



CNET1244BU

NSX-T Design for Pivotal Application Service (PAS)

#vmworld #CNET1244BU

Make
your
mark

Disclaimer

This presentation may contain product features or functionality that are currently under development.

This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.

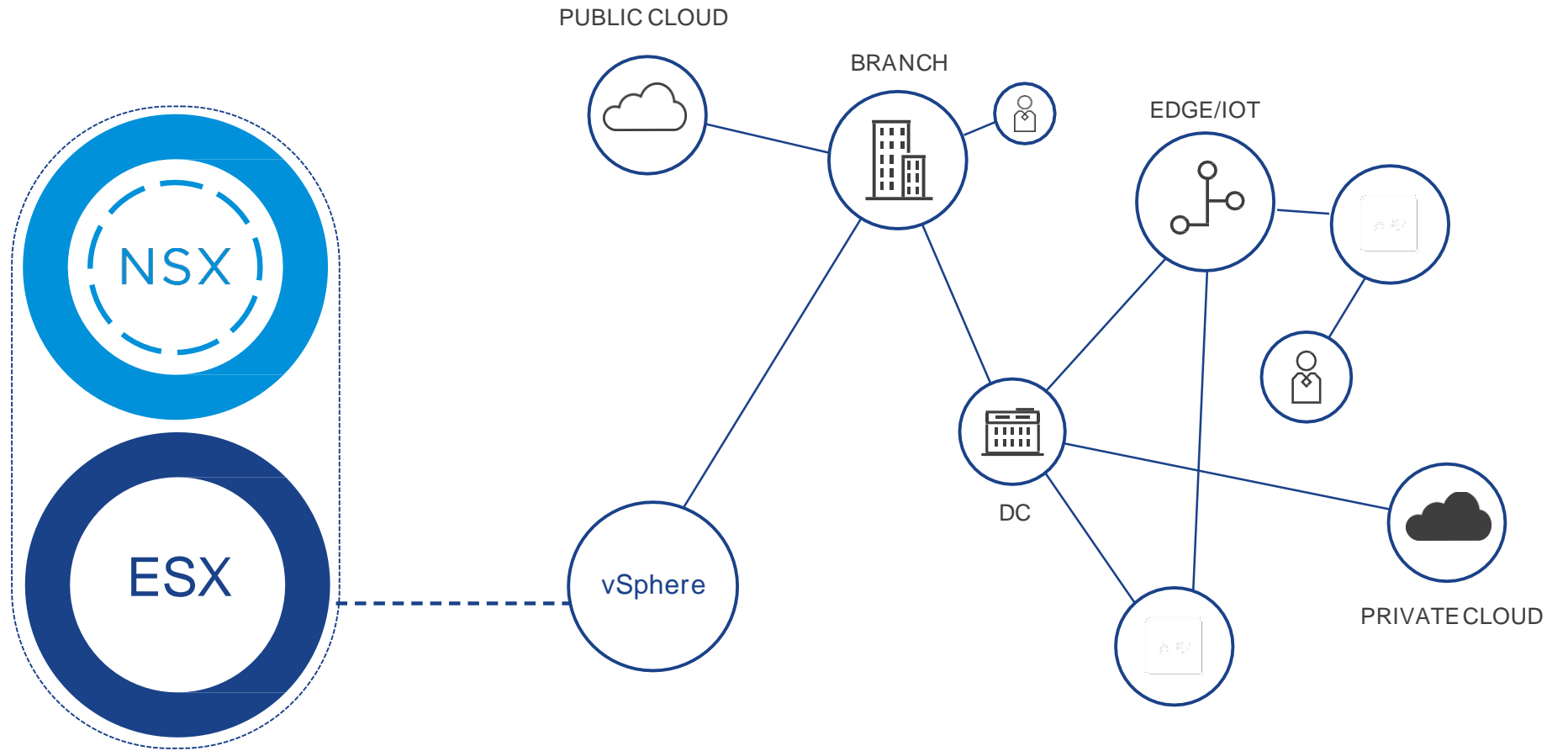
Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

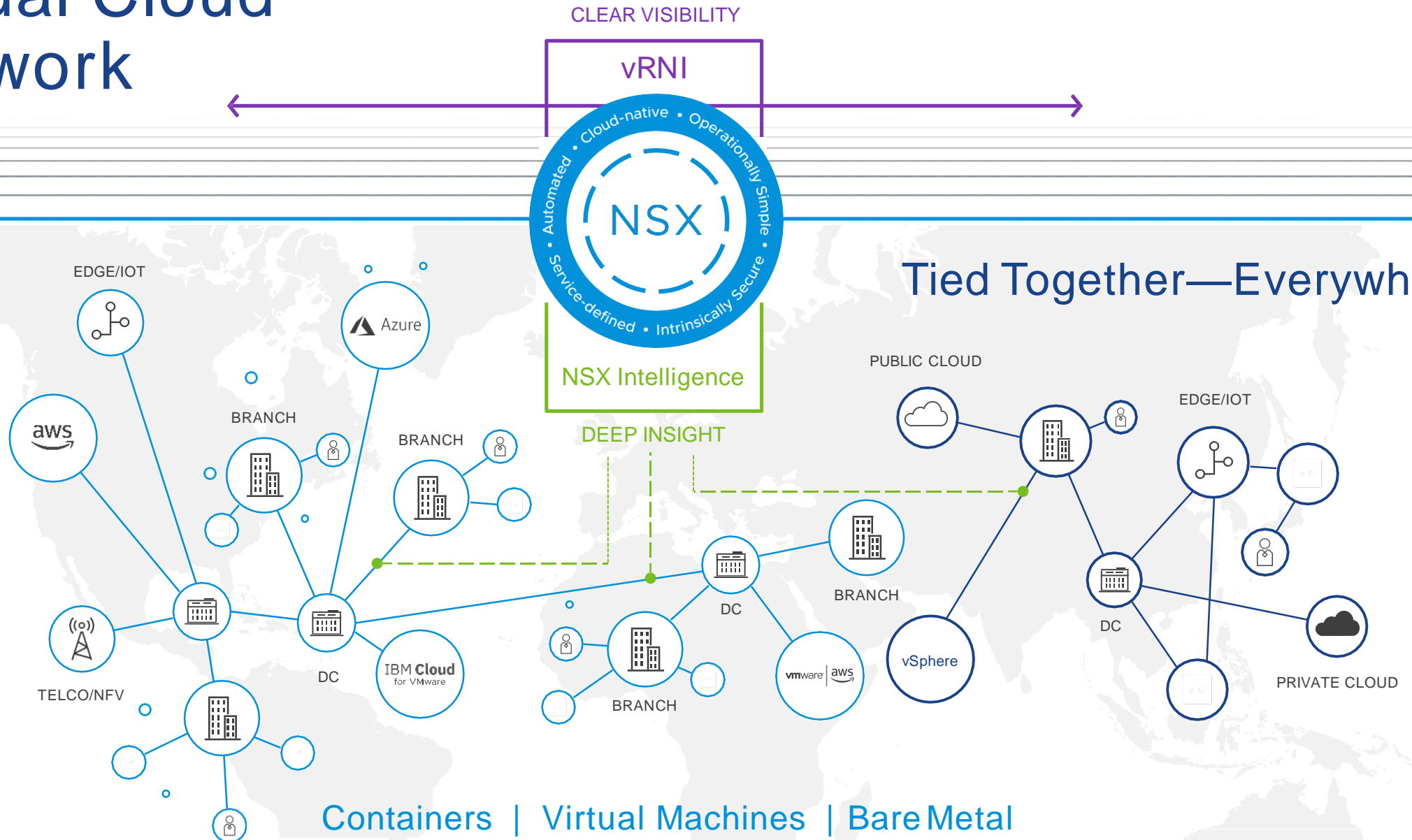
Pricing and packaging for any new features/functionality/technology discussed or presented, have not been determined.

The information in this presentation is for informational purposes only and may not be incorporated into any contract. There is no commitment or obligation to deliver any items presented herein.

NSX Evolution



Virtual Cloud Network



VMware Tanzu Portfolio

Build

Modern Applications

Traditional | COTS | Cloud Native

Run

Enterprise Kubernetes

On-premises | Public Cloud | Edge

Manage

Single Control Point

Multi-cloud
Multi-cluster
Multi-team

PaaS and PKS

PaaS
(PAS)

Developers focus on code.
PAS Platform takes care of the rest.

KaaS
(PKS)

Developer has flexibility of how/what to package code into container.
PKS takes care of running the containers.

IaaS

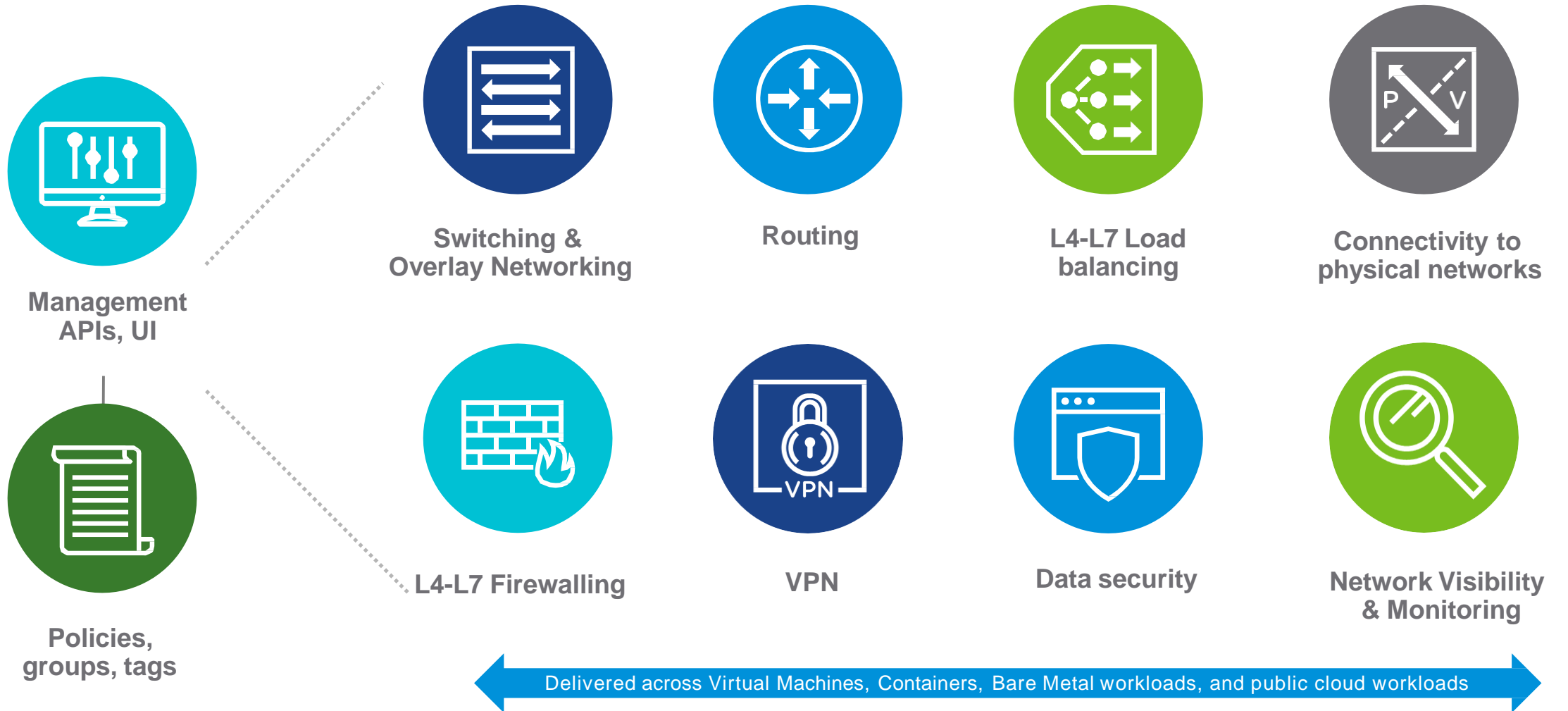
Virtualized Infrastructure

Physical Infrastructure

NSX Architecture

High Level Overview

NSX Data Center



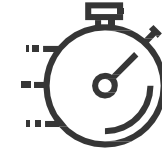
Cloud Native Application Networking



Native container
networking



Any application
framework



Speed of
delivery



Micro-segmentation
for microservices



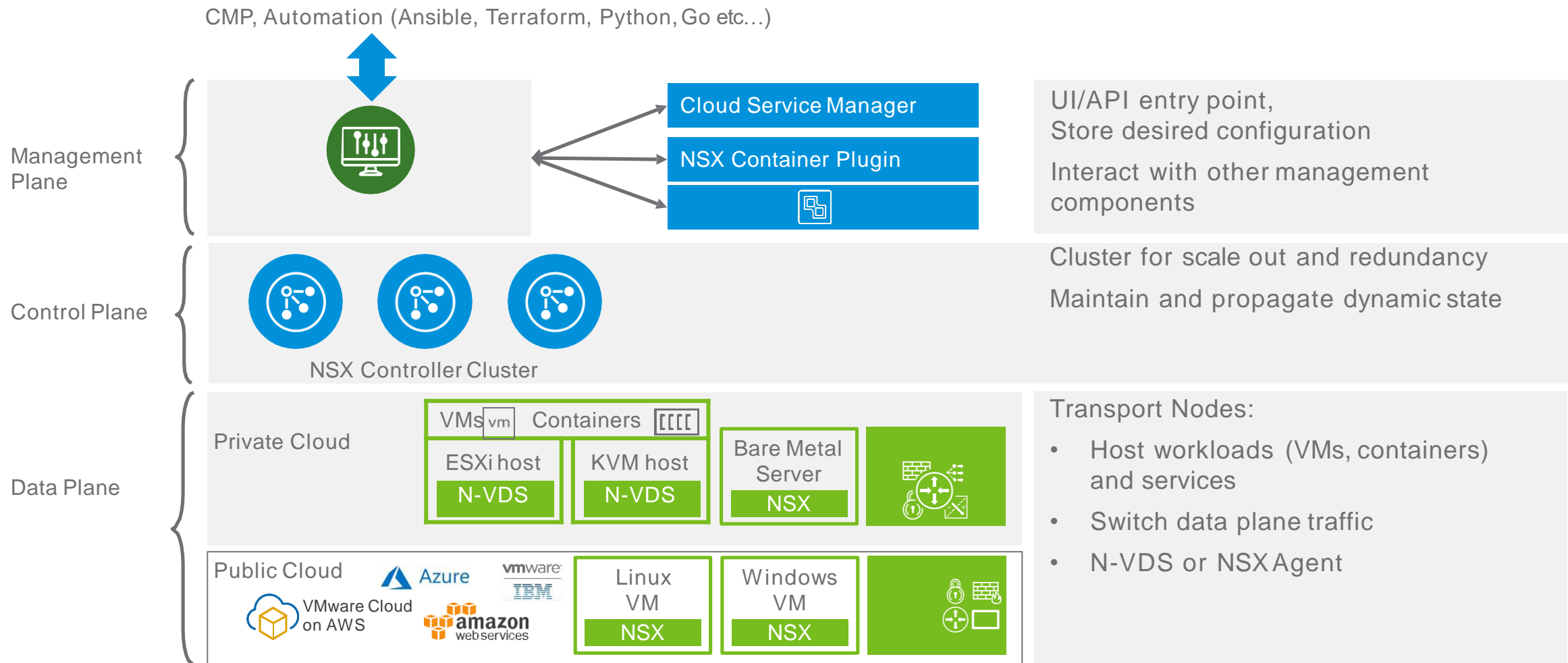
Monitoring and analytics
for microservices



Reference
designs



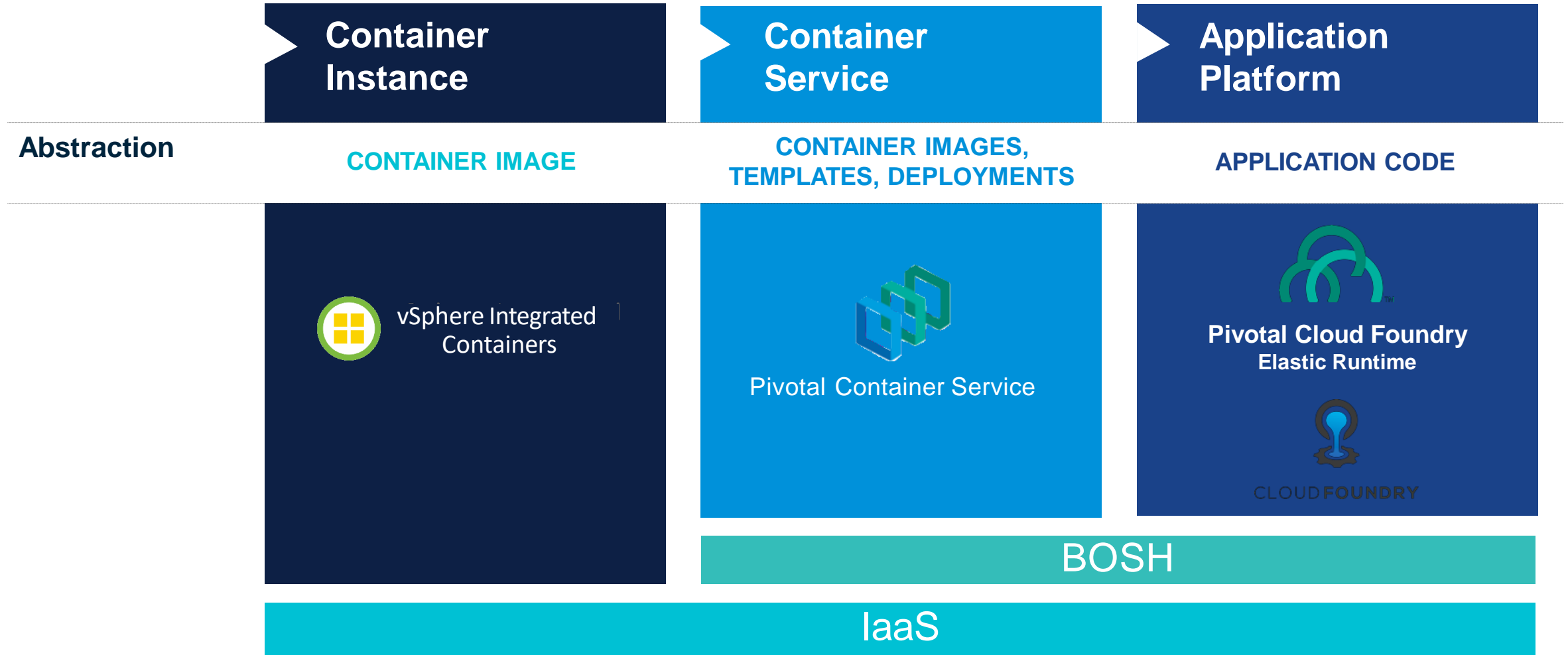
NSX-T Components



PaaS (PAS) and KaaS (PKS)

Intro and Use Cases

Choosing the Right Tool for the Job

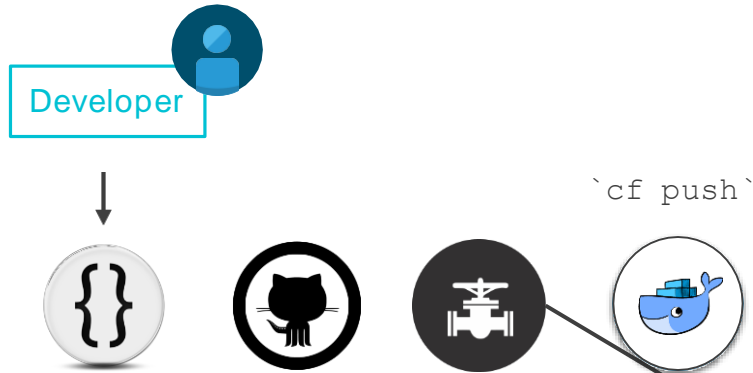


Pivotal Cloud Foundry 101

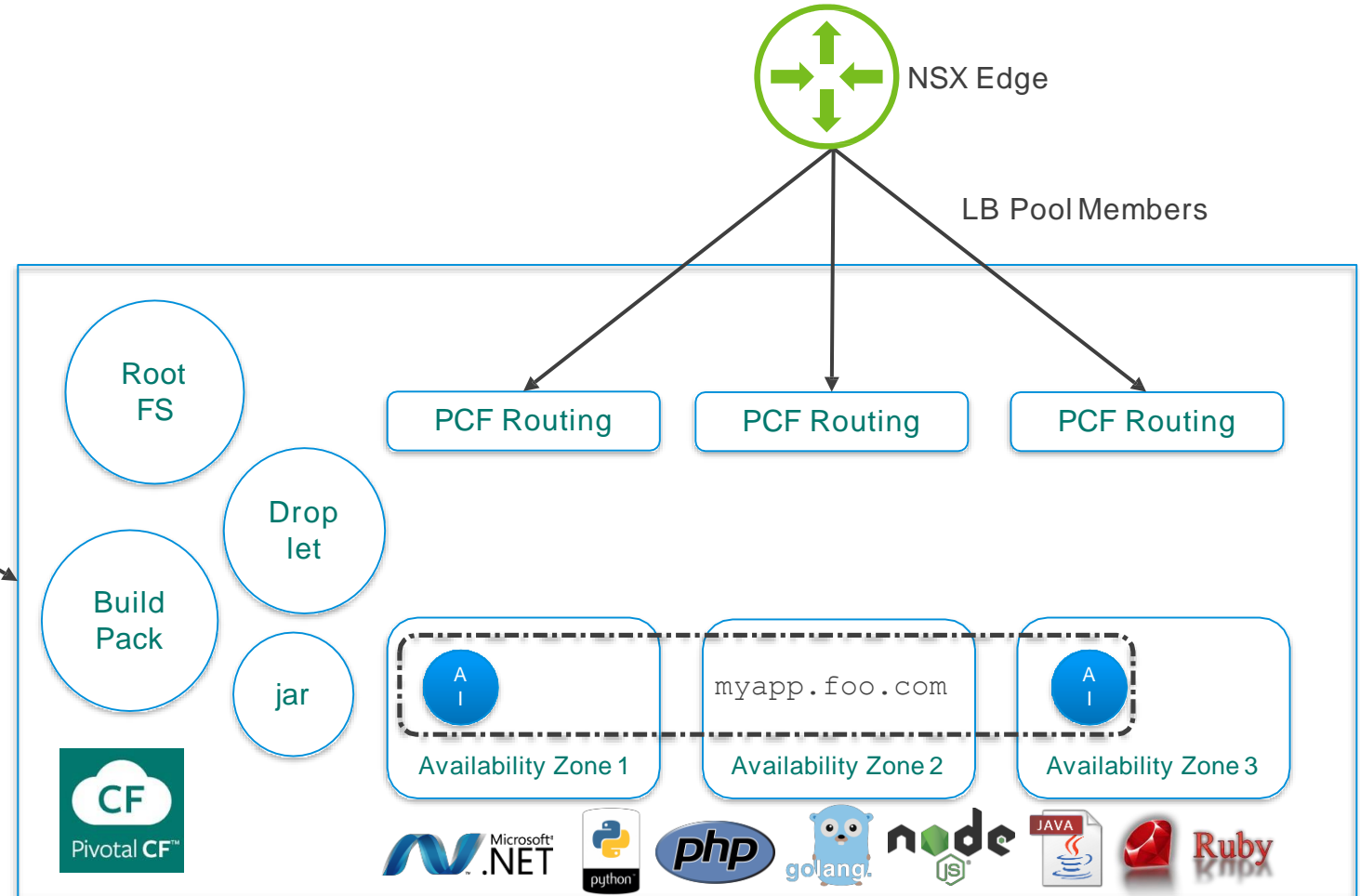
URL Request:
myapp.foo.com



*.foo.com = NSX Edge Vip



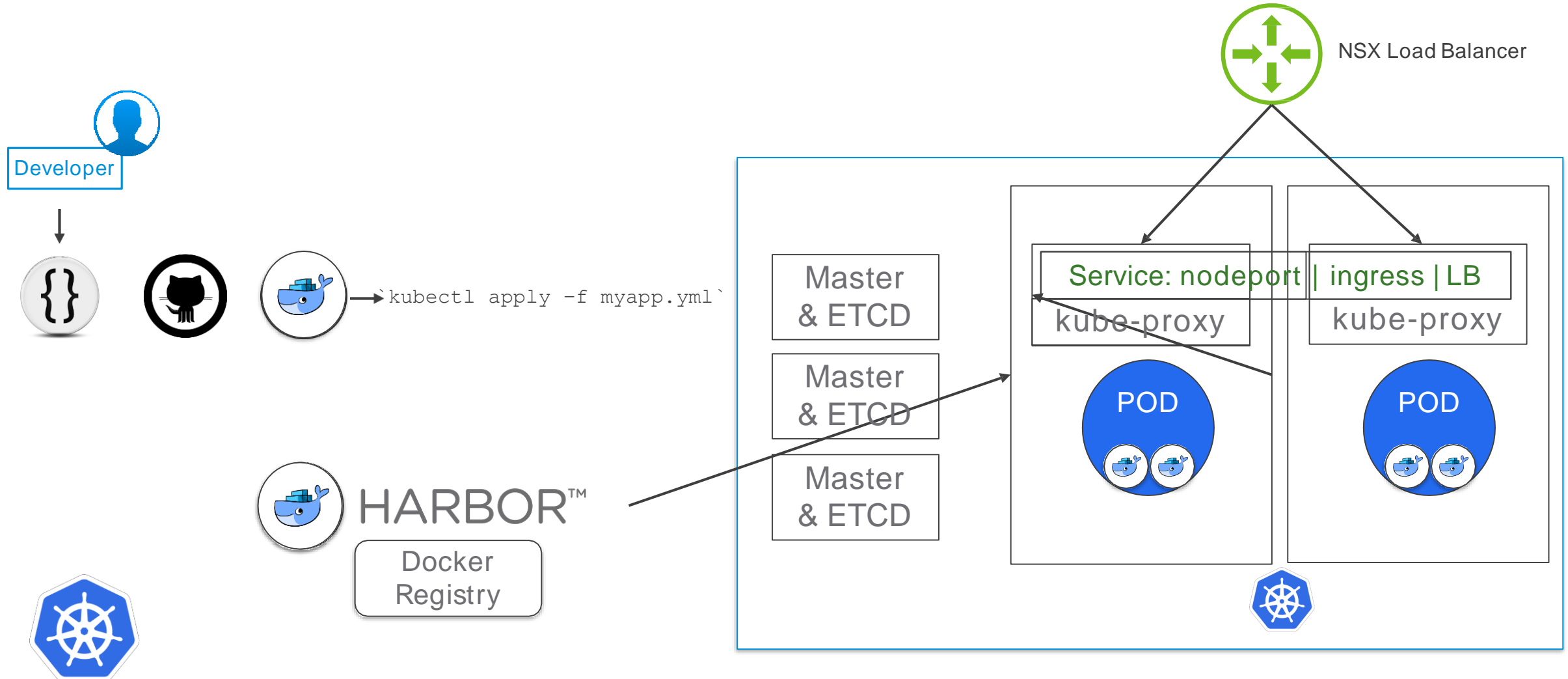
“Here is my source code
Run it on the cloud for me
I do not care how”



Kubernetes 101 (CaaS)

Containers @ Scale

URL Request:
myapp.foo.com/k8siscool



NSX Data Center highlights

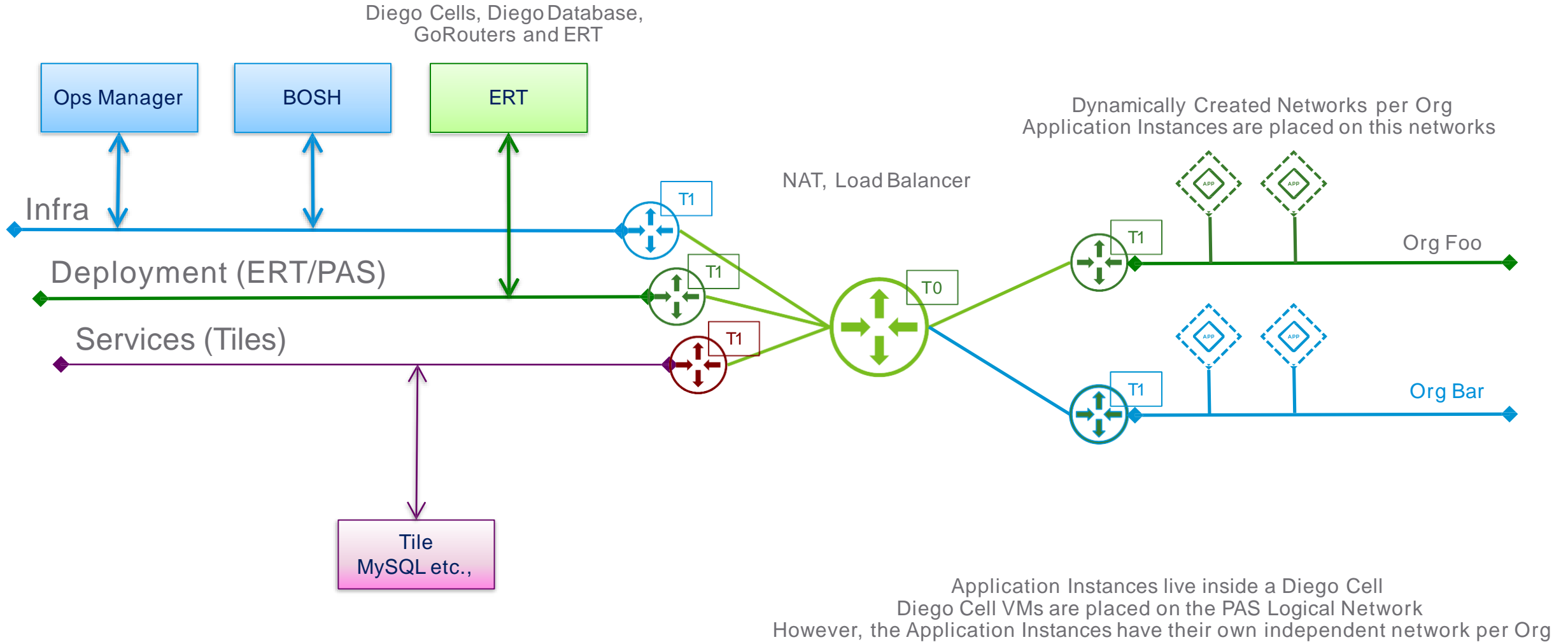
- Heterogeneity
 - PAS, K8S, VMs, Bare metal, Cloud, vSphere, KVM all in a single unified solution
- Relatability
 - Can filter traffic with physical FW/IPS on tenant level
 - Can pinpoint traffic to specific container
- Visibility
 - Monitoring and logging tools that are up to standard of DevOps/Security teams
- Programmability
 - NSX Data Center can be fully automated
 - Automated by native integration with the CNA platforms using the CNI
 - Allows integration into DevOps non disruptively

PAS Integration with NSX-T

How does it work

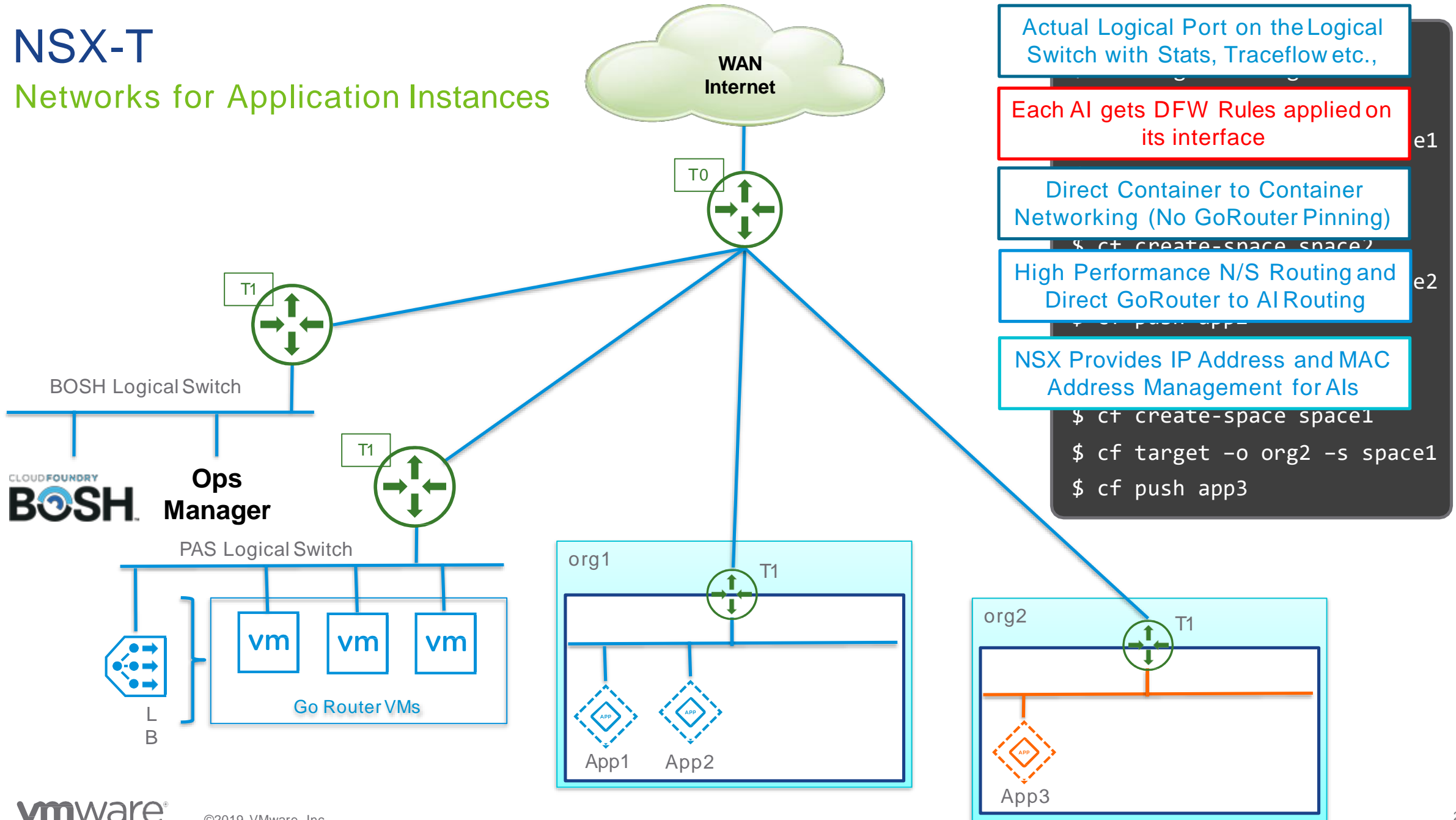
PAS

Network and Security Requirements



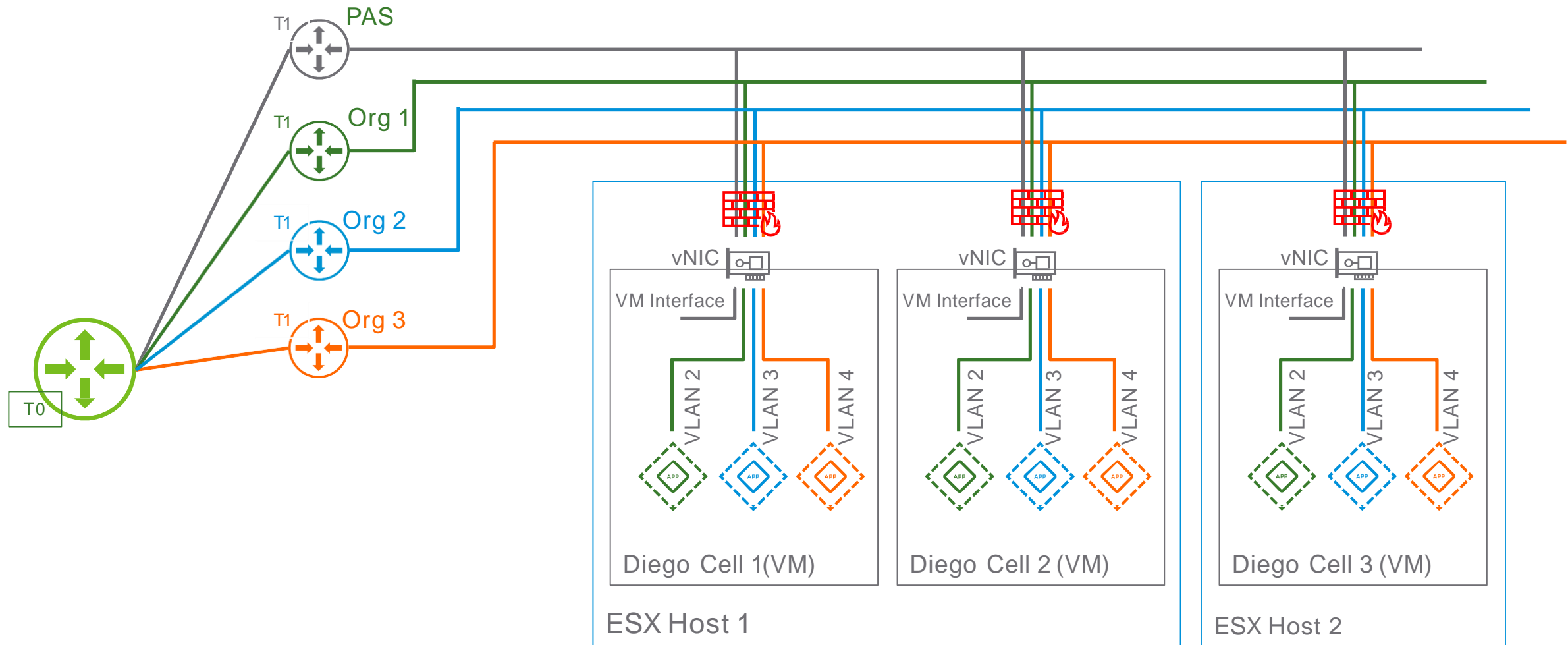
NSX-T

Networks for Application Instances



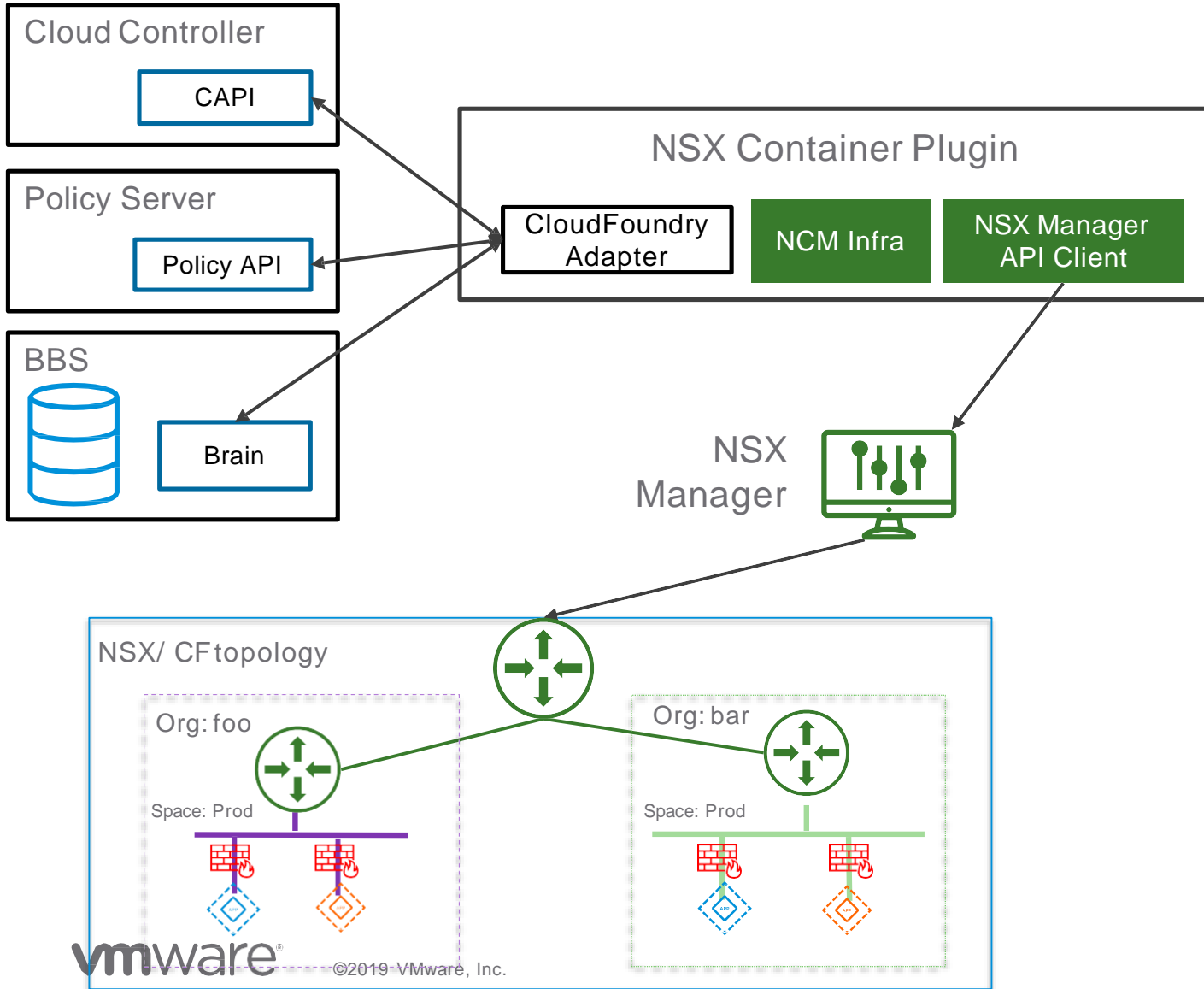
PAS on NSX-T

Network for the Application Instances



CF / NSX Components

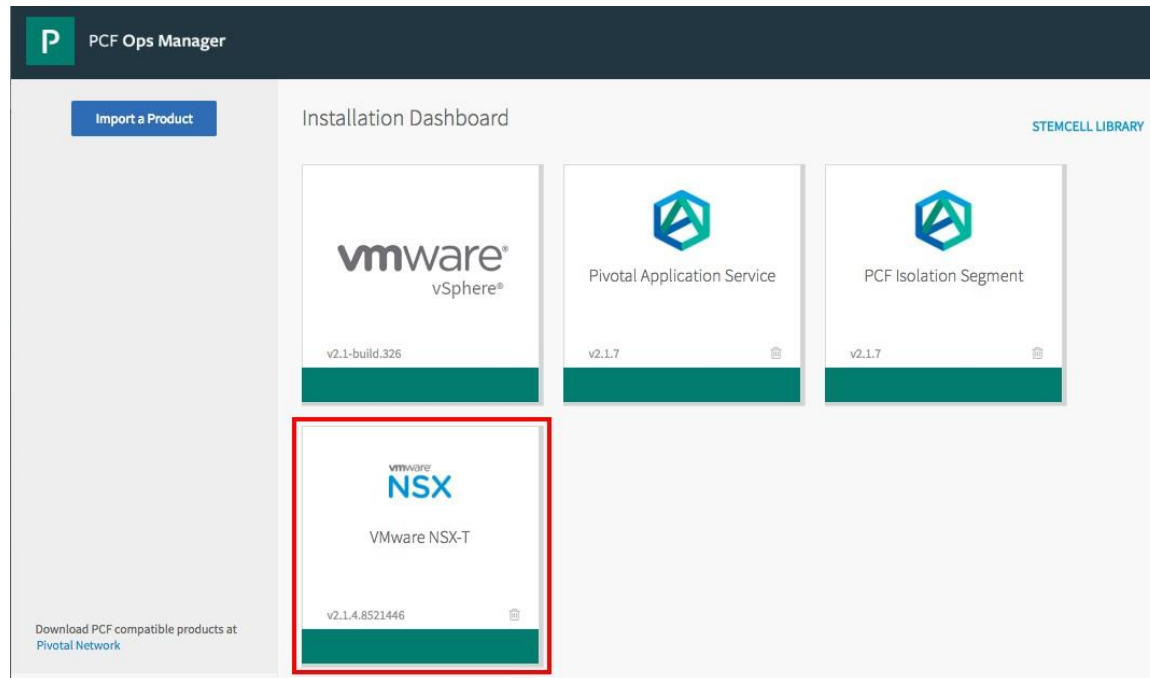
Network Container Plugin (NCP)



- NSX Container Plugin: NCP is a software component provided by VMware in form of a BOSH add-on release. It is deployed as a pair of HA VMs as part of the ERT (using a Ops Manager Tile)
- Adapter layer: NCP is built in a modular way, so that individual adapters can be added for different CaaS and PaaS systems
- NSX Infra layer: Implements the logic that creates topologies, attaches logical ports, etc. based on triggers from the Adapter layer
- NSX API Client: Implements a standardized interface to the NSX API

PAS on NSX-T

Integration Intro








On Diego Cell:

- Following NSX processes are installed:
 - ovssdb-server
 - ovs-vswitchd
 - NSX-Node-Agent
- Following silk processes are removed from diego cells:
 - iptables-logger
 - netmon
 - silk-daemon
 - vxlan-policy-agent

NSX Container Plugin (NCP) deployed as a pair of HA VMs as part of PAS

NSX Container Plugin (NCP)

Auto-provisioning NSX-T Data Center Objects on PAS/PKS Operations

	NSX-T Data Center	PAS (On creating a new org) Runtime (Garden)	PKS (On creating a new K8s Cluster) Runtime (K8s)
	Logical Switches	1per each Org	3 (1x for namespace, 1x for LB, 1x for Masters/Workers)
	Logical Router (T1)	1per each Org	3 (1x for namespace, 1x for LB, 1x for Masters/Workers)
	NSX-T LB	None (Manually add links to GoRouter)	1NSX-T LB 1x Virtual Server for K8s Control Plane API 1x Server Pool containing 3 K8s MasterNodes 1x Virtual Server for Ingress Controllers (HTTP) 1x Virtual Server for Ingress Controllers (HTTPS) Allocate IP from Floating IP Pool
	Distributed Firewall	1x Rule for Org	1x Rule for kubernetes dashboard 1x Rule for Kube-dns
	NSX-T DDI/IPAM	1x /24 from Nodes IP Block for Als	1x /24 from Nodes IP Block (for Master and Worker nodes) 1x /24 from Pods IP Block for K8s namespace
	NAT	1x SNAT for the Org	1x SNAT for the K8s namespace

DFW Rules are auto-generated on applying any security policy via “cf network-policy” or K8s apply

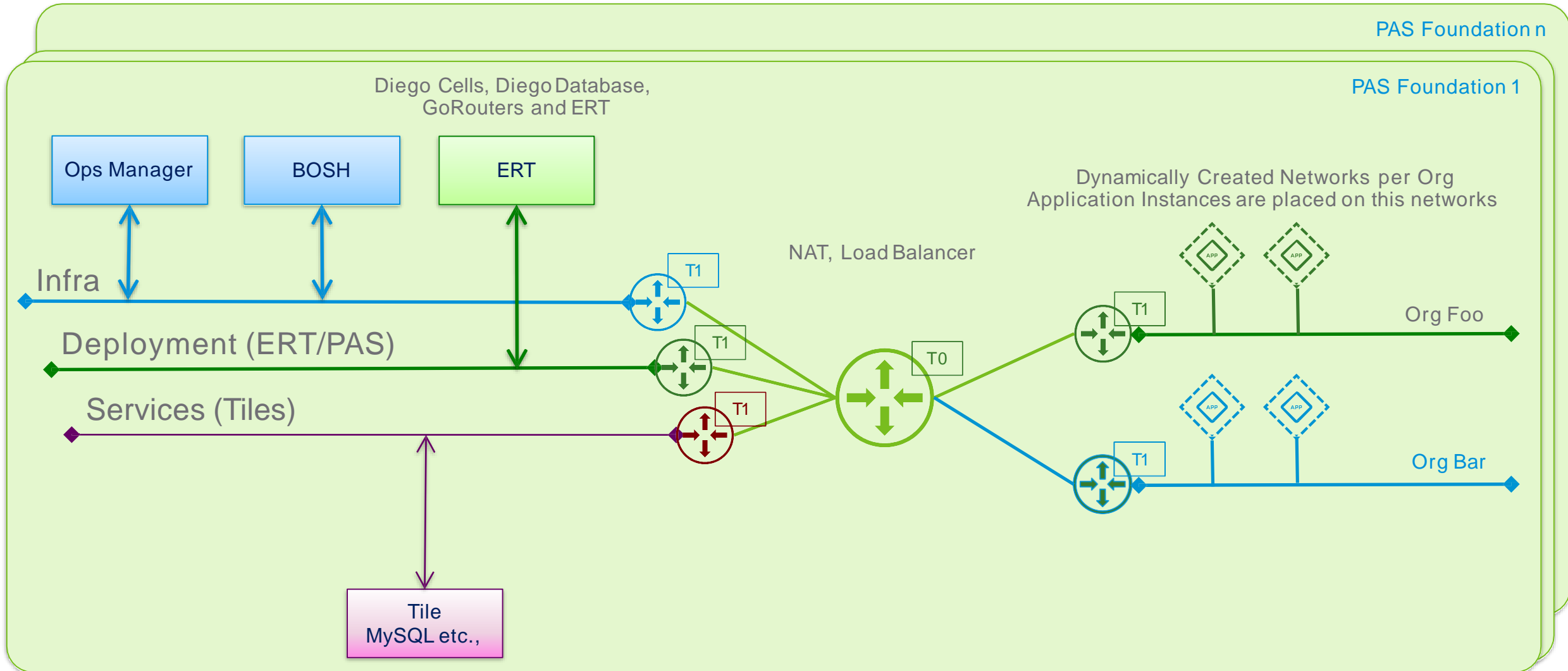
NCP also auto-deletes NSX-T Data Center objects – when the corresponding objects are deleted in PAS / PKS

PAS/PKS Integration with NSX-T

Logical Design Options

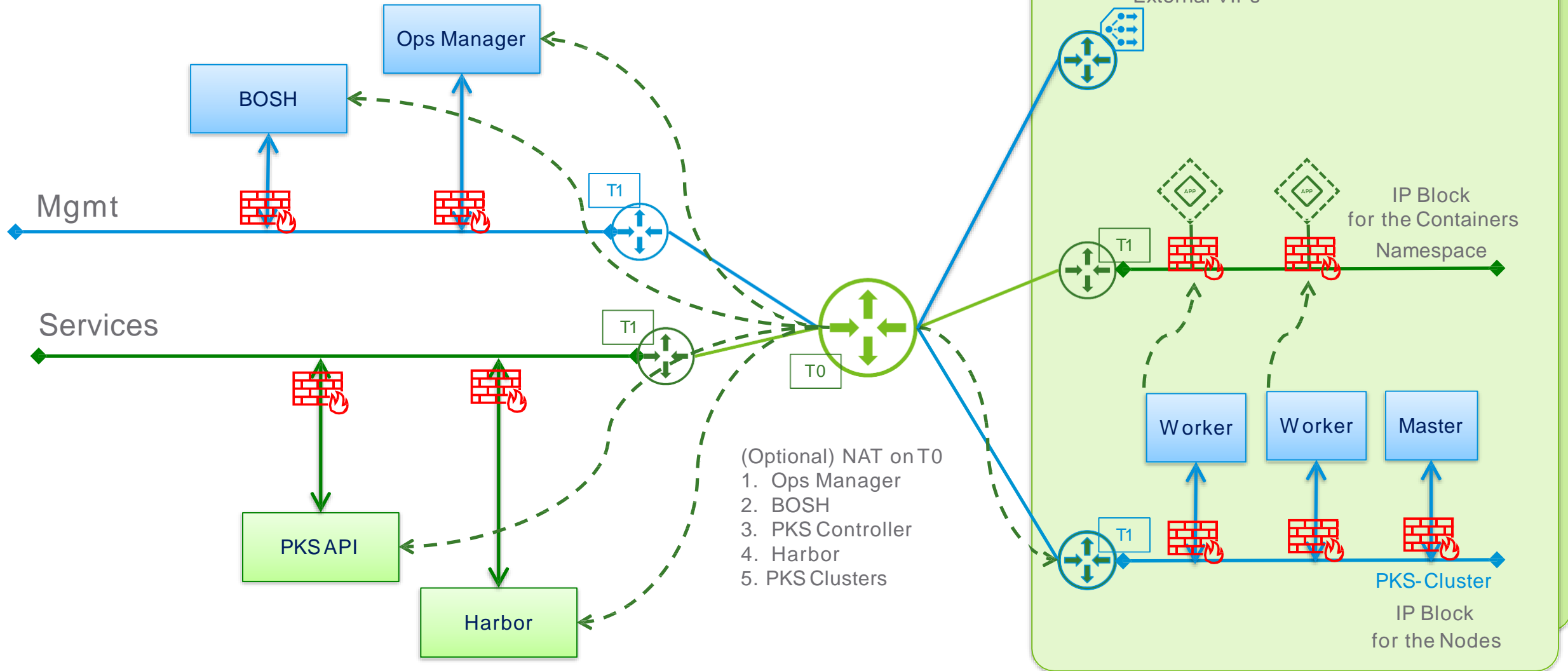
PAS on NSX-T

Network and Security Components



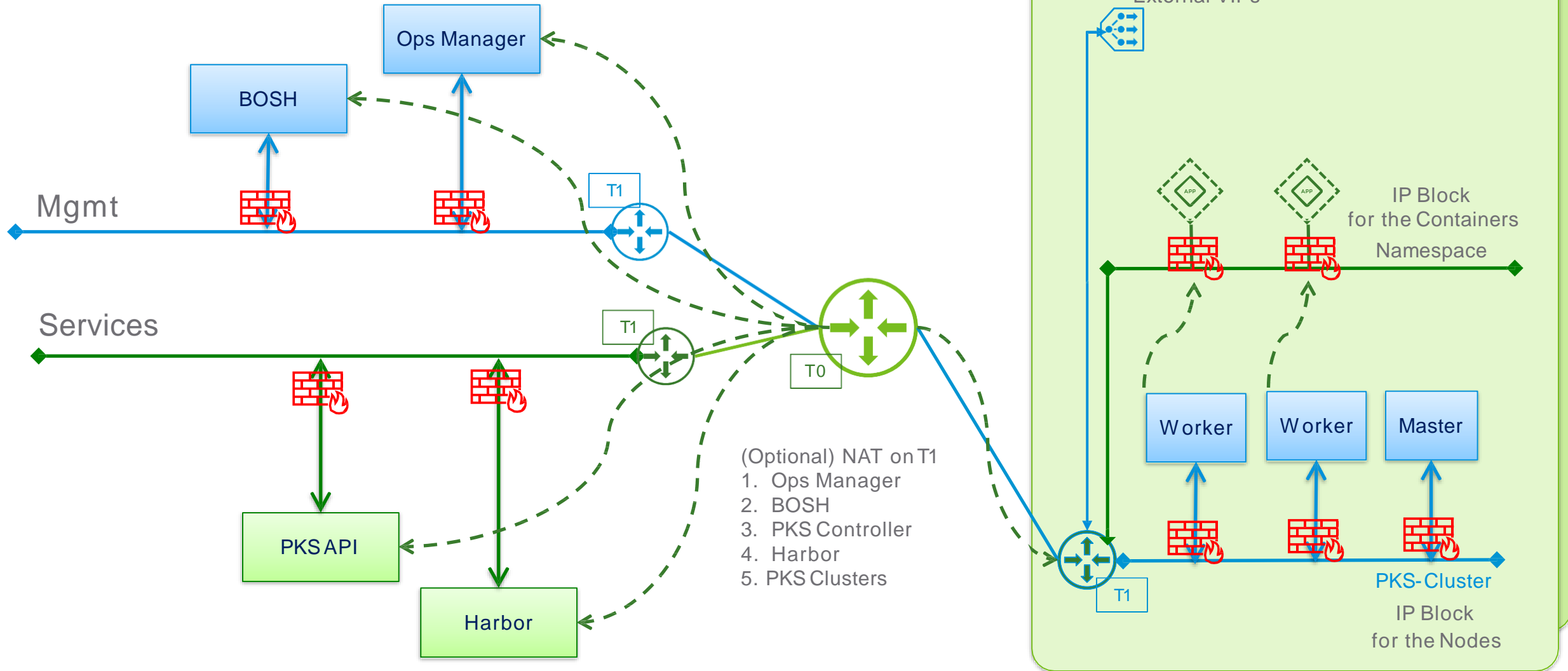
PKS on NSX-T (Dedicated T1)

Network and Security Components



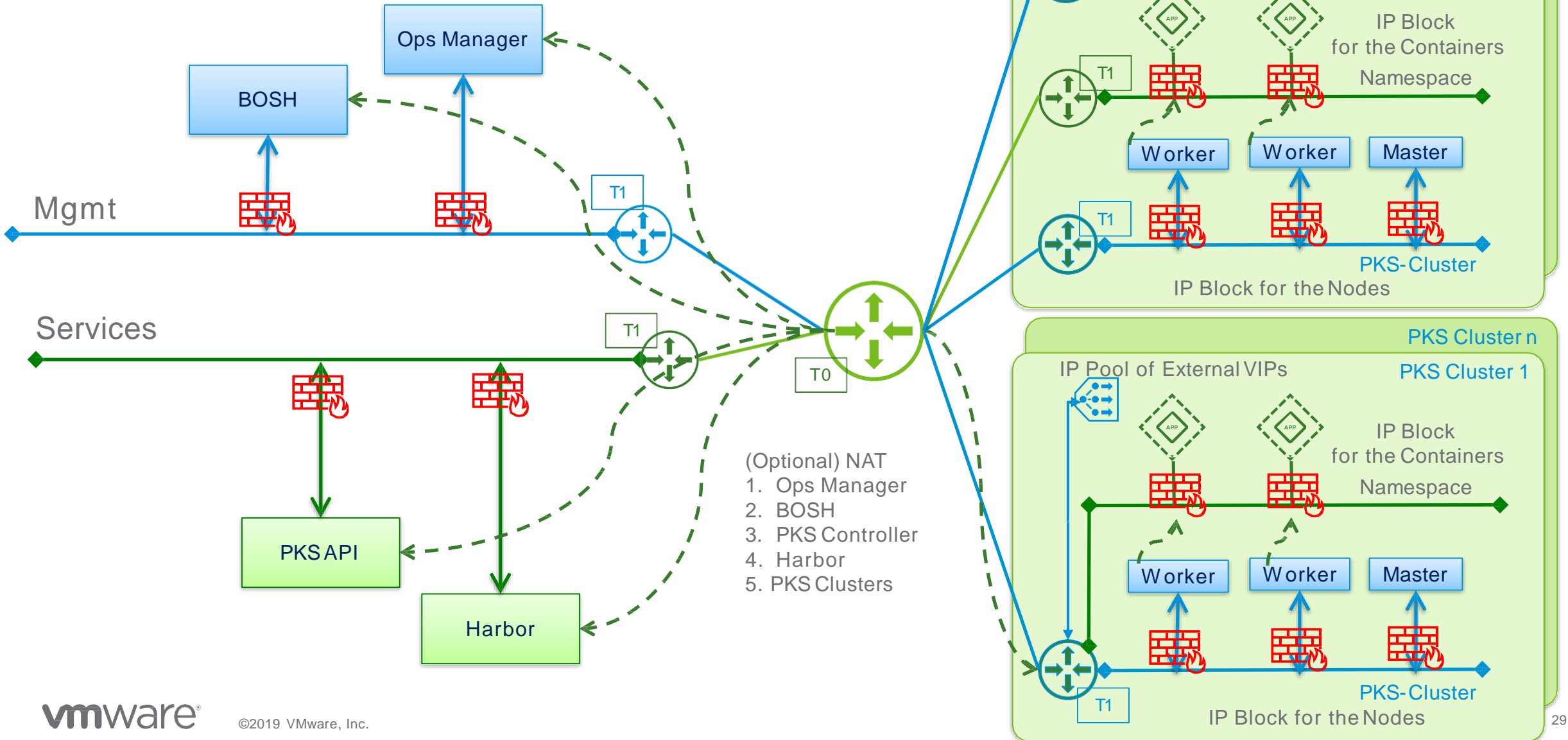
PKS on NSX-T (Shared T1)

Network and Security Components



PKS on NSX-T (Shared T1)

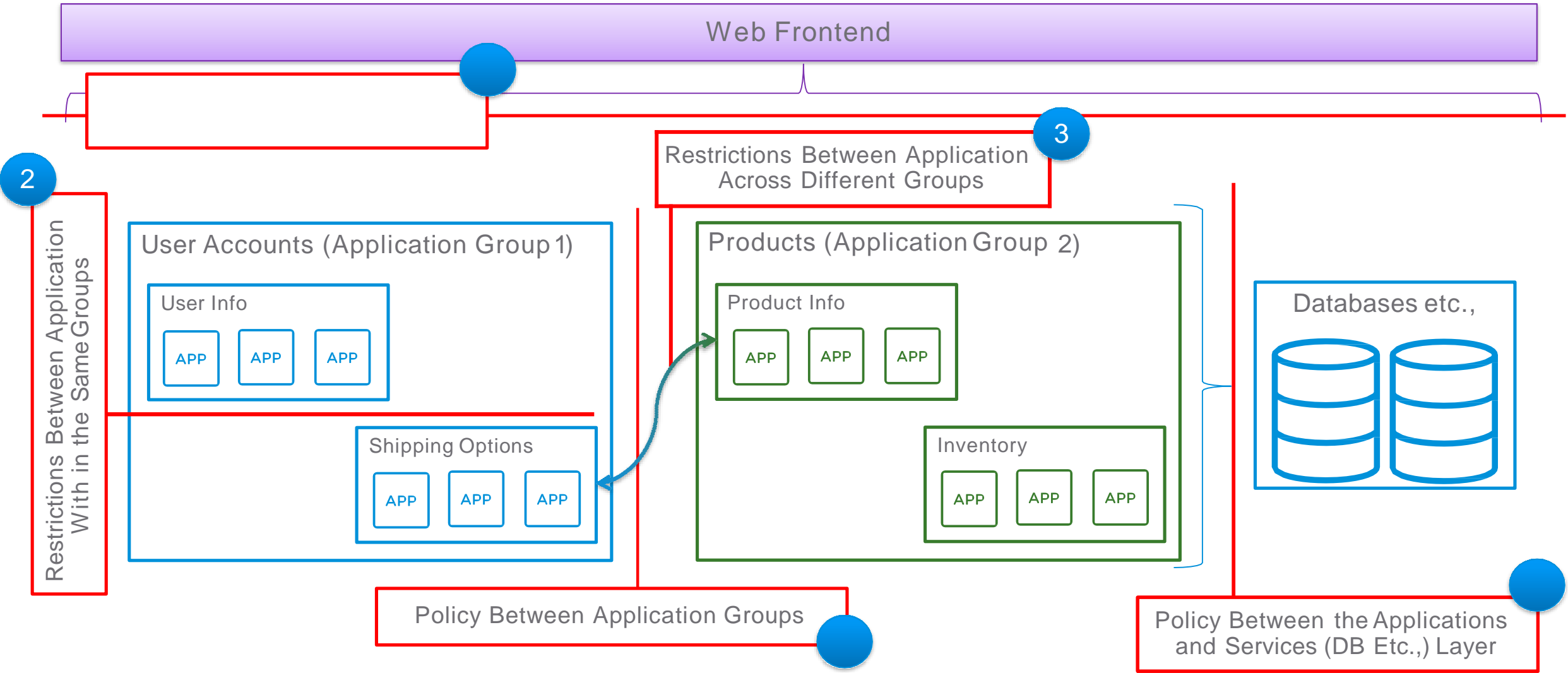
Network and Security Components



Security Considerations

NSX-T Data Center Distributed Firewall

Application Level



Application Context

Dynamically created tags: identify the application and its placement within a foundation

The image shows a screenshot of the VMware vCenter interface. On the left, a table lists various tags for an application. On the right, a detailed view of the 'secOps-pas-fd1-product-info' membership criteria is shown. Arrows point from specific rows in the table to explanatory text boxes.

Scope	Tag
ncp/version	1.0.0
ncp/cluster	pas-fd1
ncp/cf_org_guid	65d4ca29-fdf6-4b1f-a3e9-65d33c90c09e
ncp/cf_org_name	online-store
ncp/cf_space_guid	6cc51b1e-c689-49b2-85de-4f37c293c19a
ncp/cf_space_name	product-app-group
ncp/cf_app_guid	15a73bba-b866-43ce-949a-17dc744102fa
ncp/cf_app_name	product-info
ncp/cf_instance_id	788b666f-244c-4fad-796d-fcbe
ncp/cf_phase	running
ncp/cf_index	0

secOps-pas-fd1-product-info

Overview **Membership Criteria** Members Ap

Membership Criteria | [EDIT](#)

1. Logical Port

- Tag Equals pas-fd1
- Scope Equals ncp/cluster
- Tag Equals product-info
- Scope Equals ncp/cf_app_name

Tags for staging security posture before applications are actually deployed

Tags for security posture after applications are deployed – used by cf network policy

Application Context

Dynamically created tags: identify the application and its placement within a foundation

```
# kubectl get pod --show-labels
```

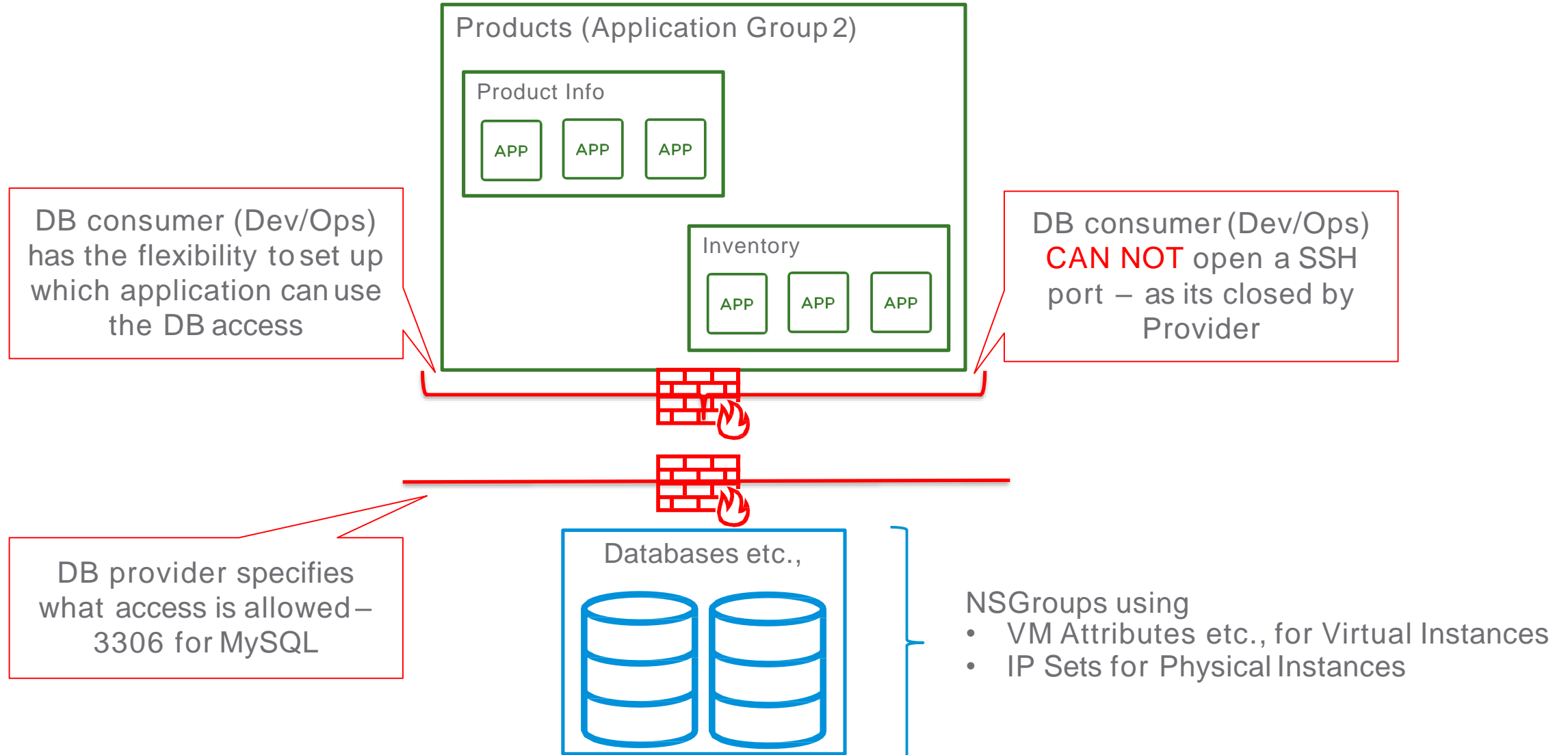
```
NAME                READY STATUS  RESTARTS AGE LABELS
wordpress-77f8677bb8-qq9jw  1/1 Running 0    45h app=wordpress,pod-template-hash=77f8677bb8,tier=frontend
```

▼ **Tags** | [MANAGE](#)

Tag	Scope
1.2.0	ncp/version
pks-a5864a15-6481-4c0c-b142-e374c2eca4ab	ncp/cluster
default	ncp/project
wordpress-77f8677bb8-qq9jw	ncp/pod
a536be10-c837-11e9-899f-00505680edd8	ncp/pod_uid
frontend	tier
wordpress	app

NSGroups

For resources external to PAS



Firewall Sections

NCP Section Placement

Top Firewall Section Marker

ncp-top

Bottom Firewall Section Marker

ncp-bottom

The screenshot shows the VMware NSX Firewall configuration interface. The left sidebar contains a navigation menu with the following items: Dashboard, Getting Started, Tools, Load Balancing, Firewall (highlighted), Encryption, Routing, and DDI. The main content area is titled 'General' and displays a table of firewall sections. The table has columns for #, Name, ID, Sources, Destinations, Services, Action, Applied To, Log, and Stats. The sections listed are:

#	Name	ID	Sources	Destinations	Services	Action	Applied To	Log	Stats
+	SecOps-Rules	70926b8a-d3a2-4baa-...		Stateful	Applied To: 4				2 Rules
	DB-Provider	374af3f1-94db-49f2-9...		Stateful	Applied To: 0				0 Rules
+	ncp-top	7906814c-7f59-4800-...		Stateful	Applied To: 0				0 Rules
+	np-pas-fd1-inventory	361232df-641a-49d9-9...		Stateful	Applied To: 0				1 Rule

Micro-Segmentation Granularity

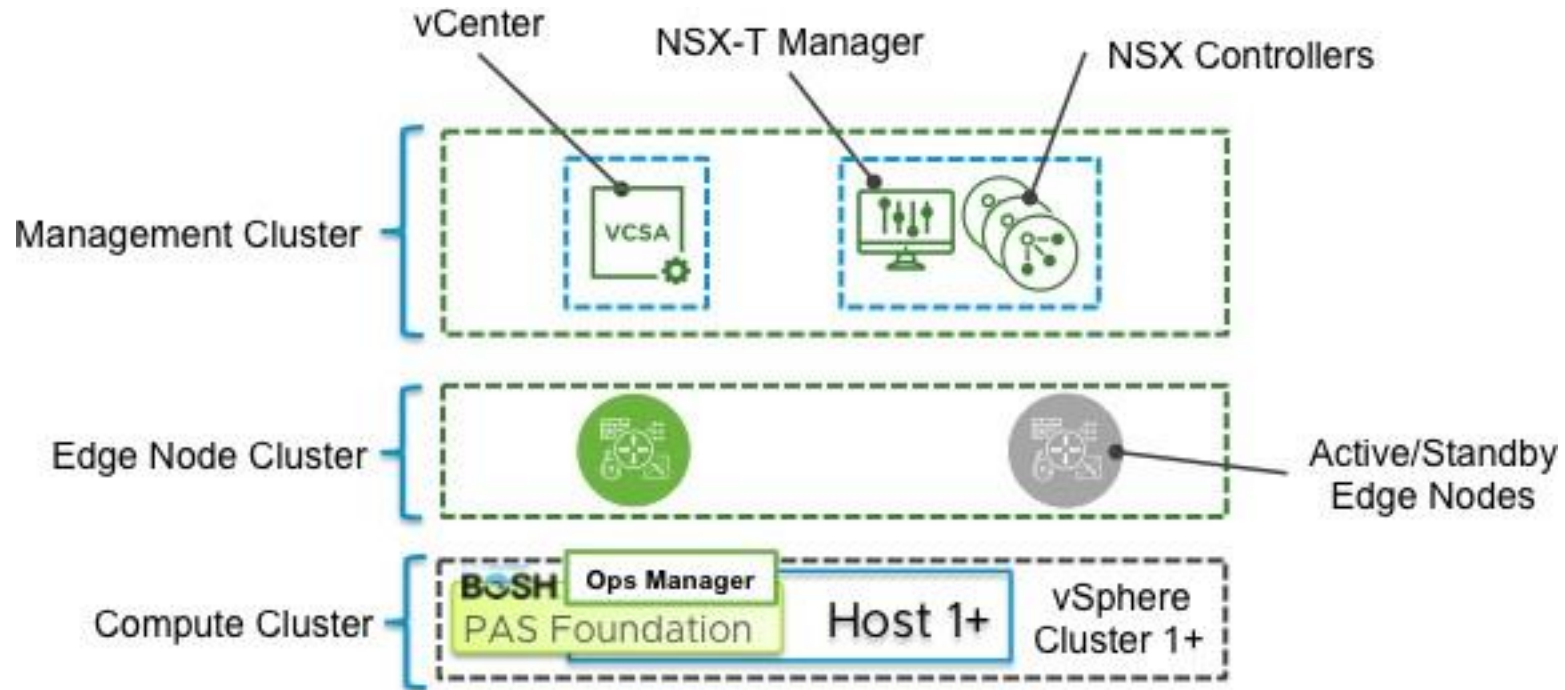
Coarse to fine-grained ...

Granularity	Purpose	Pros/Cons
Foundation level / Instance Level	<ol style="list-style-type: none">1. Allow Foundation level access to Databases2. Allow access between Foundations3. Allow access between Foundations and other external resources	Coarse security posture
Org/Space Level	<ol style="list-style-type: none">1. Org level access to external resources such as Databases2. Allowing access between intra and inter foundation Orgs	Fine grained security posture while maintaining operational simplicity

Design Considerations

PAS Design

Building Blocks



PAS Foundation

- BOSH, Ops Manager and
- Go Routers, Diego Cells, AI, etc.,

Three+ vSphere Clusters

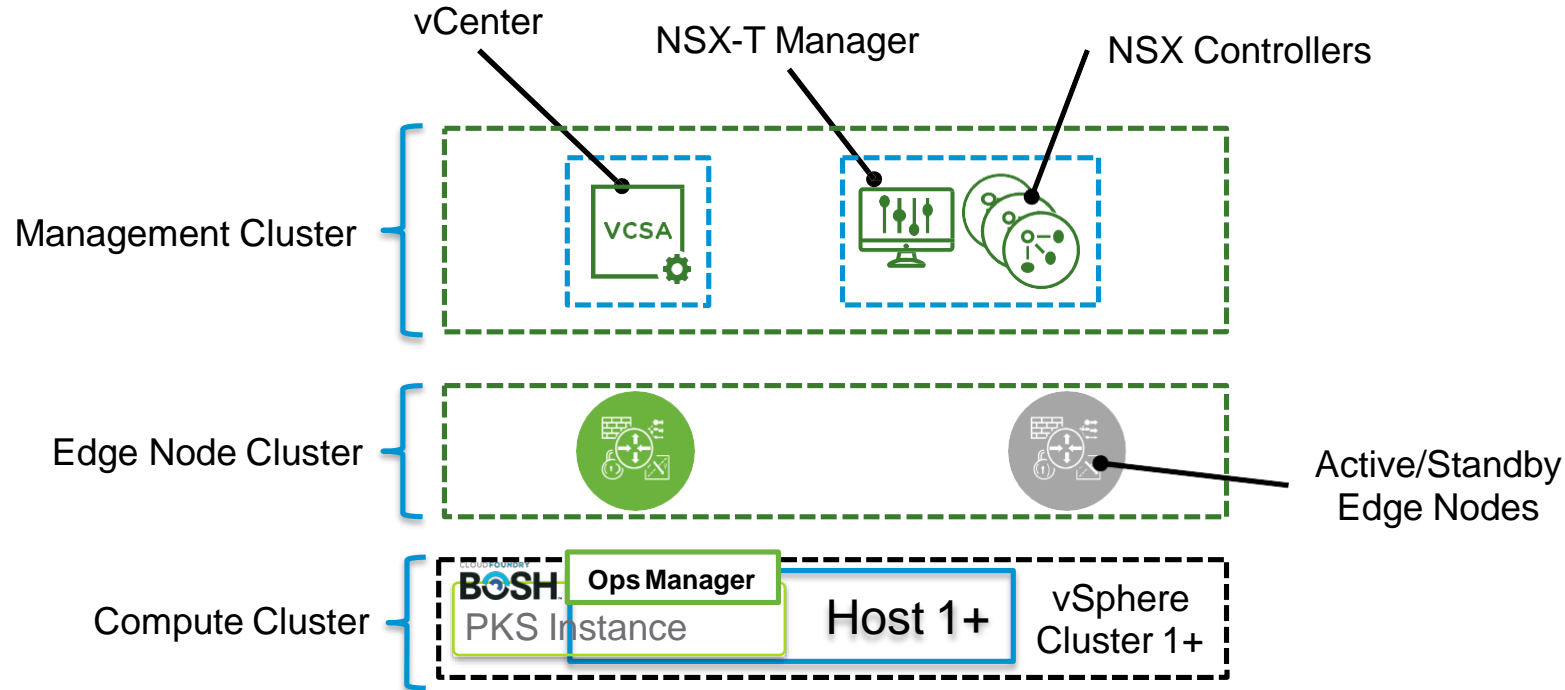
- Dedicated Management Cluster
- Dedicated Edge Cluster
- 1or more Compute Clusters
 - Ideally 3+ Compute Clusters

Other Options

- Collapse Edge and Management Cluster
- Collapse Edge and Compute Cluster

PKS Design

Building Blocks



PKS Instance

- BOSH, Ops Manager, PKS API, Harbor etc., and
- Masters, Worker and PODs

Three+ vSphere Clusters

- Dedicated Management Cluster
- Dedicated Edge Cluster
- 1or more Compute Clusters
 - Ideally 3+ Compute Clusters

Other Options

- Collapse Edge and Management Cluster
- Collapse Edge and Compute Cluster

vSphere Cluster Considerations

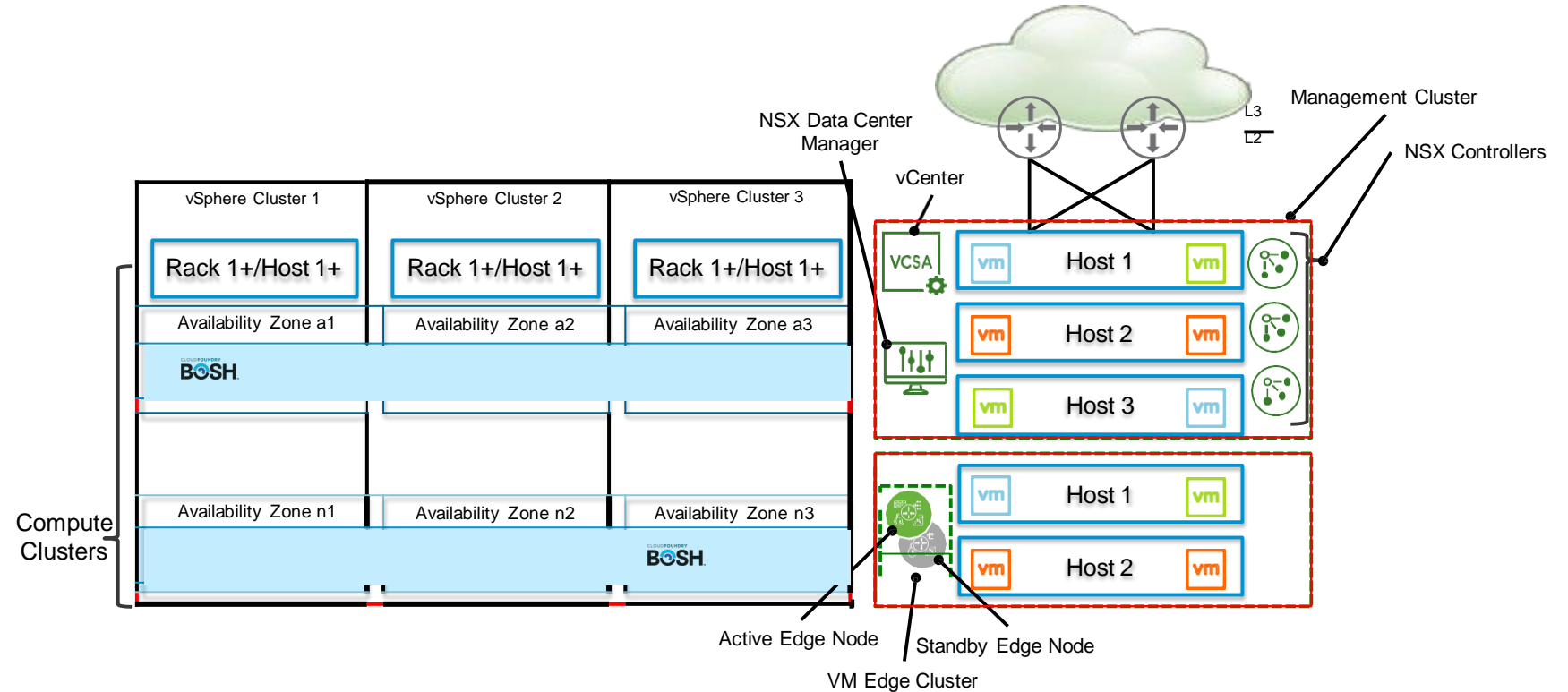
Sharing NSX-T across multiple PAS Foundations

vSphere Cluster Design

- Clusters are not striped across racks
- This is specific to PAS workloads only
- Not recommended when mixing PAS with other traditional workloads on the same cluster

Separate cluster for

- Management (NSX, vCenter)
- Edge Nodes



vSphere Cluster Considerations

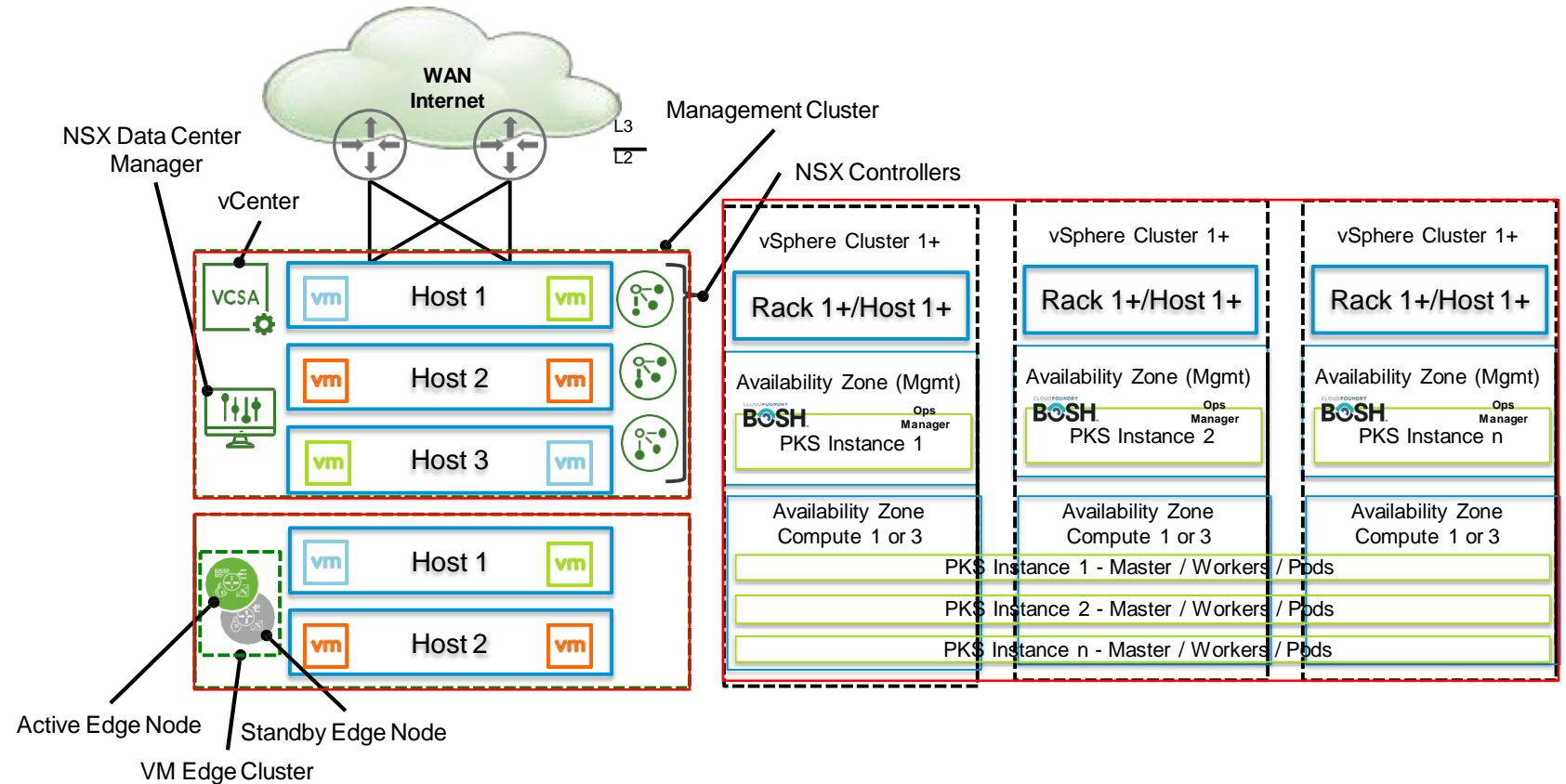
Sharing NSX-T across multiple PKS Instances

vSphere Cluster Design

- Clusters are not striped across racks
- This is specific to PAS & PKS workloads only
- Not recommended when mixing PAS & PKS with other traditional workloads on the same cluster

Separate cluster for

- Management (NSX, vCenter)
- Edge Nodes



vSphere Cluster Considerations

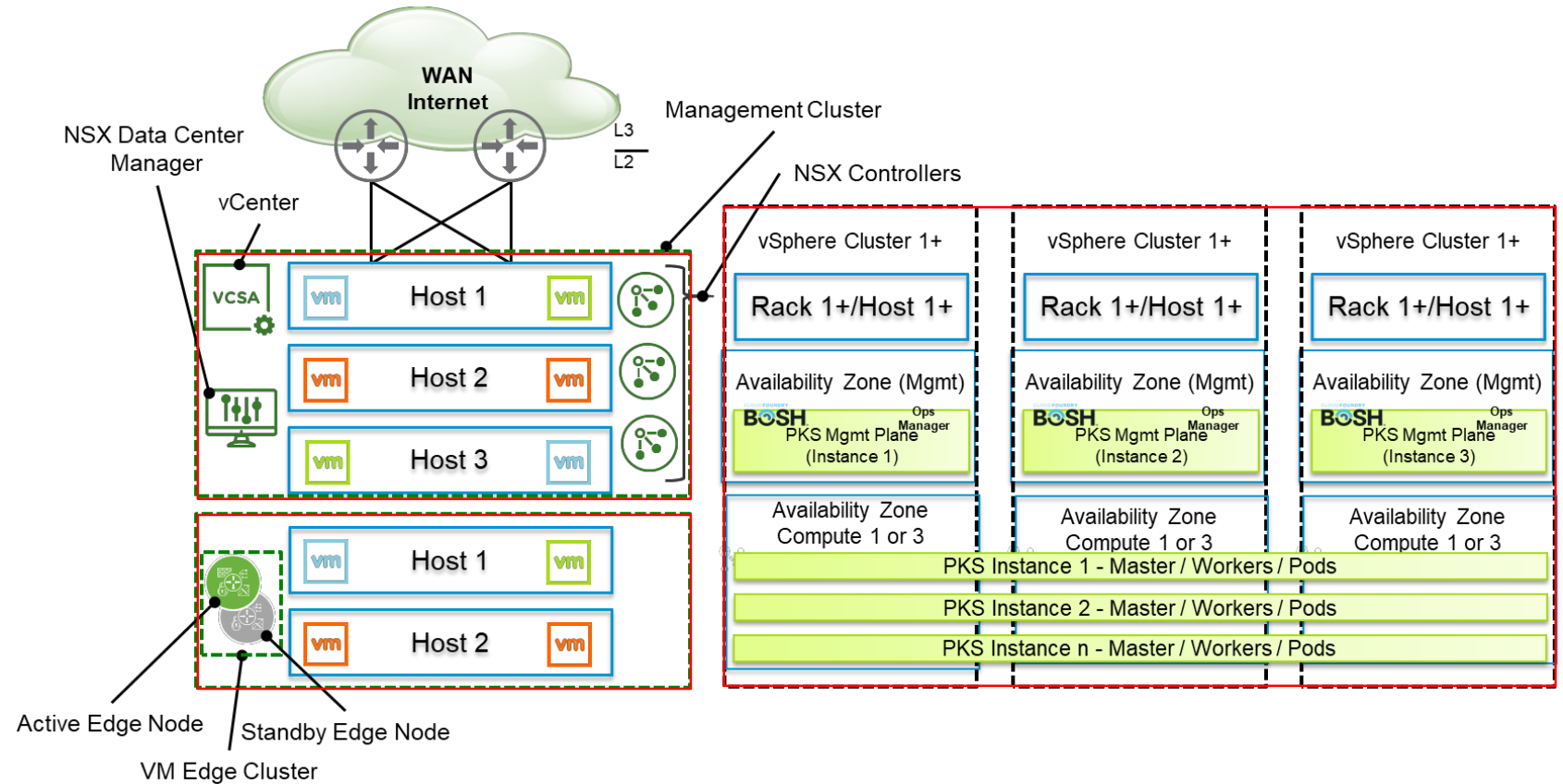
Sharing NSX-T across multiple PKS Instances

vSphere Cluster Design

- Clusters are not striped across racks
- This is specific to PAS & PKS workloads only
- Not recommended when mixing PAS & PKS with other traditional workloads on the same cluster

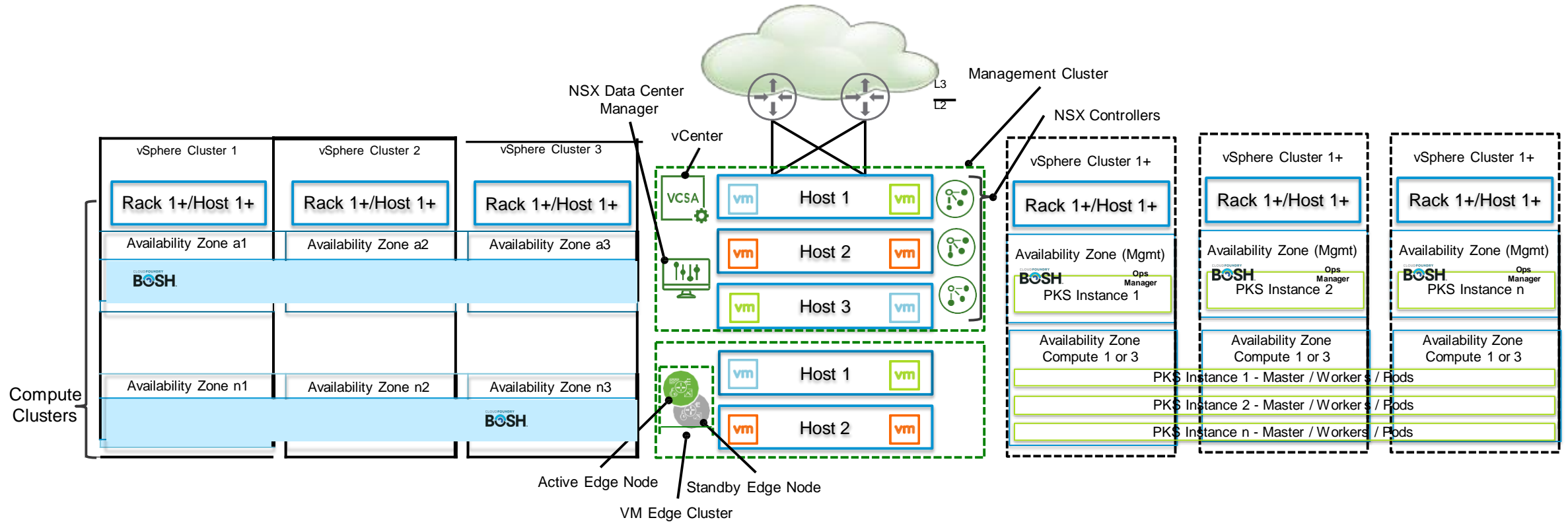
Separate cluster for

- Management (NSX, vCenter)
- Edge Nodes



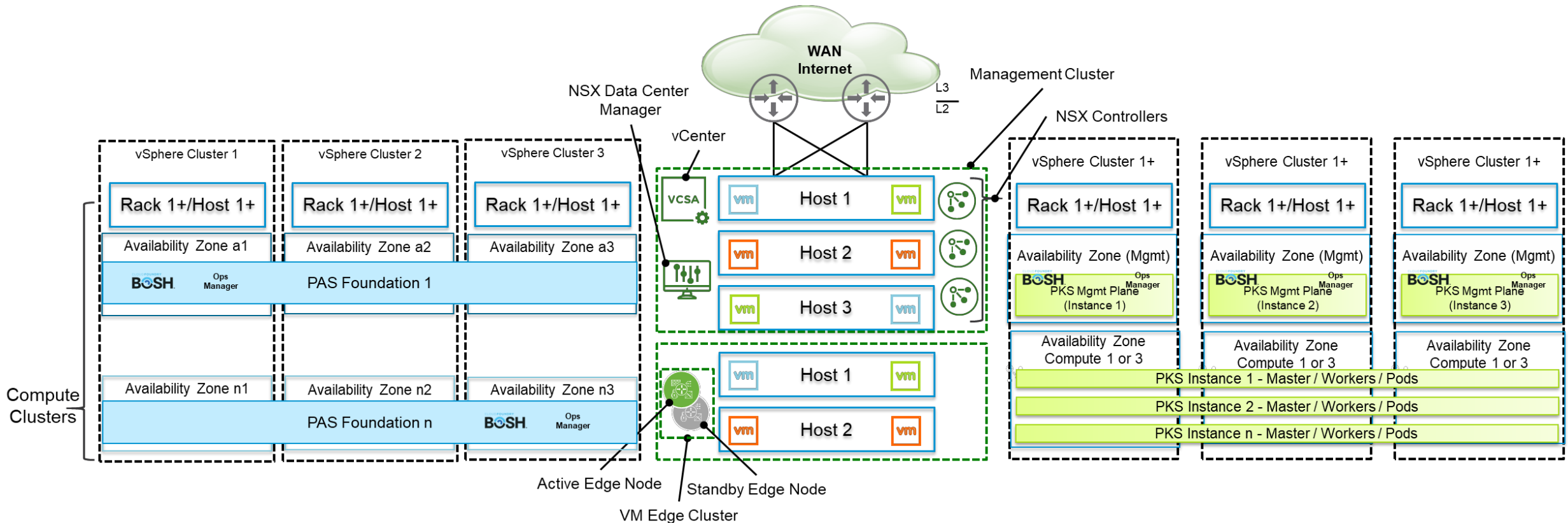
vSphere Cluster Considerations

Sharing NSX-T across multiple PAS Foundations and PKS Instances



vSphere Cluster Considerations

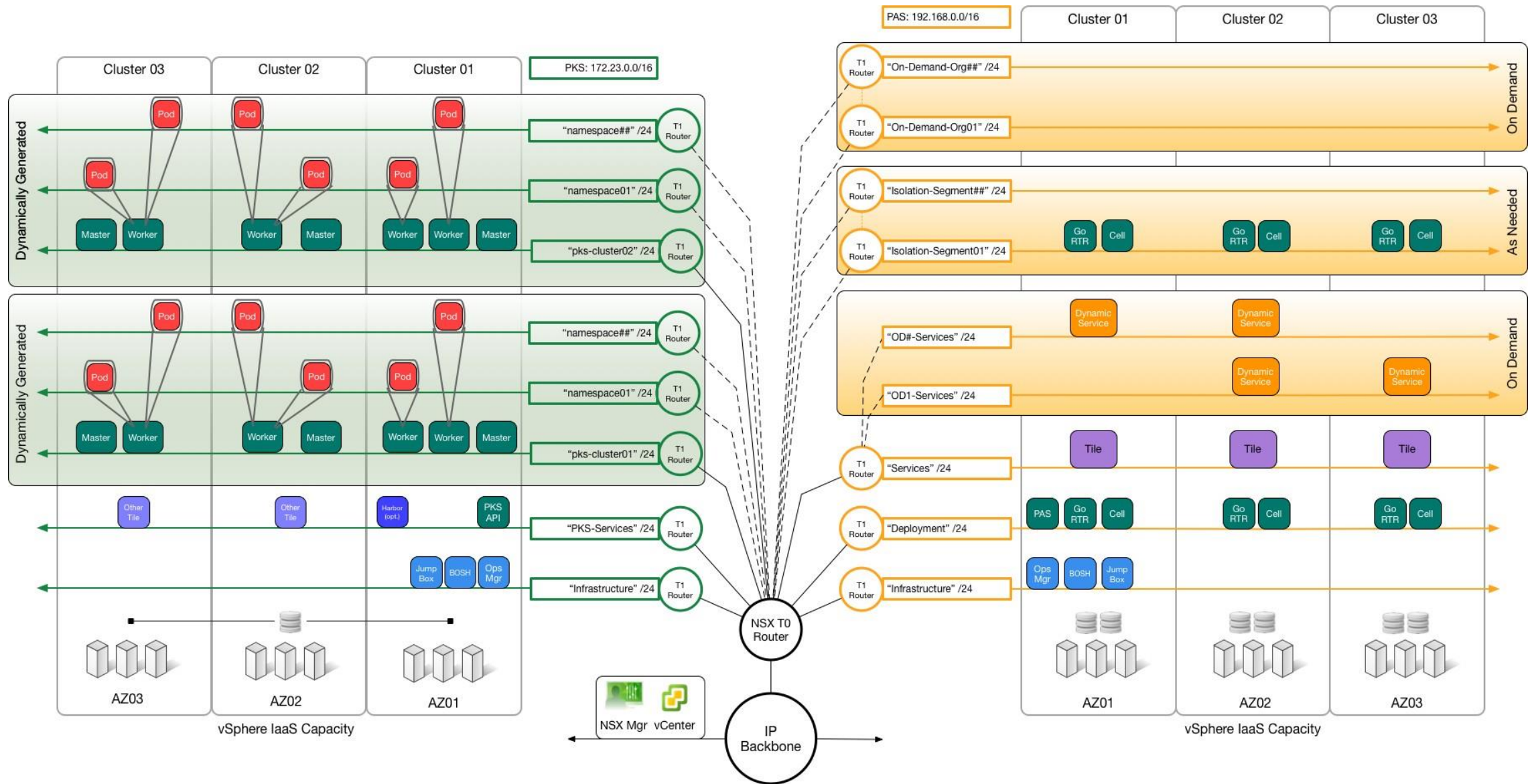
Sharing NSX-T across multiple PAS Foundations and PKS Instances



NSX Data Center for PCF

Pivotal's Ref architecture

Pivotal's Ref Architecture



Pivotal Ready Architecture

- <https://pivotal.io/pivotal-ready-architecture>

Pivotal Ready Architecture: What's in the Box

Pivotal Ready Architecture by Dell EMC is built on the Dell EMC VxRail appliance to provide a simple way to run Pivotal Cloud Foundry on-premises. Further, Pivotal Ready Architecture provides you with greater control and flexibility over your PCF deployment, with infrastructure that allows you to start small and grow.



Dell EMC VxRail is the only fully integrated, preconfigured, and pretested VMware hyper-converged infrastructure appliance family on the market. It dramatically simplifies IT operations while lowering overall capital and operational costs.



Dell EMC Networking top-of-rack (ToR) high-density switches offer ultra-low latency and line rate performance designed for data centers.



Pivotal Cloud Foundry handles infrastructure, OS patching and container orchestration. The platform also provides built-in tools for scaling, metrics, microservice patterns and modern stream processing, so teams can focus on activities that drive business value.



VMware vSphere® optimizes performance, availability and efficiency from infrastructure and applications, making it the ideal foundation for any cloud environment.



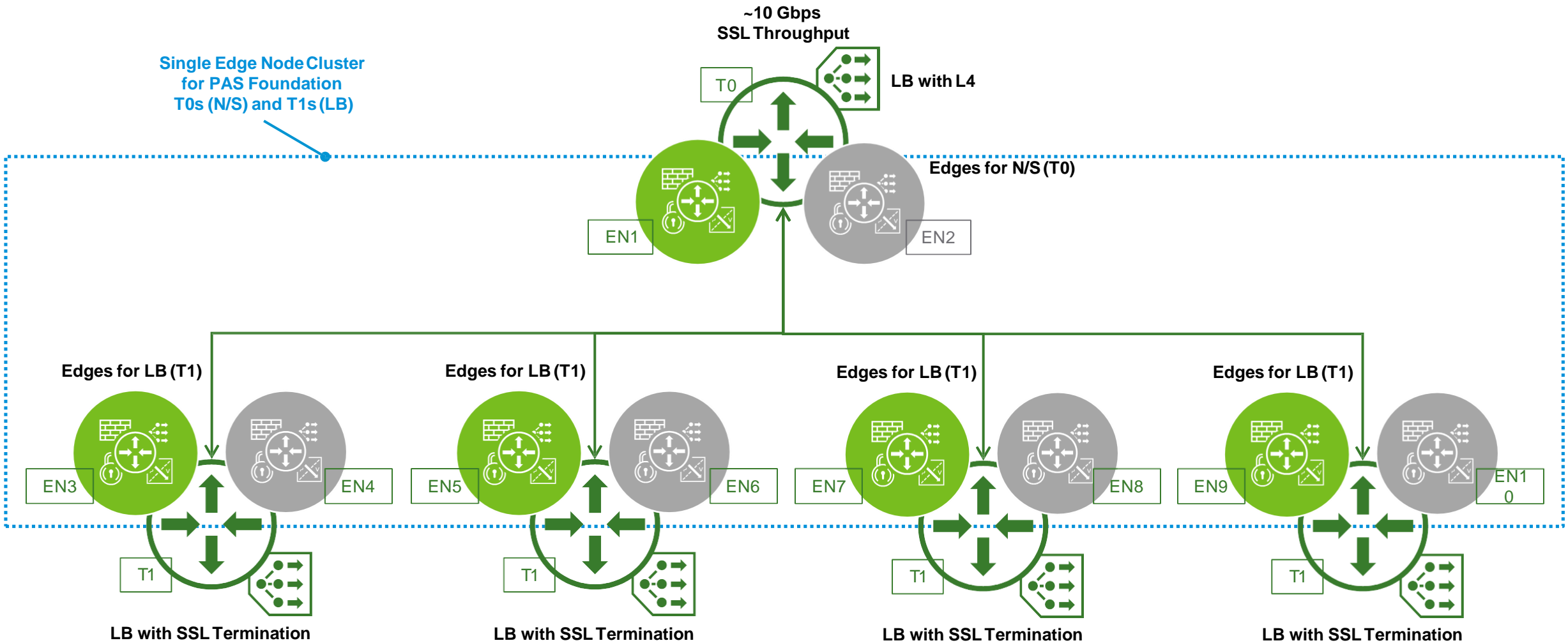
VMware NSX-T® provides an agile software-defined infrastructure to build cloud-native application environments.

Load Balancer

Edge Nodes Design

For Maximizing SSL Throughput

Single Edge Node Cluster
for PAS Foundation
T0s (N/S) and T1s (LB)



Overview

Concourse Pipeline

Pipelines?

Pipelines are defined as a single declarative config file composing together just three core concepts.

Concourse Primitives

Resources

Track Versions of external artifacts used for CI / CD

Any entity that can be checked for new versions, pulled down at a specific version, and/or pushed to

Can be one of many types of built in resources; git repositories, Amazon S3, Buckets, Docker Images or a custom implementation

Jobs

Represent the plan for a build step within a pipeline

Can contain operations against resources, or tasks as steps.

Builds of a job's plan can be triggered manually or trigger on new versions of resource

Tasks

Allow the execution of arbitrary scripts against a set of resources

Can output directories representing a new resource version

Run in a container using a configurable container image

Concourse Primitives

Example: Use ovftool to Install NSX-T

Resources

```
- name: ovftool
  type: file-url
  source:
    url: ((nsx_image_webserver))/((ovftool_file_name))
    filename: ((ovftool_file_name))
    skip_ssl_verification: true
```

Jobs

```
- name: install-nsx-t
  plan:
    - aggregate:
      - get: nsx-t-gen-pipeline
      - get: nsxt-ansible
      - get: ovftool
```

Tasks

```
- task: install-nsx-t
  file: nsx-t-gen-pipeline/tasks/install-nsx-t/task.yml
  params: *nsx-t-gen-params
```

Install NSX-T

With Concourse Pipeline

Concourse Pipeline

NSX-T Installation and Configuration

Install NSX-T Components

NSX-T Manager

Self-signed cert with FQDN

- Generate and Register

NSX-T Controllers

Configure NSX-T

Transport Nodes

- Compute
- Edges

Host-switches

Uplink Profiles

Transport Zones

Create Logical Topology

T0 Router

T1 Routers

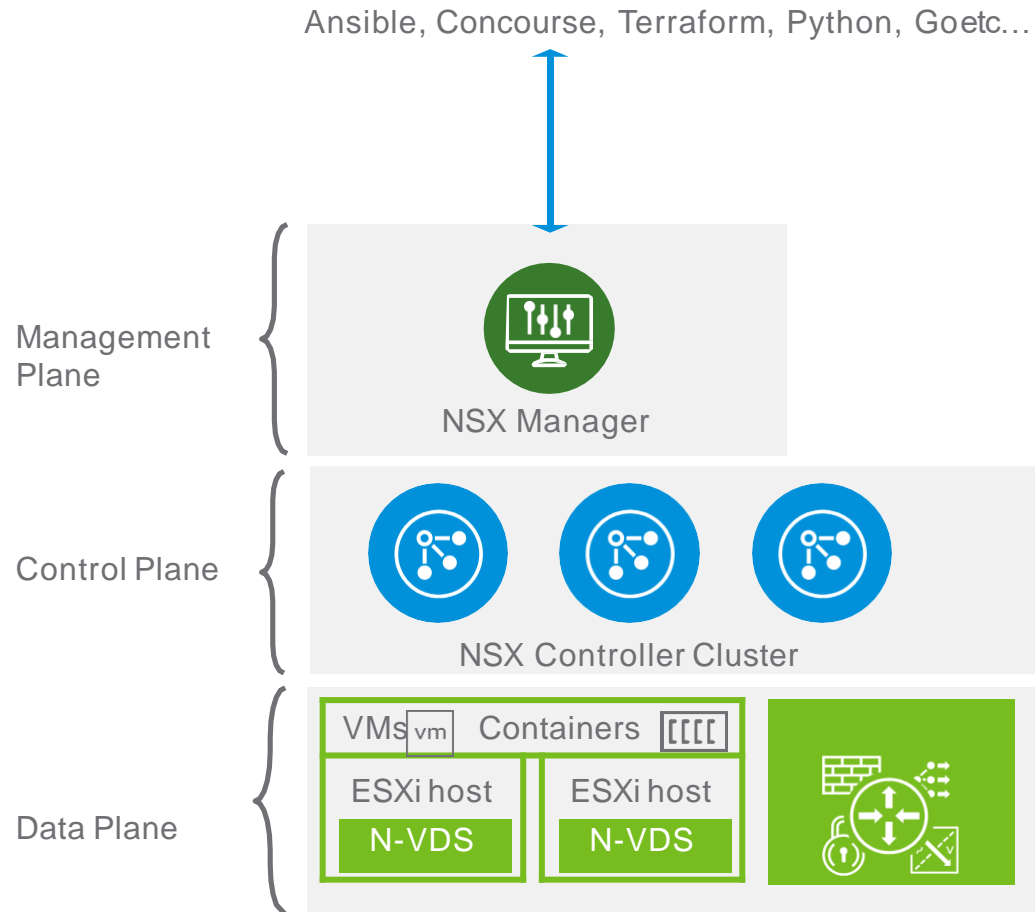
Logical Switches

IP Pools

IP Blocks

NAT Rules

NSX-T Components



UI/API entry point,
Store desired configuration
Interact with other management
components

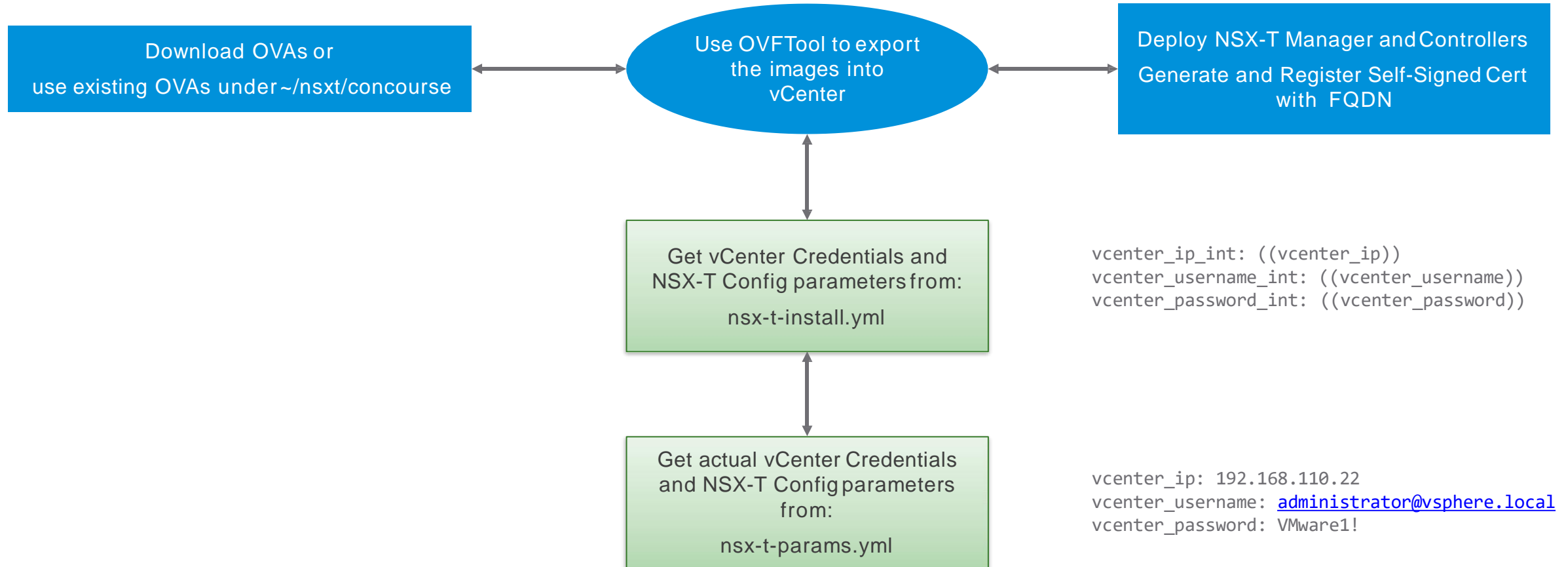
Cluster for scale out and redundancy
Maintain and propagate dynamic state

Transport Nodes:

- Host workloads (VMs, containers) and services
- Switch data plane traffic
- N-VDS or NSX Agent

Install NSX-T

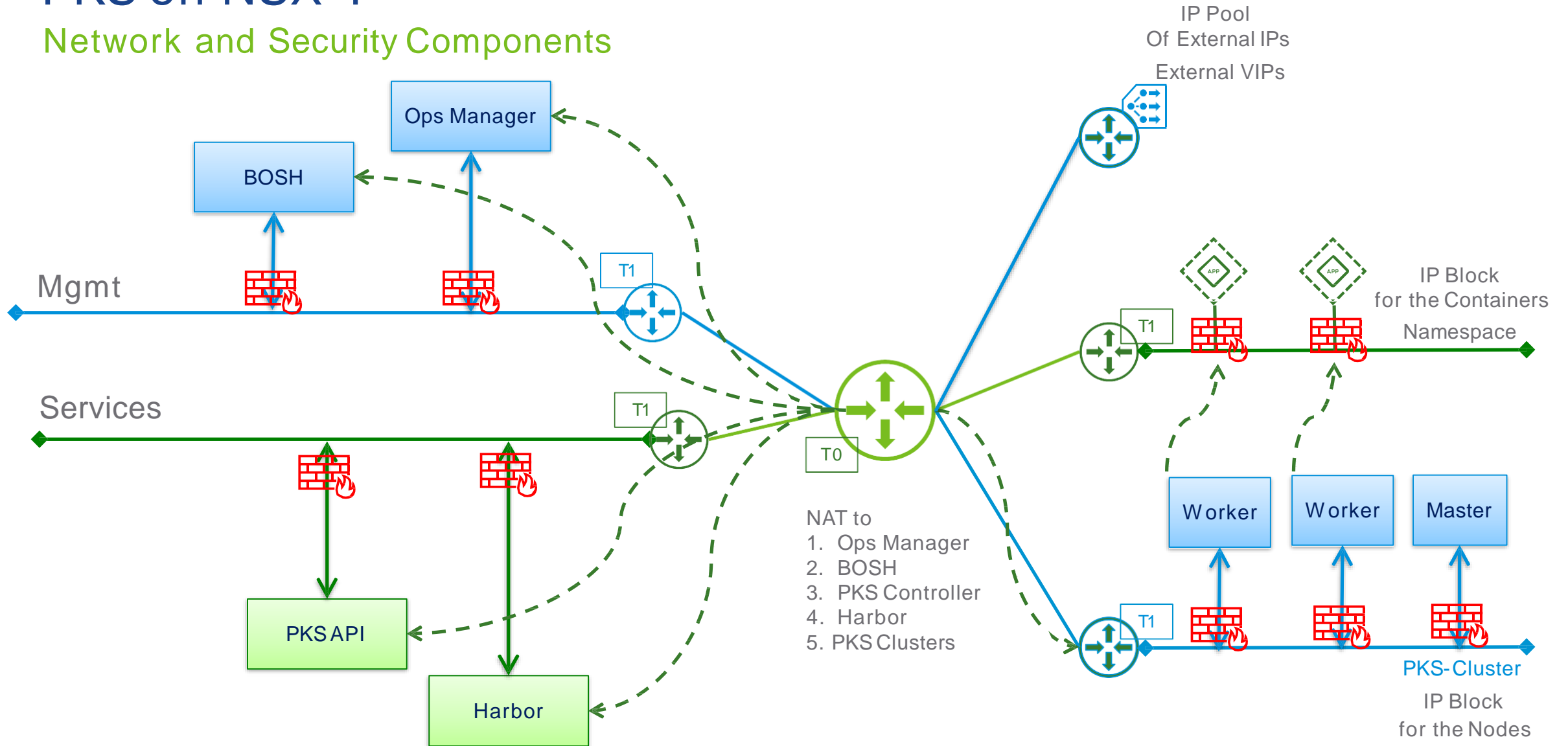
Deploy NSX-T Manager and Controllers



Prepare NSX-T for PKS

PKS on NSX-T

Network and Security Components



T0 Routers

Create and Configure

nsx-t-install.yml

```
nsx_t_t1router_logical_switches_spec_int: ((nsx_t_t1router_logical_switches_spec))
```

→ nsx-t-params.yml

```
# Tier 0 router - t0-PKS  
tier0_router_name: t0-pks  
tier0_uplink_port_ip: 192.168.200.3  
tier0_uplink_port_subnet: 24  
tier0_uplink_next_hop_ip: 192.168.200.1  
tier0_uplink_port_ip_2: # 192.168.200.5  
tier0_ha_vip: # 192.168.200.3
```



T0 - Router

Create and Configure

Tier 0 router - t0-PKS

```
tier0_router_name: t0-pks
tier0_uplink_port_ip: 192.168.200.3
tier0_uplink_port_subnet: 24
tier0_uplink_next_hop_ip: 192.168.200.1
tier0_uplink_port_ip_2: # 192.168.200.5
tier0_ha_vip: # 192.168.200.3
```

vm NSX

- Search
- Dashboard
- Getting Started
- Networking
- Security
- Partner Services
- Tools
- Fabric

- Switching
- Routers
- NAT
- DHCP
- IPAM

Routing

Routers NAT

+ ADD ▾ EDIT DELETE ACTIONS ▾

<input type="checkbox"/>	Logical Router ↑	ID	Type	Connected To
<input type="checkbox"/>	t0-pks	1e87...6ced	Tier-0	
<input type="checkbox"/>	t1-pks-mgmt	f27d...decb	Tier-1	t0-pks
<input type="checkbox"/>	t1-pks-service	a46d...b4aa	Tier-1	t0-pks

T1 Routers and Logical Switches

For Management and Services

nsx-t-install.yml

```
nsx_t_t1router_logical_switches_spec_int: ((nsx_t_t1router_logical_switches_spec))
```

nsx-t-params.yml

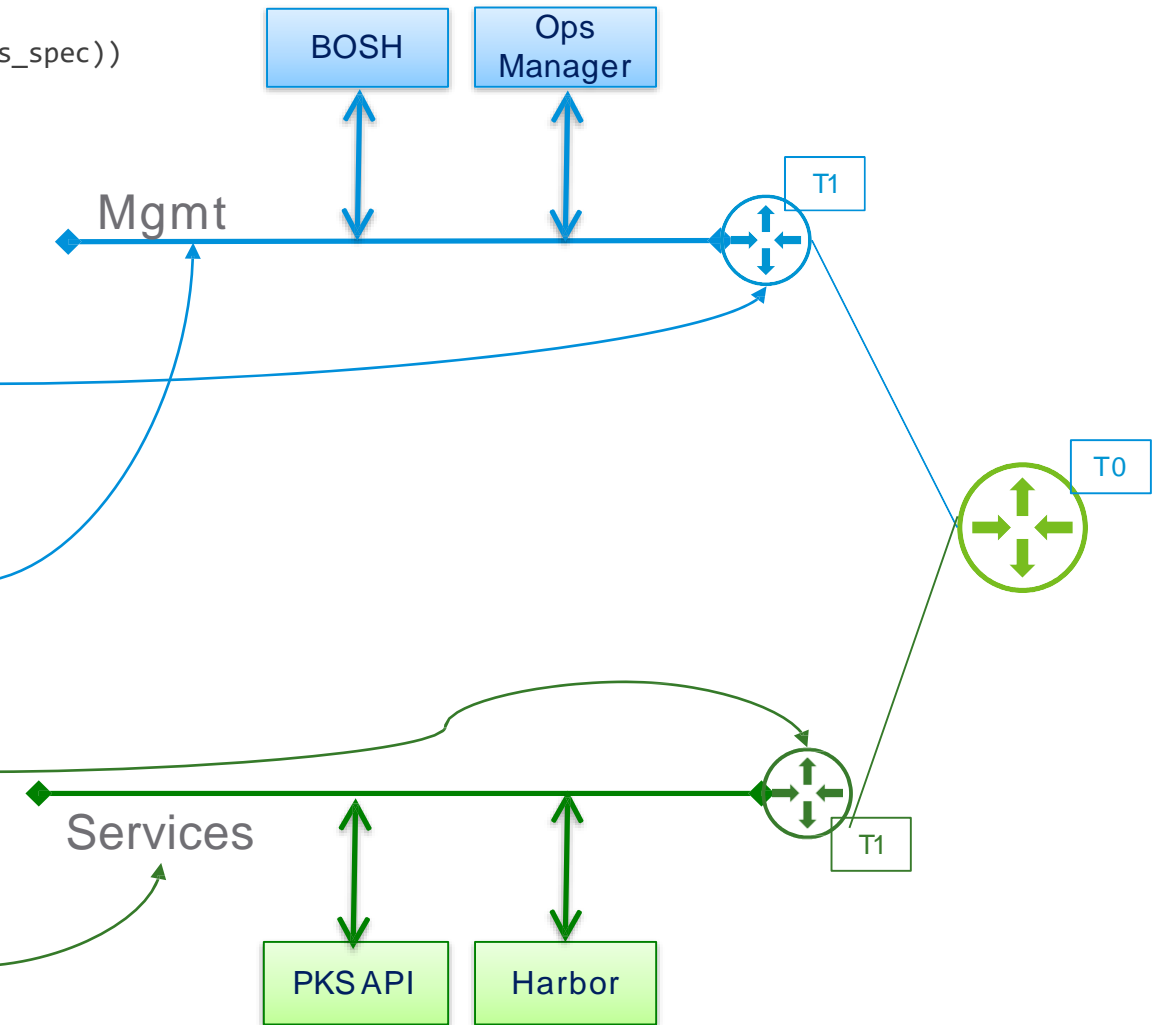
```
nsx_t_t1router_logical_switches_spec: |
  t1_routers:
```

```
# PKS Mgmt Router
- name: t1-pks-mgmt
```

```
# PKS Mgmt Switches
switches:
- name: ls-pks-mgmt
  logical_switch_gw: 172.31.0.1
  subnet_mask: 24
```

```
# PKS Services T1 Router
- name: t1-pks-service
```

```
# PKS Services Logical Switch
switches:
- name: ls-pks-service
  logical_switch_gw: 172.31.2.1
  subnet_mask: 24
```

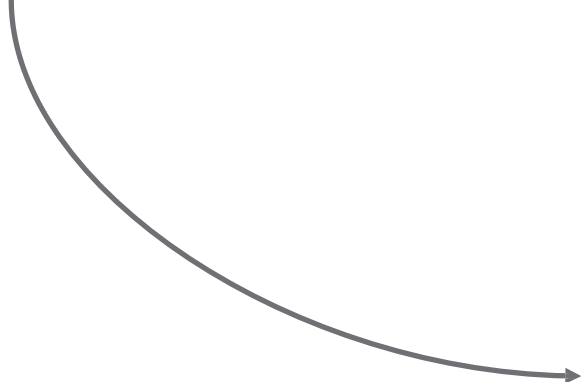


NAT

Configure NAT Rules for PKS

nsx-t-install.yml

```
nsx_t_nat_rules_spec_int: ((nsx_t_nat_rules_spec))
```



nsx-t-params.yml

```
nsx_t_nat_rules_spec: |
  nat_rules:
    # PKS Cluster Network CIDR
    - t0_router: t0-pks
      nat_type: snat
      source_network: 172.31.0.0/24           # PKS Clusters network cidr
      translated_network: 10.40.14.12        # SNAT External: PKS Networks
      rule_priority: 1024                    # Priority

    # PKS Ops manager
    - t0_router: t0-pks
      nat_type: dnat
      destination_network: 10.40.14.3       # External IP: PKS OpsManager
      translated_network: 172.31.0.3        # Internal IP: PKS OpsManager
      rule_priority: 1024                    # Priority

    # BOSH
    - t0_router: t0-pks
      nat_type: dnat
      destination_network: 10.40.14.4       # External IP: BOSH
      translated_network: 172.31.0.4        # Internal IP: BOSH
      rule_priority: 1024                    # Priority

    # PKS Controller
    - t0_router: t0-pks
      nat_type: dnat
      destination_network: 10.40.14.5       # External IP: PKS Controller
      translated_network: 172.31.0.5        # Internal IP: PKS Controller
      rule_priority: 1024                    # Priority

    # Harbor
    - t0_router: t0-pks
      nat_type: dnat
      destination_network: 10.40.14.6       # External IP: Harbor
      translated_network: 172.31.0.6        # Internal IP: Harbor
      rule_priority: 1024                    # Priority
```


NAT

Configure NAT Rules for PKS

The screenshot displays the VMware NSX Manager interface. On the left is a dark navigation sidebar with the 'Networking' menu item circled in red. The main content area is titled 'Routing' and 'NAT'. A left-hand pane shows a list of logical routers: 'Logical Router ↑', 't0-pks' (selected), 't1-pks-mgmt', and 't1-pks-service'. The main pane shows the configuration for the 't0-pks' NAT rule. It includes tabs for 'Overview', 'Configuration', 'Routing', and 'Services'. Below these are statistics for 'Active sessions', 'Packet count', and 'Bytes Data'. A table lists NAT rules with columns for ID, Action, Match (Protocol, Source IP, Source F, Destination IP, Destinat), Translated (IP, Ports), and Appl. A blue box highlights the first five rows of the table.

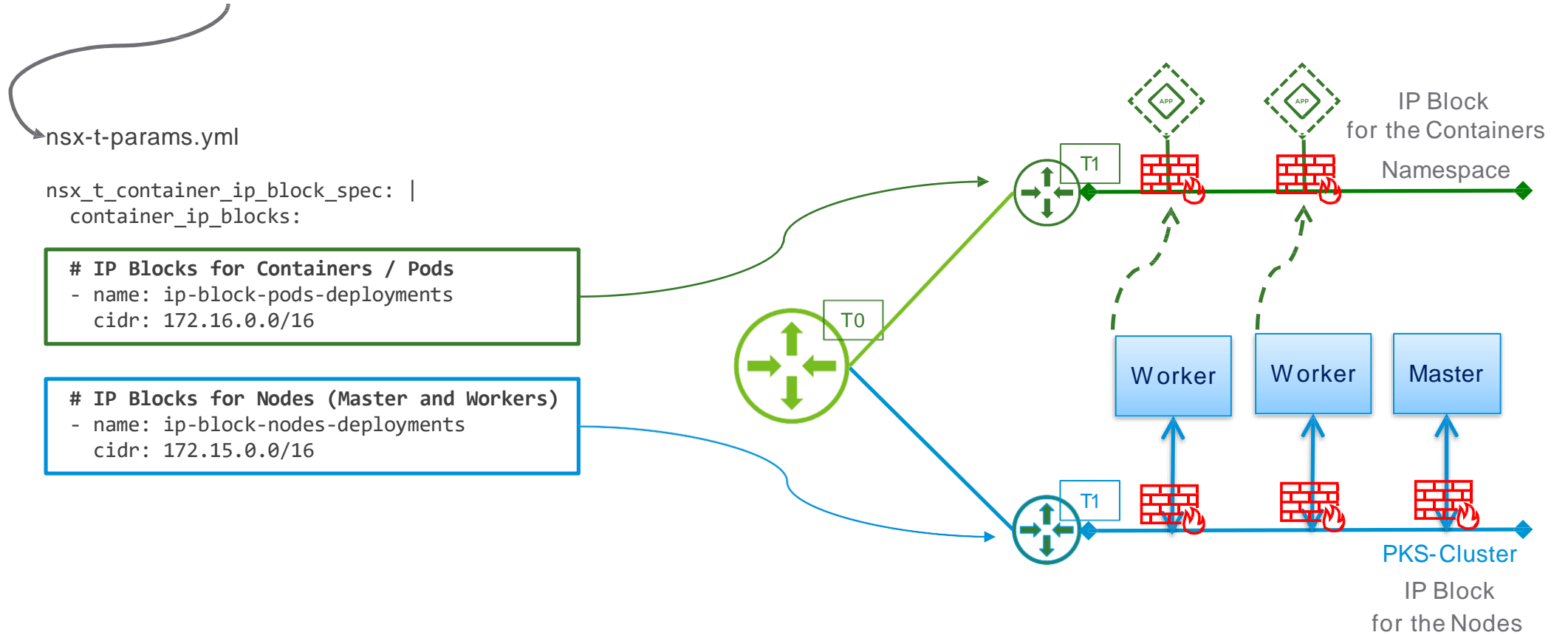
ID	Action	Match					Translated		Appl
		Protocol	Source IP	Source F	Destination IP	Destinat	IP	Ports	
1025	SNAT	Any	172.31.0.0/24	Any	Any	Any	10.40.14.12	Any	
1026	DNAT	Any	Any	Any	10.40.14.3	Any	172.31.0.3	Any	
1027	DNAT	Any	Any	Any	10.40.14.4	Any	172.31.0.4	Any	
1028	DNAT	Any	Any	Any	10.40.14.5	Any	172.31.0.5	Any	
1029	DNAT	Any	Any	Any	10.40.14.6	Any	172.31.0.6	Any	

IP Blocks

Containers & Nodes

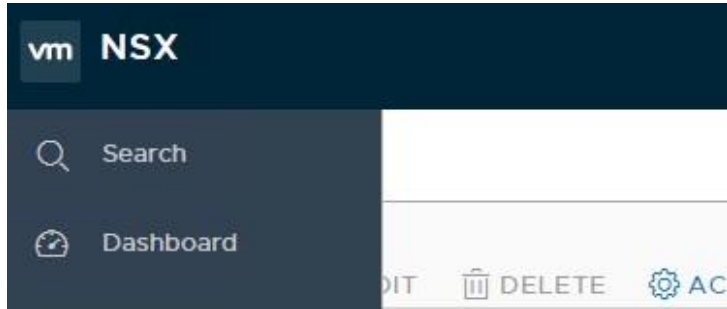
nsx-t-install.yml

```
nsx_t_container_ip_block_spec_int: ((nsx_t_container_ip_block_spec))
```



IP Blocks

Containers and Nodes



```
# IP Blocks for Containers / Pods
- name: ip-block-pods-deployments
  cidr: 172.16.0.0/16
# IP Blocks for Nodes (Master and Workers)
- name: ip-block-nodes-deployments
  cidr: 172.15.0.0/16
```

IPAM

+ ADD EDIT DELETE ACTIONS

IP Blocks	ID	CIDR
ip-block-nodes-deployments	60f5...a68f	172.15.0.0/16
ip-block-pods-deployments	1f9e...6cc1	172.16.0.0/16



External IPs

Used for VIPs on LB

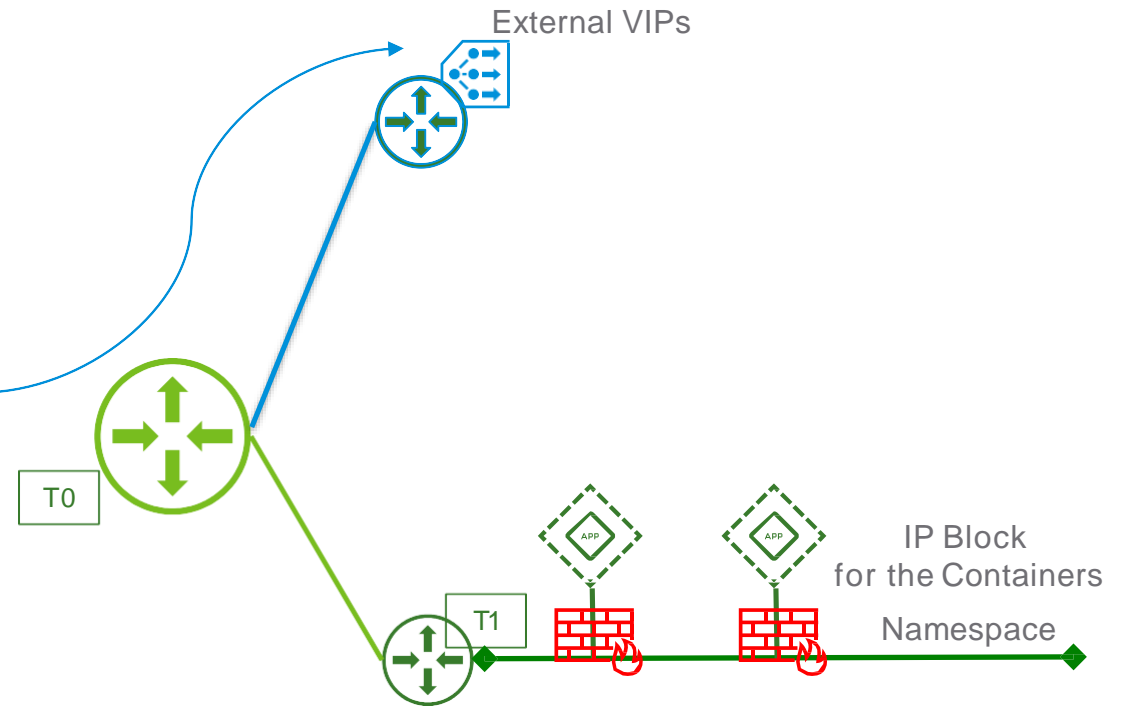
nsx-t-install.yml

```
nsx_t_external_ip_pool_spec_int: ((nsx_t_external_ip_pool_spec))
```

nsx-t-params.yml

```
nsx_t_external_ip_pool_spec: |  
  external_ip_pools:
```

```
# External IP Pool to use for VIPs  
- name: ip-pool-vips  
  cidr: 10.40.14.32/27  
  gateway: 10.40.14.33  
  start: 10.40.14.34 # Should not include gateway  
  end: 10.40.14.62 # Should not include gateway
```



External IPs

Used for VIPs on LB

External IP Pool to use for VIPs

- name: ip-pool-vips
- cidr: 10.40.14.32/27
- gateway: 10.40.14.33
- start: 10.40.14.34 # Should not include gateway
- end: 10.40.14.62 # Should not include gateway

Groups

Groups IP Sets **IP Pools** MAC Sets

Subnets : ip-pool-vips

IP Ranges	Gateway	CIDR	DNS Servers	DNS suffix
10.40.14.34 - 10.40.14.62		10.40.14.32/27		
<input checked="" type="checkbox"/> ip-pool-vips	483f...369c		1	0 of 29
<input type="checkbox"/> tep-ip-pool	2fae...50a7		1	7 of 25

Virtual Machines

Resources

<https://github.com/vmware/nsx-t-datacenter-ci-pipelines>

Quick Intro: <https://www.youtube.com/watch?v=wU6FW1eC5B8>

Resources

How to get started



Learn

Design Guides
Demos



Try

Take a
Hands-on Lab



Connect

Join VMUG, VMware
Communities (VMTN)

nsx.techzone.vmware.com

@VMwareNSX
#runNSX

