Architecture Design Document
VirtualDesignMaster - Season 2

Challenge 3

Prepared by: Daemon Behr

Date: 2014–07-28

Revision History

Date	Rev	Author	Comments	Reviewers
2014-07-28	R1	Daemon Behr	Challenge 3	

Design Subject Matter Experts

The following people provided key input into this design.

Name	Email Address	Role/Comments
Daemon Behr	daemonbehr@gmail.com	Infrastructure Architect

Contents

1. Purpose and Overview

1.1 Executive Summary

Space ship depots continue to come online, with launches to the Moon occurring daily. The moon bases have been stabilized, and humans are beginning to settle in.

Many island nations fared better than expected during the outbreak. Their isolation could be very valuable if we face a third round of infection before the earth has been evacuated.

We need to get them back on the grid as soon as possible. Japan, Madagascar, and Iceland are first on the list for building infrastructures. Local teams have managed to get some equipment, but all you'll have to start with is one repository and blank hardware. As we've learned while building the depots, travel is dangerous and difficult. You will need to create your infrastructure in a lab first, to ensure it will be able to be quickly deployed by a local team. Once the process has been deemed successful, we will establish a satellite link to the islands to get everything we need to the local repositories.

The infrastructure must be built from scratch using a VMware cluster including shared storage, HA, DRS and OpenStack must manage it.

Deployment of instances will be done by using the OpenStack Horizon dashboard only. OpenStack Ice House will be used with the overlay network Neutron.

The VM images will include one Linux and one Windows instance. In order to remove single points of failure due to the availability of human resources, a second hypervisor will be integrated. Two VM images will be hosted by this hypervisor, one Linux and one Windows.

Documentation of the topology and provide visual proof of working deployments using video are required.

1.2 Summary Analysis

The purpose of this design is to build a robust and open infrastructure with multiple hypervisors and an automation and management engine (Openstack). This infrastructure will need to be configured and tested beforehand and then deployed remotely. These Island sites are being used as repositories for the existing knowledge of humanity. They will not have high compute requirements, but will have large storage requirements.

1.3 Design Interpretation

There are few key constraints to the deployment of the infrastructure. The hardware that will be used cannot be preconfigured because it has to be deployed by local teams on the island nations. There are three zones, with datacenters located at the Universities of Tokyo, Reykjavik and Madagascar. Zone 3 (Tokyo) will have the Openstack management cluster for all the zones because of the greater availability of technical staff.

1.4 Intended Audience

This document is meant for the key stakeholders in the project as well as technical staff leads required for a successful deployment.

1.5 Requirements

Below are the requirements as defined by the scenario document as well as additional communication with judges and creators in clarification emails.

1.5.1 Availability

Availability can be defined as “the proportion of the operating time in which an entity meets its in-service functional and performance requirements in its intended environment”. The criticality of the environment requires 100% availability as a service level objective (SLO).

Availability can be understood by understanding the relationship of Maintainability and Reliability. The chance a system will fail is based on Reliability. How quickly it can be fixed is due to it’s Maintainability. The combination of those two provide us with:

MTBF – Mean Time Between Failures (Reliability)

MTTR – Mean Time To Repair (Maintainability)

Availability is equal to MTTR/MTBF over the period evaluated.

R001	**Production systems require a 99% availability SLO**

1.5.2 Maintainability

Maintainability is defined as “the ability of an entity to facilitate its diagnosis and repair”. This is a key factor in availability.

R002	**The infrastructure must be quickly diagnosed and easily repaired**
R003	**The infrastructure must be managed by Openstack**
R004	**The infrastructure must use multiple hypervisors**

1.5.3 Integrity

System integrity is defined as “when an information system performs its function in an unimpaired manner, free from deliberate or inadvertent manipulation of the system”. In this context, it means that adds / moves / changes are not done on production systems.

R005	**The infrastructure must have adequate protection to avoid any data loss**

1.5.4 Reliability

Reliability can be defined by having an absence of errors. Errors in a code base do occur, but there must be a method to ensure that they are tested, identified and resolved before they are put in production. This prevents the errors from affecting the overall application infrastructure.

In addition, infrastructure component configuration errors can cause reliability issues and faults. A method must be in place to ensure that all component configurations are correct.

R006	**A system must be in place to identify errors in real time**

1.5.5 Scalability

Although the scope of the scalability has not been defined, the ability for it to occur with minimal additional design required.

R007	**The system must be scalable**

1.6 Constraints

C001	**The infrastructure is not physically accessible by design staff**
	Designs need to be tested and validated before they are deployed
C002	**Knowledgeable technical staff must be available during the deployment period.**
	Without the required staff, the mission will be delayed.

C003	**Communications are limited between core facilities and island facilities**
	Internet connectivity is limited and unreliable when using wireless communication.
C004	**Automation must be used for initial deployment.**
	In order accommodate the needs of all the other requirements, automation and orchestration is required for deployment.

1.7 Risks

R001	**The virus spread may occur more quickly than anticipated.**
	Time to prep would be reduced and facility security would need to be increased.
R002	**Lack of resources may delay timelines**
	Delays will cost human lives.
R003	**Adequate power may not be available**
	This would impact the ability to scale
R004	**Staffing levels may not be adequate**
	This would put more strain on available staff and increase possibility of human errors when tired.
R005	**Staff may not have the skill to manage the environment**
	This may reduce the manageability of the environment.

1.8 Assumptions

A001	**The required staff will be available and have been trained to support the environment and work as a team.**
	The minimum required is indicated in D006
A002	**All required hardware for scalability is available**
	This should be in a facility nearby that is accessible by the technical staff and secure from zombies. All tools required for deployment are also available.
A003	**Adequate power is available**

	This includes clean power, multiple UPS battery banks and redundant generators equivalent to a tier3 data center.
A004	**All staff in Primary roles have adequate supporting supplies**
	This consists of a large stockpile of high-quality coffee. Lack of this may cost human lives.
A005	**All staff in Secondary roles have adequate protective equipment**
	This should consist of standard equipment: 12 gauge shotgun with 24 shell bandolier, flamethrower (2 per 8 person shift), concussion grenades (8), high power rifle (7.62mm round) with 5 x 30 round magazines. An adequate number of chainsaws Katanas should also be made available.
A006	**All equipment in this design is new and has been burn-in tested**
	A period of 1 week was used and all tests were passed without issue.
A007	**All component firmware is at the identical revision**
	This is for each component type.

2. Architecture Design

2.1 Design Decisions

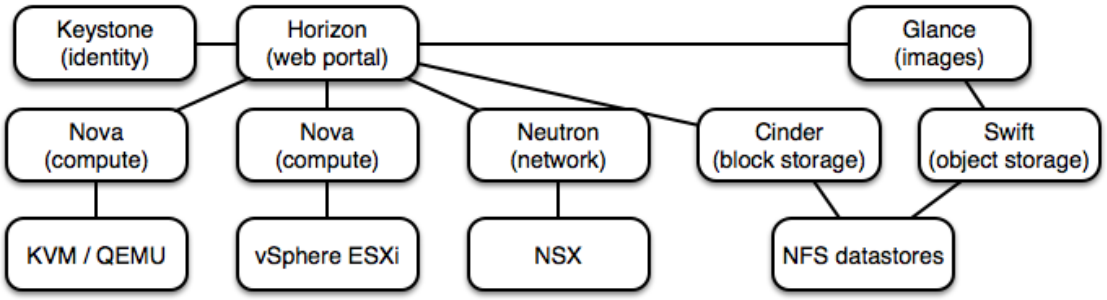
The architecture is described by a logical design, which is independent of hardware-specific details.

The following design decisions are presented with their justification.

D001	**The system will have the fewest number of logical components possible**
	Complexity needs to be reduced to allow for easier maintainability and quicker diagnosis of issues.
D002	**Automation will be done via Openstack. Heat will not be used**
	This relates back to D001. The fewest number of components possible .
D003	**There will be 3 availability zones**
	This is for Tokyo, Reykjavik and Madagascar. Each zone will have an identical copy of the infrastructure.
D004	**The facility will be fortified**

	Fortification will be done, where possible and where staffing permits.
D005	**Continuous Integration and Continuous Delivery will be implemented**
	Iterations of the infrastructure will evolve over time. Changes will need to be maintained and scheduled between all zones.
D003	**There will be three availability zones**
	This is to meet the requirements of R003 and R004
D004	**The facility will be fortified**
	There were two possible methods to meet R007. Fortify the facility, or distribute the infrastructure between several sites. Fortification was the decided on method because it reduces complexity and requires less staff and time.
D005	**Continuous Integration and Continuous Delivery will be implemented**
	Iteration of application code builds and infrastructure designs will occur hourly. This is not to say that changes will occur every hour, but it does mean that adds / moves / changes will occur at that time.

2.2 Conceptual Design



OpenStack APIs allow users to customize and configure down to the network level. VMware NSX is one of the most advanced and feature rich SDN solutions available today working seamlessly with OpenStack, ESXi and KVM.

This solution provides an end-to-end architecture with Cisco, Canonical Ubuntu, VMware vSphere and OpenStack technologies including Cinder and Swift for storage. Along with NetApp, these demonstrate high availability and redundancy along with ease of deployment and use.

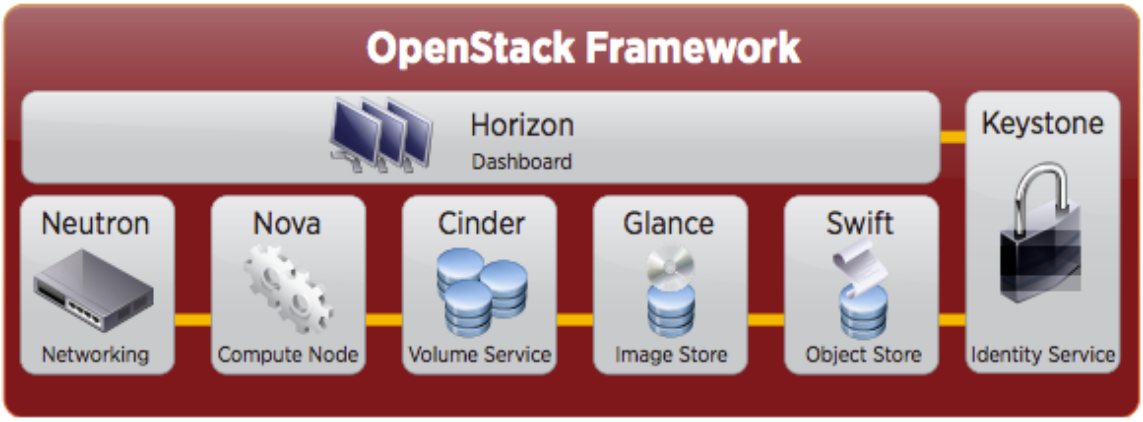
The following are the components used for the design and deployment:

• Cisco Unified Compute System (UCS) 2.2(1c)
• Cisco B-series Unified Computing System servers for compute and storage needs
• Cisco UCS VIC adapters
• Cisco Nexus 5000 series switches
• Canonical Ubuntu 12.04 LTS
• VMware vSphere 5.5
• OpenStack IceHouse architecture

The solution is designed to host scalable, mixed application workloads. The scope of this design is limited to the infrastructure pieces of the solution; the design does not address the vast area of OpenStack components and multiple configuration choices available there.

The components being used are listed below:

• Keystone – Identity service
• Horizon – Web GUI
• Nova – Compute service
• Glance – Image service
• Neutron – Network services (formerly called Quantum)
• Cinder – Block storage service
• Swift – Object storage service



The Cisco Unified Computing System is a next-generation data center platform that unites computing, network, storage access, and virtualization into a single cohesive system.

The main components of the Cisco UCS are:

• Computing—Blade servers based on Intel Xeon E5-2600 V2 Series Processors.
• Network—The system is integrated onto a low-latency, lossless, 10-Gbps network fabric.

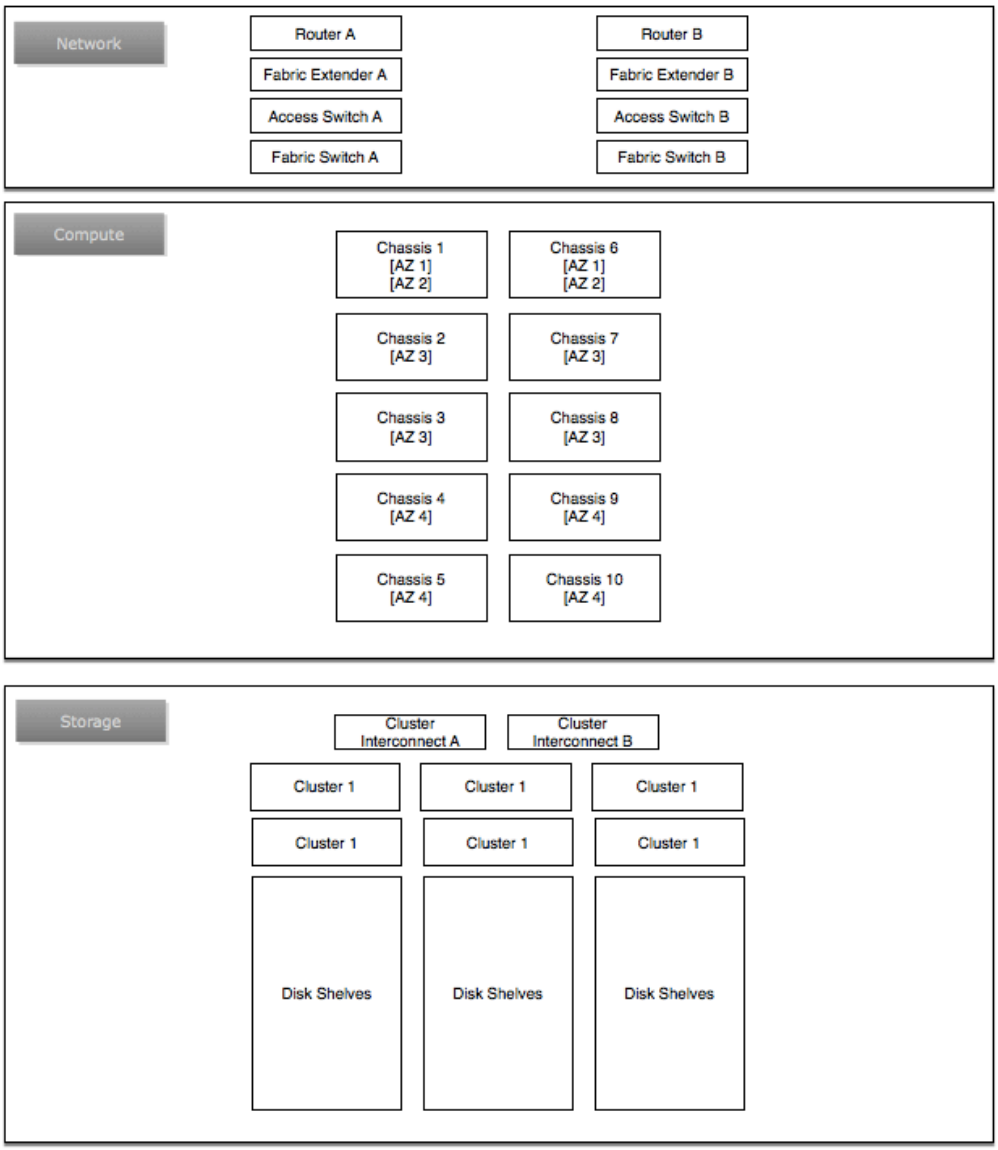
The NetApp filers are high performance storage devices that are able to scale up and scale out. This deployment will make use of Clustered Data OnTap 8.2.1 on FAS3250 units.

2.3 Logical Design

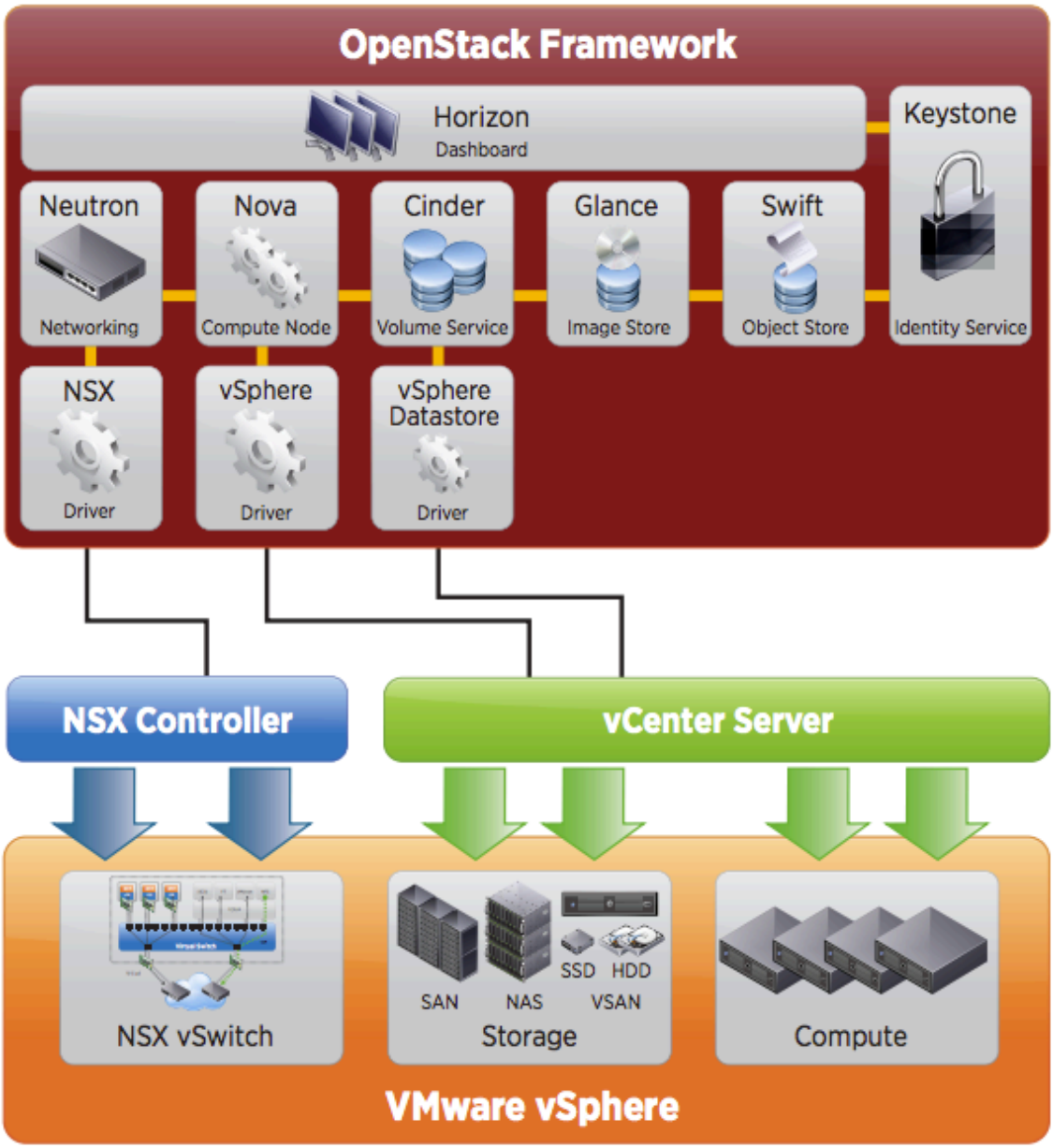
The Logical Design provides a more detailed view of the Conceptual Design components to meet the requirements. The architecture building blocks are defined without the mapping of specific technologies.

Below is the logical design for the 3 regions. Each region will have an identical configuration of hardware. There are 4 availability zones:

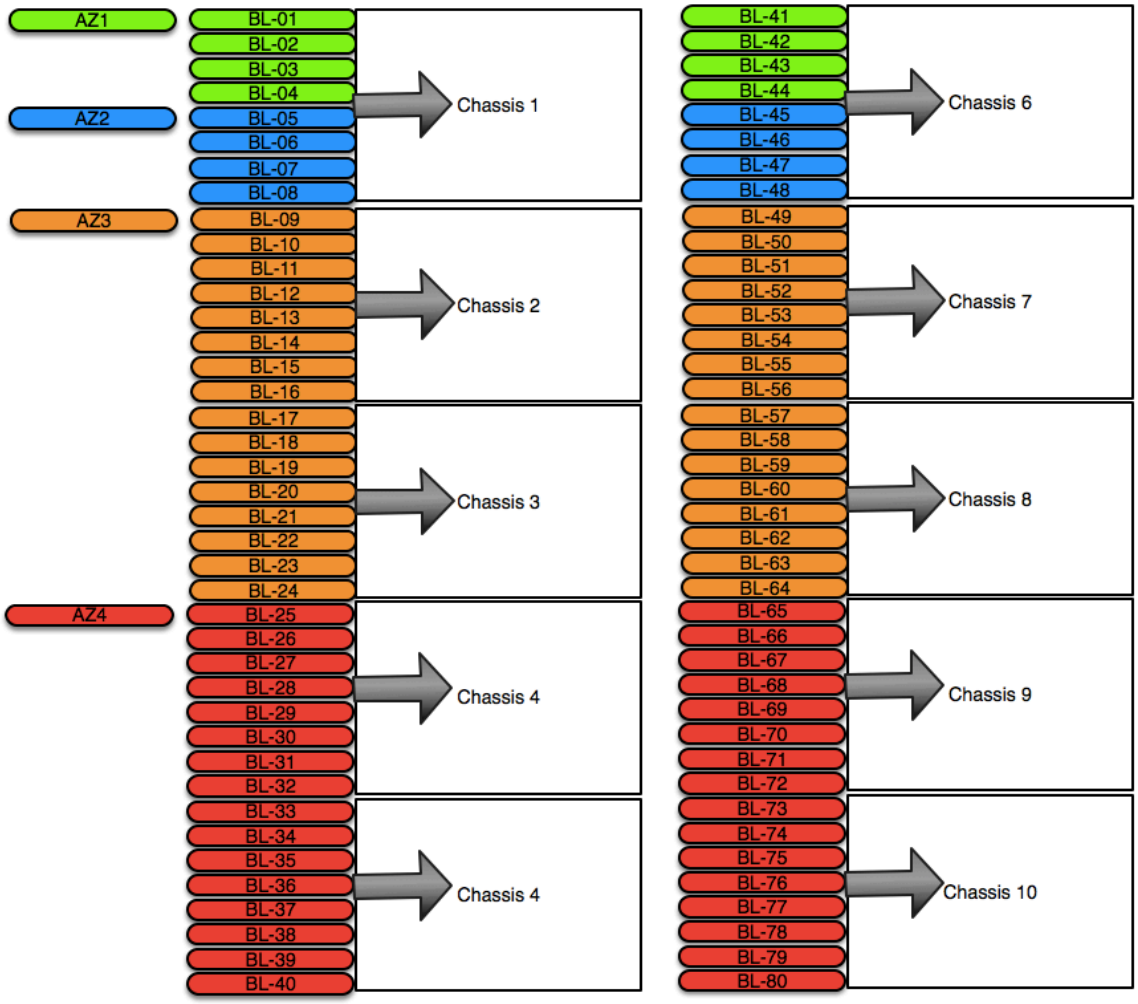
AZ1 – vSphere MGMT
AZ2 – Openstack MGMT
AZ3 – vSphere Compute
AZ4 – KVM Compute.



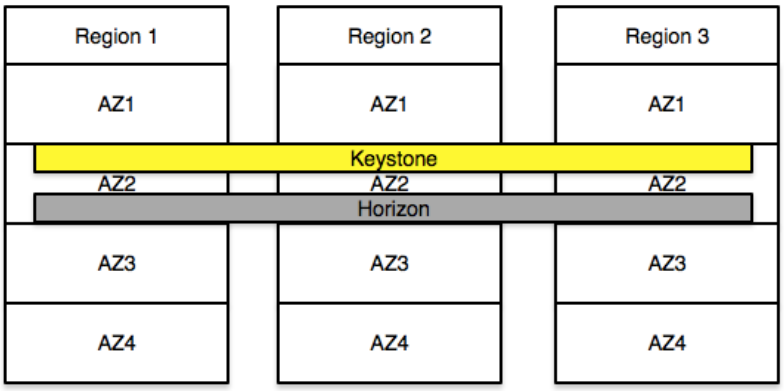
2.3.1 Openstack to vSphere Logical Design



2.3.2 Availability Zone to Blade Logical Design



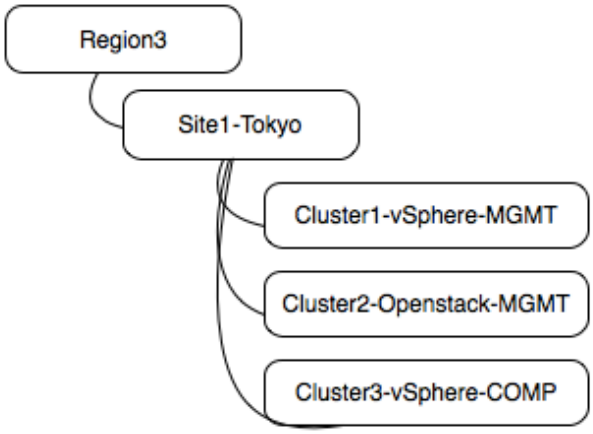
AZ1 – vSphere Management (vSphere cluster 1)
AZ2 – Openstack Management (vSphere cluster 2)
AZ3 – vSphere Compute Cluster (vSphere cluster 3)
AZ4 – KVM Compute Cluster (Baremetal Ubuntu hosts with KVM)



The active Keystone and Horizon instances will be hosted in AZ2, Region 3 (Tokyo). They will manage the other 2 regions centrally. Since all Openstack management components are running as VMware virtual machines, they can be replicated from site to site using NetApp snapmirror.

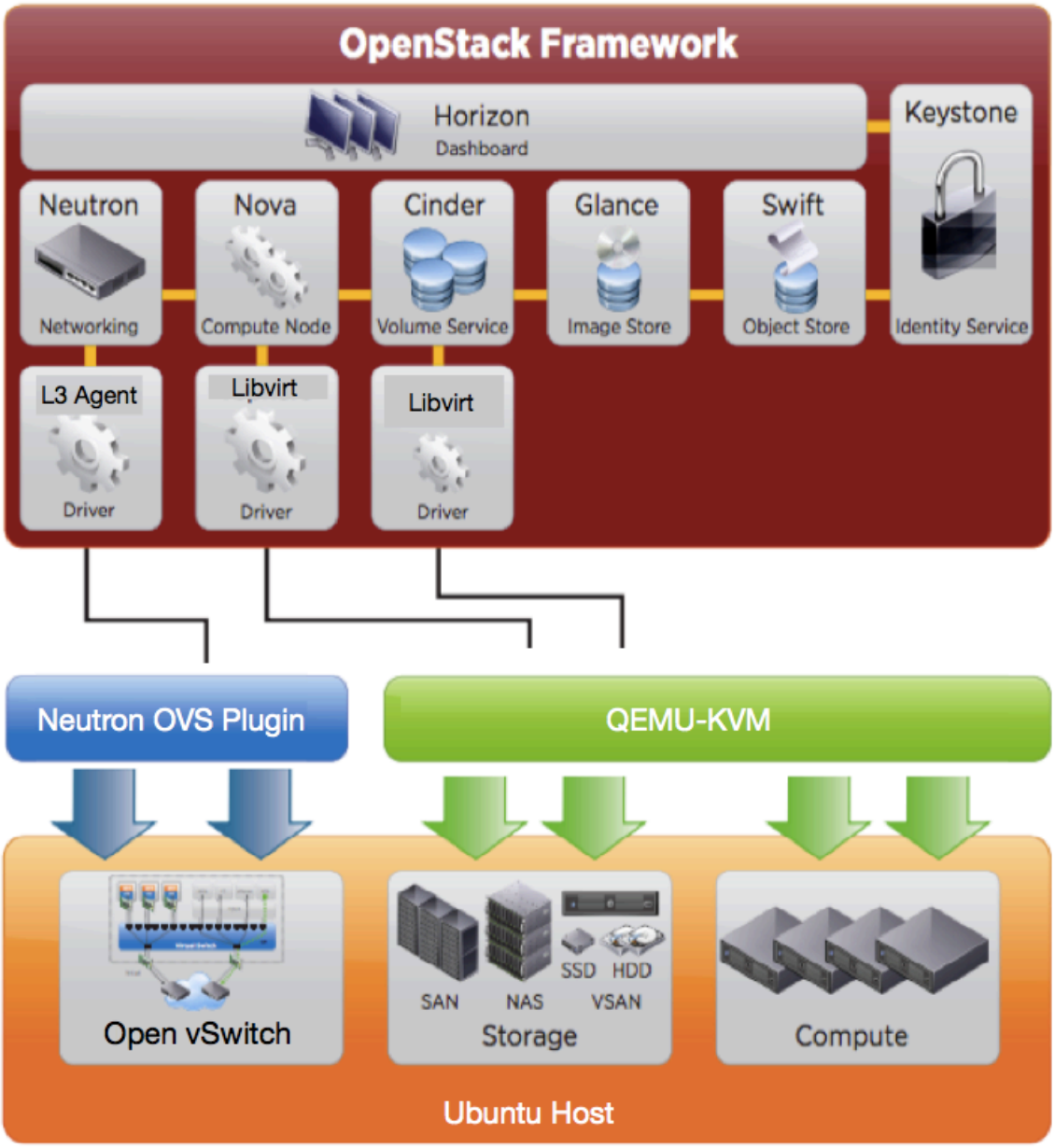
Snapmirror will replicate images located in Glance, and created volumes. All other Openstack management components will only run locally. In the case of a DR situation, the VMs just need to be registered and started in vSphere at the other sites.

2.3.3 vSphere cluster Logical Design

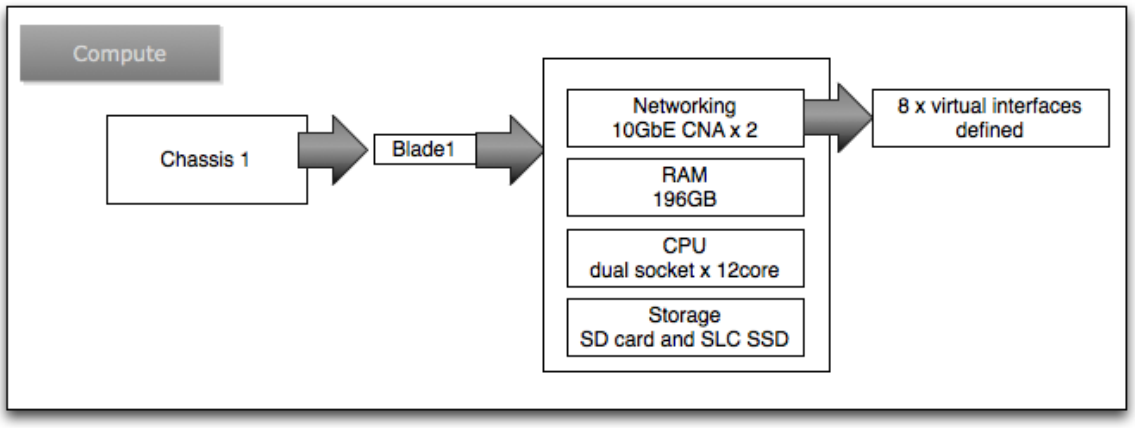


All management VMs will run in HA/DRS clusters within vSphere. Management clusters are physically separated from compute clusters. Each region will have one vCenter server.

2.3.4 Openstack to KVM Logical Design

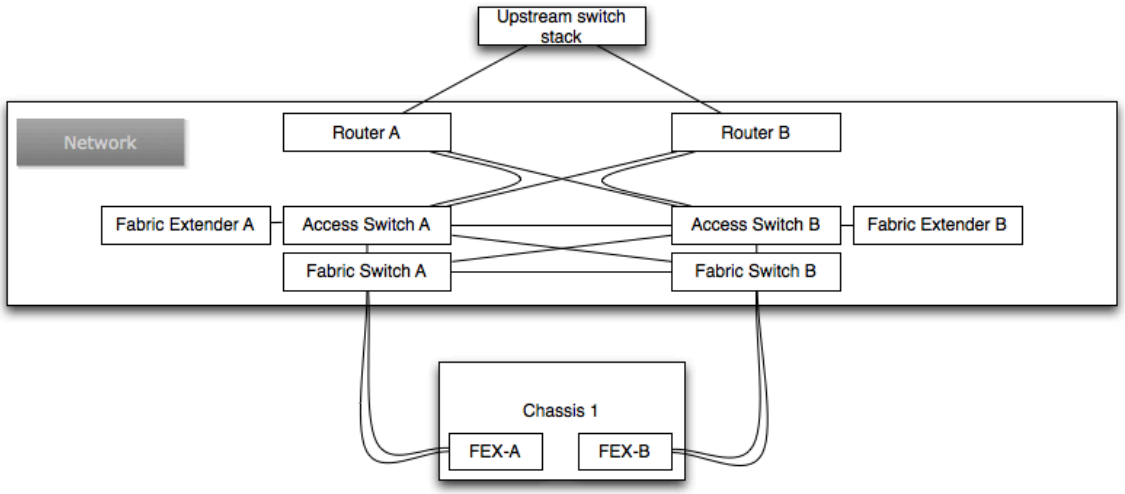


Each region will be provisioned with a 2 cabinet pod of blade chassis and 3 cabinets of storage. Each compute cabinet will contain 5 chassis. Each chassis will have 8 blade servers. In vSphere there will be 2 x 8 host clusters for management and 2 x 32 host clusters for compute. The clusters will be spanned across chassis and cabinets. As seen in the diagram below, this allows for no more than 50% cluster resource failure from a chassis malfunction on the smaller clusters, or 25% on large cluster. A single host failure is 12.5% for the 8 host cluster and 3% for a 32 host cluster.



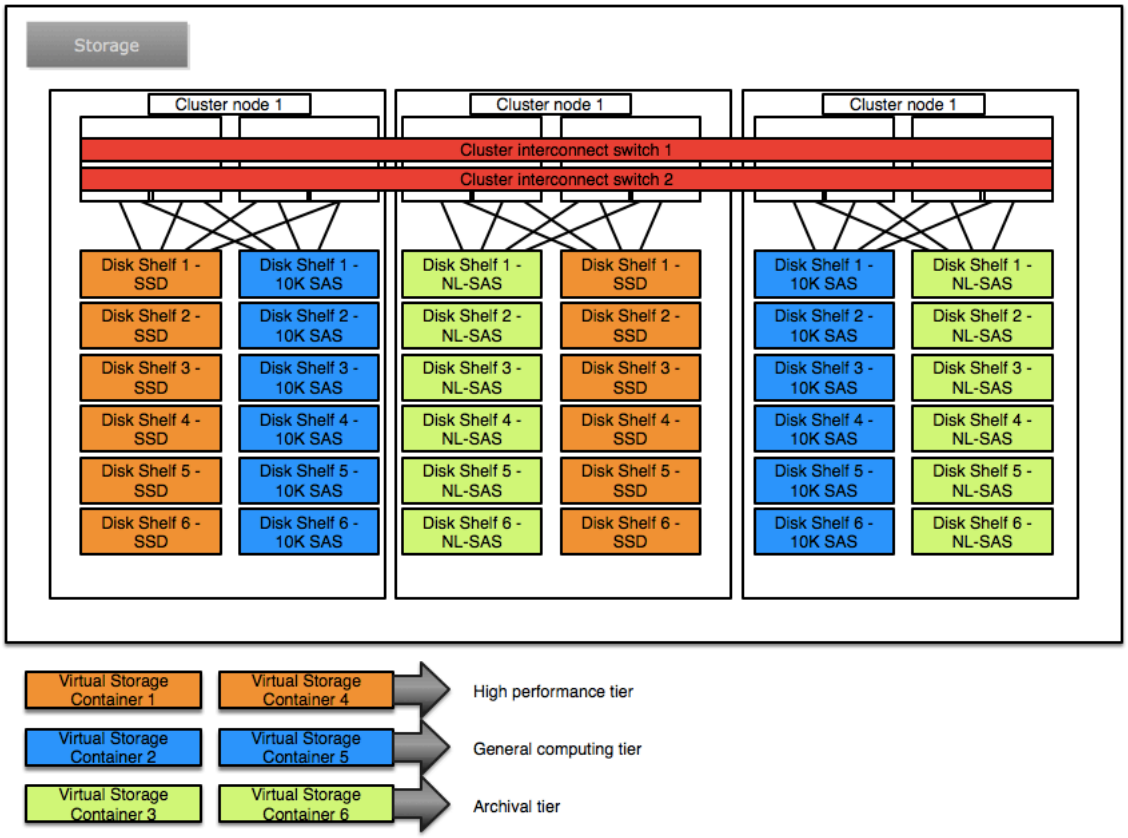
The chassis will uplink via fabric extenders to the fabric switches. The fabric switches will work in HA and can support a maximum of 20 chassis per pod. Each chassis will 2 fabric extenders, each having 2 x 10GbE uplinks to it's associated switch. The blades will use an onboard 10GbE adapter as well as a 10GbE Mezzanine card. The aggregate bandwidth of each chassis is 20Gbps per fabric. The aggregate bandwidth of the blade servers within the chassis is 80GbE per fabric. Therefore the over-subscription ratio is 4/1.

The network is converged, carrying both storage and data traffic within the same physical devices. Connectivity from the chassis FEXs will be aggregated in a port channel with the fabric switches. The fabric switches will aggregate bandwidth via a virtual port channel across both access switches. The access switches have a LAG group between then to accommodate interswitch traffic. The fabric extenders connected directly to the access switches provide 1Gbs connectivity for any users that need direct connectivity. Otherwise user access will be routed through the upstream switch stack.



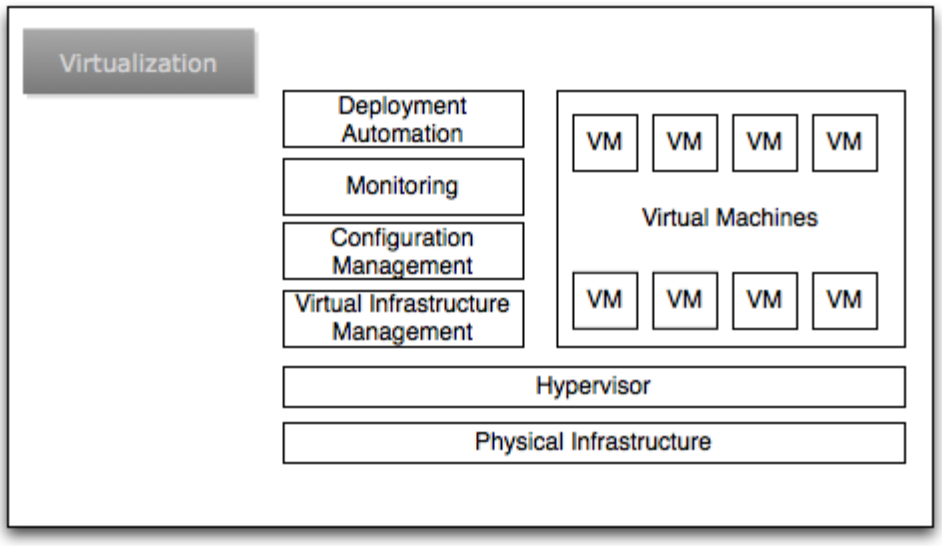
Storage will be comprised of clustered nodes in a scale out model. Each node will have active / active controllers. Nodes will be responsible for their own disks, but they can take over the disks of another node in the event of a failure.

Virtual storage containers are spanned across the aggregates of “like” performance tiers in each node cluster. This allows for expandability and performance. Volumes will be defined within the containers.



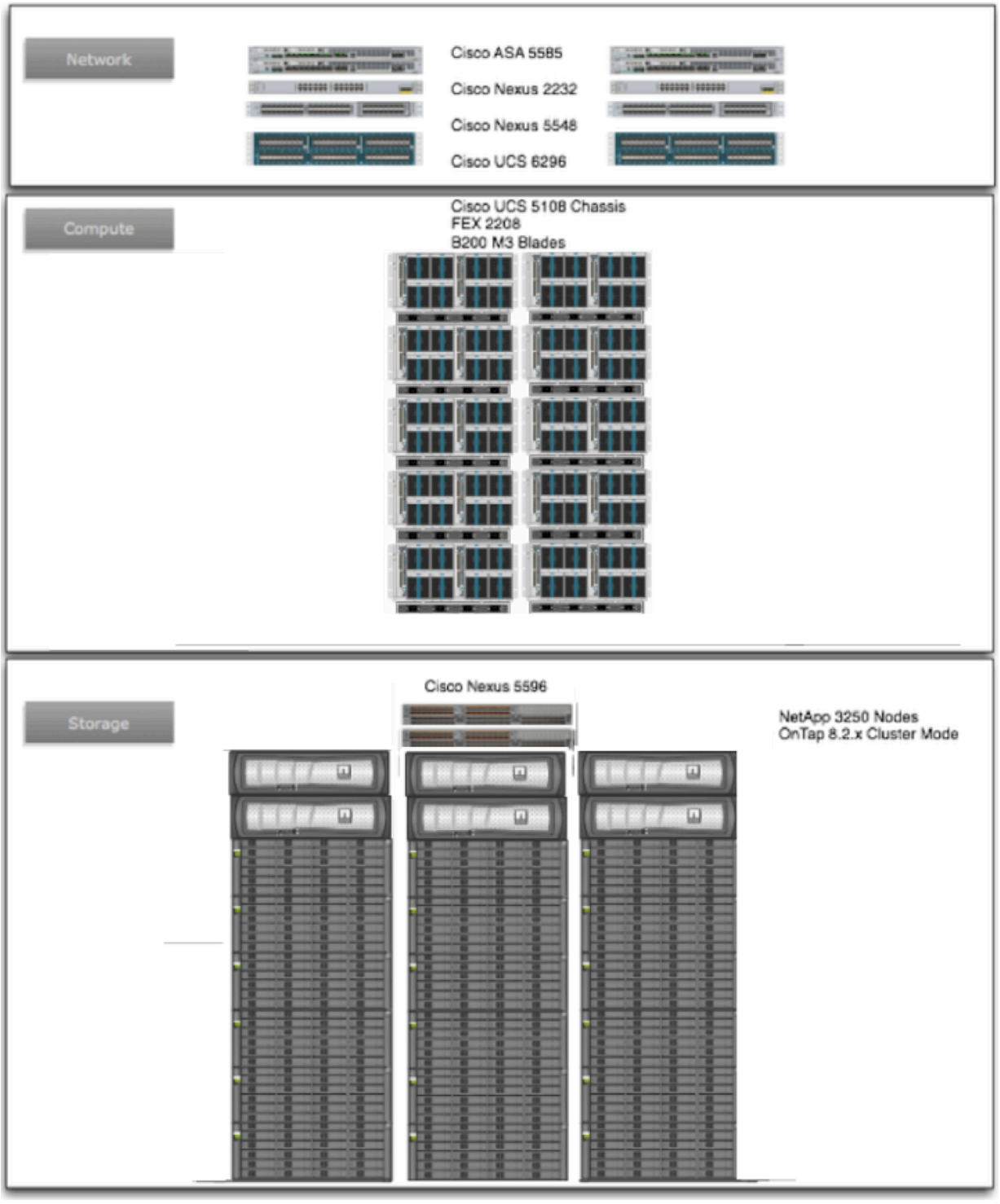
An additional storage tier exists outside of the clustered SAN environment. That is server side caching by making use of internal SLC SSDs within the blades.

The vSphere virtualization layer consists of the following components; a hypervisor, an infrastructure management system, configuration management, deployment automation and monitoring.

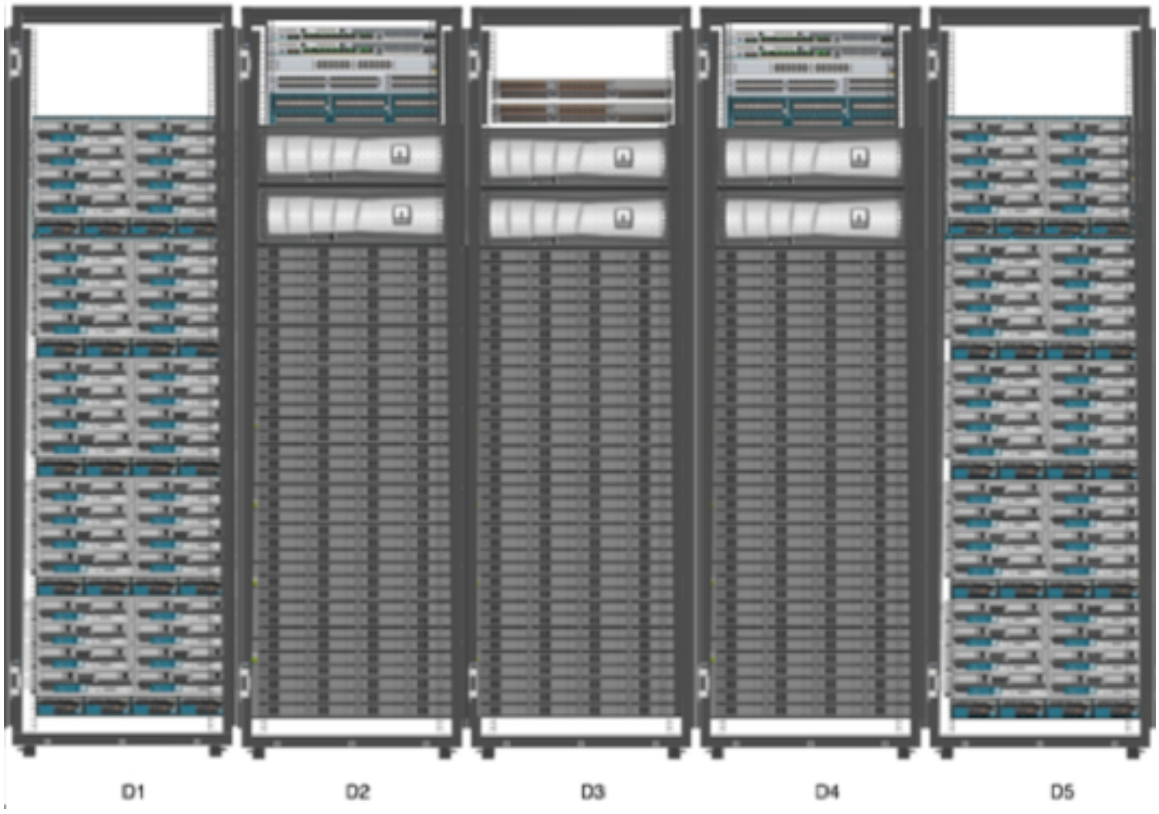


2.4 Physical Design

The physical design will be examined from first a datacenter row, then cabinet, then the technology area. The logical design for the divisional pod will be mapped to a row.


Network
Cisco ASA 5585
Cisco Nexus 2232
Cisco Nexus 5548
Cisco UCS 6296
Compute
Cisco UCS 5108 Chassis
FEX 2208
B200 M3 Blades
Storage
Cisco Nexus 5596
NetApp 3250 Nodes
OnTap 8.2.x Cluster Mode

2.4.1 Physical rack layout of region pod



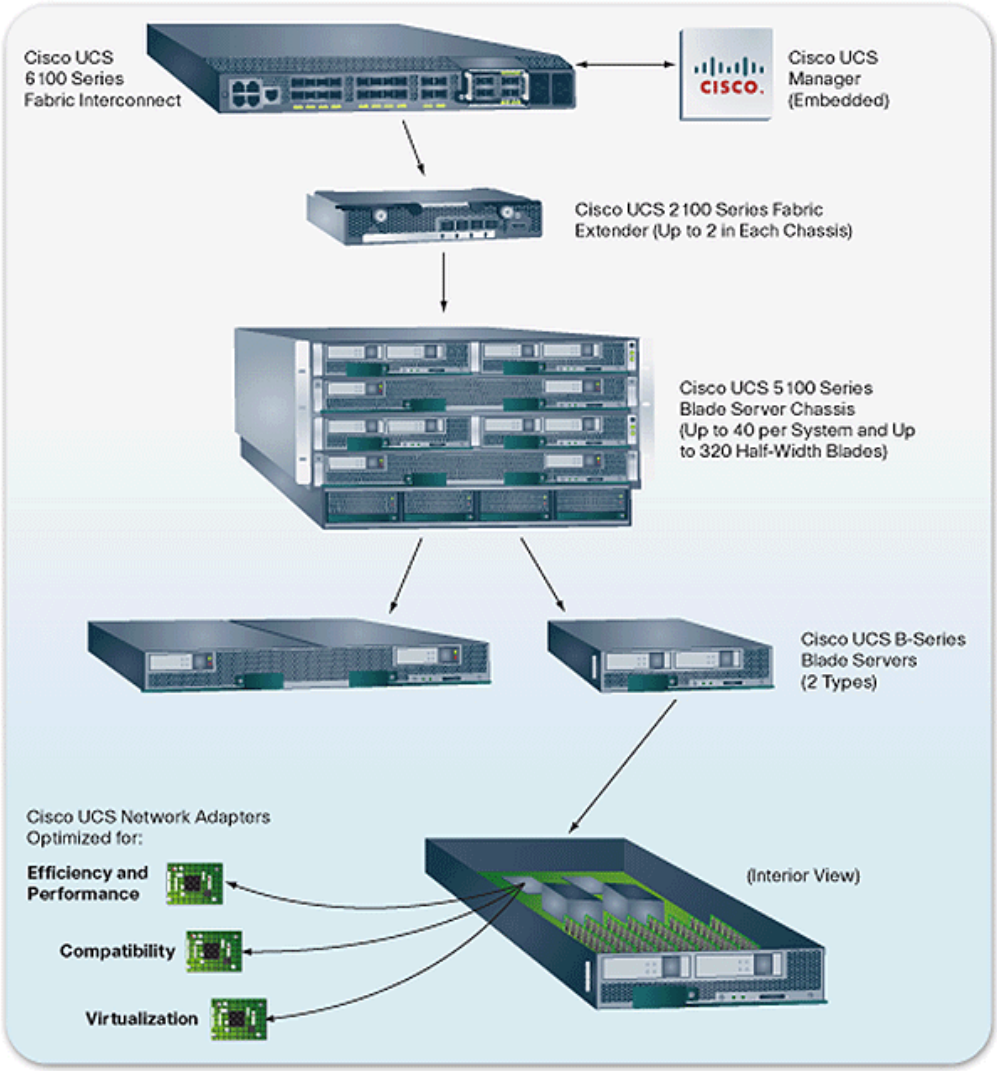
Above is the rack diagram for region3 site 1

The Cisco UCS infrastructure was chosen for the compute and network environment because of the ease of management, deployment and scalability.

NetApp was chosen for the shared storage because of the resiliency and scalability when using Clustered OnTap.

The server side caching will be done by PernixData FVP, making use of the SLC SSDs in the blades.

2.4.2 Cisco UCS Architecture

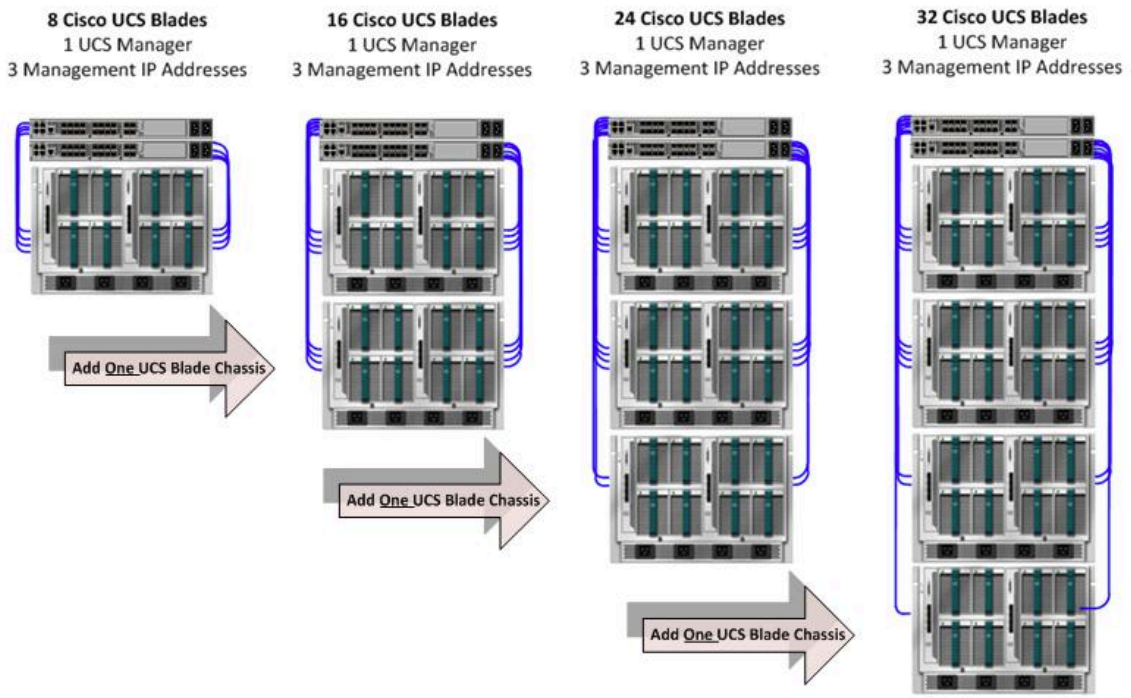


The Cisco Unified Computing System is a data center server platform composed of computing hardware, virtualization support, switching fabric, and management software.

The Cisco 6200 Series switch (called a "Fabric Interconnect") provides network connectivity for the chassis, blade servers and rack servers connected to it through 10 Gigabit converged network adapter.

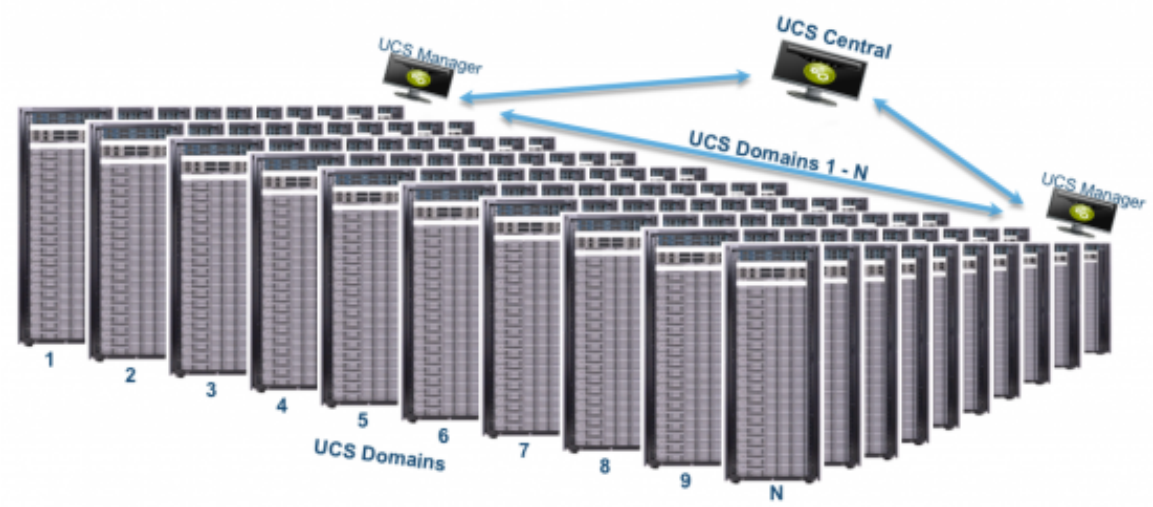
The Cisco UCS Manager software is embedded in the 6200 series Fabric Interconnect handles. The administrator accesses the interface via a web browser.

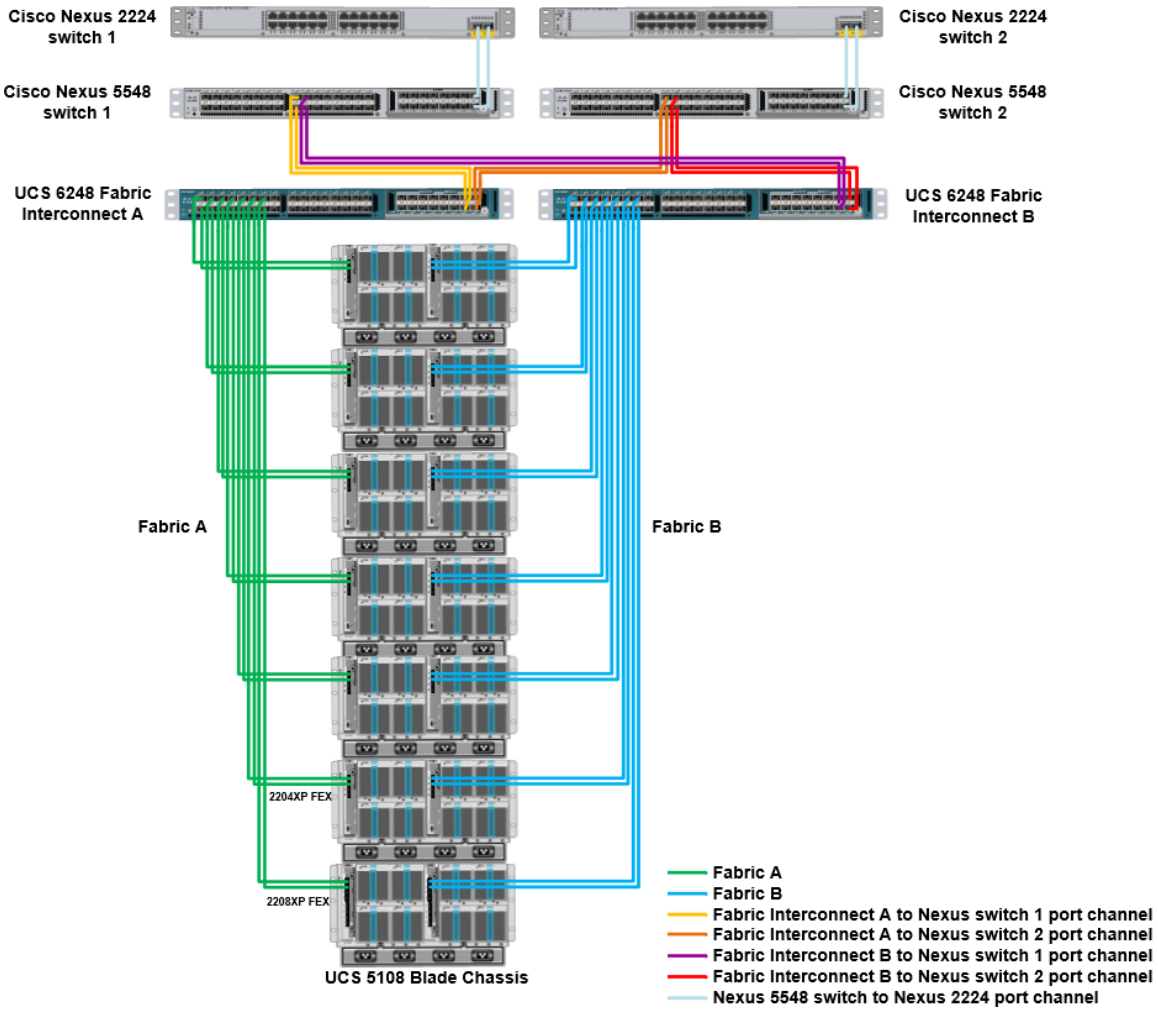
A key benefit is the concept of Stateless Computing, where each compute node has no set configuration. MAC addresses, UUIDs, firmware and BIOS settings for example, are all configured on the UCS manager in a Service Profile and applied to the servers.



In the diagram above you are able to see how to scale chassis in a UCS domain.

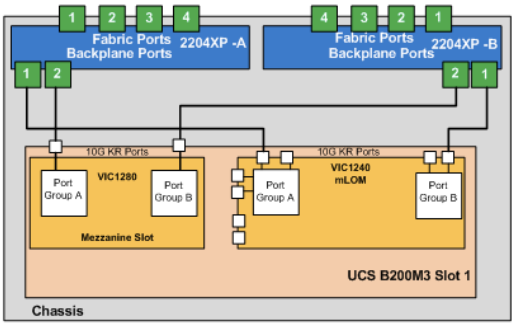
There is a maximum of 20 chassis per UCS domain. To scale beyond that, UCS central will be used to manage (n) domains.





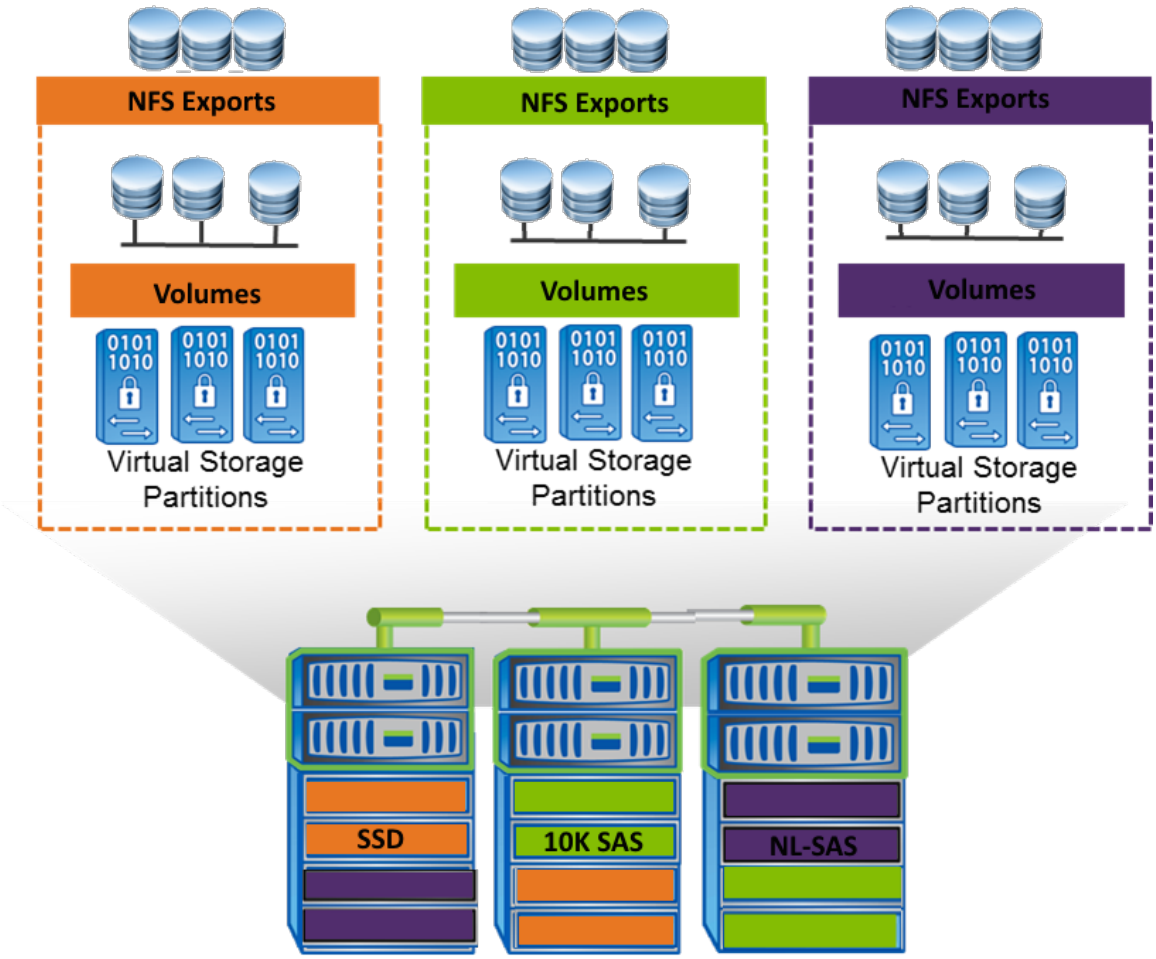
The above cabling diagram shows an example connectivity map with 7 chassis connecting to UCS 6248 Fabric Interconnects. The FIs then have a virtual port channel with the Nexus 5548 switches and then LAG connections with the fabric extenders.

The networking within the blade server is as below:



Mez	IOM	BandWidth
VIC 1280	2204XP	20Gb Per Fabric

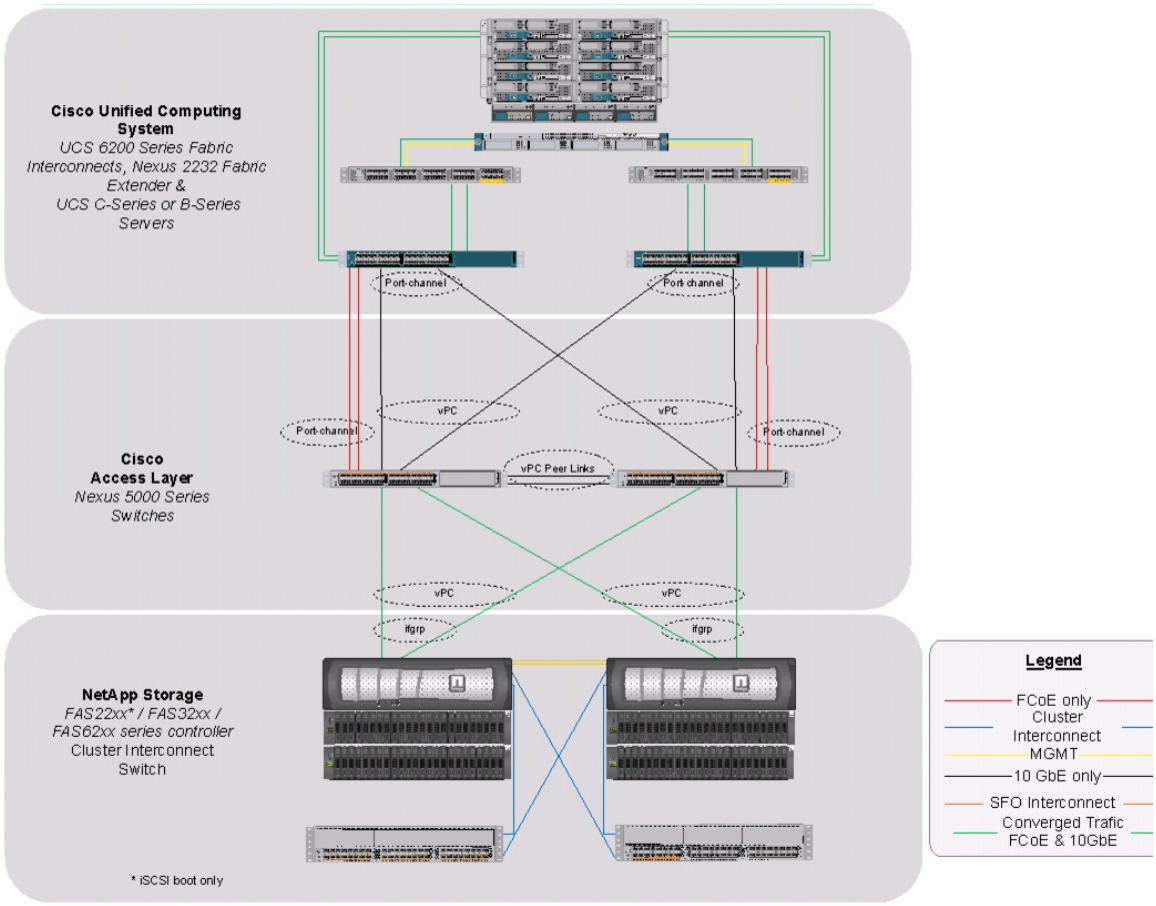
2.4.3 NetApp Architecture



The NetApp OnTap Cluster-Mode OS a horizontal scale out architecture. In the diagram above, you will see several cluster nodes. These nodes have the ability to span the filesystem across multiple aggregates and HA-Pairs by use of an SVM (Storage Virtual Machine). The SVM acts as a storage hypervisor and turns all the disks into resource pools that can be expanded and contracted as required.

By doing this, the system resilience and scalability has been increased. There is a risk that if not managed correctly that the storage system can start sprawling and troubleshooting performance issues is more time consuming due to the added complexity.

An example of the architecture is seen below.



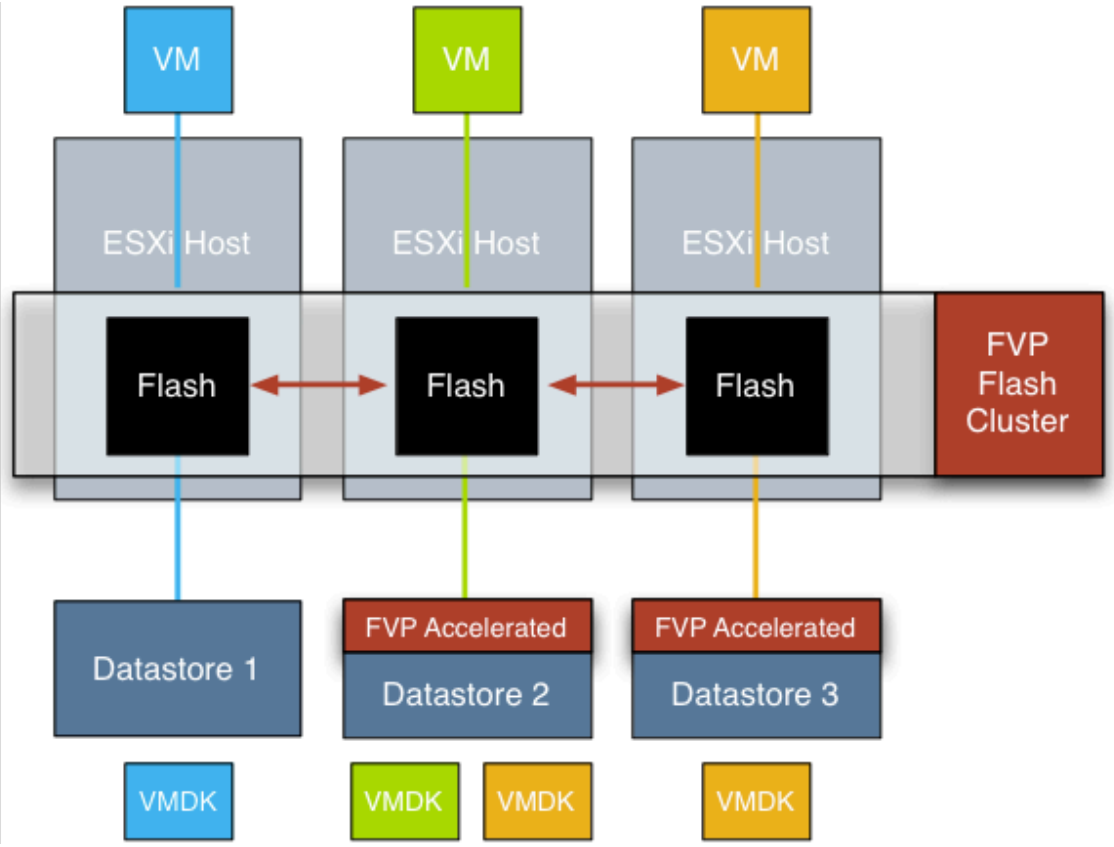
Detailed configuration guides for Cisco UCS, Cluster-Mode NetApp and Nexus switching can be reviewed in the Cisco Validated Designs (UCS_CVDs).

http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/esxi51_ucsm2_Clusterdeploy.html

2.4.4 PernixData Architecture

PernixData FVP virtualizes server-side caching that enables IT administrators to scale storage performance independent of capacity. Each server in the FVP Flash Cluster will make use of local SLC SSDs to cache read and write I/O of the storage system.

The VMs that require the extra IOPs will be placed on the FVP accelerated datastores.



2.5 Virtualization Network Layer

2.5.1 High Level Network Design Network Segmentation and VLANs

Within the Cisco UCS blades there are 2 VICs (virtual interface cards), an mLOM and a Mezzanine card. One is embedded in the server, the other is an add on card. These VICs then virtualize the physical NICs in the blade server so that any host OS running on the baremetal will see it as configured. In this design, we have presented 8 NICs to the host. These will be divided into the following VLANs and port groups.

VLAN 1011 – Management
VLAN 1020 – Storage Communication (data channel)
VLAN 1030 – vMotion
VLAN 2001-2999 – VM networks
Out of band (OOB) communication will be done on the Management network

The VIC to VNIC mapping is as follows:

VNIC0 – mLOM – Fabric A
VNIC1 - mLOM – Fabric B
VNIC2 - mLOM – Fabric A
VNIC3 - mLOM – Fabric B
VNIC4 - Mezz – Fabric A
VNIC5 - Mezz – Fabric B
VNIC6- Mezz – Fabric A
VNIC7- Mezz – Fabric B

2.5.2 Virtual Switches & Virtual Distributed Switches

There will be 2 vSwitches; a standard vSwitch for host management and a distributed vSwitch for all other communication. The uplinks will be as follows:

vSwitch0 is a VSS. Uplinks are:

VNIC0
VNIC5
There will be one management port group named VSS-1011-MGMT and one VMkernel, both on VLAN 1011

This is for interface redundancy. One is on the mLOM and the other is on the Mezzanine adapter.

vSwitch 1 is a VDS. Uplinks are

VNIC1
VNIC2
VNIC3
VNIC4
VNIC6
VNIC7

The VDS will have uplinks defined per port group.

The following port groups will be defined:

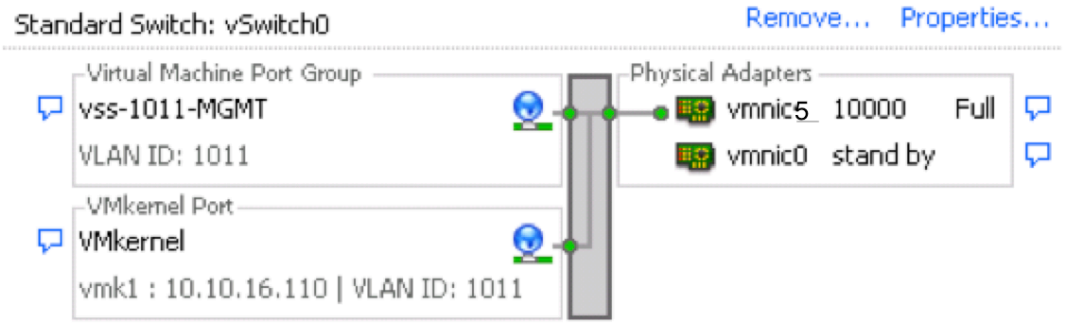
Name	VLAN	Purpose	Uplinks
VDS-1020-Storage	1020	Storage Connectivity	1,4
VDS-1030-vMotion	1030	vMotion	2,6
VDS-2001-VM	2001	VM Network	3,7

VM Network VLANs will span from VLAN 2001 to 2999 as required.

There is a VMKernel on VLAN 1020 for connectivity to the datastores and another on VLAN 1030 for vMotion.

2.5.3 NIC Teaming

The uplinks will be in active / standby configuration



2.5.4 Network I/O Control

Network IO control will not be applied on the VDS as there is already segmentation with the separate virtual interfaces in UCS. QoS will be applied at the UCS level by the Class of Service. Management will be Bronze, VM Data is Best Effort, and NFS is Gold. This is applied to the specific interfaces within UCS.

2.5.5 Physical Switches

The physical switches will be:

- Cisco UCS 6296 for the fabric interconnects
- Nexus 5548 for the upstream switches
- Nexus 2232 Fabric Extenders for any devices that require 1GB connectivity to the environment
- Nexus 5596 for the NetApp cluster Interconnects

2.5.6 DNS and Naming Conventions

Domain will be zombee.local
Hostnames will be constructed by region, site, role, numerical ID.

Example:
Region: Japan
Location: Tokyo University, Site1
Role: ESXi server
Numeric ID: 001

FQDN: JPN-TKY1-ESXi-001.zombee.local

2.6 ESXi Host Design

2.6.1 ESXi Host Hardware Requirements

Each blade server will have:

2 sockets with intel E52600 series CPUs (8 core).
512GB RAM
2 x 600GB SLC SSD drives
Cisco VIC1280 Mezzanine Card

The first blade in every chassis will have dual SD cards to boot ESXi from. This for the eventuality that a total power loss occurs and the vCenter server does not start. vCenter will have DRS affinity rules to those hosts.

2.6.2 Virtual Data Center Design

Each division will have it's own vCenter instance, named as per the naming conventions, ie: JPN-TKY1-VC-001

There will be one datacenter defined by the division name.

2.6.3 vSphere Single Sign On

Single Sign-on will be used and authenticated to Active Directory

2.6.4 vCenter Server and Database Systems (include vCenter Update Manager)

The vCenter Server Appliance (VCSA) will be used, as the total number of hosts is within the maximums. Update Manager will be deployed on it's own VM.

2.6.5 vCenter Server Database Design

The embedded database will be used for the vCenter server. A separate Windows server 2008 R2 server with SQL 2008 standard will be used for the shared database for other components, such as VUM.

2.6.6 vCenter AutoDeploy

vCenter AutoDeploy will be used to deploy all hosts in the compute cluster. All management clusters will be booting from SD cards.

2.6.7 Clusters and Resource Pools

There will be 3 clusters spanned across the 2 UCS cabinets and 10 chassis. 8 hosts will be in the 2 smaller clusters and 32 hosts will be in the larger cluster.

a. Enhanced vMotion Compatibility

2.6.8 Fault Tolerance (FT)

FT will not be used.

2.7 DRS Clusters

HA and DRS will be enabled and set to aggressive.

vCenter will have affinity rules to the first server in the chassis for the cluster it is in.
Any clustered application servers or databases will have anti-host affinity or if required, anti-chassis affinity rules.

HA admission control will be set to 25% tolerance.

2.7.1 Multiple vSphere HA and DRS Clusters

Each cluster will have the same roles, so the rules will stay the same. The only exception is where vCenter is located, which is cluster-01

2.7.2 Resource Pools

Resource pools will not be used unless required.

2.8 Management Layer Logical Design

2.8.1 vCenter Server Logical Design

The vCenter server will be on the VCSA with 32GB of RAM allocated to support all the hosts. Active directory will be installed on a VM named JPN-TKY1-DC-001. A shared SQL server for VUM will be located on a VM named JPN-TKY1-SQL-001. vCenter Update manager will be named JPN-TKY1-VUM-001.

2.8.2 Management and Monitoring

vCenter Operations Manager will be used to monitor the environment in detail.

Log insight will be used to provide real-time analysis and speedy root cause analysis.

2.9 Virtual Machine Design

2.9.1 Virtual Machine Design Considerations

Operating systems will be comprised of a system volume and one or more data volumes. System volumes will not be larger than 100GB in size and will be thin provisioned by default.

Swap files for all VMs will be located on a swap datastore.
No RDMs will be used.

Virtual Hardware Version 10 will be used by default.
All operating systems that support VMXNET3 will use it as the network adapter of choice.

2.9.2 Guest Operating System Considerations

All VMs will be provisioned from templates and configuration policies applied afterwards. No configuration changes will be applied manually.

2.9.3 General Management Design Guidelines

All management of the environment will be done by authorized personnel only, from a dedicated VM named JPN-TKY1-MGMT-001. All actions will be audited and reviewed.

2.9.4 Host Management Considerations

The UCS servers have a built in IP KVM in their management framework. The UCS plugin will be added to vCenter so that baremetal host management can occur from there.

2.9.5 vCenter Server Users and Groups

Active Directory will be used as the user management system. Roles based on access levels and job function will be applied to the appropriate groups.

2.9.6 Management Cluster

Cluster 1 will be the management cluster. All VMs for management purposes, whether virtual infrastructure, storage or otherwise, will run on cluster 1.

2.9.7 Management Server Redundancy

vCenter server is protected by HA across the cluster. In a worst case scenario, the blades with SD cards will still be able to boot and start the vCenter server if the storage is accessible.

2.9.8 Templates

All virtual machines will be spawned from a master image in Openstack. Using VAAI, rapid cloning of virtual machines can be done.

2.9.9 Updating Hosts, Virtual Machines, and Virtual Appliances

vCenter update manager will be used to update hosts, VMs and appliances.

2.9.10 Time Synchronization

A GPS synced time server will be used for accurate time measurement. All hosts will connect to that NTP server. Active Directory VMs will be set to obtain the time of their hosts, then provide it to network computers.

2.9.11 Snapshot Management

VM snapshots will use a 10% volume reserve and occur 30min intervals. Retention will be hourly, daily, weekly and monthly. 10 snapshots for each type will be used.

2.10.1 Performance Monitoring

Performance monitoring will be done by vCOPs for the infrastructure and vFabric Hyperic for application analysis.

2.10.2 Alarms

All default vCenter alarms are enabled and the recipient is set to an internal email user.

2.10.3 Logging Design Considerations

Log insight will be deployed as well as the dump collector. Logs will be configured to go to a shared datastore.

2.11 Infrastructure Backup and Restore

2.11.1 Compute (ESXi) Host Backup and Restore

Host configuration will be maintain in a single host profile and backed up with a PowerCli script.

2.11.2 vSphere Replication

vSphere replication will not be used, at this time.

2.11.3 vSphere Distributed Switch Backup and Restore

The VDS config will be backed up from the vSphere client

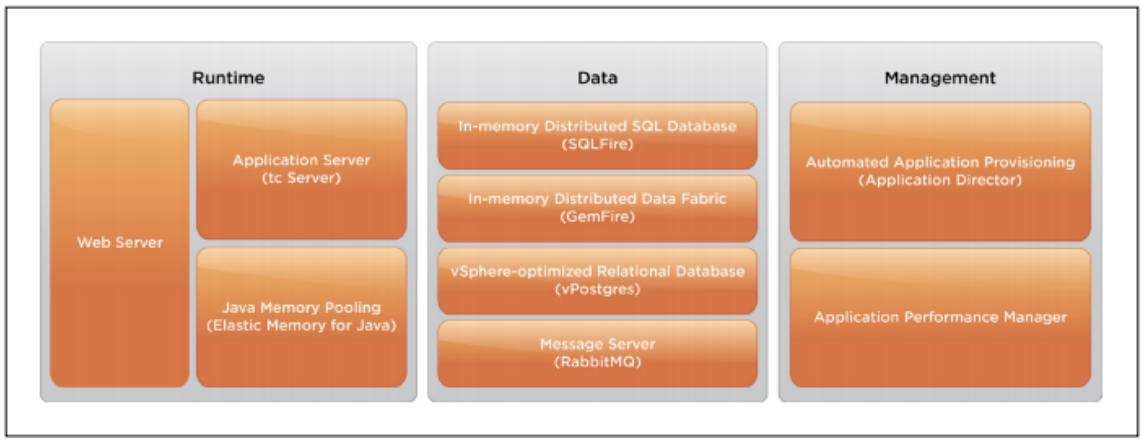
2.11.4 vCenter Databases

vCenter uses the internal VCSA database. The other components that require external databases are using JPN-TKY1-SQL-001

2.12. Application provisioning automation

All VMs will be provisioned from templates then puppet will apply the appropriate configuration. For custom-built applications, the vFabric Suite will be used.

2.12.1 vFabric Overview



vFabric is a development and automation framework for creating multi-tier application and managing code release cycles. With the combination of vFabric Hyperic, (for testing application performance) and vFabric Application Director (for scaling and deployment automation), complex applications can be created, tested and deployed using a Continuous Integration and Continuous Deployment model.