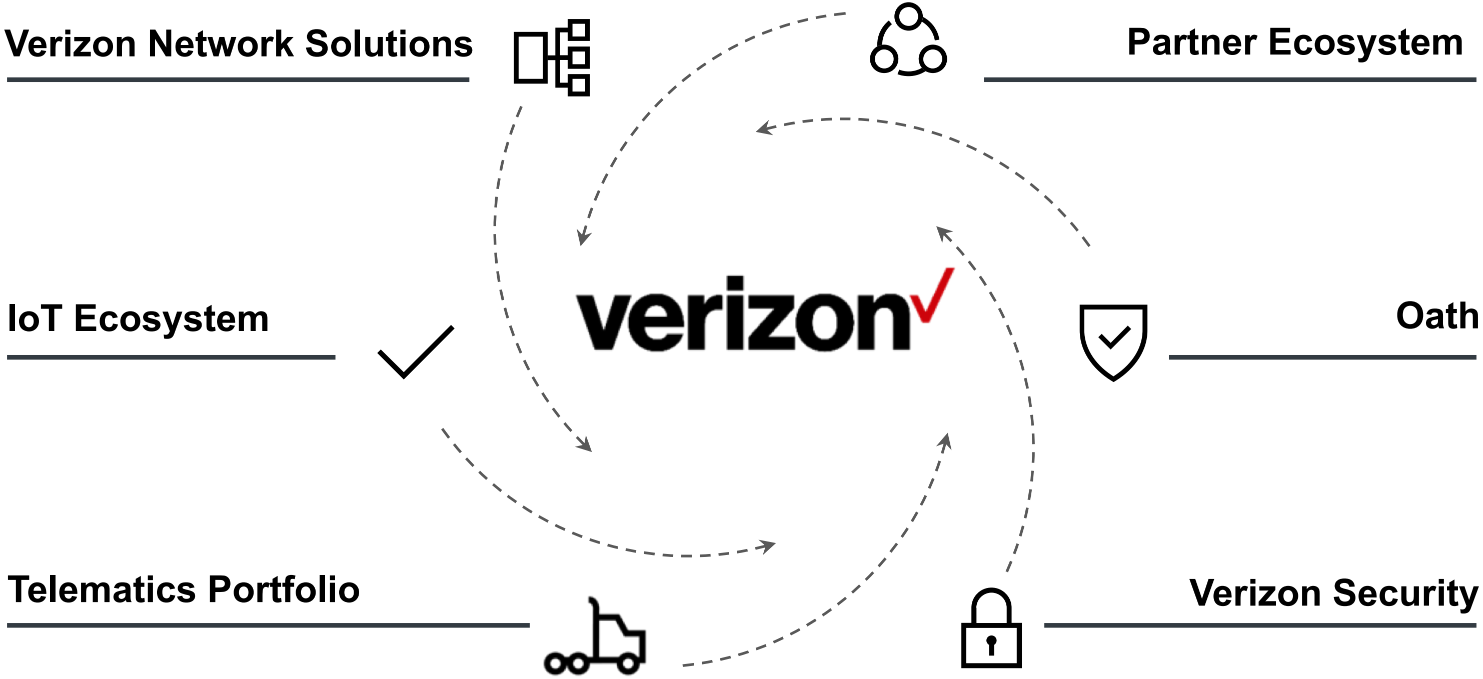
Telematics Container Cloud Platform

TCCP Architecture & Deployment

2018-03-05: Rick Davis

The power of Verizon



Verizon Network Solutions

Partner Ecosystem

IoT Ecosystem

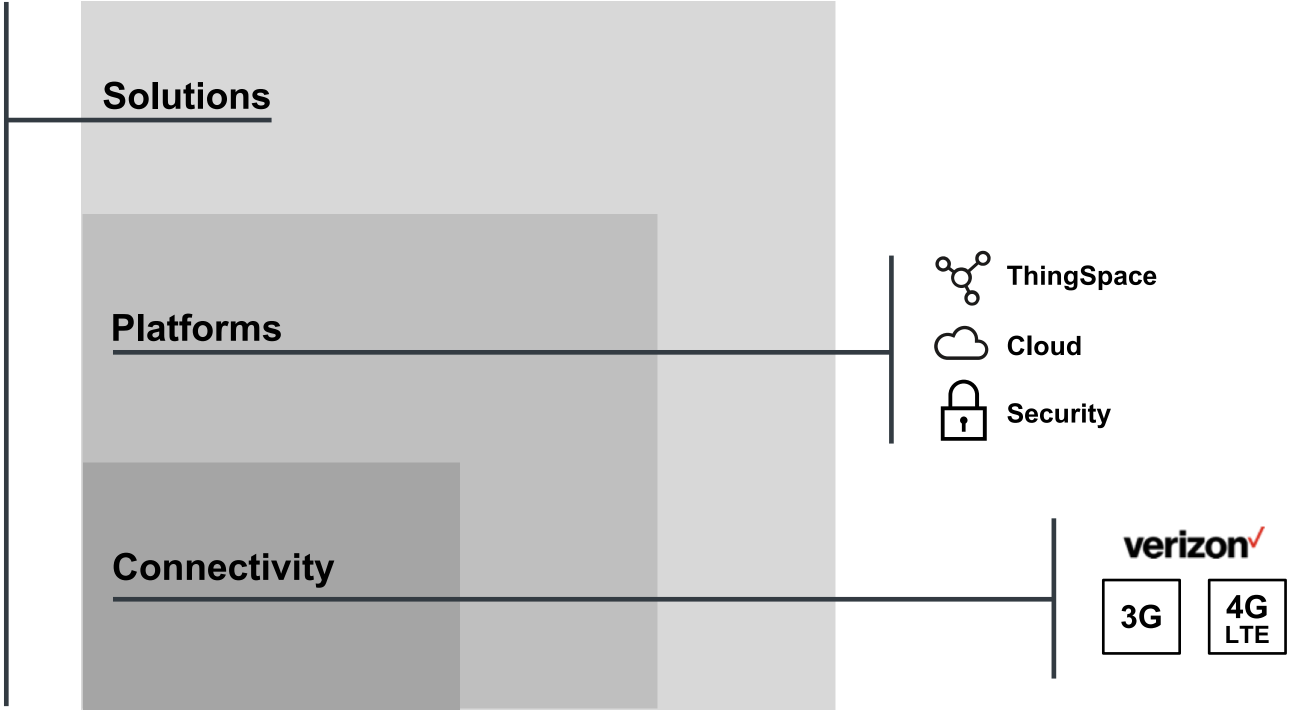
Oath

Telematics Portfolio

Verizon Security

Our technology portfolio is second to none

- IoT-enabled solutions
- Telematics solutions
- Advanced communications
- Managed mobility
- Secure networking



Solutions

Platforms

Connectivity

ThingSpace

Cloud

Security

Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

Agenda

1. Goals
2. Architecture
3. Automation
4. Collaboration & Lessons Learned
5. Q&A

Goals

What do we want to do?

Business Drivers

Increase Development Velocity

- Monolith to Microservice
- Latest Software Frameworks
- Containerized
- 100% CICD

Increase Infrastructure Velocity

- Site Rebuild & App Deploys < 12hr
- Repeatable/Predictable
- 100% CICD

Future Proofing

- Flexible & Extensible
- Easily Scalable
- Cloud Adaptable & Agnostic

Infrastructure Goals

Architecture

- Cost Conscious
  - Open Hardware
  - Open Source Software, Commercial Support Optional
- Global Deployment Capability w/Vendor Support
- Public & Private Cloud Adaptable

Automation

- Server & Network Provisioning
- Integrated Validations
- Infrastructure-as-Code

Extensibility

- Platform with Swappable Components
- Avoid Lock-in, Everywhere
  - Hardware, Software, and Cloud

Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

Architecture

How are we going to do it?

Architecture

eBGP Clos Fabric

- One Protocol/Simplified OPS
- Quagga with ECMP on Hosts
- Anycast Support

Open Source Routing Quagga

Spine/Leaf Topology

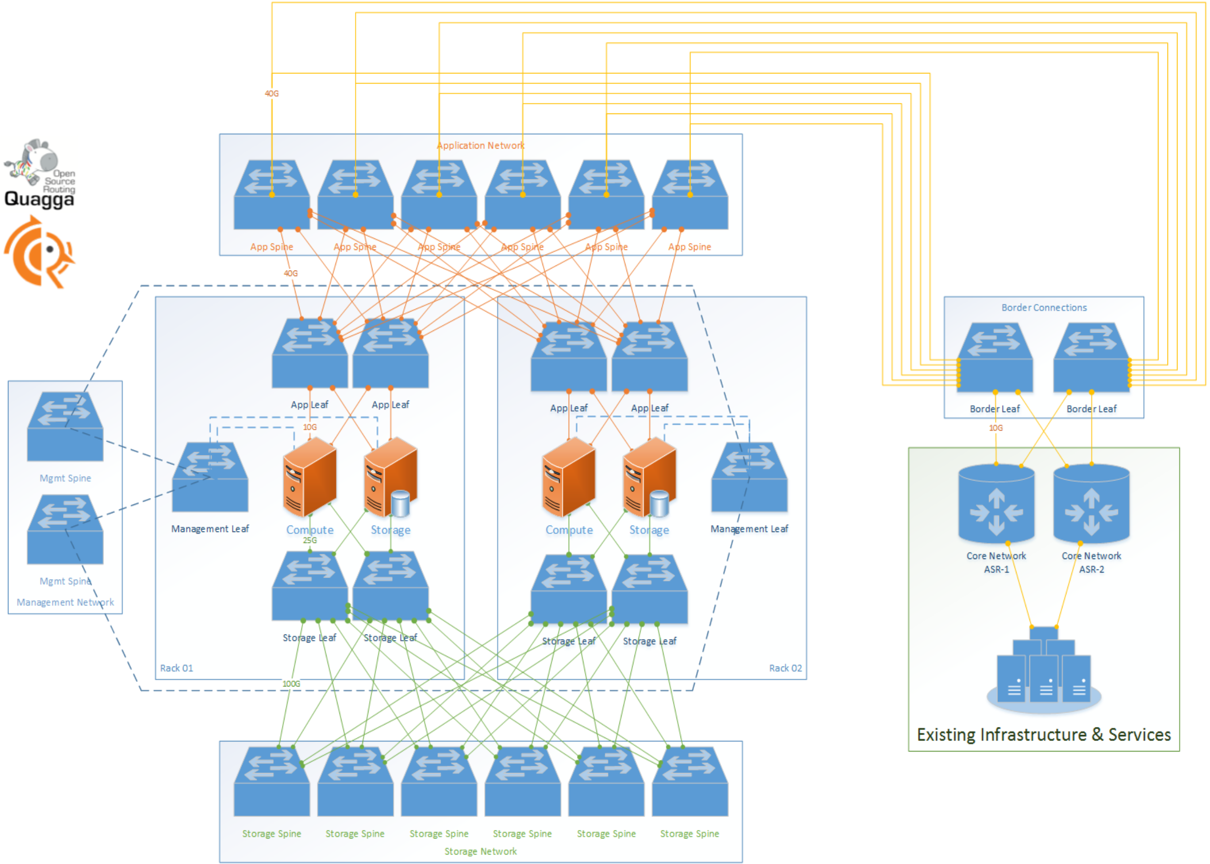
- Dedicated Storage Network
- Dedicated Border/Peering

Simple Scalability

- 100’s of Racks/Site

Open Hardware & Software, Commercial Support Optional

Open Switch

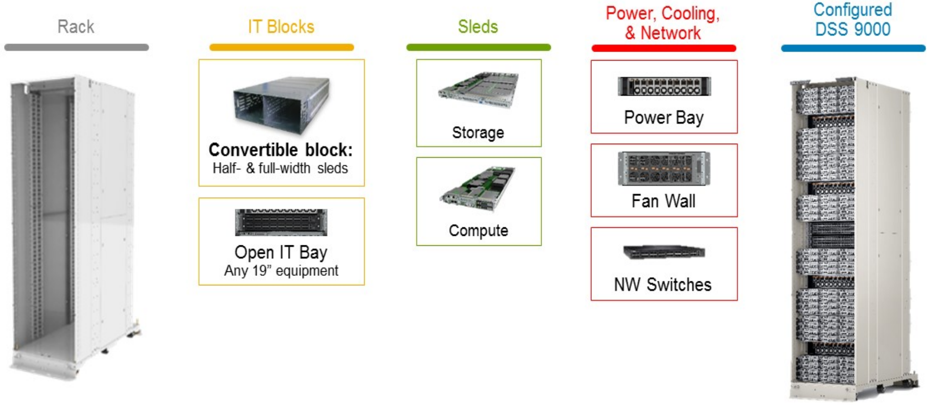


Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

Hardware Platforms

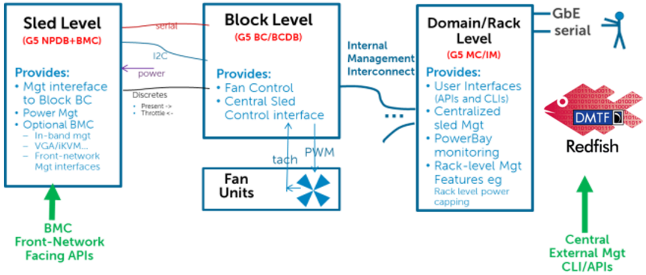
Dell EMC DSS 9000 Servers & Storage

- Open Compute based infrastructure avoids lock-in
- Rack-Level Hyperscale Infrastructure
- Rack-Level Management – IPMI/Redfish
- Flexible Compute & Storage Configurations
- OEM support & reliability, ODM pricing



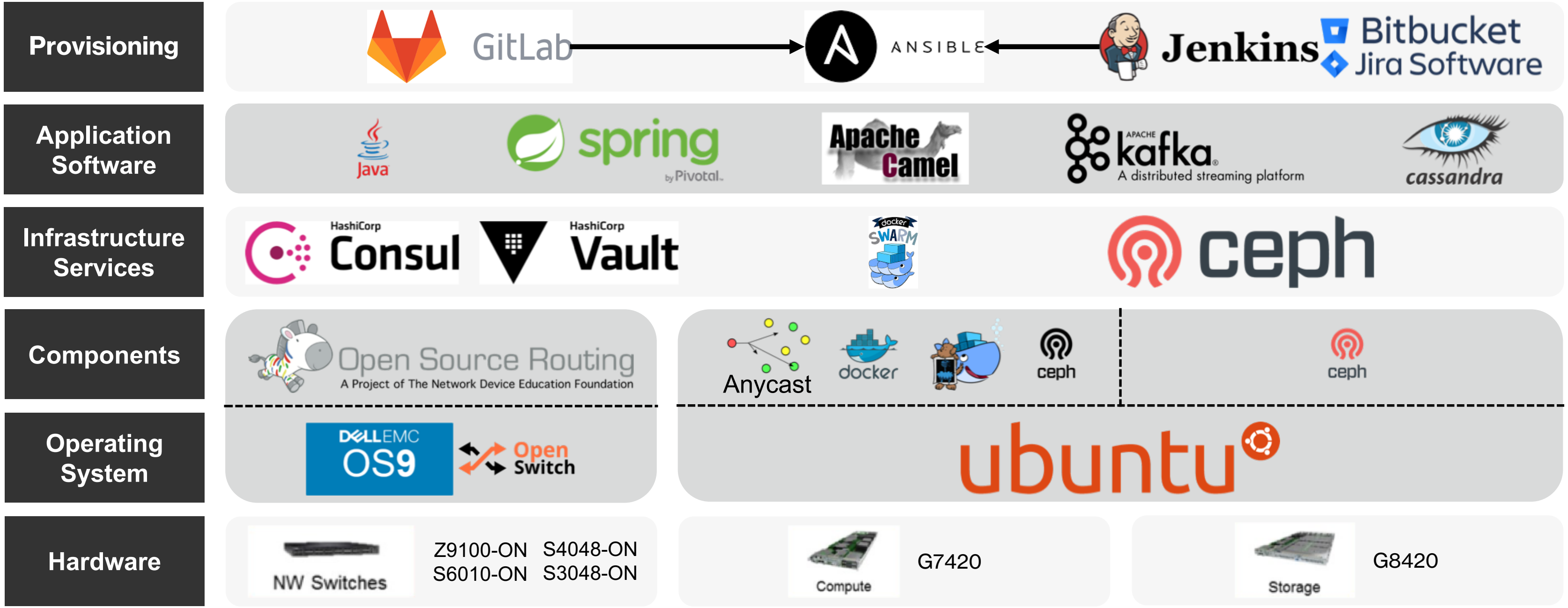
Dell EMC Open Networking Switches

- Swappable Network OS via ONIE
- Latest high performance platform (e.g. Tomahawk 4x25G/100G)
- Commodity silicon avoids vendor lock-in (Z9100 ~= FB Wedge100)
- OEM support & reliability, ODM pricing



Source: Dell EMC ESI

Infrastructure Stack



Provisioning
GitLab → ANSIBLE ← Jenkins
Bitbucket
Jira Software

Application Software
Java
spring by Pivotal
Apache Camel
Apache kafka A distributed streaming platform
cassandra

Infrastructure Services
HashiCorp Consul
HashiCorp Vault
docker SWARM
ceph

Components
Open Source Routing A Project of The Network Device Education Foundation
Anycast
docker
ceph
ceph

Operating System
DELL EMC OS9
Open Switch
ubuntu

Hardware
NW Switches
Z9100-ON S4048-ON
S6010-ON S3048-ON
Compute
G7420
Storage
G8420

Logical Architecture

Service-by-Service Failover

Service-by-Service Isolation

- VXLAN/MACVLAN Isolation
- Line Rate ACL's in Silicon

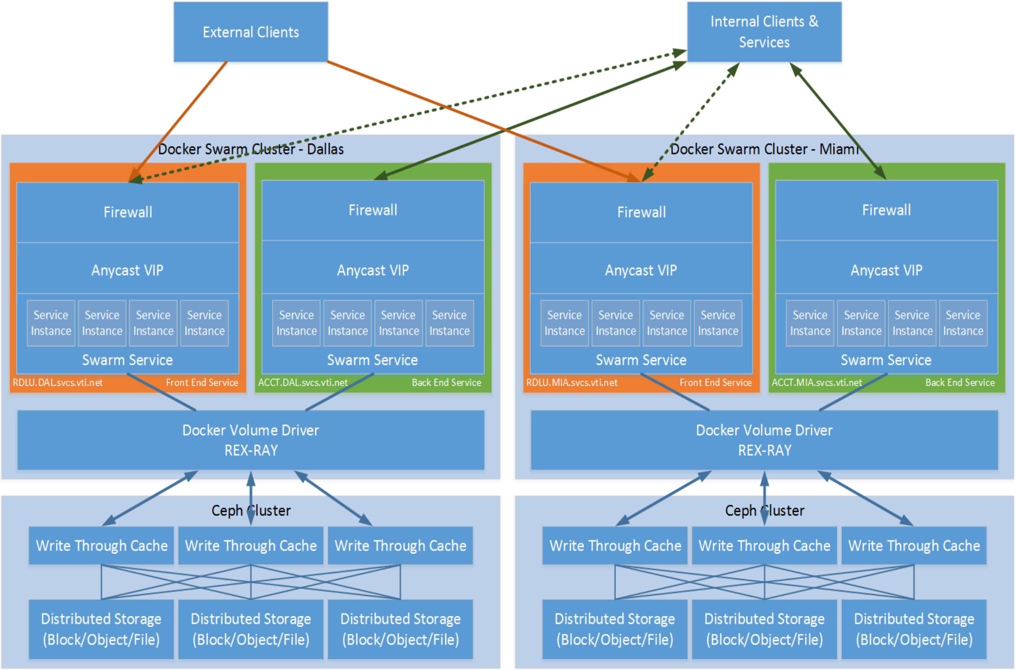
Anycast Load Balancing

Ceph Block/Object/File Store

- Persistent Container Storage via REX-Ray

Independent Clusters

- Failure Domain Isolation
- Site-to-Site Replication in App/DB Control



External Clients

Internal Clients & Services

Docker Swarm Cluster - Dallas

Firewall

Anycast VIP

Service Instance | Service Instance | Service Instance | Service Instance

Swarm Service

RDLU.DAL.svcs.vti.net — Front End Service

Firewall

Anycast VIP

Service Instance | Service Instance | Service Instance | Service Instance

Swarm Service

ACCT.DAL.svcs.vti.net — Back End Service

Docker Volume Driver
REX-RAY

Ceph Cluster

Write Through Cache | Write Through Cache | Write Through Cache

Distributed Storage (Block/Object/File) | Distributed Storage (Block/Object/File) | Distributed Storage (Block/Object/File)

Docker Swarm Cluster - Miami

Firewall

Anycast VIP

Service Instance | Service Instance | Service Instance | Service Instance

Swarm Service

RDLU.MIA.svcs.vti.net — Front End Service

Firewall

Anycast VIP

Service Instance | Service Instance | Service Instance | Service Instance

Swarm Service

ACCT.MIA.svcs.vti.net — Back End Service

Docker Volume Driver
REX-RAY

Ceph Cluster

Write Through Cache | Write Through Cache | Write Through Cache

Distributed Storage (Block/Object/File) | Distributed Storage (Block/Object/File) | Distributed Storage (Block/Object/File)

Docker Swarm – Our Perspective (Early 2016)

Decision Factors

- Developer Simplicity & Talent Pool
- Blue/Green Deployment & Rolling Upgrades
- Network, Volume, Logging Drivers
- API Compatibility & Ecosystem Tools
- Windows/Mac/Linux Hosts
- Windows & Linux Containers

Reduced Complexity

Swappable Components

Large Ecosystem

Sounds good…what if it doesn't work?

Swap Components

- **Need cloud?**
  - Swap provisioning to cloud
- **Need Kubernetes?**
  - Enable in Docker or swap it
- **Need etcd?**
  - Add it or swap it

Handover

- **Certify hardware with other VZ platforms**
- **Handover few racks for platform deployment**
- **Convert apps**
- **Validate, update CICD**
- **Deploy in PROD**
- **Convert remaining racks**

Migrate & Repurpose

- **Validate/convert apps and infrastructure services**
  - Service Discovery, Logging, Monitoring, etc, etc
- **Validate, update CICD**
- **Deploy in PROD**
- **Repurpose existing racks**

Automation

How do we keep doing it?

Automation Goals

Simplified

Fully Automated

Easily Repeatable & Predictable

Zero Touch & Zero Downtime

Standardized

Ansible First

Bash/Python Second

Same CI Pipeline, Every Site

End-to-End

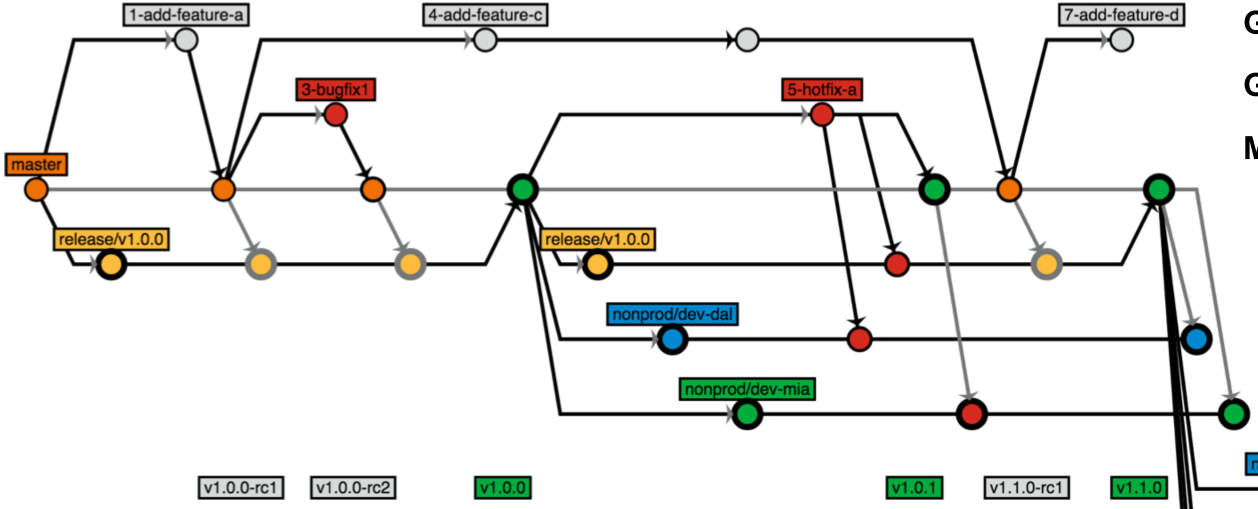
Server AND Network

- Configuration AND Provisioning

Public & Private Cloud Adaptable

Automated Maintenance & Healing

CI Pipeline & Deployment Management



GitLab Flow based model

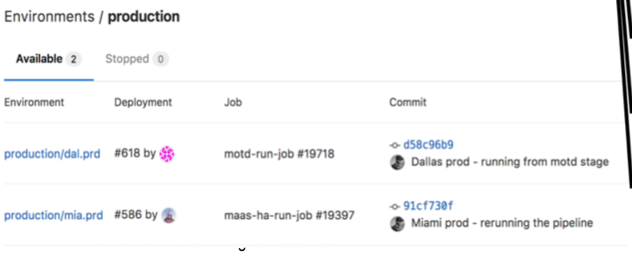
Git Tags = Immutable

Maintenance Automation

Fully Automated: Eliminate human error

Code/OPS/Change Review: Merge Request

Branch Protections: RBAC


Environments / **production**

Environment	Deployment	Job	Commit
production/dal.prd	#618 by	motd-run-job #19718	d58c96b9 Dallas prod - running from motd stage
production/mia.prd	#586 by	maas-ha-run-job #19397	91cf730f Miami prod - rerunning the pipeline

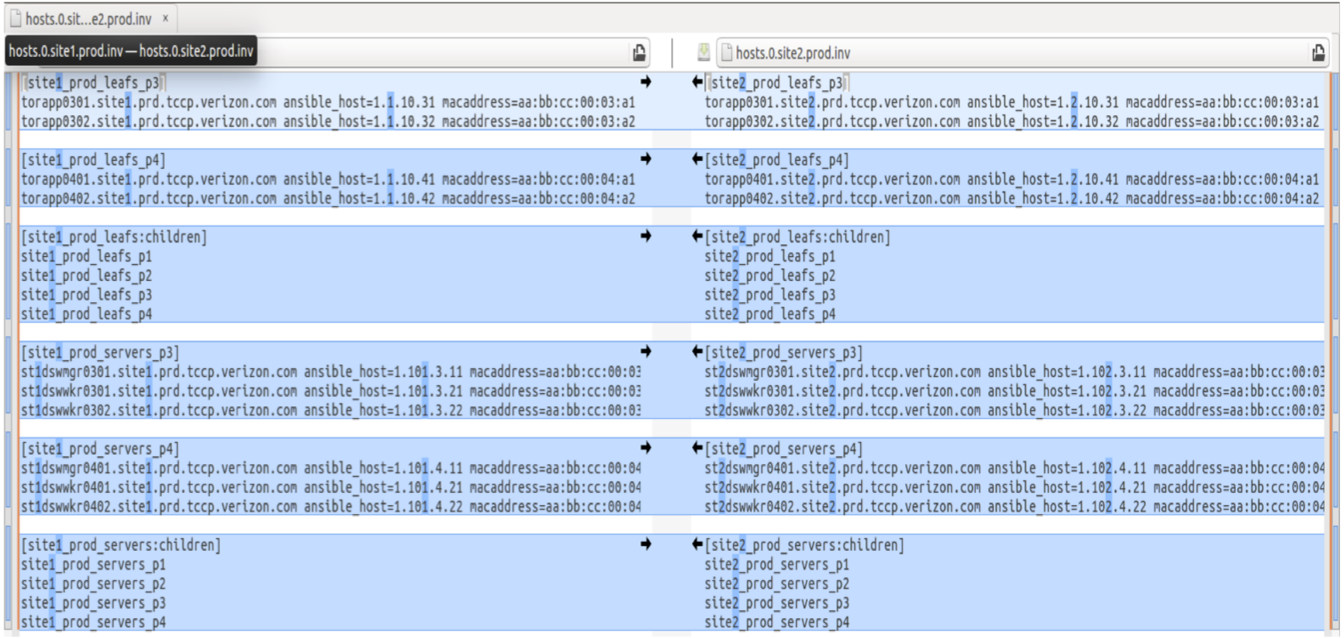
Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

Simple Scalability – Adding Racks

Add rack group (P4)

Update device details

Execute CI Pipeline


hosts.0.sit...e2.prod.inv ×
hosts.0.site1.prod.inv — hosts.0.site2.prod.inv
hosts.0.site2.prod.inv

```
[site1_prod_leafs_p3]
torapp0301.site1.prd.tccp.verizon.com ansible_host=1.1.10.31 macaddress=aa:bb:cc:00:03:a1
torapp0302.site1.prd.tccp.verizon.com ansible_host=1.1.10.32 macaddress=aa:bb:cc:00:03:a2

[site1_prod_leafs_p4]
torapp0401.site1.prd.tccp.verizon.com ansible_host=1.1.10.41 macaddress=aa:bb:cc:00:04:a1
torapp0402.site1.prd.tccp.verizon.com ansible_host=1.1.10.42 macaddress=aa:bb:cc:00:04:a2

[site1_prod_leafs:children]
site1_prod_leafs_p1
site1_prod_leafs_p2
site1_prod_leafs_p3
site1_prod_leafs_p4

[site1_prod_servers_p3]
st1dswmgr0301.site1.prd.tccp.verizon.com ansible_host=1.101.3.11 macaddress=aa:bb:cc:00:03
st1dswwkr0301.site1.prd.tccp.verizon.com ansible_host=1.101.3.21 macaddress=aa:bb:cc:00:03
st1dswwkr0302.site1.prd.tccp.verizon.com ansible_host=1.101.3.22 macaddress=aa:bb:cc:00:03

[site1_prod_servers_p4]
st1dswmgr0401.site1.prd.tccp.verizon.com ansible_host=1.101.4.11 macaddress=aa:bb:cc:00:04
st1dswwkr0401.site1.prd.tccp.verizon.com ansible_host=1.101.4.21 macaddress=aa:bb:cc:00:04
st1dswwkr0402.site1.prd.tccp.verizon.com ansible_host=1.101.4.22 macaddress=aa:bb:cc:00:04

[site1_prod_servers:children]
site1_prod_servers_p1
site1_prod_servers_p2
site1_prod_servers_p3
site1_prod_servers_p4
```

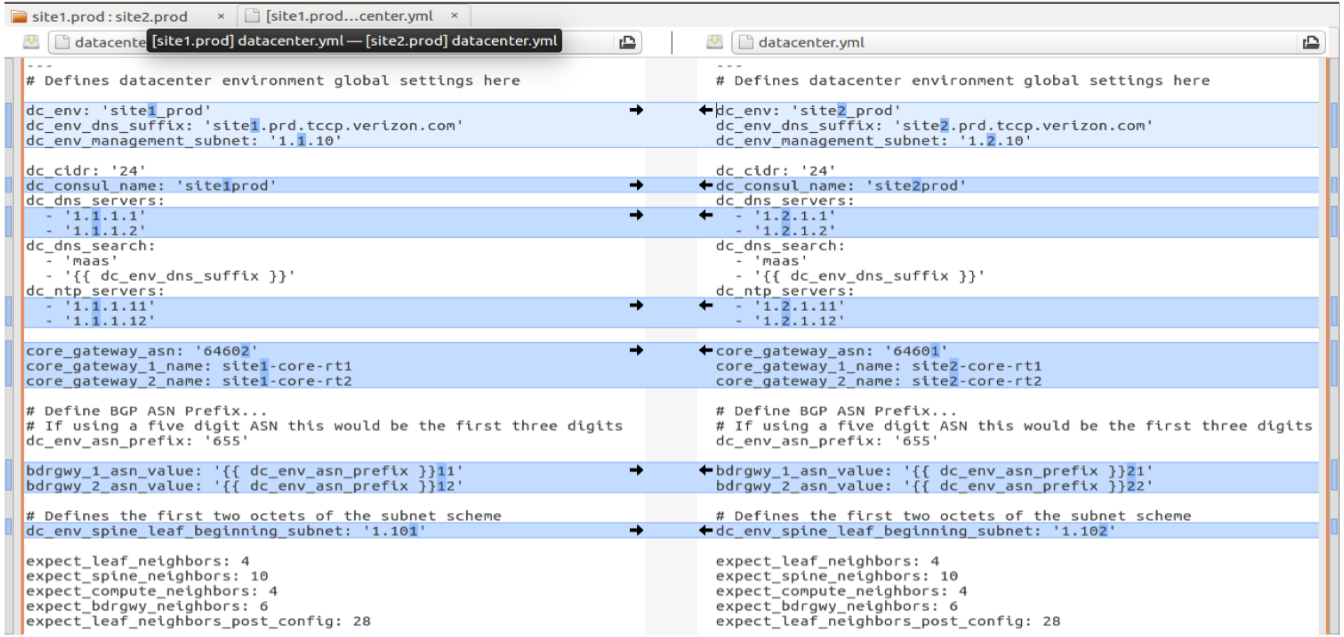
```
[site2_prod_leafs_p3]
torapp0301.site2.prd.tccp.verizon.com ansible_host=1.2.10.31 macaddress=aa:bb:cc:00:03:a1
torapp0302.site2.prd.tccp.verizon.com ansible_host=1.2.10.32 macaddress=aa:bb:cc:00:03:a2

[site2_prod_leafs_p4]
torapp0401.site2.prd.tccp.verizon.com ansible_host=1.2.10.41 macaddress=aa:bb:cc:00:04:a1
torapp0402.site2.prd.tccp.verizon.com ansible_host=1.2.10.42 macaddress=aa:bb:cc:00:04:a2

[site2_prod_leafs:children]
site2_prod_leafs_p1
site2_prod_leafs_p2
site2_prod_leafs_p3
site2_prod_leafs_p4

[site2_prod_servers_p3]
st2dswmgr0301.site2.prd.tccp.verizon.com ansible_host=1.102.3.11 macaddress=aa:bb:cc:00:03
st2dswwkr0301.site2.prd.tccp.verizon.com ansible_host=1.102.3.21 macaddress=aa:bb:cc:00:03
st2dswwkr0302.site2.prd.tccp.verizon.com ansible_host=1.102.3.22 macaddress=aa:bb:cc:00:03

[site2_prod_servers_p4]
st2dswmgr0401.site2.prd.tccp.verizon.com ansible_host=1.102.4.11 macaddress=aa:bb:cc:00:04
st2dswwkr0401.site2.prd.tccp.verizon.com ansible_host=1.102.4.21 macaddress=aa:bb:cc:00:04
st2dswwkr0402.site2.prd.tccp.verizon.com ansible_host=1.102.4.22 macaddress=aa:bb:cc:00:04

[site2_prod_servers:children]
site2_prod_servers_p1
site2_prod_servers_p2
site2_prod_servers_p3
site2_prod_servers_p4
```

Simple Scalability – Adding Sites

Add datacenter config

Add racks

Execute CI Pipeline



```
---
# Defines datacenter environment global settings here

dc_env: 'site1_prod'
dc_env_dns_suffix: 'site1.prd.tccp.verizon.com'
dc_env_management_subnet: '1.1.10'

dc_cidr: '24'
dc_consul_name: 'site1prod'
dc_dns_servers:
  - '1.1.1.1'
  - '1.1.1.2'
dc_dns_search:
  - 'maas'
  - '{{ dc_env_dns_suffix }}'
dc_ntp_servers:
  - '1.1.1.11'
  - '1.1.1.12'

core_gateway_asn: '64602'
core_gateway_1_name: site1-core-rt1
core_gateway_2_name: site1-core-rt2

# Define BGP ASN Prefix...
# If using a five digit ASN this would be the first three digits
dc_env_asn_prefix: '655'

bdrgwy_1_asn_value: '{{ dc_env_asn_prefix }}11'
bdrgwy_2_asn_value: '{{ dc_env_asn_prefix }}12'

# Defines the first two octets of the subnet scheme
dc_env_spine_leaf_beginning_subnet: '1.101'

expect_leaf_neighbors: 4
expect_spine_neighbors: 10
expect_compute_neighbors: 4
expect_bdrgwy_neighbors: 6
expect_leaf_neighbors_post_config: 28
```

```
---
# Defines datacenter environment global settings here

dc_env: 'site2_prod'
dc_env_dns_suffix: 'site2.prd.tccp.verizon.com'
dc_env_management_subnet: '1.2.10'

dc_cidr: '24'
dc_consul_name: 'site2prod'
dc_dns_servers:
  - '1.2.1.1'
  - '1.2.1.2'
dc_dns_search:
  - 'maas'
  - '{{ dc_env_dns_suffix }}'
dc_ntp_servers:
  - '1.2.1.11'
  - '1.2.1.12'

core_gateway_asn: '64601'
core_gateway_1_name: site2-core-rt1
core_gateway_2_name: site2-core-rt2

# Define BGP ASN Prefix...
# If using a five digit ASN this would be the first three digits
dc_env_asn_prefix: '655'

bdrgwy_1_asn_value: '{{ dc_env_asn_prefix }}21'
bdrgwy_2_asn_value: '{{ dc_env_asn_prefix }}22'

# Defines the first two octets of the subnet scheme
dc_env_spine_leaf_beginning_subnet: '1.102'

expect_leaf_neighbors: 4
expect_spine_neighbors: 10
expect_compute_neighbors: 4
expect_bdrgwy_neighbors: 6
expect_leaf_neighbors_post_config: 28
```

verizon connect

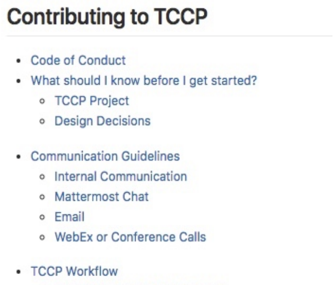
Collaboration & Lessons Learned

What should we have done?

Collaboration

Collaboration

- Issues/Merge Requests, Milestones, and Boards
- GitLab Flow Based Model
- Source Controlled CI
- Reusable repositories

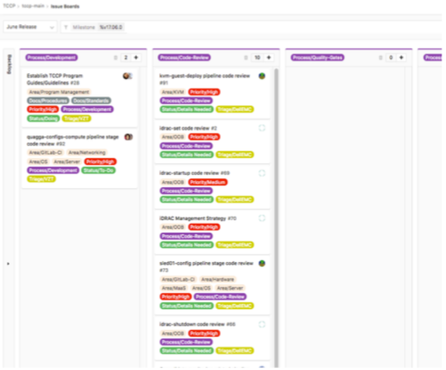


Contributing to TCCP

- Code of Conduct
- What should I know before I get started?
  - TCCP Project
  - Design Decisions
- Communication Guidelines
  - Internal Communication
  - Mattermost Chat
  - Email
  - WebEx or Conference Calls
- TCCP Workflow

Communication

- Wiki & GitLab Pages
- ChatOps
- CI Analytics



TCCP › tccp-main › Issue Boards

June Release | Milestone %v17.06.0

Process/Development 2

Establish TCCP Program Guides/Guidelines #28

quagga-configs-compute pipeline stage code review #92

Process/Code-Review 10

kvm-guest-deploy pipeline code review #91

idrac-set code review #2

idrac-startup code review #69

iDRAC Management Strategy #70

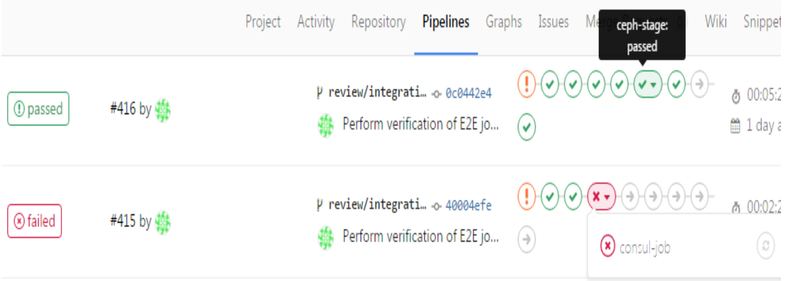
sled01-config pipeline stage code review #73

idrac-shutdown code review #66

Process/Quality-Gates 0

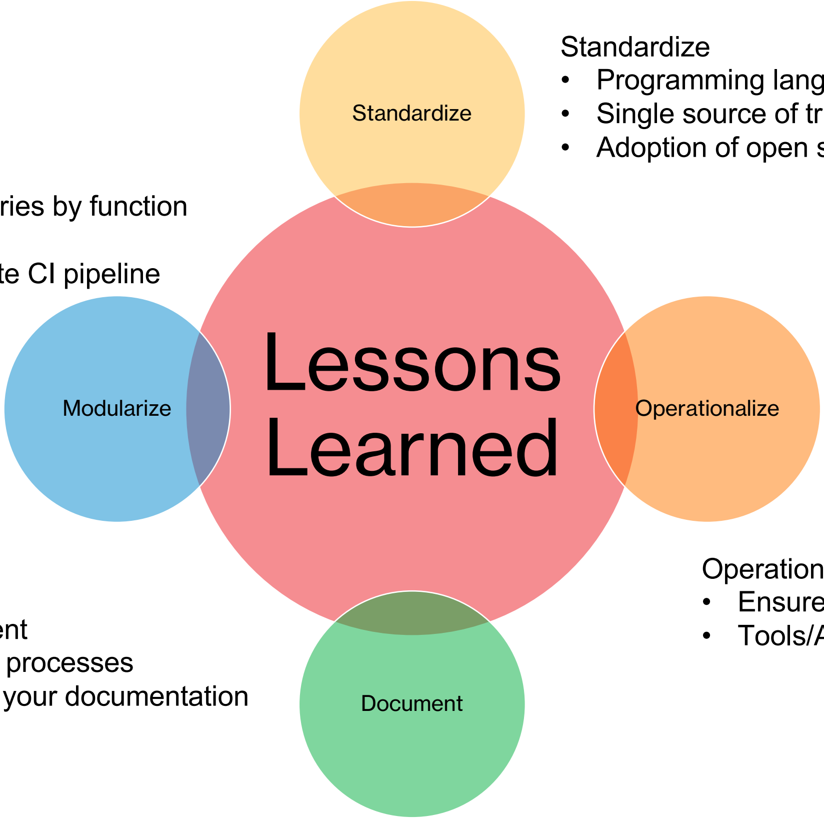
Flexibility

- Single Solution, Multiple Integrations
- Push & Pull Git Mirroring
- Granular Permissions by Team/Group/Project
- Multi-project CI pipelines



passed #416 by review/integrati… 0c0442e4 Perform verification of E2E jo… 00:05:2 1 day a

failed #415 by review/integrati… 40004efe Perform verification of E2E jo… 00:02:2 consul-job



Standardize

- Programming language
- Single source of truth – Ansible inventory
- Adoption of open source roles

Modularize

- Separate repositories by function
- Consolidate roles
- Logically segregate CI pipeline

Operationalize

- Ensure idempotency
- Tools/Alerts/Dashboards in roles

- Document
- Change processes
- Code is your documentation

Q&A

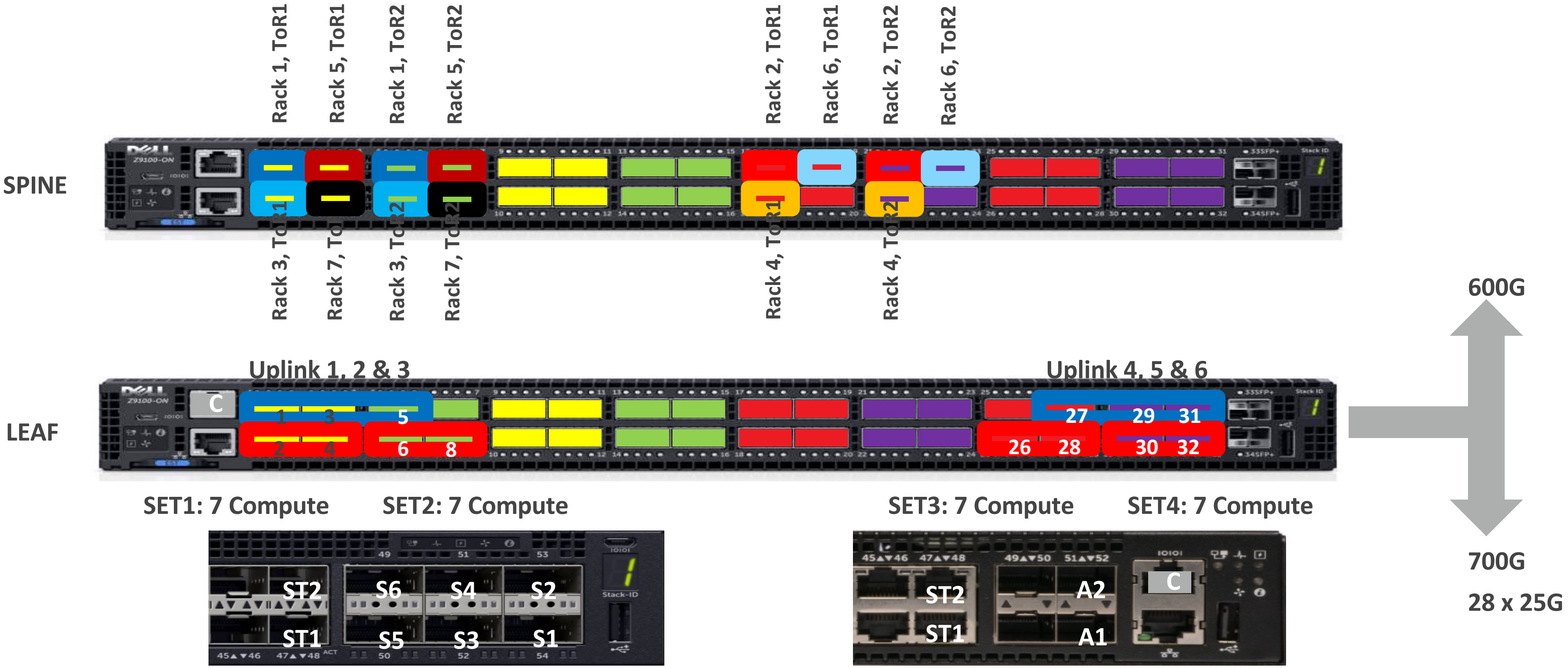
verizon✓
connect

Thank you.

verizon✓
connect

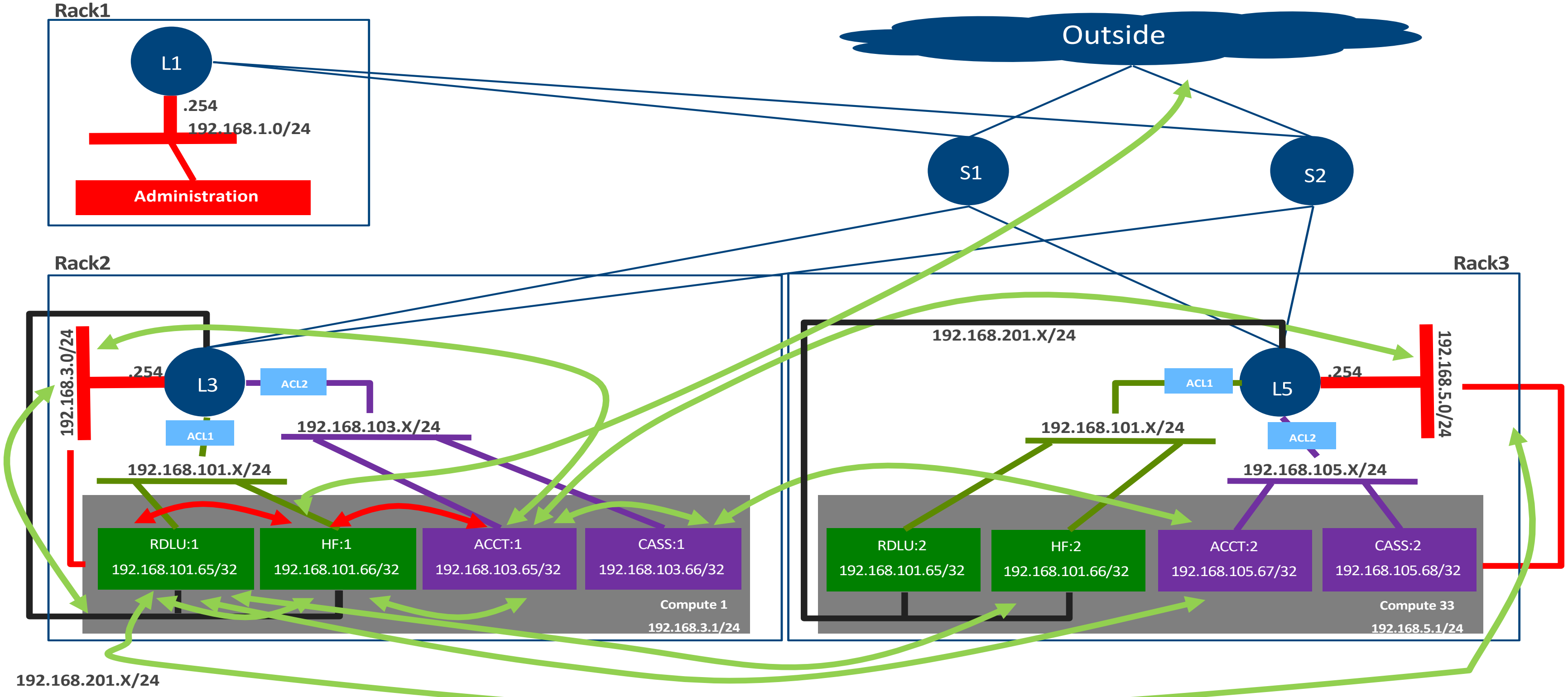
BACKUP

Z9100 Port Mapping & Buffer Zone Utilization



Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

MACVLAN & Anycast is Easy :)


Rack1
L1
.254
192.168.1.0/24
Administration
Outside
S1
S2
Rack2
Rack3
192.168.3.0/24
.254
L3
ACL2
ACL1
192.168.103.X/24
192.168.101.X/24
192.168.201.X/24
ACL1
L5
.254
192.168.5.0/24
192.168.101.X/24
ACL2
192.168.105.X/24
RDLU:1
192.168.101.65/32
HF:1
192.168.101.66/32
ACCT:1
192.168.103.65/32
CASS:1
192.168.103.66/32
Compute 1
192.168.3.1/24
RDLU:2
192.168.101.65/32
HF:2
192.168.101.66/32
ACCT:2
192.168.105.67/32
CASS:2
192.168.105.68/32
Compute 33
192.168.5.1/24
192.168.201.X/24