- 12 half-height or 6 full-height compute modules. The Synergy design additionally allows for the inclusion of double-wide modules as well, such as the D3940 Storage Module.
- Ten fans and a single Frame Link Module for in-band and out-of-band management.
- Up to six 2650 watt power supplies.
- Up to six interconnect modules for full redundancy of three fabrics.

The Synergy 12000 features a fully automated and managed composer module using HPE OneView, contained within the HPE Synergy Composer module. OneView handles all the setup, provisioning, and management both at the physical and logical level.

Figure 3. HPE Synergy Architecture

HPE Synergy 480 Gen 10 Compute Module

The HPE Synergy 480 Compute Module delivers superior capacity, efficiency, and flexibility in a two-socket, half-height, single-wide form factor to support demanding workloads. Powered by the latest Intel® Xeon® E5-2600 v4 processors and featuring support for up to 1.5TB of HPE DDR4 SmartMemory, flexible storage controller options, three I/O connectors, and designed to create a pool of flexible compute capacity within a composable infrastructure, the HPE Synergy 480 Gen 10 Compute Module is an ideal platform for general-purpose enterprise workload performance now and in the future.

The solution as presented in this Reference Architecture white paper contains two Synergy 12000 frames in two different racks. Each Synergy 12000 frames in a rack consist of four (4) HPE Synergy 480 Gen10 Servers.

Figure 4. HPE Synergy 480 Gen 10 Compute Module

HPE Synergy D3940 Storage Module

The HPE Synergy D3940 Storage Module is a directed attached storage module with 40 Small Form Factor (SFF) drive bays designed for use in HPE 12000 Synergy frames. Through the HPE Synergy 12Gb SAS Connection module it provides composable storage for up to 10 compute modules in a single frame. Synergy storage is optimized for use as either a direct attached storage array or as software-defined storage. HPE Synergy D3940 Storage Modules support a family of 12G SAS or 6G SATA HDD and SSD Smart Drives.

Figure 5. HPE Synergy Storage module

HPE Virtual Connect SE 40Gb F8 Module for Synergy

The HPE Virtual Connect SE 40Gb F8 Module, master module based on composable fabric, is designed for composable infrastructure. The disaggregated, rack-scale design uses a Master/Satellite architecture to consolidate data center network connections, reduce hardware, and scale network bandwidth across multiple HPE Synergy 12000 Frames. The master module contains intelligent networking capabilities that extends connectivity to satellite frames through Interconnect Link Modules. This decreases top of rack switch needs and substantially reduces costs. The components reduction simplifies fabric management at scale while consuming fewer ports at the data center aggregation layer.

The HPE Virtual Connect SE 40Gb F8 Module for Synergy eliminates network sprawl at the edge with one device that converges traffic inside the HPE Synergy 12000 Frames, and directly connects to external LANs.

HPE Synergy 20Gb Interconnect Link Module

The HPE Synergy 20Gb Interconnect Link Module (satellite module) is designed for composable infrastructure. Based on a disaggregated, rack-scale design, it uses a Master/Satellite architecture to consolidate data center network connections, reduce hardware and scale network bandwidth across multiple HPE Synergy 12000 Frames.

Arista 7160 Switch

Arista 7160 Data Center Switch Series are purpose-built, fixed-configuration 10/25GbE and 100GbE systems with wire-speed L2/3 features, built for the highest-performance environments and the largest scale data centers. It can be deployed in a wide range of open networking solutions including largescale L2/3 cloud designs, overlay networks, virtualized or traditional enterprise data center networks.

The Arista 7160 48SFP25 6QSFP28 switch comes with 48 SFP25 and 6 QSFP28 ports. It is supported by Arista Extensible Operating System (EOS). Figure 6 shows front view of Arista 7160 48SFP25 6QSFP28 switch.

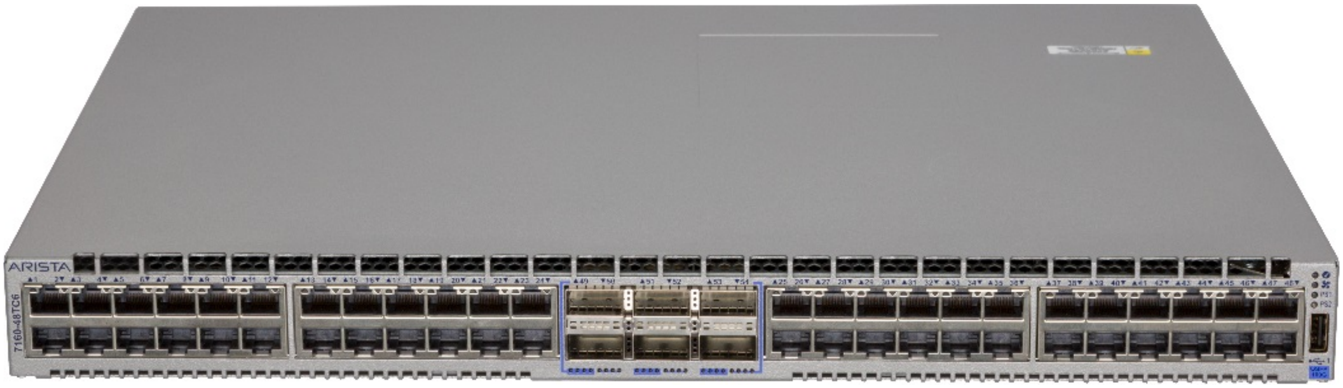


Figure 6. Arista 7160 48SFP25 6QSFP28 switch

Solution Software components

The table below lists the software components used in this Reference Architecture. Further below mentions the layers of the full solution stack.

Table 1 defines the HPE Synergy 480 Gen10 software components and versions used in this Reference Architecture.

Table 1. Software components and versions used in this Reference Architecture

Component	Version
HPE	
HPE OneView	4.0
VMware	
VMware Cloud Foundation	3.0
VMware vCenter	6.5
VMware ESXi	6.5 U2, EP8
VMware vSAN	6.6.1 EP8
Cloud Foundation Builder VM	3.0
SDDC Manager	3.0
VMware vRealize Operation	6.7
VMware vRealize Log Insight	4.6.1
VMware vRealize Automation	7.4
vRealize Suite Life Cycle Manager	1.2
VMware NSX Data Center for vSphere	6.4
VMware vSphere Integrated Containers	1.3
Microsoft	
Microsoft Windows Server 2016	Standard

VMware Cloud Foundation 3.0

VMware Cloud Foundation™ is the industry's most advanced enterprise-ready cloud platform providing a complete set of software-defined services for compute, storage, networking, security and cloud management to run enterprise apps whether it is traditional or containerized. Cloud Foundation drastically simplifies data center operations by deploying a standardized and validated architecture with built in lifecycle automation of the cloud stack. It orchestrates, provisions, and deploys a software-defined data center (SDDC) platform by stitching together VMware vSphere, vSAN, and NSX into a natively integrated stack to deliver enterprise-ready cloud infrastructure.

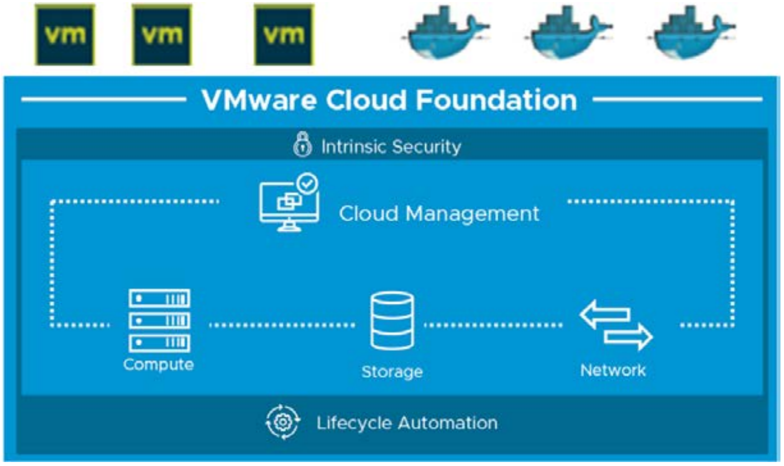


Figure 7. VMware Cloud Foundation

VMware Cloud Foundation components

The core components for VMware Cloud Foundation are explained below.

Cloud Foundation Builder VM

The Cloud Foundation Builder VM is a one-time use VM which deploys and configures the management domain and transfers inventory and control to SDDC Manager. During the deployment process, the Cloud Foundation Builder VM validates network information you provided in the deployment parameter spreadsheet such as DNS, network (VLANS, IPs, MTUs), and credentials. After the management domain is up and the SDDC Manager is running, the Cloud Foundation Builder VM must be powered off and archived. Table 4 shows Cloud Foundation Builder VM resource requirements.

Table 2. Cloud Foundation Builder VM resource requirement

Components	Requirements
CPU	4 vCPUs
Memory	4GB
Storage	350 GB

SDDC Manager

SDDC Manager manages the bring-up of the Cloud Foundation system, creates and manages workload domains, and performs lifecycle management to ensure the software components remain up-to-to date. SDDC Manager also monitors the logical and physical resources of Cloud Foundation. It allows data center administrators to configure the additional hosts and racks into a logical pool of resources and thus multiple racks can be managed as a single Cloud Foundation System. SDDC Manager controls these processes by using workflows. Each workflow comprises of a series of tasks, which are executed by SDDC Manager. There are two VMs installed by VCF 3.0 for SDDC Manager each VM is performing its own functions.

SDDC Manager Controller VM

The SDDC Manager Controller VM includes the logic for deploying the software stack, managing workload domains, managing hardware tasks, and performing lifecycle management.

SDDC Manager Utility VM

The SDDC Manager Utility VM contains the downloaded LCM update bundles, as well host-level and NSX backup files.

VMware vCenter Server

vCenter Server provides for management of a VMware virtualized environment with one or more ESXi hosts. SDDC Manager deploys one vCenter Server per workload domain. By default, all vCenter Servers are configured in enhanced linked mode.

VMware platform Services Controller

During bring-up, SDDC Manager deploys two Platform Services Controllers in the management domain. These instantiate an SSO domain. All vCenter Servers (management domain and compute workload domains) are registered with the SSO domain and configured in enhanced link mode.

VMware vSphere (ESXi)

ESXi is a type 1 hypervisor used to implement virtualization on bare metal systems. ESXi provides for compute virtualization within the software-defined data center and is a foundational building block for implementing a private cloud.

VMware vSAN

VMware vSAN™ aggregates local or direct-attached data storage devices to create a single storage pool shared across all hosts in the vSAN cluster. vSAN eliminates the need for external shared storage, simplifies storage configuration and virtual machine provisioning.

VMware NSX Data Center for vSphere

NSX is the network virtualization platform for the SDDC, delivering the operational model of a virtual machine for entire networks. With NSX, network functions including switching, routing, and firewalling are embedded in the hypervisor and distributed across the environment.

VMware NSX for vSphere is a virtualized networking component in the software-defined data center (SDDC) architecture, which programmatically creates, snapshots, deletes, and restores software-based virtual networks. With network virtualization, the functional equivalent of a network hypervisor, NSX reproduces the complete set of Layer 2 to Layer 7 networking services (e.g., switching, routing, firewalling, and load balancing) in software. It allows these services to be programmatically assembled in any arbitrary combination to produce unique, isolated virtual networks in a matter of seconds. NSX also provides a platform for various security services both network and endpoint based. NSX provides various built-in services, including L2-L4 firewall and activity monitoring. Additionally, security vendors can leverage its guest introspection and network introspection frameworks to deliver service chained next-generation firewall, IDS/IPS, agentless AV, file integrity monitoring, and vulnerability management capabilities.

VMware vRealize Log insight

vRealize Log Insight delivers heterogeneous and highly scalable log management with intuitive, actionable dashboards, sophisticated analytics and broad third-party extensibility, providing deep operational visibility and faster troubleshooting.

VMware Cloud Foundation also has the following optional components for which separate licenses are needed.

VMware vRealize Operations Manager

vRealize Operations Manager delivers intelligent operations management with application-to-storage visibility across physical, virtual, and cloud infrastructures. Using policy-based automation, operations teams automate key processes and improve IT efficiency. This is an optional component.

VMware vRealize Automation

vRealize Automation is a cloud automation tool that accelerates the delivery of IT services through automation and pre-defined policies, providing high level of agility and flexibility for developers, while enabling IT teams to maintain frictionless governance and control. This is an optional component.

vRealize Suite Orchestrator

vRealize Orchestrator is a development- and process-automation platform that provides an extensive library of workflows and a workflow engine. It simplifies the automation of complex IT tasks.

VMware vSphere Integrated Containers

vSphere Integrated Containers provides critical enterprise container infrastructure to help IT Ops run both traditional and containerized applications side-by-side on a common platform. Once vSphere Integrated Containers is installed, both developers and administrators can provision and manage containers through the VMware Cloud Foundation management tools and a command prompt.

HPE OneView for VMware vRealize Operations

HPE OneView for VMware vRealize Operations provides integrated and highly automated performance, capacity, configuration compliance, and cost management tools to the vRealize Operations custom GUI. The Plugin seamlessly integrates the manageability features of HPE Synergy with VMware's analytics engine that analyzes what is normal and then applies that baseline to a dynamic server environment.

When the HPE OneView for VMware vRealize Operations is installed, the custom HPE OneView Dashboards are added to the vRealize Operations custom GUI. The HPE OneView Dashboards allow you to monitor resources in a vRealize environment. The attributes that can be monitored include: resource health, power, temperature (server and enclosure), and system alerts. The analytics engine allows for proactive monitoring of the HPE OneView resource environment and indicates the state of the resources. If a problem occurs, an alert is triggered and displayed. The analytics engine also provides for proactive prediction which can determine the point in the future when a resource will reach a predefined critical level.

Design and configuration guidance

HPE Synergy Solution design and configuration

The hardware used to configure this Reference Architecture contains two rack, with each rack contains one HPE Synergy 12000 Frame. Each HPE Synergy 12000 Frame contains 4 HPE Synergy 480 Gen 10 Compute Modules and an HPE Synergy D3940 Storage Module. Each HPE Synergy D3940 storage module consists of 40 drive enclosures comprising a mix of 1.92 TB SATA SSD drives for capacity tier and 800 GB SAS SSD for cache tier, as per VMware vSAN requirement. The Synergy 12000 frames have a redundant pair of HPE Synergy 12Gb SAS Connection Modules, to provide powerful and redundant connectivity to the D3940 Storage Module, and a redundant pair of HPE Synergy Virtual Connect SE 40Gb F8 Modules, for high-speed uplink connectivity to multiple networks.

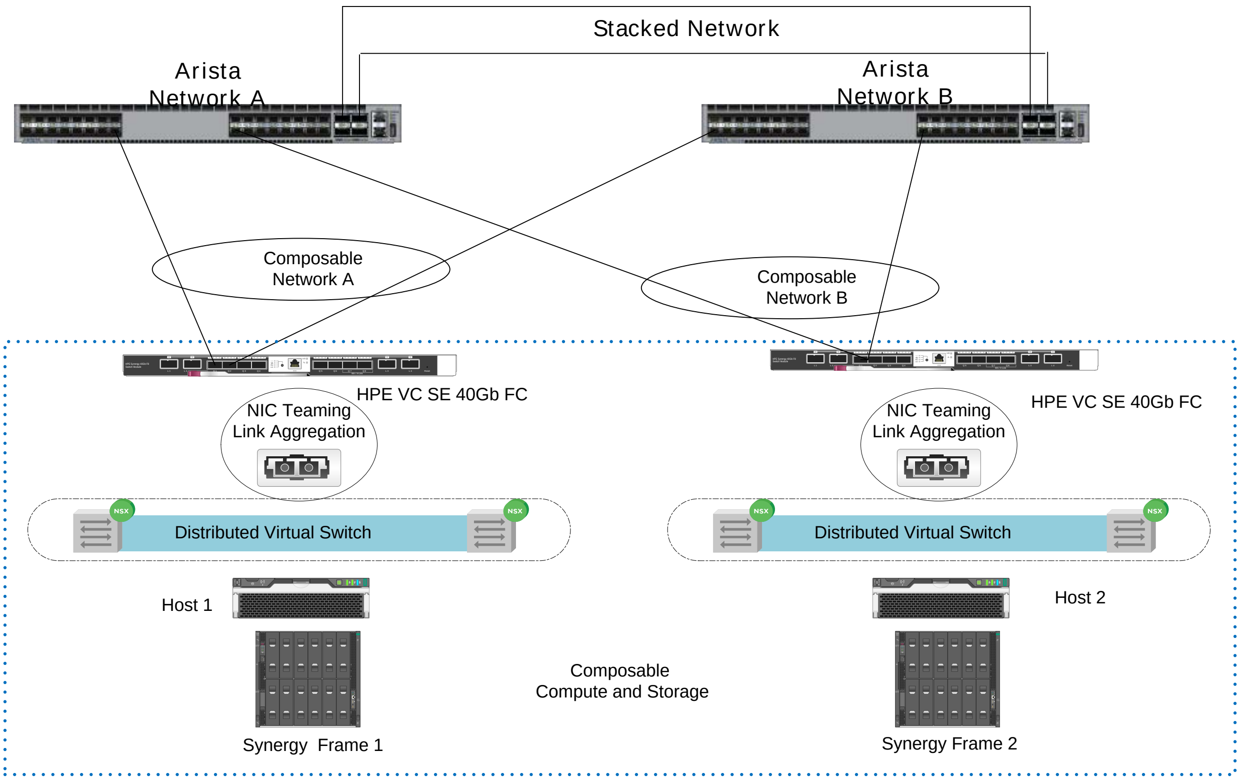


Figure 8. Solution layout

The figure 9 shows the front view of the hardware components used in the solutions as mounted in the rack.

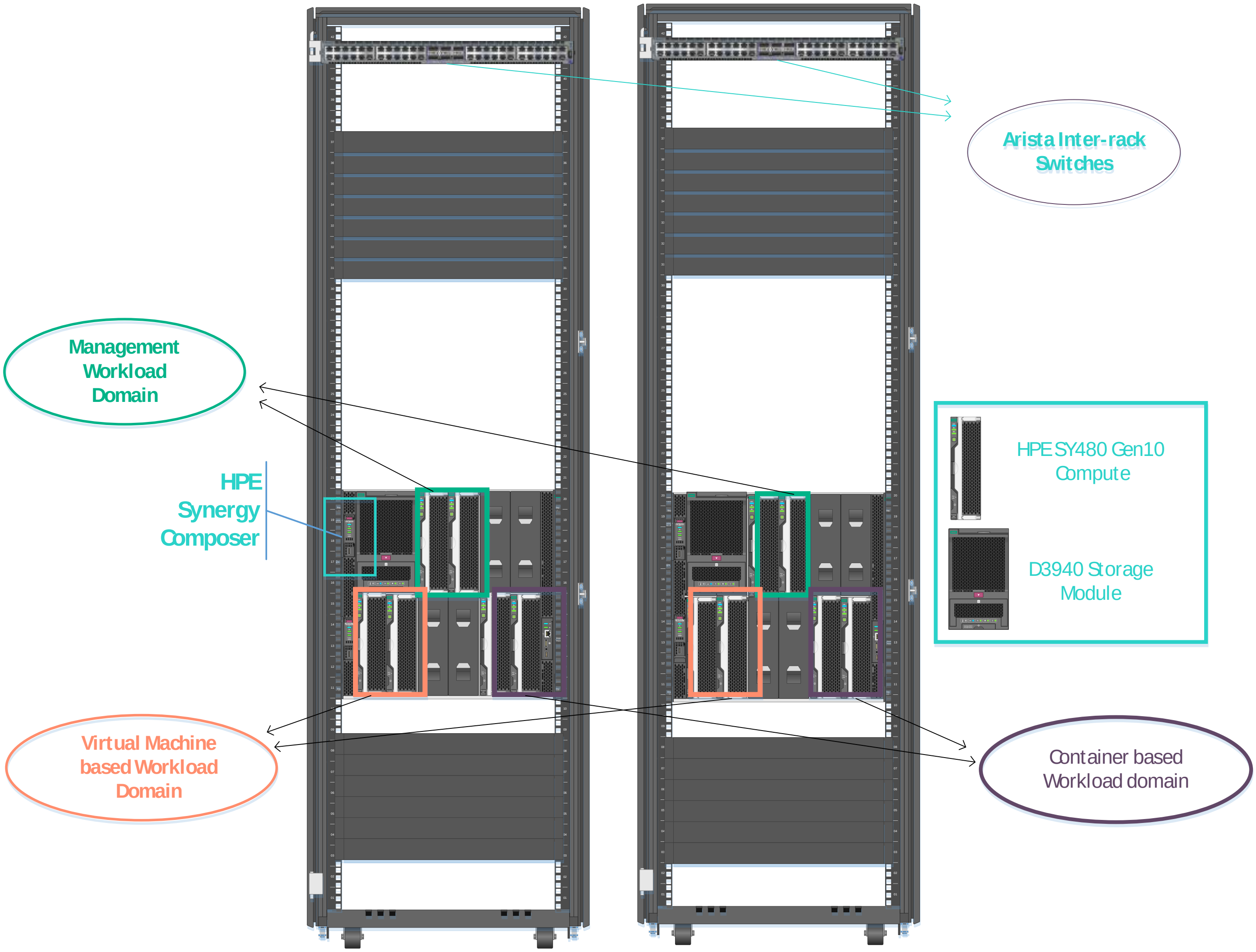

Arista Inter-rack Switches
Management Workload Domain
HPE Synergy Composer
HPE SY480 Gen10 Compute
D3940 Storage Module
Virtual Machine based Workload Domain
Container based Workload domain

Figure 9: Front view of the hardware in the rack

Table 3 defines the hardware configuration used in this Reference Architecture.

Table 3. Hardware configuration

Hardware	Details
Number of racks	2
Number of Synergy12000 Frames in each rack	1
HPE Synergy 480 Gen 10 Compute Module per Synergy 12000 Frame	4
HPE Synergy Storage D3940 Storage Module per HPE Synergy 12000 Frames	1
HPE Virtual Connect SE 40Gb F8 Module for Synergy per Synergy 12000 Frames	2 (Redundant)
HPE Synergy 20GB Interconnect Link Module per Synergy 12000 Frames	2 (Redundant)

Hardware	Details
HPE Synergy Storage Module per Synergy 12000 Frames	1*D3940
HPE Synergy Composer Module	2 (Redundant)
HPE Synergy network options	Synergy 3820C 10/20Gb CNA

Table 4 defines the HPE Synergy 480 Gen10 hardware components used in this Reference Architecture.

Table 4. HPE Synergy 480 Gen10 hardware components (quantities are per node)

Hardware	Quantity	Description
CPU	2	Intel(R) Xeon(R) Gold 6142 CPU (2.6 GHz / 16-core)
Memory	24	8*HPE 32GB 2Rx4 PC4-2666V-R Smart Kit 16*HPE 16GB 2Rx8 PC4-2666V-R Smart Kit Total 512 GB memory on each node
10/20Gb CNA	1	HPE Synergy 3820C 10/20Gb Converged Network Adapter

The VMware Cloud Foundation infrastructure needs different External Services for initial deployment and deployment of other optional components like vRealize Operations or vRealize Automation. Those services as Active Directory, Dynamic Host Configuration Protocol (DHCP), Domain name Service (DNS), Network Time Protocol (NTP) are part of customer's data center environment.

The Cloud Foundation Builder virtual machine was installed on separate ESXi host and is not part of the Synergy Environment. It is configured to have network connectivity to the management network of all ESXi hosts to be added to the VMware Cloud Foundation solution as well network connectivity to Virtual Machines providing External Services as Active Directory, Dynamic Host Configuration Protocol (DHCP), Domain name Service (DNS), Network Time Protocol (NTP).

Network traffic types within VMware Cloud Foundation are isolated from each other via the use of VLANs. Table 5 shows the VLAN IDs configured before initial deployment through HPE OneView. These VLANs and corresponding IP subnets were configured in the network devices including the Arista switches to allow traffic to pass through them.

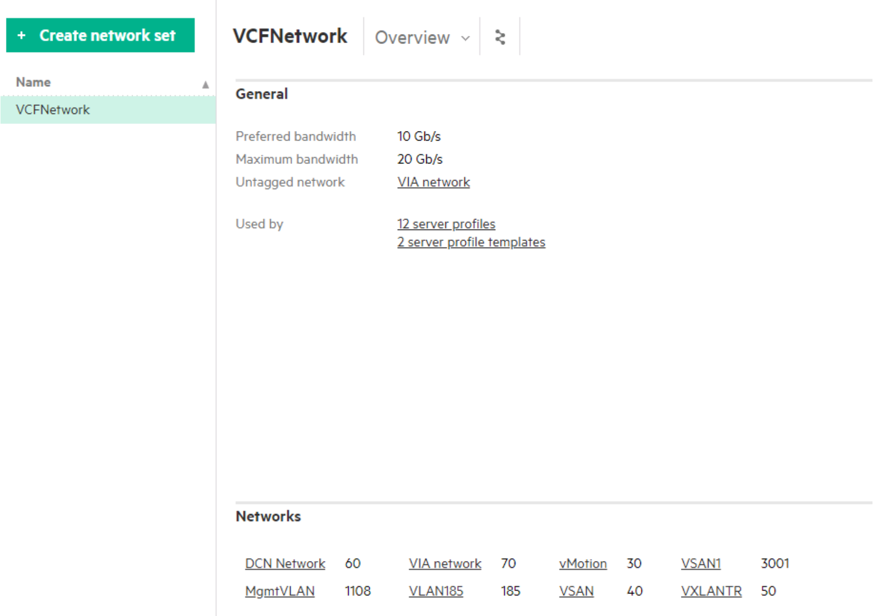
Table 4. VLAN utilized in the solution

VLAN	VLAN ID
VCF-Internal Management Network	1108
vSAN Network	40
vMotion Network	30
VXLAN Transport	50
Data center Network	60

This VLANs, as configured above, are configured in the Network Set and Logical Interconnect Groups configurations through HPE OneView for HPE Synergy.

The following steps summarize the process.

1. Place all configured VCF networks into HPE OneView Network Set as shown in Figure 7. Networks Sets allow multiple VLANs to be carried on the same physical server interface and are used by server profiles.



General

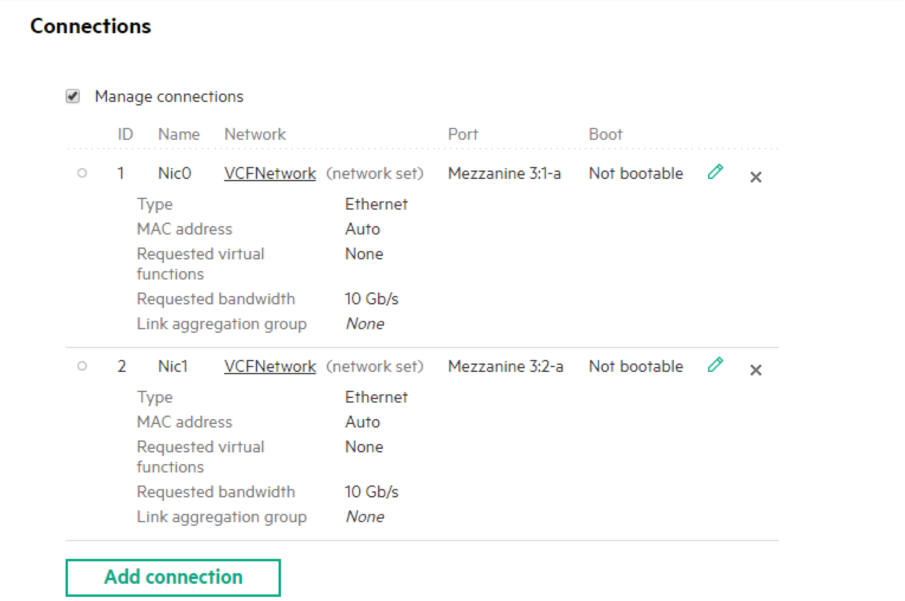
Preferred bandwidth	10 Gb/s
Maximum bandwidth	20 Gb/s
Untagged network	VIA network
Used by	12 server profiles 2 server profile templates

Networks

DCN Network	60	VIA network	70	vMotion	30	VSAN1	3001
MgmtVLAN	1108	VLAN185	185	VSAN	40	VXLANTR	50

Figure 10. HPE OneView Network Sets

2. Create HPE OneView Logical Interconnect Groups with proper uplink sets and create HPE Logical Enclosure Group.
3. Apply Logical Enclosure Groups with correct SPP to create Logical Enclosures.
4. Create HPE OneView Server Profile templates, selecting appropriate hardware type and Enclosure Group created in the previous steps.
5. Select appropriate firmware baseline consistent with vSAN requirements. Select two NICs, each with the network sets.



Connections

☑ Manage connections

	ID	Name	Network	Port	Boot
○	1	Nic0	VCFNetwork (network set)	Mezzanine 3:1-a	Not bootable
○	2	Nic1	VCFNetwork (network set)	Mezzanine 3:2-a	Not bootable

Connection 1:

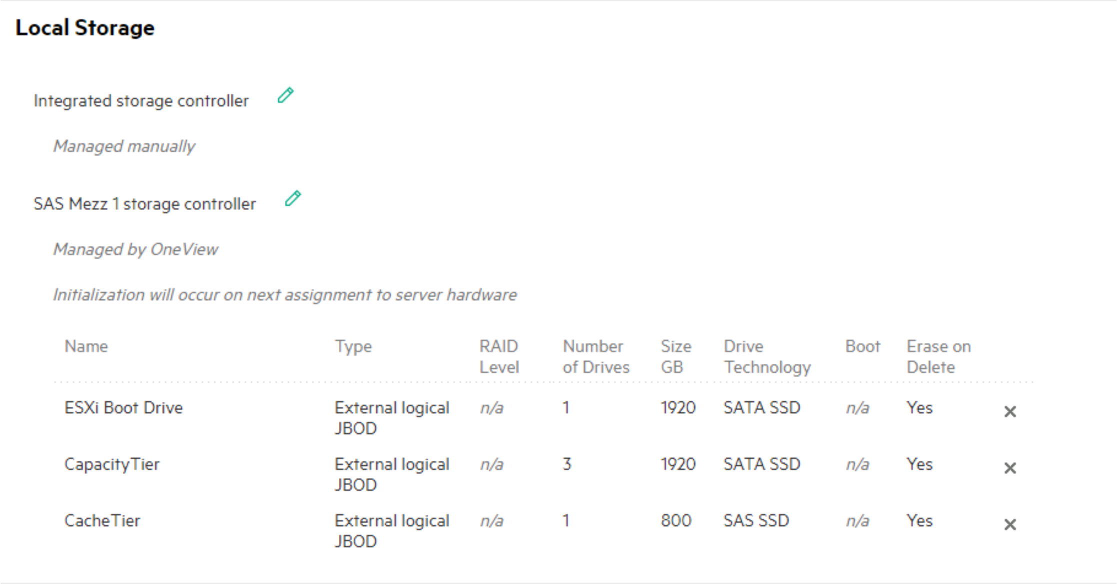
Type	Ethernet
MAC address	Auto
Requested virtual functions	None
Requested bandwidth	10 Gb/s
Link aggregation group	*None*

Connection 2:

Type	Ethernet
MAC address	Auto
Requested virtual functions	None
Requested bandwidth	10 Gb/s
Link aggregation group	*None*

Figure 11. Network configuration in HPE OneView Server Templates

6. Create HPE OneView Server Profiles for each server used for VCF using the HPE OneView Server profile template. VCF needs 4 hosts for “management” and 4 for each “workload” domain.



Local Storage

Integrated storage controller ✎

Managed manually

SAS Mezz 1 storage controller ✎

Managed by OneView

Initialization will occur on next assignment to server hardware

Name	Type	RAID Level	Number of Drives	Size GB	Drive Technology	Boot	Erase on Delete	
ESXi Boot Drive	External logical JBOD	*n/a*	1	1920	SATA SSD	*n/a*	Yes	×
CapacityTier	External logical JBOD	*n/a*	3	1920	SATA SSD	*n/a*	Yes	×
CacheTier	External logical JBOD	*n/a*	1	800	SAS SSD	*n/a*	Yes	×

Figure 12. Storage configuration in HPE OneView Server Templates

VMware Software-Defined Data Center Solution Design

The solution comprises of the following major components: VMware Cloud Foundation (VCF), VMware vRealize Automation (vRA) and vSphere Integrated Containers (VIC) as shown in the figure 11 below.

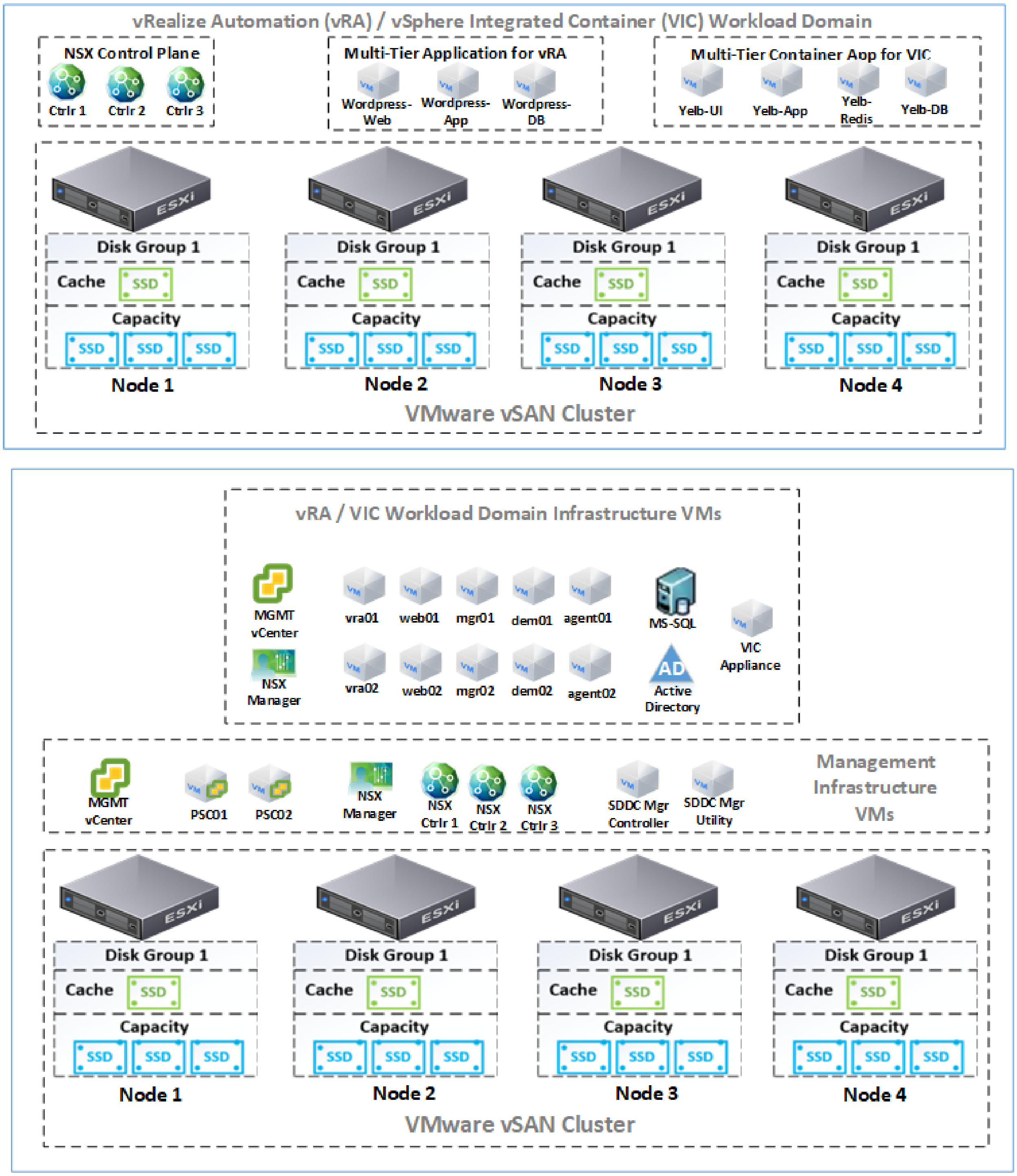


Figure 13. VMware solution components

In this Reference Architecture, we have two workload domains. There are four (4) nodes for the VCF management workload domain and four (4) nodes for the workload domain that is hosting both the WordPress and Yelb applications that are delivered via vRealize Automation blueprints. Both of these workload domains are powered by All-Flash vSAN as the software-defined storage tier with each host consisting of one (1) caching tier SSD and three (3) capacity-tier SSDs.

NSX Manager and controllers are deployed for each workload domain providing security policies that are integrated with vRealize Automation blueprints providing dynamic application level isolation for the multiple tiers within each application and isolation between different applications providing intrinsic security in a data center.

vRealize Automation virtual machines are deployed in the management workload domain as part of the infrastructure VMs and the applications are deployed in the VI workload domain within VCF.

In the following sections, we will cover design best practices for the entire solution focusing on the following aspects:

- VMware Cloud Foundation (VCF) Design building the SDDC platform and workload domains for hosting applications
- NSX providing security policies and its integration with vRealize Automation
- vSAN storage design
- vRealize Automation blueprint design defining how to provision and manage the lifecycle of resources
- VMware Integrated Containers (VIC) design delivering containerized applications within a VM construct and its integration with vRealize Automation providing customers with the ability to manage and deliver containerized applications using the same cloud management portal and familiar blueprints and policy framework

VMware Cloud Foundation (VCF) Design

In VCF, a workload domain is a policy-based resource block with separate capacity, availability, performance and security policies that combines compute (vSphere), storage (vSAN) and networking (NSX) into a single consumable entity. There are two types of workload domains:

- Management workload domain
- Virtual infrastructure workload domain

Management workload domain

The management workload domain is a special-purpose workload domain dedicated to infrastructure and management tasks. It is created during the initial VMware Cloud Foundation bring-up process based on the specifications as given in the deployment configuration spreadsheet. The deployment configuration spreadsheet is downloaded from Cloud Foundation Builder VM and converted to the JSON file format which is used during the Cloud Foundation deployment process. The management workload domain contains a management cluster with 4 VSAN ready nodes. Two HPE Synergy 480 Gen10 compute module from each HPE Synergy 12000 Frame are configured for the management workload domain. These 4 vSAN Ready Nodes are configured as Management Clusters. Within this cluster are a series of virtual machines that have been automatically deployed by Cloud Foundation. These include:

- SDDC Manager
- vCenter Server
- Platform Services Controllers
- NSX Manager
- NSX Controllers
- vRealize Log Insight

Virtual infrastructure workload domain

A virtual infrastructure (VI) workload domain is used to deploy applications. In this Reference Architecture whitepaper, a single virtual infrastructure workload domain is being created, with a single cluster using four (4) Synergy 480 Gen10 compute modules. This workload domain demonstrates both traditional virtual machined based virtualized environment for Microsoft Windows based WordPress application and new cloud native apps such as a vSphere Integrated Container (VIC) based LINUX application - Yelb.

VMware NSX design and configuration guidance

In this deployment, NSX is configured for both management and VI workload domain, to provide security of infrastructure VMs and VI workload domains use virtual machines. The NSX components shown in Figure 14 are deployed in our environment.

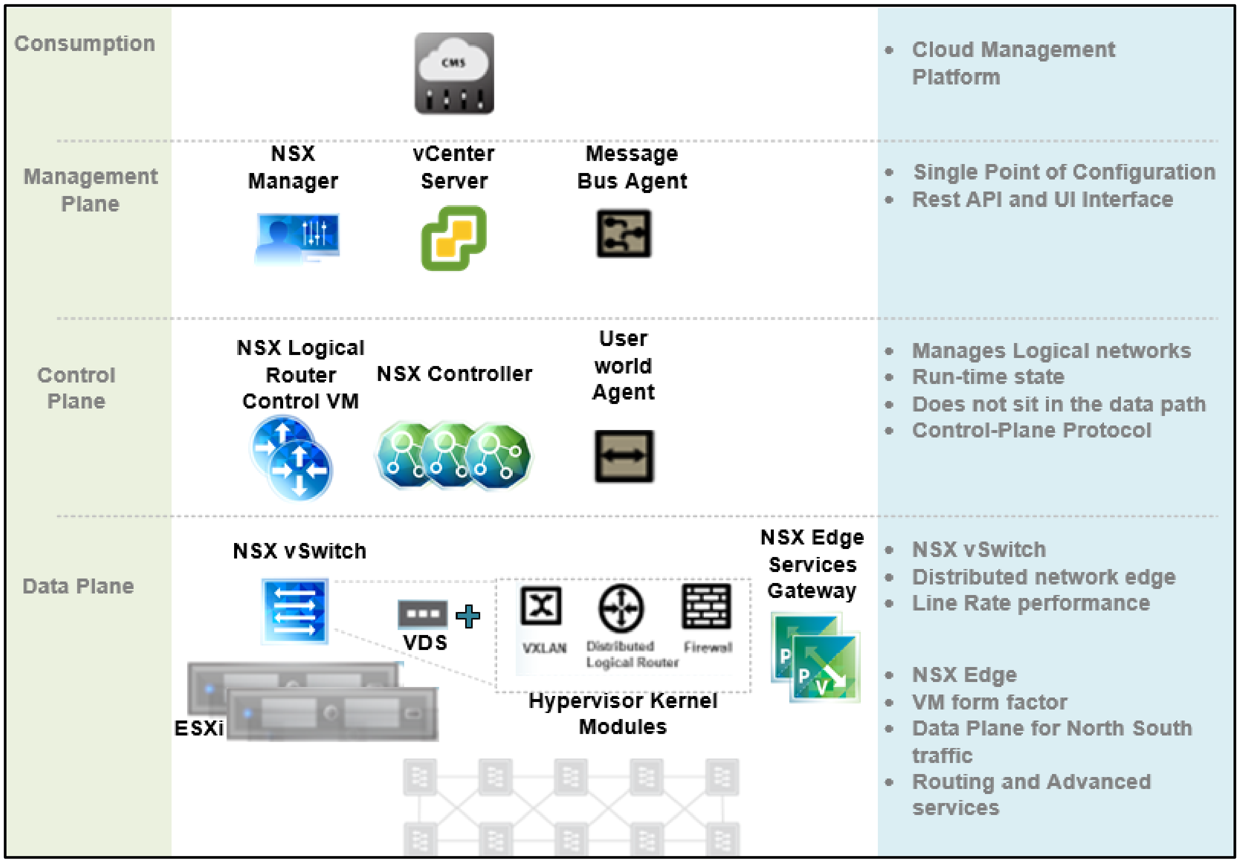


Figure 14. VMware solution component

VMware NSX Manager

NSX Manager is a centralized network management component of NSX. It is installed as virtual appliance on ESXi hosts in vCenter. There's a one to one mapping between an NSX Manager and vCenter Server. It provides a management UI and integrated with vCenter via a vSphere Web Client plugin. It is leveraged to install and configure VXLANs, Distributed Routing, Firewall kernel modules and agents on ESXi hosts. It also deploys NSX controllers and Edge Appliances.

VMware NSX Controller

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 (VXLANs), and distributed logical routers. The controller cluster is responsible for managing the distributed switching and routing modules in the hypervisors. The controller does not have any data plane traffic passing through it. Controller nodes are deployed in a cluster of three members to enable high-availability and scale. Any failure of the controller node does not impact any data-plane traffic.

VMware NSX Edge

NSX Edge can be deployed as an Edge Services Gateway (ESG) or as a Distributed Logical Router (DLR). The ESG gives you access to all NSX Edge services such as firewall, NAT, DHCP, VPN, load balancing, and high availability. ESG has uplink interfaces that connect to uplink port groups which have access to a shared corporate network or a service that provides access layer networking. Multiple external IP addresses can be configured for load balancer, site-to-site VPN, and NAT services. The internal interfaces of an ESG connect to secured port groups and act as the gateway for all protected virtual machines in the port group. On the other hand, DLR provides East-West distributed routing with tenant IP address space and data path isolation. It has uplink interface which connects to an ESG, with an intervening Layer 2 logical transit switch between the DLR and the ESG. An internal interface on a DLR peers with a virtual machine hosted on an ESX hypervisor with an intervening logical switch between the virtual machine and the DLR.

The following diagram comprises of one NSX Manager and three Controllers. NSX Manager maps to a single vCenter Server environment. Therefore, each workload domain includes one NSX Manager instance and three NSX Controller instances. Data center administrators can use the vSphere Web Client to perform additional NSX configuration required by the specific VMs deployed within the workload domain.

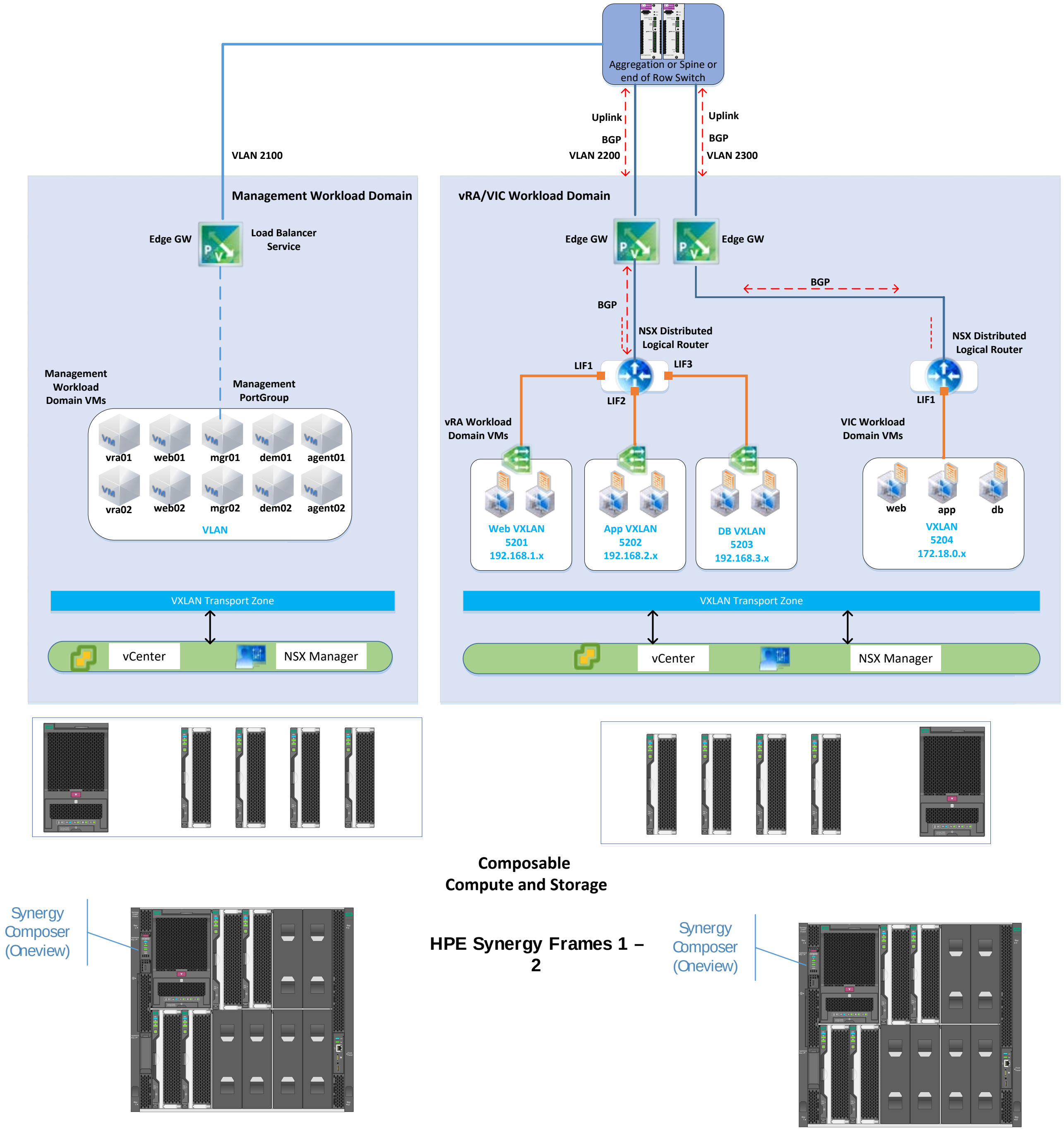


Figure 15. Solution design for VMware NSX

VMware vSAN design

In this Reference Architecture, both management and vRA/VIC workload domains are built with four (4) node vSAN clusters. Each vSAN Cluster is designed with a single disk group per host consisting of one (1) Cache tier SSD and three (3) capacity tier SSDs. The SDDC Manager provides full lifecycle management automation for vSAN.

For the underlying HPE Synergy 480 Gen 10 Compute Modules, 1.92TB SATA SSD and 800 GB SAS SSD is provided by the D3940 Storage module as shown in the figure below.

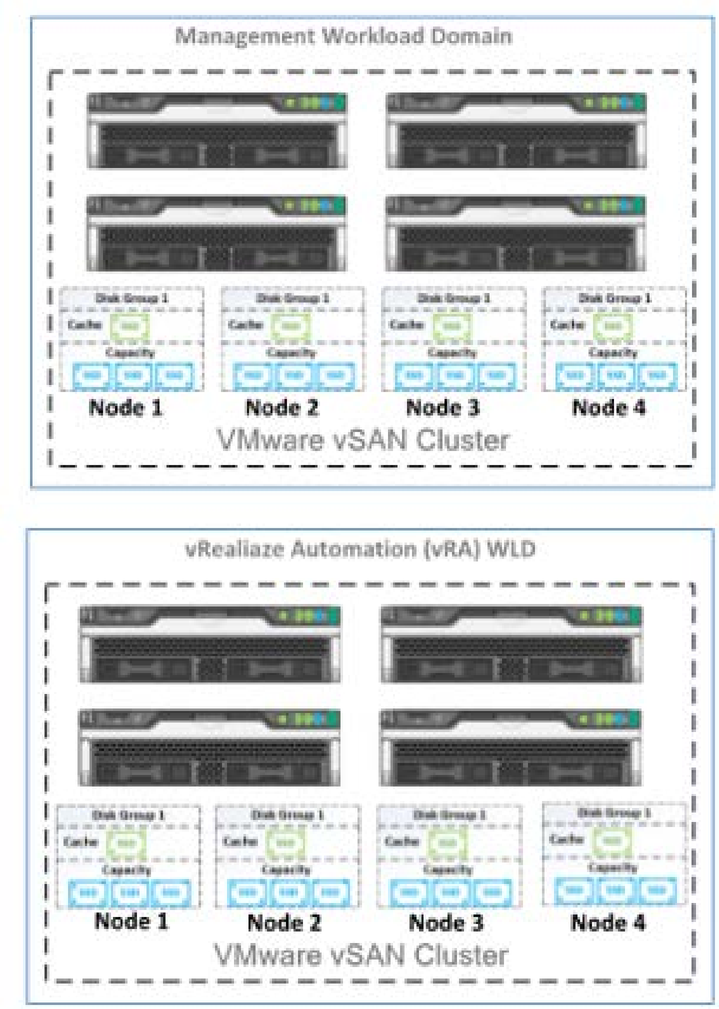


Figure 16. Solution design for VMware vSAN

Table 5 describes details about the vSAN configuration used in this Reference Architecture design.

Table 5. vSAN configuration

Workload Domains	No. of nodes in the vSAN Cluster	No. Of disk groups in each node	No of cache disk in each disk group	No of capacity disk in each disk group
Management Workload domain	4	1	1	3
vRA/VIC Workload domain	4	1	1	3

vRealize Automation blueprint design

vRealize Automation is the cloud management portal for traditional VM workloads and containerized workloads. With vRA integration with NSX, virtualized networks and integrated container networks can co-exist in a single environment.

For this Reference Architecture, multi-machine application vRA blueprint for multi-tier application is used, WordPress is created within vRA's blueprint designer as well as for a blueprint for Yelb which is delivered via VMware Integrated Containers. Blueprint designer is tightly integrated with NSX and contains NSX components like load balancer and security groups that can be deployed for the respective applications.

The following steps provide detail for the multi-machine blueprint for WordPress and Yelb.

WordPress

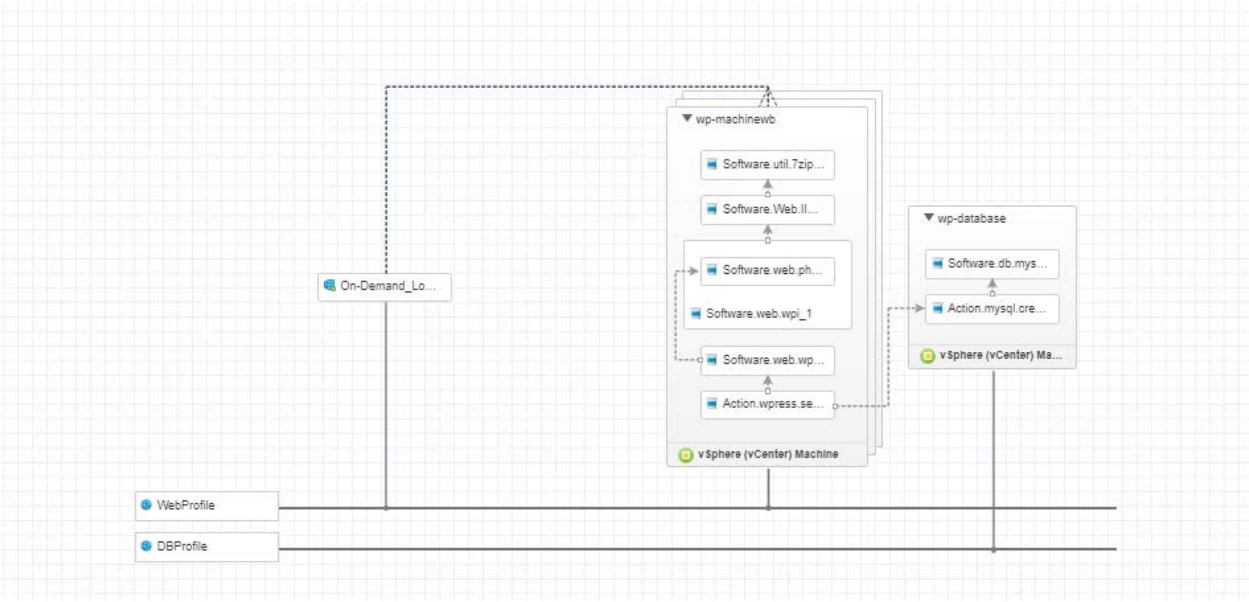


Figure 17. Multi-machine blueprint for WordPress.

Table 6 describes details about virtual machine configuration for WordPress application used in this Reference Architecture design.

Table 6. WordPress Virtual Machine configuration

VM Name	Operating System	Number of VM's	vCPU	RAM	Storage
wp-machineweb	Windows 2012 Server	Min -2 Max -3	1	2 GB	50 GB
wp-database	Windows 2012 Server	1	1	2 GB	50 GB
On-Demand_Load_Balancer	Linux	1	1	2 GB	1 GB

Note

A private repository should be created to host the software components. Modify the blueprint configuration and add the repository URL for every component

Yelb

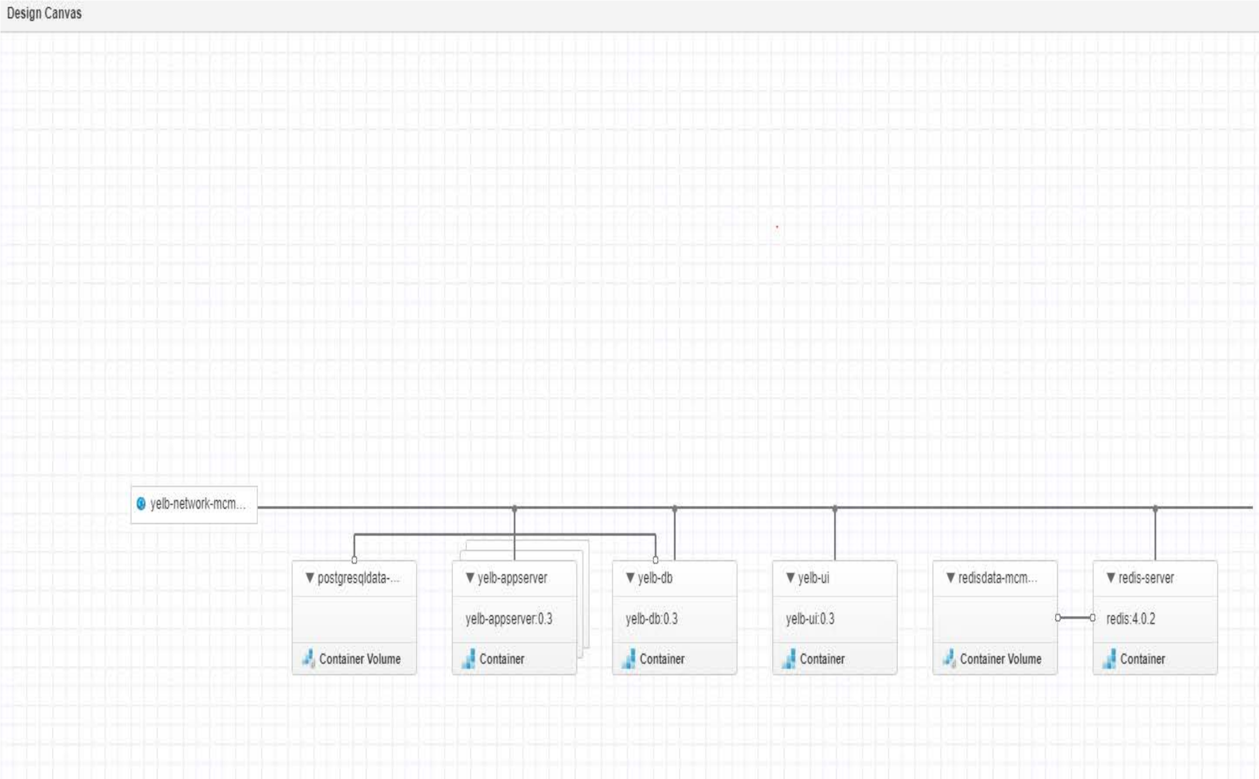


Figure 18. Multi-machine blueprint for Yelb

Table 7 describes details about Yelb configuration used in this Reference Architecture design.

Table 7. Yelb configuration

VM Name	vCPU	RAM	Network	Number of VM's
Yelb-Appserver	2	2 GB	Bridge	2
Yelb-DB	2	2 GB	Bridge	1
Yelb-UI	2	2 GB	Bridge	1
Redis Server	2	2 GB	Bridge	1

vSphere Integrated Containers (VIC) design

vSphere Integrated Containers is delivered as an ova appliance that comprises of the following major components:

- vSphere Integrated Containers Engine, a container runtime for vSphere that allows you to provision containers as virtual machines, offering the same security and functionality of virtual machines in VMware ESXi™ hosts or vCenter Server® instances.
- vSphere Integrated Containers Plug-In for vSphere Client, that provides information about your vSphere Integrated Containers setup and allows you to deploy virtual container hosts directly from the vSphere Client.
- vSphere Integrated Containers Registry, an enterprise-class container registry server that stores and distributes container images. vSphere Integrated Containers Registry extends the Docker Distribution open source project by adding the functionalities that an enterprise requires, such as security, identity and management.
  - vSphere Integrated Containers Management Portal, a container management portal that provides a UI for DevOps teams to provision and manage containers, including the ability to obtain statistics and information about container instances. Cloud administrators can manage container hosts and apply governance to their usage, including capacity quotas and approval workflows. Cloud administrators can create projects and assign users and resources such as registries and virtual container hosts to those projects.

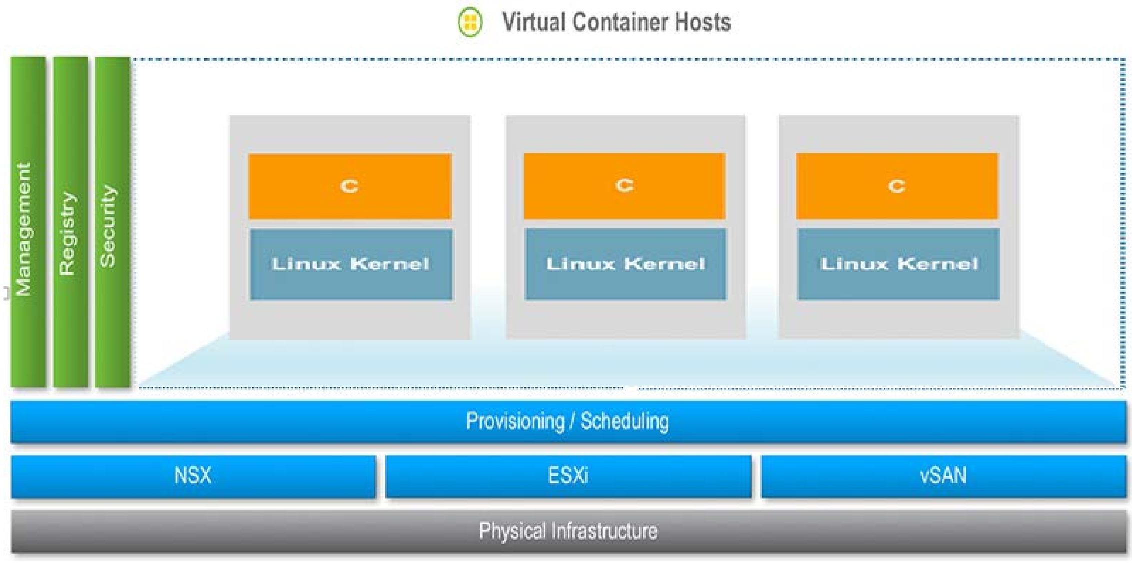


Figure 19. vSphere Integrated Container Design

Virtual Container Host - A Virtual Container Host (VCH) is a multi-functional appliance that you deploy as a vApp in a vCenter Server cluster or as a resource pool on an ESXi host. The vApp or resource pool provides a useful visual parent-child relationship in the vSphere Web Client so that you can easily identify the container VMs that are provisioned into a VCH. You can also specify resource limits on the vApp. Multiple VCHs can be provisioned onto a single ESXi host, into a vSphere resource pool, or into a vCenter Server cluster.

vSphere Integrated Container Networks - The vSphere Integrated Containers Engine uses different network types for different purposes:

- Management network: This network is dedicated to communication between the VCH, vCenter Server, and ESXi hosts.
- Public network: This network, which is mandatory, connects containers to the Internet.

Note
If the lab where VCH is getting deployed doesn't have internet access then a private Docker repository needs to be created. This repository will host the Docker images for containerized application going to be deployed on VCH.

- Client network: This network connects Docker clients to Docker endpoints and isolates the endpoints from the public network.
- Bridge network: This network allows the containers and the VCH to communicate with each other. Each VCH requires a unique bridge network.
- Container network: This type of network is used to connect containers directly to vSphere networks without routing through the VCH endpoint VM using NAT

vSphere Integrated Container Storage - Virtual container hosts (VCHs) require a datastore in which to store container image files, container VM files, and the files for the VCH itself. One or more datastores can be specified while deploying VCH.

In this Reference Architecture, a VCH blueprint is designed in vRA to deploy VCH. For networking, one (1) VCH public network DVS port group and one (1) NSX logical switch for container network are created. For storage, persistent volumes are used to achieve data persistence for application deployment. The vRA blueprint created for containerized application, Yelb, includes both - persistent volumes and NSX logical switch port configurations.

Use Cases for the solution

The following Use Cases are implemented for this Reference Architecture.

- Use Case 1: Demonstrate a traditional multi-tier application and a containerized application deployment
- Use Case 2: Auto Provisioning of Network uplink and downlink ports using Ansible scripts
- Use Case 3: Demonstrate ease of monitoring and reporting of VCF infrastructure using HPE OneView for vRealize Operations

Use Case 1: Demonstrate a traditional multi-tier application and a containerized application deployment

In this Reference Architecture, we have deployed WordPress as a traditional multi-tier application using vRA and leveraged NSX integration to provide security and the required application isolation between the web, app and db tier. This security framework can be expanded to provide isolation between multiple applications. We have deployed Yelb as a containerized application using VIC and vRA integration. The following sections provide details on both of these use cases:

Deploying WordPress with vRA and NSX

WordPress is an application developed in PHP. The database used is MySQL and the front end is managed by using PHP. Instead of storing static pages, WordPress creates the website pages dynamically using PHP and MySQL.

In this solution, an external network profile and routed network profile are configured for the application deployment with 1:1 mapping between external profile and existing NSX logical switches. This network will be part of the blueprint which is used to provision WordPress. NSX Edge Services Gateway (ESG) with on-demand load balancing service running is automatically deployed as part of the WordPress blueprint.

The following table describes the communication flow enabled by NSX Micro-segmentation polices that allowed between multiple tiers of WordPress with a default "deny all" rule at the bottom ensuring that any traffic not explicitly allowed is denied.

Table 8. NSX Distributed Firewall rules -Communication flow by NSX Micro-segmentation policies

Name	Source	Destination	Service	Action	Applied To
Allow to NTP, DNS	Any	NTP, DNS	DNS, NTP	Allow	
Allow Web	Web LB	Web Servers	80	Allow	Web-Tier
Allow Web to APP	Web Servers	App LB	8080	Allow	Web-Tier, App-Tier
Allow App	App LB	App Servers	8080	Allow	App-Tier
Allow App to DB	App Server	DB Server	3306	Allow	App-Tier, DB-Tier
Deny other	Any	Any	Any	Deny	

These NSX distributed firewall rules are defined within NSX Service composer establishing dynamic membership rules for each tier. When the VMs pertaining to specific tiers get deployed, they will be automatically placed in their respective security groups. For accessing the application externally, only a set of services on a specific tier (WEB) will be allowed on the required ports. Based on these policies, the multi-tier application with traffic flow is depicted in the below diagram:

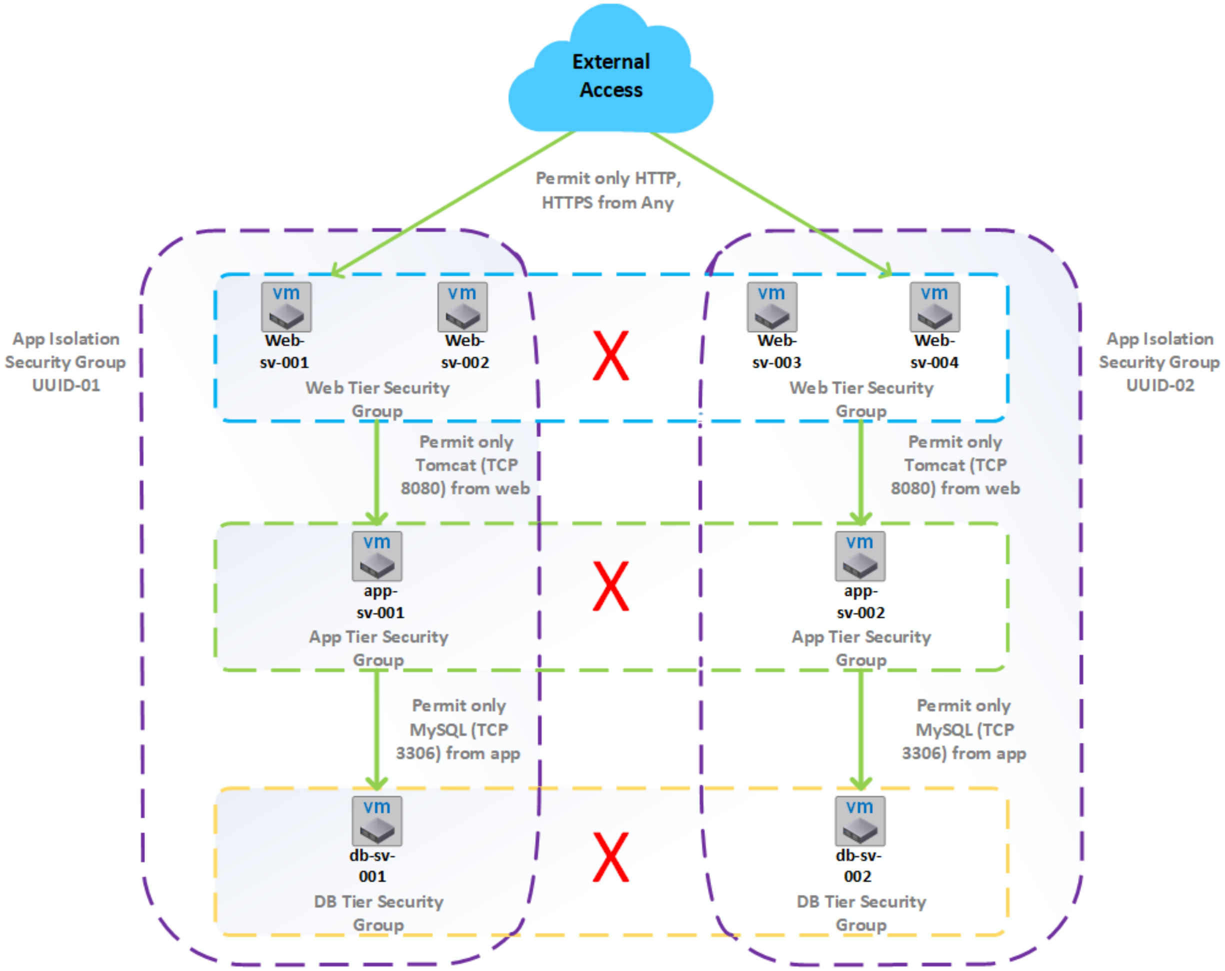

External Access
Permit only HTTP, HTTPS from Any
App Isolation Security Group UUID-01
App Isolation Security Group UUID-02
vm
Web-sv-001
Web-sv-002
Web-sv-003
Web-sv-004
Web Tier Security Group
Web Tier Security Group
X
Permit only Tomcat (TCP 8080) from web
Permit only Tomcat (TCP 8080) from web
app-sv-001
app-sv-002
App Tier Security Group
App Tier Security Group
X
Permit only MySQL (TCP 3306) from app
Permit only MySQL (TCP 3306) from app
db-sv-001
db-sv-002
DB Tier Security Group
DB Tier Security Group
X

Figure 20. Traffic flow

The NSX On-demand load balancer component in vRA blueprint takes care of the traffic flow amongst the VMs to a specific tier. It is deployed as an Edge Gateway in NSX with load balancing services enabled and algorithm set as Round-Robin.

Deploying Yelb with vSphere integrated containers and vRA

Yelb application is a multi-tiered containerized voting application which consists of 4 containers: yelb-ui, yelb-appserver, Redis-server, yelb-db. The current architecture layout for the Yelb application is as shown below.

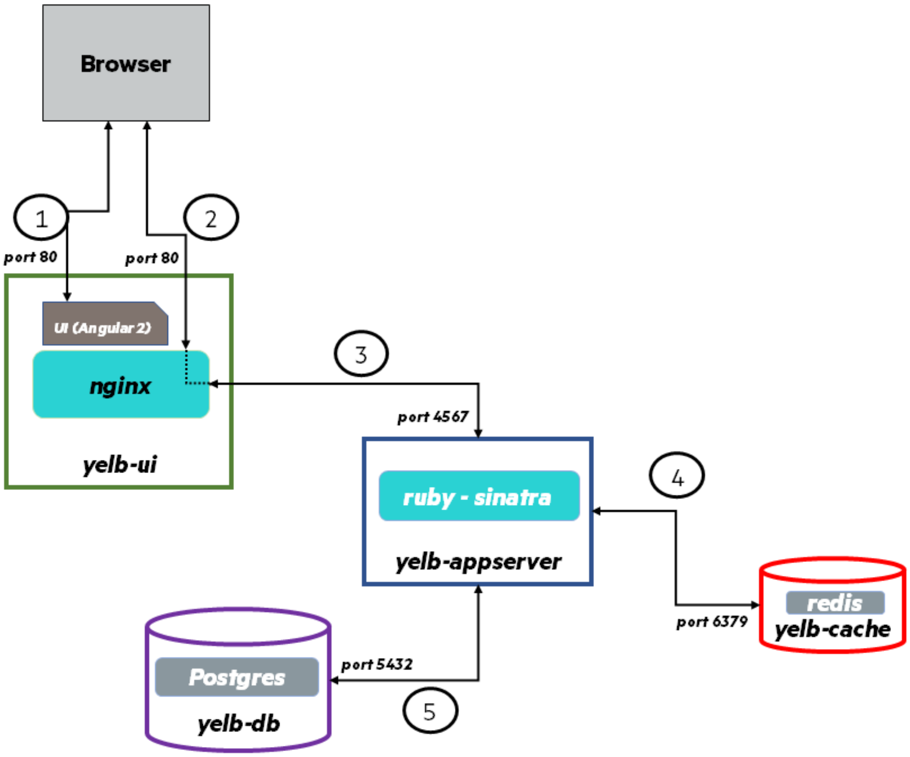
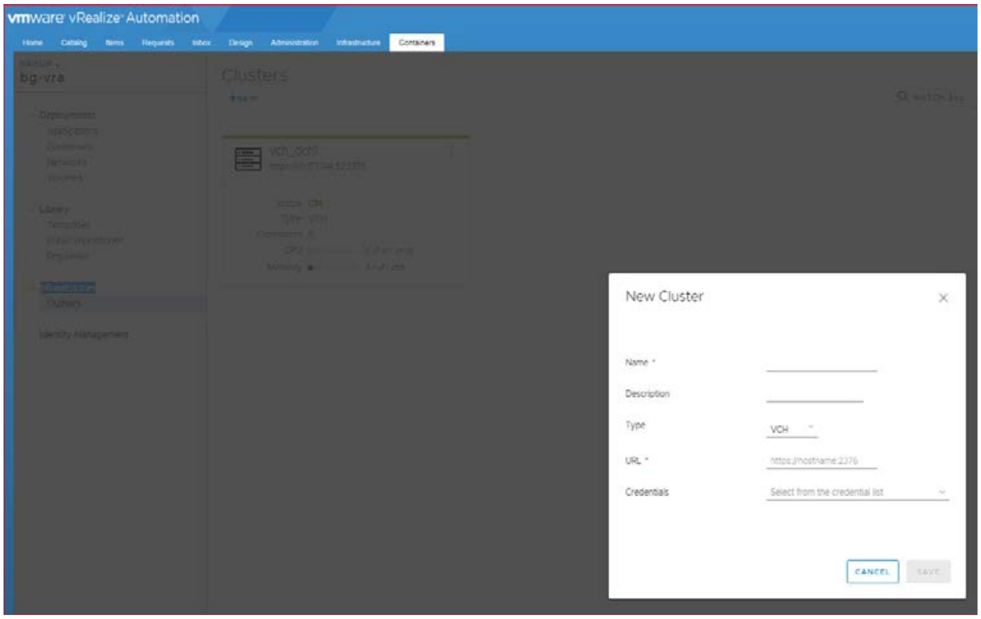


Figure 21. Architecture layout for Yelb application

Yelb-ui is a frontend component of Yelb that fulfills a couple of roles. The first role is to host the Angular 2 applications that is the UI of Yelb application. When the browser connects to this layer it downloads the JavaScript code that builds the UI itself. Subsequent requests and calls to other application components are proxied via the nginx service running on yelb-ui. Yelb-appserver is a Sinatra application that basically read and write to a cache server (redis-server) as well as a Postgres backend database (yelb-db). Redis Server is used to store the number of page views whereas Postgres is used to persist the votes.

The first step is to deploy VIC and its management portal by following steps mentioned here. Once VIC is deployed, next step is to deploy VMware Container host (VCH) which enables provisioning of the containers, VCH can be deployed by following the steps mentioned here. After deploying VCH, add it as a container infrastructure into vRA to be able to leverage a common blueprint and management framework. This is accomplished through the integration between VIC and vRA as shown below.



New Cluster

Name *

Description

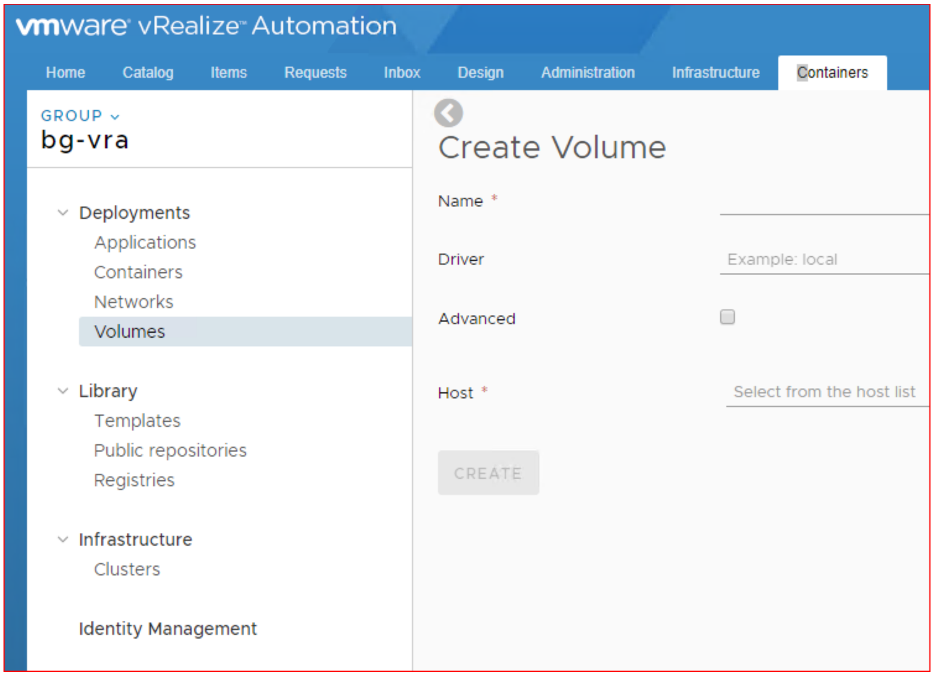
Type VCH

URL * https://hostname:2376

Credentials Select from the credential list

Figure 22. Add a new VCH cluster

For achieving storage persistency for Yelb, persistent volumes are leveraged and created in vRA through the 'Containers-> Volumes' tab.



Create Volume

Name *

Driver Example: local

Advanced ☐

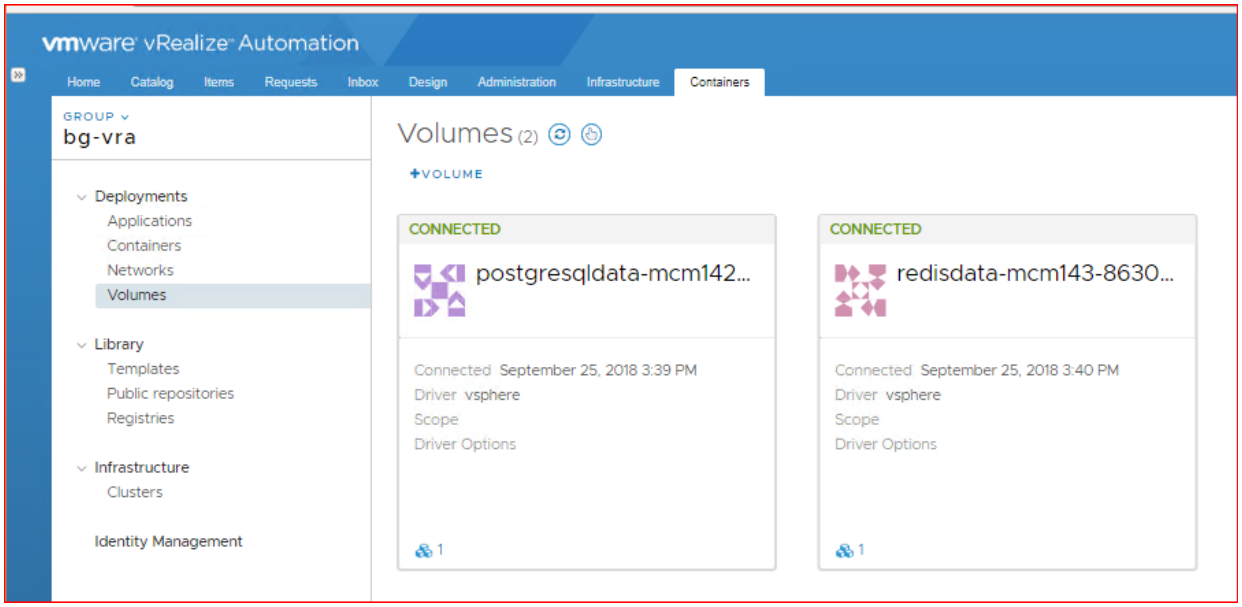
Host * Select from the host list

Figure 23. Create persistent volume

Note

The host in above image represents VCH.

For Yelb, we have created two persistent volumes, one each for Redis-server and for yelb-db container as shown below.



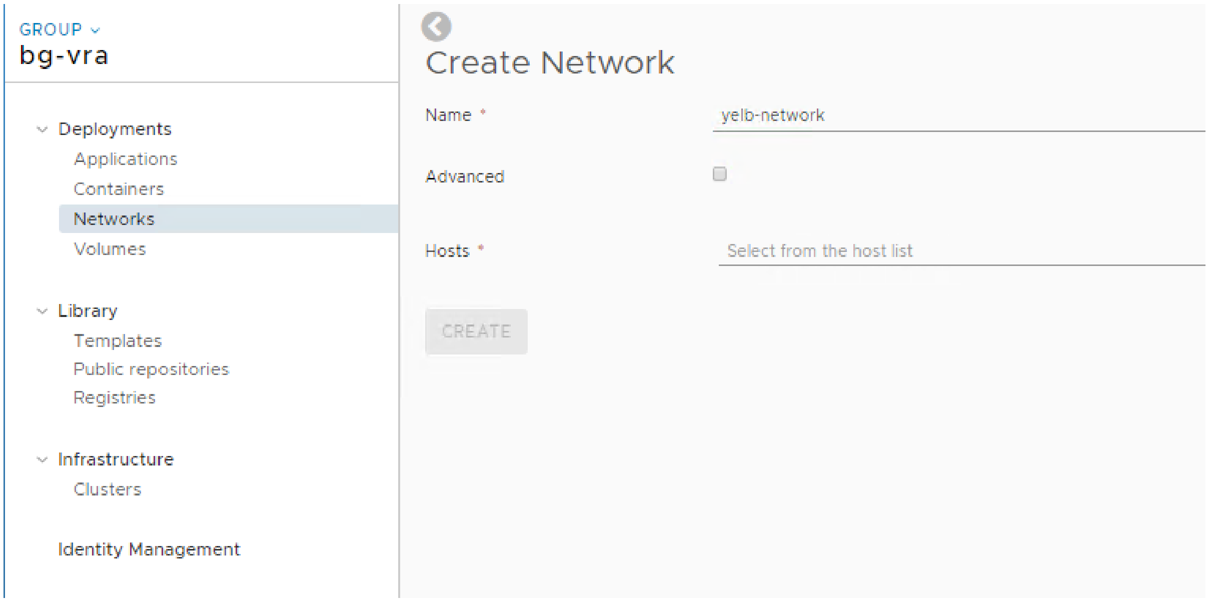
Volumes (2)

CONNECTED
postgresqldata-mcm142...
Connected September 25, 2018 3:39 PM
Driver vsphere
Scope
Driver Options

CONNECTED
redisdata-mcm143-8630...
Connected September 25, 2018 3:40 PM
Driver vsphere
Scope
Driver Options

Figure 24. Persistent volumes for containers

For networking we need to create bridge network for yelb application which can be done through ‘Container-> Networks’ tab as shown below.



Create Network

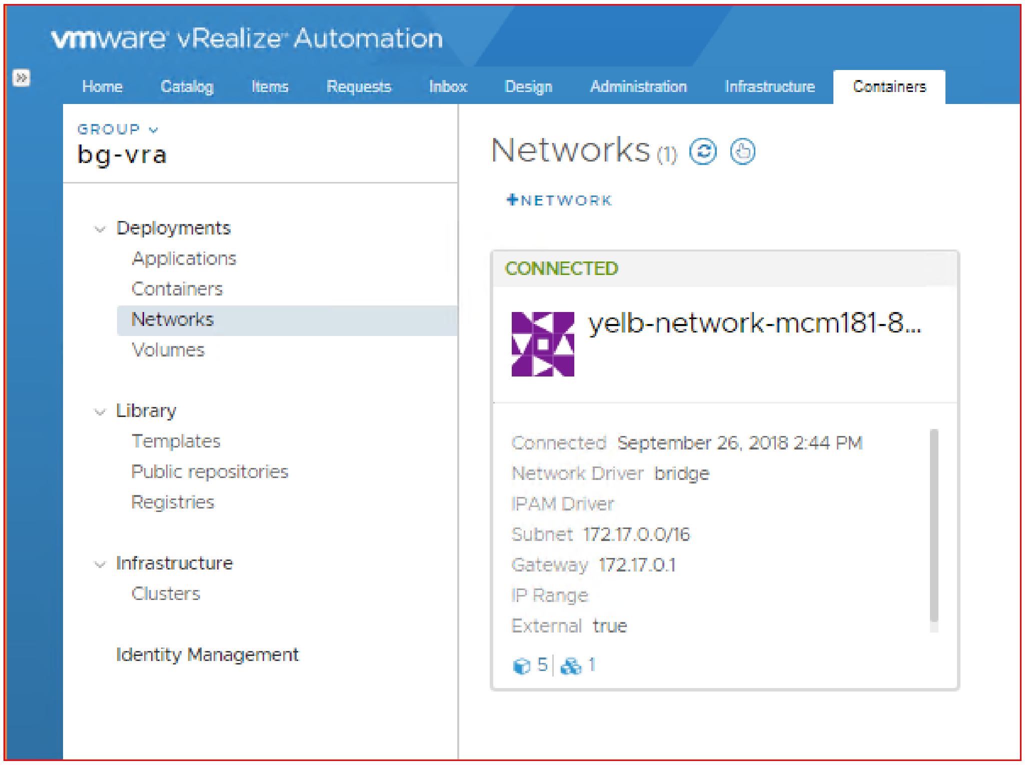
Name * yelb-network

Advanced □

Hosts * Select from the host list

Figure 25. Create bridge network for application

Enter the name of the network and select the VCH host created in in the previous step. After creating the network, it should look like the image below.



yelb-network-mcm181-8...

Connected September 26, 2018 2:44 PM
Network Driver bridge
IPAM Driver
Subnet 172.17.0.0/16
Gateway 172.17.0.1
IP Range
External true

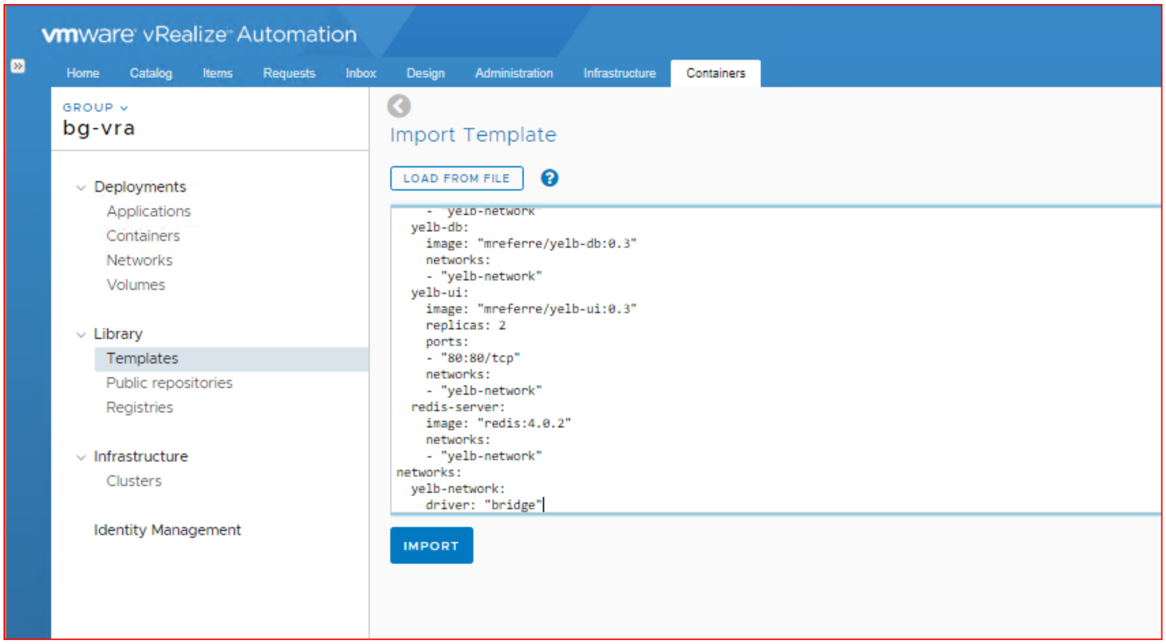
Figure 26. Bridge network for containerized application

The configured network, as shown in Figure 26, will use same the port groups or logical switches of NSX which had been used by VCH. After configuring storage and network, we can now import the containerized application template (.yaml) into vRA. The content of the template file is as mentioned below.

```
---
version: ”2”
services:
  yelb-db:
    image: ”mreferre/yelb-db:0.3”
    volumes:
    - ”postgresqldata-mcm142-86308784973:/var/lib/postgresql/data”
    networks:
    - ”yelb-network-mcm181-86391873781”
  yelb-appserver:
    image: ”mreferre/yelb-appserver:0.3”
    restart: ”always”
    networks:
    - ”yelb-network-mcm181-86391873781”
  yelb-ui:
    image: ”mreferre/yelb-ui:0.3”
    ports:
    - ”80:80/tcp”
    networks:
```

```
    - ”yelb-network-mcm181-86391873781”
  redis-server:
    image: ”redis:4.0.2”
    volumes:
    - ”redisdata-mcm143-86308821985:/data”
    restart: ”no”
    networks:
    - ”yelb-network-mcm181-86391873781”
networks:
  yelb-network-mcm181-86391873781:
    external: true
volumes:
  postgresqldata-mcm142-86308784973:
    external: true
  redisdata-mcm143-86308821985:
    external: true
```

Modify the highlighted values in the file as per your environment. Namely, the network and volumes created previously. Import the template in vRA by pasting the contents of the file as shown below.



```
    - "yelb-network"
  yelb-db:
    image: "mreferre/yelb-db:0.3"
    networks:
    - "yelb-network"
  yelb-ui:
    image: "mreferre/yelb-ui:0.3"
    replicas: 2
    ports:
    - "80:80/tcp"
    networks:
    - "yelb-network"
  redis-server:
    image: "redis:4.0.2"
    networks:
    - "yelb-network"
networks:
  yelb-network:
    driver: "bridge"
```

Figure 27. Import containerized application template

After importing, you should see a template named ‘yelb’ under ‘Templates’. Click on it and it should look like the image below.

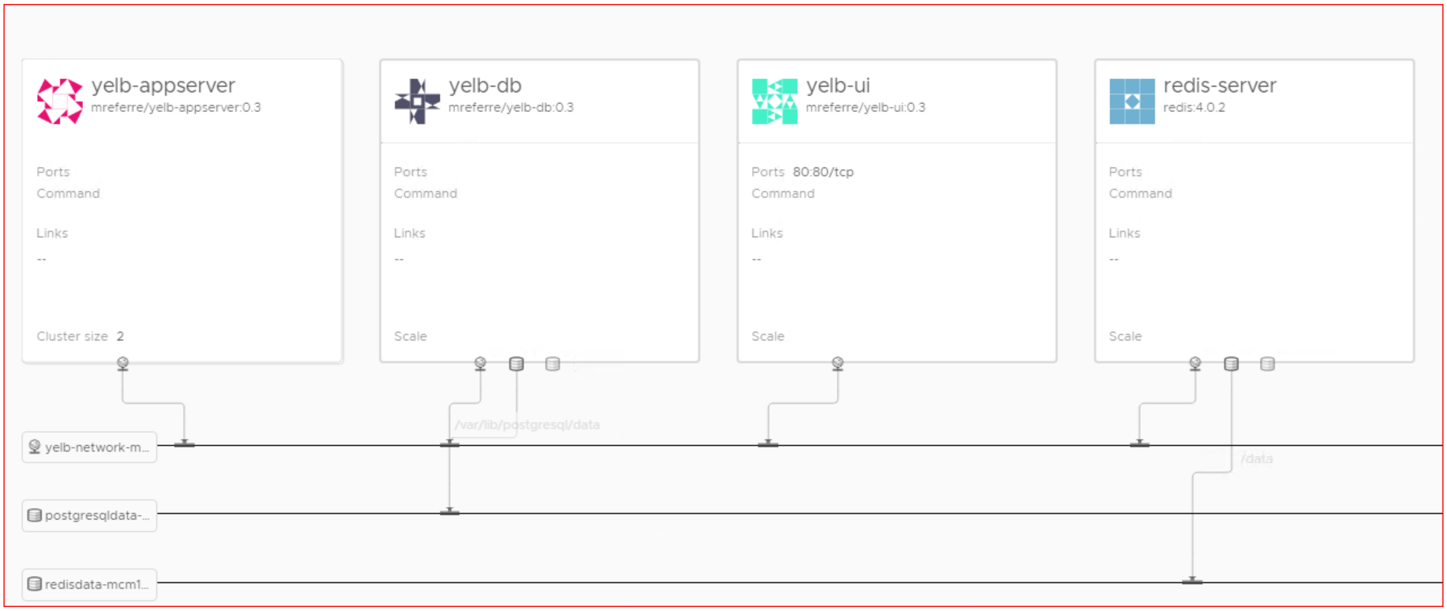
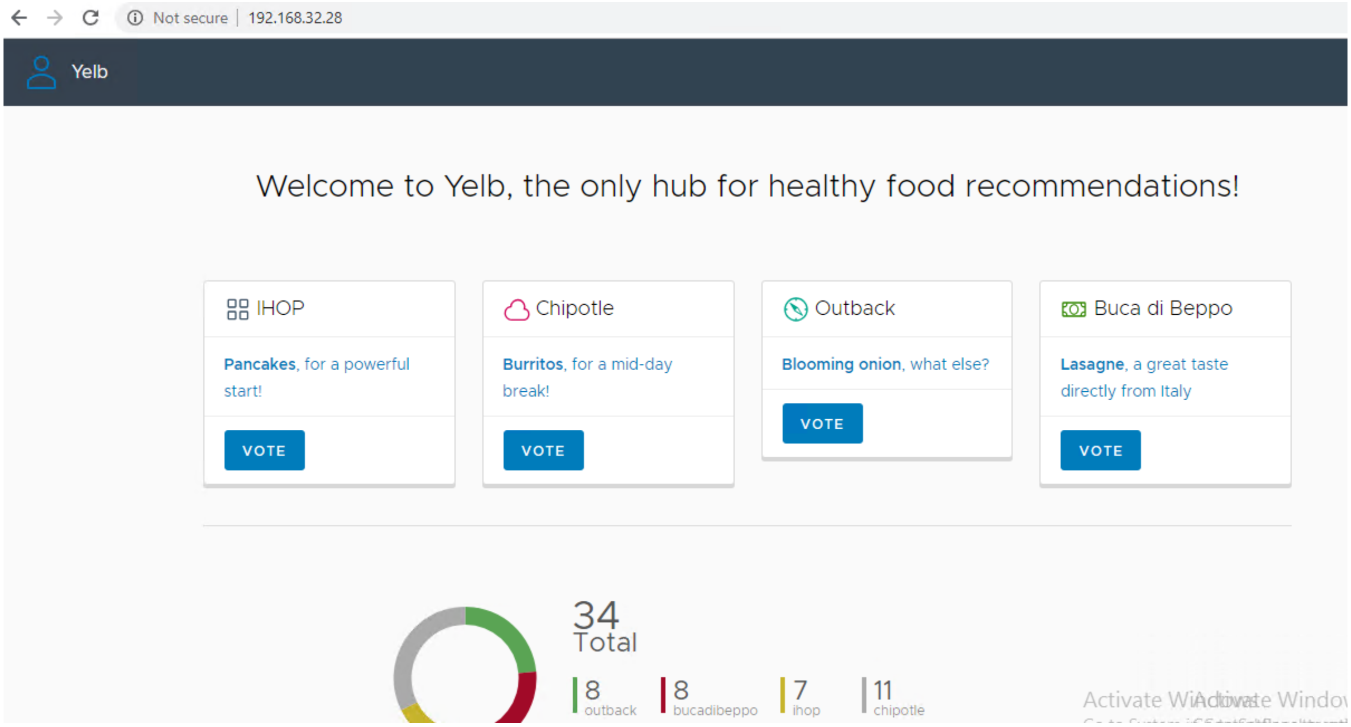


Figure 28. Yelb application template

This template is now leveraged to provision and publish vRA blueprint for Yelb. Export this template as a YAML Blueprint and it should then be visible under ‘Design->Blueprints’. After publishing this blueprint and making it available as a service to consume from vRA catalog, request and submit the application for deployment. Upon successful deployment Yelb can be accessed through VCH IP:Port as defined.



Welcome to Yelb, the only hub for healthy food recommendations!

IHOP
Pancakes, for a powerful start!
VOTE

Chipotle
Burritos, for a mid-day break!
VOTE

Outback
Blooming onion, what else?
VOTE

Buca di Beppo
Lasagne, a great taste directly from Italy
VOTE

Figure 29. Successful application deployment

Use Case 2: Auto provisioning of Network uplink and downlink ports using Ansible scripts

This Reference Architecture uses the Ansible Playbook for various configuration activity for the inter-rack Arista switches. The following is the list of activity which can achieved using the playbook for automation.

- Create VLAN in the switch
- Add the VLAN to a Trunk group
- Shut down an interface and bring up a switch interface
- Show technical details (for diagnostics).

The following list shows the pre-requisites for running the playbook.

- EOS_config module should be loaded.
- The Arista Switch Privilege EXEC Mode should be enabled with “No Secret”.
- The Arista Switch should support transport: eapi

Details of the Playbook are provided in the Appendix B: Ansible Playbook.

Use case 3: Demonstrate ease of monitoring and reporting of VCF infrastructure using HPE OneView for vRealize Operations

HPE OneView for VMware vRealize Operations provides an integrated monitoring and reporting tool for VCF Infrastructure.

When the HPE OneView for VMware vRealize Operations is installed, the custom HPE OneView Dashboards are added to the vRealize Operations custom GUI. The HPE OneView Dashboards allow you to proactively monitor HPE Synergy hardware resources and also shows the object relationship with other objects in the environments. A proactive monitoring of the HPE Synergy hardware used for VMware Cloud Foundation helps in improve productivity, efficient use of the resources and hence minimizing cost.

Following are the dashboards available for use:

- HPE OneView Infrastructure Dashboard
- HPE OneView Networking Dashboard
- HPE OneView Servers Overview Dashboard
- HPE OneView Enclosure Overview Dashboard
- HPE OneView Uplink Port Overview Dashboard

The following sections show examples of four among above five dashboards.

HPE OneView Infrastructure dashboard

The HPE OneView Infrastructure dashboard provides an overview of the entire HPE Synergy infrastructure managed by HPE OneView. It displays the status of HPE OneView managed hardware and allows you to see how the hardware relates to your virtual environment. Figure 30 shows an example of HPE OneView Infrastructure dashboard. It displays the entire VCF environment physical and virtual resources and interrelation between each objects in the environments. Thus the administrator gets a quick summary of the environment as well as an idea how an objects it related to other objects. It also shows metric charts demonstrating the performance and usage of the resources.

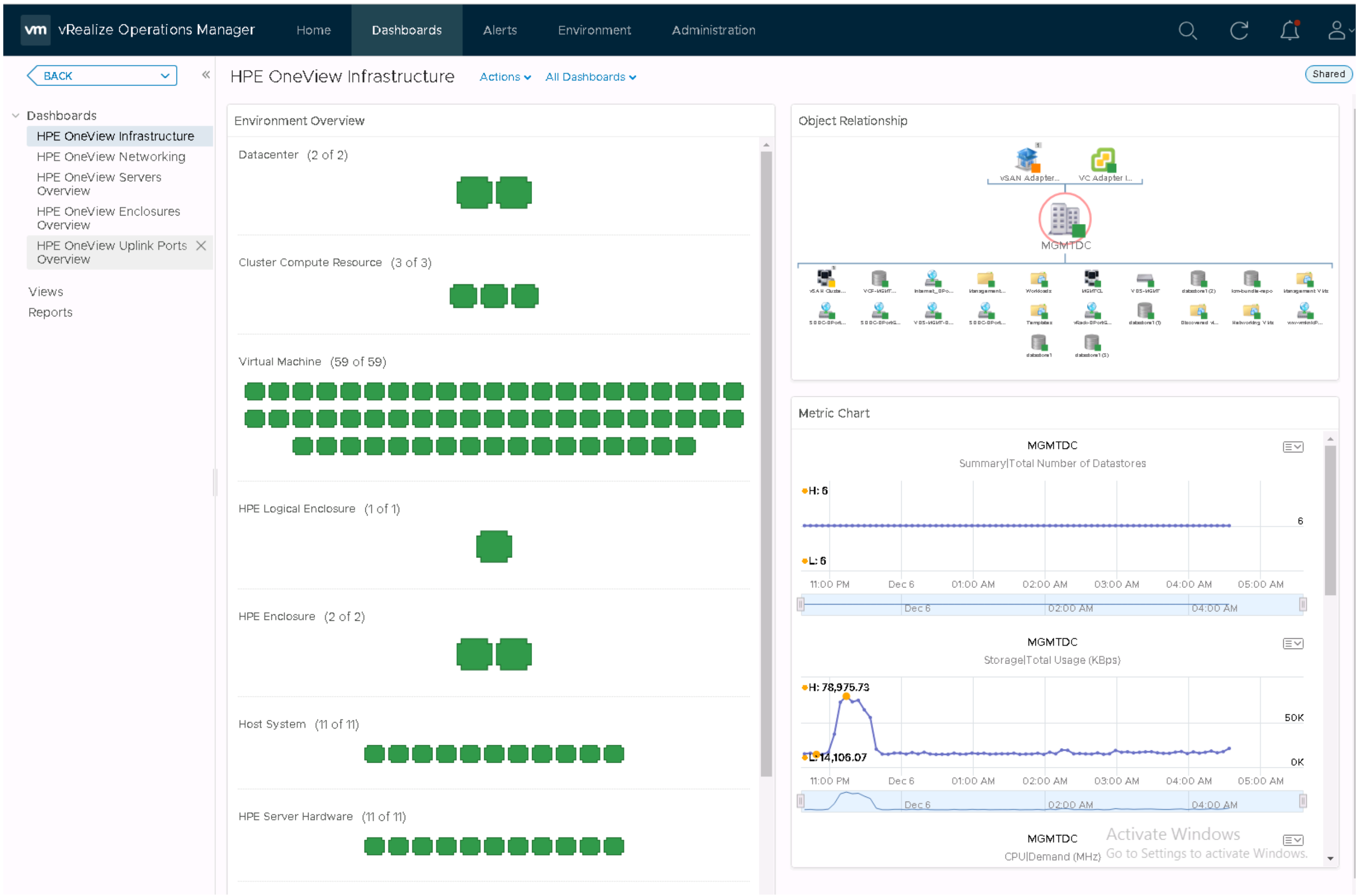


Figure 30. HPE OneView Infrastructure dashboard

HPE OneView Networking dashboard

The HPE OneView Networking dashboard provides an overview of the HPE OneView networking along with its connection to the virtual environment. Selecting an object in the Environment Overview allows you to see how this object relates to other objects and generates a graph for each metric collected. Figure 31 shows an example of HPE OneView Networking dashboard with VCF-Node1 Server selected and its necessary connections.

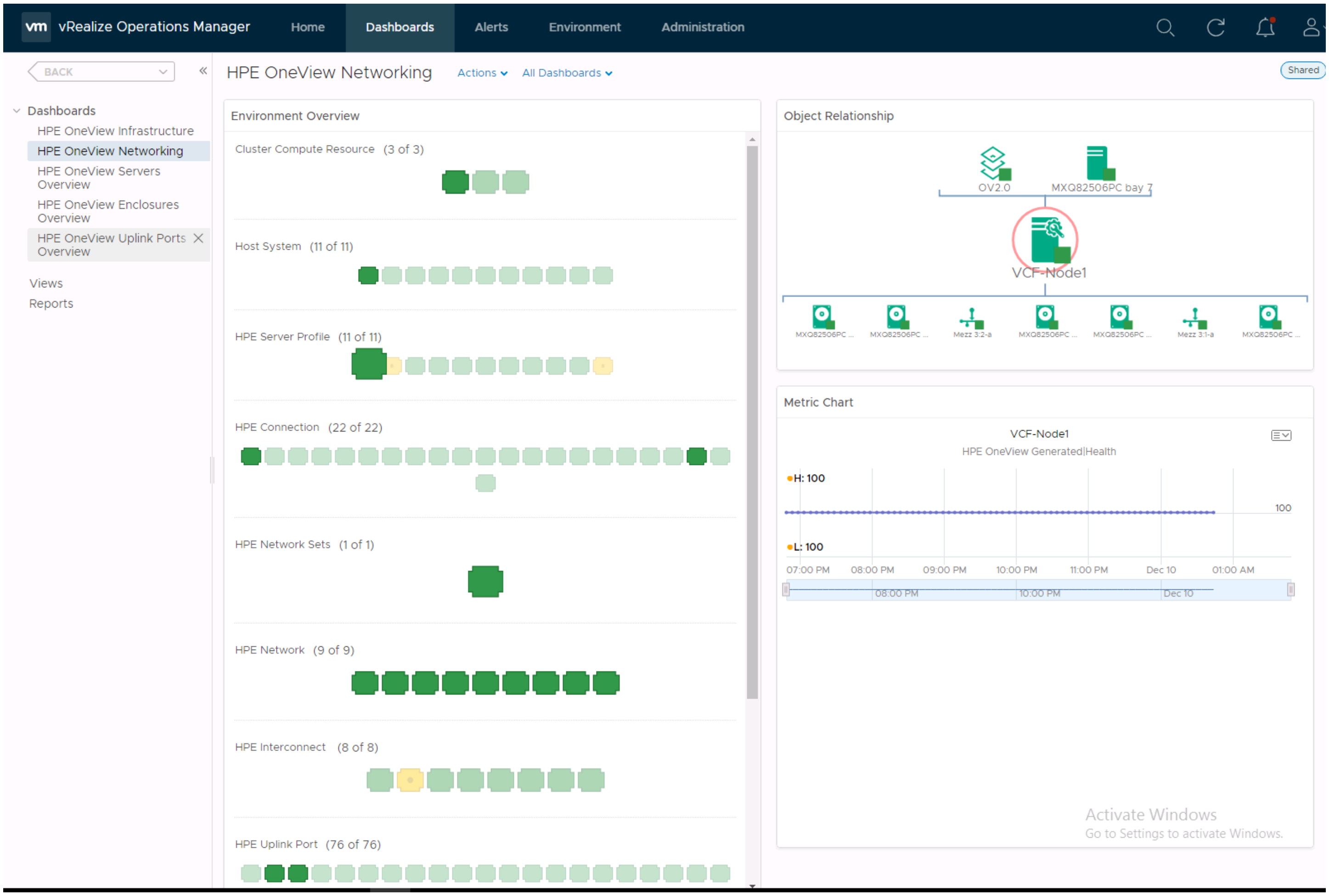
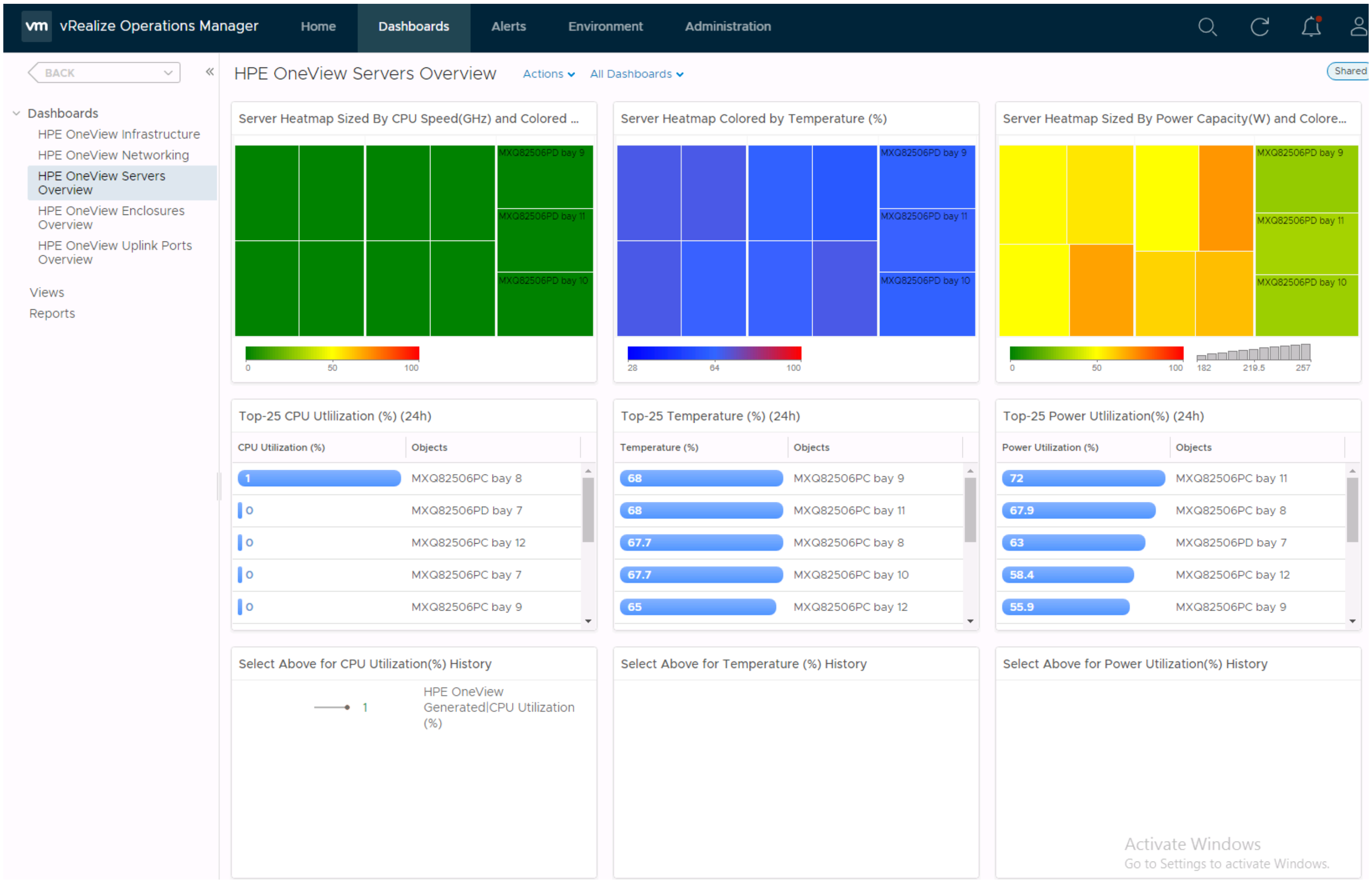


Figure 31. HPE OneView Networking dashboard

HPE OneView Servers Overview dashboard

The HPE OneView Servers Overview dashboard provides a summary and metrics information pertaining to CPU utilization, temperature, and power utilization of physical servers in the VCF environment. The Heatmaps allow you to quickly compare these metrics.



HPE OneView Servers Overview

Server Heatmap Sized By CPU Speed(GHz) and Colored ...

Server Heatmap Colored by Temperature (%)

Server Heatmap Sized By Power Capacity(W) and Colore...

Top-25 CPU Utlilization (%) (24h)

CPU Utilization (%)	Objects
1	MXQ82506PC bay 8
0	MXQ82506PD bay 7
0	MXQ82506PC bay 12
0	MXQ82506PC bay 7
0	MXQ82506PC bay 9

Top-25 Temperature (%) (24h)

Temperature (%)	Objects
68	MXQ82506PC bay 9
68	MXQ82506PC bay 11
67.7	MXQ82506PC bay 8
67.7	MXQ82506PC bay 10
65	MXQ82506PC bay 12

Top-25 Power Utlilization(%) (24h)

Power Utilization (%)	Objects
72	MXQ82506PC bay 11
67.9	MXQ82506PC bay 8
63	MXQ82506PD bay 7
58.4	MXQ82506PC bay 12
55.9	MXQ82506PC bay 9

Select Above for CPU Utilization(%) History

1 HPE OneView Generated|CPU Utilization (%)

Select Above for Temperature (%) History

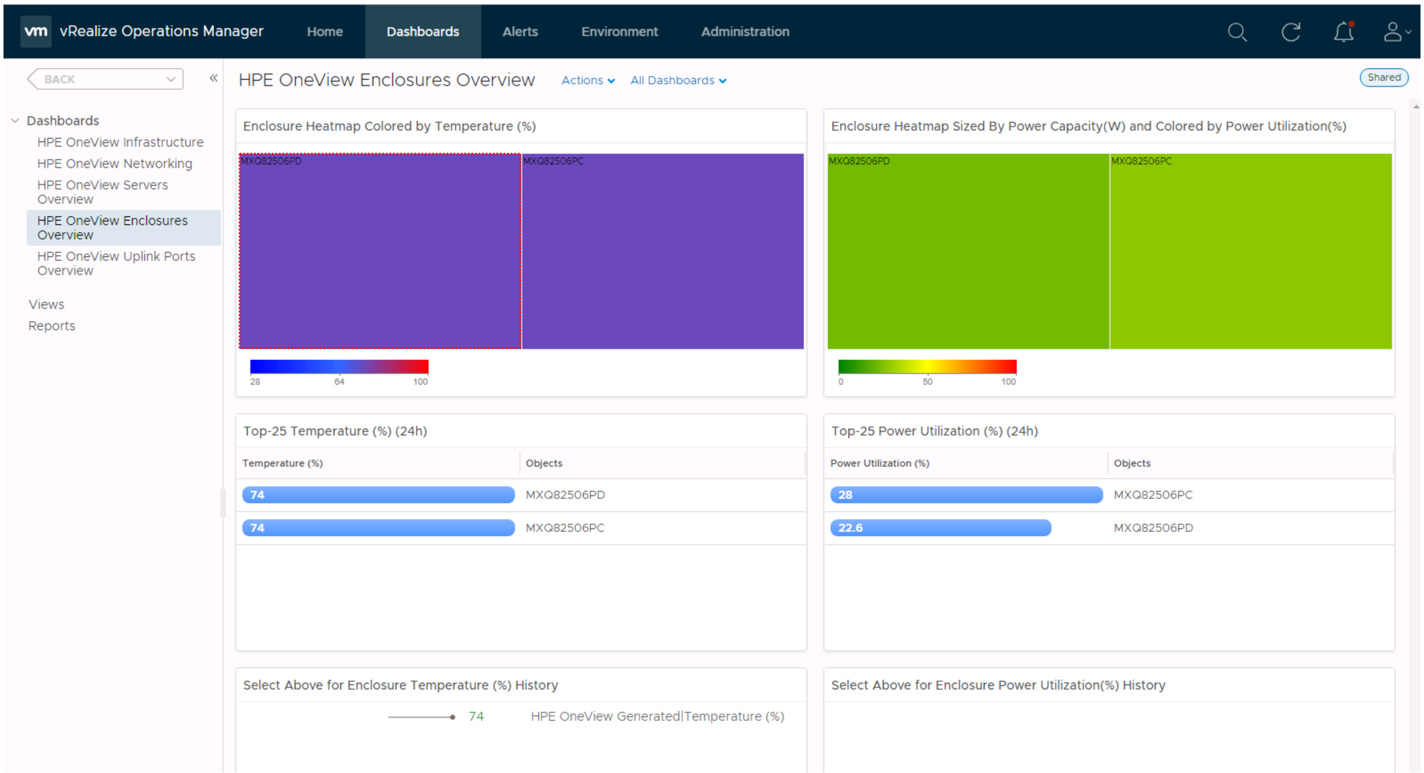
Select Above for Power Utilization(%) History

Figure 32. HPE OneView Servers dashboard

HPE OneView Enclosures Overview dashboard

The HPE OneView Enclosures Overview dashboard provides summary and metrics information pertaining to the temperature and power utilization of the HPE Synergy Enclosures. The Heatmaps allow you to quickly compare these metrics. Under each Heatmap, there is a comparative representation between the two enclosures as used for this Reference Architecture. Selecting an enclosure from this list generates a sparkline chart displaying the metric history.

Figure 33 shows HPE OneView Enclosure dashboard with the two enclosure as used in this Reference Architecture solution.


vm vRealize Operations Manager
Home
Dashboards
Alerts
Environment
Administration
BACK
Dashboards
HPE OneView Infrastructure
HPE OneView Networking
HPE OneView Servers Overview
HPE OneView Enclosures Overview
HPE OneView Uplink Ports Overview
Views
Reports
HPE OneView Enclosures Overview
Actions
All Dashboards
Shared
Enclosure Heatmap Colored by Temperature (%)
MXQ82506PD
MXQ82506PC
28
64
100
Enclosure Heatmap Sized By Power Capacity(W) and Colored by Power Utilization(%)
MXQ82506PD
MXQ82506PC
0
50
100

Top-25 Temperature (%) (24h)

Temperature (%)	Objects
74	MXQ82506PD
74	MXQ82506PC

Top-25 Power Utilization (%) (24h)

Power Utilization (%)	Objects
28	MXQ82506PC
22.6	MXQ82506PD

Select Above for Enclosure Temperature (%) History

74 HPE OneView Generated|Temperature (%)

Select Above for Enclosure Power Utilization(%) History

Figure 33. HPE OneView Enclosure Overview Dashboard

Summary

HPE and VMware can deliver a software-defined solution running on modular infrastructure across compute, storage, network, security, and cloud management. Trust this next-generation solution to run all your enterprise apps—both traditional and containerized—in cloud environments.

HPE Synergy offers a unique design for running VMware private clouds, providing the right IT platform that matches VMware Cloud Foundation characteristics—automated, software-driven, and flexible. HPE Synergy is the only modular infrastructure to run VMware Cloud Foundation deployments, and it provides a foundation for supporting hybrid configurations.

HPE Synergy:

- Reduces infrastructure complexity and cost.
- Eliminates top-of-rack switching and deploy.
- Has rack-scale fabric with HPE Virtual Connect.
- Efficiently scales fabrics across multiple frames.
- Provisions and manages the physical fluid resources for SDDC deployments through HPE OneView.

This Reference Architecture demonstrates a secured and scalable private cloud solutions built using VMware Cloud Foundation on HPE Synergy. It showcases the ability to:

- simplify deployment and operations of on-boarding traditional and container based workloads on VMware Cloud Foundation.
- ease the deployment of traditional multitier applications and containerized applications delivered through vRealize Automation.
- automate Arista end of row switch operations, by adding uplinks trunk group ports for bringing up VMware VCF deployment with Ansible Playbooks.
- simplify and automate expanding VMware Workload clusters and firmware updates using HPE OneView for VMware vRealize Orchestrator.
- proactively monitor for performance, health and capacity of a private cloud environment using HPE OneView For VMware vRealize Operations.

Appendix A: Bill of materials

Note
Part numbers are at time of publication and subject to change. The bill of materials does not include complete support options or complete rack and power requirements. For questions regarding ordering, consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html

Table 9. Bill of materials

Qty	Product	product
		Rack and power
1	P9K10A	HPE 42U 600mmx1200mm G2 Kitted Advanced Shock Rack with Side Panels and Baying
1	P9K10A 001	HP Factory Express Base Racking Service
1	804938-B21	HPE Synergy Frame Rack Rail Kit
1	H6J85A	HPE Rack Hardware Kit
1	BW932A	HPE 600mm Rack Stabilizer Kit
1	BW932A B01	HPE 600mm Rack include with Complete System Stabilizer Kit
		Synergy Frames
2	797740-B21	HPE Synergy 12000 Configure-to-order Frame with 1x Frame Link Module 10x Fans
2	779218-B21	HPE Synergy 20Gb Interconnect Link Module
2	794502-B23	HPE Virtual Connect SE 40Gb F8 Module for Synergy
1	798096-B21	HPE 6x 2650W Performance Hot Plug Titanium Plus FIO Power Supply Kit
1	798096-B21	HPE 6x 2650W Performance Hot Plug Titanium Plus FIO Power Supply Kit
2	804353-B21	HPE Synergy Composer
2	804942-B21	HPE Synergy Frame Link Module
1	804943-B21	HPE Synergy Frame 4x Lift Handles
1	859493-B21	Synergy Multi Frame Master1 FIO
4	804101-B21	HPE Synergy Interconnect Link 3m Active Optical Cable
2	720199-B21	HPE BladeSystem c-Class 40G QSFP+ to QSFP+ 3m Direct Attach Copper Cable
2	861412-B21	HPE Synergy Frame Link Module CAT6A 1.2m Cable
8	871940-B21	HPE Synergy 480 Gen10 Configure-to-order Compute Module

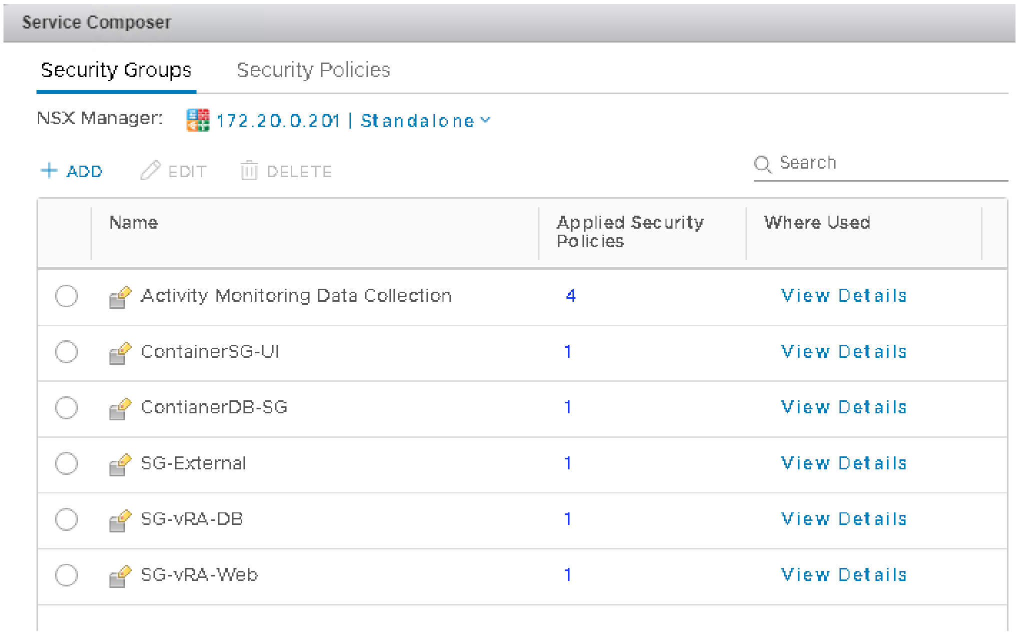
Qty	Product	product
8	872138-B21	HPE Synergy 480/660 Gen10 Intel Xeon-Gold 6142 (2.6GHz/16-core/150W) Processor Kit
8	872138-L21	HPE Synergy 480/660 Gen10 Intel Xeon-Gold 6142 (2.6GHz/16-core/150W) FIO Processor Kit
64	815100-B21	HPE 32GB (1x32GB) Dual Rank x4 DDR4-2666 CAS-19-19-19 Registered Smart Memory Kit
128	835955-B21	HPE 16GB (1x16GB) Dual Rank x8 DDR4-2666 CAS-19-19-19 Registered Smart Memory Kit
8	P01367-B21	HPE 96W Smart Storage Battery (up to 20 Devices) with 260mm Cable Kit
8	804424-B21	HPE Smart Array P204i-c SR Gen10 (4 Internal Lanes/1GB Cache) 12G SAS Modular Controller
8	741279-B21	HPE 8GB Dual microSD Flash USB Drive
8	777430-B21	HPE Synergy 3820C 10/20Gb Converged Network Adapter

		Storage
2	835386-B21	HPE Synergy D3940 12Gb SAS CTO Drive Enclosure with 40 SFF (2.5in) Drive Bays
20	872376-B21	HPE 800GB SAS 12G Mixed Use SFF (2.5in) SC 3yr Wty Digitally Signed Firmware SSD
60	875513-B21	HPE 1.92TB SATA 6G Read Intensive SFF (2.5in) SC 3yr Wty Digitally Signed Firmware SSD
2	757323-B21	HPE Synergy D3940 Redundant I/O Adapter

Appendix B: NSX Security Policies Configuration

The following steps describe the NSX Security Policies configuration details.

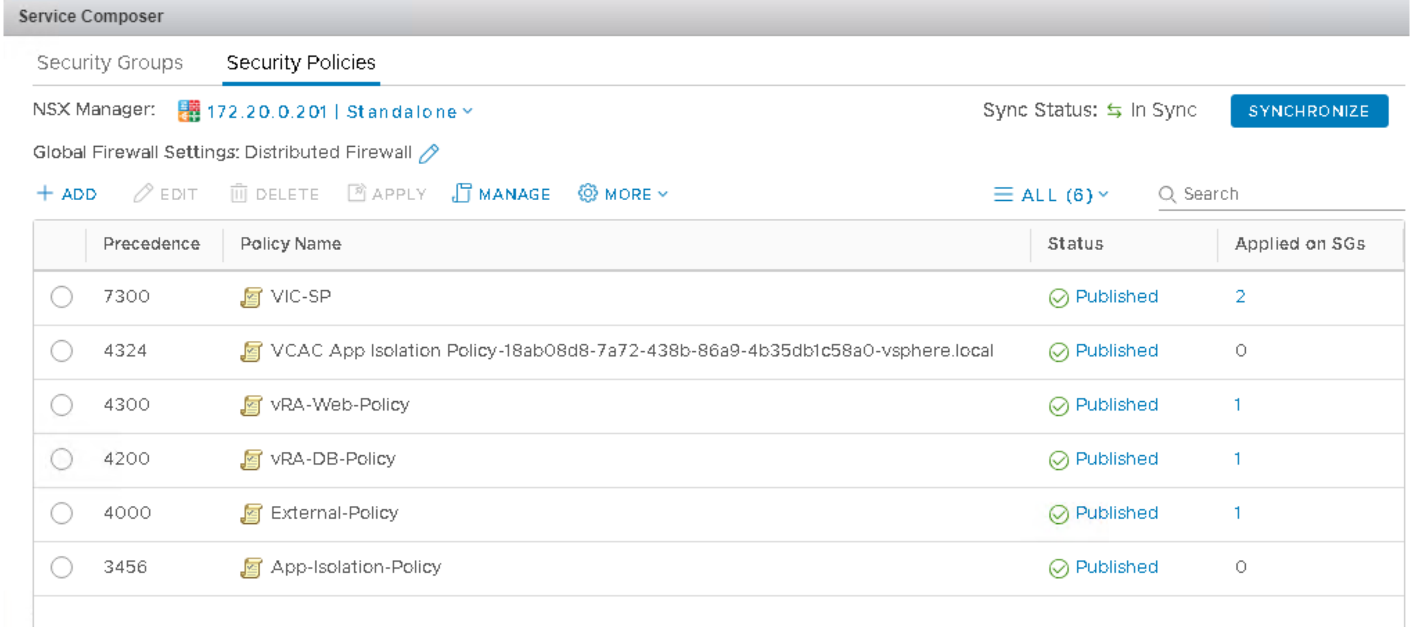
1. Define security groups in NSX service composer for WEB, APP and DB tier.



	Name	Applied Security Policies	Where Used
○	Activity Monitoring Data Collection	4	View Details
○	ContainerSG-UI	1	View Details
○	ContianerDB-SG	1	View Details
○	SG-External	1	View Details
○	SG-vRA-DB	1	View Details
○	SG-vRA-Web	1	View Details

Figure 34. Security groups for multiple tiers of application

2. Define a dynamic membership rule in the security groups for each tier in NSX. The rules are based on the machine prefix defined in vRA, for e.g. – For WEB tier define a rule that matches the VM names starting from 'web-'. This will add it to the security group dynamically while the VMs get deployed through vRA.



NSX Manager: 172.20.0.201 | Standalone

Sync Status: In Sync

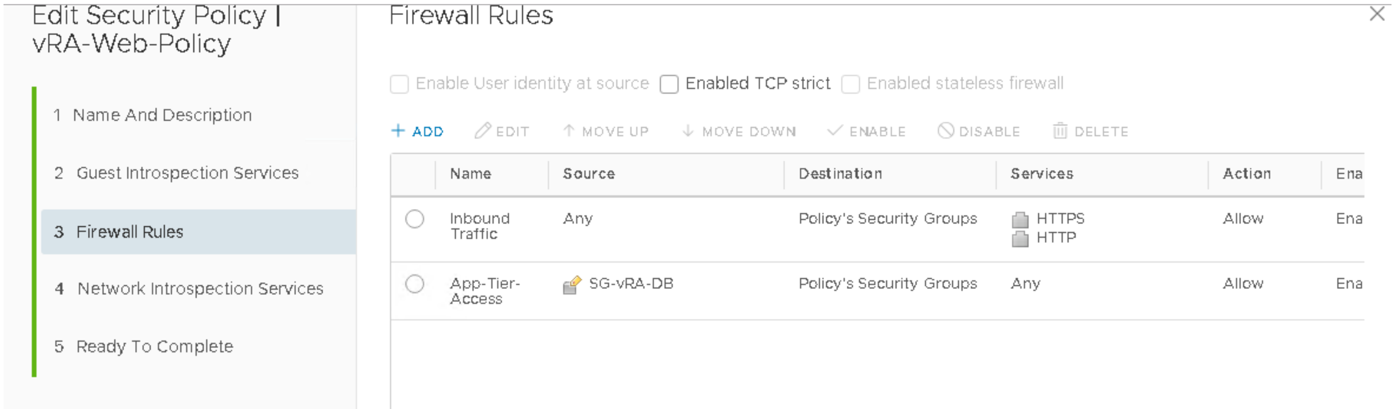
Global Firewall Settings: Distributed Firewall

	Precedence	Policy Name	Status	Applied on SGs
○	7300	VIC-SP	Published	2
○	4324	VCAC App Isolation Policy-18ab08d8-7a72-438b-86a9-4b35db1c58a0-vsphere.local	Published	0
○	4300	vRA-Web-Policy	Published	1
○	4200	vRA-DB-Policy	Published	1
○	4000	External-Policy	Published	1
○	3456	App-Isolation-Policy	Published	0

Figure 35. NSX security policies

3. Create security policies to define how traffic flows between specific tiers.

   a. Create the 'WEB-Tier-Policy' and apply it to 'WEB-TIER-SG' security group. This policy has a weight of 4300.

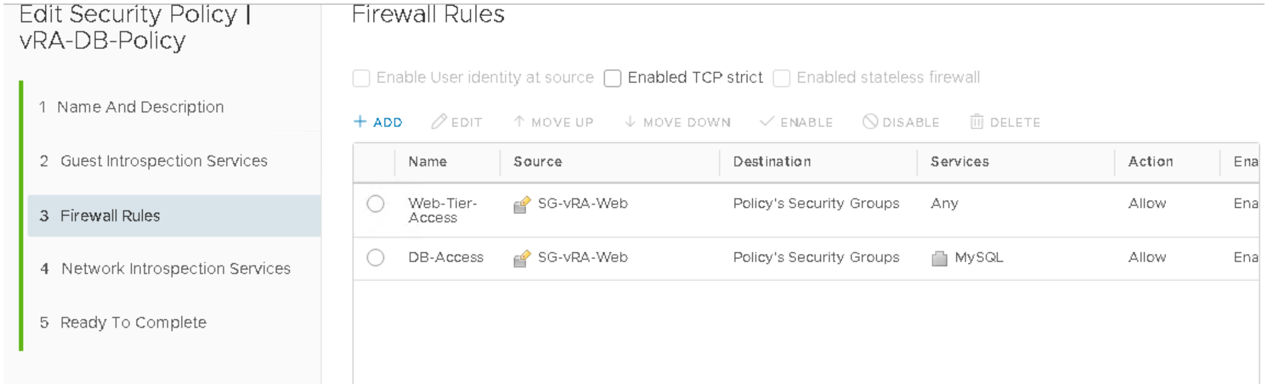


□ Enable User identity at source □ Enabled TCP strict □ Enabled stateless firewall

	Name	Source	Destination	Services	Action	Ena
○	Inbound Traffic	Any	Policy's Security Groups	HTTPS HTTP	Allow	Ena
○	App-Tier-Access	SG-vRA-DB	Policy's Security Groups	Any	Allow	Ena

Figure 36. Firewall rules in Web security policy

b. Create the ‘DB-Tier-Policy’ and apply it to the ‘DB-TIER-SG’ security group. This policy has a weight of 4100.



Edit Security Policy | vRA-DB-Policy

1 Name And Description
2 Guest Introspection Services
3 Firewall Rules
4 Network Introspection Services
5 Ready To Complete

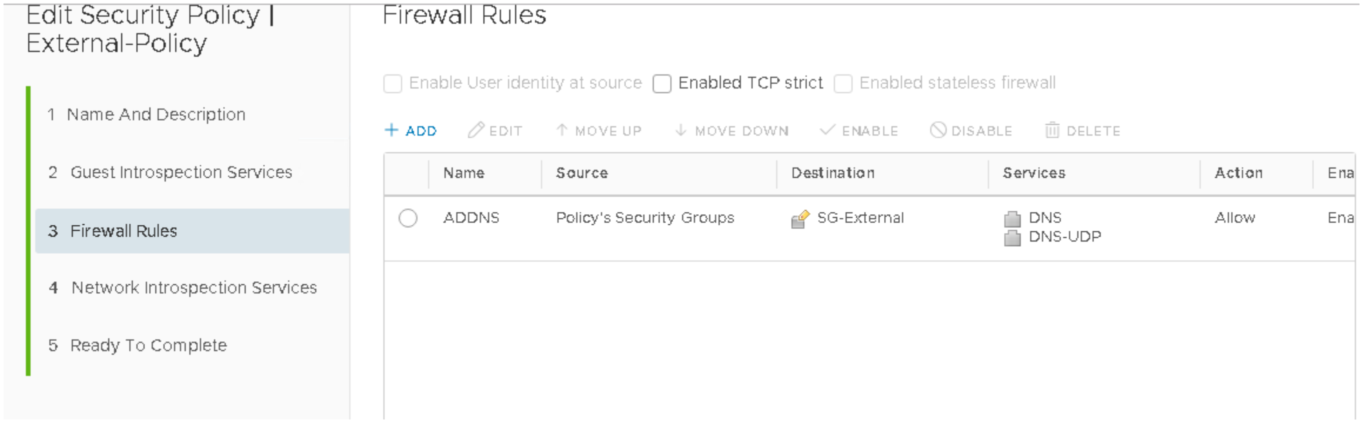
Firewall Rules

□ Enable User identity at source □ Enabled TCP strict □ Enabled stateless firewall

	Name	Source	Destination	Services	Action	Ena
○	Web-Tier-Access	SG-vRA-Web	Policy's Security Groups	Any	Allow	Ena
○	DB-Access	SG-vRA-Web	Policy's Security Groups	MySQL	Allow	Ena

Figure 37. Firewall rules in DB security policy

c. Create a default security policy that enables the VMs to talk to AD, DNS, NTP servers and apply it to ‘EXTERNAL-SG’ security group.



Edit Security Policy | External-Policy

1 Name And Description
2 Guest Introspection Services
3 Firewall Rules
4 Network Introspection Services
5 Ready To Complete

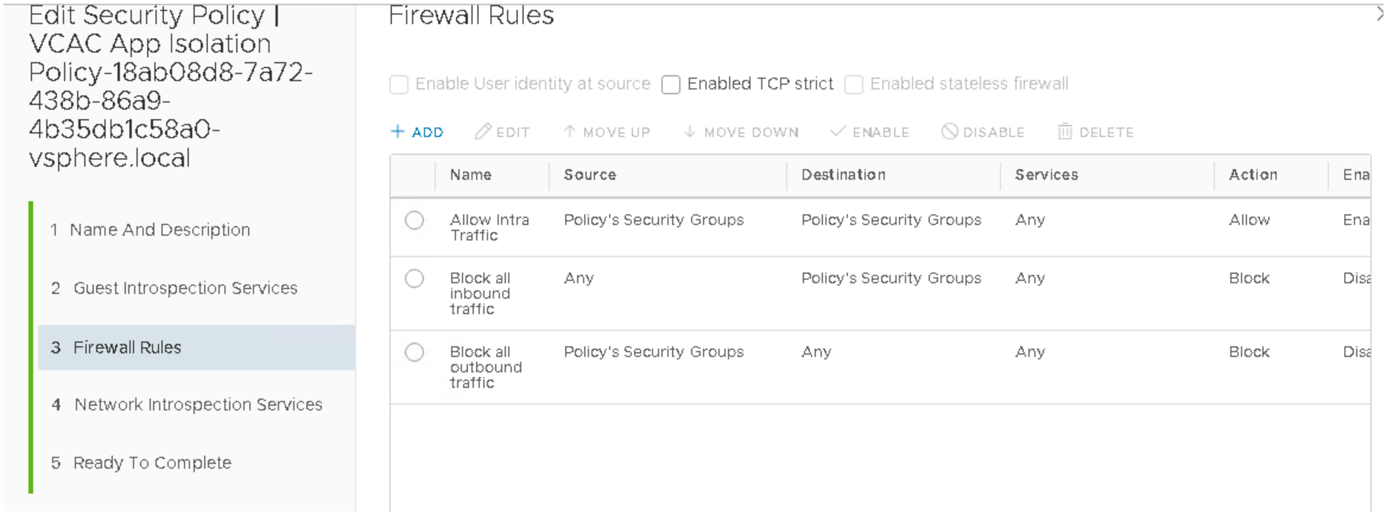
Firewall Rules

□ Enable User identity at source □ Enabled TCP strict □ Enabled stateless firewall

	Name	Source	Destination	Services	Action	Ena
○	ADDNS	Policy's Security Groups	SG-External	DNS DNS-UDP	Allow	Ena

Figure 38. Firewall rules for external components access

4. In vRA multi-machine blueprint, enable ‘App Isolation’. This will create a security policy automatically in NSX when the blueprint gets deployed. This policy will block all incoming and outgoing traffic. This policy has a weight of 3456.



Edit Security Policy | VCAC App Isolation Policy-18ab08d8-7a72-438b-86a9-4b35db1c58a0-vsphere.local

1 Name And Description
2 Guest Introspection Services
3 Firewall Rules
4 Network Introspection Services
5 Ready To Complete

Firewall Rules

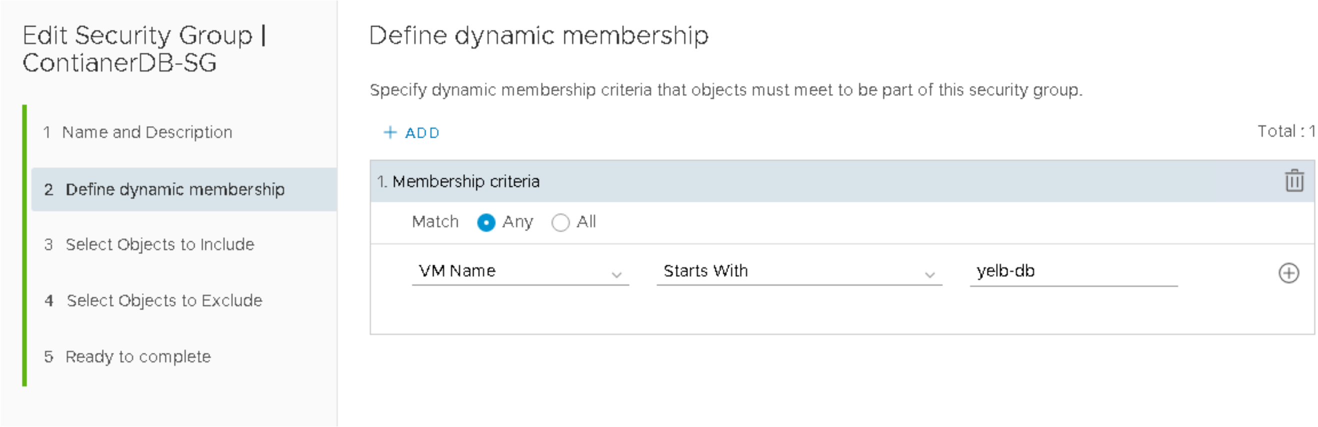
□ Enable User identity at source □ Enabled TCP strict □ Enabled stateless firewall

	Name	Source	Destination	Services	Action	Ena
○	Allow Intra Traffic	Policy's Security Groups	Policy's Security Groups	Any	Allow	Ena
○	Block all inbound traffic	Any	Policy's Security Groups	Any	Block	Disa
○	Block all outbound traffic	Policy's Security Groups	Any	Any	Block	Disa

Figure 39. Default firewall rules generated via vRA app isolation feature

The following steps describe configuring NSX Security Policies for Container workloads.

5. Define a dynamic membership rule in the security groups for each tier in NSX. The rules are based on the machine prefix defined in vRA, for e.g. – For yelb-DB define a rule that matches the Container VM names starting from ‘yelb-db’. This will add it to the security group dynamically while the container gets deployed through vRA.



Edit Security Group | ContianerDB-SG

1 Name and Description
2 Define dynamic membership
3 Select Objects to Include
4 Select Objects to Exclude
5 Ready to complete

Define dynamic membership

Specify dynamic membership criteria that objects must meet to be part of this security group.

Total : 1

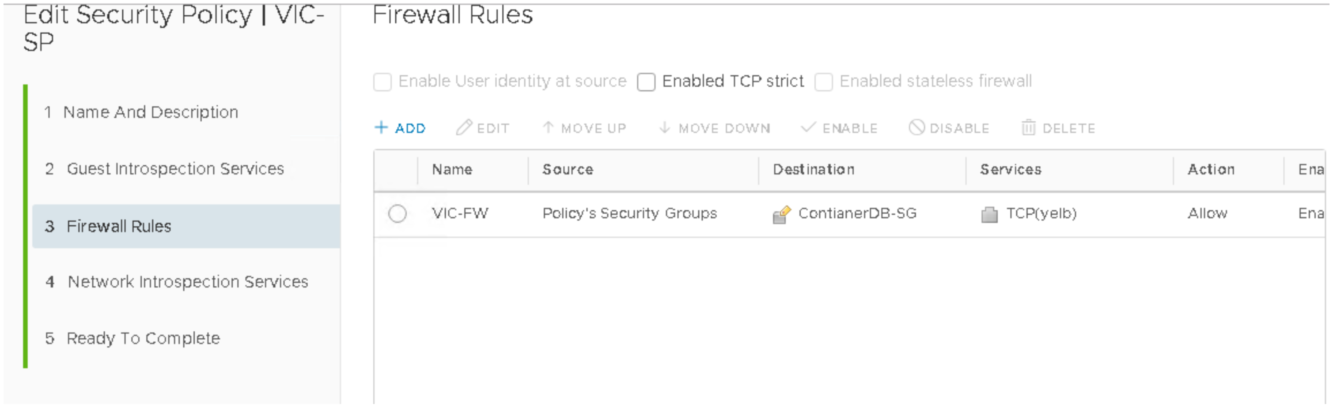
1. Membership criteria

Match ● Any ○ All

VM Name | Starts With | yelb-db

Figure 40. Dynamic membership criteria for yelb-db container VM

6. Create a security policy with a firewall rule that blocks TCP service over port 80. Apply this policy to Container DB-SG security group. This policy has a weight of 7300.



Edit Security Policy | VIC-SP

1 Name And Description
2 Guest Introspection Services
3 Firewall Rules
4 Network Introspection Services
5 Ready To Complete

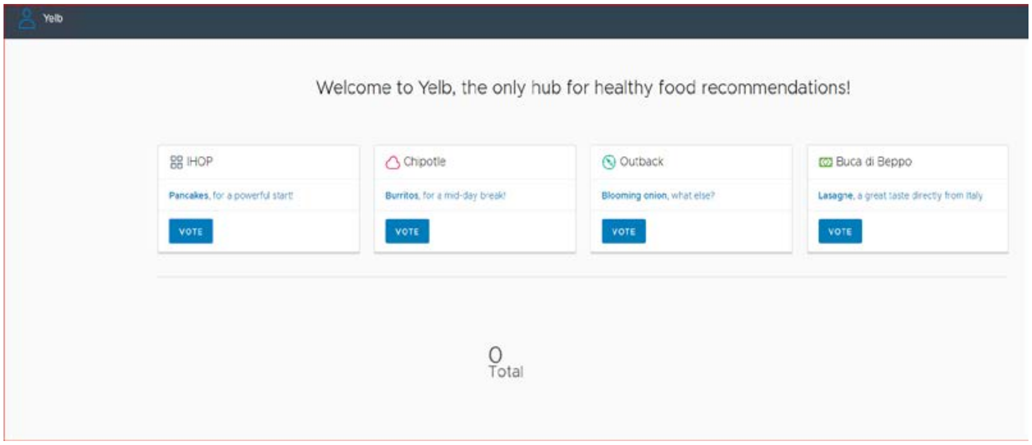
Firewall Rules

□ Enable User identity at source □ Enabled TCP strict □ Enabled stateless firewall

	Name	Source	Destination	Services	Action	Ena
○	VIC-FW	Policy's Security Groups	ContianerDB-SG	TCP(yelb)	Allow	Ena

Figure 41. Security policy for db container

7. Once security policy is applied and published successful it will block all the TCP traffic to yelb-db container due to which vote count display will become 0 or not visible as shown in the below figure.



Welcome to Yelb, the only hub for healthy food recommendations!

IHOP
Pancakes, for a powerful start!
VOTE

Chipotle
Burritos, for a mid-day break!
VOTE

Outback
Blooming onion, what else?
VOTE

Buca di Beppo
Lasagne, a great taste directly from Italy
VOTE

0
Total

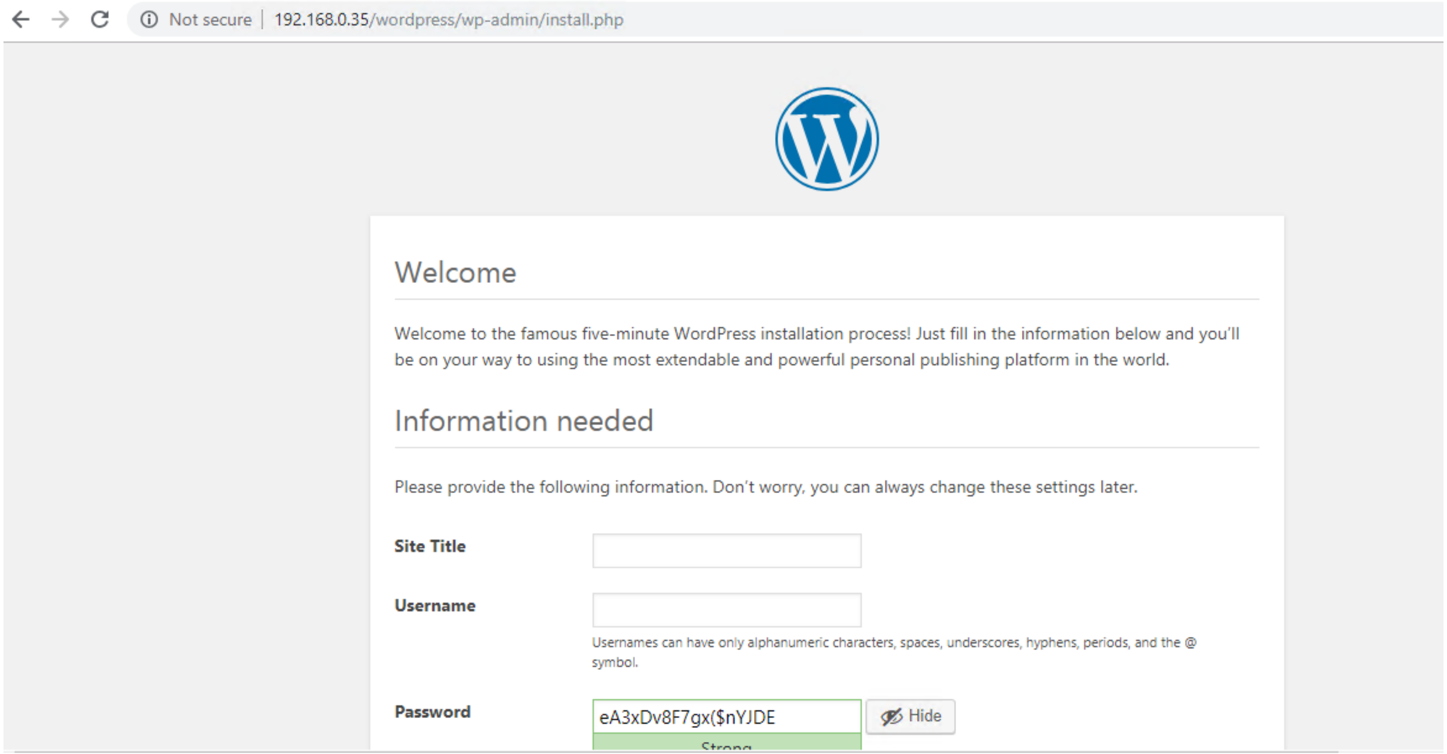
Figure 42. Application behavior after applying the security policy

8. If you want to enable the vote count, view as an administrator, then change the firewall rule action to 'Allow' in 'SecurityPolicy-VIC' as defined in Step 3.

Appendix C: vRA Blueprint Configuration

1. Create a vRA Blueprint with two vSphere machines and attach the Windows 2012 template from vSphere.
2. Define the minimum required number of VMs for a specific tier while configuring the multi-machine blueprint. Also, define the startup/shutdown order, for e.g. – database VM will start first, followed by application/web VM.
3. Attach the security groups and policies to the specific VMs.
4. Configure an On-demand load balancer in the blueprint canvas and attach it to the web tier VM.
5. Publish the blueprint as a Catalog Item in vRA.

6. After the blueprint gets deployed successfully, in vSphere, verify that the VMs are placed in appropriate security groups. Also, you should see an NSX edge gateway automatically deployed with Load balancer services enabled and pool members added to it.
7. Grab the VIP from load balancer and access the deployed application..



Not secure | 192.168.0.35/wordpress/wp-admin/install.php

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: eA3xDv8F7gx($nYJDE

Figure 43. WordPress application

Appendix D: Ansible Playbook

This section provides details about Ansible Playbook operations on Arista Switch for the following switch functionality.

1. Add vLAN trunk Group
2. Bringing up Interface
3. Shutdown Interface
4. Show Tech

Details of each operations is as follows:-

- Add vLAN trunk Group

Ansible Play

```
----------------------------------------------------------------------------------------------------------------------------------------------------

hosts: localhost

  connection: local

  vars_files:
    - "/home/vcf/playbooks_08052018/add_vlan_trunk_group.vars"

  tasks:

    - name: Add vlan

      eos_config:

        authorize: yes

        lines: 'vlan {{ vlan }}'

        save_when: always

        provider: '{{ eos_connection }}'

      register: eos_command_output


    - name: Add vlan to trunk group

      eos_config:

        authorize: yes

        lines: 'trunk group {{ trunk_group }}'

        parents: 'vlan {{ vlan }}'

        save_when: always

        provider: '{{ eos_connection }}'

      register: eos_command_output
```

Variable File

```
eos_connection:

  host: "192.168.47.77"

  authorize: yes

  username: "vcf"

  password: "vcf"

  transport: eapi

  validate_certs: no

vlan: 3000

trunk_group: VCF
```

- Bringing up Interface

Ansible Play

```
---------------------------------------------------------------------------------------------------------------------------------------------------------------------

- hosts: localhost

  connection: local

  vars_files:

    - "/home/vcf/playbooks_08052018/bring_up_ints.vars"

  tasks:

    - name: Show all interfaces

      eos_command:

        commands: 'show interfaces status '

        provider: '{{ eos_connection }}'

      register: eos_command_output

    - local_action: copy content={{ eos_command_output.stdout }} dest={{ playbook_dir }}/aux1.txt

    - shell: " sed 's/{/\\\n/g' {{ playbook_dir }}/aux1.txt > {{ playbook_dir }}/aux2.txt"

    - shell: " sed 's/}/\\\n/g' {{ playbook_dir }}/aux2.txt > {{ playbook_dir }}/aux1.txt"

    - shell: "awk -F\\\" '{if ((NF>6) && ($(NF-7) == \"linkStatus\") && ($(NF-5) == \"disabled\")) {print antes} else if ((NF > 1) && ($2 != \"vlanInformation\") && ($2 != \"interfaceMode\") && ($2 != \"vlanExplanation\")) {antes = $(NF - 1)}}' {{ playbook_dir }}/aux1.txt > {{ playbook_dir }}/aux2.txt"

    - shell: "cat {{ playbook_dir }}/aux2.txt"

      register: down_interfaces

    - name: Show filtered interfaces

      eos_command:

        commands: 'show interfaces {{ item }} description'

        provider: '{{ eos_connection }}'

      with_items: "{{ down_interfaces.stdout_lines }}"

    - name: Bring up interfaces

      eos_config:

        authorize: yes

        lines:

          - no shutdown

        parents: interface {{ item }}
        save_when: always
        provider: '{{ eos_connection }}'
      with_items: "{{ down_interfaces.stdout_lines }}"
    - shell: "rm -f {{ playbook_dir }}/aux1.txt"
    - shell: "rm -f {{ playbook_dir }}/aux2.txt"
```

Variable File

```
eos_connection:
  host: "192.168.47.77"
  authorize: yes
  username: "vcf"
  password: "vcf"
  transport: eapi
  validate_certs: no
```

- Shutdown Interface

Ansible Play

```
- hosts: localhost
  connection: local
  vars_files:
    - "/home/vcf/playbooks_08052018/shut_int.vars"
  tasks:
    - name: Shut down interface
      eos_config:
        authorize: yes
        lines:
          - shutdown
        parents: interface {{ interface }}
        save_when: always
        provider: '{{ eos_connection }}'
      register: eos_command_output
```

Variable File

```
------------------------------------------------------------------------------------------------------------------------------------------------------
eos_connection:
  host: "192.168.47.77"
  authorize: yes
  username: "vcf"
  password: "vcf"
  transport: eapi
  validate_certs: no
interface: Ethernet1
------------------------------------------------------------------------------------------------------------------------------------------------------
```

- Show Tech

Ansible Play

```
------------------------------------------------------------------------------------------------------------------------------------------------------
- hosts: localhost
  connection: local
  vars_files:
    - "/home/vcf/playbooks_08052018/show_tech.vars"
  tasks:
    - name: Show tech
      eos_command:
        commands: 'show tech-support | no-more'
        provider: '{{ eos_connection }}'
      register: eos_command_output
------------------------------------------------------------------------------------------------------------------------------------------------------
```

Variable File

```
------------------------------------------------------------------------------------------------------------------------------------------------------
eos_connection:
  host: "192.168.47.77"
  authorize: yes
  username: "vcf"
  password: "vcf"
  transport: eapi
  validate_certs: no
vlan: 3000
trunk_group: VCF
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Resources and additional links

HPE Reference Architecture, hpe.com/info/ra

HPE Synergy, hpe.com/info/synergy

HPE and VMware, hpe.com/partners/vmware

HPE Networking, hpe.com/networking

HPE Servers, hpe.com/servers

HPE Enterprise Information Library, hpe.com/info/convergedinfrastructure

HPE Technology Consulting Services, hpe.com/us/en/services/consulting.html

HPE OneView for VMware vCenter with operation Manager and Log Insight, hpe.com/hpeoneviewforvcenter

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

a00064661enw, February 2019