Telco Cloud Reference Architecture Guide 3.0

VMware Telco Cloud Platform 3.0
VMware Telco Cloud Platform RAN 3.0
VMware Telco Cloud Infrastructure 3.0

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About the Telco Cloud Reference Architecture Guide

1

This Telco Cloud Reference Architecture Guide provides guidance for designing and deploying a Telco Cloud Platform 5G Edition (Core) or Telco Cloud Platform RAN solution. This guide replaces the existing individual reference architecture guides for the Telco Cloud Infrastructure (Cloud Director and OpenStack Editions), Telco Cloud Platform 5G Edition, and Telco Cloud Platform RAN solutions.

You can use this combined telco cloud reference architecture guide as a single reference point for Telco Cloud network designs that encompass multiple components distributed across the VMware Telco Cloud Platform ecosystem.

Intended Audience

This guide is intended for telecommunications and solution architects, sales engineers, field consultants, advanced services specialists, and customers who are responsible for the design, deployment, and operations of Telco Clouds, Virtualized Network Functions (VNFs), Cloud Native (or containerized) Network Functions (CNFs).

Acronyms and Definitions

The following table lists the acronyms used in this guide.

Acronym	Definition
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
BSS	Business Support System
CAPV	Cluster API Provider vSphere
CBRS	Citizens Broadband Radio Service
CMK	CPU Manager for Kubernetes
CNCF	Cloud Native Computing Foundation, a Linux Foundation project designed to help advance the container technology
CNF	Cloud Native Network Function, executing within a Kubernetes environment
CNI	Container Network Interface

Acronym	Definition
CNS	Cloud Native Storage, a storage solution that provides comprehensive data management for Kubernetes stateful applications.
CNTT	Common NFVI Telco Task Force
CPI	Cloud Provider Interface
CRAN	Cloud (or Centralized) Radio Access Network
CSAR	Cloud Service Archive
CSI	Container Storage Interface. vSphere CSI exposes vSphere storage to containerized workloads on container orchestrators, such as Kubernetes. It enables vSAN and other types of vSphere storage.
CSP	Communications Service Provider
C-VDS	Converged VDS, a vSphere managed Virtual Distributed Switch that is leveraged by NSX (formerly NSX-T Data Center)
CVDS (E)	Converged VDS, a vSphere managed Virtual Distributed Switch that is leveraged by NSX and configured with Enhanced Data Path switching
DHCP	Dynamic Host Configuration Protocol
DPDK	Data Plane Development Kit, an Intel-led packet processing acceleration technology
DRAN	Distributed Radio Access Network
EDP	Enhanced Data Path
ETSI	European Telecommunications Standards Institute
K8s	Kubernetes
LCM	Life Cycle Management
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Infrastructure
NRF	NF Repository Function
OSS	Operational Support System
PCIe	Peripheral Component Interconnect Express
PCF	Policy Control Function
PCRF	Policy and Charging Rule Function
PSA	Pod Security Admission
PV	Persistent Volume
PVC	Persistent Volume Claim
QCI	Quality of Service Class Identifier
RAN	Radio Access Network
RRU	Remote Radio Unit
RU	Radio Unit
SMF	Session Management Function
SBA	Service-Based Architecture

Acronym	Definition
SBI	Service-Based Interface
SR-IOV	Single Root Input/Output Virtualization
STP	Spanning Tree Protocol
SVI	Switched Virtual Interface
SVNFM	Specialized VNF Manager
TCA	Telco Cloud Automation
TCA-CP	Telco Cloud Automation-Control Plane
ToR Switch	Top-of-Rack Switch
UDM	Unified Data Management
UDR	Unified Data Repository
VDS	vSphere Distributed Switch
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager

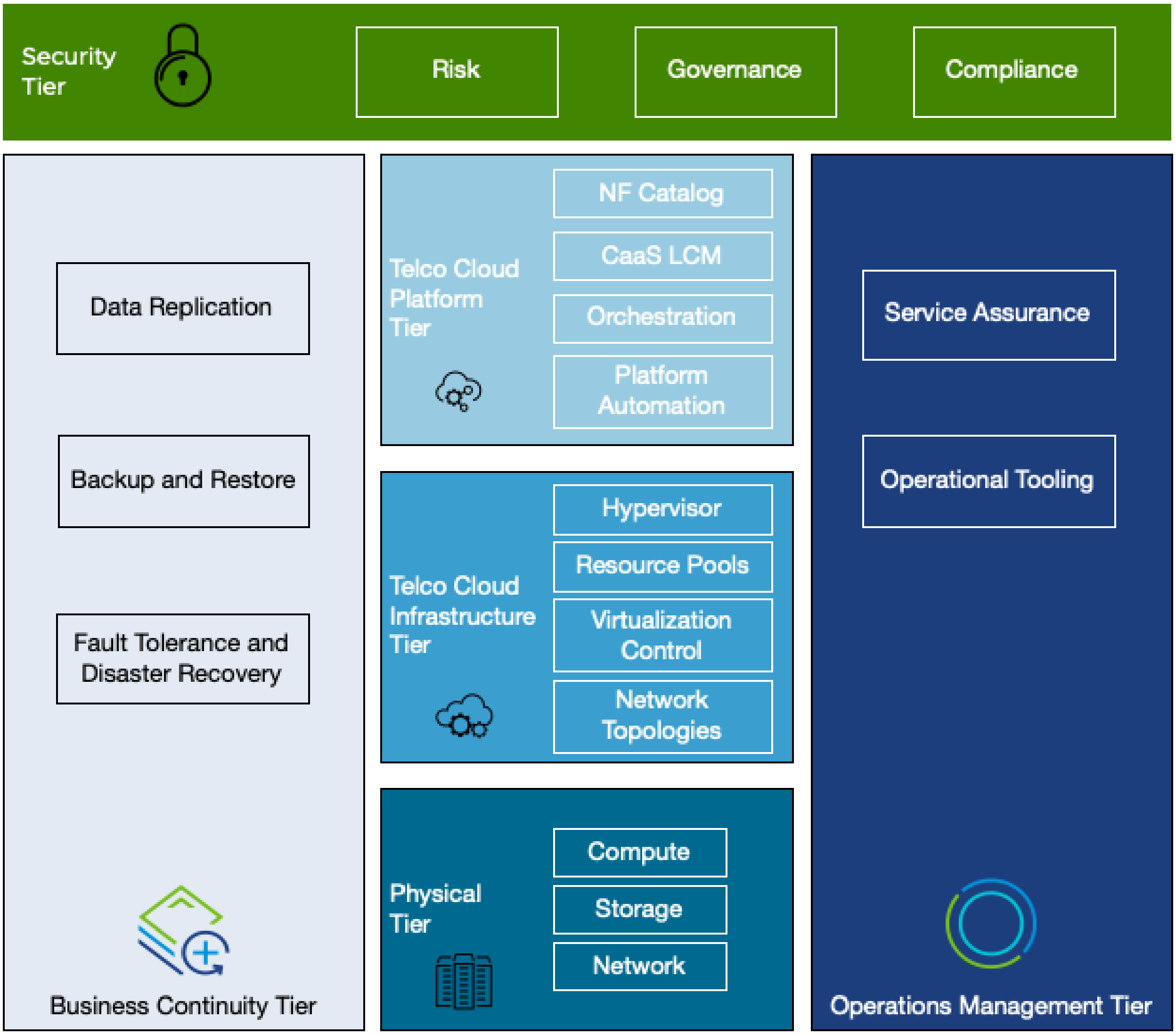
Overview of Telco Cloud

2

VMware Telco Cloud is a purpose-built, carrier-grade cloud services platform, containing NFV features that are designed to support the Communication Service Provider (CSP) requirements for any telco workload including OSS/BSS, VAS, and 4G (LTE), 5G core, and RAN network functions.

The following diagram illustrates the Telco Cloud reference model, with each tier representing an abstraction of the Telco Cloud functional components.

Figure 2-1. Abstraction Layers of VMware Telco Cloud



Telco Cloud Layer

Description

Physical Tier

Represents compute hardware, storage, and physical networking as the underlying pool of shared resources.

Telco Cloud Infrastructure Tier

The lowest tier of VMware Telco Cloud. The Telco Cloud Infrastructure layer provides the virtualization run-time environment with network functions and resource isolation for VM and container workloads. Virtualized compute, storage, and networking are delivered as an integrated solution through vSphere, vSAN, and NSX.

The infrastructure tier is aimed at VM-based Virtual Network Functions (VNFs) and is delivered through a Virtual Infrastructure Manager (VIM) layer. The VIM layer provides tenancy, resource isolation and guarantees for operating VNFs in a multi-vendor environment.

The Infrastructure tier is optimized for telco-class workloads to enable the delivery of quality and resilient services. Infrastructure high availability, performance, and scale considerations that are built into this tier serve as the foundation for upper-level tiers.

Telco Cloud Platform Tier

Provides service management and control functions that bridge the virtual resource orchestration and physical functions to deliver Container Infrastructure Service Management and onboarding / instantiation of Network Functions and Services (both VNF and CNF).

The Telco Cloud platform tier is aimed at Cloud-Native Network functions. These functions are delivered through the CaaS management capabilities of Telco Cloud Automation, including Kubernetes cluster creation and lifecycle management as well as function design, instantiation, and lifecycle management.

The design, instantiation of VNFs from the Telco Cloud Platform tier is also supported, allowing integration with the VIM layer from the Telco Cloud Infrastructure tier.

The Telco Cloud Platform Tier is a centralized control and management function embedded with automation and optimization capabilities and leverages the capabilities provided by the Telco Cloud Infrastructure Layer.

Telco Cloud Operations Tier

An integrated operational intelligence for infrastructure day 0, 1, and 2 operations that spans across all tiers. The functional components of the operations management tier provide bare-metal provisioning of the host OS, logging, topology discovery, observability frameworks, health monitoring, alerting, issue isolation, and closed-loop automation and remediation for 4G, 5G Core, and RAN environments.

Business Continuity Tier

A suite of processes, design considerations, and applications are designed to support Data Replication, Backup, and Restore capabilities alongside a Business Continuity and Disaster Recovery (BCDR) plan.

Security Tier

The Security Tier is responsible for implementing/applying end-to-end security policies such as firewalling, micro-segmentation, and intrusion detection across all tiers within the Telco Cloud.

Read the following topics next:

- Telco Cloud 5G Architecture

Telco Cloud 5G Architecture

This section provides an overview of the deployment topologies and considerations for 5G Core and RAN architectures commonly used by Communication Service Providers (CSPs).

5G Multi-Tier and Distributed Architecture

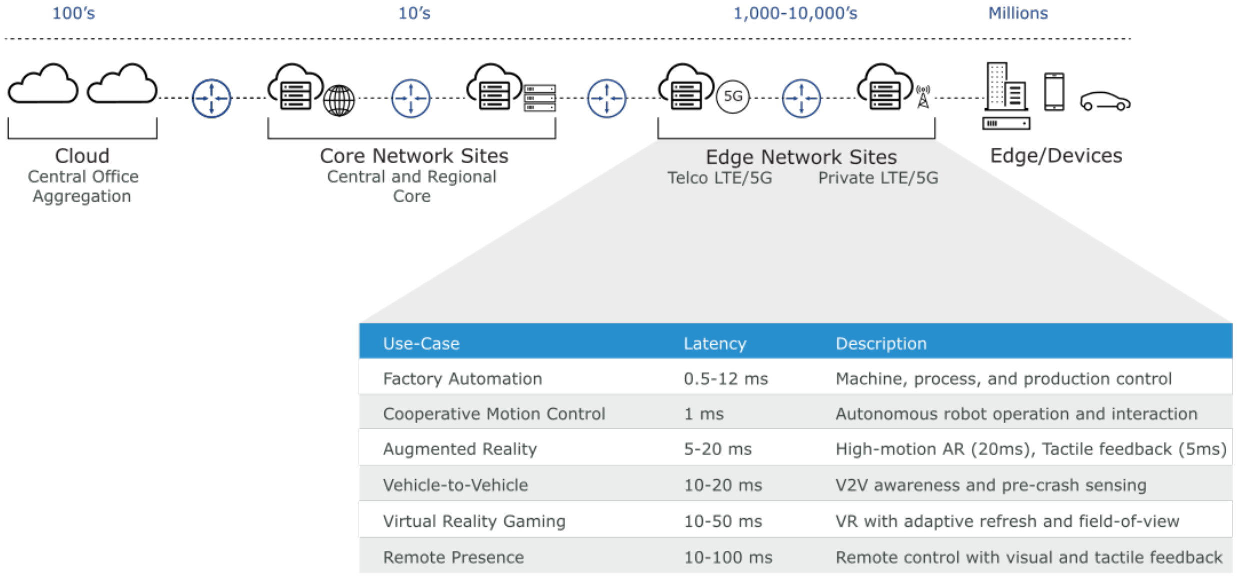
VMware Telco Cloud is a common platform from Core to RAN. This common platform can adapt and scale automatically depending on the workload deployed through VMware Telco Cloud Automation™.

To deploy all workload types such as VNFs and CNFs from 4G and 5G Core to RAN (including 4G Control functions, 4G EPC functions, 5G Core functions, and RAN DU/CU functions), the same infrastructure components, automation platforms, operational tooling, and CaaS infrastructure is leveraged end-to-end.

The 5G network must be dynamic and programmable to achieve business objectives. Network operators must be able to provision virtual network slices on-demand with QoS. This helps meet SLAs, provision functions to increase capacity using industry-standard APIs, and re-route traffic during congestion pro-actively and securely.

To handle the massive data traffic, 5G is designed to separate the user plane from the control plane and to distribute user plane functions as close to the end user device as possible. As the user traffic increases, operators can add more user plane services without changing the control plane capacity. This distributed architecture can be realized by building the data center and network infrastructure based on hierarchical layers. The following diagram illustrates a hierarchical 5G design.

Figure 2-2. Distributed Telco Cloud Architecture



Use-Case	Latency	Description
Factory Automation	0.5-12 ms	Machine, process, and production control
Cooperative Motion Control	1 ms	Autonomous robot operation and interaction
Augmented Reality	5-20 ms	High-motion AR (20ms), Tactile feedback (5ms)
Vehicle-to-Vehicle	10-20 ms	V2V awareness and pre-crash sensing
Virtual Reality Gaming	10-50 ms	VR with adaptive refresh and field-of-view
Remote Presence	10-100 ms	Remote control with visual and tactile feedback

Applications such as RAN cell sites, sensors, and smart devices can reside on the network edge.

- **Far Edge**: The network far edge is the aggregation point for the geographically distributed radio sites hosting RAN and IP routing aggregators. It might also host mobile edge computing software to support private 5G use cases for factory automation, remote presence, and so on. Access mobility and user plane termination functions of the 5G core can also reside in the far edge. The type and number of applications that can be hosted in the far edge sites are limited by available power and space.
- **Near Edge**: The near edge is the aggregation point for far edge sites. It hosts many of the services as the far edge. It also serves as a peering point to access the Internet or other infrastructure-related cloud services. In a distributed 5G Core, the UPF can be deployed at the near-edge for distributed break-out points for efficient internet and off-ramp connectivity.

The central or core sites hosts infrastructure components such as the VMware Telco cloud management components, Kubernetes Management clusters, CICD toolchains, Operational Support Systems (OSS), Observability platforms, Kubernetes image repository, and so on.

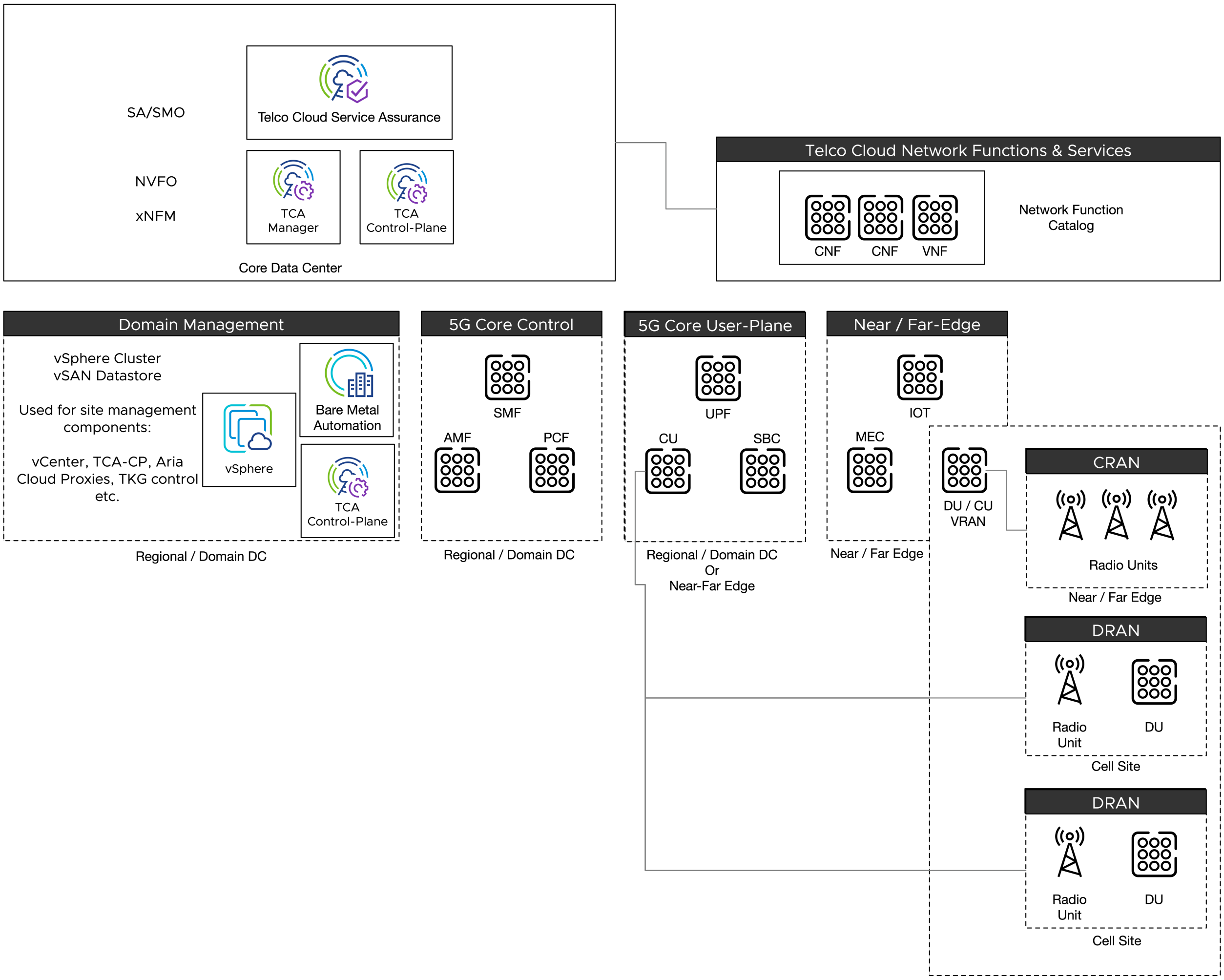
Depending on the 5G deployment, control plane functions of the 5G core, subscriber database, and so on, can reside in the core or regional data center.

RAN Architectures

The RAN network must adhere to the multi-tier and distributed architecture of the 5G Core and the specifications of the RAN environment.

The model of the Telco Cloud architecture allows CSPs to scale 5G deployments based on application requirements and user load. Modern Telco architecture consists of four levels:
5G subscriber databases, data repositories, resource orchestration, and service assurance are typically hosted in the Central, or core data centers. The central and regional data centers also serve as peering points for lawful intercept points. For redundancy, a pair of central data centers are deployed in geographically diverse sites.

Figure 2-3. End-To-End Architecture Covering 5G Core and RAN



In this diagram, the regional or domain data centers host the 5G core user plane function, voice services functions, and non-call processing infrastructure such as IPAM, DNS, and NTP servers. Inbound and outbound roaming traffic can also be routed from the regional data center, out towards the edge. With the inclusion of RAN/DRAN, the distribution of user plane functions becomes more visible.

The Telco Cloud Platform is a compute workload domain that can span from the Core and Regional / Domain Data Centers to individual cell sites.

- **DRAN architecture**: The Cell site uses the Distributed RAN (DRAN) architecture. This architecture uses single hosts, distributed across thousands of remote cell sites. However, more complex architectures for cell-sites (with redundancy) can be used.
- **CRAN architecture**: Near edge sites can implement a Centralized RAN (CRAN) architecture. This architecture uses a cluster of hosts for high availability, resiliency, and scale. In the CRAN model, RAN functions are deployed outside of the cell sites.

For more information about these two architectural models, see Telco Cloud - RAN Domains.

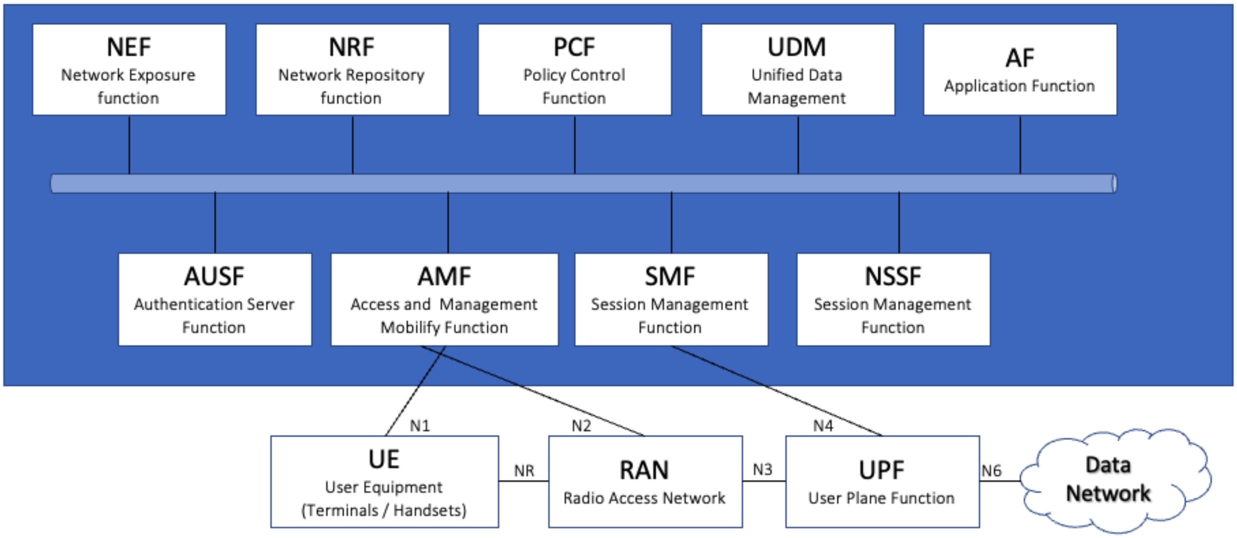
To support new applications and devices that require ultra-low latency, localized processing, and high-throughput networks, CSPs can push the 5G user-plane closer to the application edge. At the same time, RAN dis-aggregation enables efficient hardware utilization and pooling gain and increases deployment flexibility while reducing the Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) of Radio Access.

Note The core sites (Central, Regional, and Edge sites) are part of the 5G Core architecture, while the far edge or cell sites are also commonly part of the RAN architecture for CRAN and DRAN.

5G Service-Based Architecture

5G comes with the specification of a Service-Based Architecture (SBA). The basic principles of SBA are independent of vendors, products, and technologies. A service is a discrete unit of functionality that can be accessed remotely, acted upon, and updated independently. SBAs improve the modularity of products. The network functions creating the 5G service can be broken down into communicating services. With this approach, users can deploy services from different vendors into a single product.

The following diagram illustrates various 5G Core Control Plane functions. The UPF and RAN components are User Plane Functions responsible for radio control, packet routing, Deep Packet Inspection (DPI), and other router-based functions.


NEF
Network Exposure function
NRF
Network Repository function
PCF
Policy Control Function
UDM
Unified Data Management
AF
Application Function
AUSF
Authentication Server Function
AMF
Access and Management Mobilify Function
SMF
Session Management Function
NSSF
Session Management Function
N1
N2
N4
UE
User Equipment (Terminals / Handsets)
NR
RAN
Radio Access Network
N3
UPF
User Plane Function
N6
Data Network

Note Containers are used as a portable and lightweight virtualization solution for 5G Service-Based Architecture (SBA). Kubernetes is one of the components to consider when delivering Carrier-Grade Container as a Service (CaaS). 5G Core network functions are deployed in a cloud-native form although VM-based components can also exist.

A Carrier-Grade CaaS platform requires a complex ecosystem of solutions and functions to form a pre-set business and operating model. The cloud infrastructure modernization changes not only the business model in service agility and metered revenue models but also challenges the silo operating model.

5G Core and RAN Connectivity Considerations

The following core and edge connectivity considerations are required to support different deployment models of 5G RAN:

- **Core and Edge connectivity**: Core and Edge connectivity have a significant impact on the 5G core deployment and it provides application-specific SLAs. The radio spectrum type, connectivity, and available bandwidth also have an impact on the placement of CNFs.

- **WAN connectivity and Bandwidth**: In the centralized deployment model, the WAN connectivity must be reliable between the sites. All 5G control traffic travels from the edge to the core, so any unexpected WAN outage prevents 5G user sessions from being established.

  The fronthaul traffic forwarding from the Remote Radio Unit (RRU) to the DU can be significant in a DRAN environment, so an appropriate bandwidth and infrastructure sizing is required. The WAN sizing and redundant connectivity requirements need to be based on the maximum expected throughput, as required Quality of Service (QoS) can be deployed to protect high-priority traffic between the RAN and Far Edge / Core sites.

- **Components deployment in Cell Site**: Due to the physical constraints of remote Cell Site locations, deploy only the required functions at the Cell Site and the remaining components centrally. For example, observability and logging functions are often deployed centrally to provide universal visibility and control. Non-latency-sensitive user metrics are often forwarded centrally for processing.

- **Network Routing and Local Break-out**: Each cell site or far-edge site routes the user plane and Internet traffic through the local Internet gateways, while the device communication involving management and non-real-time sensitive applications leverage the core.

Telco Cloud Architecture Overview

3

The Telco Cloud reference architecture is intended for Communication Service Providers (CSPs) to deliver Telco Grade services in a single environment that supports a wide range of telco Network Functions (NFs) in both Virtual (VM-based) Network Functions (VNFs) and Cloud-Native (Container-based) Network Functions.

The VMware Telco cloud provides an automated approach to deploy telco applications, from legacy applications to 5G Core and RAN functions.

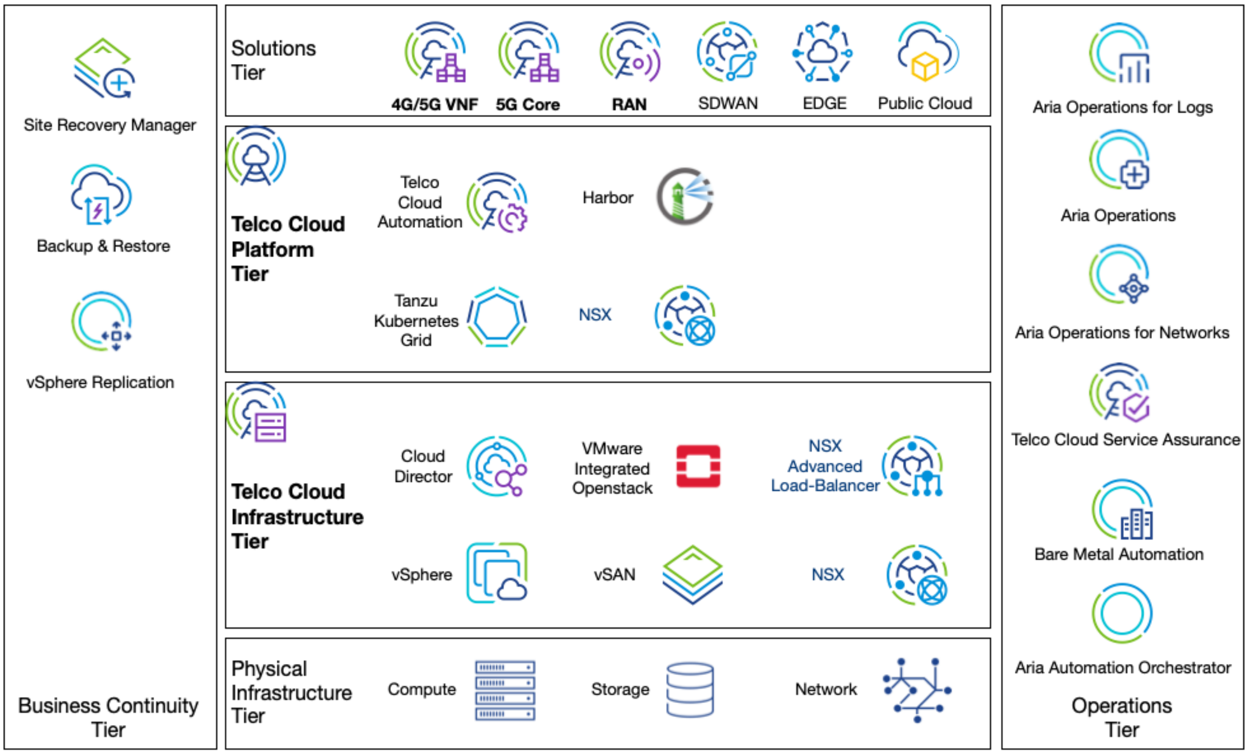
The modularized Telco Cloud architecture is based on various building blocks. This allows for components to be interchanged to create modular, customized design with a specific set of outcomes as determined by the CSPs.

Elements are not exclusively allocated to a single tier, so elements within each tier are flexible and can be consumed as required. The architecture overview diagram represents one way of building a Telco Cloud.

The modules and elements are rigorously tested, beyond the traditional interoperability to ensure stability and wide-ranging compatibility, the ultimate goal is to create an architecture in such a way as to ensure the Communication Service Provider outcomes and goals are achieved.

In relation to the Chapter 2 Overview of Telco Cloud diagram, the following diagram illustrates the building blocks of the Telco Cloud and the VMware products that exist within each tier.

Figure 3-1. Building Blocks of the Telco Cloud



Note Not all components in each Tier are necessary, the Telco Cloud design and implementation are based on specific customer requirements. In this diagram, Telco Cloud Automation includes the Airgap server (if required) for deploying Tanzu Kubernetes Grid in a restricted environment.

Read the following topics next:

- Greenfield and Brownfield Deployments
- Physical Infrastructure Tier
- Telco Cloud Infrastructure Tier
- Telco Cloud Platform Tier
- Telco Cloud Operations Tier

Greenfield and Brownfield Deployments

The Telco Cloud can be deployed as a greenfield deployment or as an extension to an existing brownfield deployment.

Greenfield Deployment

Depending on the requirements, greenfield deployments can include components of the Telco Cloud Infrastructure Tier and the Telco Cloud Platform Tier. The combination of these two tiers allows for the deployment of 4G, 5G Core, or RAN environments and legacy telco applications.

- **VNFs only**: For greenfield deployments where the required outcome is to deploy VNFs (VM-based workloads), the Infrastructure tier is recommended. In the infrastructure tier, you can use VMware Cloud Director or VMware Integrated OpenStack as a Virtual Infrastructure Manager (VIM). NSX functions as the Software-Defined Networking stack for the infrastructure platform. NSX Advanced Load Balancer (formerly Avi Networks) can be used to provide L4 load balancing and can also be integrated with the VIM for tenant control
- **CNFs only**: For greenfield deployments that focus only on Cloud-Native Network Functions (CNFs) deployed on Tanzu Kubernetes Grid, the Platform tier is recommended. The Compute and Storage tiers are provided through vSphere and vSAN in the Infrastructure tier. In addition, NSX is used for 5G Core deployments. Depending on the CNF requirements, NSX Advanced Load Balancer (formerly Avi Networks) can be used to provide L4 load balancing and L7 ingress functionalities.
- **VNFs and CNFs**: For mixed-mode deployments, a combination of Infrastructure and Platform tiers is recommended. Telco Cloud Automation in the Platform tier can be used to onboard, instantiate, and life cycle management (LCM) not only CNFs but VNFs deployed to one of the Infrastructure endpoints. Depending on the design goals, NSX and or NSX Advanced Load Balancer can be used for additional networking functionality. For CNFs, the traditional VIM component is not used. Considerations for resource isolation and multi-tenancy differences must also be considered for VNFs and CNFs.
- **RAN-specific CNFs only**: For RAN-only deployments that are focused on the DU and CU network functions, the Platform tier is recommended. Some components notably vSAN and NSX are not included in this deployment model. Due to the architecture of RAN-only environments, these components are not commonly used. The Telco Cloud Platform RAN bundle uses fewer components from both the Infrastructure and Platform tiers although a common set of management and operational components exist in all deployment models.

Note For more information about the design specifics for 5G Core, RAN, and legacy environments, see Telco Cloud Architectures.

Brownfield Deployments

After deploying a Telco Cloud as part of vCloud NFV, Telco Cloud Infrastructure, or cloud-native Telco Cloud Platform, additional components from the portfolio can be leveraged to extend or enhance the existing Telco cloud environment.

- **Existing vCloud NFV Base**: To upgrade an existing vCloud NFV environment to the Telco Cloud Infrastructure layer, the Infrastructure Components (such as vSphere, vSAN, NSX, VMware Cloud Director, VMware Integrated OpenStack) must be upgraded. Depending on the design requirements, additional components such as NSX Advanced Load Balancer can be leveraged.

  The existing vCloud NFV deployment can also be upgraded and evolved to incorporate the Telco Cloud Platform tier, enabling new cloud-native network functions and services through Telco Cloud Automation and Tanzu Kubernetes Grid.

- **Adding Capabilities to an Existing Telco Cloud Infrastructure Deployment**: According to the design requirements, an existing Telco Cloud Infrastructure deployment can be enhanced to support cloud-native network functions. This might require upgrades to one or more of the Infrastructure components to ensure compatibility between the components of the Telco Cloud Platform tier and the Telco Cloud Infrastructure tier.

When expanding the scope of an existing environment, mixed mode environments are supported. This allows upgraded vCenter Servers to be used for both deployments on the new ESXi cluster while co-existing with the existing environment on the original ESXi versions.

Physical Infrastructure Tier

This section covers the physical architecture elements of the Telco Cloud, this is broken into compute, storage, and network elements.

Figure 3-2. Physical Infrastructure Tier

Compute Pod Architecture

The Telco Cloud design uses modular, reusable compute building blocks called Pods. Each Pod can include a combination of compute, storage, and networking platforms.

Note Compute pods are different from Kubernetes Pods.

A pod architecture provides hardware resources that allows for redundancy and availability. Each pod is connected to the underlay network fabric. This data center fabric (and the subsequent WAN) connects pods and facilitates efficient data transfer.

As a logical component of the Telco Cloud, the pod must be aligned physically to a rack within the data center. Edge and cell-site deployments have less host count than typical pod deployments.

The pod and rack mapping allows for a repeatable L2 design. VLANs that are created for the various management functions are re-used across pods or racks. However, the VLAN is not spanned across pods or racks. Each pod or rack uses locally significant VLANs to create a simple, reusable architecture.

A single rack can include a single or multiple smaller pods. However, when creating large pods that span across multiple racks, ensure the use of fault-domains. When using vSAN as the pod storage mechanism, the VLAN provisioning must also be planned to avoid L2 stretching where possible.

It is also common to separate out resources for control-plane and user-plane functions. To maximize the resource utilization and provide maximum performance and throughput, user-plane functions are typically isolated from control-plane functions.

The following pod types are part of the Telco Cloud deployment:

- Management pod
- Compute pods (Control-Plane, User-Plane, or Mixed Mode)
- Network edge pods
- Near/Far edge pods
- Cell sites

Note In smaller environments, some pod types can be combined to save resources. For example, the compute and network edge pods can be combined to create a single, shared network edge and compute pod.

Management Pod

The management pod is responsible for the management of the Telco Cloud. The servers in the management pod host vCenter Servers, NSX Managers and edge nodes, NSX Advanced Load Balancer controllers and services edges, Virtual Infrastructure Managers (VIMs), Telco Cloud Automation, and all the components from the Operations and Business Continuity tiers.

Note Typically NSX edge nodes or Service Engines from NSX ALB are deployed into the edge Pod. However, for some use-cases, a limited number of these are deployed into the management pod.

The management pod hosts the critical components of the Telco Cloud, so it must be deployed without any single points of failure across the server, storage, and networking configurations.

The management pod can be deployed using various methods. The release bundle for Telco Cloud Infrastructure or Telco Cloud Platform specifies a fixed set of releases for the VMware applications such as vCenter, NSX, and so on.

The management pod does not require the same level of feature-rich functionality as the workload domains. The management pod and the components it uses can deviate from the release bundle but they must be compatible. Ensure that the appliance hardware version is compatible with the Management vSphere versions. For more information, see the VMware Interoperability Matrix.

The management pod has its own Management vCenter, typically its own set of NSX Managers and an NSX Advanced Load-Balancer deployment that control the management pod. This allows for the full vCenter functionality (HA, DRS, vSAN) within the management pod and the ability to provide Network Overlay and Load Balancer services for applications within the management pod (such as Cloud Director or Aria operations).

Additional Resource vCenter Servers and NSX Managers that control the compute/edge pods must be aligned to the recommended or compatibility versions based on the Telco Cloud release notes.

The two common methods of deploying the management pod:

- Manual deployment of ESXi servers, vCenter, NSX Managers, and so on.
- Automated deployment using tools such as VMware Cloud Foundation (VCF)

Note VMware Cloud Foundation can be used only for the management pod. vCenter Servers, NSX managers, and other components for the Compute environment must not be deployed using VMware Cloud Foundation. Ensure that the component releases deployed by vCloud Foundation are aligned with those recommended in the Telco Cloud bundles.

Compute Pods

Compute pods host the network functions (VNFs or CNFs). The compute pods can share a common design. However, due to the high throughput and performance requirements for user plane applications (such as UPF or other packet gateways), different hardware might be used to delineate between different types of compute pods.

All the resources such as vCenter Servers and NSX Managers that are required to run the compute pods are deployed in the management domain. With this architecture, only network functions and services run in a remote data center, while the components managing the infrastructure reside in the management pod.

Edge Pods

Edge pods host NSX Edge nodes that enable north/south routing from the Telco Cloud to the upstream DC Gateway nodes. The Edge pod can be comprised of NSX Edge nodes running on ESXi servers or servers running directly as Bare Metal Edges.

Edge pods host stateful services and upstream BGP peering towards the network, while many components run in a distributed fashion. Egress traffic is handled by the edge pod.

Note Edge pods can be also used to deploy the Service Engine created by NSX Advanced Load Balancer. The NSX Advanced Load Balancer Controllers reside in the management domain while the Service engines can be distributed across the Edge pods.

Near and Far Edge Pods

Near and Far Edge pods are similar to compute pods. The most common difference is in the size of the pod. While regular compute pods are 8, 16, or more hosts, the near/far-edge pods are lower in count.

The far edge pods are commonly used in 5G Core or RAN use cases for distributed components such as UPF or CU nodes.

Cell Sites

A cell site is a single server rather than a pod. The cell site design is important from the connectivity and reusability perspectives, enabling automation and consistency in the deployment of RAN applications such as the DU.

Cell sites from different hardware vendors or with different hardware (accelerator cards, Network Interface cards with onboard Global Network Satellite System interfaces) might have different physical designs.

Network Architecture

The DC fabric architecture is tightly coupled to the pod architecture. The physical design of the DC fabric can be a L2 fabric, a traditional 3-tier Data Center design, or a Leaf/Spine based architecture. The network architecture might have an impact on the overall performance, convergence, and simplicity of operations. The pod architectures can integrate with multiple network architectures; however, the most common and recommended design is the Leaf/Spine architecture.

For a responsive and high-performance Telco Cloud, the physical network must have the following characteristics:

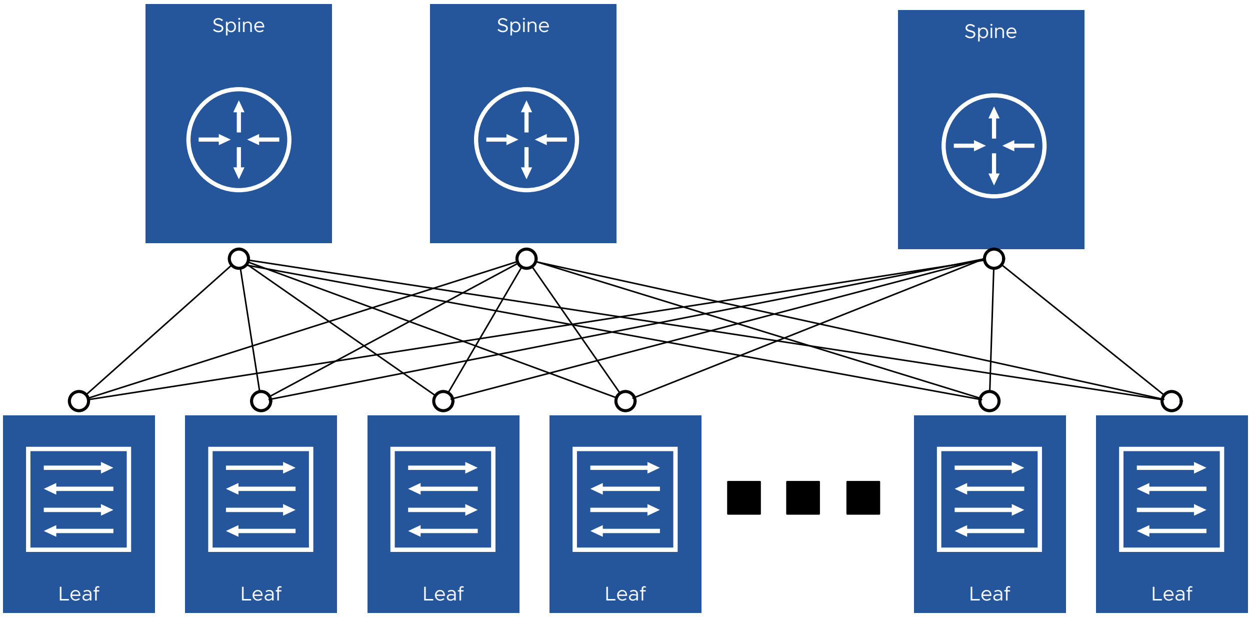
- Simple design
- Highly scalable
- High bandwidth and interface count
- Support for QoS, Low-Latency Queuing, and so on
- Fault tolerant design with no Single Point of Failure (SPOF)
- Minimal or no over-subscription between the leaf and spine

In the Leaf/Spine architecture, the leaf switch is located within the rack and is responsible for the physical connectivity between the servers within the same rack. The rack can include more than one single Leaf switch for redundancy and networking resiliency

The physical network implementation can leverage L2 or L3 based transport but L3 design is recommended, most scalable, and commonly deployed.

The spine switch provides single-hop connectivity between all Leaf switches. An IGP is used between the leaf and spine switches to ensure resiliency when a Leaf or Spine switch fails.

Figure 3-3. Leaf/Spine Physical Network Design

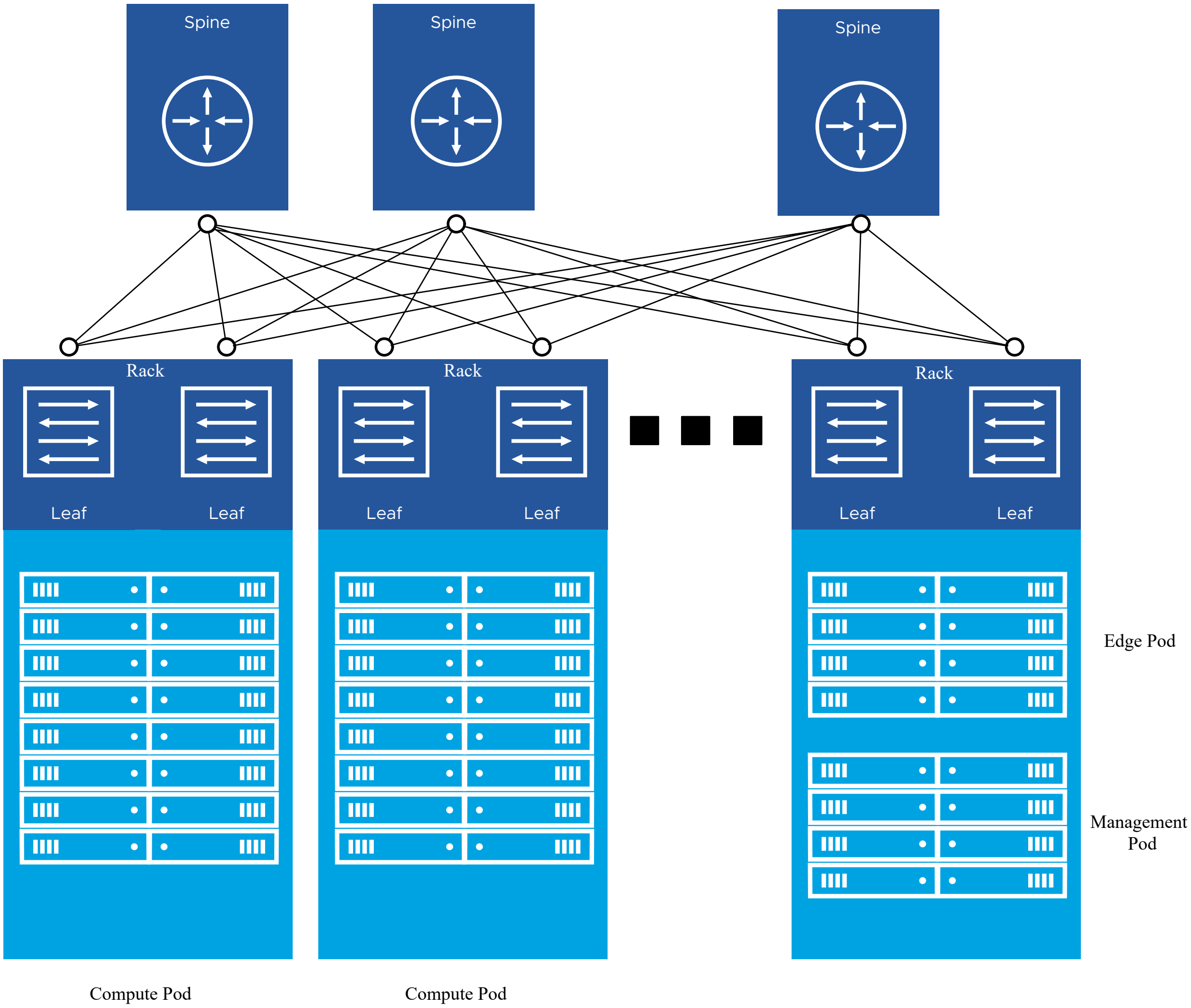


The main benefits of an L3 Transport Leaf/Spine based architecture are as follows:

- Wide range of products are available from various vendors
- Commonly-used, highly-scalable, and cost-effective design
- Enables re-use of configuration across leaf switches due to L2 termination and subsequent routing at the Leaf
- L3 from leaf to spine; VLAN trunking is not required

The following diagram illustrates the rack design for compute pods. Due to cost and power constraints, cell sites do not have redundant leaf switches, ToR switches, or a network spine but rather operate independently. For more information, see Network Virtualization Design.

Figure 3-4. Pod Design



To prevent uplink over-subscription, the physical links between the leaf switches and the spine operate at a higher speed than those between the servers and the leaf switches. Over-subscription is calculated by the total leaf port bandwidth or total uplink bandwidth. For 16 servers in a rack with 25 GB connectivity (total 400 GB), a minimum of four 100 GB uplinks are required. The uplinks must be distributed equally and consistently to allow Equal-Cost Multipath (ECMP) capabilities.

Storage Architecture

vSAN is the recommended storage architecture for Management and Compute pods. vSAN provides a highly scalable, high performing, and fault-tolerant storage architecture to deliver storage to the network functions. vSAN uses the internal disks installed in each server to create a high-available clustered datastore that can be shared across all hosts within a vSphere cluster. For more information, see the vSAN Design Guide.

Note

- The Original Storage Architecture (OSA) vSAN model uses a combination of Solid-State Drives (SSDs) and Magnetic Disk Drives (HDDs) organized into storage tiers to create the datastore.
- The Express Storage Architecture vSAN model (available in vSphere 8.0 or higher) uses storage pools instead of storage tiers. This model requires all NVME-based storage devices and higher-speed NICs for data transfer.

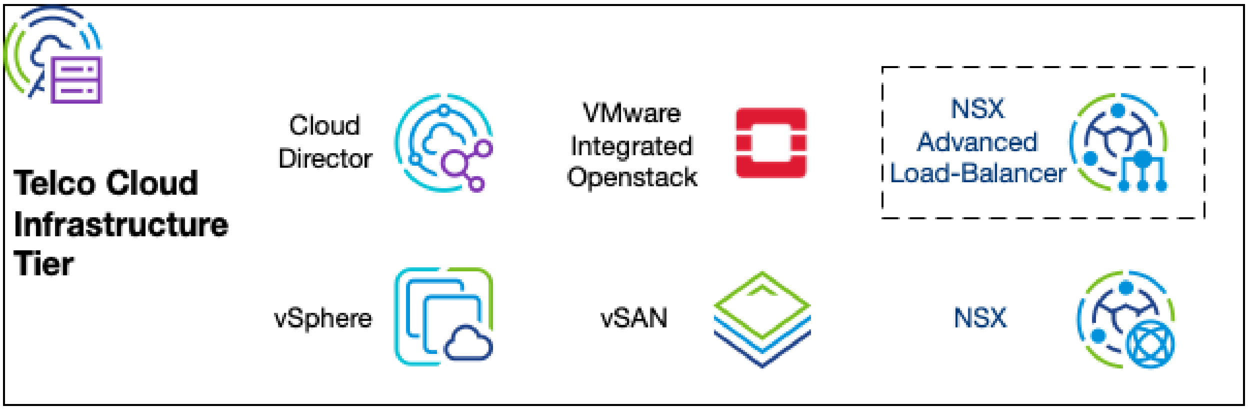
vSAN is the recommended storage architecture for the Telco Cloud but other storage solutions are also supported.

- A wide range of external storage solutions such as iSCSI, NFS, and Fiber Channel are supported by the pods. For more information, see the VMware Compatibility Guide.
- Some storage solutions might not offer direct cloud-native storage options, for example, RWX persistent volumes for Kubernetes. Ensure that the storage solution meets the requirements of the network functions being deployed to the Telco Cloud.

Telco Cloud Infrastructure Tier

The Telco Cloud Infrastructure (TCI) Tier is the foundation for the Telco Cloud and is used by all Telco Cloud derivatives including Telco Cloud Infrastructure, Telco Cloud Platform 5G Core, and Telco Cloud Platform RAN.

Figure 3-5. Telco Cloud Infrastructure Tier


Telco Cloud Infrastructure Tier
Cloud Director
VMware Integrated Openstack
NSX Advanced Load-Balancer
vSphere
vSAN
NSX

The Telco Cloud Infrastructure Tier focuses on the following major objectives:

- Provide the Software Defined Infrastructure
- Provide the Software Defined Storage
- Provide the Software Defined Networking
- Implement the Virtual Infrastructure Manager (VIM)

The physical tier focuses on infrastructure (servers and network switches), while the infrastructure tier focuses on the applications that form the Telco Cloud foundations.

In the Telco Cloud infrastructure tier, access to the underlying physical infrastructure is controlled and allocated to the management and network function workloads. The Infrastructure tier consists of hypervisors on the physical hosts and the hypervisor management components across the virtual management layer, network and storage layers, business continuity and security areas.

The Infrastructure tier is divided into pods, classified as domains such as Management Domain and Compute domain).

Management Pod

The Management pod is crucial for the day-to-day operations and observability capabilities of the Telco Cloud. The management pod hosts the Management and Operational applications from all tiers of the Telco cloud. Applications such as Telco Cloud Automation that is part of the Platform tier reside in the management domain.

The following table lists the components associated with the entire Telco Cloud. Some components are specific to Telco Cloud Infrastructure and VNF deployments (such as VMware Cloud Director). Other components are exclusive to Telco Cloud Platform for 5G Core and RAN stacks and CNF deployments (such as VMware Telco Cloud Automation and its associated components).

Component	Description
Management vCenter Server	Controls the management workload domain
Resource vCenter Servers	Controls one or more workload domains
Management NSX Cluster	Provides SDN functionality for the management cluster such as overlay networks, VRFs, firewalling, and micro-segmentation. Implemented as a cluster of 3 nodes
Resource NSX Clusters	Provides SDN functionality for one or more resource domains such as overlay networks, VRFs, firewalling, and micro-segmentation. Implemented as a cluster of 3 nodes
VMware Cloud Director Cells (VIM)	Provides tenancy and control for VNF based workloads
VMware Integrated OpenStack (VIM)	Provides tenancy and control for VNF based workloads

Component	Description
NSX Advanced Load Balancer management and edge nodes	Load Balancer controller and service edges provide L4 load balancing and L7 ingress services.
Aria Operations Cluster	Collects metrics and determines the health of the Telco Cloud
Aria Operations for Logs Cluster	Collects logs for troubleshooting the Telco Cloud
VMware Telco Cloud Manager and Control-Plane Nodes	ETSI SOL-based NFVO, G-VNFM, and CaaS management platform are used to design and orchestrate the deployment of Kubernetes clusters and onboard / lifecycle manage Cloud-Native Network Functions.
Aria Operations for Networks	Collects multi-tier networking metric and flow telemetry for network-level troubleshooting
VMware Site Recovery Manager	Facilitates implementation of BCDR plans
Aria Automation Orchestrator	Runs custom workflows in different programming or scripting languages
VMware Telco Cloud Service Assurance	Performs monitoring and closed-loop remediation for 5G Core and RAN deployments
Telco Cloud Automation Airgap servers	Facilitates Tanzu Kubernetes Grid deployments in air-gapped environments with no Internet connectivity

Note For architectural reasons, some deployments might choose to deploy specific components outside of the Management cluster. This approach can be used for a distributed management domain as long as the required constraints are met, that is, 150ms latency between the ESXi hosts and various management components.

Each component has a set of constraints around latency and connectivity requirements. Additional components can be hosted in the Management Cluster. For example, Rabbit MQ or NFS VM is used by VMware Cloud Director. Additional elements such as Directory service and NTP can also be hosted in the management domain if required.

To ensure a responsive management domain, do not oversubscribe the management domain in terms of CPU or Memory. To ensure that all components run within a single management domain in case of a failure in one management location, considerations must also account for management domain failover.

The management domain can be deployed in a single availability zone (single rack), across multiple availability zones, and across regions in an active/active or active/standby site. The configuration varies according to the availability and failover requirements.

Workload / Compute Pods

The compute pods or Far/Near edge pods are used to host the Network functions. Some additional components from the management domain can be in the workload domain. For example, Cloud Proxies for remote data collection from Aria Operations.

The workload domain is derived from one or more compute pods. A compute pod is synonymous with a vSphere cluster. The pod can include a minimum of two hosts (four when using vSAN) and a maximum of 96 hosts (64 when using vSAN). The compute pod deployment in a workload domain is aligned with a rack.

Note Configuration maximums can change. For information about current maximums, see VMware Configuration Maximums.

The only distinction between compute pods and far/near-edge pods is the pod size. The far/near edge pods are smaller and distributed throughout the network, whereas the compute pods are larger and co-located within a single data center.

In addition to the pod, the RAN environment is a combination of far/near edge pods and single-host cell sites.

Note Far edge sites can deploy a 2-node vSAN with an external witness for small vSAN-based deployments but is not considered for compute workload pods.

The distribution and quantity of cell sites is higher than far/near-edge or compute pods. Due to factors such as cost, power, cooling and space, the cell site has fewer active components (single servers) in a restricted environment.

Each workload domain can host up to 2,500 ESXi hosts and 40,000 VMs. The host can be arranged in any combination of compute or far/near-edge pods (clusters) or individual cell sites.

Note

- Cell sites are added as standalone hosts in vSphere, and they do not have cluster features such as HA, DRS, and vSAN.
- To maintain marginal room for growth and to avoid overloading a component, the maximums of the component must not exceed more than 75-80%.

The Edge pod is deployed as an additional component in the workload domain. The Edge pod provides North/South routing from the Telco Cloud to the IP Core and other external networks.

The deployment architecture of an edge pod is similar to the workload domain pods when using ESXi as the hypervisor. This edge pod is different from NSX Bare Metal edge pods.

Virtual Infrastructure Management

The Telco Cloud Infrastructure tier includes a common resource orchestration platform called Virtual infrastructure Manager (VIM) for traditional VNF workloads. The two types of VIM are as follows:

- VMware Cloud Director (CD)
- VMware Integrated OpenStack (VIO)

The VIM interfaces with other components of the Telco Cloud Infrastructure (vSphere, NSX) to provide a single interface for managing VNF deployments.

VMware Cloud Director

VMware Cloud Director is used for the cloud automation plane. It supports, pools, and abstracts the virtualization platform in terms of virtual data centers. It provides multi-tenancy features and self-service access for tenants through a native graphical user interface or API. The API allows programmable access for both tenant consumption and the provider for cloud management.

The cloud architecture contains management components deployed in the management cluster and resource groups for hosting the tenant workloads. Some of the reasons for the separation of the management and compute resources include:

- Different SLAs such as resource reservations, availability, and performance for user plane and control plane workloads
- Separation of responsibilities between the CSP and NF providers
- Consistent management and scaling of resource groups

The management cluster runs the cloud management components, vSphere resource pools are used as independent units of compute within the workload domains.

VMware Cloud Director is deployed as one or more cells in the management domain. Some of the common terminologies used in VMware Cloud Director include:

- **Catalog** is a repository of vAPP / VM templates and media available to tenants for VNF deployment. Catalogs can be local to an Organization (Tenant) or shared across Organizations.
- **External Networks** provide the egress connectivity for Network Functions. External networks can be VLAN-backed port groups or NSX networks.
- **Network Pool** is a collection of VM networks that can be consumed by the VNFs. Network pools can be Geneve or VLAN / Port-Group backed for Telco Cloud.
- **Organization** is a unit of tenancy within VMware Cloud Director. An organization represents a logical domain and encompasses its own security, networking, access control, network catalogs and resources for consumption by network functions.
- **Organization Administrator** manages users, resources, services, and policy in an organization.
- **Organization Virtual Data Center** (OrgVDC) is a set of compute, storage and networking resources provided for the deployment of a Network Function. Different methods exist for allocating guaranteed resources to the OrgVDC to ensure that appropriate resources are allocated to the Network Functions.

  Organization VDC Networks are networking resources that are available within an organization. They can be isolated to a single organization or shared across multiple organizations. Different types of Organization VDC networks are Routed, Isolated, Direct, or Imported network segments.

- **Provider Virtual Data Center** (pVDC) is an aggregate set of resources from a single VMware vCenter. pVDC contains multiple resource pools or vSphere clusters and datastores. OrgVDCs are carved out of the aggregate resources provided by the pVDCs.
- **vAPP** is a container for one or more VMs and their connectivity. vAPP is a common unit of deployment into an OrgVDC from within VMware Cloud Director.

VMware Integrated OpenStack (VIO)

VMware Integrated OpenStack integrates with the vCenter Server and NSX components to form a single open-api based programmable interface.

OpenStack is a cloud framework for creating an Infrastructure-as-a-Service (IaaS) cloud. It provides cloud-style APIs and a plug-in model that activates virtual infrastructure technologies. OpenStack does not provide the virtual technologies, instead leverages the underlying hypervisor, networking, and storage from different vendors.

VMware Integrated OpenStack is a VMware production-grade OpenStack that consumes industry-leading VMware technologies. It leverages an existing VMware vSphere installation to simplify installation, upgrade, operations, monitoring, and so on. VMware Integrated OpenStack is OpenStack-powered and is validated to provide API compatibility for OpenStack core services.

OpenStack Core Service	OpenStack Project	Coverage
Block Storage API and extensions	Cinder	Full
Compute Service API and extensions	Nova	Full
Identity Service API and extensions	Keystone	Full
Image Service API	Glance	Full
Networking API and extensions	Neutron	Full
Object Store API and extension	Swift	Tech Preview
Load Balancer	Octavia	Full
Metering and Data Collection Service API	Ceilometer (Aodh, Panko, Gnocchi)	Full
Key Manager Service	Barbican	Full

The VMware Integrated OpenStack architecture aligns with the pod/domain architecture:

- **Management Pod**: OpenStack Control Plane and OpenStack Life Cycle Manager are deployed in the Management Pod.
- **Edge Pod**: OpenStack Neutron, DHCP, NAT, Metadata Proxy services reside in the Edge Pod.
- **Compute Pod**: Tenant VMs and VNFs provisioned by OpenStack reside in the Compute Pod.

The core services of VMware Integrated OpenStack run as containers in an internally deployed VMware Tanzu Kubernetes Grid Cluster. This cluster is deployed using the VIO Lifecycle Manager on top of PhotonOS VMs.

In a production-grade environment, the deployment of VMware Integrated OpenStack consists of:

- **Kubernetes Control Plane VMs** are responsible for Life Cycle Management (LCM) of the VIO control plane. The control plane is responsible for the Kubernetes control processes, the Helm charts, the cluster API interface for Kubernetes LCM, and the VIO Lifecycle Manager web interface.
- **VIO Controller Nodes** host the OpenStack components and are responsible for integrating the OpenStack APIs with the individual VMware components.

OpenStack services are the interface between OpenStack users and vSphere infrastructure. Incoming OpenStack API requests are translated into vSphere system calls by each service.

For redundancy and availability, OpenStack services run at least two identical Pod replicas. Depending on the load, Cloud administrators can scale the number of pod replicas up or down per service. As OpenStack services scale out horizontally, API requests are load-balanced natively across Kubernetes Pods for even distribution.

OpenStack Service	VMware Component	Driver
Nova	vCenter Server	VMware vCenter Driver
Glance	vCenter Server	VMware VMDK Driver
Cinder	vCenter Server	VMware VMDK Driver
Neutron	NSX Manager	VMware NSX Driver / Plug-in

Nova Compute Pods

In VMware Integrated OpenStack, a vSphere cluster represents a single Nova compute node. This differs from traditional OpenStack deployments where each hypervisor (or server) is represented as individual nova compute nodes.

The Nova process represents an aggregated view of compute resources. Each vSphere cluster represents a single nova compute node. Clusters can come from one or more vCenter Servers.

All incoming requests to OpenStack are handled by the Nova Driver, which is integrated with the VMware vCenter Driver through the nova compute pods.

RabbitMQ Pod

RabbitMQ is the default message queue used by all VIO services. It is an intermediary for messaging, providing applications with a common platform to send and receive messages. All core services of OpenStack connect to the RabbitMQ message bus. Messages are placed in a queue and cleared only after they are acknowledged.

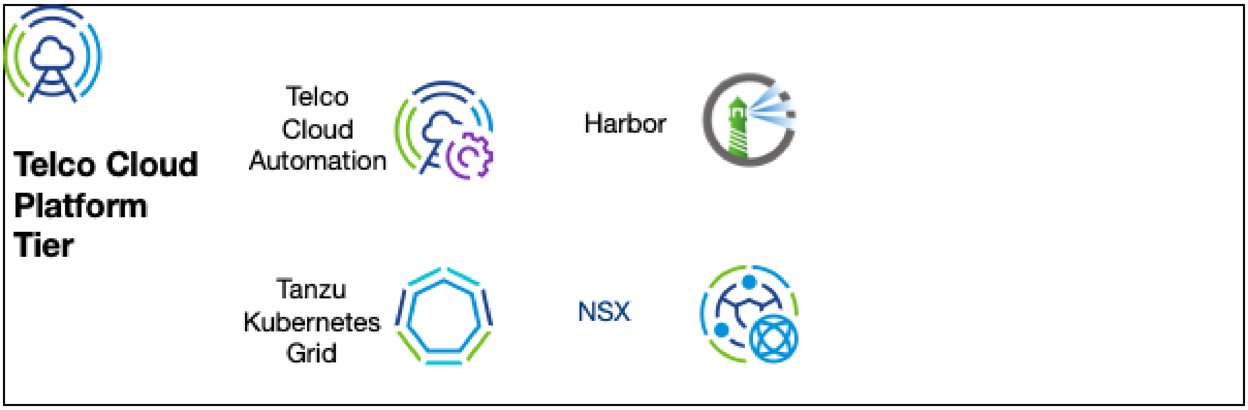
The VIO RabbitMQ implementation runs in active-active-active mode in a cluster environment with queues mirrored between three RabbitMQ Pods.

Each OpenStack service is configured with IP addresses of the RabbitMQ cluster members and one node is designated as primary. If the primary node is not reachable, the Rabbit Client uses one of the remaining nodes. Because queues are mirrored, messages are consumed identically regardless of the node to which a client connects. RabbitMQ provides high scalability.

Telco Cloud Platform Tier

The Telco Cloud Platform (TCP) tier comprises the Automation and CaaS components of the Telco Cloud. While the Infrastructure tier focuses on IaaS, the Platform tier focuses on CaaS and the automated deployment of CaaS clusters, CNFs, and VNFs.

Figure 3-6. Telco Cloud Platform Tier



The major objectives of the Telco Cloud Platform tier are as follows:

- CaaS deployment and life cycle management
- On-boarding, instantiation, and life cycle management of VNFs and CNFs
- Implementing the container repository

The components of the Telco Cloud Platform tier reside in the management domain created as part of the Telco Cloud Infrastructure.

Component	Description
Telco Cloud Automation (Manager and Control-Plane nodes)	Responsible for creation of Kubernetes clusters, onboarding, and lifecycle management of CNFs, VNFs, and Network Services
Telco Cloud Automation (Airgap Server)	Deployed in air-gapped environments to create and customize Kubernetes deployments
Harbor	Provides the OCI repository for images and helm charts

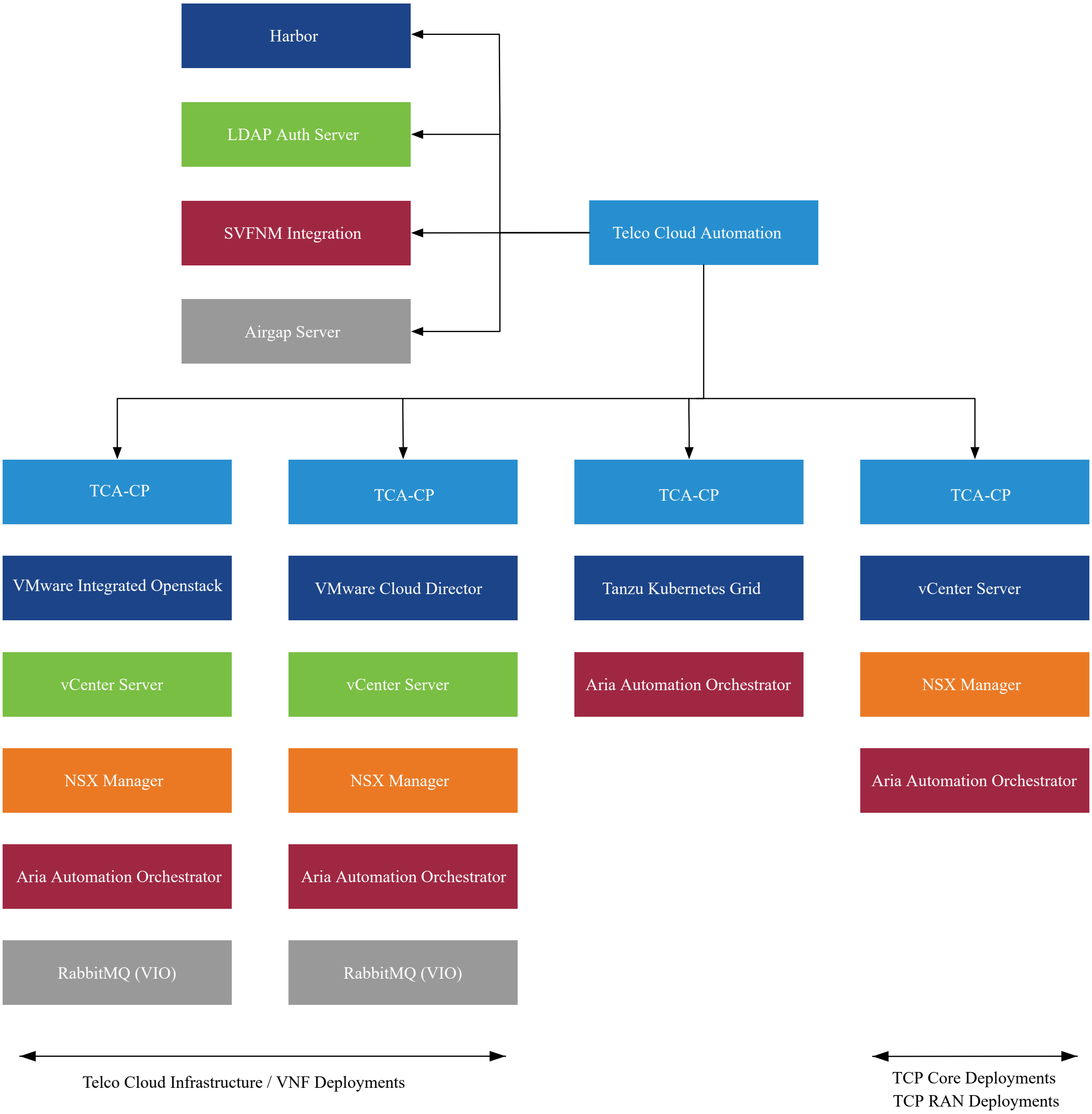
Component	Description
Workload Domain NSX Cluster	Provides SDN functionality for the management cluster such as overlay networks, VRFs, firewalling, and micro-segmentation. Implemented as a cluster of 3 nodes
Aria Automation Orchestrator	Runs custom workflows in different languages

Note It is not necessary to deploy NSX as part of Telco Cloud Infrastructure and Telco Cloud Platform. A common NSX Manager can be used for the workload domains.

Telco Cloud Automation Architecture

Telco Cloud Automation provides orchestration and management services for Telco clouds.

Figure 3-7. Telco Cloud Automation - Virtual Infrastructure Endpoints



- **TCA-Manager**: The Telco Cloud Automation manger is the heart of the Telco Cloud platform. It manages the CaaS cluster deployments, Network Function and services catalog, network function and services inventory, and connectivity to partner systems (sVNFM and Harbor). The TCA manager also manages user authentication and role-based access.

- **TCA-Control Plane**: The virtual infrastructure in the Telco edge, aggregation, and core sites are connected using the Telco Cloud Automation-Control Plane (TCA-CP). The TCA-CP provides the infrastructure for placing workloads across clouds using TCA. It supports several types of Virtual Infrastructure Manager (VIM) such as VMware vCenter Server, VMware Cloud Director, VMware Integrated OpenStack (VIO), and Kubernetes. Each TCA-CP node supports only one VIM type.

  The TCA Manager connects with TCA-CP to communicate with the VIMs. Both TCA Manager and TCA-CP nodes are deployed as an OVA.

  To deploy VNFs, the TCA Manager uses a Control Plane for VMware Cloud Director or VIO, although direct VNF deployments to vCenter is also supported. When deploying Kubernetes, the TCA Manager leverages a Control Plane node configured for vCenter - this is used for new Tanzu Kubernetes Grid deployments.

  A dedicated TCA-CP is required for each non-Kubernetes VIM.

- **SVNFM**: Registration of supported SOL 003 based SVNFMs.
- **NSX Manager**: Telco Cloud Automation communicates with NSX Manager through the VIM layer. A single instance of the NSX Manager can support multiple VIM types.
- **Aria Automation Orchestrator**: Aria Automation Orchestrator registers with TCA-CP and runs customized workflows for VNF and CNF onboarding and day-2 life cycle management.
- **RabbitMQ**: RabbitMQ tracks VMware Cloud Director and VMware Integrated OpenStack notifications. It is not required for Telco Cloud Platform when using Kubernetes-based VIMs only.

Telco Cloud Automation Persona

The following key stakeholders are involved in the end-to-end service management, life cycle management, and operations of the Telco cloud native solution:

Persona	Role
CNF Vendor / Partners	CNF vendors supply HELM charts and container images
	Read access to NF Catalog
CNF Deployer / Operator	Read access to NF Catalog
	Responsible for CNF LCM through Telco Cloud Automation (ability to self-manage CNF)
CNF Developer / Operator	Develops CNF CSAR by working with vendors, including defining Dynamic Infrastructure Provisioning requirements.
	Maintains CNF catalog
	Responsible for CNF LCM through TCA

Persona	Role
	Updates Harbor with HELM and Container images
Tanzu Kubernetes Cluster admin	Kubernetes Cluster Admin for one or more Tanzu Kubernetes clusters
	Creates and maintains Tanzu Kubernetes Cluster template.
	Deploys Kubernetes clusters to pre-assigned Resource Pool
	Assigns Kubernetes clusters to tenants Assigns Developers and Deployers for CNF deployment through TCA
	Works with CNF Developer to supply CNF dependencies such as Container images, OS packages, and so on.
	Performs worker node dimensioning
	Deploys CNF monitoring/logging tools such as Prometheus and fluentd.
	API and CLI access to Kubernetes clusters associated with CNF deployment
Telco Cloud Automation Admin / System Admin	Onboards Telco Cloud Automation users and partners
	Adds new VIM Infrastructure and associates it with TCA-CP
	Infrastructure monitoring through Aria Operations.
	Creates and maintains vCenter Resource Pools for Tanzu Kubernetes clusters.
	Creates and maintains Tanzu Kubernetes Cluster templates.
	Deploys and maintains the Harbor repository
	Deploys and maintains Tanzu Kubernetes bootstrap process
	Performs TCA, TCA-CP, Harbor, infrastructure upgrades, and infrastructure monitoring

Tanzu Kubernetes Grid

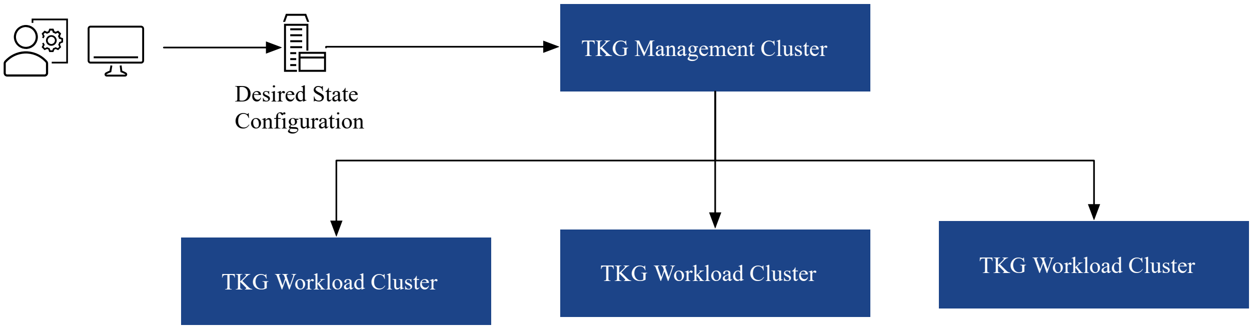
VMware Tanzu Standard for Telco provisions and performs life cycle management of Tanzu Kubernetes clusters.

A Tanzu Kubernetes cluster is an opinionated installation of the Kubernetes open-source software that is built and supported by VMware. With Tanzu Standard for Telco, administrators provision Tanzu Kubernetes clusters through Telco Cloud Automation and consume them in a declarative manner that is familiar to Kubernetes operators and developers.

Within the Telco Cloud, the desired state of Tanzu Kubernetes Grid deployments is configured as a cluster template and the template is used to create the initial management cluster through an internal bootstrap. After the management cluster is created, the cluster template is passed to the Tanzu Kubernetes management cluster to instantiate the Tanzu Kubernetes workload clusters using the Cluster-API (CAPI) for vSphere.

The following diagram shows different hosts and components of the Tanzu Kubernetes Grid architecture:

Figure 3-8. Tanzu Kubernetes Grid Architecture



Cluster API brings declarative, Kubernetes style APIs for application deployments and makes similar APIs available for cluster creation, configuration, and management. The Cluster API uses native Kubernetes manifests and APIs to manage bootstrapping and lifecycle management (LCM) of Kubernetes clusters.

The Cluster API relies on a pre-defined cluster YAML specification that describes the desired state of the cluster and attributes such as the class of VM, size, and the total number of nodes in the Kubernetes cluster, the node pools, and so on.

Tanzu Kubernetes Grid - Control Plane components

The Kubernetes control plane runs as pods on the Kubernetes Control Plane nodes. The Kubernetes Control Plane consists of the following components:

- **Etcd**: Etcd is a simple, distributed key-value store that stores the Kubernetes cluster configuration, data, API objects, and service discovery details. For security reasons, etcd must be accessible only from the Kubernetes API server.
- **Kube-API-Server**: The Kubernetes API server is the central management entity that receives all API requests for managing Kubernetes objects and resources. The API server serves as the frontend to the cluster and is the only cluster component that communicates with the etcd key-value store.

For redundancy and availability, place a load balancer for the control plane nodes. The load balancer performs health checks on the API server to ensure that external clients such as kubectl connect to a healthy API server even during the cluster degradation.

- **Kube-Controller-Manager**: The Kubernetes controller manager is a daemon that embeds the core control loops shipped with Kubernetes. A control loop is a non-terminating loop that regulates the state of the system. In Kubernetes, a controller is a control loop that watches the shared state of the cluster through the API server and moves the current state to the desired state.
- **Kube-Scheduler**: Kubernetes schedulers know the total resources available in a Kubernetes cluster and the workload allocated on each worker node in the cluster. The API server invokes the scheduler every time there is a need to modify a Kubernetes pod. Based on the operational service requirements, the scheduler assigns the workload on a node that best fits the resource requirements.

Virtual Infrastructure Management

Telco Cloud Automation administrators can onboard, view, and manage the entire Telco Cloud through the Telco Cloud Automation console. Details about each cloud such as Cloud Name, Cloud URL, Cloud Type, Tenant Name, Connection Status, and Tags can be displayed as an aggregate list or graphically based on the geographic location.

Telco Cloud Automation administrators use roles, permissions, and tags to provide resource separation for the VIM, Users (Internal or external), and Network Functions.

CaaS Subsystem

The CaaS subsystem allows Telco Cloud operators to create and manage Kubernetes clusters for cloud-native 5G and RAN workloads. VMware Telco Cloud Platform uses VMware Tanzu Standard for Telco to create Kubernetes clusters.

Telco Cloud Automation administrators create the Kubernetes management and workload cluster templates to reduce repetitive tasks associated with Kubernetes cluster creation and standardize the cluster sizing and capabilities. A template can include a group of workers nodes through node pool, control and worker node sizing, container networking, storage class, and CPU manager policies.

Kubernetes clusters can be deployed using the templates or directly from the Telco Cloud Automation UI or API. In addition to the base cluster deployment, Telco Cloud Automation supports various PaaS add-ons that can be deployed during cluster creation or at a later stage.

Add-on Category	Add-on	Description
CNI	Antrea	Antrea CNI
	Calico	Calico CNI
CSI	vSphere-CSI	Allows the use of vSphere datastores for PVs

Add-on Category	Add-on	Description
	NFS	NFS Client mounts NFS shares to worker nodes
Monitoring	Prometheus	Publishes metrics
	Fluent-bit	Formats and publishes logs
Networking	Whereabouts	Used for cluster-wide IPAM
	Multus	Allows more than a single interface to a pod
	NSX ALB / Ingress	Integrates LB or Ingress objects with VMware NSX Advanced Load Balancer
System	Cert-Manager	Provisions certificates
	Harbor	Integrates harbor add-on to Tanzu Kubernetes clusters
	Systemsettings	Used for password and generic logging
TCA-Core	nodeconfig	Used as part of dynamic infrastructure provisioning
Tools	Velero	Performs backup and restore of Kubernetes namespaces and objects.

The CaaS subsystem access control is backed by RBAC. The Kubernetes Cluster administrators have full access to Kubernetes clusters, including direct SSH access to Kubernetes nodes and API access through the Kubernetes configuration. CNF developers and deployers can have deployment access to Kubernetes clusters through the TCA console in a restricted role.

Cloud-Native Networking

In Kubernetes networking, each Pod has a unique IP that is shared by all the containers in that Pod. The IP of a Pod is routable from all the other Pods, regardless of the nodes they are on. Kubernetes is agnostic to reachability. L2, L3 or overlay networks can be used as long as the traffic reaches the desired pod on any node.

CNI is a container networking specification adopted by Kubernetes to support pod-to-pod networking. The specification defines the Linux network namespace. The container runtime allocates a network namespace to the container and passes numerous CNI parameters to the network driver. The network driver then attaches the container to a network and reports the assigned IP address to the container runtime. Multiple plug-ins might run at a time with the container joining networks driven by different plug-ins.

Telco workloads require a separation of the control plane and data plane. A strict separation between the Telco traffic and the Kubernetes control plane requires multiple network interfaces to provide service isolation or routing. To support those workloads that require multiple interfaces in a Pod, additional plug-ins are required. A CNI meta plug-in or CNI multiplexer that attaches multiple interfaces supports multiple Pod NICs.

- **Primary CNI**: The CNI plug-in that serves pod-to-pod networking is called the primary or default CNI, a network interface that every Pod is created with. In case of network functions, the primary interface is managed by the primary CNI.
- **Secondary CNI**: Each network attachment created by the meta plug-in is called the secondary CNI. SR-IOVs or VDS (Enhanced Data Path) NICs configured for pass-through are managed by secondary CNIs.

While there are several container networking technologies and different ways of deploying them, Telco Cloud and Kubernetes admins want to eliminate manual CNI provisioning and configuration in containerized environments and reduce the overall complexity. Calico or Antrea is often used as the primary CNI plug-in. MACVLAN and Host-devices can be used as secondary CNI together with a CNI meta plug-in such as Multus.

Airgap Server

In the non-air-gapped design, VMware Telco Cloud Automation uses external repositories for Harbor and the PhotonOS packages to implement the VM and NodeConfig operators, new kernel builds, or additional packages to the nodes. Internet access is required to pull these additional components.

The Airgap server is a Photon OS VM that is deployed as part of the telco cloud. The airgap server is then registered as a partner system within the platform and is used in the air-gapped (internet-restricted) environments.

The airgap server operates in two modes:

- **Restricted mode**: This mode uses a proxy server between the Airgap server and the internet. In this mode, the Airgap server is deployed in the same segment as the Telco Cloud Automation VMs in a one-armed mode design.
- **Air-gapped mode**: In this mode, the airgap server is created and migrated to the air-gapped environment. The airgap server has no external connectivity requirements. You can upgrade the airgap server through a new Airgap deployment or an upgrade patch.

Harbor

5G (Core and RAN) consists of CNFs in the form of container images and Helm charts from 5G network vendors. Container images and Helm charts must be stored in an OCI compliant registry that is highly available and easily accessible. Public or cloud-based registries lack critical security compliance features to operate and maintain carrier-class deployments. Harbor addresses these challenges by providing trust, compliance, performance, and interoperability.

VNF and CNF Management

Virtual Network Function (VNF) and Cloud-native Network Function (CNF) management encompasses onboarding, designing, and publishing of the SOL001-compliant CSAR packages to the TCA Network Function catalog. Telco Cloud Automation maintains the CSAR configuration integrity and provides a Network Function Designer for CNF developers to update and release new NF iterations in the Network Function catalog.

Network Function Designer is a visual design tool within VMware Telco Cloud Automation. It generates SOL001-compliant TOSCA descriptors based on the 5G Core or RAN deployment requirements. A TOSCA descriptor consists of CNF instantiation parameters and operational characteristics for life cycle management.

Network functions from each vendor have their unique infrastructure requirements. A CNF developer specifies infrastructure requirements within the CSAR package to instantiate and operate a 5G Core or RAN CNF.

VMware Telco Cloud Automation customizes worker node configurations based on the application requirements by using the VMware Telco NodeConfig Operator. The NodeConfig and VMconfig Operators are Kubernetes operators that manage the node OS, VM customization, and performance tuning. Instead of static resource pre-allocation during the Kubernetes cluster instantiation, the operators defer resource binding of expensive network resources such as SR-IOV VFs, DPDK package installation, and Huge Pages to the CNF instantiation. This allows the control and configuration of each cluster to be bound to the application requirements. This customization is automated by Telco Cloud Automation in a zero-touch process.

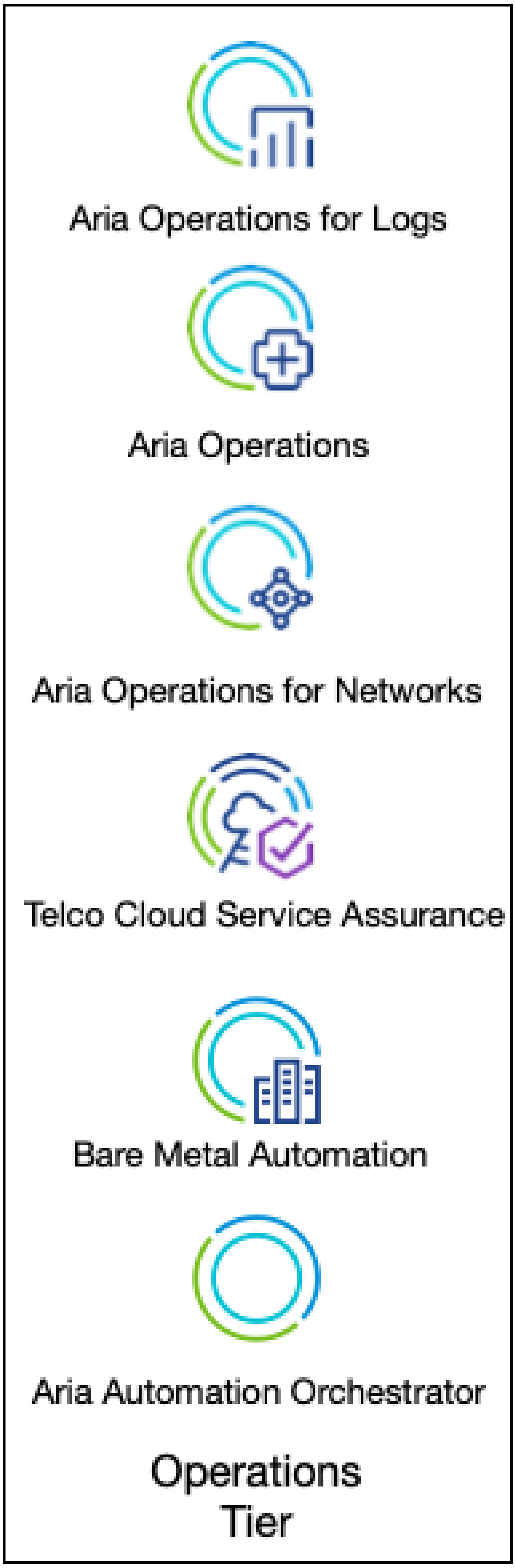
Access policies for CNF Catalogs and Inventory are based on roles and permissions. TCA administrators create custom policies in Telco Cloud Automation to offer self-managed capabilities required by the CNF developers and deployers.

Telco Cloud Operations Tier

The Telco Cloud Operations tier supports centralized data monitoring and logging for the telco cloud solution.

The Telco Cloud Operations tier monitors the Physical, Infrastructure, and Platform tiers. It collects information about various operations to provide observability into the platform efficiency and insights into networking infrastructure and Kubernetes clusters.

Figure 3-9. Telco Cloud Operations Tier


Aria Operations for Logs
Aria Operations
Aria Operations for Networks
Telco Cloud Service Assurance
Bare Metal Automation
Aria Automation Orchestrator
Operations
Tier

Important VMware vRealize suite is being rebranded to VMware Aria suite. Throughout this Telco Cloud Reference Architecture guide, the Aria naming convention is used.

VMware Bare Metal Automation for VMware Telco Cloud Platform automates the provisioning and configuration of physical servers, including server configuration and imaging.

Note The Operations Tier is optional in the Telco Cloud. In addition to existing components such as Aria Operations (formerly vRealize Operations), Aria Operations for Logs (formerly vRealize Log Insight), new components such as Aria operations for Networks (formerly vRealize Network Insight), Telco Cloud Service Assurance, and Bare Metal Automation are added to the Telco Cloud Platform.

Aria Operations for Networks

Aria Operations for Networks is a network monitoring tool that collects and analyzes operational information about network data sources such as NSX, vSphere, Kubernetes deployments, and so on. Users can access this information through dashboards.

Aria Operations for Networks provides capabilities for application discovery, application visibility, and enhanced troubleshooting capabilities by collecting and analyzing inventory, metadata, and flow telemetry of the infrastructure traffic using sFlow/IPFIX. Aria Operations for Networks provides detailed traffic distribution patterns and real-time views of network traffic and patterns.

Major use-cases and benefits of Aria Operations for Networks:

- Application Discovery and visibility
- Security and migration planning
- Visual troubleshooting aids for day 2 operations

Note

- Aria Operations for Networks is an optional component and is leveraged as a SaaS service or on-premise for any Telco Cloud deployment.
- The telco cloud reference architecture supports only the on-premise version of Aria Operations for Networks.

Aria Operations for Logs

Aria Operations for Logs collects unstructured data from the Telco Cloud Platform by using the syslog protocol. It has the following capabilities:

- Connects to other VMware products such as vCenter Server and ESXi hosts to collect events, tasks, and alarm data.
- Integrates with Aria Operations to send notification events and enable launch in context.
- Functions as a collection and analysis point for any system that sends syslog data.

To collect additional logs, you can install an ingestion agent on Linux or Windows servers or use the preinstalled agent on specific VMware products. Preinstalled agents are useful for custom application logs and operating systems such as Windows that do not natively support the syslog protocol.

As the Kubernetes and Container adoptions are increasing in the Telco Cloud, Aria Operations for Logs can also be the centralized log management platform for Tanzu Kubernetes clusters. Cloud Administrators can easily configure container logs to forward to Aria Operations for Logs using industry-standard Open-Source log agents such as FluentD and Fluentbit. Any logs that the container pod writes to standard output (stdout) are sent to Aria Operations for Logs by the log agent, with no changes to the CNF.

Aria Operations

VMware Aria Operations is a unified AI-powered self-driving operations management platform for Private, Hybrid, and Multi-Cloud environments. It tracks and analyzes the operations of multiple data sources using specialized analytic algorithms. These algorithms help Aria Operations learn and predict the behavior of every object it monitors.

Aria Operations supports collecting information through additional management packs. Some management packs are native to Aria Operations and others are add-ons to provide relevant, contextual information about the Telco cloud components. The recommended management packs include:

- vSphere and vSAN
- NSX
- Cloud Director
- Kubernetes Management Pack

Users access the information ingested by Aria operations by using views, reports, and dashboards. Aria Operations is customizable to suit the telco cloud operations requirements, and it supports the creation of custom reports and dashboards.

Bare Metal Automation

VMware Bare Metal Automation is a bare metal provisioning platform used to deploy and bootstrap the server with the appropriate OS, in the case of the Telco Cloud Bare Metal Automation is used to deploy and configure ESXi on the customer server of choice.

In addition to ESXi deployment, Bare Metal Automation performs server BIOS configuration management and ensures the deployment of correct firmware revisions to the BIOS, Network Interface Cards, and other components within the server.

VMware Bare Metal Automation provides and end-to-end server automation. Workflows can be connected to other Telco Cloud components such as VMware Telco Cloud Automation to continue the deployment process after the Bare Metal provisioning process is completed.

Telco Cloud Service Assurance

VMware Telco Cloud Service Assurance is a holistic service assurance solution that allows Communications Service Providers (CSPs) and large enterprises to monitor and manage both the traditional physical infrastructure and new virtual and containerized network services together. The micro-services architecture enables flexibility and scale in VMware Telco Cloud Service Assurance.

Telco Cloud Service Assurance provides end-to-end service assurance capabilities across multiple domains including the Network underlay, the virtualized infrastructure and service level monitoring of the 5G Core and RAN applications

VMware Telco Cloud Service Assurance provides an automated approach to operational intelligence to reduce operational expenses, increase uptime, meet SLAs, and operationalize new services faster. It automatically discovers the topology of a complex, multivendor network including the physical, virtual, and services layers, and presents the user with a comprehensive, graphical topology view.

VMware Telco Cloud Service Assurance provides the following capabilities:

- Single pane of glass providing the CSP Operations teams with rapid insights
- Automated root-cause analysis across service, physical, and virtualized networks
- Auto-discovery of physical and virtual topologies
- Dashboard and reporting
- RAN assurance to consume fault and metric data from the RAN environment
- Closed-loop remediation to take automated actions on infrastructure and service failures
- Data collector SDK to allow the addition of custom CNF collectors
- Observability for TCA pipelines results

Telco Cloud Solution Design

4

This section contains the design and deployment considerations for Telco Cloud Platform. Unlike earlier versions of the Telco Cloud Reference Architecture guides where each stack (Telco Cloud Infrastructure, Telco Cloud Platform 5G Core and Telco Cloud Platform RAN) was treated separately, this revised Telco Cloud Reference architecture guide combines the design elements from all the stacks into a single telco cloud solution design.

Most brownfield deployments of VMware Telco Cloud evolved over time and now incorporate cloud-native functionality and RAN architectures. Greenfield deployments might be skewed toward a particular use case. However, even in a Telco Cloud RAN only deployment, common and foundational components such as the management domain and the compute resource domains exist.

Read the following topics next:

- Telco Cloud Platform End-To-End Deployment Architecture
- Prerequisites for Telco Cloud Service
- Physical Infrastructure Design
- Telco Cloud Infrastructure Design
- Telco Cloud Platform Design
- Telco Cloud Operations Design

Telco Cloud Platform End-To-End Deployment Architecture

This section describes the end-to-end deployment architecture of the Telco Cloud Platform, covering both centralized and multi-site designs. These deployment models encompass IaaS services for 4G/5G VNFs as well as 5G Core and RAN CNFs.

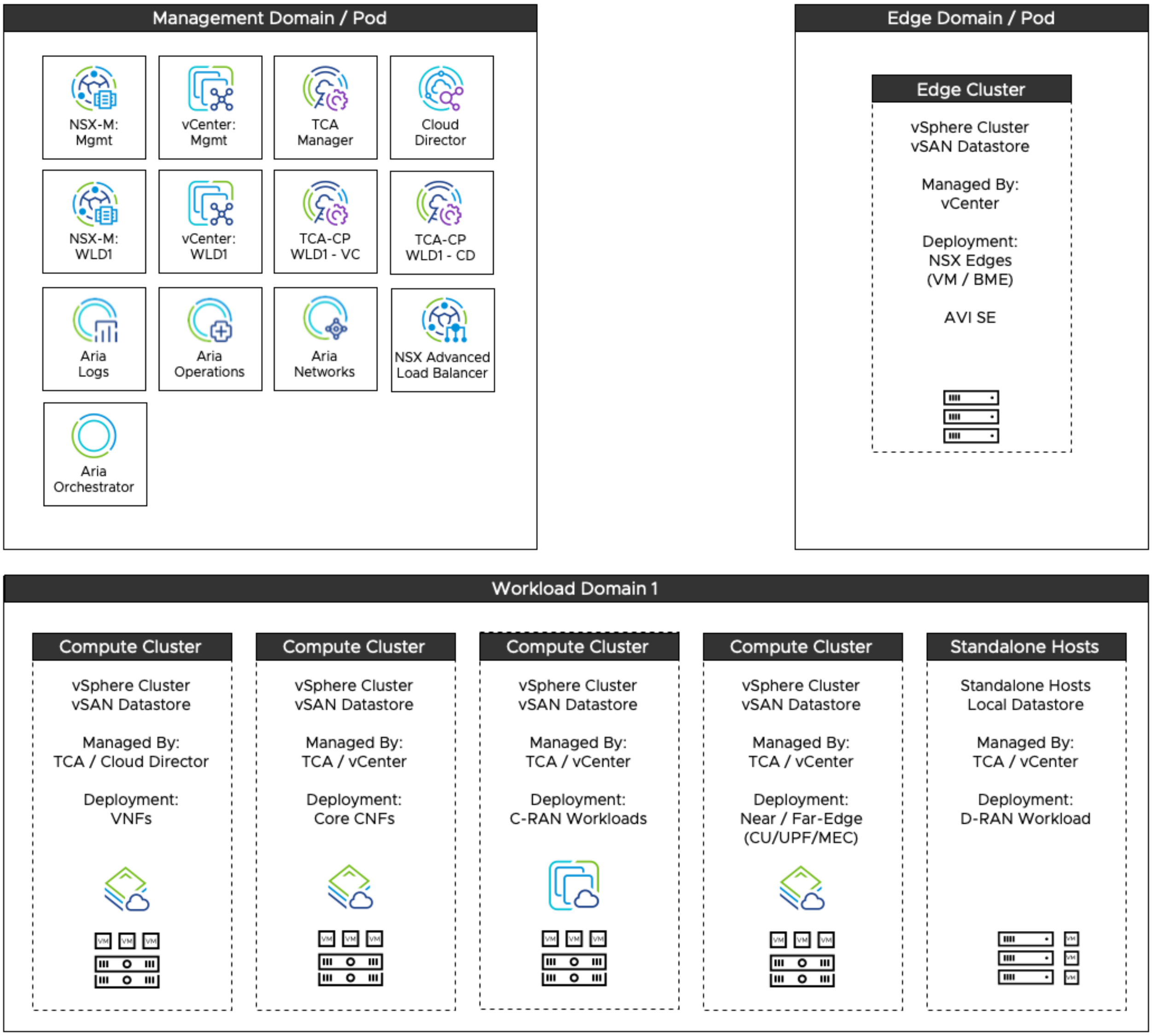
Centralized Deployment Model

In the Centralized deployment model, the Telco Cloud is divided into various domains. Each domain serves a specific purpose to the applications running on the Telco Cloud.

The Management pod / domain hosts all the components necessary for the instantiation and operation of the Telco Cloud. Each workload domain hosts one or more compute environments that allow the placement of workloads such as 4G VNFs, 5G Core and RAN CNFs.

Note Depending on the overall use case and design requirements of the Telco Cloud, the Management domain components vary for each Telco Cloud deployment. For example, a RAN-only deployment does not require components such as NSX and VMware Cloud Director.

Figure 4-1. Centralized Deployment Model of Telco Cloud Platform



In the management domain, a single vCenter manages each workload domain. The expansion of workload domains requires additional vCenter Servers and TCA-CP nodes.

A single management domain is deployed in the main data center (central data center). It can stretch across multiple data centers in an active-active or active-standby configuration leveraging the BCDR solution for management domain failover.

The telco cloud architecture can be scaled out and scaled up across the cloud, within a specific workload domain or across multiple workload domains.

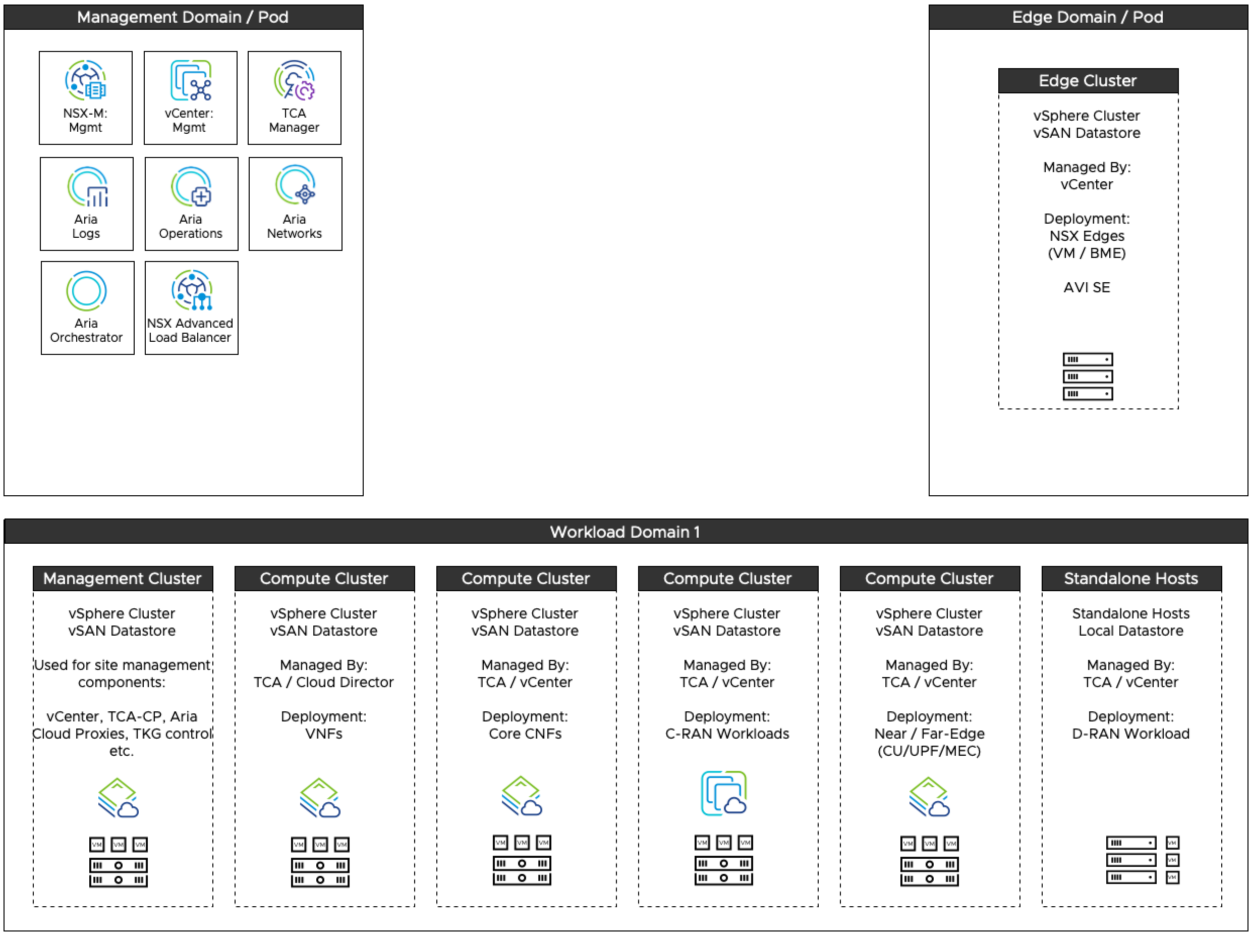
- The telco cloud can be scaled out by adding more workload domains with new vCenter Server, Telco Cloud Control Plane, and NSX deployments.

  Note When adding more workload domains, the Management domain may require additional resources for new vCenter, NSX, and NSX Advanced Load Balancer deployments.

- A single workload domain can be scaled up by adding more compute clusters to be consumed by VMware Cloud Director, VMware Integrated OpenStack, or Telco Cloud Automation, and by adding more cell sites (standalone hosts) for RAN deployments.

Multi-Site Deployment Model

The multi-site deployment model of the telco cloud platform provides a distributed approach to the management pod such that the components can be distributed throughout the network. The central management domain can host the main components of the Telco Cloud. A localized, smaller management domain can be added to each site for independent site management.


Management Domain / Pod
NSX-M: Mgmt
vCenter: Mgmt
TCA Manager
Aria Logs
Aria Operations
Aria Networks
Aria Orchestrator
NSX Advanced Load Balancer
Edge Domain / Pod
Edge Cluster
vSphere Cluster vSAN Datastore
Managed By: vCenter
Deployment: NSX Edges (VM / BME)
AVI SE
Workload Domain 1
Management Cluster
vSphere Cluster vSAN Datastore
Used for site management components:
vCenter, TCA-CP, Aria Cloud Proxies, TKG control etc.
Compute Cluster
vSphere Cluster vSAN Datastore
Managed By: TCA / Cloud Director
Deployment: VNFs
Compute Cluster
vSphere Cluster vSAN Datastore
Managed By: TCA / vCenter
Deployment: Core CNFs
Compute Cluster
vSphere Cluster vSAN Datastore
Managed By: TCA / vCenter
Deployment: C-RAN Workloads
Compute Cluster
vSphere Cluster vSAN Datastore
Managed By: TCA / vCenter
Deployment: Near / Far-Edge (CU/UPF/MEC)
Standalone Hosts
Standalone Hosts Local Datastore
Managed By: TCA / vCenter
Deployment: D-RAN Workload
VM

The following components can be deployed in a localized management cluster within a distributed environment:

- Local vCenter to manage the site resources
- Local NSX Advanced Load Balancer to manage site networking
- Local Telco Cloud Automation-Control Plane for site control
- Local Aria Operations Cloud Proxies for remote data collection and forwarding to the central management domain
- Local Aria Operations for Logs collectors for remote log collection and forwarding to the central management domain

Note Additional components can be deployed in the localized management domain based on the requirements for function distribution and locality of management.

In both the centralized and multi-site deployment designs, it is not necessary to have all workloads on a single site. A workload domain can host multiple compute clusters or RAN nodes from multiple physical locations. The multi-site design applies the concept of workload distribution across multiple DCs to the management components as well.

The characteristics of a management domain in an end-to-end deployment architecture:

- Each workload domain has its own vCenter hosted on the centralized or multi-site management domain.
- Each workload domain has one or more TCA-CPs hosted on the centralized or multi-site management domain.
- Each workload domain has its own NSX manager, if required. NSX is not required for a RAN workload only domain.
- The VIM options must not share a single cluster. Separate clusters are recommended for VNFs, CNFs, or RAN workloads. Sharing a single cluster across multiple VIMs is not recommended.
- Different vSphere switches can be created for different clusters. The vSwitch architecture for RAN sites uses a different vSwitch design for 5G Core and VNF based clusters.
- When using Tanzu Kubernetes Grid, deploy the control-plane nodes in the workload cluster. Alternately, the control-plane nodes can be deployed in a separate cluster that is part of the same vCenter.
- A workload domain does not have to host all cluster types. Depending on the requirements, a workload domain may consist entirely of standalone hosts for RAN workloads (along with a near/far edge cluster for storing the control plane nodes) or the workload domain may consist of vSphere clusters for legacy VNF placement through Cloud Director.

Prerequisites for Telco Cloud Service

This section describes common external services such as DNS, DHCP, NTP, and NFS required for the Telco Cloud.

Various external services are required for the deployment of the Telco Cloud components and Tanzu Kubernetes Grid clusters. If you deploy the Telco Cloud solution in a greenfield environment, you must first deploy your Central Data Center and Management Domain, and then onboard workload domains as required.

The following table lists the required external services and dependencies for the Telco Cloud:

Service	Purpose
Domain Name Services (DNS)	Provides name resolution for various components of the Telco Cloud Platform
Dynamic Host Configuration Protocol (DHCP)	Provides automated IP address allocation for Tanzu Kubernetes clusters throughout the workload domain **Note**: Ensure that the DHCP service is available local to each site for optimal deployment configuration.
Network File System (NFS)	Provides shared storage and data transfer between Cloud Director cells.
Network Time Protocol (NTP)	Performs time synchronization between the Telco Cloud management components

Note LDAP is not a hard requirement, although it is the predominant solution used for providing a centralized user management platform across all components of the telco cloud.

DNS

When you deploy the telco cloud platform, each component from the management domain, including the Tanzu Kubernetes Grid clusters within the domain, require the DNS to be configured for proper addressing through the application FQDN.

DNS resolution must be available for all the components in the solution, including servers, Virtual Machines (VMs), and virtual IPs for Load-Balancer services. Before you deploy the Telco Cloud management components or create workload domains, ensure that both forward and reverse DNS resolutions are created for each component.

DHCP

Dynamic Host Configuration Protocol (DHCP) is required to automatically configure Tanzu Kubernetes Cluster nodes with an IPv4 address. For each Workload domain, DHCP services must be provided locally (for example, through NSX) or remotely from outside the workload domain.

The DHCP scope must be defined and made available to accommodate all the initial and future Kubernetes workloads used in the Telco Cloud Platform.

Note After deploying the control plane nodes, swap the DHCP allocated addresses of the control plane nodes to a static reservation. Thus, the node always receives the same address upon reboot. This is important to maintain kube-vip stability.

For more information about Tanzu Kubernetes Grid IP Addressing, see Tanzu Kubernetes Cluster Design

NTP

All the management components of the Telco Cloud must be synchronized against a common time by using the Network Time Protocol (NTP):

- vCenter Servers and ESXi Host
- NSX Managers and edge nodes
- NSX Advanced Load Balancer and service engines
- Cloud Director cells
- Telco Cloud Automation Manager and Control Plane nodes
- Aria Operations components

Note The Telco Cloud components such as vCenter Server Single Sign-On (SSO) are sensitive to a time drift between distributed components. The synchronized time between various components also assists troubleshooting efforts.

The following guidelines apply to the NTP sources:

- The IP addresses of NTP sources can be provided during the initial deployment of Telco Cloud management components
- The NTP sources must be reachable by all the components in the Telco Cloud Platform.
- Time skew between NTP sources must be less than 5 minutes.

Physical Infrastructure Design

The Physical Infrastructure Tier includes the design of physical ESXi hosts, storage, and networking requirements across the Telco Cloud.

Management Network Overlay and Load Balancing Services

Some services are optional in the Management Domain. For example, NSX-based overlay services are not mandatory in the management domain. However, depending on the components deployed in the management domain, Load-Balancer services might be required.

NSX is not a mandatory component in the management domain. By using NSX in the management domain, the following service can be leveraged:

- Overlay networking for isolation between management components
- Micro-segmentation between management components on a common management subnet.

Note Do not place mission-critical management components on overlay networking. Edge routing or networking issues can prevent access to these components until the issues are resolved.

Load Balancing in the Management Domain

Several applications in the Management domain require a Load Balancer deployed in a highly available configuration:

- Aria Operations: Aria Operations requires a load-balancer to balance requests into the UI or API and for service availability.
- Aria Automation Orchestrator: To provide a high-available Aria Automation Orchestrator deployment, multiple Orchestrator VMs must be front-ended by a load-balancer.
- Cloud Director: To ensure availability in a multi-cell Cloud Director deployment, the service must be front-ended by a load-balancer.
- RabbitMQ: For a high-available external RabbitMQ deployment required for Cloud Director, the service must be front-ended by a load-balancer.

The recommended Load-Balancer service for the management domain is VMware NSX Advanced Load Balancer (AVI). For more information about the Load-Balancer design, see Load Balancer Design - NSX Advanced Load Balancer.

vSphere Host and Cluster Design

This section describes the physical specifications of ESXi servers used to create workload clusters or standalone hosts (for RAN deployments) for the successful deployment and operation of the Telco Cloud.

Physical Design Specification Fundamentals

The configuration and assembly process for each system is standardized, with all components installed in the same way on each ESXi host. The standardization of the physical configuration across the ESXi hosts helps you operate an easily manageable and supportable infrastructure. This standardization applies to the ESXi hosts for each cluster in the workload domain. Components of each workload domain might have different physical requirements based on the applications.

As an example, RAN Distributed Unit (DU) workloads deployed to standalone hosts leverage look-aside or in-line accelerator cards. Deploying these PCI cards to Core or VNF workload pods residing in the same workload domain is expensive and unnecessary .

Therefore, ESXi hosts in a cluster must have identical configurations, including storage and networking configurations. For example, consistent PCI card slot placement, especially for network controllers, is essential for the accurate alignment of physical to virtual I/O resources. By using identical configurations, you can balance the VM storage components across storage and compute resources.

The sizing of physical servers running ESXi requires special considerations depending on the workload. Traditional clusters use vSAN as the primary storage system, hence the vSAN requirements must be considered.

The section outlines the generic and specific recommendations for each workload type such as 5G Core and RAN.

ESXi Host Memory

The amount of memory required for vSphere compute clusters varies according to the workloads running in clusters. When sizing the memory of hosts in a compute cluster, consider the admission control setting (n+1) that reserves the resources of a host for failover or maintenance. In addition, leave a memory budget of 8-12 GB for ESXi host operations.

The number of vSAN disk groups and disks managed by an ESXi host determines the memory requirements. To support the maximum number of disk groups, up to 100 GB RAM is required for vSAN. For more information about the vSAN configuration maximums, see VMware Configuration Maximums.

ESXi Boot Device

The following considerations apply when you select a boot device type and size for vSAN:

- vSAN does not support stateless vSphere Auto Deploy.
- Device types supported as ESXi boot devices:
  - SATADOM devices. The size of the boot device per host must be at least 32 GB.
  - USB or SD embedded devices. The USB or SD flash drive must be at least 8 GB.

vSphere 8.0 is deprecating the ability to boot from SD disks. A dedicated boot disk with at least 128 GB is required to include optimal support for ESX-OSData partitions. This local disk can also be RAID-1 Mirrored to provide boot disk resiliency; this is handled at the server layer.

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Use vSAN ready nodes.	Ensures full compatibility with vSAN	Hardware choices might be limited.	Management domain, Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX-Edge. Not applicable to RAN sites.
All ESXi hosts must have a uniform configuration across a cluster in a workload environment.	A balanced cluster has more predictable performance even during hardware failures. In addition, the impact on performance during re-sync or rebuild is minimal.	As new models of servers become available, the deployed model phases out. So, it becomes difficult to keep a uniform cluster when adding hosts later.	Management domain, Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX edge. For RAN sites, ensure that all RAN sites in a workload domain have uniform configuration.
Set up the management cluster with a minimum of four ESXi hosts.	Provides full redundancy for the management cluster when vSAN is used	Additional ESXi host resources are required for redundancy.	Management Domain
Set up the workload and edge clusters with a minimum of four ESXi hosts.	Provides full redundancy for the workload cluster when vSAN is used	Additional ESXi host resources are required for redundancy. If Not using vSAN the clusters can be minimally sized according to the workload requirements.	Compute clusters, VNF ,CNF ,C-RAN, Near/Far Edge, and NSX Edge deployments
Set up each ESXi host in the management cluster with a minimum of 256 GB RAM.	Ensures that the management components have enough memory to run during a single host failure. Provides a buffer for future management or monitoring of components in the management cluster.	In a four-node cluster, only the resources of three ESXi hosts are available as the resources of one host are reserved for vSphere HA. Depending on the products deployed and their configuration, more memory per host (or more hosts) might be required.	Management Domain

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Size the RAM on workloads clusters/ hosts appropriately for the planned applications	Ensures that the Network Functions have enough memory to run during a single host failure. Provides a buffer for scaling or additional network functions to be deployed to the cluster.	In a four-node cluster, only the resources of three ESXi hosts are available as the resources of one host are reserved for vSphere HA. Depending on the Functions deployed and their configuration, more memory per host might be required.	Compute Clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge deployments
Use a disk or redundant Boot Optimized Server Storage unit as the ESXi boot disk.	Enables a more stable deployment than USB or SD cards Stores coredumps and rotating logs	Requires additional disks for the ESXi boot disk	All domains

Management Cluster

The management cluster runs multiple VMs, from multiple instances of vCenter or NSX to scaled deployments of Aria Operations, Aria Operations for logs, and so on.

When deploying new functions to the management domain, ensure that the management cluster has the failover capacity to handle host component failures or downtime caused by host upgrades.

Compute Clusters

The compute cluster sizing depends on various factors. Control plane and user plane workloads are deployed to unique compute clusters (different vSphere clusters). To maximize the overall performance, the user plane workloads require more resources for features such as Latency Sensitivity, NUMA awareness, and infrastructure awareness.

A compute pod design must be rack-aligned with 16/24/32 servers per rack, depending on the server RU size, the power available to the rack, and so on.

Network Edge Clusters

Network Edge Clusters are smaller in size than the Management or Core Compute clusters. The core compute clusters consume an entire rack. The network edge cluster primarily hosts NSX Edge Gateway VMs and NSX Advanced Load Balancer Service Gateways. It must be sized according to the edge domain requirements.

Near/Far Edge Clusters

Near/Far Edge clusters are smaller in size than the Management or Core Compute clusters. The Near/Far Edge clusters are used in a distributed environment. The use cases for this cluster type include:

- C-RAN Deployments (hosting multiple DUs in a single location)

- Edge services (UPF Breakout)
- Multi-Access Edge Computing (MEC) use cases for enterprise verticals or co-located offerings

The hardware that is installed in the Near/Far Edge clusters might also differ from the general-purpose compute based on the service offerings. GPUs can be deployed for AI/ML services and inline/look-aside accelerators can be deployed for C-RAN type services.

Note The standardization of physical specifications across the cluster applies to all cluster types.

RAN Cell Sites

RAN Cell Sites run fewer workloads than other cluster types; however, these workloads require low latency and more resources in real time.

ESXi Host memory must be sized to accommodate both the Tanzu Kubernetes Grid RAN worker nodes and ensure that memory is available for the hypervisor. Standalone hosts do not require RAM overhead for vSAN.

The minimum memory reservation for RAN worker nodes is 12 GB RAM and 2 physical cores per NUMA node. Remove these resources from the overall host capacity when performing function sizing and capacity planning and include overhead for ESXi and the Guest OS or Kubernetes processing requirements.

Physical Network Design

The physical network design includes defining the network topology for connecting physical switches and the ESXi hosts, determining switch port settings for VLANs, and designing routing or services architecture.

Top-of-Rack Physical Switches

When configuring Top-of-Rack (ToR) switches, consider the following best practices:

- Configure redundant physical switches to enhance availability.
- Configure switch ports that connect to ESXi hosts manually as trunk ports. Virtual switches are passive devices and do not support trunking protocols such as Dynamic Trunking Protocol (DTP).
- Modify the Spanning Tree Protocol (STP) on any port that is connected to an ESXi NIC to reduce the time it takes to transition ports over to the forwarding state, for example, using the Trunk PortFast feature on a Cisco physical switch.
- Configure jumbo frames on all switch ports, Inter-Switch Link (ISL), and Switched Virtual Interfaces (SVIs).

Top-Of-Rack connectivity and Network Settings

Each ESXi host is connected redundantly to the network fabric ToR switches through various physical interfaces. For information about the design and number of these interfaces, see the Network Virtualization Design section.

The ToR switches are configured to provide all necessary VLANs through an 802.1Q trunk. These redundant connections use the features of vSphere Distributed Switch to guarantee that a physical interface is not overrun and redundant paths are used if they are available.

- **Leaf-Spine architecture**: Modern data centers are deployed with a leaf-spine based architecture, where each leaf switch is connected to all available spine switches, resulting in an efficient configuration with maximum redundancy and load sharing.
- **Spanning Tree Protocol** (STP): Although the recommended design does not use STP, switches usually include STP configured by default. Designate the ports connected to ESXi hosts as trunks and utilize specific features such as trunk portfast to reduce uplink convergence.
- **Trunking**: Configure the switch ports as members of an 802.1Q trunk.
- **MTU**: Set MTU for all switch ports, VLANs, and SVIs to support jumbo frames for consistency. The host DC is typically configured to an MTU of 9100.

Jumbo Frames

IP storage, network function payload, and other services can benefit from the configuration of jumbo frames. Increasing the per-frame payload from 1500 bytes to a jumbo frame setting improves the efficiency of data transfer. Jumbo frames must be configured end-to-end. When enabling jumbo frames on an ESXi host, select an MTU size that matches the MTU size of the physical switch ports.

The workload determines whether to configure jumbo frames on a VM. If the workload consistently transfers large amounts of network data, configure jumbo frames. Also, ensure that both the VM operating system and the VM NICs support jumbo frames. Jumbo frames also improve the performance of vSphere vMotion.

Note The vSwitches are configured to support Jumbo Frames (9000 MTU), while the vMotion and vSAN VMKernel ports benefit from increased throughput by using Jumbo Frames. To avoid MTU or MSS mismatches from management components to the ESXi host management interface, leave the Management VMKernel port (VMK0) with the default MTU (1500).

Recommended Physical Network Design

Design Recommendation	Design Justification	Design Implication
Use the Layer 3 Leaf-Spine Data Center Architecture.	You can select layer 3 switches from different vendors for the physical switching fabric. This approach is cost-effective because it uses only the basic functionality of the physical switches.	VLANs are restricted to a single rack. VLANs can be reused across racks without spanning across domains.
Implement the following physical network architecture: ■ Two physical interfaces, one per NUMA node, on each ToR switch for ESXi host uplinks. ■ Layer 3 device such as a Data Center Gateway with BGP support	■ Guarantees availability during a switch failure ■ Provides compatibility with vSphere host profiles because they do not store link-aggregation settings ■ Supports BGP as the dynamic routing protocol ■ BGP is the only dynamic routing protocol supported by NSX.	Hardware choices might be limited.
Use two ToR switches for each rack.	This design uses multiple physical interfaces per NUMA node on each ESXi host. Provides redundancy and reduces the overall design complexity.	Two ToR switches per rack can increase costs.
Use VLANs to segment physical network functions.	■ Supports physical network connectivity without requiring many NICs. ■ Isolates different network functions of the Software-Defined Data Center (SDDC) so that you can have differentiated services and prioritized traffic as needed.	Requires uniform configuration and presentation on all the switch ports made available to the ESXi hosts.
Assign static IP addresses to all management components.	Ensures that interfaces such as management and storage always have the same IP address. In this way, you provide support for continuous management of ESXi hosts using vCenter Server and for provisioning IP storage by storage administrators.	Requires precise IP address management.
Create DNS records for all ESXi hosts and management VMs to enable forward, reverse, short, and FQDN resolution.	Ensures consistent resolution of management components using both IP address (reverse lookup) and name resolution.	Adds administrative overhead.

Design Recommendation	Design Justification	Design Implication
Use an NTP or Precision Time Protocol time source for all management components.	It is critical to maintain accurate and synchronized time between management components.	None
Configure the MTU size to at least 9000 bytes (jumbo frames) on the physical switch ports, VLANs, SVIs, vSphere Distributed Switches, and VMkernel ports.	Improves traffic throughput. A minimum of 1700 MTU is required for NSX Data Center deployments.	When you adjust the MTU size, you must also configure the entire network path (VMkernel port, distributed switch, physical switches, and routers) to support the same MTU size.

Physical Storage Design

The Physical Storage Design uses vSAN to implement software-defined storage as the primary storage type. Different domains or workload clusters can implement other storage options depending on the requirements and constraints.

vSAN is a hyper-converged storage solution built within the ESXi hypervisor. vSAN (Original Storage Architecture) creates disk groups consisting of hard disk drives and flash devices or all-flash devices in the local ESXi host. It provides a highly resilient shared storage datastore to the vSphere Cluster.

By using vSAN storage policies, you can control capacity, performance, and availability on a per virtual disk basis.

While vSAN is the recommended solution, the physical storage can be based on the extensive list of supported storage providers. This reference architecture focuses on vSAN requirements from a physical storage perspective.

Caution When using a storage platform, evaluate the supported use cases across the telco cloud, including file and block storage and cloud-native support considerations such as Read-Write Many persistent volumes.

Table 4-1. vSAN Infrastructure Requirements

Category	Requirements
Number of ESXi hosts	■ Minimum of 3 ESXi hosts providing storage resources to the vSAN cluster. This can be 3 ESXi hosts or 2 ESXi hosts and 1 vSAN witness. ■ Minimum of 4 ESXi hosts for vSAN automated rebuild. ■ Maximum 200 VMs per host in a vSAN cluster.
vSAN configuration	vSAN can be configured in all-flash or hybrid mode: ■ **All-flash mode**: All-flash vSAN configuration requires flash devices for both the caching and capacity tiers. ■ **Hybrid mode**: vSAN hybrid storage configuration requires both magnetic devices and flash caching devices.
Individual ESXi hosts that provide storage resources.	■ Minimum of one flash device. The flash cache tier must be at least 10% of the size of the capacity tier. ■ Minimum of two HDDs for hybrid mode, or an additional flash device for an all-flash configuration. ■ RAID controller that is compatible with vSAN. ■ Minimum 10 Gbps network for vSAN traffic or 25 Gbps when using the vSAN ESA architecture. ■ Host isolation response of vSphere High Availability is set to power off VMs.

vSAN Disk Groups

Disk group sizing is an important factor in the vSAN design. If more ESXi hosts are available in a cluster, more failures are tolerated in the cluster. This capability adds cost because additional hardware is required for the disk groups.

More available disk groups can increase the recoverability of vSAN during a failure. When deciding on the number of disk groups per ESXi host, consider these points:

- Amount of available space on the vSAN datastore
- Number of failures that can be tolerated in the cluster

The optimal number of disk groups is a balance between the hardware and space requirements for the vSAN datastore. More disk groups increase space and provide high availability but it can be expensive.

Note The disk groups concept is not applicable for the vSAN Express Storage Architecture.

Storage Controllers

The storage I/O controllers are as important as the selection of disk drives to a vSAN configuration. vSAN supports SAS, SATA, and SCSI adapters in either pass-through or RAID 0 mode. vSAN supports multiple controllers per ESXi host.

- **Multi-Controller Configuration**: Multiple controllers can improve performance and mitigate a controller or SSD failure to a smaller number of drives or vSAN disk groups.

- **Single-Controller Configuration**: With a single controller, all disks are controlled by one device. A controller failure impacts all storage, including the boot media (if configured).

Controller queue depth is an important aspect of performance. All I/O controllers in the VMware vSAN Hardware Compatibility Guide have a minimum queue depth of 256. If you increase the queue depth to a value higher than 256, ensure that you consider the regular day-to-day operations in your environment. Examples of events that require higher queue depth are as follows:

- VM deployment operations
- Re-sync I/O activity because of automatic or manual fault remediation

Physical Storage Recommendations

Design Recommendation	Design Justification	Design Implication
Use all-flash (SSD / NVMe) vSAN in all vSphere clusters.	Provides the best performance with low latency. When using all-flash vSAN, you can enable de-duplication and compression that saves space on the datastores.	Flash storage costs more than traditional magnetic disks.
For the management cluster, provide a vSAN configuration with at least 6 TB of usable space.	Provides all the required space for this solution while allowing the deployment of additional monitoring and management components in the management cluster.	More space is required on day 1.
For the edge cluster, provide a vSAN configuration with at least 500 GB of usable space.	Provides required storage to run NSX Edge Nodes and NSX ALB Service Engines.	None
For the compute clusters, size the vSAN datastore according to the current workloads plus five years of expected growth.	Ensures that the storage solution is not required to be upgraded as it can cause downtime to workloads.	More space is required on day 1.
If using vSAN do not use HCI Mesh or vSAN File Services	HCI Mesh and vSAN FS are not supported with vSAN using ESA architecture.	

Telco Cloud Infrastructure Design

The platform design includes software components for providing software-defined storage, networking, and compute. These components include the hypervisor, virtualization management, storage virtualization, and network virtualization.

The Telco Cloud Infrastructure tier provides the foundation capabilities for all workload type such as 4G, 5G Core, and RAN. The Telco Cloud Infrastructure tier is leveraged primarily for VNF-based workloads. However, the common, horizontal Telco Cloud is built from the foundational building blocks of the Infrastructure tier.

vCenter Server Design

The vCenter Server design encompasses all the vCenter Server instances, including the number of instances, their sizes, networking configuration, vSphere cluster layout, redundancy, and security configuration.

According to the site design, overall scale, number of VMs, and continuity requirements for your environment, a vCenter Server deployment for the Telco Cloud consists of two or more vCenter Server instances with one vCenter for the Management Domain and at least one additional vCenter for the workload domain.

The vCenter Server system is the central point of management and monitoring. Use the following methods to protect vCenter Server according to the maximum downtime tolerated:

- Automated protection using vSphere HA
- Automated protection using vCenter Server HA

vCenter Server Sizing

You can size the resources and storage for the Management vCenter Server Appliance and the Compute vCenter Server Appliance according to the expected number of VMs in the environment.

Table 4-2. Recommended Sizing for the Management vCenter Server

Attribute	Specification
Appliance Size	Small (up to 100 hosts or 1000 VMs)
Number of vCPUs	4
Memory	21 GB
Disk Space	579 GB

Table 4-3. Recommended Sizing for Workload Domain vCenter Servers

Attribute	Specification
Appliance Size	X-Large (up to 2,000 hosts or 35,000 VMs)
Number of vCPUs	24
Memory	59 GB
Disk Space	2,283 GB

Note vCenter sizing depends on the site or workload domain scale.

TLS Certificates in vCenter Servers

By default, vSphere uses TLS or SSL certificates that are signed by VMware Certificate Authority (VMCA). These certificates are not trusted by end-user devices or browsers. As a security best practice, replace at least all user-facing certificates with certificates that are signed by a third-party or enterprise Certificate Authority (CA).

vCenter RAN Considerations

In a RAN-only deployment, the ESXi hosts are added to vCenter as standalone hosts. Scaling of RAN deployments is significant in a domain.

Using dedicated vCenter Servers for RAN deployments with a high host count has the benefit of separating lifecycle management of RAN and Core workload domains. This approach requires additional vCenter deployments and is determined based on the overall Telco Cloud design considerations and constraints.

Recommended vCenter Designs

Design Recommendation	Design Justification	Design Implication
Deploy at least two vCenter Server systems: - One vCenter Server supports the management workloads - Another vCenter Server supports the compute workloads	- Isolates vCenter Server failures to management or compute workloads. - Isolates vCenter Server operations between management and compute workloads. - Supports a scalable vSphere cluster design where you might reuse the management components as more compute workload domains are added. - Simplifies capacity planning for compute workloads because you do not consider management workloads for the Compute vCenter Server. - Improves the ability to upgrade the vSphere environment and related components by separating the maintenance windows. - Supports separation of roles and responsibilities to ensure that only authorized administrators can handle the management workloads. - Facilitates quicker troubleshooting and problem resolution.	- Requires licenses for each vCenter Server instance. - Deployment location of vCenter Server depends on the centralized or multi-site deployment design.
Protect all vCenter Servers by using vSphere HA.	Supports the availability objectives for vCenter Server without the required manual intervention during a failure event.	vCenter Server becomes unavailable during the vSphere HA failover.
Replace the vCenter Server machine certificate with a certificate signed by a third-party Public Key Infrastructure.	Infrastructure administrators connect to the vCenter Server instances using a web browser to perform configuration, management, and troubleshooting. The default certificate results in certificate warning messages.	Replacing and managing certificates is an operational overhead.

Workload Domains and vSphere Cluster Design

The vCenter Server functionality is distributed across a minimum of two workload domains and two vSphere clusters. The telco cloud solution uses two multiple Server instances: one for the management domain and another for the first compute workload domain. The compute workload domain can contain multiple cell site ESXi hosts.

When designing a cluster for workloads, consider the types of workloads that the cluster handles. Different cluster types have different characteristics.

When designing the cluster layout in vSphere, consider the following guidelines:

- Use a few large-sized ESXi hosts or more small-sized ESXi hosts for Core clusters
  - A scale-up cluster has few large-sized ESXi hosts.
  - A scale-out cluster has more small-sized ESXi hosts.
- Use the ESXi hosts that are sized appropriately for your Cell Site locations.
- Consider the total number of ESXi hosts and cluster limits as per vCenter Server maximums.
- Consider the applications deployed to the cluster or workload domain.
- Consider additional space for rolling upgrades of Kubernetes nodes.
- Consider HA constraints and requirements for host outages

In a D-RAN scenario, hosts are not added to a vSphere cluster. When deployed through Telco Cloud Automation Cell Site Groups, the hosts are added into a folder structure aligned with the Cell Site Grouping in vCenter and not as a member of the cluster.

In a C-RAN deployment, depending on the workload deployments (DU / CU), a combination of standalone hosts and clusters may be necessary to accommodate availability requirements.

vSphere High Availability

If an ESXi host failure, vSphere High Availability (vSphere HA) protects VMs by restarting them on other hosts in the same cluster. During the cluster configuration, the ESXi hosts elect a primary ESXi host. The primary ESXi host communicates with the vCenter Server system and monitors the VMs and secondary ESXi hosts in the cluster.

The primary ESXi host detects different types of failure:

- ESXi host failure, for example, an unexpected power failure
- ESXi host network isolation or connectivity failure
- Loss of storage connectivity
- Problems with the virtual machine OS availability

The vSphere HA Admission Control Policy allows an administrator to configure how the cluster determines available resources. In a small vSphere HA cluster, a large proportion of the cluster resources is reserved to accommodate ESXi host failures, based on the selected policy.

Note As D-RAN deployments do not leverage vSphere clusters, the vSphere HA is not a consideration on the D-RAN deployment. If the cell site architecture deploys more than one ESXi host, they are added as individual hosts and not as a cluster.

The following vSphere HA policies are available for workload clusters:

- **Cluster resource percentage**: Reserves a specific percentage of cluster CPU and memory resources for recovery from host failures. With this type of admission control, vSphere HA ensures that a specified percentage of aggregate CPU and memory resources is reserved for failover.
- **Slot policy**: vSphere HA admission control ensures that a specified number of hosts can fail and sufficient resources remain in the cluster to failover all the VMs from those hosts.
  - A slot is a logical representation of memory and CPU resources. By default, the slot is sized to meet the requirements for any powered-on VM in the cluster.
  - vSphere HA determines the current failover capacity in the cluster and leaves enough slots for the powered-on VMs. The failover capacity specifies the number of hosts that can fail.
- **Dedicated failover hosts**: When a host fails, vSphere HA attempts to restart its VMs on any of the specified failover hosts.

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Use vSphere HA to protect all VMs against failures.	vSphere HA provides a robust level of protection for VM availability.	You must provide sufficient resources on the remaining hosts so that VMs can be migrated to those hosts in the event of a host outage.	Management domain, Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites
Set the Host Isolation Response of vSphere HA to Power Off and Restart VMs.	vSAN requires that the HA Isolation Response is set to Power Off and the VMs are restarted on available ESXi hosts.	VMs are powered off in case of a false positive and an ESXi host is declared isolated incorrectly.	Management domain, Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites

vSphere Distributed Resource Scheduler

The distribution and usage of CPU and memory resources for all hosts and VMs in the cluster are monitored continuously. The vSphere Distributed Resource Scheduler (DRS) compares these metrics to an ideal resource usage based on the attributes of the cluster's resource pools and VMs, the current demand, and the imbalance target. DRS then provides recommendations or performs VM migrations accordingly.

DRS supports the following modes of operation:

- Manual
  - Initial placement: Recommended host is displayed.
  - Migration: Recommendation is displayed.
- Partially Automated
  - Initial placement: Automatic
  - Migration: Recommendation is displayed.

- Fully Automated
  - Initial placement: Automatic
  - Migration: Recommendation is run automatically.

Note

- The configuration of DRS modes can vary between clusters. Some workloads or applications may not fully support automated vMotioning of VMs that occurs when DRS is optimizing the cluster.
- Due to the nature of the host deployments and the Platform Awareness functionality leveraged in the RAN (SR-IOV or Accelerator pass-through), DRS is not a consideration for RAN deployments.

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Enable vSphere DRS in the management cluster and set it to Fully Automated, with the default setting (medium).	Provides the best trade-off between load balancing and excessive migration with vSphere vMotion events.	If a vCenter Server outage occurs, mapping from VMs to ESXi hosts might be difficult to determine.	Management domain
Enable vSphere DRS in the edge and compute clusters and set it to Partially Automated mode.	Enables automatic initial placement Ensures that the latency-sensitive VMs do not move between ESXi hosts automatically.	Increases the administrative overhead in ensuring that the cluster is properly balanced.	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites

Resource Pools

A resource pool is a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies and used to hierarchically partition available CPU and memory resources.

Each DRS cluster or standalone host has an invisible root resource pool that groups the resources of that cluster. The root resource pool does not appear because the resources of the cluster and the root resource pool are always the same.

You can create child resource pools from the root resource pool. Each child resource pool owns some of the parent's resources and can, in turn, have a hierarchy of child resource pools to represent successively smaller units of computational capability.

A resource pool can contain child resource pools, VMs, or both. You can create a hierarchy of shared resources. The resource pools at a higher level are called parent resource pools. Resource pools and VMs that are at the same level are called siblings. The cluster represents the root resource pool. If you do not create child resource pools, only the root resource pools exist.

Scalable Shares allows the resource pool to dynamically scale as VMs are added or removed from the resource pool hierarchy.

The resource pool is a key component for both Cloud Director and Tanzu Kubernetes Grid deployments. The Organizational Virtual Data Centers (OrgVDCs) that are created by VMware Cloud Director are manifested as resource pools on vCenter, depending on the configuration of the components in Cloud Director. The resource pool can be configured across multiple clusters and the placement decision is made by Cloud Director level to determine which resource pool to deploy into.

For Tanzu Kubernetes Grid, the resource pool is the target endpoint for a nodepool of the Tanzu Kubernetes cluster. Control plane and node pools can be deployed to separate vSphere clusters and resource pools, but a single node pool or the control nodes cannot be deployed across different endpoints.

VMware Telco Automation does not create resource pools as with Cloud Director. Therefore, the resource pools for Tanzu Kubernetes Grid deployments must be manually created and sized with appropriate limits and reservations on vCenter Server before the Tanzu Kubernetes cluster creation.

Note When vSphere resources are added to Cloud Director as resource pools (instead of the entire cluster), resource pools for Cloud Director and resource pools for Tanzu Kubernetes Grid can exist on the same cluster. This level of cluster sharing is not recommended.

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Do not share a single vSphere cluster between Cloud Director and TKG resources	Provides a better isolation between VNF and CNFs within a single vCenter	Requires separate cluster for VNF and CNF workloads	Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX-Edge
Create TKG resource pools and allocate appropriate resources for each network functions	Ensures proper admission control for TKG clusters hosting control-plane workloads.	Requires understanding of the application sizing and scale considerations to effectively configure the resource pool	Compute clusters. VNF, CNF, C-RAN, Near/Far Edge, and NSX-Edge The root resource pool can be used for D-RAN deployments.

Workload Cluster Scale

Workload cluster scale depends on the management architectural design. In both the centralized management domain and multi-site configuration, independent vCenter Servers can manage a domain or site.

Additional clusters can be added to the resource vCenter, either at the same location or remote. The definition of a 'site' is different from a workload domain. A workload domain can spread across multiple locations. The Near/Far Edge locations must be managed by a vCenter within the local workload domain.

In an NSX design, NSX Manager and vCenter Server must have 1:1 mapping. NSX Manager manages not only the local or co-located environment but also remote locations connected to the domain.

Network Virtualization Design

The network virtualization design uses the vSphere Distributed Switch (VDS) and associated features such as the Converged VDS (C-VDS) through the NSX and vSphere integration.

Network Virtualization Design Goals

The following high-level design goals apply regardless of your environment:

- **Meet diverse needs**: The network must meet the diverse requirements of different entities in an organization. These entities include applications, services, storage, administrators, and users.
- **Reduce costs**: Server consolidation reduces network costs by reducing the number of required network ports and NICs, but a more efficient network design is required. For example, configuring two 25 GbE NICs might be more cost-effective than four 10 GbE NICs.
- **Improve performance**: You can achieve performance improvement and decrease the maintenance time by providing sufficient bandwidth, which in turn reduces the contention and latency.
- **Improve availability**: A well-designed network improves availability by providing network redundancy.
- **Support security**: A well-designed network supports an acceptable level of security through controlled access and isolation, where required.
- **Enhance infrastructure functionality**: You can configure the network to support vSphere features such as vSphere vMotion, vSphere High Availability, and vSphere Fault Tolerance.

Note In specific use-cases such as D-RAN deployments, the support of enhanced infrastructure functionality may be different. In single-node deployments, elements such as vMotion and vSphere HA are removed in favor of network performance and throughput.

Network Virtualization Best Practices

The following network virtualization best practices can be considered throughout your environment:

- Separate the network services to achieve high security and better performance. This requires multiple vSwitches to differentiate between management functions and workload traffic.
- Use the Network I/O Control and traffic shaping to guarantee bandwidth to critical VMs. During the network contention, these critical VMs receive a high percentage of the bandwidth.

  Note Do not use this best practice for user plane telco workloads as it might increase latency and impact overall throughput.
- Separate the network services on a vSphere Distributed Switch by attaching them to port groups with different VLAN IDs.

- Keep vSphere vMotion traffic on a separate network. When a migration using vSphere vMotion occurs, the memory of the guest operating system is transmitted over the network. You can place vSphere vMotion on a separate network by using a dedicated vSphere vMotion VLAN and a dedicated Telco Cloud Platform stack for vMotion.
- Ensure that physical network adapters that are connected to the same vSphere Standard or Distributed Switch are also connected to the same physical network.

Network Virtualization Segmentation

In many cases, separating different types of traffic is recommended for access security and to reduce the contention and latency.

High latency on any network can have a negative impact on the performance. Some components are more sensitive to high latency than others. For example, high latency IP storage and the vSphere Fault Tolerance logging network can negatively affect the performance of multiple VMs.

While a single vSwitch design can deliver the appropriate segmentation, segmentation at the vSwitch level for management functions provides increased resiliency to the platform and maximizes overall throughput at the cost of additional NICs or ports. By segmenting the following management functions, the platform can maximize the overall throughput of the infrastructure:

- ESXi Management
- vMotion
- vSAN (OSA or ESA)
- IP Storage

According to the application or service, high latency on specific VM networks can also negatively affect performance. Determine which workloads and networks are sensitive to high latency by using the information gathered from the current state analysis and by interviewing key stakeholders and SMEs. Determine the required number of network cards, ports, uplinks, and VLANs depending on the type of traffic.

vSphere Distributed Switching

The core of any virtualized infrastructure is the networking that backs the VMs and the ESXi hosts.

Within vSphere environment, VMware provides two types of virtual switches: standard switch and vSphere-managed Virtual Distributed Switch (vSphere VDS). The Converged VDS that integrates NSX and vSphere Distributed Switch functionalities is based on the underlying vSphere VDS.

The vSphere Distributed Switch functions as a single switch to which all hosts in a workload domain are connected. This architecture allows network configurations to be consistently applied across all attached hosts.

The vSphere Distributed Switch comprises two components:

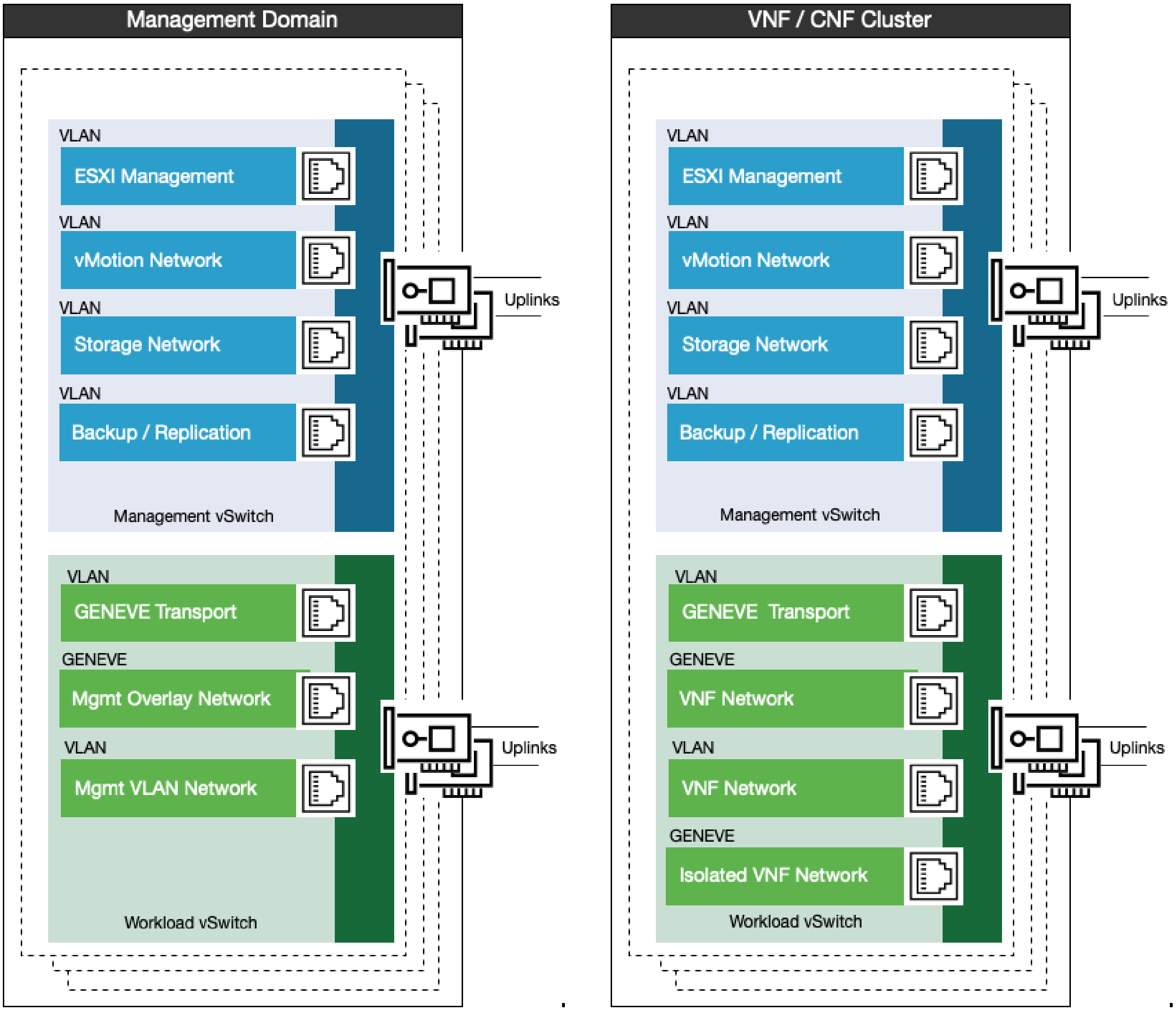
- **Port groups**: The port groups, to which the VMs are connected, enable inter-VM communication. Each port group can have its own configuration such as Port Group name, VLAN, teaming policies, and so on.
- **Uplink adapters**: The uplink adapters determine the physical interfaces that are used to connect a VDS to the underlay network. The uplink configuration can incorporate elements such as Link Aggregation Control Protocol (LACP).

vSphere Distributed Switch instances offer several enhancements compared to the legacy standard virtual switches. Because vSphere Distributed Switch instances are centrally created and managed, the virtual switch configuration can be made consistent across ESXi hosts. Centralized management saves time, reduces mistakes, and lowers operational costs.

The number of vSwitches deployed throughout the Telco cloud depends on various factors such as design constraints, physical NIC or port count, throughput requirements, network redundancy requirements, and so on.

The following vSwitch design shows the dual vSwitch model for the management and general VNF/CNF workload clusters.

Figure 4-2. Dual vSwitch Model



In this dual vSwitch model, the management vSwitch is used for host management, with VLAN for ESXi, vMotion, Storage (vSAN or IP storage) and backup/replication. The workload vSwitch is used only for workload traffic, a Similar design can also be used for the edge pods.

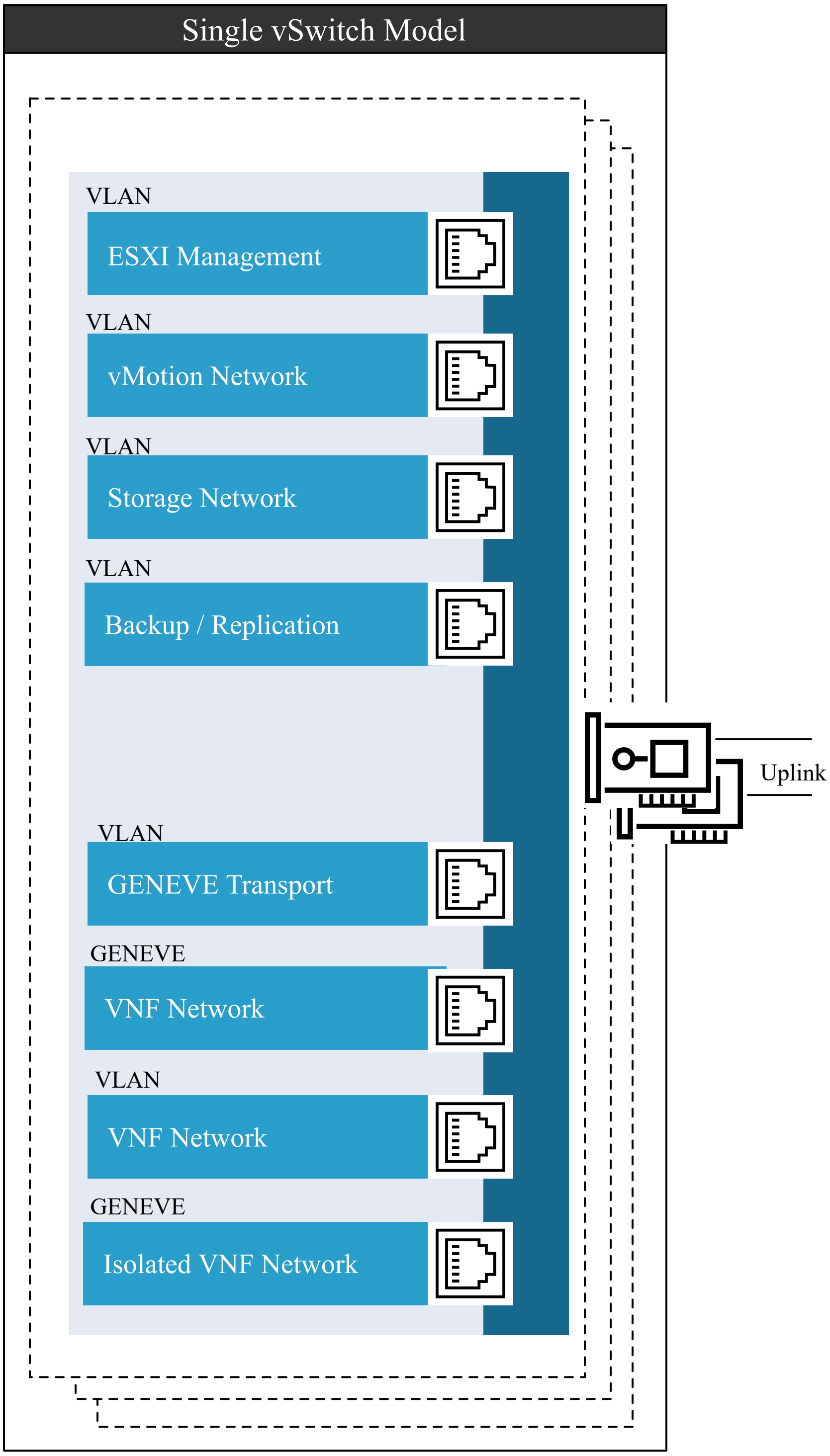
Two vSwitches are used to separate management traffic from workload traffic even if the hosts are not prepared with NSX. This would remove the GENEVE and Mgmt Overlay segments but the Management VLAN networks exist regardless of the state of the NSX in the management domain.

Note

- If both vSAN and other storage platforms are both deployed, separate VLANs must be created for different storage switches.
- The management vSwitch must leverage Network I/O control to provide guarantees for the storage network. If vSAN ESA is used, each network uplink must be at least 25 GB.

The following vSwitch design shows the single vSwitch model for the general VNF/CNF workload clusters

Figure 4-3. Single vSwitch Model



In the single vSwitch model, a common vSwitch is used for host management, with VLAN for ESXi, vMotion, Storage (vSAN or IP storage), backup/replication, and workload traffic.

When using a single vSwitch, configure Network I/O control to ensure guarantees to both management and workload traffic.

The vSwitch allows the separation of different traffic types, providing access security and reducing the contention and latency.

High latency on any network can negatively affect the performance. Some components are more sensitive to high latency than others. For example, high latency IP storage and the vSphere Fault Tolerance logging network can negatively affect the performance of multiple VMs.

Depending on the CSP requirements, constraints, and hardware, you can use either of these vSwitch models for the deployment and configuration of the network virtualization design. Ensure that you understand the impact.

vSwitch and NUMA Considerations

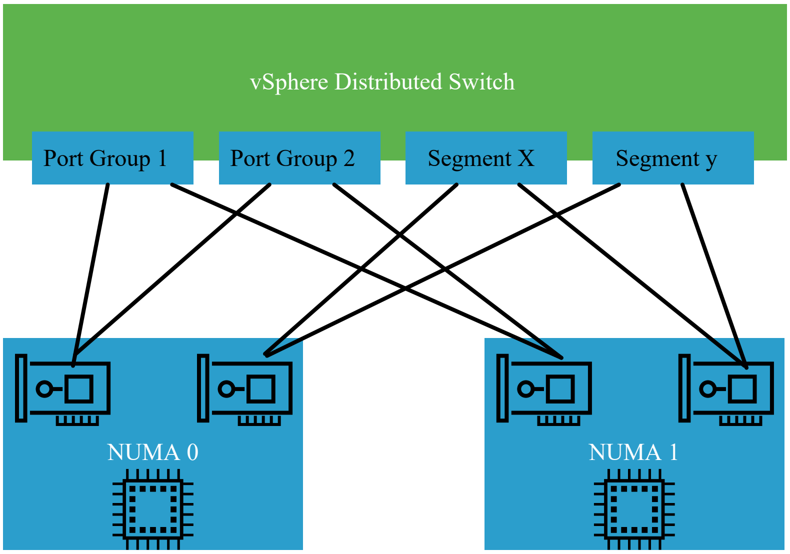
For User-Plane intensive workloads, such as the UPF, the alignment of the networking with the CPUs used for VM execution is important to maximize performance.

To perform NUMA alignment on the vSwitching layer, at least two physical NICs are required. One NIC must be installed in a PCI slot that aligns to socket 0 (NUMA Node 0), and the other must be installed in a PCI slot that aligns to socket 1 (NUMA Node 1).

Important When using NSX Enhanced Data Path (EDP) with multiple NICs that are NUMA aligned, NSX ensures (through the converged VDS) that the traffic is NUMA aligned. The network traffic is aligned with the NUMA node from where the VM vCPUs and memory are allocated.

For fully NUMA redundant designs, 4 NICs (2 per NUMA) are required.

Figure 4-4. NUMA-aligned vSwitch



NIC Teaming and Failure Detection

NIC teaming policies increase the network bandwidth available to network paths and provide redundancy by avoiding single points of failure for network segments.

NIC teaming is achieved by assigning multiple physical NICs to a virtual switch. Any NIC can be used. However, the NIC team built using ports from multiple NICs and motherboard interfaces provides optimal protection, decreases the chances of failure, and aligns with the NUMA considerations.

In addition to NIC teaming, the Telco Cloud administrators can use the following types of failure detection methods:

- **Link status only**:
  - Relies only on the link status that the network adapter provides.
  - Detects failures such as removed cables and physical switch power failures.

Caution Link status will not detect switch ports put into a blocking state by the Top-Of-Rack Switch, nor will it detect VLAN mismatches between the ESXi host and the switch.

- **Beacon probing**: Sends out and listens for Ethernet broadcast frames or beacon probes that physical NICs send to detect link failure in all physical NICs in a team. ESXi hosts send beacon packets every second. Beacon probing is most useful to detect failures in the physical switch closest to the ESXi host, where the failure does not cause a link-down event for the host. However, to properly function, beacon probing requires three uplinks. Link Status is the recommended failure detection mechanism.

Network I/O Control

When Network I/O Control (NIOC) is enabled, the distributed switch allocates bandwidth for different traffic types. vSphere 6.0 and later supports NIOC version 3.

When network contention occurs, Network I/O Control enforces the share value specified for different traffic types. Network I/O Control applies the share values set to each traffic type. Hence, less important traffic, as defined by the share percentage or network pools, is throttled while granting more network resources to more important traffic types.

Network I/O Control supports bandwidth reservation for system traffic based on the capacity of physical adapters on an ESXi host. It also enables fine-grained resource control at the VM network adapter. Resource control is similar to the CPU and memory reservation model in vSphere DRS.

NIOC is enabled on the management switches to assign bandwidth reservations for vMotion or vSAN. However, it can add latency that might result in a reduction in network performance. Hence, NIOC is not recommended on workload vSwitches.

TCP/IP Stacks for vMotion

Use the vMotion TCP/IP stack to isolate the traffic for vSphere vMotion and to assign a dedicated default gateway for the vSphere vMotion traffic.

By using a separate TCP/IP stack, you can manage vSphere vMotion and cold migration traffic according to the network topology, and as required by your organization.

- Route the traffic for the migration of VMs (powered on or off) by using a default gateway. The default gateway is different from the gateway assigned to the default stack on the ESXi host.
- Assign a separate set of buffers and sockets.
- Avoid the routing table conflicts that might appear when many features are using a common TCP/IP stack.
- Isolate the traffic to improve security.

Recommended Network Virtualization Design

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Use two physical NICs for vSwitch uplinks on all vSwitches.	Provides redundancy to all port groups.	None	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge
Provide NUMA-aligned uplinks to the workload vSwitch.	Maximizes workload throughput in dual-socket servers	Requires at least one or two uplink ports per NUMA for maximum redundancy	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge
Use the Route based on the physical NIC load teaming algorithm for all port groups.	Reduces the complexity of the network design Increases resiliency and performance.	None	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge
Separate uplinks across physical cards	Provides high availability in the event of a NIC failure	Requires at least two NIC cards per NUMA if redundancy is required per NUMA	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge
Use vSphere Distributed Switches.	Simplifies the management of the virtual network.	Migration from a standard switch to a distributed switch requires a minimum of two physical NICs to maintain redundancy.	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge D-RAN or C-RAN deployments through cell site groups
Use Converged VDS when using NSX	Allows NSX functionality to be leveraged as part of the workload vSwitches	Converged VDS for workload	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Workload vSwitches only
Use ephemeral port binding for the management port group.	Provides the recovery option for the vCenter Server instance that manages the distributed switch.	Port-level permissions and controls are lost across power cycles, and no historical context is saved.	Management Domain Management vSwitch Only

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Use static port binding for all non-management port groups.	Ensures that a VM connects to the same port on the vSphere Distributed Switch. This allows for historical data and port-level monitoring.	None	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge
Enable Network I/O Control on non-workload distributed switches.	Increases the resiliency and performance of the network	If configured incorrectly, Network I/O Control might impact the network performance for critical traffic types.	Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Management vSwitch Only
Use the vMotion TCP/IP stack for vSphere vMotion traffic.	By using the vMotion TCP/IP stack, vSphere vMotion traffic can be assigned a default gateway on its own subnet and can go over Layer 3 networks.	None	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Management vSwitch only
Configure all vSwitches with 9000 MTU.	Increases throughput for vMotion / vSAN and workloads	Requires ToR switches also to support Jumbo Frames	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Both vSwitches must be configured with Jumbo Frames.
Use Link status as the link failure detection mechanism.	Beacon probing requires 3 NICs in a NIC team.	Certain types of failures further down the path in the underlay network cannot be detected. Monitoring for the physical networking must be able to capture and respond to those failures.	Compute clusters, VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Both vSwitches must be configured with Link Status.

RAN vSwitch Design

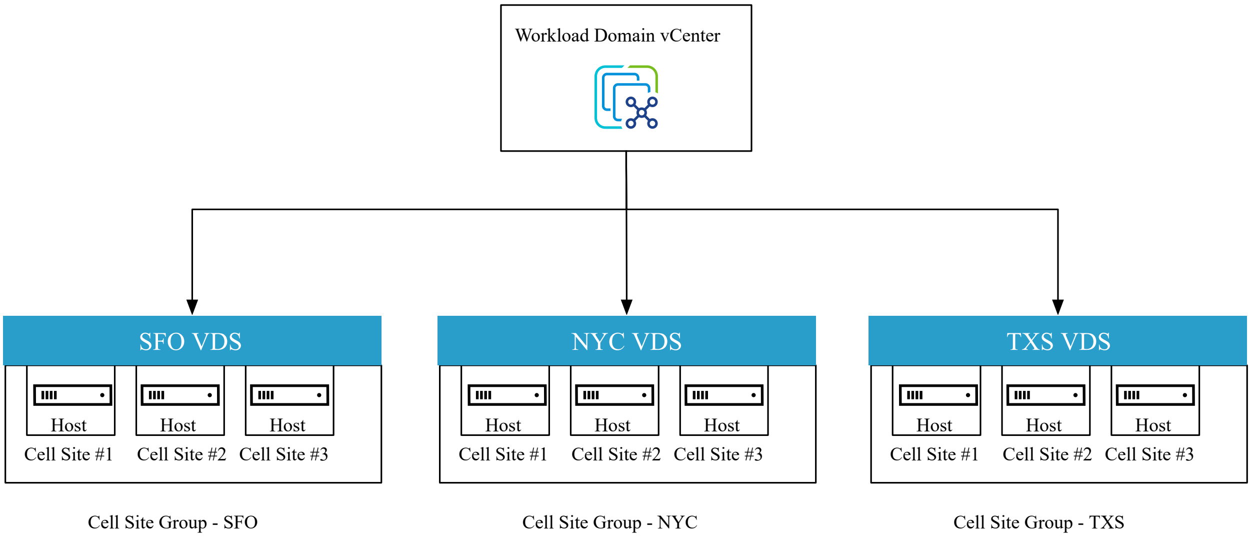
In a RAN environment, specifically the D-RAN environment at cell sites, the vSwitch design is different from the management and general cluster design.

In cell site ESXi hosts, create a single virtual switch per cell site group. The virtual switch can manage each type of network traffic and configure a port group to simplify the configuration and monitoring.

When leveraging Telco Cloud Automation to onboard RAN cell sites, the concept of a Cell Site Group is created. This Cell Site Group defines the vSwitch configuration. When the host is added to the cell site group, the vSwitch configuration is applied across all hosts in the cell site group.

The Distributed vSwitch concept eases the management burden by treating the network as an aggregated resource. Individual host-level virtual switches are abstracted into one large VDS spanning multiple hosts. In this design, the data plane remains local to each VDS but the management plane is centralized.

Figure 4-5. Dedicated vSwitches per Cell Site Group



The management traffic on the RAN Cell Site must be less than 150ms from the ESXi host back to the vCenter.

Important Follow these considerations for the RAN cell sites:

- The maximum number of unique vSwitches that are supported on a single vCenter is 128.
- Each VDS can manage up to 2000 hosts. However, the design for each Cell Sites group consumes a unique vSwitch.

RAN SR-IOV Considerations

Due to the challenges in resource optimization, vMotion, and so on, SR-IOV is not recommended for the Core or Edge of the Telco Cloud. However, in the RAN (especially D-RAN) where a single server is deployed, SR-IOV can improve overall RAN performance.

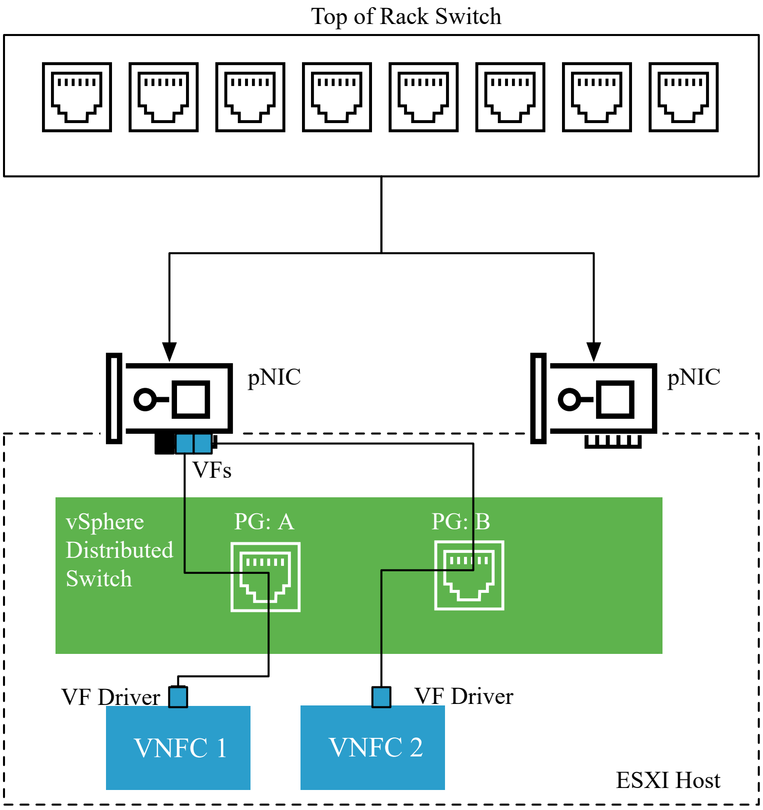
SR-IOV is a specification that allows a single Peripheral Component Interconnect Express (PCIe) physical device under a single root port to appear as multiple separate physical devices to the hypervisor or the guest operating system.

SR-IOV uses Physical Functions (PFs) and Virtual Functions (VFs) to manage global functions for the SR-IOV devices. PFs are full PCIe functions that configure and manage the SR-IOV functionality. VFs are lightweight PCIe functions that support data flow but have a restricted set of configuration resources. The number of VFs provided to the hypervisor or the guest operating system depends on the device. SR-IOV enabled PCIe devices require appropriate BIOS, hardware, and SR-IOV support in the guest operating system driver or hypervisor instance.

Note In vSphere 8.0, the number of VFs that can be attached to a VM is increased to accommodate RAN deployment requirements. For more information about configuration maximums, see VMware Configuration Maximums.

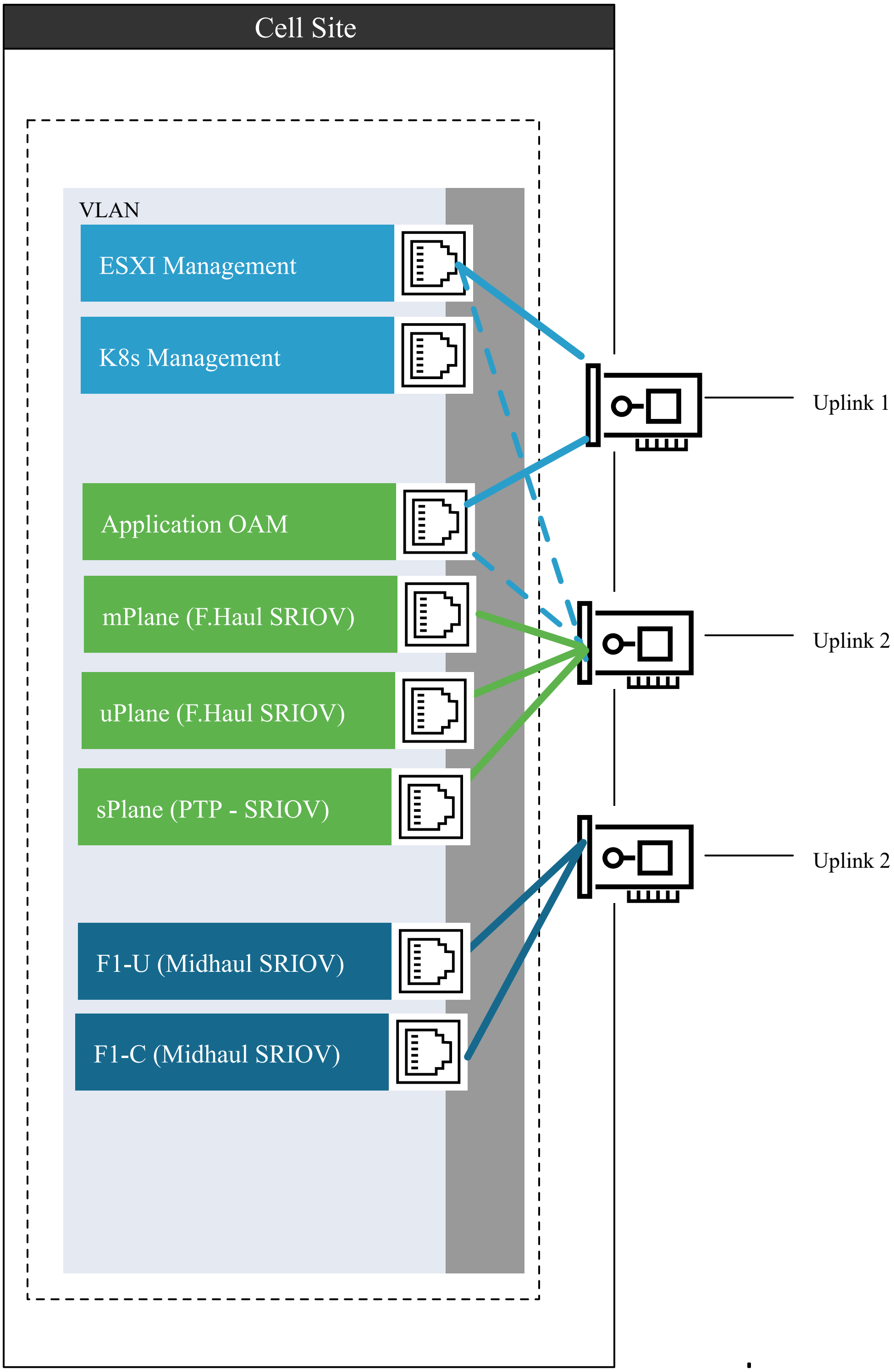
In vSphere, a VM can use an SR-IOV virtual function for networking. The VM and the physical adapter exchange data directly without using the VMkernel stack as an intermediary. Bypassing the VMkernel for networking reduces the latency and improves the CPU efficiency for high data transfer performance. As the VM does not have access to the Physical Function, a message box or similar solution is created to allow the VFs that are exposed to the guest OS to send communication messages to the Physical Function. This allows for hardware configuration or alerting messages from within the Guest OS back to ESXi.

Figure 4-6. Logical View of SR-IOV Configuration



SR-IOV makes the vSwitch design in a RAN cell site different from the traditional vSwitch model. The following diagram illustrates the VDS design in a RAN cell site:

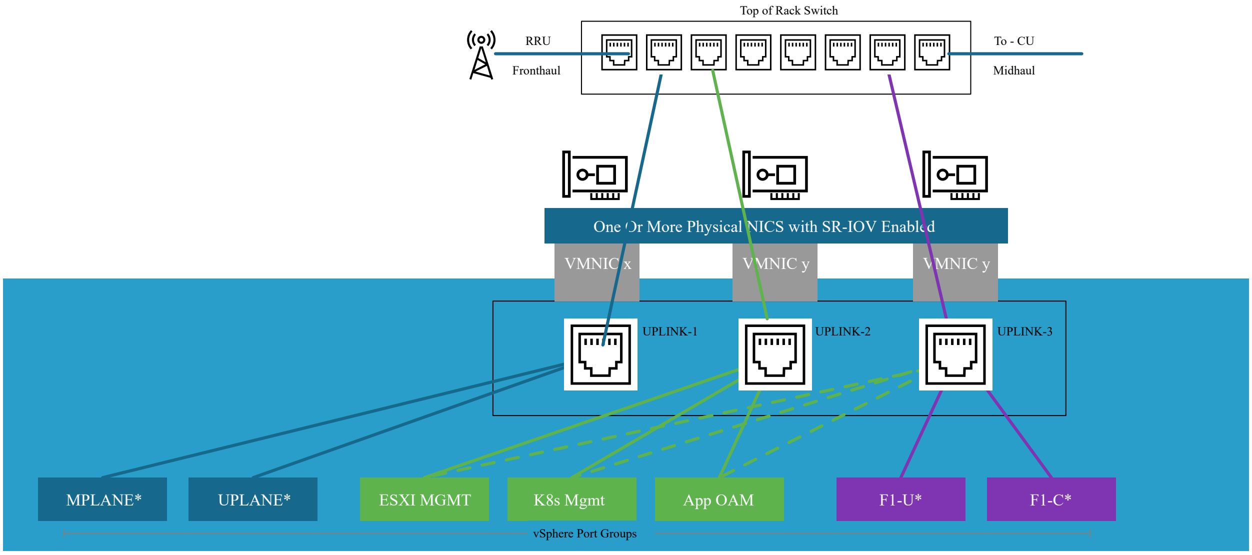
Figure 4-7. Cell Site VDS



In the Cell Site, the layout and configuration of the vSwitch is different. Because of the SR-IOV interfaces for Fronthaul (from RU to DU) and Midhaul (from DU to CU), fixed uplinks are used for different connectivity requirements.

The following diagram illustrates an alternate representation of the RAN Cell Sites switch:

Figure 4-8. Alternate Cell Site vSwitch View



PTP timing is an important consideration in the RAN. In the model shown above, the PTP Grandmaster is connected to the Top of Rack switch. This design is typical for LLS-C3 type configurations.

Depending on the hardware installed and the workload, the PTP Timing interface can be configured at the cell site in one of the two ways:

- Passthrough of the PF to the worker node: This must be the first physical interface on the NIC (the first port on a specific NIC). When using this configuration, an unused VMNIC port must be used for PT.
- PTP over VF: In this method, the S-Plane is passed to the worker node over an additional SR-IOV interface. The S-Plane interface is carried over VMNIC.

For more information about PTP configuration, see A1: PTP Overview in the Telco Cloud Automation User Guide.

In an LLS-C1 configuration, the DU nodes connect directly to additional ports on the ESXi server. The NICs need to provide an onboard GNSS solution to take clocking directly from a GPS antenna rather than from the Top of Rack Switch.

This model requires additional NICs on the physical server.

Note When using SR-IOV along with NUMA alignment, Telco Cloud Automation ensures that the SR-IOV VF is taken from a physical NIC that is aligned to the NUMA node where the VM will be placed.

The option to relax the NUMA alignment can be selected. However, this is used when the SR-IOV VF is used for PTP and not for user plane interfaces.

RAN Network Virtualization Recommendations

Design Recommendation	Design Justification	Design Implication
Use two physical NICs in Cell Site ESXi host for workloads.	Provides redundancy to all port groups	None
Use a minimum of one SR-IOV VF or physical NIC in Cell Site ESXi hosts for PTP time synchronization.	Provides time synchronization service	Needs to be determined if NIC card can provide PTP over SR-IOV VF or if pass-through is necessary
Create Cell Sites using the Cell Site Grouping mechanism in Telco Cloud Automation	Provides a central configuration point for adding cell site hosts to the environment	
When using multiple cell site groups to segment a market or large RAN deployment, be aware of the 128 vSwitch limit.	Allows for proper dimensioning of cell site groups.	Maximum 128 vSwitches supported in a single vCenter. These vSwitches needs to be shared between management vSwitches, workloads, and RAN
Use at least 3 ports for the vSwitch uplink.	Allows separation of management, fronthaul, and midhaul traffic	Requires additional ports
Ensure that fronthaul and midhaul ports are assigned to different uplinks in the Port Group configuration.	Separates the fronthaul and midhaul traffic, allowing for maximal throughput	

NSX Design

NSX is an implementation of software-defined networking. It provides network services such as switching, routing, load balancing, firewall, and Virtual Private Networking (VPN).

NSX focuses on providing networking, security, automation, and operational simplicity for the underlying physical network. NSX is a non-disruptive solution and can be deployed on any IP network such as traditional networking models and next-generation fabric architectures, regardless of the vendor. This is accomplished by decoupling the virtual networks from their physical counterparts.

NSX Manager

The NSX Manager is the centralized network management component of VMware NSX . It implements the management and control planes for the NSX infrastructure.

NSX Manager provides the following functions:

- The Graphical User Interface (GUI) and the RESTful API for creating, configuring, and monitoring NSX components, such as segments and gateways.
- An aggregated system view
- A method for monitoring and troubleshooting workloads attached to virtual networks
- Configuration and orchestration of the following services:
  - Logical networking components, such as logical switching and routing

  - Networking and edge services
  - Security services and distributed firewall
- A RESTful API endpoint to automate consumption. Because of this architecture, you can automate all configuration and monitoring operations using any cloud management platform, security vendor platform, or automation framework.

Some of the components of the NSX Manager are as follows:

- **NSX Management Plane Agent** (MPA): Available on each ESXi host. The MPA persists the desired state of the system and communicates Non-Flow-Controlling (NFC) messages such as configuration, statistics, status, and real-time data between transport nodes and the management plane.
- **NSX Controller**: Controls the virtual networks and overlay transport tunnels. The controllers are responsible for the programmatic deployment of virtual networks across the entire NSX architecture.
- **Central Control Plane** (CCP): Logically separated from all data plane traffic. A failure in the control plane does not affect existing data plane operations. The controller provides configuration to other NSX Controller components such as the segments, gateways, and edge VM configuration.
- **Local Control Plane** (LCP): The LCP runs on transport nodes. It is adjacent to the dataplane it controls and is connected to the CCP. The LCP is responsible for programming the forwarding entries and firewall rules of the data plane.

Virtual Distributed Switch

vSphere Distributed switch (vSphere 7 and later versions) supports NSX Distributed Port Groups. NSX and vSphere integration consolidates the use of NSX on VDS, and this model is known as a Converged Virtual Distributed Switch (C-VDS).

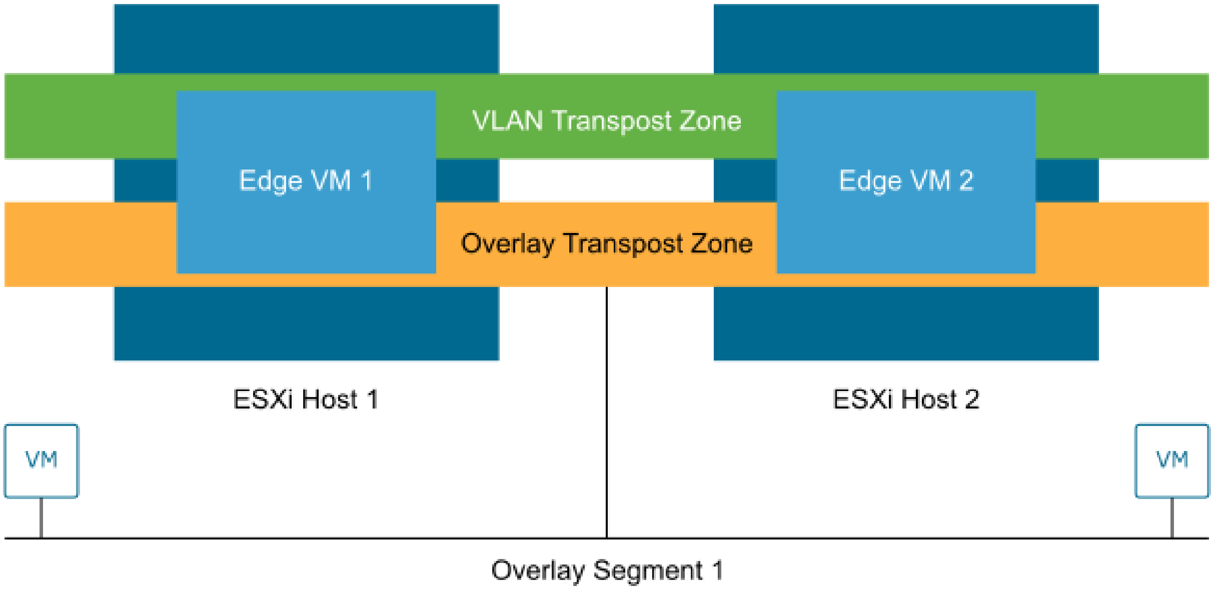
NSX implements each logical broadcast domain by tunneling VM-to-VM traffic and VM-to-gateway traffic using the Geneve tunnel encapsulation mechanism. The network controller has a global view of the data center and ensures that the virtual switch flow tables in the ESXi host are updated as the VMs are created, moved, or removed.

NSX implements virtual switching in Standard and Enhanced Data Path (EDP) modes. EDP provides better network performance for telco workloads. The EDP mode supports both overlay and VLAN based network segments.

Transport Zones

Transport zones determine which hosts can consume a particular network. A transport zone identifies the type of traffic, such as VLAN, overlay, and the teaming Policy. You can configure one or more transport zones. A transport zone does not represent a security boundary.

Figure 4-9. Transport Zones



In NSX, when an ESXi host is converted into a transport node, it is attached to one or more transport zones. The switch type and mode are configured during the host provisioning or within the transport node profiles.

Note Only a single overlay transport zone is supported for ESXi hosts. Multiple VLAN transport zones can be configured.

Logical Switching

NSX Segments create logically abstracted segments to which the workloads can be connected. A single segment is mapped to a unique Geneve segment ID (or VLAN ID) that is distributed across the ESXi hosts (and NSX edge nodes) within a given transport zone. NSX Segments support switching in the ESXi host without the constraints of VLAN sprawl or spanning tree issues.

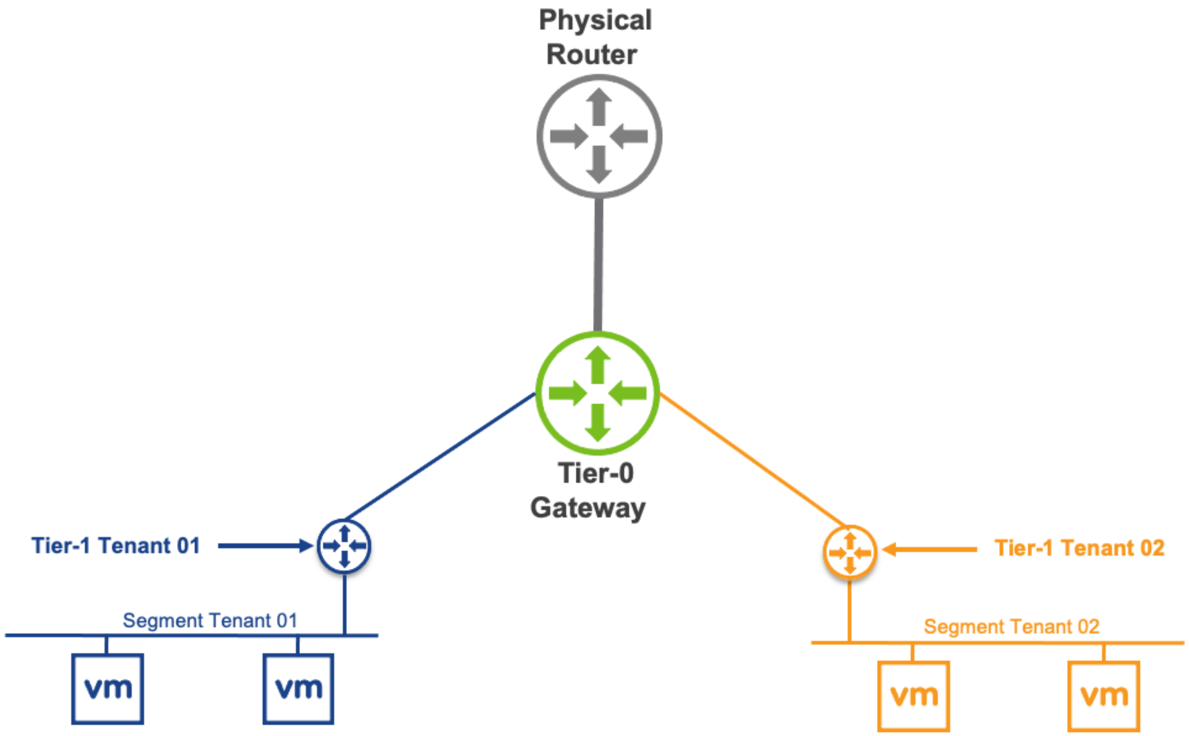
Gateways

NSX Gateways provide the North-South connectivity for the workloads to access external networks and the East-West connectivity between different logical networks.

A gateway is a configured partition of a traditional network hardware router. It replicates the functionality of the hardware, creating multiple routing domains in a single router. Gateways perform a subset of the tasks that are handled by the physical router. Each gateway can contain multiple routing instances and routing tables. Using gateways can be an effective way to maximize the use of routers.

- **Distributed Router**: A Distributed Router (DR) spans across all ESXi nodes to which VMs are connected to the NSX gateway. The DR construct also expands to the edge nodes. Functionally, the DR is responsible for one-hop distributed routing between segments and other gateways connected to the NSX gateway.
- **Service Router**: A Service Router (SR) implements stateful services such as Border Gateway Protocol (BGP), stateful Network Address Translation (NAT). These services cannot be implemented in a distributed way. A gateway always has a DR. A gateway has SRs when it is a Tier-0 Gateway or a Tier-1 Gateway. It is configured with services such as load balancing, NAT, or Dynamic Host Configuration Protocol (DHCP).

Figure 4-10. Traditional NSX Routing

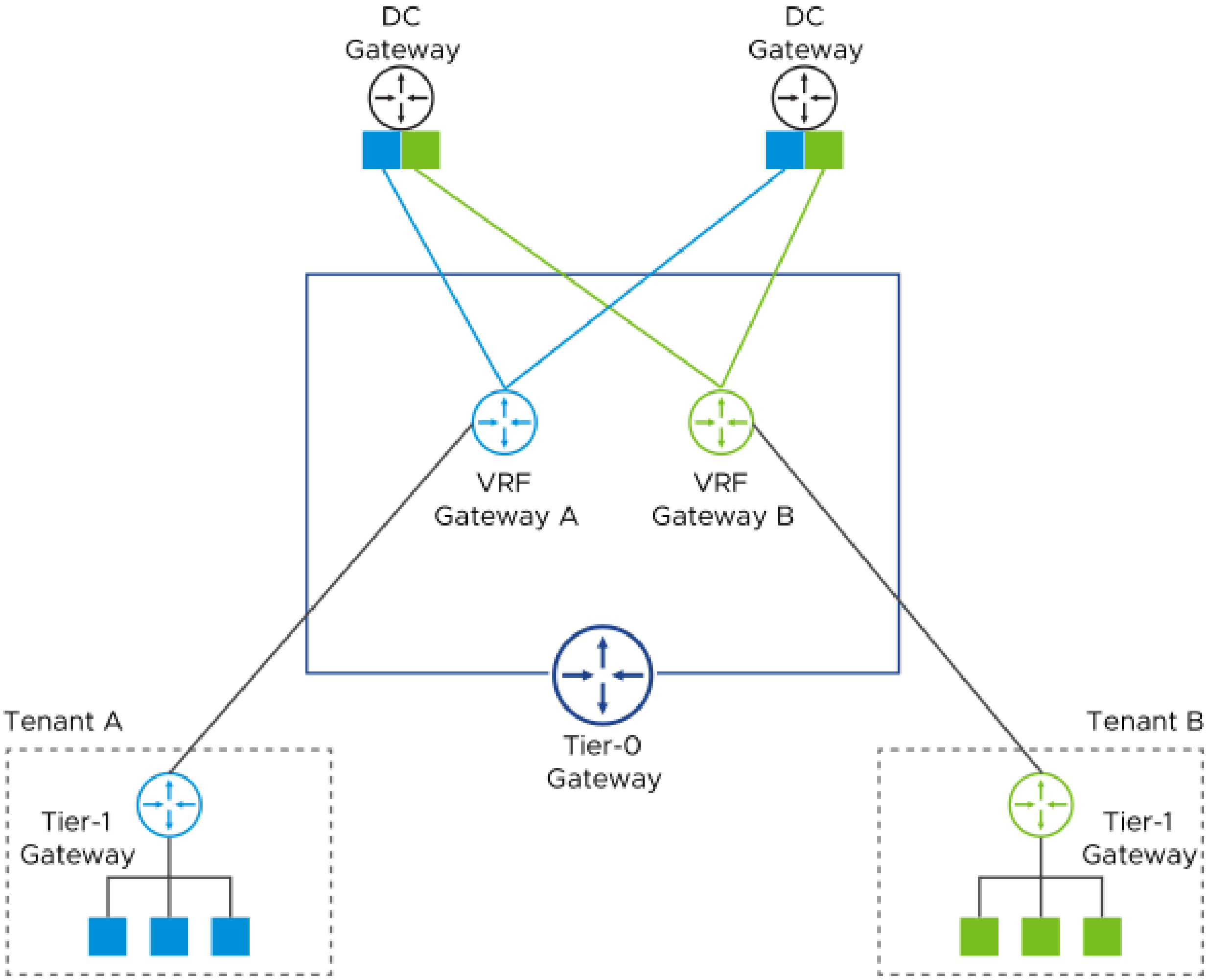


Virtual Routing and Forwarding

A Virtual Routing and Forwarding (VRF) gateway enables multiple instances of a routing table to exist simultaneously within the same gateway. A VRF is a layer 3 equivalent of a VLAN. A Tier-1 router or network segment must be linked to a tier-0 VRF gateway. From the tier-0 VRF gateway the configuration inherits the parent T0 configuration.

In a multi-tenant solution, VRFs allow a single Tier-0 gateway to be deployed and managed while keeping the routing tables between tenants isolated. Each VRF can peer to a different eBGP neighbor and autonomous system (AS).

Figure 4-11. NSX VRF Routing



Enhanced Data Path

NSX Enhanced Data Path (EDP) is a networking stack mode, which provides better network performance. It is primarily designed for data plane intensive workloads.

- When EDP mode is enabled on a transport node, a combination of technologies is used. The technologies include mbuf packet representation, DPDK fastpath, polling, flow-cache, dedicated CPUs, and vmxnet3 optimizations. These technologies together significantly accelerate the network traffic for the network function workloads.
- The NSX EDP mode virtual switches use the same emulated virtual NIC adapter, VMXNET3, as the Standard or Distributed vSphere Virtual Switch. Unlike other solutions such as SR-IOV, you can consume EDP with no loss of functionality such as vMotion.
- The NSX implementation of EDP remains in the ESXi Hypervisor space (vmkernel). Hence the user-space security concerns associated with open-source DPDK networking stacks are not applicable to EDP.

When using EDP, CPU cores must be pre-allocated to the infrastructure dimensioning plan. The CPU cores are assigned per EDP switch instance, per host, and per NUMA. The quantity depends on the throughput required by the workloads. SR-IOV, in comparison, does not require any CPU cores in the infrastructure compute budget. However, the benefits of EDP outweigh the CPU cores saved using SR-IOV.

The following table provides a high-level comparison between the two design choices:

- **NSX EDP only**: A single virtual switch is used to manage data plane and non-data plane traffic.
- **NSX & SR-IOV**: NSX Standard mode switch is used for non-data plane traffic. SR-IOV is used for data plane traffic.

Note Uplinks that are used for EDP switching cannot also be leveraged for SR-IOV VFs.

Table 4-4. NSX EDP and SR-IOV Comparison

	NSX EDP only	**NSX + SRIOV**
CPU	2-4 physical cores per CPU Additional Core could be considered	None **Note**: The assumption is that the NSX threads use the cores reserved for ESXi.
NIC	2 ports per socket	2 ports for NSX 2 ports per socket for SR-IOV
Throughput	The lowest throughput based on the physical network, EDP virtual switch, or the workload.	The lower of the physical network or the workload.
Programmable L2/L3 networks (SDN)	Fully programmable	Not supported
Programmable Edge	Fully programmable, high-performance Edges	Not supported

Table 4-4. NSX EDP and SR-IOV Comparison (continued)

	NSX EDP only	NSX + SRIOV
End-to-end visibility and analytics	Available	Not supported
Provisioning and resiliency	NIC teaming, vMotion, HA, DRS	Not supported
Driver Standardization	Available (VMXNET3)	Not supported **Note**: Application vendor-specific drivers are required.
NIC support	VMware Hardware Compatibility List	VMware + App Vendor HCL for SR-IOV
Upgrade compatibility	Available across NSX releases	Vendor-specific for SR-IOV
Support model	VMware Carrier-Grade Support	Complex multivendor resolution

Note EDP is supported in conjunction with the Virtual Hyperthreading (vHT) functionality provided by vSphere 8. When configuring EDP interfaces as part of Dynamic Infrastructure Provisioning (DIP) in VMware Telco Cloud Automation. These interfaces can be consumed while configuring vHT.

The provisioning of EDP requires the assignment of lCores. Theses lCores must be considered for the overall resource budget.

Ethernet VPN (EVPN)

Ethernet VPN (EVPN) is a standards-based BGP control plane that enables the extension of Layer 2 and Layer 3 connectivity.

In the Route Server mode, EVPN allows workloads such as an Evolved Packet Core (EPC) that supports BGP peering and high throughput and low latency to bypass the NSX edge node and route traffic directly to the physical network.

The NSX edge resides in the control path and not the data path. The NSX edge peers to both the workload and the physical network. The data path bypasses the NSX edge node and routes directly to the physical network using VXLAN encapsulation, enabling high throughput and low latency required by this class of applications.

Figure 4-12. EVPN Route Server Mode

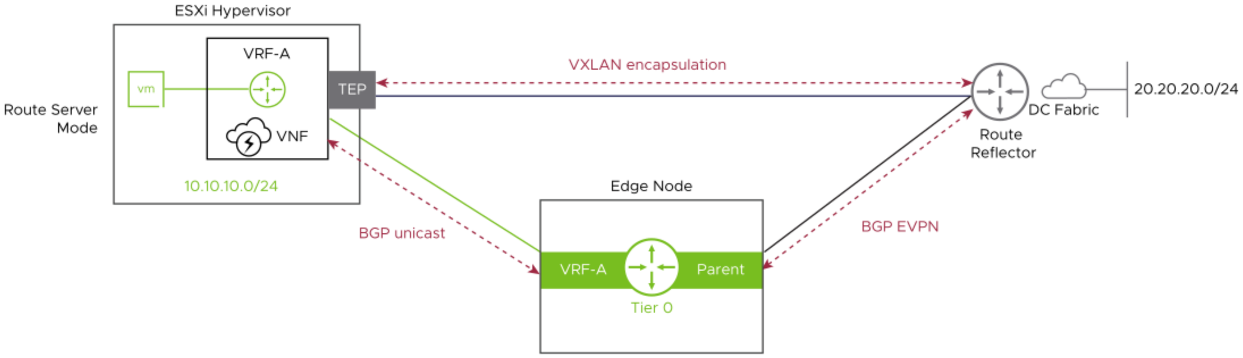
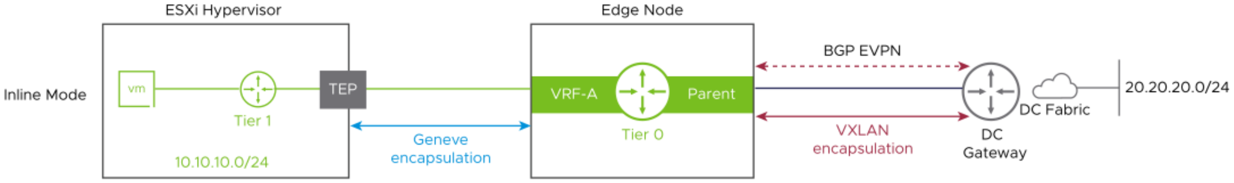


Figure 4-13. EVPN Inline Mode



Note EVPN route server mode is implemented using a specific set of RFCs. These RFCs must be supported on the network underlay devices for the service to work end-to-end.

Table 4-5. Recommended NSX Design

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Deploy a three-node NSX Manager cluster using the large-sized appliance to configure and manage all NSX-based compute clusters.	The large-sized appliance supports more than 64 ESXi hosts. The small-sized appliance is for proof of concept and the medium size supports up to 64 ESXi hosts only.	The large-sized deployment requires more resources in the vSphere management cluster.	Management domain, Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites
Apply vSphere Distributed Resource Scheduler (DRS) anti-affinity rules to the NSX Manager/Controller cluster nodes.	Using DRS prevents Manager or Controller nodes from running on the same ESXi host, and thereby risking their high availability.	Additional configuration is required to set up anti-affinity rules and the rules must be maintained in the event of node restores.	
Create a VLAN and Overlay Transport zone.	Ensures that all segments are available to all ESXi hosts and edge VMs are configured as Transport Nodes.	None	Management domain, Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites

Table 4-5. Recommended NSX Design (continued)

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Configure ESXi hosts to use the vSphere Distributed Switch with EDP mode in each NSX compute cluster.	Provides a high-performance network stack for NFV workloads.	EDP mode requires more CPU resources, and compatible NICs compared to standard or ENS interrupt mode.	
Use large-sized NSX Edge VMs.	The large-sized appliance provides the required performance characteristics if a failure occurs.	Large-sized Edges consume more CPU and memory resources.	
Deploy at least two large sized NSX Edge VMs in the vSphere Edge Cluster.	Creates the NSX Edge cluster to meet availability requirements.	None	
Create an uplink profile with the load balance source teaming policy with two active uplinks for ESXi hosts.	For increased resiliency and performance, supports the concurrent use of two physical NICs on the ESXi hosts by creating two TEPs.	None	
Create a second uplink profile with the load balance source teaming policy with two active uplinks for Edge VMs.	For increased resiliency and performance, supports the concurrent use of two virtual NICs on the Edge VMs by creating two TEPs.	None	
Create a Transport Node Policy with the VLAN and Overlay Transport Zones, VDS settings, and Physical NICs per vSphere Cluster.	Allows the profile to be assigned directly to the vSphere cluster and ensures consistent configuration across all ESXi hosts in the cluster.	You must create all required Transport Zones before creating the Transport Node Policy.	
Create two VLANs to enable ECMP between the Tier-0 Gateway and the Layer 3 device (ToR or upstream device). The ToR switches or the upstream Layer 3 devices have an SVI on one of the two VLANs. Each edge VM has an interface on each VLAN.	Supports multiple equal-cost routes on the Tier-0 Gateway and provides more resiliency and better bandwidth use in the network.	Extra VLANs are required.	

Table 4-5. Recommended NSX Design (continued)

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Deploy an Active-Active Tier-0 Gateway.	Supports ECMP North-South routing on all edge VMs in the NSX Edge cluster.	Active-Active Tier-0 Gateways cannot provide services such as NAT. If you deploy a specific solution that requires stateful services on the Tier-0 Gateway, you must deploy a Tier-0 Gateway in Active-Standby mode.	
Deploy a Tier-1 Gateway to the NSX Edge cluster and connect it to the Tier-0 Gateway.	Creates a two-tier routing architecture that supports load balancers and NAT. Because Tier-1 is always Active/Standby, the creation of services such as load balancers or NAT is possible.	None	
Deploy Tier-1 Gateways with Non-Preemptive setting.	Ensures that when the failed Edge Transport Node comes back online it does not move services back to itself resulting in a small service outage.	None	
Replace the certificate of the NSX Manager instances with a certificate that is signed by a third-party Public Key Infrastructure.	Ensures that the communication between NSX administrators and the NSX Manager instance is encrypted by using a trusted certificate.	Replacing and managing certificates is an operational overhead.	
Replace the NSX Manager cluster certificate with a certificate that is signed by a third-party Public Key Infrastructure.	Ensures that the communication between the virtual IP address of the NSX Manager cluster and NSX administrators is encrypted using a trusted certificate.	Replacing and managing certificates is an operational overhead.	

Load Balancer Design - NSX Advanced Load Balancer

VMware NSX Advanced Load Balancer (formerly Avi Networks) is a software defined platform that provides centrally managed dynamic pool of load balancing resources on commodity x86 servers, VMs, or containers to deliver granular services close to individual applications.

NSX Advanced Load Balancer has three core components:

- Admin Console
- Controller

- Service Engines

NSX Advanced Load Balancer - Admin Console

The NSX Advanced Load Balancer Admin Console is a modern web-based user interface that provides role-based access to control, manage, and monitor applications. Its capabilities are also available through CLI and REST API that can be integrated with other systems such as Tanzu Kubernetes Grid.

NSX Advanced Load Balancer Controller

The NSX Advanced Load Balancer Controller is the single point of management and control that serves as the "brain" of the platform. It supports high availability and is deployed as a three-node cluster. The leader node performs load balancing configuration management for the cluster. The follower nodes collaborate with the leader node to perform data collection from Service Engines and process analytic data.

The NSX Advanced Load Balancer controllers continually exchange information securely with the service engines and with one another. The health of servers, client connection statistics, and client-request logs collected by the service engines are regularly offloaded to the controllers. These controllers share the work of processing the logs and aggregating analytics. The controllers also send commands such as configuration changes to the service engines. Controllers and service engines communicate using their management IP addresses.

NSX Advanced Load Balancer Service Engine

NSX Advanced Load Balancer Service Engines (Service Engines) are VM-based applications that handle all data plane operations by receiving and executing instructions from the Controller. The Service Engines perform load balancing for all client- and server-facing network interactions. It also collects real-time application telemetry from application traffic flows.

The hardware requirements of a Service Engine can be customized in a specific Service Engine group.

Virtual Service

Within NSX Advancer Load Balancer, each Load Balancer service is associated with one or multiple Virtual Services (VSs)- that serve this load balancer. The Virtual Services are placed on Service Engines and its affinity depends on the Virtual Service placement setting in a specific group.

Virtual service placement mechanisms:

- **Compact**: In this mechanism, NSX Advanced Load Balancer prefers to spin up and use the minimum number of Service Engines. The Virtual Services are placed on Service Engines that are already running.

- **Distributed**: In this mechanism, NSX Advanced Load Balancer maximizes Virtual Service performance by avoiding placements on existing Service Engines. Instead, it places Virtual Services on newly spun-up Service Engines, up to the maximum number of Service Engines per group.

If a Virtual Service is scaled out across multiple Service Engines, the Virtual Service placement setting is used to determine which Service Engines to use.

Recommended NSX Advanced Load Balancer Design

Design Recommendation	Design Justification	Design Implication	Domain Applicability
Deploy NSX ALB controller cluster with three controllers.	Provides high availability on the control plane	Requires additional networking and resources capacity	Management domain, Compute clusters VNF, CNF, C-RAN, Near/Far Edge, and NSX Edge Not applicable to RAN sites
Deploy an NSX ALB controller cluster with three controllers in each domain in a multi-site deployment.	Provides control plane services in local site even during the primary site failure	Resources are required to deploy NSX ALB controller cluster in each site.	
Use the Write access mode that allows NSX ALB to dynamically create and scale Service Engines.	Enables automation of Service Engines deployments	User with sufficient permissions must be provided in vCenter SSO. Sufficient resources must be available for dynamic scaling.	
Set the Virtual Service placements over Service Engines to distributed.	Provides more efficient use of Service Engines Optimizes throughput performance of the load balancer	Requires more resources	
Minimum of two Virtual Services for each specific load balancer service.	Increases the availability of the Virtual Service	Requires more resources	
BGP with ECMP and RHI are used to advertise VIPs.	BGP and RHI are required for the dynamic advertisement in the event of load balancer scaling.	Specific configuration is needed both on NSX ALB and on peer routers to set up the BGP.	
N+M mode is used for Service Engine Groups.	Provides better compromise of availability and scaling for the Virtual Services.	Minimum number of Service Engines per Service Engine group must be three.	

Design Recommendation	Design Justification	Design Implication	Domain Applicability
To tolerate one Service Engine failure, M is set to 1 (M=1). This means that the equivalent resources of one Service Engine in a Service Engine Group are reserved for an HA event.	Provides better availability and performance during a Service Engine failure event	Additional resources are required to host the HA Service Engine.	
Service Engine data-path failure detection is not enabled. BGP+BFD is used instead.	Faster failure detection and load distribution using ECMP.	BGP needs to be used in the environment, and BFD timers need to be aligned.	

IPv4, IPv6, and Dual-Stack Considerations

Every component of the Telco Cloud stack has its own IPv4-IPv6 support matrix.

IPv6 was defined by the Internet Engineering Task Force (IETF) to overcome the IPv4 limitations like the number of available address space. IPv6 also provides improved address autoconfiguration between nodes.

In the Telco Cloud, VNFs and CNFs can communicate in multiple ways using IPv4 and IPv6 protocols, leveraging the capabilities of the platform components:

- VNFs deployed on top of VMware Cloud Director or VMware Integrated OpenStack deployments: separate vNICs for each address family or a dual stack configuration with DHCP or static assignment
- CNFs deployed through VMware Telco Cloud Automation on to the CaaS infrastructure.
- NSX: single or dual stack for overlay or VLAN segment with Standard or EDP configuration
- NSX Advanced Load Balancer: single or dual stack for Load Balancer

Several options are available at the VNF and CNF level to consume IPV4 and IPv6 communications. Some constrains apply on the management component interactions. The following table summarizes the address family capabilities of the management components. IPv4 is standard for all components.

Table 4-6. IPv6 and Dual Stack Readiness of Managements Components

Components	Release	IPv6 only	Dual Stack	Notes
ESXi	**8.0b**	**X**	**X**	
vCenter Server	8.0b		X	
NSX	4.1.0.2		X	Management Plane and Control Plane TEP supports only IPv4.
Telco Cloud Automation	2.3	X	X	

Table 4-6. IPv6 and Dual Stack Readiness of Managements Components (continued)

Components	Release	IPv6 only	Dual Stack	Notes
ESXi	**8.0b**	**X**	**X**	
Tanzu Kubernetes Grid	2.1.1	X		Management and Workload clusters must have the same Address Family.
Aria Operations	8.10	X	X	All nodes in the cluster must follow the Address Family.
Aria Operations for Logs	8.10	X	X	
Aria Automation Orchestrator	8.11			
Aria Operations for Networks	6.9	X	X	
VMware Cloud Director	10.4.1			
Bare Metal Automation	3.0			
VMware Integrated OpenStack	7.3			
NSX Advanced Load Balancer	22.1.3			
Avi Kubernetes Operator	1.9.2	X	X	Management plane to NSX ALB can be IPv4 only.

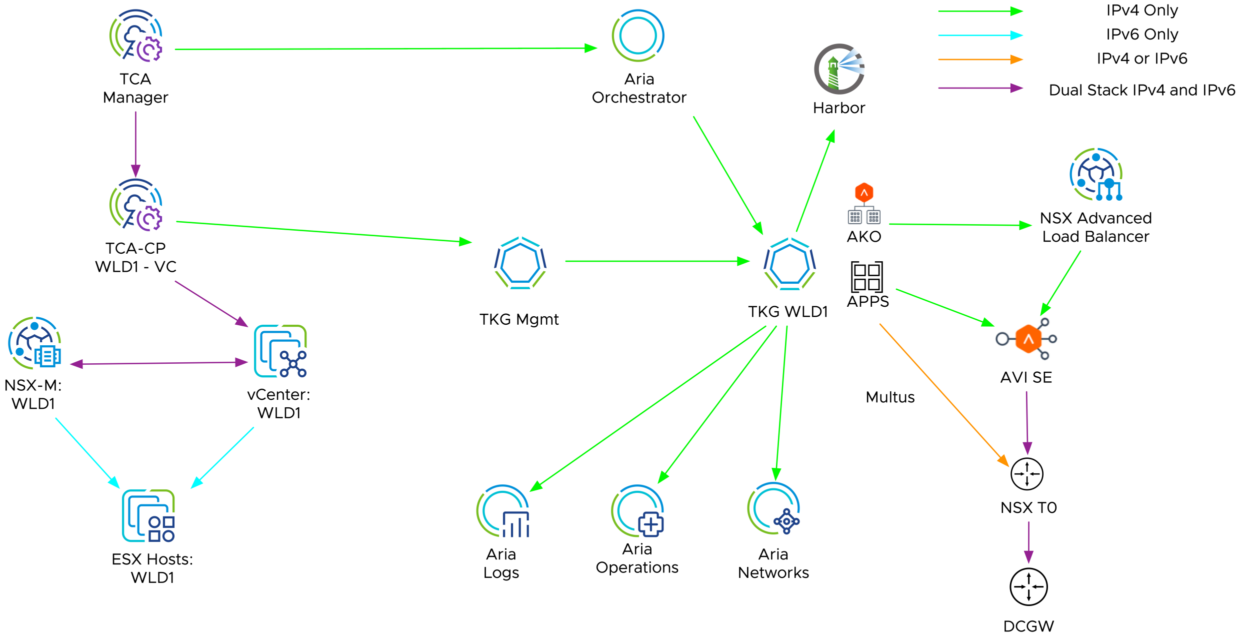
Note Due to the interaction of the components in a deployment and the limitations of some components, only part of the stack can be deployed in an IPv6 or a Dual Stack scenario.

Telco Cloud IPv6 Ingress in CaaS

The following diagram illustrates a deployment where IPv4 and IPv6 coexist. Some components are configured with dual stack and can communicate with components that have limitations in supported address families. Aria Automation Orchestrator is one such component that supports only IPv4. In the diagram, ESXi hosts are configured in IPv6 only since they are connected to vCenter, NSX, and TCA that are configured in Dual Stack.

Even if the TKG Workload is configured in IPv4, only the NSX Advanced Load Balancer enables to export IPv4, IPv6, and Dual Stack routes. This allows CNFs to export ingress VIPs in IPv4 and IPv6. The additional interfaces consumed by the CNFs with multus can be configured in IPv4 or IPv6 but not in a Dual Stack configuration. Both ingress and egress CNF communications of any address family can be aggregated over NSX gateways.

Figure 4-14. Telco Cloud IPv6 Ingress in CaaS



Telco Cloud IPv6 VNF

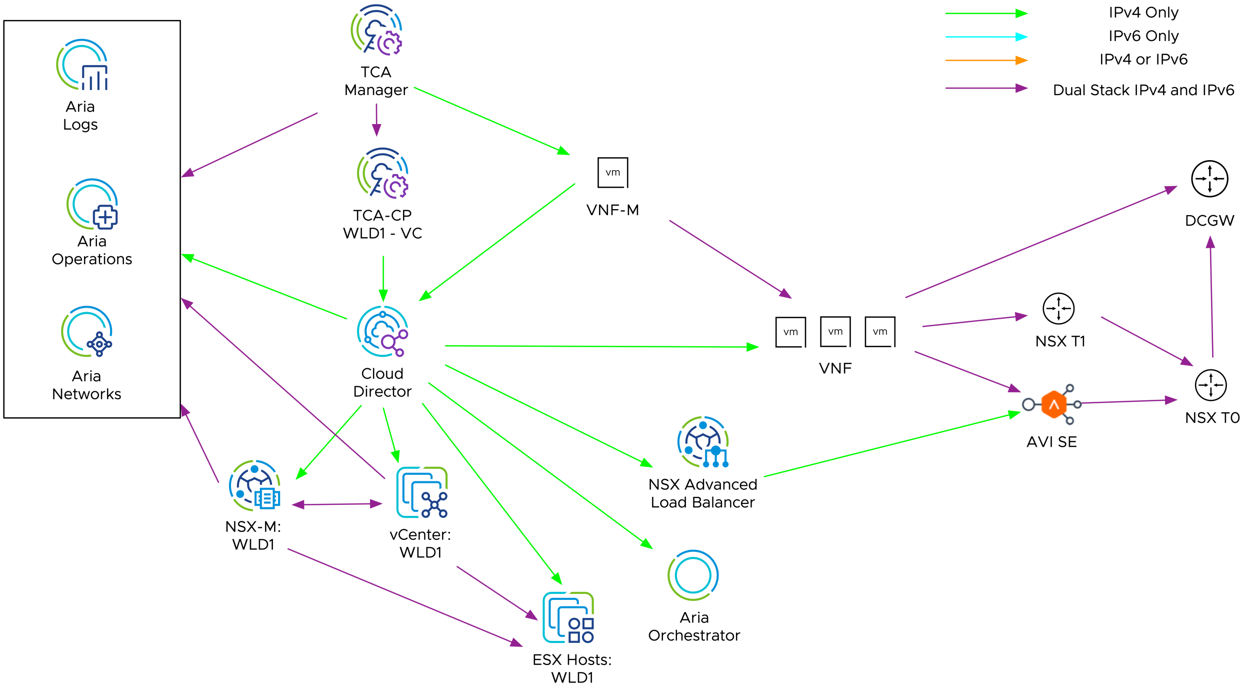
The following diagram illustrates a deployment where most of the management components are configured with IPv4. This is the result of limitation in some components such as VMware Cloud Director and Aria Automation Orchestrator.

Even if the management communications are managed with IPv4, the hosted VNFs can be configured with IPv4, IPv6, or dual stack. Several options are available to be consumed by VNFs:

- Dual Stack Overlay segments
- Dual stack Trunk VLANs configured with EDP
- Dual stack load balancer configured on NSX Advanced Load Balancer

The VNF Manager can be configured with Dual Stack and it can communicate with the VNF over IPv6 and with VMware Cloud Director over IPv4.

Figure 4-15. Telco Cloud IPv6 VNF

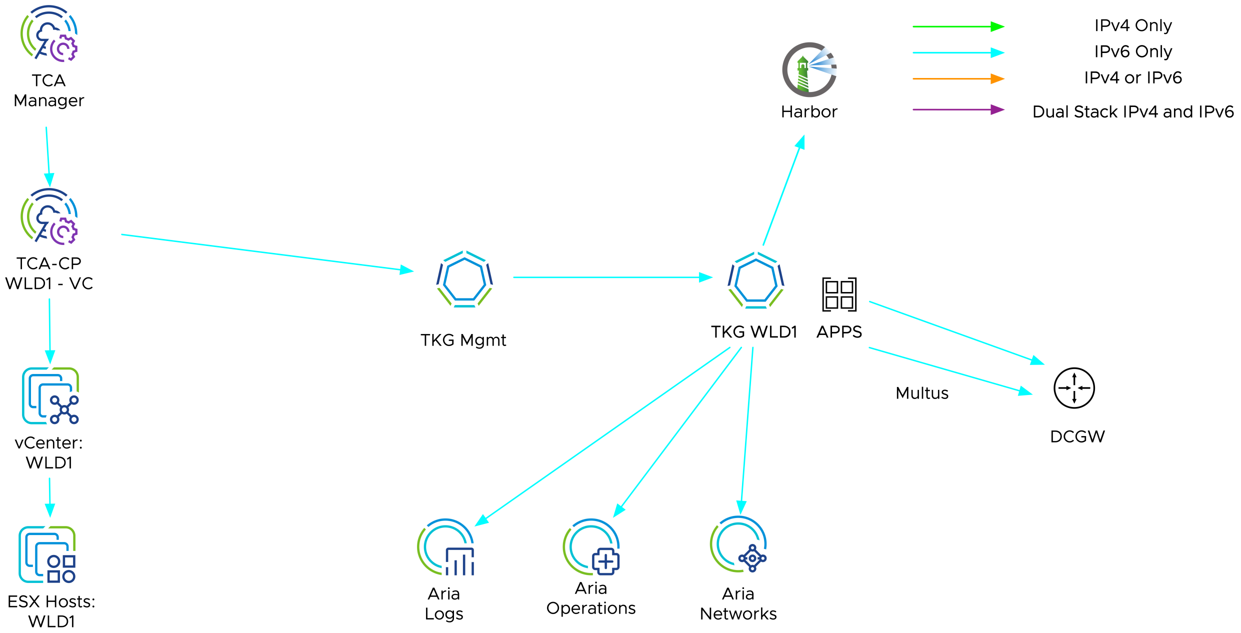


Telco Cloud RAN IPv6

The following diagram illustrates an IPv6 environment where all the components are connected with IPv6 interfaces. Some components such as TCA Manager, TCA control plane, and vCenter might be configured in a dual stack configuration. Other components such as TKG Management and TKG Workload are both configured in IPv6 only since they are not supporting dual stack.

Note Aria Automation Orchestrator is not used in this example since it supports only IPv4. CNFs can use Javascript-based workflows directly from TCA.

Figure 4-16. Telco Cloud RAN IPv6



Virtual Infrastructure Manager Design

The Virtual Infrastructure Manager (VIM) is a key component in the Telco Cloud framework. The VIM abstracts and manages the underlying infrastructure components and presents them to external cloud orchestration platforms. The VIM provides the foundations for multi-tenancy, compute scheduling, and resource allocation across the telco cloud.

.

VMware Cloud Director

VMware Cloud Director is a Telco Cloud component that functions as an interface to VNF services. It uses vCenter Server and NSX Manager to orchestrate compute, storage, and networking from a single programmable interface.

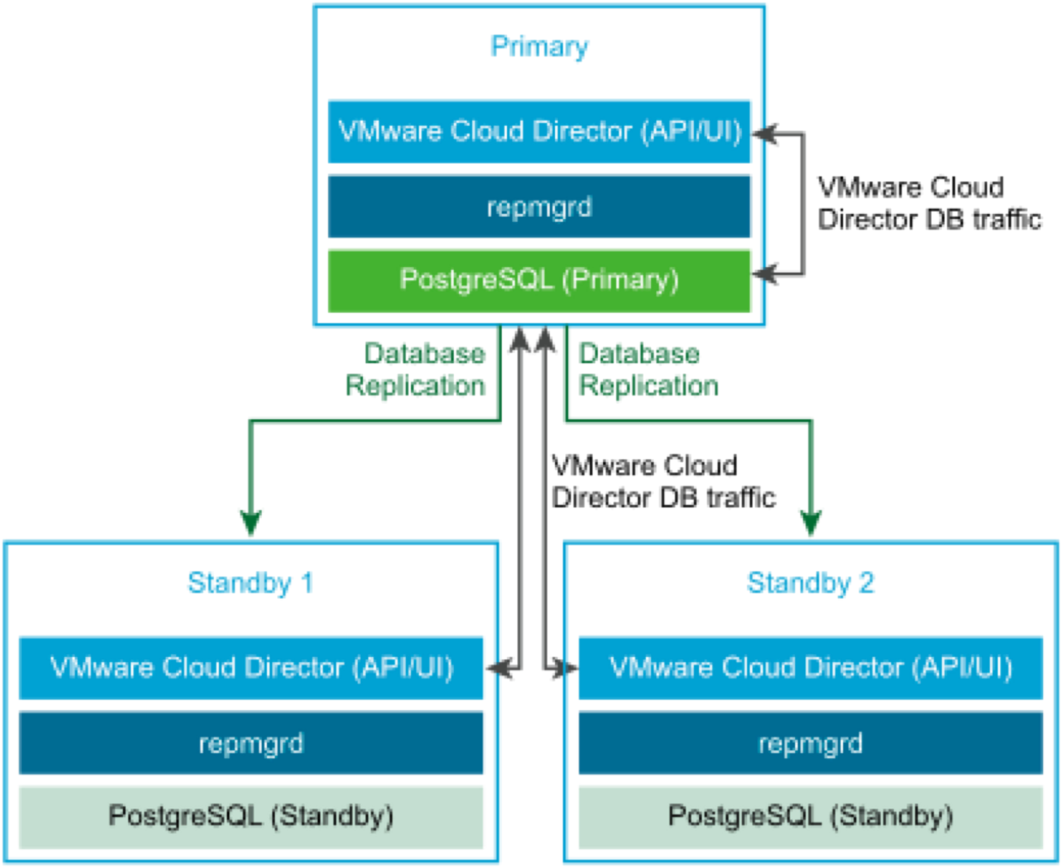
Cloud Director Cell Design

The Cloud Director design is based on two deployment components: database nodes (primary and secondary nodes) and cells.

The Cloud Director database cluster must be deployed in a HA cluster. It leverages an embedded postgreSQL database and incorporates replication to provide a highly-available cluster of Cloud Director appliances.

The database cluster must be configured in 3 nodes: one Primary Node and two secondary nodes. Additional scaling of the Cloud Director deployment can be achieved by deploying individual application cells.

Figure 4-17. Cloud Director HA Database Architecture


Primary
VMware Cloud Director (API/UI)
repmgrd
PostgreSQL (Primary)
VMware Cloud Director DB traffic
Database Replication
Database Replication
VMware Cloud Director DB traffic
Standby 1
VMware Cloud Director (API/UI)
repmgrd
PostgreSQL (Standby)
Standby 2
VMware Cloud Director (API/UI)
repmgrd
PostgreSQL (Standby)

Each server in the group runs a collection of services called a VMware Cloud Director cell. All cells share a single VMware Cloud Director database, transfer server storage, and connect to the vSphere and network resources. The installation and configuration process of VMware Cloud Director creates the cells, connects them to the shared database, and transfers server storage.

As opposed to the database nodes, Cloud Director cells provide the cloud managment components. In a stateless appliance, all updates from the cells are written to the postgres database stored in the HA database cluster.

The cells communicate with each other through an ActiveMQ message bus on the primary interface. They also share a common Cloud Director database where the cells persist configuration and state data. The transfer service requires that all cells have access to a common NFS mount.

VMware Cloud Director appliances come in different sizes (medium, large, and extra-large). The recommended deployment size for Cloud Director is large. If Cloud Director is integrated with Aria Operations, extra-large is recommended.

All Cloud Director cells provide the portal, API, and VMware Remote Console (VMRC) access. Thus, a load balancer must be deployed to load balance the traffic between the cells.

Any load balancer can be used as long as it is configured for sticky sessions to reduce the Cloud Director session database API calls. However, the recommended load balancer is NSX Advanced Load Balancer. The Load Balancer configuration can terminate the SSL and re-create new SSL connections to the Cloud Director cells based on the algorithm such as least_connections.

Note Both the Cloud Director API and the Remote Console share the same port, so separate certificates are no longer necessary.

Table 4-7. Recommended VMware Cloud Director Cell Design

Design Recommendation	Design Justification	Design Implication
Deploy nodes using the VMware Cloud Director Appliance.	Ensures a consistent deployment across cells.	None
Deploy at least 3 database nodes: one Primary and two Secondary of extra-large size.	Provides resiliency across nodes and tolerates node failure	Consumes more resources in the management cluster
Configure VMware Cloud Director for manual Primary Cell failover.	Simplifies the BCDR configuration and process	Can increase the downtime of VMware Cloud Director by eliminating automated failover of the Primary Cell.
Ensure that cells can communicate with each other through the message bus on the primary network interface.	Ensures that cells do not enter a split-brain state.	None
Use the same consoleproxy certificate on all cells.	Ensures that users can connect to the consoleproxy service regardless of the cell they are connected to.	You must manually install the certificate on each cell.
Verify that the cell transfer share is accessible for all cells.	Required for the proper functioning of VMware Cloud Director.	None
Configure NTP for all Database nodes, cells, and NFS server	Ensures accurate clocking across all components, including NFS	None

Table 4-8. Recommended Sizing for Cloud Director

Attribute	Specification
Overall Appliance Size	Extra Large
Database Cells	
Number of vCPUs	24
Memory	32 GB
Disk Space	Minimum 120 GB
Cell Appliance	
Number of vCPUs	8

Table 4-8. Recommended Sizing for Cloud Director (continued)

Attribute	Specification
Memory	8 GB
Disk Space	Minimum 120 GB

Cloud Director NFS - Transfer Storage

Cloud Director requires an NFS volume to act as a shared storage to all nodes within a Cloud Director deployment. This shared storage is used for cluster management and providing temporary storage for uploads, downloads, and in some conditions catalog storage.

One of the common approaches to provide the NFS storage is by using vSAN File Services. Use vSAN File Services to provide NFS 3.0 / 4.x shares that can be leveraged by the Cloud Director deployment. Other approaches include external storage arrays providing NFS capabilities.

Note The solution that you use must be highly available.

The transfer storage is used when multiple cells are deployed. In this case, the Cloud Director cells use the NFS server as a shared, temporary repository for all uploads, downloads, and cloning operations. The data is removed from the transfer storage location when the operation is complete. However, the overall size of the NFS storage depends on the number of concurrent operations and the workload size.

Authentication

You can integrate VMware Cloud Director with an external identity provider and import users and groups to your organizations. You can configure an LDAP server connection at a system or organization level and a SAML integration at an organizational level.

- **LDAP Server**: You can configure an organization to use the system LDAP connection as a shared source of users and groups. An organization can also use a separate LDAP connection as a private source of users and groups.
- **SAML Identity Provider**: To import users and groups from a SAML identity provider to your system organization, you must configure your system organization with this SAML identity provider. Imported users can log in to the system organization with the credentials established in the SAML identity provider.

  To configure VMware Cloud Director with a SAML identity provider, establish a mutual trust by exchanging SAML service provider and identity provider metadata.

Table 4-9. Recommended Roles and Authentication Design for VMware Cloud Director

Design Recommendation	Design Justification	Design Implication
Use the default VMware Cloud Director roles, unless necessary.	Simplifies the user rights management and configuration	Custom roles might be required for some cases where the built-in roles do not work.
Configure a system LDAP connection.	■ Enables centralized account management by leveraging the existing LDAP infrastructure. ■ Provides high security, as you do not need to create local accounts that can be left unused.	Requires manual user import and role assignment
Use the system LDAP connection for organizations.	■ Enables centralized account management by leveraging the existing LDAP infrastructure. ■ Provides a high level of security as local accounts are not required.	Requires manual user import and role assignment

Cloud Director Scaling Considerations

A single Cloud Director deployment manages resources from multiple data centers. While all the Cloud Director nodes must reside in a single management domain due to latency constraints, the vCenter Servers that the Cloud Director manages can be distributed geographically.

Considerations for managing remote resource vCenter Servers:

- vCenter Server and ESXi Latency: Maximum 150ms RTT between Cloud Director cells and vCenter servers / ESXi hosts.
- NSX Latency: Maximum 150ms RTT between Cloud Director cells and NSX Managers.

Cloud Director supports scale-up and scale-out models. When a large deployment size is used, the cloud director cell can be taken out of service, powered off, and resized. Upon reboot, the node is reconfigured to reflect the new sizing.

The most common approach to scaling is to add more cells to the system. The recommended number of cells is n+1, where n is the number of resource vCenter Servers managed by Cloud Director.

Cloud Director Tenancy

Multi-tenancy is one of the key constructs in Cloud Director. This section covers the constructs of Provider Virtual Data Centers, Organization, and Organizational Virtual Data Centers.

Provider Virtual Data Center

A provider Virtual Data Center (pVDC) makes the vSphere compute, memory, and storage resources available to VMware Cloud Director. The pVDC aggregates the resources within a single vCenter Server by mapping several or all the vSphere clusters. The pVDC also is associated with Datastores (vSAN or otherwise) and storage policies are made available within the pVDC. The pVDC can be bound to a NSX network pool or vSphere VLAN-backed network pools to create networks.

Note A pVDC network can be created without the network pool backing. In this case, the networks cannot be auto-created. Networks can only be consumed, implying that all network segments and port-groups must be pre-created.

Before an organization deploys VMs or creates catalogs, the system administrator must create a provider VDC and the organization VDCs that consume the Provider VDC resources. The relationship of provider VDCs to the organization VDCs they support is an administrative decision. The decision can be based on the scope of your service offerings, the capacity and geographical distribution of your vSphere infrastructure, and similar considerations.

Note When the deployment of VNFs and CNFs into a converged cluster is supported, an effective approach is to map the vSphere resource pools to a pVDC instead of adding the entire cluster. However, in most deployments, the entire vSphere cluster is added to the pVDC.

Because a pVDC provides an abstraction layer against which resources (compute, RAM, storage) are allocated to tenants, system administrators can create pVDCs that provide different classes of service, based on performance, capacity, and features. Tenants can then be provisioned with organization VDCs that deliver specific classes of service as defined by the pVDC. Before you create a provider VDC, consider the vSphere capabilities that you plan to offer your tenants. Some of these capabilities can be implemented in the primary resource pool of the pVDC.

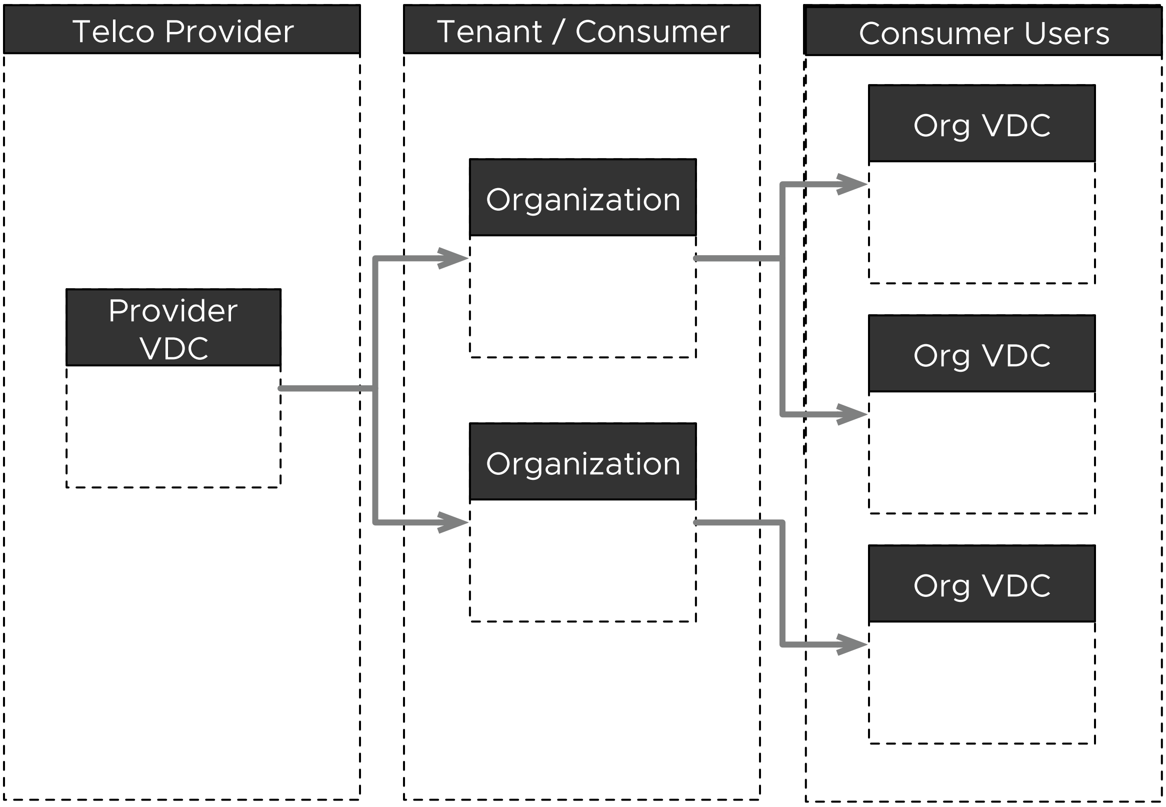
One such example of this is to create a separate PVDC for User-Plane versus Control plane workloads. However, pVDCs are more commonly used to represent different resource domains. The different workload types can be placed inside the Tenant Constructs (Organization / Organization VDC) and different resource allocation models can be used to separate out the different workload types.

The following diagram illustrates the relationship between Provider VDC, Organization, and Organization VDC.

Note

- An Organization can be built using resources from multiple Provider VDCs and is not tied to the Provider VDC construct.
- The Organization VDC construct is backed by the resources from a single pVDC, although different OrgVDCs within the same organization can be deployed across different pVDCs.

Figure 4-18. Relationship Between Cloud Director Constructs



Organization

The Organization construct is the key component of multi-tenancy in Cloud Director. The Organization is the logical security boundary within a Cloud Director tenant. The Organization is the construct within which all tenant resources reside. These resources include Organizational Virtual Data Centers, Users and IDM / SAML integration points, Role-Based Access Control, catalogs, networking constructs, and so on.

Organization Virtual Data Centers

Within an organization, units of resources are carved out of the Provider VDC (pVDC) in the form of an Organizational Virtual Data Center (OrgVDC). The pVDC creates the required OrgVDC as resource pools within the vCenter Server allocated to the pVDC. The configured resources of the OrgVDC (Compute, Memory, Storage Policies, and Network) have mapping constructs within vSphere to provide the required capabilities.

Note OrgVDC allows the tenant workloads to be instantiated and powered on.

Allocation Models

To allocate resources to an organization, you must create an OrgVDC. An OrgVDC obtains its resources from a pVDC. The allocation model determines how and when the compute and memory resources of pVDC are committed to the OrgVDC. The OrgVDC maps to resource pools on the underlying vSphere environment.

The allocation model defines the amount of available resources and the commitment of those resources to the VMs contained within the OrgVDC.

The following table describes the vSphere resource distribution settings at the VM or resource pool level based on the OrgVDC allocation model:

Allocation Model	Resource Pool Setting	Virtual Machine Setting
Allocation Pool	■ A percentage of resources is guaranteed. ■ Maximum limit is set on the resource pool	Resource guarantee and limits are inherited from the resource pool.
Pay-As-You-Go	No resource guarantee or limits at the resource pool level	Resource limits are set at the VM level based on the configuration of the OrgVDC.
Reservation Pool	■ Resources Guarantee is 100%, and the resource limit is equal to reservation. ■ Resource sets in the reservation pool are dedicated.	Resource Guarantees and Limits are not defined by default; however they can be configured per workload.
Flex	See the following Note.	See the following Note.

Note The Flex model can be implemented with compute and placement policies.

CPU and RAM configuration can be configured at both the OrgVDC and VM level. However, the placement and compute policies provide more flexibility in workload placement and sizing.

The Flex model supports all configurations possible through other allocation models.

Each allocation model can be used for different levels of performance control and management. The suggested uses of each allocation model are as follows:

- **Flex Model**:
  - With the flex model, you can achieve a fine-grained performance control at the workload level. VMware Cloud Director system administrators can manage the elasticity of individual organization VDCs. Cloud providers can have better control over memory overhead in an organization VDC and can enforce a strict burst capacity use for tenants.

    Note The flex allocation model uses policy-based management of workloads.

- **Allocation Pool Model**:
  - Use the allocation pool model for long-lived, stable workloads, where tenants subscribe to a fixed compute resource consumption and cloud providers can predict and manage the compute resource capacity. This model is optimal for workloads with diverse performance requirements.
  - With this model, all workloads share the allocated resources from the resource pools of vCenter Server.

  - Regardless of whether you activate or deactivate elasticity, tenants receive a limited amount of compute resources. Cloud providers can activate or deactivate the elasticity at the system level and the setting applies to all allocation pool organization VDCs. If you use the non-elastic allocation pool, the organization VDC pre-reserves the VDC resource pool and tenants can overcommit vCPUs but cannot overcommit any memory. If you use the elastic pool allocation, the organization VDC does not pre-reserve any compute resources, and capacity can span through multiple clusters.

    Note Cloud providers manage the overcommitment of physical compute resources and tenants cannot overcommit vCPUs and memory.

- **Pay-as-You-Go Model**:
  - Use the pay-as-you-go model when you do not have to allocate compute resources in vCenter Server upfront. Reservation, limit, and shares are applied on every workload that tenants deploy in the VDC.
  - With this model, every workload in the organization VDC receives the same percentage of the configured compute resources reserved. In VMware Cloud Director, the CPU speed of every vCPU for every workload is the same and you can only define the CPU speed at the organizational VDC level. From a performance perspective, because the reservation settings of individual workloads cannot be changed, every workload receives the same preference.
  - This model is optimal for tenants that need workloads with different performance requirements to run within the same organization VDC.
  - Because of the elasticity, this model is suitable for generic, short-lived workloads that are part of autoscaling applications.
  - With this model, tenants can match spikes in compute resources demand within an organization VDC.
- **Reservation Pool Model**:
  - Use this model when you need a fine-grained control over the performance of workloads that are running in the organization VDC.
  - From a cloud provider perspective, this model requires an upfront allocation of all compute resources in vCenter Server.

    Note This model is not elastic.

  - This model is optimal for workloads that run on hardware dedicated to a tenant. In such cases, tenant users can manage use and overcommitment of compute resources.

Traditionally, the Pay-As-You-Go model was used for control-plane based network functions, as the resource guarantees are a percentage of the overall requested resources. With the PAYG model, resources can be consumed continually within the pVDC limits.

User-Plane workloads were traditionally deployed into a reservation pool, although this pool does not support elastic resouces. Only a single resource from the pVDC can be used as available resources for this pool.

The flex model allows all the configuration settings (as listed in the table below) to be combined in different ways and function in the same method as other models but with additional flexibility and more granular control of VMs and compute policies.

The following table describes the configuration settings across different resource pool models.

	Flex Allocation Model	**Elastic Allocation Pool Model**	**Non-Elastic Allocation Pool Model**	**Pay-As-You-Go Model**	**Reservation Pool Model**
Elastic (Can consume multiple resource pools)	The Elastic setting is based on the organization VDC configuration.	Yes	No	Yes	No
vCPU Speed	If a VM CPU limit is not defined in a VM sizing policy, vCPU speed might impact the VM CPU limit within the VDC.	Impacts the number of running vCPUs in the Organizational VDC.	Not Applicable	Impacts the VM CPU limit.	Not Applicable
Resource Pool CPU Limit	The CPU limit of an Organizational VDC is defined based on the number of VMs in the resource pool.	Organization VDC CPU allocation	Organization VDC CPU allocation	Unlimited	Organization VDC CPU allocation
Resource Pool CPU Reservation	The CPU reservation of an Organization VDC is defined based on the number of vCPUs in the resource pool. Organization VDC CPU reservation equals the organization VDC CPU allocation times the CPU guarantee.	Sum of powered-on VMs and equals the CPU guarantee multiplied by the vCPU speed, multiplied by the number of vCPUs.	Organization VDC CPU allocation multiplied by the CPU guarantee	None, expandable	Organization VDC CPU allocation
Resource Pool Memory Limit	The memory limit of an Organizational VDC is apportioned based on the number of VMs in the resource pool.	Unlimited	Organization VDC RAM allocation	Unlimited	Organization VDC RAM allocation

	Flex Allocation Model	Elastic Allocation Pool Model	Non-Elastic Allocation Pool Model	Pay-As-You-Go Model	Reservation Pool Model
Resource Pool Memory Reservation	The RAM reservation of an Organization VDC is apportioned based on the number of VMs in the resource pool. The organization VDC RAM reservation equals the organization VDC RAM allocation times the RAM guarantee.	Sum of RAM guarantee times vRAM of all powered-on VMs in the resource pool. The resource pool RAM reservation is expandable.	Organization VDC RAM allocation times the RAM guarantee	None, expandable	Organization VDC RAM allocation
VM CPU Limit	Based on the VM sizing policy of the VM	Unlimited	Unlimited	vCPU speed times the number of vCPUs	Custom
VM CPU Reservation	Based on the VM sizing policy of the VM	0	0	Equals the CPU speed times the vCPU speed, times the number of vCPUs	Custom
VM RAM Limit	Based on the VM sizing policy of the VM	Unlimited	Unlimited	vRAM	Custom
VM RAM Reservation	Based on the VM sizing policy of the VM	0	Equals vRAM times RAM guarantee plus RAM overhead	Equals vRAM times RAM guarantee plus RAM overhead	Custom

Table 4-10. Recommended Tenancy Considerations

Design Recommendation	Design Justification	Design Implication
Create at least a single Provider VDC.	Required to create Organizations and corresponding Organization VDCs.	None
Create an Organization per vendor.	Ensures isolation between different vendors in the environment.	None
Create at least a single Organization VDC per Organization.	Allows the Organization to deploy workloads.	If not sized properly, an Organization VDC can have unused or overcommitted resources.

Table 4-10. Recommended Tenancy Considerations (continued)

Design Recommendation	Design Justification	Design Implication
Use the Flex allocation model.	Allows for fine-grained control of resource allocations to each organization VDC.	None
Configure storage and runtime leases for production VDCs to not expire.	Production workloads must be run until their end of life and then decommissioned manually	By not setting any leases, workloads that are no longer being used continue to run, resulting in wasted resources.

Cloud Director Policies

Administrators can configure Provider VDC compute policies that provide Host affinity type rules, allowing VMs to run on a constrained set of hosts. Such constraints can include the requirements for SR-IOV, GPU-enabled nodes, licensing issues, or other concerns the provider might have to address. These policies align with DRS VM/Host mappings within the vSphere environment.

The OrgvDC compute policies consist of placement and sizing policies. Placement policies allow the OrgVDC to consume one or more provider placement policies. This capability provides the flexibility to place different network functions or even different elements within the same network function to different placement policies. This simplifies and ensures the placement of User-Plane and Control-Plane policies.

The Sizing policy allows the provider to provide machine sizes in terms of CPU and Memory resources. The Sizing policy allows the tenant to choose the sizing policy on a per VM basis to suit the deployment model. When using a compute policy, the provider can specify the number of vCPUs, the clock speed, and the amount of memory, and the reservation and limits.

Note These policies in conjunction with the Flex allocation model ensure pre-determined placement and sizing for tenant workloads.

A single OrgVDC compute policy (a placement policy or a sizing policy) can be applied as the organisation default through the UI.

Without a default sizing policy, reservations and CPU speeds are configured as per the OrgVDC default configuration.

Cloud Director Networking

Cloud Director networking enable the provider and tenants to create and consume networking constructs from a vSphere or NSX environment. Hence, tenants can create network segments and configure network services including DHCP, NAT firewalling and load-balancer integrations.

Note This section describes how the vSphere and NSX networking constructs can be created and consumed within Cloud Director. The general networking concepts of vSphere or NSX are not covered.

Network Pools

A network pool is a group of networks available for an organization VDC to create routed networks and certain types of organization VDC networks. VMware Cloud Director uses network pools to create NAT-routed and internal organization VDC networks and vApp networks. Network traffic on each network in a pool is isolated from all other networks at Layer 2.

Each organization VDC in VMware Cloud Director can have one network pool. Multiple organization VDCs can share a network pool. The network pool for an organization VDC provides the networks to satisfy the network quota for an organization VDC.

Every provider VDC that is backed by NSX includes a Geneve network pool. When you create a an NSX-backed provider VDC, you can associate that provider VDC with an existing Geneve network pool or create a new Geneve network pool for the provider VDC.

VMware Cloud Director Geneve networks has various benefits:

- Logical networks spanning Layer 3 boundaries
- Logical networks spanning multiple racks on a single Layer 2
- Broadcast containment
- High performance
- Increased scaling (up to 16 million network addresses)

If NSX is not in use, a VLAN-backed network pool can also be leveraged. When using a VLAN-backed network pool, the provider specifies the VLAN ranges that can be consumed by the pool. When an OrgVDC network is created, a port-group is created on the vSwitch with the next available VLAN from the network pool.

External Networks

A Cloud Director external network provides an uplink interface that connects networks and VMs in the system to a network outside of the system. For example, a VPN, a corporate intranet, or the public Internet.

External networks are provider-level configurations and they are not created at the tenant level. When creating an External Network in Cloud Director, the vSphere or NSX Adminstrator must provision the network segment first so Cloud Director can consume them.

Note The range of IP addresses defined for the external network are allocated either to an edge gateway or to the VMs that are directly connected to the network. Hence, the IP addresses must not be used outside of VMware Cloud Director.

An external network can be backed by an NSX tier-0 logical router. You can also create an external network that is backed by a VRF-lite tier-0 gateway in NSX. A VRF gateway is created from a parent tier-0 gateway. It has its own routing tables. Multiple VRFs can exist within the parent tier-0 gateway. This allows VDCs to have their own external network without deploying multiple tier-0 gateways.

Organizational VDC Networks

OrgVDC networks can be created within the tenant space of Cloud Director, so that tenant workloads can have both internal and external connectivity. OrgVDC networks are typically used for internal traffic, however an OrgVDC network can be connected directly to external networks to create different networking topologies.

The following OrgVDC network types are used for workload traffic:

- **Direct Networks**: The direct network allows an exit from within the Organization VDC to an external network. The direct network is connected directly to a vSphere port-group that hosts the network configuration.

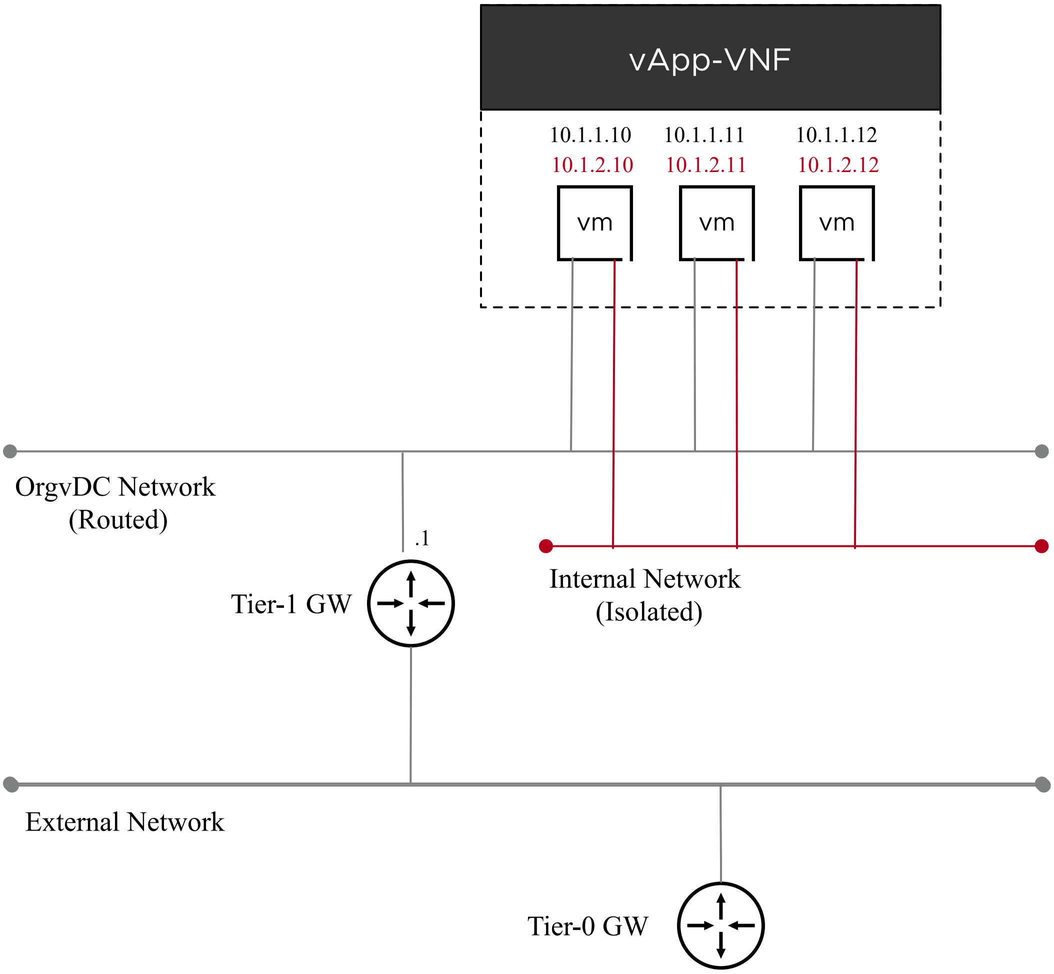
  Note When using NSX (without N-VDS), the NSX segments show up as port-groups in the vSphere vSwitch. You can also create an NSX overlay segment as an external network and then create a direct network that is backed by this network.

  This approach allows direct VM to T0 communication and avoids the T1 construct created within the Cloud Director tenant.

- **Internal Networks**: The internal network is created within an organisation and is not connected to any constructs. It is neither backed by an external network nor connected to a tenant edge. The internal network is not routed, so all VMs that need to communicate must have an attachment to this network.
- **Routed Network**: The routed network is a network segment that is connected to an NSX Edge. The routed networkis attached to a tenant edge gateway and can be configured as routable through the external connection provided by the provider gateway configuration.

The following figure illustrates a single Organisation within which a vApp with a router network is connected to an NSX Tier-1 gateway (deployed through Cloud Director) and an isolated network that is not connected outside of the organisation.

Figure 4-19. Cloud Director OrgVDC Networks



Depending on the use case, networking constructs within Cloud Director can be shared across multiple OrgVDCs within an organisation, allowing networks to span beyond the OrgVDC construct.

Cloud Director - NSX Networking

An NSX edge gateway provides a routed organization VDC network or a data center group network with connectivity to external networks and IP management properties. It can also provide services such as firewall, NAT, IPSec VPN, DNS forwarding, and DHCP, which is enabled by default.

Provider Gateways: The provider gateway brings an NSX Tier-0 or Tier-0 VRF construct into Cloud Director. The provider gateway is created at the provider level.

When building a provider gateway, the administrator selects the NSX manager and the T0 construct to bring into Cloud Director and the user also specifies an IP Block.

Note When creating a provider gateway, nothing new is created on NSX. Tenant edges that are connected to the provider gateway create an instantiation of a Tier-1 Gateway within NSX.

Note Mixed-mode Provider Gateways is not supported. For example, you cannot have a Provider Gateway that is using T0-VRFs and also the same parent T0.

Edge Gateways: The edge gateway represents the instantiation of a Tier-1 gateway within an Organizational VDC. The edge gateway connects to the Provider Gateway created by the administrator. The edge gateway belongs to a single OrgVDC.

When creating the edge gateway, the specified provider gateway can be marked as dedicated. This enables some additional configuration options within Cloud Director such as Route Advertisment control and BGP routing configuration. Any BGP configuration is applied at the provider gateway as the edge gateway (as a Tier-1 Gateway) does not support BGP.

By using route advertisement, you can create a fully-routed network environment in an OrgVDC. You can decide which network subnets that are attached to the NSX edge gateway can be advertised to the dedicated external network. If a subnet is not added to the advertisement filter, the route to it is not advertised to the external network and the subnet remains private. Route advertisement is automatically configured on the NSX edge gateway.

VMware Cloud Director supports automatic route redistribution when you use route advertisement on an NSX edge gateway. Route redistribution is automatically configured on the tier-0 logical router that represents the dedicated external network. You can configure an external or internal Border Gateway Protocol (eBGP or iBGP) connection between an NSX edge gateway that has a dedicated external network and a router in your physical infrastructure.

In an edge gateway that is connected to an external network backed by a VRF gateway, the local BGP AS number and graceful restart settings are inherited from the parent Tier-0 gateway and they cannot be changed at the VRF level.

You can also attach external networks directly to the Tier-1 Gateway, by selecting the external networks. The port-group is connected to the Tier-1 Gateway as a service port that exists on the service node where the Tier-1 is deployed.

Cloud Director Recommendations

Design Recommendation	Design Justification	Design Implication
Create a Geneve Network pool.	Required to create NSX-T backed network resources.	None
Create a dedicated external network per Organization.	Allows the use of a fully routed network topology.	None

Design Recommendation	Design Justification	Design Implication
Create a VRF per Organization.	Allows all Organizations to share a single parent Tier-0 gateway, while maintaining isolation between Organizations. Sharing a single parent gateway reduces the number of NSX Edges required in the deployment.	VRFs inherit the local AS and graceful restart configuration of the parent Tier-0, such that each VRF has the same local AS.
Create one or more Edge Gateways (Tier-1 gateways) per Organization VDC.	Enables networking services for the Organization VDC.	None
Create DC Groups, if there is a need to have OrgvDC networks spanning multiple OrgVDCs in an organization.	Simplifies cross OrgVDC connectivity	Requires the creation of DC groups within the Cloud Director tenant.
Use Legacy IP Block addressing over IP Spaces feature.	**Note**: The delivery of the complete IP Spaces feature set span multiple releases.	None

Cloud Director Storage and Libraries

Cloud Director Storage enables the use storage-based policies within an Organization.

Storage-Policy Based Management

Storage-Policy Based Management (SPBM): VMware Cloud Director uses SPBM to define storage characteristics. In a software-defined data center, SPBM helps align storage with application demands of VMs. It provides a storage policy framework that serves as a single unified control panel across a broad range of data services and storage solutions.

As an abstraction layer, SPBM abstracts storage services delivered by vVols, vSAN, I/O filters, and other storage entities. Instead of integrating with each type of storage and data services, SPBM provides a universal framework for different types of storage entities.

SPBM provides the following mechanisms:

- Advertisement of storage capabilities and data services offered by storage arrays and other entities such as I/O filters.
- Bidirectional communications between ESXi and vCenter Server on one side and between storage arrays and entities on the other side.
- VM provisioning based on VM storage policies.

VMware Cloud Director uses the SPBM policies defined in vCenter Server. These policies are assigned to a Provider VDC and are available to Organization VDCs managed by the Provider VDC.

By using SPBM, a provider can have different storage tiers within the same Provider vDC.

IOPS: You can enable the I/O operations per second (IOPS) setting for a storage policy so that tenants can set per-disk IOPS limits.

Managed read/write performance in physical storage devices and virtual disks is defined in units called IOPS, which measure read/write operations per second. To limit I/O performance, a provider VDC storage policy that includes storage devices with a configured IOPS allocation must back an organization VDC storage policy. Afterwards, a tenant can configure disks to request a specified level of I/O performance. A storage profile configured with IOPS support delivers its default IOPS value to all disks that use it. The disks include the ones that are not configured to request a specific IOPS value. A hard disk configured to request a specific IOPS value cannot use a storage policy that has a maximum IOPS value lower than the requested value or a storage policy that is not configured with IOPS support.

You can edit the default IOPS settings. For example, you can set limits on IOPS per disk or IOPS per storage policy. The IOPS limits per disk are set based on the disk size in GB so that you grant more IOPS to larger disks. Tenants can set custom IOPS on a disk within these limits. You can use IOPS limiting with or without IOPS capacity considerations for placement.

You cannot enable IOPS on a storage policy backed by a Storage DRS cluster. The storage policies depends on the workload being deployed, so generic design recommendations become irrelevant. Instead, create storage policies based on the storage type and workload demands such that the vendor requirements are met.

Cloud Director Catalogs

Catalogs allow both Providers and Tenants to create catalogs and provide a simple way to handle the storage of ISO files and OVF templates.

Public catalogs can be created by the provider and shared amongst multiple tenants. Public catalogs can include standard customer images such as Jumphosts, PaaS services, and other provider-specific VMs that can be consumed by one or more tenants (organizations).

The primary use of Catalogs within an Organization is to allow the tenants in the organization to upload their applications ready for deployment. When an Organizational-level catalog is created, it can be configured to use a specific storage policy.

Organization catalogs can be private, shared, or published:

- Shared: The shared model allows the catalog to be viewed only by a specific set of users or groups within the current organisation. This allows different catalogs to be shared with different tenant users.
- Published: The publish model allows a catalog to be published. Any organzation can subscribe to this catalog, if it has the published URL for the catalog.

Cloud Director RBAC

VMware Cloud Director uses roles and associated rights to determine whether a user or group is authorized to perform an operation. Most of the procedures documented in the VMware Cloud Director guides include a predefined role. This predefined role includes a specific set of rights.

System administrators can use rights bundles and global tenant roles to manage the rights and roles in each organization.

Predefined Provider Roles:

- **System Administrator**: Exists only in the provider organization. It includes all rights in the system. A System Administrator can create additional system administrators and user accounts in the provider organization.
- **Multisite System**: Runs the heartbeat process for multisite deployments. It includes only one right 'Multisite: System Operations' to make a Cloud Director OpenAPI request. This request retrieves the status of the remote member of a site association.

Predefined Global Tenant Roles:

- **Organization Administrator**: Manages users and groups in organizations and assign them roles, including the predefined Organization Administrator role. Roles created or modified by an Organization Administrator are not visible to other organizations.

  Note After creating an organization, a System Administrator can assign the role of Organization Administrator to any user in the organization.

- **Catalog Author**: Creates and publishes catalogs
- **vApp Author**: Uses catalogs and creates vApps
- **vApp User**: Uses existing vApps
- **Console Access Only**: Views VM state and properties and uses the guest OS
- **Defer to Identity Provider**:
  - The rights associated with this role are determined based on the information received from the user's OAuth or SAML Identity Provider.
    - If an OAuth Identity Provider defines a user, the user is assigned the roles named in the roles array of the user's OAuth token.
    - If a SAML Identity Provider defines a user, the user is assigned the roles named in the SAML attribute. The SAML attribute name appears in the RoleAttributeName element, which is in the SamlAttributeMapping element in the organization's OrgFederationSettings.
  - When assigning this role to a user or group, the user or group name provided by the Identity Provider must match the role or group name defined in your organization. Otherwise, the user or group is not qualified for inclusion.
    - If a user is assigned the Defer to Identity Provider role but no matching role or group name is available in your organization, the user can log in to the organization but has no rights.

- If an Identity Provider associates a user with a system-level role such as System Administrator, the user can log in to the organization but has no rights. You must manually assign a role to such users.

Note Except the Defer to Identity Provider role, each predefined role includes a set of default rights. Only a System Administrator can modify the rights in a predefined role. If a System administrator modifies a predefined role, the modifications propagate to all instances of the role in the system.

Cloud Director Authentication

You can integrate VMware Cloud Director with an external identity provider and import users and groups to your organizations. You can configure an LDAP server connection at a system or organization level and a SAML integration at an organization level.

- **LDAP**: An organization can use the system LDAP connection as a shared source of users and groups or a separate LDAP connection as a private source of users and groups.
- **SAML**: If you want to import users and groups from a SAML identity provider to your system organization, configure the system organization with the SAML identity provider. Imported users can log in to the system organization with the credentials established in the SAML identity provider.

  To configure VMware Cloud Director with a SAML identity provider, you must establish a mutual trust by exchanging SAML service provider and identity provider metadata.

Table 4-11. Recommended Roles and Authentication Design for VMware Cloud Director

Design Recommendation	Design Justification	Design Implication
Use the default VMware Cloud Director roles, unless necessary.	Simplifies the user rights management and configuration.	Custom roles might be required for some cases where the built-in roles do not work.
Configure a system LDAP connection.	■ Enables centralized account management by leveraging the existing LDAP infrastructure. ■ Provides high security, as you do not need to create local accounts that can be left unused.	Requires manual user import and role assignment
Use the System LDAP connection for Organizations.	Enables centralized account management by leveraging the existing LDAP infrastructure. Provides a high level of security as local accounts, which can be left when a user leaves, do not need to be created.	Requires manual user import and role assignment

Cloud Director and Telco Cloud Automation Integration

VMware Telco Cloud Automation uses the Virtual Infrastructure Managers (VIMs) as an endpoint for VNF deployments. This section describes the considerations for using Telco Cloud Automation and a VIM for ETSI-based VNF deployments and lifecycle management.

When integrating Cloud Director and Telco Cloud Automation, complete the following steps before instantiating any VNFs.

- Install and configure a dedicated Telco Cloud Automation Control Plane (TCA-CP) node to communicate with Cloud Director. After configuring the TCA-CP node, it can be added as a Virtual Infrastructure account within Telco Cloud Automation.
  - Use a system adminstrator account to configure the TCA-CP node with Cloud Director.
  - Use the administrator account from the Cloud Director Organizaation to add the virtual infrastructure accounts.
- Deploy and configure RabbitMQ to enable Cloud Director communicate inventory to the TCA-CP node.
- Add a Virtual Infrastructure Account into Telco Cloud Automation, using the Telco Cloud Automation Control Plane - Cloud Director endpoint.
  - Use the Organizational Administrator credentials when adding a Cloud Director endpoint.
- Add compute profiles to the Cloud Director Virtual Infrastructure account.

Note The Virtual Infrastructure account links to the specific Organization, where as the compute profile links to the individual OrgVDCs within an Organization.

Cloud Director and RabbitMQ

This section describes the requirements for creating Cloud Director Virtual Infrastructure (VI) and integrations between VMware Cloud Director, RabbitMQ (AMPQ), and VMware Telco Cloud Automation.

- RabbitMQ is a message broker supporting the Advanced Message Queueing Protocol (AMQP). Different software components can connect to Rabbit MQ and transfer messages to specific queues. Other software components can then take those messages off the queue for processing.
- If Telco Cloud Automation is used to instantiate VNFs to Cloud Director, Cloud Director posts notification messages to RabbitMQ. The messages include notifications of inventory, inventory changes, events, and so on.

Cloud Director must be configured under the Administration -> Extensibility tab to send messages to the RabbitMQ deployment. Non-Blocking AMQP notifications must be configured to enable Telco Cloud Automation to receive messages from Cloud Director.

Deploying VNFs with Telco Cloud Automation and Cloud Director

When instantiating Cloud Director and Telco Cloud automation for VNFs, a NF can be instantiated in various ways.

- Using templates on the vCenter Server
- From a Cloud Director catalog using individual vAPP templates
- From a Cloud Director catalog as a VNF

When using the vAPP model, each VDU created in the NF designer is instantiated as an individual VM/vAPP. This creates vAPP sprawl within Cloud Director.

Using the VNF Model, each VDU is created as a VM within a single vAPP. When the VNF is scaled-out, more VMs are created for the component scaled within the same vAPP.

Note In this model, the Name value (not image name) as configured in the VDU is used to reference the vAPP in the vCD Catalog. Within the vAPP template inside the Cloud Director catalog, the VM name must equal the name (not image name) as specified for each VDU within the CSAR.

When designing a VNF through Telco Cloud Automation and when specifying the Virtualization Deployment Units (VDUs), VNF configurations can have additional components that are managed by VMware Telco Cloud Automation. These additional components include:

- Internal Connection Points: Depending on the Telco Cloud configuration, internal connection points are created as isolated OrgVDC networks within Cloud Director.
- External Connection Points: The External connection points allow a VNF to connect to an existing external network within Cloud Director and server as the traffic ingress interface.
- Additional OVF properties
- Scaling Policies (Scaling Aspects and Instantation levels): Scaling policies in ETSI control the scaling of VDUs. Scaling can be defined in different steps. When using Aspects, more control over the scaling of individual elements of the NF can be achieved..

Table 4-12. Recommendations for Cloud Director Integration

Design Recommendation	Design Justification	Design Implication
Create a highly available RabbitMQ deployment.	Required for integration of Cloud Director and Telco Cloud Automation	Requires additional resources and expertise to deploy and configure RabbitMQ
Deploy the RabbitMQ Managment plug-in.	Allows UI access to RabbitMQ for troubleshooting	None
Deploy a TCA-CP node for each unique Cloud Director deployment.	Allows TCA to deploy VNFs to Cloud Director OrgvDCs	
Use tagging and RBAC for each VI and compute profile.	Enables extension of the Cloud Directory tenancy constructs into TCA RBAC.	Requires appropriate RBAC planning in TCA

VMware Integrated OpenStack

VMware Integrated Openstack (VIO) leverages the benefits of the vSphere Software Defined Data Center with the capabilities, API features, and components offered by OpenStack. The Cloud Automation design includes VMware OpenStack services and components that allow the consumption of the software-defined storage, networking, and compute.

Nova Compute Design

Nova provisions virtual compute instances on top of the SDDC infrastructure. Nova comprises a set of daemons running as Kubernetes Pods on top of the Tanzu Kubernetes cluster to provide the compute provisioning service.

Nova compute consists of the following daemon processes:

- **nova-api**: Accepts and responds to end-user compute API requests such as VM boot, reset, resize, and so on.
- **nova-compute**: Creates and terminates VM instances.
- **nova-scheduler**: Takes a VM instance request from the queue and determines where compute must run it.
- **nova-conductor**: Handles requests that need coordination and acts as a database proxy. Nova conductor communicates between Nova processes.

Nova Compute

Unlike a traditional KVM-based approach where each hypervisor is represented as a nova compute, the VMware vCenter driver activates the nova-compute service to communicate with a VMware vCenter Server instance.

The vCenter driver aggregates all ESXi hosts within each cluster and presents one large hypervisor to the nova scheduler. VIO deploys a nova-compute Pod for each vSphere ESXi cluster that it manages. Because individual ESXi hosts are not exposed to the nova scheduler, Nova scheduler assigns hypervisor compute hosts at granularity of the vSphere clusters. vCenter selects the ESXi host within the cluster based on the advanced DRS placement settings. Both automated and partially automated DRS are supported for standard VM workloads. DRS must be deactivated in the case of SR-IOV.

Nova Host Aggregates

A nova host aggregate is a grouping of hypervisors or nova computes. Groupings can be done based on the host hardware similarity. For example, clusters with SSD storage backing can be grouped into one aggregate and clusters with magnetic storage into another aggregate. If hardware attributes are similar, grouping can also be based on the physical location of the cluster in the form of availability zones. If there are N data centers, all ESXi clusters within a data center can be grouped into a single aggregate. An ESXi cluster can be in more than one host aggregate.

Host Aggregates provide a mechanism to allow administrators to assign key-value pairs, also called metadata, to compute groups. The nova scheduler can use this key-value pair and metadata to select the hardware that matches the client request. Host aggregates are visible only to administrators. Users consume aggregates based on the VM flavor definition and availability zone.

Nova Scheduler

VMware Integrated OpenStack uses the nova-scheduler service to determine where to place a new workload or a modification to an existing workload request, for example, during a live migration or when a new VM starts up. A nova-scheduler is simply a filter. Based on the type of request, it eliminates nova-computes that cannot achieve the workload request and returns those that can.

Nova scheduler also controls host CPU, memory, and disk over-subscription. Over-subscription places multiple devices to the same physical resource to optimize usage. Over-subscription can be defined based on the host aggregate. The following filters when activated controls aggregates-level over-subscription management:

- **AggregateCoreFilter**: Filters hosts by CPU core numbers with a per-aggregate cpu_allocation_ratio value.
- **AggregateDiskFilter**: Filters hosts by disk allocation with a per-aggregate disk_allocation_ratio value.
- **AggregateRamFilter**: Filters hosts by RAM allocation of instances with a per-aggregate ram_allocation_ratio value.

Note If the per-aggregate value is not found, the value falls back to the global setting. If the host is in more than one aggregate and thus more than one value is found, the minimum value is used.

Instead of over-subscription, Cloud Administrators can assign dedicated compute hosts by OpenStack tenants. You can use the `AggregateMultiTenancyIsolation` filter to control VM placement based on the OpenStack tenant. In this context, Tenant is defined as an OpenStack project. If an aggregate has the `filter_tenant_id metadata` key, the hosts in the aggregate create instances only from that tenant or list of tenants. No other tenant is allowed on these hosts.

As stated in the **Nova Compute** section, individual ESXi hosts are not exposed to the nova scheduler. Therefore, ensure that the nova schedule filters align with the underlying vSphere resource allocation.

Nova Compute Scaling

As workloads increase, the cluster must be scaled to meet new capacity demands. While vCenter Server can have a maximum cluster size of 64 ESXi hosts, VIO cluster scaling varies depending on the use case. You can add new capacity to a VIO deployment in two ways:

- **Vertical scaling**: Increase the number of hosts in a cluster.

- **Horizontal scaling**: Deploy a new vCenter Server cluster and add the lcuster as a new nova-compute Pod to an existing or new nova compute aggregate.

The implementation of vertical or horizontal scaling must be based on the use case and the number of concurrent operations against the OpenStack API. The following table outlines the most frequently-used deployment scenarios:

Use Case	Expected Parallel OpenStack Operations	Scaling Model
Traditional Enterprise	Low	Horizontal or Vertical
Direct API access to the infrastructure **Example**: CICD workflow	High	Horizontal
Direct API access to the infrastructure **Example**: Terraform Automation workflow	Low	Horizontal or Vertical
NFV deployment	Low	Horizontal or Vertical
Cloud Native workload running on top of Kubernetes	Low	Horizontal or Vertical

All VIO deployments must leverage capacity trend analysis of Aria Operations and Aria Operations for Logs. To identify the expansion strategy for your deployment, consult the VMware Professional Services Organization (PSO).

Nova Flavours and Compute Performance Tuning

To consume a nova host aggregate, cloud admins must create and expose VM offering so that users can request VMs that match their application vCPU, memory, and disk requirements. In OpenStack, a flavor represents various types of VM offerings.

- An OpenStack flavor defines the compute, memory, and storage capacity of the computing instances.
- In addition to capacity, a flavor such as SSD, spinning disks, CPU types, CPU family, and so on can also indicate the hardware profile. Hardware profiles are often implemented through flavor extra-specs.
- Nova flavor extra-specs are key-value pairs that define which compute or host aggregate a flavor can run on. Based on extra-spec, the nova-scheduler locates the hardware that matches the corresponding key-value pairs on the compute node.

Data-plane intensive workloads require VM-level parameters for maximum performance. VM-level parameters can also be handled using nova flavors. The vCloud NFV Performance Tuning Guide outlines the recommended VM-level parameters to be set when deploying data-plane intensive workloads:

- Virtual CPU Pinning
- NUMA alignment

- CPU/Memory reservation setting
- Selective vCPU Pinning
- Tenant VDC
- Huge Page
- Passthrough Networking

VIO supports the following VM-level parameters for VNF performance tuning:

Table 4-13. VNF Tuning Parameters

VM Parameter	Flavor Metadata Category	Metadata Values
CPU Pinning	CPU Pinning policy	hw:cpu_policy=dedicated
	VMware Policies	vmware:latency_sensitivity_level=high.
	VMware Quota	quota:cpu_reservation_percent and quota:memory_reservation_percent=100
Selective vCPU Pinning	Custom	vmware:latency_sensitivity_per_cpu_high="<cpu-id1>,<cpu-id2>"
	CPU Pinning policy	hw:cpu_policy=dedicated
	VMware Quota	quota:cpu_reservation_percent and quota:memory_reservation_percent=100
	VMware Policies	vmware:latency_sensitivity_level=high.
NUMA	VMware Policies	numa.nodeAffinity="numa id"
Huge Pages	Guest Memory Backing	hw:mem_page_size="size"
	VMware Quota	quota:memory_reservation_percent=100
Tenant vDC	VMware Policies	vmware:tenant_vdc=UUID
CPU Memory Reservation	VMware Quota	For more details, see Supported Flavor Extra Spec.

Table 4-14. Nova Compute Design Recommendations

Design Recommendation	Design Justification	Design Implication
Create nova host aggregates to group vSphere clusters sharing similar characteristics.	Host Aggregates provide a mechanism to allow administrators to group compute clusters.	None
Assign key-value pairs to host aggregates based on the hardware profile and data center affinity.	The nova scheduler uses the key-value pair and metadata to select the hardware that matches the client request.	None

Table 4-14. Nova Compute Design Recommendations (continued)

Design Recommendation	Design Justification	Design Implication
Use AggregateFilter to control nova compute (vSphere cluster) over-subscription.	Over-subscription leads to greater resource utilization efficiency.	You must ensure that the over-subscription ratio does not conflict with the Tenant VDC reservation. Over-subscription can lead to degraded SLA.
Use AggregateMultiTenancyIsolation filter to place VMs based on the tenant ID.	Used when an entire cluster must be reserved for a specific OpenStack Tenant.	Unbalanced resource consumption
When using a tenant VDC, create a new default VDC with no reservation and map all default flavors to the new VDC.	From a vSphere resource hierarchy perspective, VMs created using default flavors belong to the same resource hierarchy as Tenant VDC. If Tenant VDCs and VMs share the parent resource pool, it is not guaranteed that VMs with no resource reservation will not use CPU share from Tenant VDC with full reservation. Mapping default VIO flavors to a child VDC without resource resolution alleviates this issue.	None
Use OpenStack metadata extra specs to set VM-level parameters for data plane workloads in the compute flavor.	Metadata extra specs translate to VM settings on vSphere.	None
When adding more ESXi resources, consider building new vSphere clusters instead of adding to the existing cluster in an environment with a large churn.	New vSphere clusters introduce more parallelism when supporting a large number of concurrent API requests.	New clusters introduce new objects to manage in the OpenStack database.

Neutron Networking Design

VMware Integrated OpenStack networking consists of Neutron services integrated with NSX to build rich networking topologies and configure advanced network policies in the cloud.

A Telco VNF deployment requires the following networking services:

- **L2 services**: Activates tenants to create and consume the L2 networks.
- **L2 trunk**: Activates tenants to create and consume L2 trunk interfaces
- **L3 services**: Activates tenants to create and consume their IP subnets and routers. These routers can connect intra-tenant application tiers and can also connect to the external domain through NATed and non-NATed topologies.

- **Floating IPs**: A DNAT rule that maps a routable IP on the external side of the router (External network) to a private IP on the internal side (Tenant network). This floating IP forwards all ports and protocols to the corresponding private IP of the instance (VM) and is typically used in cases where there is IP overlap in tenant space.
- **DHCP Services**: Activates tenants to create their DHCP address scopes.
- **Security Groups**: The ability for tenants to create their firewall policies (L3 or L4) and apply them directly to an instance or a group of instances.
- **Load-Balancing-as-a-Service** (LBaaS): Activates tenants to create their load-balancing rules, virtual IPs, and load-balancing pools.
- **High throughput data forwarding**: Activates the configuration of passthrough or DPDK interfaces for data plane intensive VNFs.
- **Data Plane Separation**: Separates the traffic between different Tenants at L2 and L3.

Object Mapping in VMware Integrated OpenStack NSX Neutron Plug-in

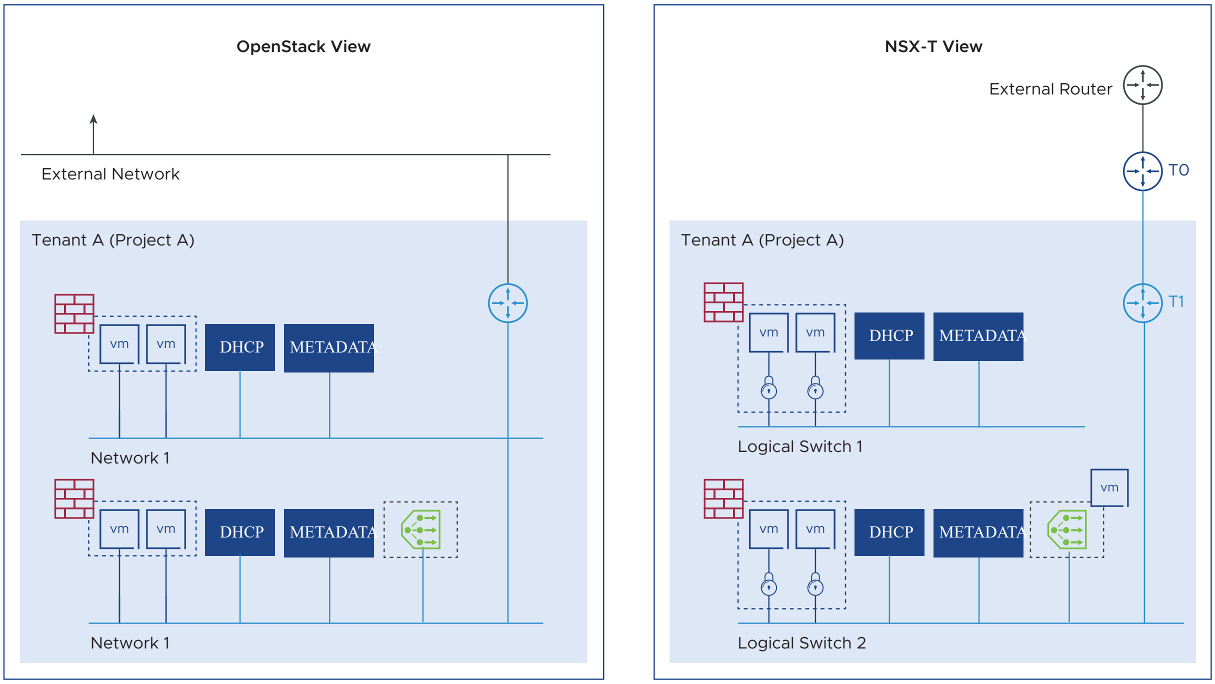
The NSX Neutron plug-in is an open-source project and can be used with any OpenStack implementation. VMware Integrated OpenStack natively integrates the NSX neutron plug-in out-of-the-box. VMware Integrated OpenStack neutron services leverage NSX neutron plug-in to invoke API calls to NSX Manager, which is the API provider and management plane of NSX.

The following table summarizes the mapping between the standard Neutron objects and VMware NSX Neutron plug-in.

Neutron Object	Mapping in NSX-T Neutron Plugin
Provider Network	VLAN
Tenant Network	Logical switch
Metadata Services	Edge Node Service
DHCP Server	Edge Node Service
Tenant Router	Tier 1 router
External Network	Tier 0 router (add VRF)
Floating IP	NSX-T Tier-1 NAT
Trunk Port	NSX-T Guest VLAN tagging
LBaaS	NSX-T Load Balancer through Tier-1 router
FWaaS/Security Group	NSX-T Microsegmentation / Security Group

The following figure illustrates the realization of VIO Objects with the NSX fabric:

Figure 4-20. Summary Toplogy View



Supported Network Topologies

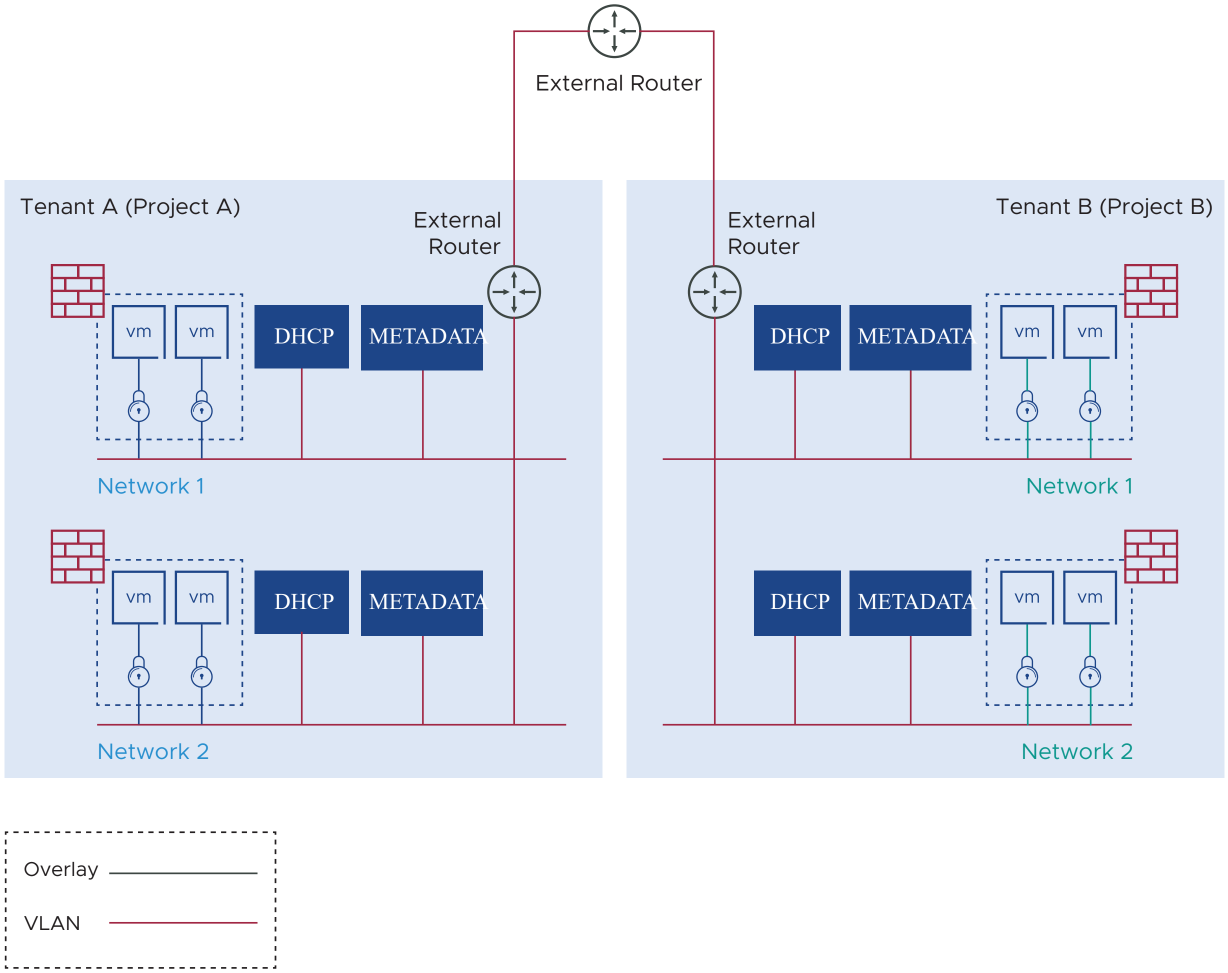
The common topologies for deploying Telco VNF deployments are as follows:

Use Case	L2	L3	OpenStack Network Security Policy
VLAN-backed Network	VLAN	Physical-Router	Security Group Only
Overlay-Backed Network with NSX L3 Services and NAT	Overlay	NSX Edge	FWaaS and Security Group
Overlay-Backed Network with NSX L3 Services and No-NAT	Overlay	NSX-Edge	FWaaS and Security Group

VLAN-Backed Networks

VLAN-backed network is often known as an OpenStack Provider network. VNF booted in the provider network gets its DHCP lease and metadata relay information from the NSX but relies on the physical network infrastructure to provide default gateway or first-hop routing services. Because the physical infrastructure handles routing, only security groups are supported in the provider network. LBaaS, FWaaS, NAT, and so on are implemented in the physical tier, outside of VIO.

Figure 4-21. VLAN-Backed Network



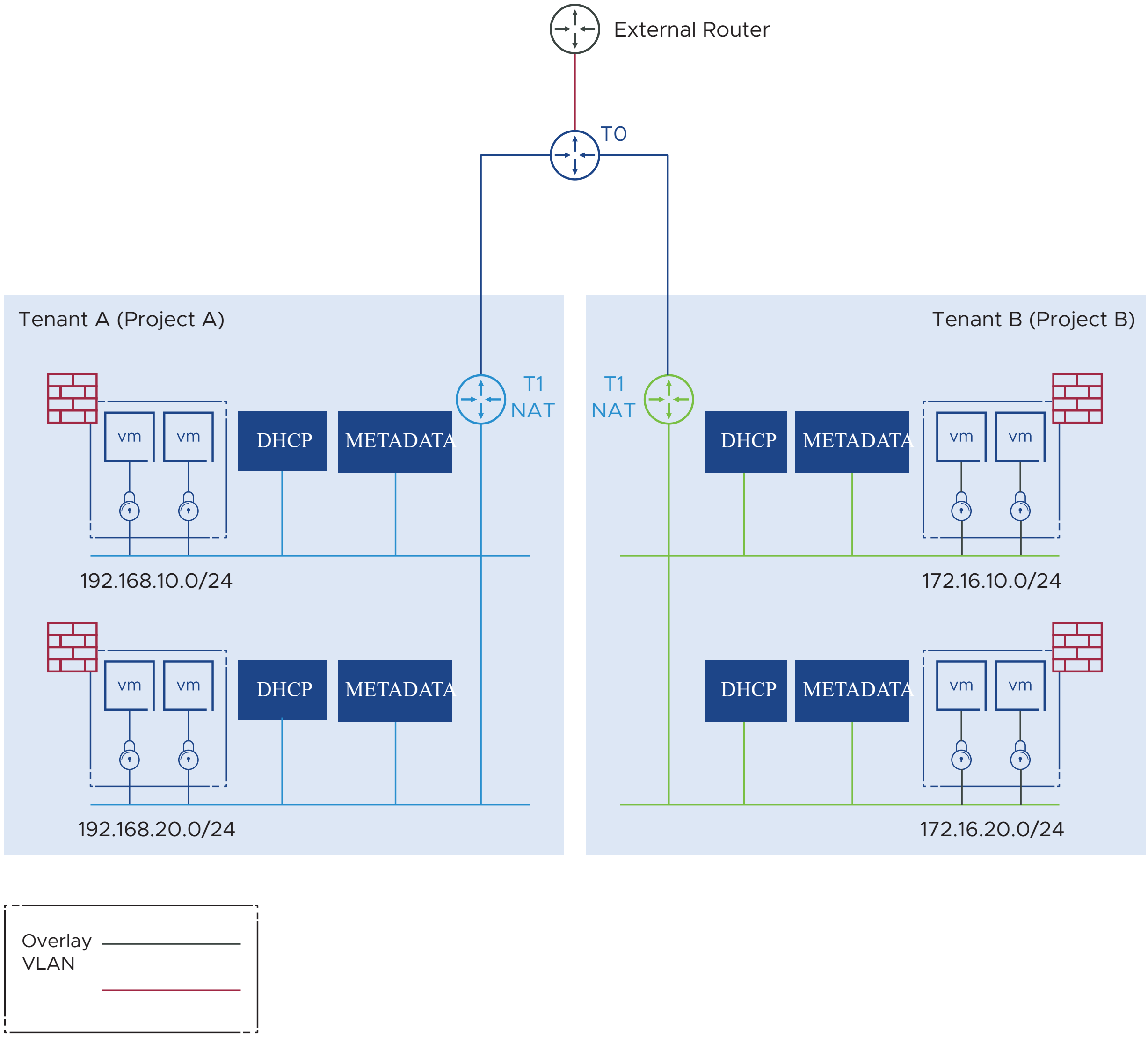
NAT Topology

The NAT topology activates tenants' ability to create and consume private RFC 1918 IP subnets and routers. The NAT topology supports overlapping IP subnets between tenants. Inter-tenant and connections to the external domain must source NAT by the NSX tier gateway backing the OpenStack tenant router.

The NSX Tier-0 router maps to the OpenStack external network and can be shared between tenants or dedicated to a single tenant.

A dedicated DHCP server is created in the NSX backend for each tenant subnet and segment. Neutron services such as LBaaS, FWaaS, and Floating IPs are fully supported.

Figure 4-22. NAT Topology NSX View



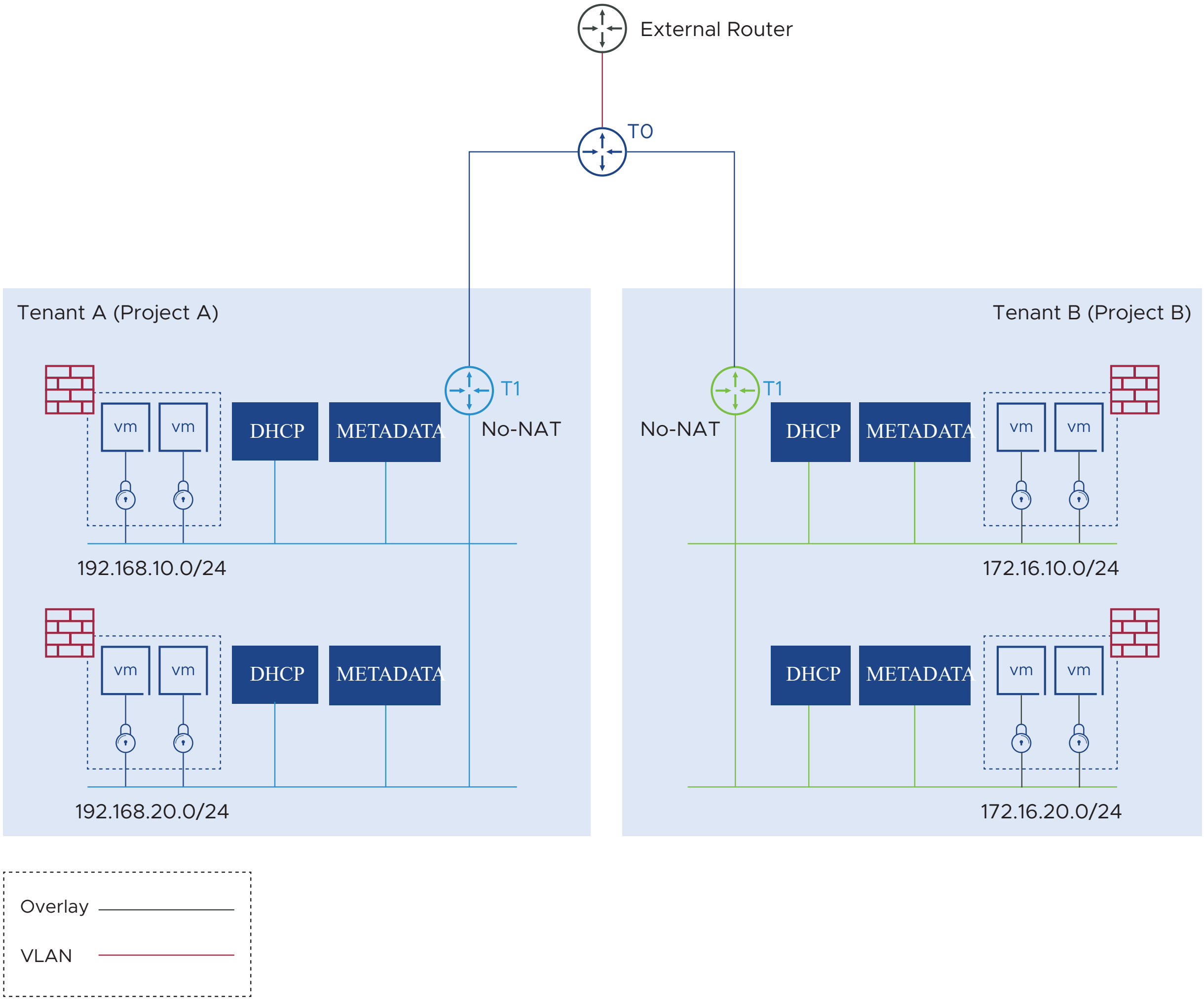
No-NAT Topology

In the No-NAT topology, NSX- performs L2 switching, security segmentation, L3 routing, load balancing, and other tasks. The source IP addresses of the VMs are preserved for inter-tenant communication and for communication to the external domain. IP subnets cannot overlap as there is no NAT.

Similar to the NAT topology, the NSX Tier-0 router maps to the OpenStack external network and can be shared between tenants or dedicated to a single tenant.

A dedicated DHCP server is created in the NSX backend for each tenant subnet and segment. Neutron services such as LBaaS, FWaaS, and Floating IPs are fully supported.

Figure 4-23. No-NAT Topology



Passthrough Networking

VMware Integrated OpenStack supports SR-IOV through Passthrough Networking. You can configure a port to allow SR-IOV passthrough and then create OpenStack instances that use physical network adapters. Starting with VMware Integrated OpenStack 7.0, multiple ports with different SR-IOV physical NICs are also supported. Multiple SR-IOVs provide vNIC redundancy for a VM, ensuring that the SR-IOV ports are assigned by different physical NICs in the ESXi server host. The Multiple SRIOV Redundancy setting is defined as part of the nova flavor definition.

Passthrough networking can be used for data-intensive traffic but it cannot use virtualization benefits such as vMotion, DRS, and so on. Hence, VNFs employing SR-IOV become static hosts. A special host aggregate can be configured for such workloads.

Port security is not supported for passthrough-configured ports and is automatically deactivated for the port created.

Data Plane Isolation with VIO and NSX

VIO uses an NSX Tier-0 router to represent OpenStack External Networks. An External network can be shared across all OpenStack tenants or dedicated to a tenant through the neutron RBAC policy.

When associating an OpenStack tenant router with an external network, the NSX Neutron plugin-in attaches the OpenStack tenant router (Tier-1) with the associated external network (Tier-0).

If you use the VRF Lite feature in NSX, each VRF instance can be defined as a unique OpenStack External Network. Through the Neutron RBAC policy, the external network (VRF instance) can be mapped to specific tenants. VRF maintains dedicated routing instances for each OpenStack tenant and offers data path isolation required for VNFs in a multi-tenant deployment.

Network Security

Security is enforced at multiple layers with VMware Integrated OpenStack:

- Tenant Security Group: Tenant Security group is defined by Tenants to control what traffic can access the Tenant application. It has the following characteristics:
  - Every rule has an ALLOW action, followed by an implicit deny rule at the end.
  - One or more Security Groups can be associated with an OpenStack Instance.
  - Tenant Security Groups have lower precedence over Provider Security Groups.
- Provider Security Group: Provider security groups are defined and managed only by OpenStack Administrators and are primarily used to enforce compliance. For example, blocking specific destinations and enforcing protocol compliance used by a tenant application. The Provider security group has the following characteristics:
  - Every rule has a DENY action.
  - One or more Provider Security Groups can be associated with an OpenStack Tenant.
  - Provider Security Groups are with higher precedence over Tenant Security Groups.
  - Provider Security Groups are automatically attached to new tenant Instances.
- Firewall-as-a-Service: FWaaSv2 is a perimeter firewall controlled by Tenants to control security policy at a project level. With FWaaSv2, a firewall group is applied at the router port level to define ingress/egress policies but not at the router level (all ports on a router).
- Port Security: Neutron Port security prevents IP spoofing. Port Security is activated by default with VIO. Depending on the application, port security settings might need to be activated or deactivated.
  - Scenario I: First hop routing protocol such as VRRP relies on Virtual IP (VIP) address to send and receive traffic. Port security policy must be updated so that the traffic to and from the VIP address can be allowed.
  - Scenario II: Port Security policy must be deactivated to support virtual routers.

Octavia Load-Balancer Support

The Octavia LBaaS workflow includes the following capabilities:

- Create a Load Balancer.
- Create a Listener.
- Create a Pool.
- Create Pool Members.
- Create a Health Monitor.
- Configure Security Groups to allow the Health Monitor to work from Load Balancer to Pool Members.

Octavia Load Balancer requires the backing of an OpenStack Tenant router. This tenant router must be attached to an OpenStack External Network. The following types of VIPs are supported:

- **VIP using External network subnet pool**: VIP allocated from an external network is designed to be accessed from public networks. To define a VIP that is publicly accessible, you must do one of the following:
  - Create a load balancer on an external network
  - Create a load balancer on a tenant network and assign floating IP to the VIP port.

  Server pool members can belong to the different OpenStack tenant subnets. The Load balancer admin must ensure reachability between the VIP network and server pool members.
- **VIP using Internal Network subnet pool**: Load balancer VIP is allocated from the tenant subnet you selected when creating the load balancer, typically a private IP. Server pool members can belong to different OpenStack subnets. The load balancer admin must ensure reachability between the VIP network and server pool members.

VIO Neutron Design Recommendations

Design Recommendation	Design Justification	Design Implication
When deploying OpenStack provider networks, do not deploy an additional DHCP server on the VLAN backing the provider network.	DHCP instance is created automatically in the NSX when the Provider network subnet is created.	None
When running with No-NAT on the OpenStack Tenant router, NSX-T0 must be pre-configured to advertise (redistribute) NAT IP blocks either as /32 or the summary route to the physical router.	T0 route advertisement updates physical L3 device with Tenant IP reachability information.	None
When running with No-NAT on the OpenStack Tenant router, NSX-T0 must be pre-configured to advertise connected routes.	T0 route advertisement updates physical L3 device with Tenant IP reachability information.	None

Design Recommendation	Design Justification	Design Implication
When deploying FWaaS, ensure that the Tenant quota allows for at least one OpenStack Tenant router.	FWaaS requires the presence of a Tenant router.	None
When deploying LBaaS, ensure that the Tenant quota allows for at least one OpenStack Tenant router.	Neutron NSX-T plugin requires a Logical router with the external network attached.	None
When SR-IOV is required for VNF workloads, use Multiple SR-IOV to provide vNIC redundancy to VNFs.	SR-IOV NIC redundancy protects VNF from link-level failure.	VNFs must be configured to switch over between vNICs in case of link failure.
Place all SR-IOV workloads in a dedicated ESXi cluster. Deactivate DRS and vMotion for this cluster.	SR-IOV cannot benefit from virtualization features such as vMotion and DRS.	Resource consumption might be imbalanced in the SR-IOV cluster as DRS is not available to rebalance workloads.
Create a dedicated Neutron External Networks backed by VRF per tenant.	Data path isolation and separate routing instances per tenant.	VRFs must be created manually before VMware Integrated OpenStack can use them.
Use allowed_address_pairs to allow traffic to and from VNF that implements the first hop routing protocol such as VRRP.	Default Neutron Port security permits traffic to and from the IP address assigned to vNIC only, not VIP.	None
Deactivate Port Security on attachments that connect to virtual routing appliances.	Default Neutron Port security permits traffic to and from the IP address assigned to vNIC only, not Transit traffic.	None

Cinder Design

Volumes are block storage devices that you attach to instances to activate persistent storage.

Block storage in VMware Integrated OpenStack is provided by Cinder. The Cinder services (cinder-api, cinder-volumes, cinder-scheduler) run as Pods in the VIO control plane Kubernetes cluster. The Cinder services functionality is not modified from the OpenStack mainline and a vSphere-specific driver is developed to support vSphere-centric storage platforms.

Shadow VM and First Class Disks (FCD)

In OpenStack, Cinder services rely on first-class block storage. However, before vSphere 6.5, the vCenter object model and API do not support block storage to native Cinder usage in OpenStack. The required functionality is provided in VIO through Shadow VMs. Shadow VM works by mimicking a first-class block storage object and is created using the following steps:

- When provisioning Cinder volume, the vSphere Cinder driver creates a new Cinder volume object in the OpenStack database.
- When attaching the volume to a virtual instance, the Cinder driver requests provisioning of a new virtual instance, called a shadow VM.
- The shadow VM is provisioned with a single VMDK volume whose size matches the requested Cinder volume size.

- After the shadow VM is provisioned successfully, Cinder attaches the shadow VMs VMDK file to the target virtual instance and Cinder treats the VMDK as the Cinder volume.

For each Cinder volume, a shadow VM is created with a VMDK attached. Therefore, many powered-off VMs are created for Cinder volumes.

The First Class Disks (vStorageObject) feature introduced in vSphere 6.7 activates virtual disk-managed objects. FCDs perform life-cycle management of VM disk objects, independent of any VMs. FCDs can create, delete, snapshot/backup, restore, and perform life-cycle tasks on VMDK objects without requiring them to be attached to a VM. The conversion between FCD and existing VMDK volumes based on Shadow VM is done manually. To convert a VMDK volume manually, see Manage a Volume.

Both FCD and Shadow VM VMDK drivers can co-exist in VIO. If FCD is activated, a new cinder-volume backend is created and it co-exists with the default VMDK driver-based cinder-volume backend. The shadow VM-based volumes and FCD-based volumes can be attached to the same Nova instance.

Figure 4-24. Cinder Volume Driver Co-Existence

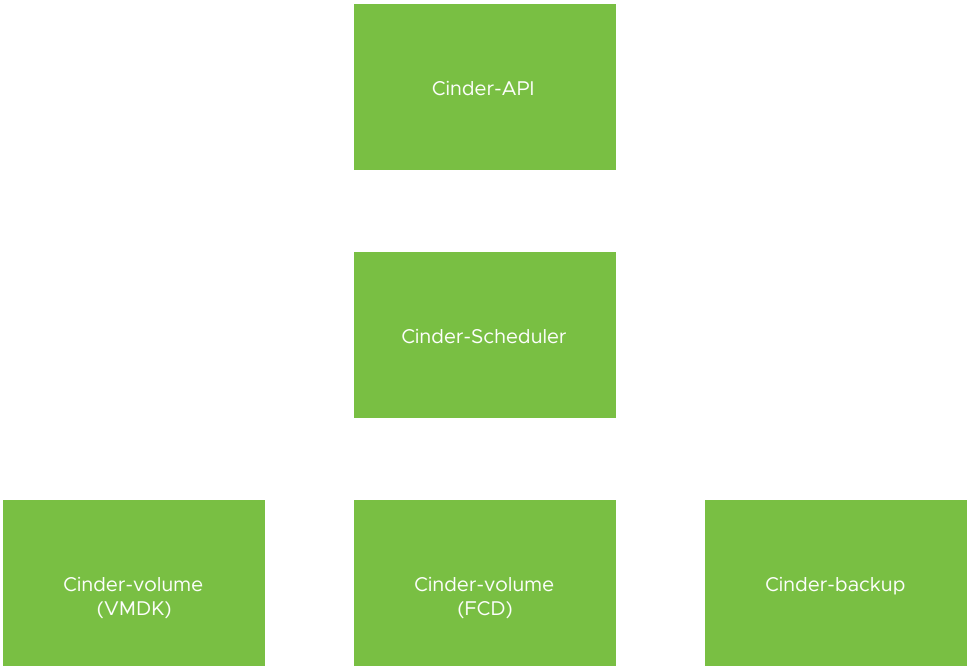


Table 4-15. Supported Volume Operations based on Driver

	FCD Driver	VMDK Driver
Create or Delete Snapshot	Yes	Yes
Copy Image to Volume.	Yes ■ sparse ■ streamOptimized ■ preallocated ■ vSphere template	Yes ■ sparse ■ streamOptimized ■ preallocated ■ vSphere template

Table 4-15. Supported Volume Operations based on Driver (continued)

	FCD Driver	VMDK Driver
Retype	Yes (must be unattached)	Yes (must be unattached)
Attach or De-attach	Yes	Yes
Multi-Attach	No	Yes
Create or restore Volume backup	Yes	Yes
Adapter Type	■ IDE ■ lsiLogic ■ busLogic ■ lsiLogicsas ■ paraVirtual	■ IDE ■ lsiLogic ■ busLogic ■ lsiLogicsas ■ paraVirtual
VMDK Type	■ thin ■ thick ■ eagerZerodThick	■ thin ■ thick ■ eagerZerodThick
Extension	Yes (must be unattached)	Yes (must be unattached)
Transfer	Yes (must be unattached)	Yes (must be unattached)

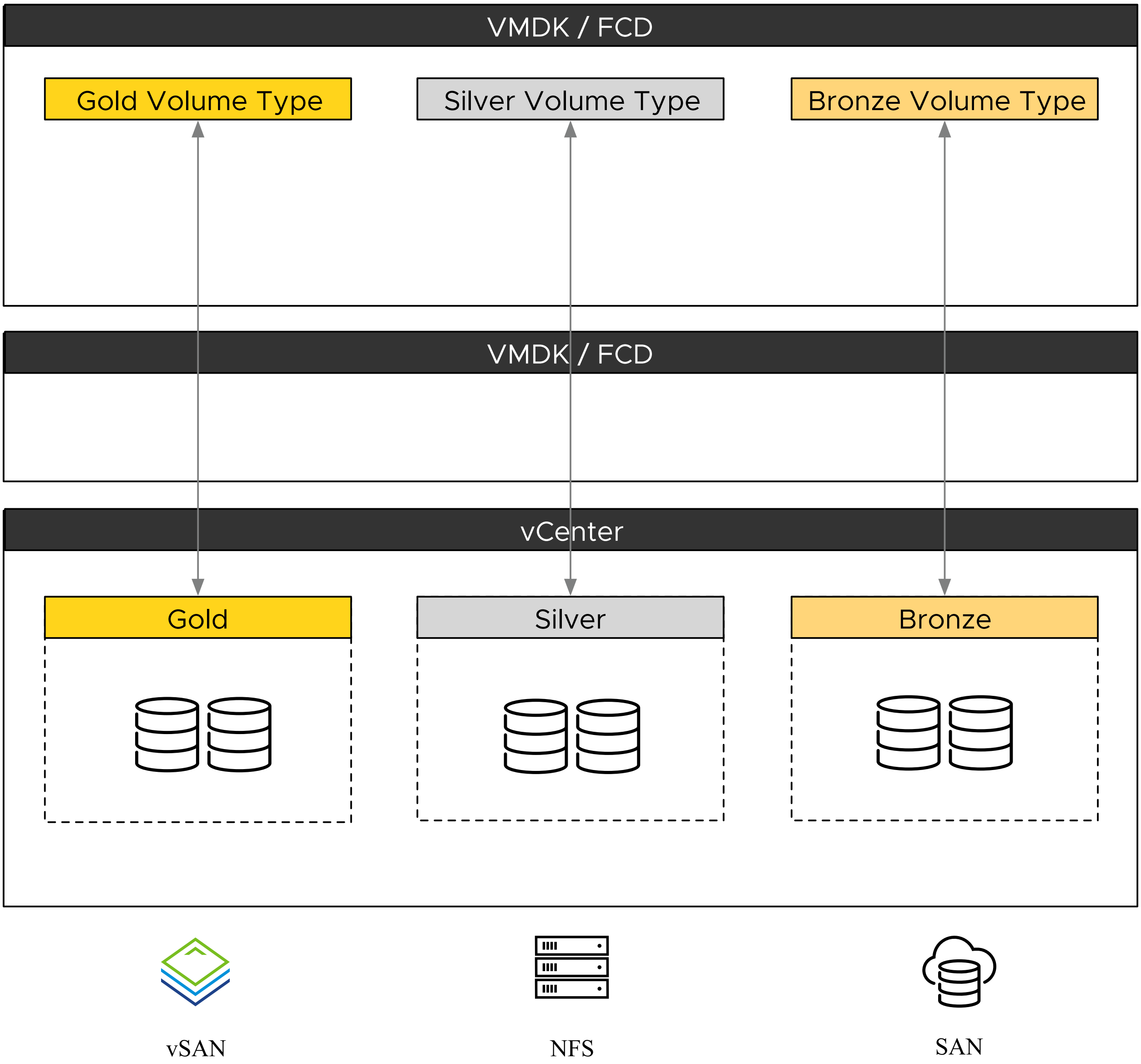
Storage Policy Based Management (SPBM) and Cinder

SPBM (storage polices) is a vCenter Server feature that controls which type of storage is provided for the VM and how the VM is placed within the storage. vSphere offers default storage policies. You can also define policies and assign them to the VMs. In a custom storage policy, you can specify various storage requirements and data services such as caching and replication for virtual disks.

When the SPBM storage policy is integrated with VIO, Tenant users can create, clone, or migrate cinder volumes by selecting the corresponding storage volume type exposed by the Cloud Admin.

The SPBM mechanism assists with placing the VM in a matching datastore and ensures that the cinder virtual disk objects are provisioned and allocated within the storage resource to guarantee the required level of service.

Figure 4-25. SPBM and Cinder Integration



vSphere Cinder Volume

The vSphere Cinder driver supports the following datastore types:

- NFS
- VMFS
- vSAN
- VVOL

The VIO Cinder service uses cinder-scheduler to select a datastore to place a new volumes request. The following are some of the VMDK driver characteristics:

- Storage vMotion and Storage DRS are supported when running VMware Integrated OpenStack. However, there is no integration between cinder-schedulers. A scheduler neither has information nor understands other schedulers.
- vSphere SPBM policies can be applied to provide storage tiering by using metadata.
- Snapshots of cinder volumes are placed on the same datastore as the primary volume.
- Storage over-subscription through the `max_over_subscription_ratio` option or activating thin or thick provisioning support through the `thin_provisioning_support` and `thick_provisioning_support` options do not apply to the Cinder VMDK driver.
- By default, the VMDK driver creates Cinder volumes as thin-provisioned disks. Cloud Administrators can change the default setting using extra specifications. To expose more than one type, administrators can create corresponding disk types with the `vmware:vmdk_type` key set to either **`thin`**, **`thick`**, or **`eagerZeroedThick`**. Disk types can be created from the UI (Horizon under Admin → Volumes), CLI, or API.

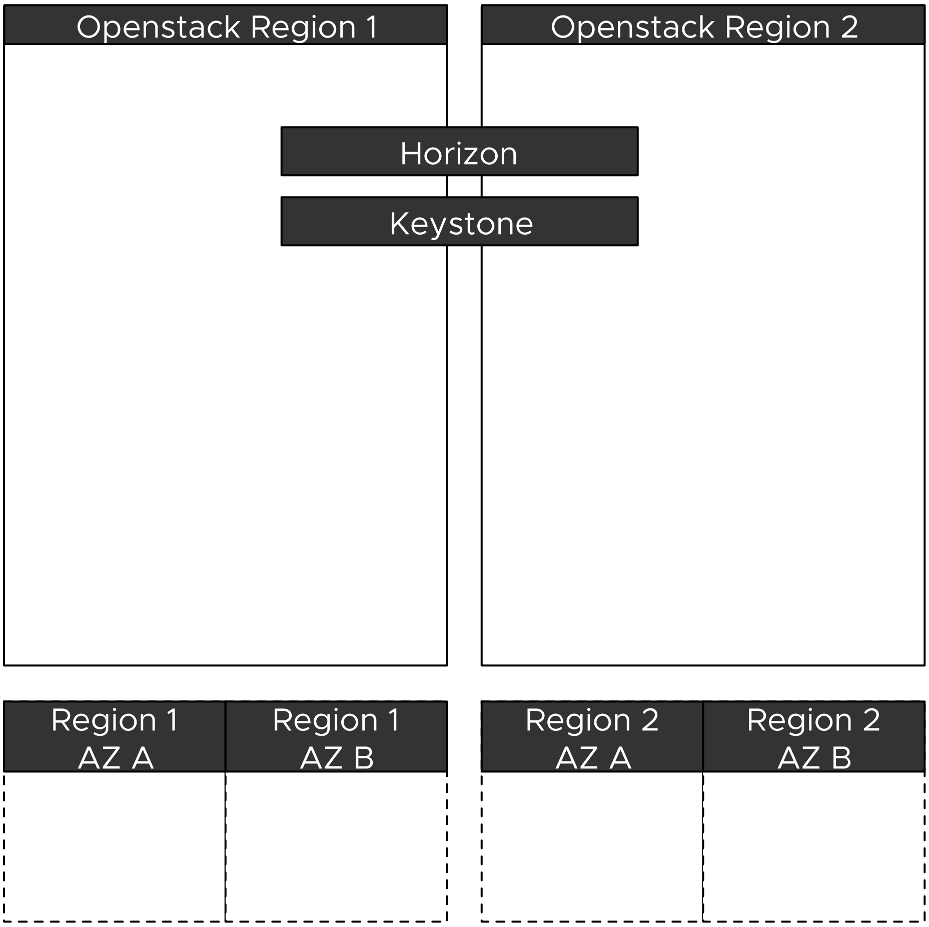
Table 4-16. Volume Types

Type	Metadata
thick_volume	vmware:vmdk_type=thick
thin_volume	vmware:vmdk_type=thin
eagerZeroedThick	vmware:vmdk_type=eagerZeroedThick
QoS	Based on SPBM policy

Availability Zone and Region Design

An OpenStack Cloud is categorized into Regions, Availability Zones, and Host Aggregates. This section describes the high-level design considerations for each category.

Figure 4-26. VIO Region and Availability Zones



OpenStack Region

A Region in VIO is a full OpenStack deployment, including a dedicated VIO control plane, API endpoints, networks, and compute SDDC stack. Unlike other OpenStack services, Keystone and Horizon services can be shared across Regions. Shared Keystone service allows for unified and consistent user onboarding and management. Similarly, for Horizon, OpenStack users and administrators can access different regions through a single pane of glass view.

OpenStack regions can be geographically diverse. Because each region has independent control and data planes, the failure in one region does not impact the operational state of another. Application deployment across regions is an effective mechanism for geo-level application redundancy. With a geo-aware load balancer, application owners can control application placement and failover across geo regions. Instead of active/standby data centers with higher-level automation (Heat templates, Terraform, or third-party CMP), applications can dynamically adjust across regions to meet SLA requirements

Keystone Design

Keystone federation is required to manage Keystone services across different regions. VIO supports two configuration models for federated identity. The most common configuration is with keystone as a Service Provider (SP), using an external Identity Provider such as the vIDM as the identity source and authentication method. The two leading protocols for external identity providers are OIDC and SAML2.0.

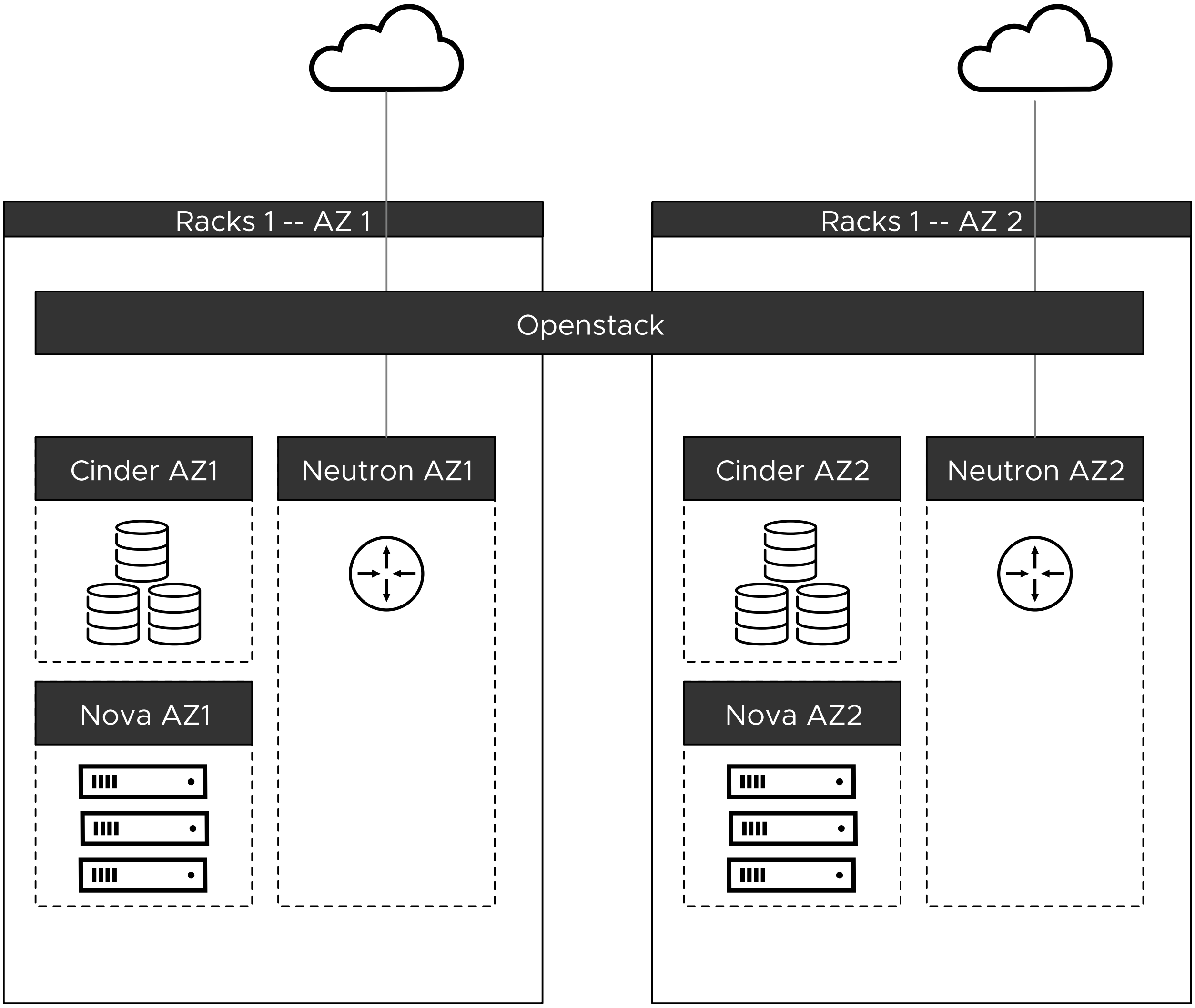
The second type of integration is “Keystone to Keystone”, where multiple keystones are linked with one acting as the identity source for all regions. The central Keystone instance integrates with the required user backends such as AD for user onboarding and authentication.

If an identity provider is not available to integrate with VIO, cloud administrators can use an LDAP-based integration.

OpenStack Availability Zone

Nova compute aggregate in OpenStack is a grouping of resources based on common characteristics such as hardware, software, or even location profile. Nova Availability Zone (AZ) is a special form of Nova-compute aggregate that is associated with availability. Unlike traditional compute aggregates, a single compute cluster can only be a single Nova AZ member. Cloud administrators can create AZs based on availability characteristics such as power, cooling, rack location, and so on. In case of Multi-VC, AZ can be created based on the vCenter Server instance. To place a VNF in an AZ, OpenStack users specify the AZ preference during VNF instantiation. The nova-scheduler determines the best match for the user request within the specified AZ.

Figure 4-27. OpenStack Availability Zone



Cinder Availability Zone

Availability zone is a form of resource partition and placement, so it applies to Cinder and Neutron also. Similar to nova, Cinder zones can be grouped based on location, storage backend type, power, and network layout. In addition to placement logic, cinder volumes must attach to a VNF instance in most Telco use cases. Depending on the storage backend topology, an OpenStack admin must decide if the cross AZ Cinder volume attachment must be allowed. If the storage backend does not expand across VZ, the cross AZ attach policy must be set to false.

Neutron Availability Zone

With NSX, Transport zones dictate which hosts and VMs can participate in the use of a particular network. A transport zone does this by limiting which hosts and VMs that can map to a logical switch. A transport zone can span one or more host clusters.

An NSX Data Center environment can contain one or more transport zones based on your requirements. The overlay transport zone is used by both host transport nodes and NSX Edges. The VLAN transport zone is used by the NSX Edge for its VLAN uplinks. A host can belong to multiple transport zones. A logical switch can belong to only one transport zone.

From an OpenStack Neutron perspective, you can create additional Neutron availability zones with NSX by creating a different overlay transport zone, VLAN transport zone, DHCP, and Metadata proxy Server for each availability zone. Neutron availability zones can share an edge cluster or use separate edge clusters.

In a single-edge cluster scenario, External networks along with associated floating IP pools can be shared across AZ. In a multiple edge clusters scenario, create non-overlap subnet pools and assign a unique subnet per external network.

VIO Availability Design Recommendations

Design Recommendation	Design Justification	Design Implication
When using Region as a unit of failure domain, ensure that each region maps to an independent set of SDDC infrastructure.	Regions are distinct API entry points. Failure in one region must not impact the availability of another region.	None
When using Region as a unit of failure domain, design your application deployment automation such that it is region aware and can redirect API requests across the region.	Region-aware application deployment is more cloud native and reduces the need for legacy DR type of backup or restore.	More complexity in infrastructure and deployment automation.
When using the vCenter Server instance as a unit of Availability Zone (Multi-VC), do not allow the cross AZ cinder attachment.	With Multi-VC, all hardware resources must be available locally to the newly created AZ. **Note** The scenario where the resources in AZ1 are leveraged to support AZ2 is not valid.	None
When using neutron AZ as a unit of failure domain, map each Neutron AZ to a separate Edge cluster.	Different edge clusters ensure that at least one NSX Edge is always available based on recommendations outlined in the platform design.	A floating IP cannot be moved between VMs across different AZs. Neutron AZ cannot map directly to a VRF instance.

OpenStack Tenancy and Resource Isolation

In VMware Integrated OpenStack, cloud administrators manage permissions through users, groups, and project definitions and datapath isolation across the compute, storage, and networking.

Allowing multiple users to share the VMware SDDC environment while ensuring complete separation is a key feature of VMware Integrated OpenStack. VIO offers ways to share virtual resources between tenants but maintains complete separation where needed.

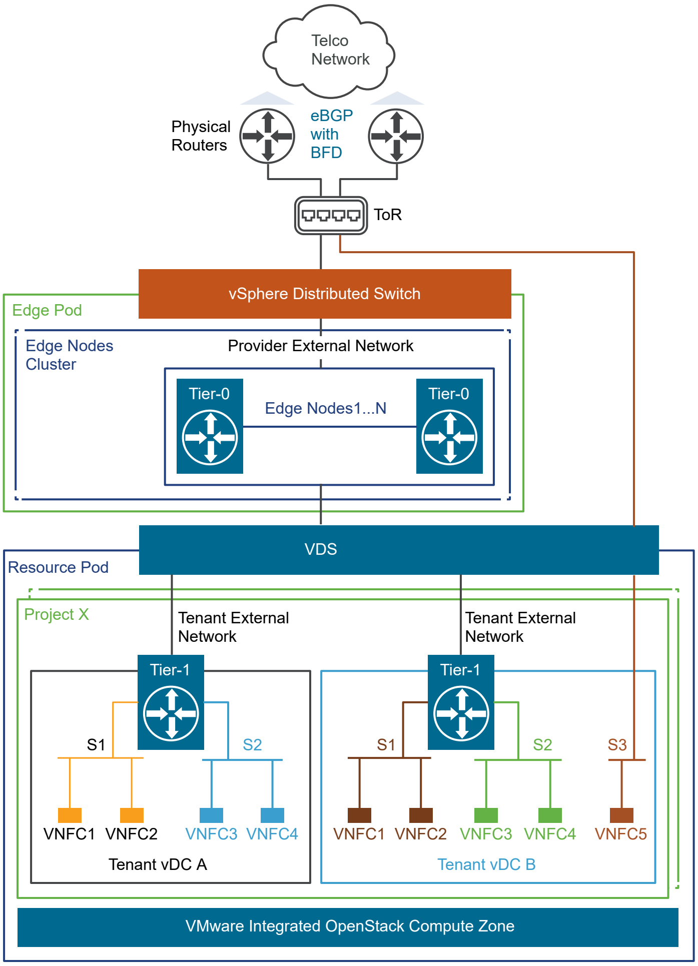
VIO performs tenant isolation in various ways including:

- Keystone API and Dashboard authentication and authorization

- Private images in Glance Image Catalog
- Network traffic isolation between user groups using VRF
- Compute isolation

The following figure illustrates how a fully integrated VMware Integrated OpenStack Compute Zone, Project and Tenant Virtual Data Center (Tenant VDC), NSX segments, and Tier-1 gateways can be leveraged to provide a multitenant environment for deploying VNFs.

Figure 4-28. Private Cloud with Multiple Tenants



This section describes user isolation using Keystone. The subsequent sections describe resource isolation as part of the OpenStack services design.

Identity

The OpenStack Keystone Identity service provides a single point of integration for managing authentication, authorization, and a catalog of services. After a user is authenticated with Keystone, an authorization token is returned by the identity service. An end-user can use the authorization token to access other OpenStack services based on the authorization rules (policy.json). The Identity service user database can be internal or integrated with some external user management systems.

As an alternative to maintaining an independent local keystone user database, VIO provides the capability to reference the enterprise Active Directory (AD) Lightweight Directory Authentication Protocol (LDAP) services for authentication. In addition to LDAP, Federated identity integration with SAML2 or OIDC with external Identity providers are also possible.

OpenStack Projects

Projects in OpenStack are equal to tenants in Telco Cloud Infrastructure. A project is an administrative container where telco workloads are deployed and managed.

An OpenStack project maps to a group of users and groups. OpenStack has a construct called Quotas that allows you to set an upper limit for resources such as vCPUs, memory, number of instances, networks, ports, subnets, and so on. Quotas are configured at the OpenStack Project level. The values used for quotas have no direct reference to the existing infrastructure capacity. Quotas on OpenStack are configured independently of the underlying platform. Quotas are enforced at OpenStack services such as Nova, Neutron, Cinder, and Glance.

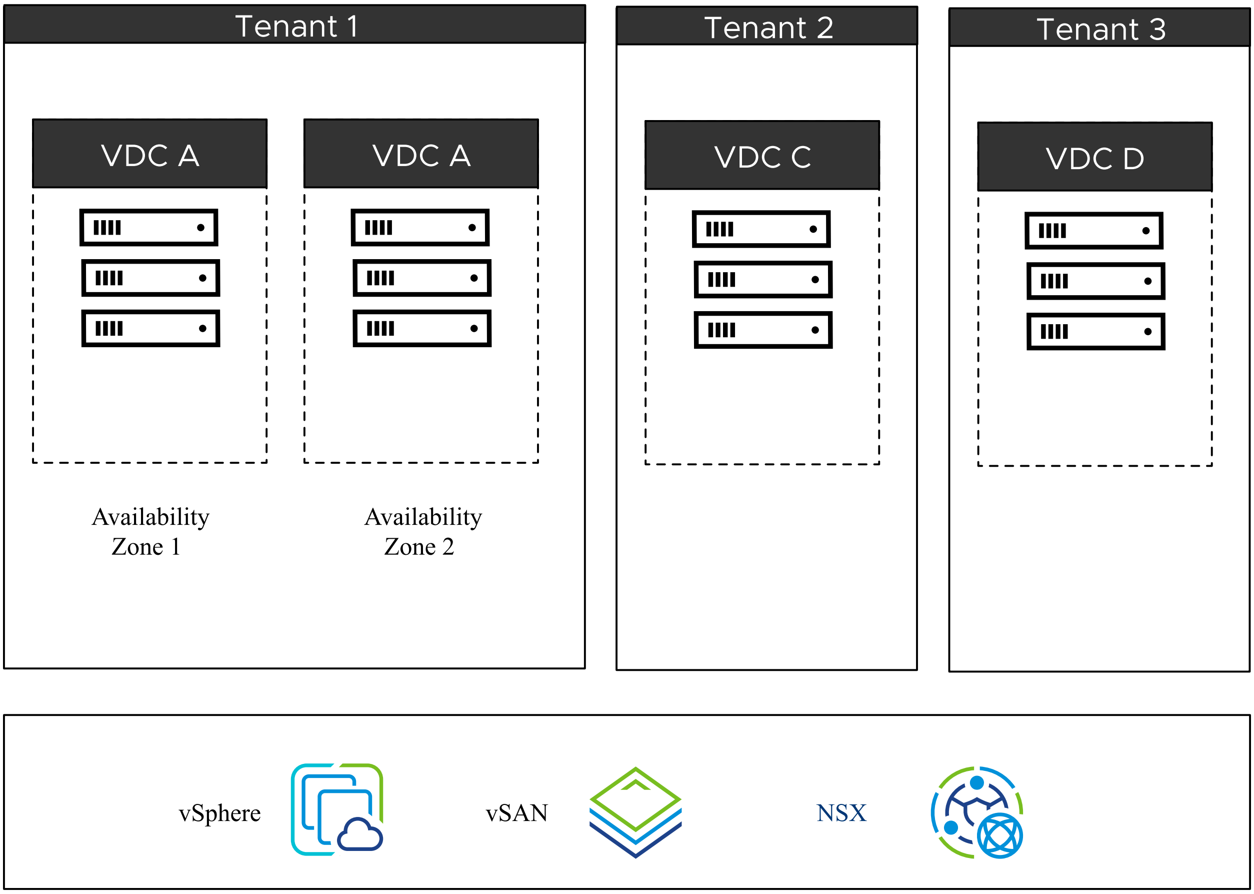
Domains are high-level grouping for projects, users, and groups. Domains can be used as a way to delegate management of OpenStack resources. A user can be a member of multiple domains if an appropriate assignment is granted. Keystone provides a default domain called Default. For most Telco use cases, a single Domain is more than sufficient for the user and group isolation.

Tenant VDC

A Tenant VDC allows the creation of virtual data centers for tenants under different compute nodes that offer specific SLA levels for each telco workload. While quotas on projects set limits on the OpenStack resources, Tenant VDCs allow limits and reservations on a specific class of resources or compute nodes based on the existing capacity and provide resource guarantees for tenants. The vSphere platform enforces limits and guarantees resources allocated to a tenant on a given compute node and completely avoids noisy neighbor scenarios in a multitenant environment. The following three types of policies are supported:

- Pay-as-you-go
- Reservation Pool
- Allocation Pool

Figure 4-29. Tenant VDC for VNF Resource Allocation



Tenancy Design Recommendations

Design Recommendation	Design Justification	Design Implication
Integrate VIO Identity service with an external user database such as AD.	Central user administrator Ability to add or revoke users globally without having to visit every deployment.	Requires a deep understanding of the corporate user directory structure to onboard only required users into VIO.
Use OpenStack domain only in scenarios where the management access has to be delegated to users and groups outside of the traditional Cloud Admin role. Otherwise, use OpenStack projects to provide Tenant isolation.	Multiple Domains complicate user management, and also it is not possible to set usage quota at a domain level.	None

Design Recommendation	Design Justification	Design Implication
Use Tenant VDCs to guarantee resources for tenants and avoid noisy neighbor scenarios	Tenant VDC complements OpenStack project Quota by providing resource guarantee on the virtualized infrastructure. Tenant VDC increases SLA per VNF.	When using Tenant VDC, create a new default VDC with no reservation and map all default flavors to the new VDC.
Avoid updating OpenStack policy.json if there are alternative ways to accomplish the same task.	Snowflake avoidance Upgrade compatibility, while many non-default changes can seem trivial. VMware cannot guarantee upstream implementation will always be backward compatible when moving between releases. Therefore, customers must maintain day-2 changes.	None

VMware Integrated OpenStack Considerations

This section outlines the scaling considerations for the VMware Integrated OpenStack (VIO) components including backup and restore.

VMware Integrated OpenStack Sizing

The sizing options for the OpenStack Control Plane, including the number of instances, sizes, and resources can be scaled horizontally.

The hardware that is required to run VMware Integrated OpenStack depends on the scale of your deployment and the size of the controller that you select.

The VIO manager comes only in a single flavor and requires 4 vCPUs, 16 GB memory, and two disks (40GB/30 GB).

For the production HA deployment, VIO supports the following controller form-factors:

Table 4-17. VIO Controller Form-Factors

Flavor	Small	Medium	Large
CPU	4	8	12
Memory (GB)	16	32	32
Controller Disk (GB)	25	50	75

Scaling VMware Integrated OpenStack

VMware Integrated OpenStack supports scaling out nodes, services, clusters, and datastores. Hence, the control and workload planes can be scaled independently as the size of a VIO deployment increases over time.

The minimum count for a highly available deployment is 3 controller nodes. However, the controllers can be scaled up to 10 to provide additional capacity.

Note Controllers of different sizes are not supported in a VIO deployment

With VIO, the sizing of the control plane VMs is not fixed. Based on the real-world conditions for your cloud, you can add additional controller VMs as a day-2 operation. The more tasks the control plane perform, the more resources are required. We recommend that medium or large-size controllers are used for production deployments.

After the deployment, monitor the CPU and memory consumptions on the controller VMs. If three controllers are consistently running high, scale them horizontally by adding additional controllers and restarting pods that are consuming more resources. Pod restart does not impact the service availability as a redundant copy is always available for each service. Pod restart triggers the Kubernetes scheduler to reassign the Pod to the newly added controller.

Sizing and Scaling Recommendations for VMware Integrated OpenStack

Design Recommendation	Design Justification	Design Implication
Use HA for production deployments.	Avoid a single point of failure.	VIO HA deployment requires a minimum of three controller nodes. A HA deployment requires 3-10 controllers.
Use the Medium or Large size for production deployment.	The amount of CPU and memory a control plane consumes is based on the number of API calls and the type of API calls that the control plane handles. Therefore, if you expect a high-churn environment with lots of API calls, move up to a larger size.	None
Deploy Large size if an optional service such as Ceilometer is used.	If the end-user never makes a call to Aodh/Gnocchi/Panko, Ceilometer uses more RAM and CPU compared to other OpenStack Services.	Increased CPU and memory.
Increase the number of Pod replicas if the API response is slow.	OpenStack Services run as Kubernetes deployments and can be scaled up/down to address application resource contention. Kubernetes offers a built-in mechanism to load balance API requests to the newly added Pod replicas.	None
Add more controller nodes if existing controllers are consistently running high in CPU and memory.	Kubernetes scheduler reassigns high utilization Pods to the newly added controller to even out the usage across nodes.	■ Pod Restart is required to move OpenStack services. ■ Node scaleout cannot be undone.
Use the Kubernetes node cordon to control the Pod to Controller assignment.	Cordon removes a node from the Kubernetes scheduler when the scheduler is placed in the Cordon state.	None

Telco Cloud Platform Design

The Telco Cloud Platform tier focuses on the components that are necessary for the deployment and management of CNF workload for 4G, 5G Core, and RAN. The foundational components such as the hypervisor, storage virtualization and network virtualization are consistent as applied across the Infrastructure Tier.

The Telco Cloud Platform Tier provides the foundational capabilities for all workload types such as 4G, 5G Core, RAN, and so on. In an environment that focuses only on 5G Core or RAN cloud-native network functions, the foundational components of the Telco Cloud Infrastructure Tier must be considered.

Telco Cloud Architectures

This section describes core and edge connectivity requirements to support different deployments within the Telco Cloud.

- **Core and Edge connectivity**: Core and Edge connectivity provides application-specific SLAs. It can have a significant impact on the 4G, 5G core, and RAN deployments. The type of radio spectrum, connectivity, and available bandwidth can have a significant impact on the placement of VNFs and CNFs.
- **WAN connectivity**: In the centralized deployment model, the WAN connectivity must be reliable between the sites. Any unexpected WAN outage prevents 5G user sessions from being established as all 5G control travels from the edge to the core.
- **Components deployment in Cell Site**: Due to the physical constraints of remote Cell Site locations, place only the required function at the Cell Site and deploy the remaining components centrally. For example, the platform monitoring and logging are often deployed centrally to provide universal visibility and control without replicating the entire core data center at the remote edge locations. Non-latency-sensitive user metrics are often forwarded centrally for processing.
- **Available WAN bandwidth**: The available WAN bandwidth between Cell Site and Central Core sites must be sized to meet the worst-case bandwidth demand. Also, when multiple classes of an application share a WAN, proper network QoS is critical.
- **Fully distributed 5G core stack**: A fully distributed 5G core stack is ideal for private 5G use cases, where the edge data center is self-contained. It survives extended outages that impact connectivity to the core data center. The Enterprise edge can be the aggregation point for the 5G Core control plane, UPF, distributed radio sites, and selective mobile edge applications. A fully distributed 5G core reduces the dependency on WAN but it increases the compute and storage requirements.
- **Network Routing in Cell Site**: Each Cell Site can locally route the user plane traffic and all the Internet traffic through the local Internet gateways, while the management and non-real-time sensitive applications leverage the core for device communication.

The following table lists the key differences between the deployments of a 5G core platform and a RAN platform. Most Telco Cloud deployments include both Core and RAN components.

Category	Core	RAN
Kubernetes Cluster	Single site	Stretched
Management Domain	Same	Same
Workload Domain composition	vSphere clusters	ESXi Hosts
Storage	vSAN/Shared storage	Local storage
Networking	Overlay / VLAN	VLAN / SRIOV

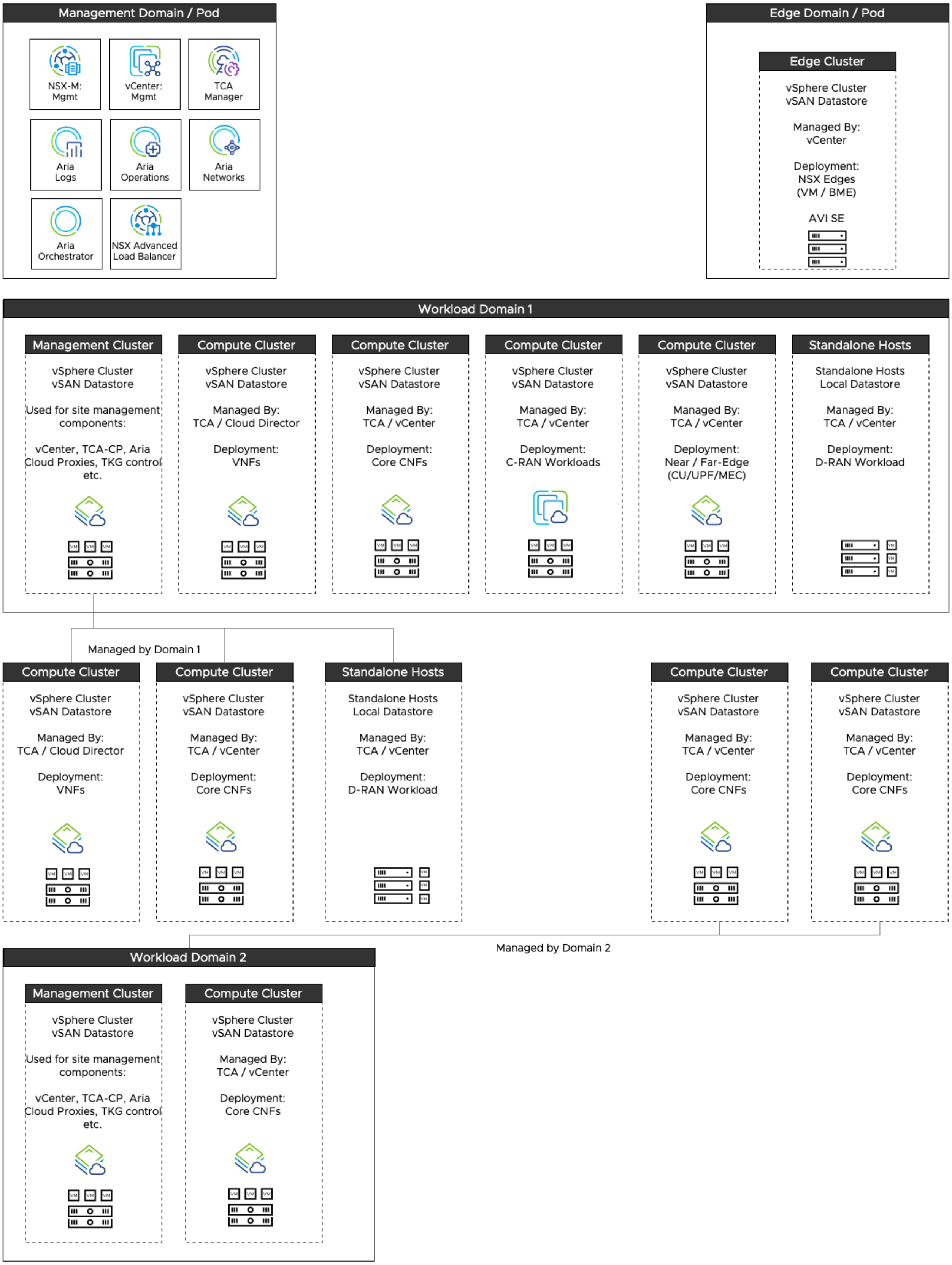
Telco Cloud - Core Domain

The Core domain for the Telco cloud leverages the 4G and 5G Core VNFs and CNFs on top of a horizontal cloud platform.

The core of the Telco Cloud, which is used for the deployment of non-RAN VNFs and CNFs, is based on the concepts of workload domains and clusters as highlighted in Workload Domains and vSphere Cluster Design.

The core of the Telco Cloud is a horizontal telco cloud architecture that is suitable for multi-function, multi-vendor implementations, and multi-cloud deployments. Repeated deployments of common, reusable modular blocks are used to create the core of the Telco Cloud.

Figure 4-30. Modular Telco Cloud Core



In the modular deployment of the Telco Cloud core, a centralized management domain is created along with two workload domains.

- Workload Domain 1 supports a full suite of Telco cloud services. It serves VNF, Core CNFs, CRAN deployments as well as edge and cell sites. An additional Edge domain is created for NSX and Load Balancer edge nodes.
- Workload Domain 2 is a smaller domain with its own Management cluster but serves only the core CNF.

In this Core architecture, each workload domain has some remote clusters. These remote clusters are connected to the workload domain vCenter but are geographically distributed from a single location for the management and compute clusters.

Telco Cloud - RAN Domains

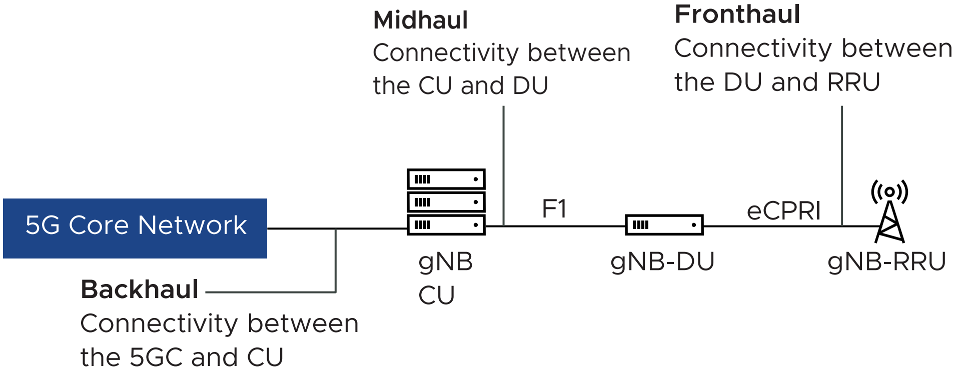
In RAN virtualization, the baseband radio functions are moved from custom-built hardware to vendor-agnostic Commercial Off-the-Shelf (COTS) hardware.

In 3GPP R15, the division of the upper and lower sections of the RAN was standardized. The higher-layer split is specified with a well-defined interface (F1) between the Centralized Unit (gNB-CU) and the Distributed Unit (gNB-DU). The CU and its functions, which are similar to the radio, have less stringent processing specifications and are more virtualization-friendly than the DU and its functions. The enhanced Common Public Radio Interface (eCPRI) links the DU to the radio.

The benefits of a fully virtualized or Open-RAN (O-RAN) are as follows:

- A single uniform hardware platform can be used across the core, RAN, and edge networks. This simplifies network management while lowering operational and maintenance costs.
- The network functions and computing hardware are isolated in a fully virtualized RAN. The network functions of the RAN can be performed on the same hardware, giving the service provider more versatility. The functionality and capacity of a virtualized RAN can be easily implemented where and when it is required, giving it more flexibility.

Figure 4-31. RAN Transport Network Terminologies



vRAN can be designed in multiple ways. The main interfaces are Fronthaul, Midhaul, and Backhaul. Clocking between the gNB-DU and the gNB-RRU is also significant. The Fronthaul can be logically split into S-Plane, M-Planes, and U-Planes for Clocking, Management, and data traffic.

The main units of the gNodeB are as follows:

- **Centralized Unit (CU)** provides non-real-time processing and access control. It manages higher layer protocols including Radio Resource Control (RRC) from the Control Plane, and Service Data Adaptation Protocol (SDAP) and Packed Data Convergence Protocol (PDCP) from the User Plane. The CU is connected between the 5G core network and the DUs. One CU can be connected to multiple DUs.

- **Distributed Unit (DU)** provides real-time processing and coordinates lower layer protocols including Physical Layer, Radio Link Control (RLC), and Media Access Control (MAC).

- **Remote Radio Unit (RRU)** does the physical layer transmission and reception, supporting technologies such as Multiple Input Multiple Output (MIMO).

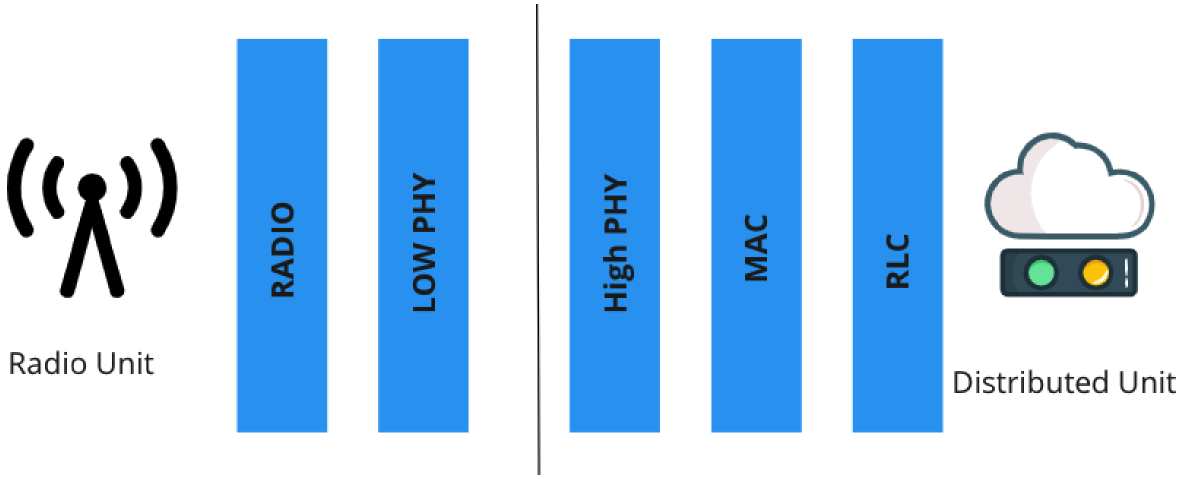
3GPP defined eight functional split options for Fronthaul networks. Options 2 and 7.x are the most commonly adopted Radio Splits.

- **Option 2**: A high-level CU and DU split. The CU handles Service Data Adaptation Protocol (SDAP) or Packet Data Convergence Protocol (PDCP) with Radio Resource Control (RRC) while L2/L1 Ethernet functions reside in the DU. Before the data is sent across the Medium haul network, aggregation and statistical multiplexing of the data are done in the DU. So, the amount of data transmitted across the interface for each radio antenna appliance is reduced.
- **Option 7.x**: A low-level DU and RU split. The DU handles the RRC, PDCP, Radio Link Control (RLC) MAC and higher Physical (PHY) functions. The RU handles the lower PHY and RF functions. A single DU is typically co-located with multiple RUs, offloading resource-intensive processing from multiple RUs. CU can be centrally located across the WAN, aggregating multiple DUs. Option 7.x lets operators simplify the deployment of the DU and RU, leading to a cost-effective solution and an ideal option for a distributed RAN deployment. Use LLC-C3 for PTP synchronization between the RU and DU.

Mobile operators require the flexibility to choose different splits based on the server hardware and fronthaul availability. Higher-layer functional splits are required for dense urban areas and scenarios, while a low fronthaul bit rate is required for a fronthaul interface.

The 7.2 split model is used commonly for O/V-RAN deployments. It distributes the L1 PHY functions between the RU and the DU, enabling a variable uplink on the F1 interface. It also allows the DU to have more functionality and process multiple sectors and sub-carriers, simplifying the RRU.

Figure 4-32. 7.2 Split Model



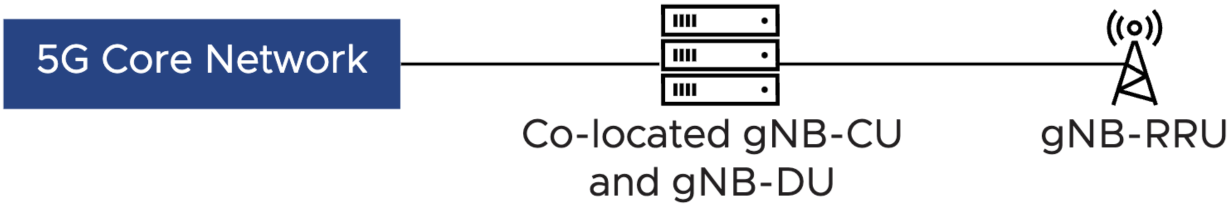
vRAN Design Approaches

The different design approaches of vRAN are as follows:

Co-located CU and DU:

A non-centralized approach utilizes the CU and DU functions co-located, with RRU physically separated.

Figure 4-33. Co-located CU / DU

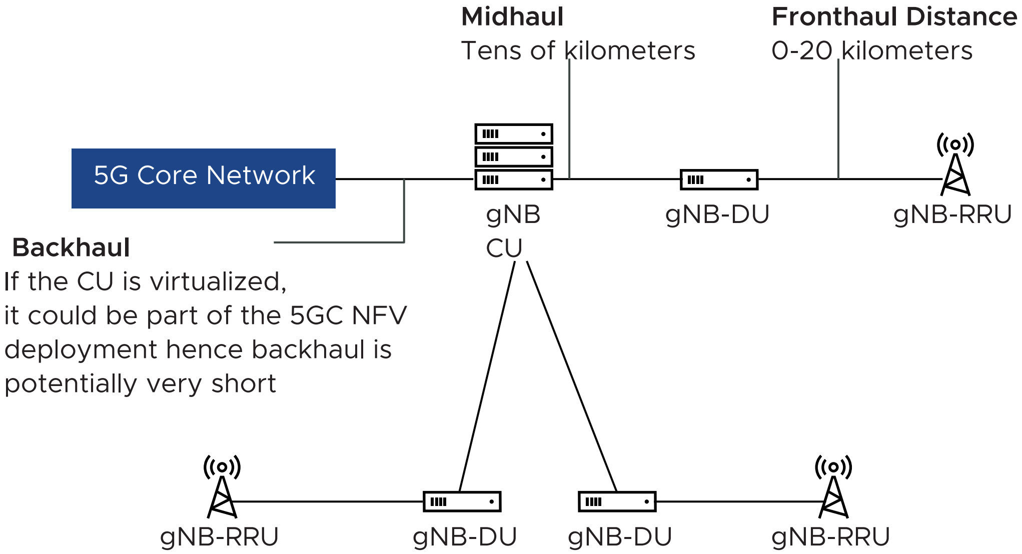


Note In this model, only fronthaul and backhaul interfaces are required, Fronthaul to the RU and backhaul from the CU.

Centralized Processing:

In the centralized approach, all functional elements of the gNB are physically separated. A single CU is responsible for several DUs. The design requirements of the Next-Generation RAN (NG-RAN) require specific transport network specifications to meet the required distances. This design model requires fronthaul, midhaul, and backhaul connectivity and is often called Centralized-RAN (C-RAN) or a BBU hotel like model.

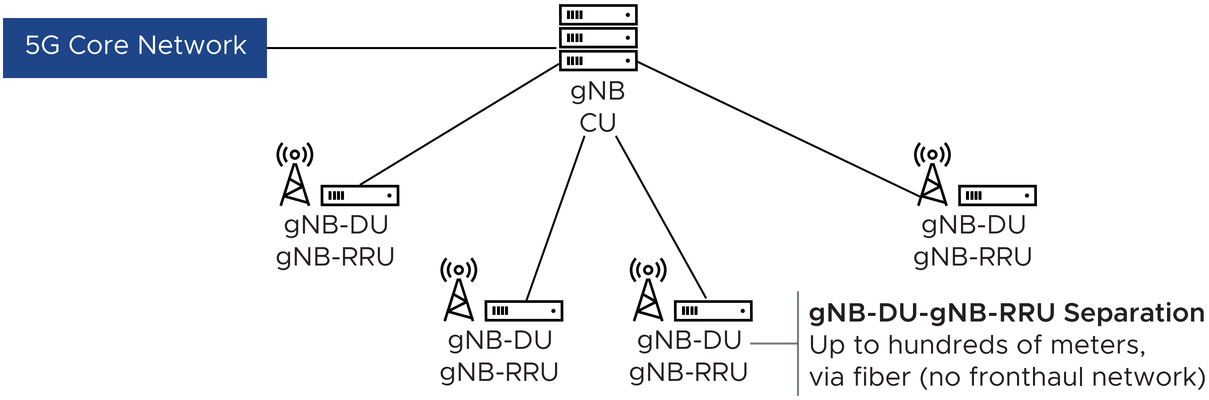
Figure 4-34. Centralized Processing (C-RAN)



DU and RRU Co-Located(D-RAN):

In this vRAN design, the DU and RRU are co-located such that they are directly connected without a fronthaul transport network. This connection is fiber-based and may span hundreds of meters, supporting scenarios where the DU and RRU are within the same building. This design requires midhaul and backhaul connectivity as shown in the following figure and is often called the Distributed RAN (D-RAN) model.

Figure 4-35. DU / RU Co-Located (D-RAN)


5G Core Network
gNB
CU
gNB-DU
gNB-RRU
gNB-DU
gNB-RRU
gNB-DU
gNB-RRU
gNB-DU
gNB-RRU
gNB-DU-gNB-RRU Separation
Up to hundreds of meters,
via fiber (no fronthaul network)

The centralized processing model introduces several potential advantages:

- **Cost Reduction**: Centralized processing capability reduces the cost of the DU function.
- **Energy Efficiency, Power and Cost Reduction**: Reducing the hardware in the cell site, reduces the power consumption and air conditioning of that site. The cost saving can be significant when you deploy tens or hundreds of cell sites.

- **Flexibility**: Flexible hardware deployment leads to a highly scalable and cost-effective RAN solution. Also, the functional split of the protocol stack impact the transport network.
- **Higher Performance**: Better performance is achieved through improved load management, cell coordination, and the future deployment of Radio interference mitigation algorithms.
- **Improved offload and content delivery**: Aggregation of processing at the CU provides an optimal place in the network for data offload and MEC application delivery.

The Distributed model (DU/RU Co-Located) also introduces several advantages:

- **Transport Costs**: This model avoids unnecessary transport (backhaul) of the RU-DU fronthaul interface. This can be significant in terms of bandwidth requirements of the fronthaul interface.
- **Proximity**: Allows the DU to be placed closer to the RU and introduces support for additional clocking methodologies such as LLS-C1.

Most RAN networks are comprised of DU nodes at the remote-cell sites, co-located with the RRU, and a smaller portion of the centralized model where the DUs are collected at a far/near-edge location. This allows maximum flexibility in terms of workload distribution and edge functionality in the future.

C-RAN Design Considerations

The Centralized RAN approach consolidates multiple ESXi hosts in a single place, aligning to the concept of a near edge or far-edge design. The following considerations are applicable for designing a C-RAN workload cluster.

- Do not use vSAN as the main datastore. The extreme real-time nature of the DU may be impacted by vSAN rebuilds or other events.
- The C-RAN TKG deployment can use a single vSphere cluster as the target endpoint. This allows the creation of a single node pool of TKG worker nodes and the provisioning of one-time Dynamic Infrastructure for all worker nodes in the cluster.

PTP Time Synchronization

RAN maintains network timing distribution as the preferred method for PTP time synchronization. For the RAN to operate effectively, the RU, DU, and CU must be time and phase synchronized. The delayed synchronization can have a negative impact on network applications. For example, low throughput, poor attachment success rate, and poor delivery success rate.

The accuracy of time synchronization depends mostly on the implementation of network connectivity and PTP protocol distribution. For example, the timestamp near interfaces and the number of hops. The O- RAN.WG4 Fronthaul networks define the following synchronization topologies for telco deployment:

- LLS-C1 Configuration
- LLS-C2 Configuration
- LLS-C3 Configuration

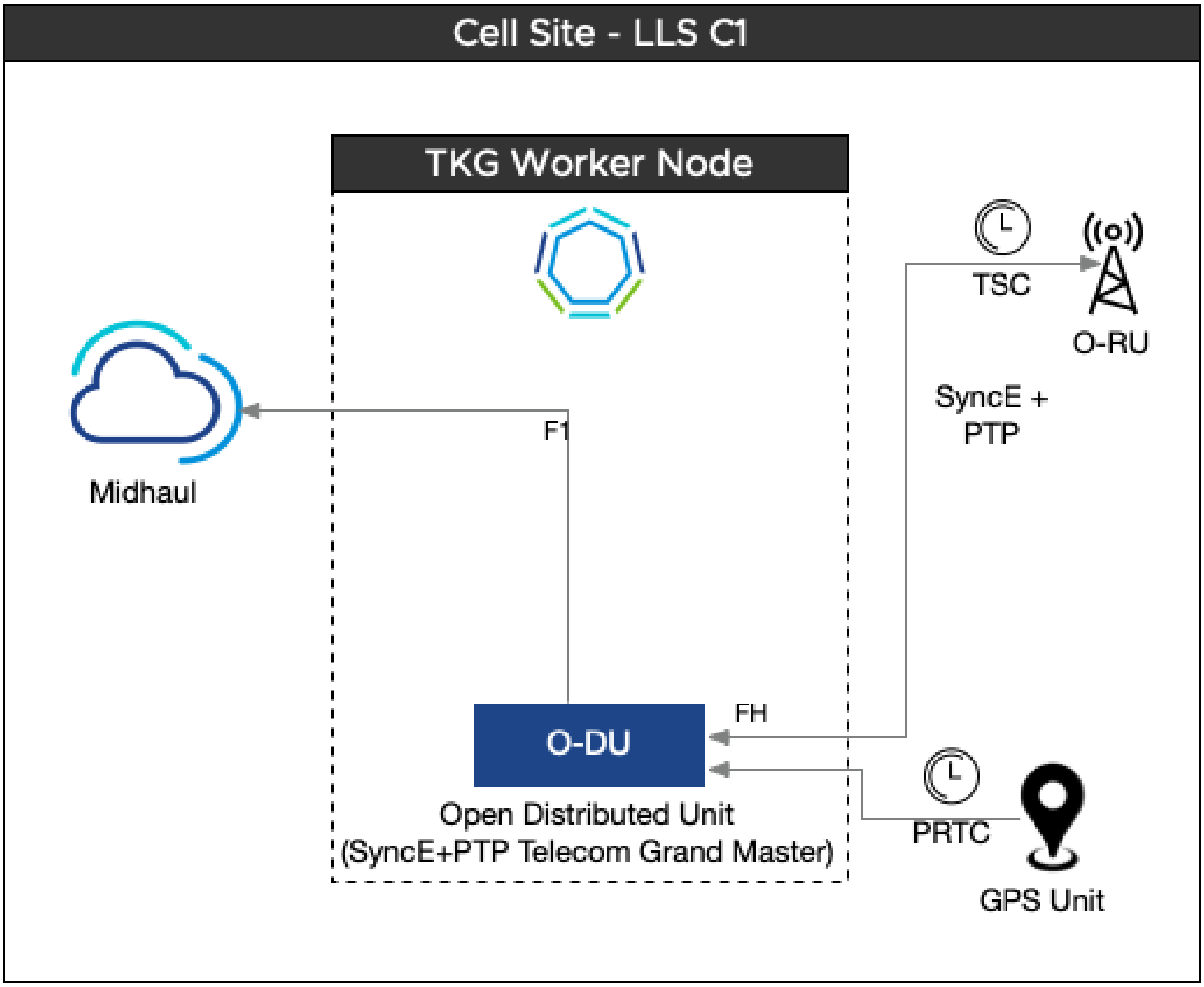
- LLS-C4 Configuration

Note Consider PTP time synchronization based on these designs. However, Telco Cloud Platform RAN supports LLS-C3 configuration and LLS-C1 when using specific hardware.

LLS-C1 Architecture

This configuration is based on the point-to-point connection between DU and RU by using the network timing option. LLS-C1 is simple to configure. In this configuration, DU operates as PTP grandmaster. The O-DU derives the time signal from (typically an onboard GNSS) Grandmaster and communicates directly with the O-RU to synchronize it.

Figure 4-36. LLS-C1



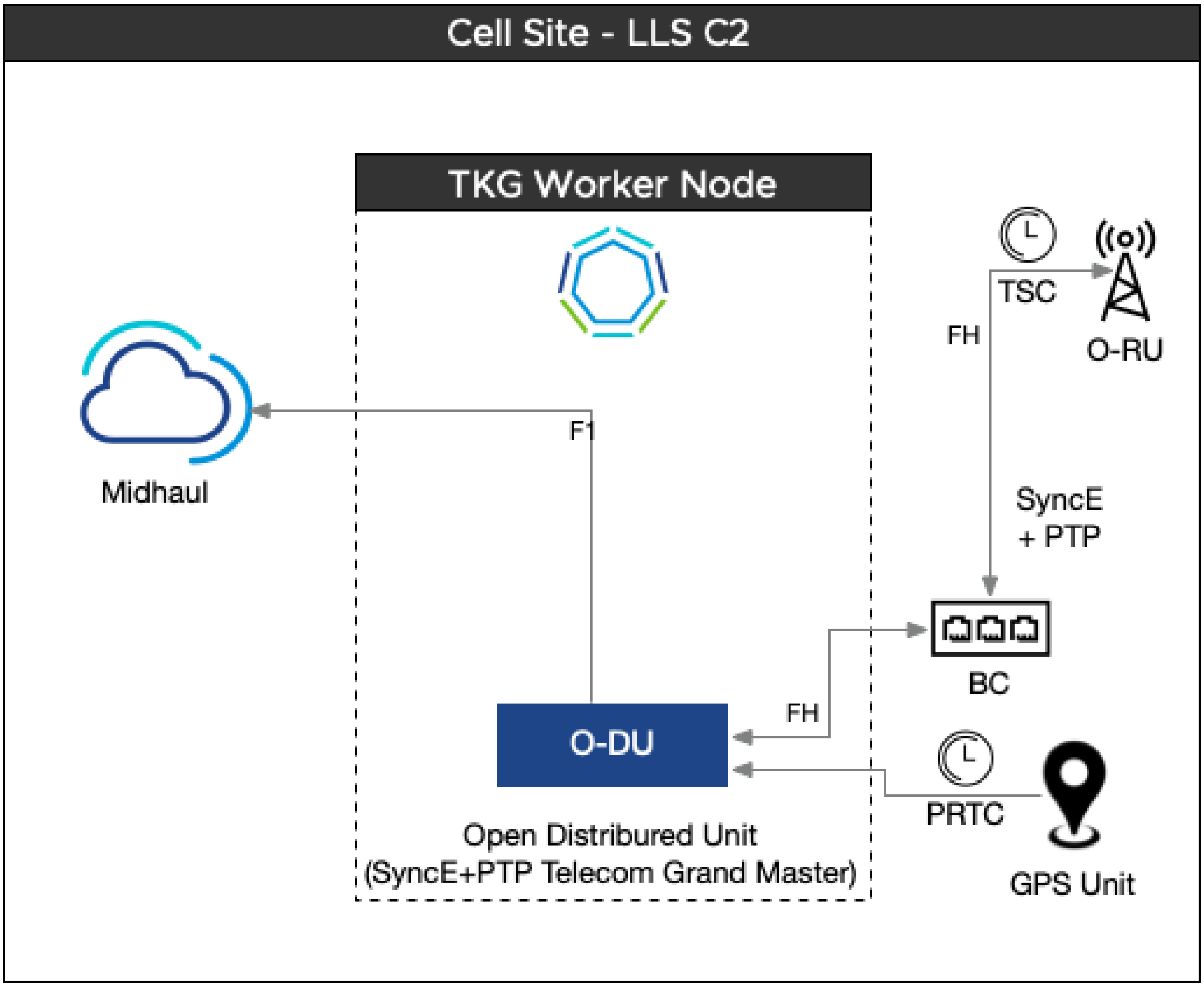
The architecture for LLS-C1 requires the O-RU units to be physically homed to the ESXi server.

Note To implement LLS-C1, specific hardware is required. Network Interface Cards with onboard GNSS are leveraged to support LLS-C1 deployments. With this configuration, the GM functionality is integrated into the server and no longer resides as an external component.

LLS-C2 Configuration

In this configuration, the O-DU acts as PTP Grandmaster allocating network timing towards the RU. One or more PTP-supported switches can be installed between the O-DU and the O-RU.

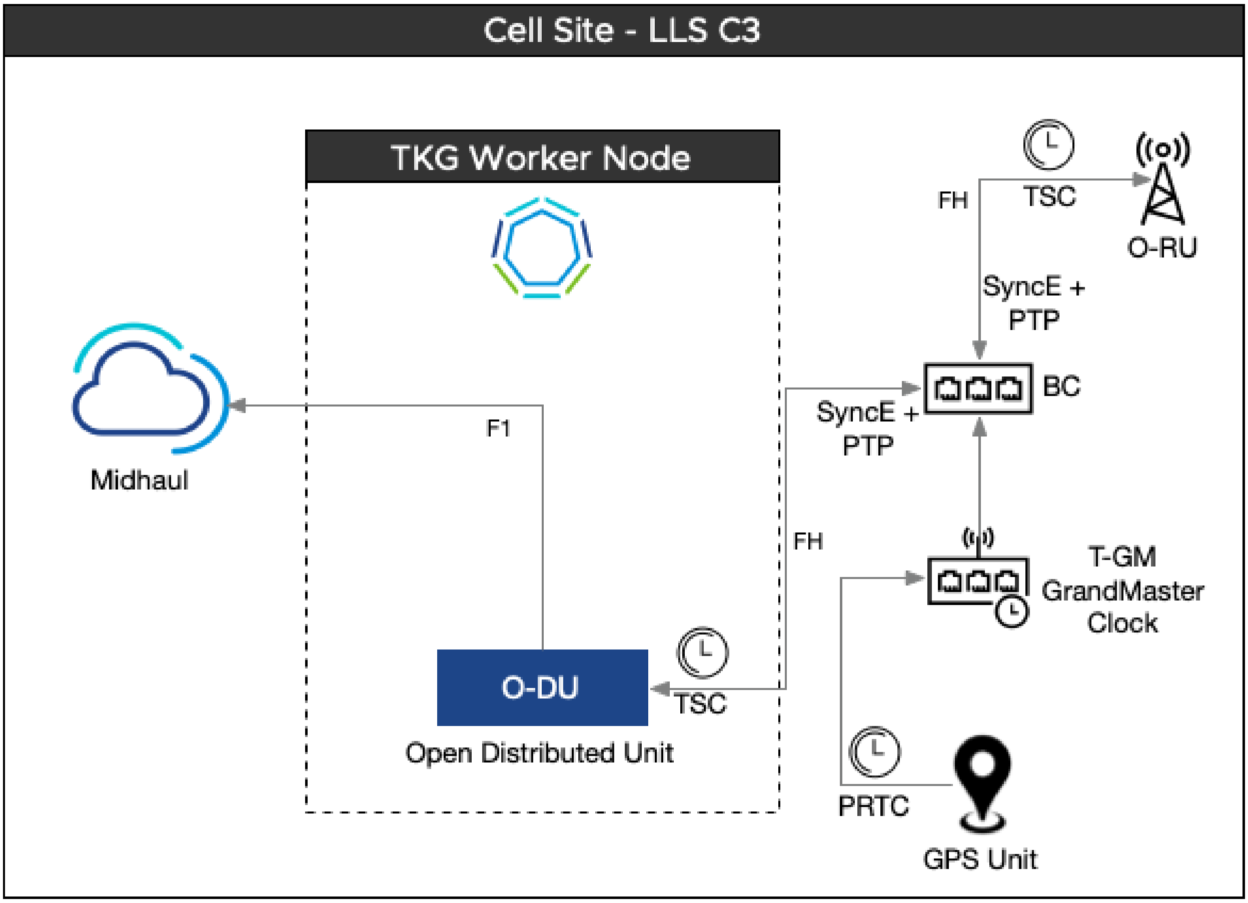
Figure 4-37. LLS-C2



LLS-C3 Configuration

In this configuration, the PTP Grandmaster performs network time-sharing between the O-DU and O-RU at Cell Sites. One or more PTP switches are allowed in the Fronthaul network to support network time-sharing. This architecture is widely adopted by introducing the PTP Grandmaster and PTP Switch, which provide the ideal solution for network time-sharing.

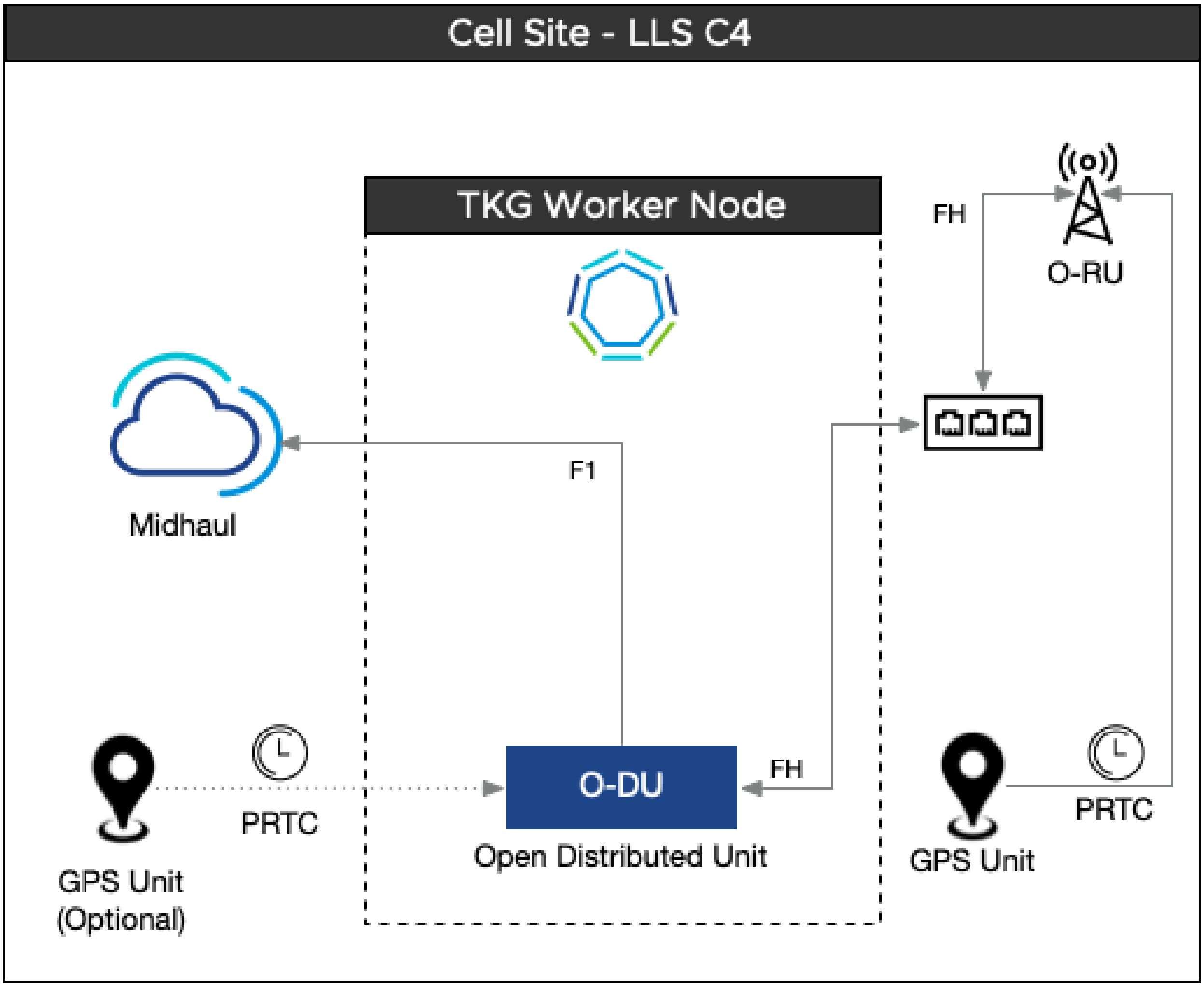
Figure 4-38. LLS-C3



LLS-C4 Configuration

In this configuration, PRTC (usually the GNSS receiver) is used locally to provide timing for the O-RU. PRTC does not depend on the Fronthaul transport network for timing and synchronization. The same or a separate GNSS can be used to provide clocking directly to the O-DU. Clock-sync is not supported using the direct link between the O-RU and O-DU.

Figure 4-39. LLS-C4



RAN Scaling Considerations

This section highlights the considerations for deploying RAN at large scale.

Out of the three common models for deploying the RAN environment, the most common are the RU-DU Co-Located (D-RAN) and the centralized processing (C-RAN) models.

Designing a RAN at scale involves various elements, each with its own scale limitations. Logical building blocks can be created and re-used at scale to provide scale-out and scale-up points based on the scale limits.

Some of the scaling considerations from the Telco Cloud Automation platform are as follows. For more details about the configuration maximums for VMware products, see VMware Configuration Maximums.

- Maximum number of TCA-CP nodes to a single TCA Manager
- Maximum VI registrations to a single TCA Manager

- Maximum number of Tanzu Kubernetes Management Clusters per single TCA manager and TCA control-plane node.
- Maximum number of workload clusters per TCA Manager
- Maximum number of Tanzu Kubernetes Workload clusters per TCA control-plane node.
- Maximum number of workers in a single node pool
- Maximum number of node pools in a single workload cluster
- Maximum number of workers nodes in a single workload cluster
- Maximum number of Network Functions managed by a single TCA control-plane node
- Maximum number of Network Functions managed by a single TCA Manager.

Other scaling considerations are based on the Telco Cloud and the vendor RAN architecture:

- Maximum number of RAN hosts per vCenter
- Maximum number of DU workloads to a single or redundant CU
- Maximum number of DU workloads to the vendor management elements.

Important While each product has its own configuration maximums, it must not exceed 70-80% of a maximum.

In addition to configuration maximums, the most prominent consideration is the blast radius. While you can deploy 2,500 ESXi servers into a single vCenter, the blast radius must be relatively wide when the vCenter experience an outage. The connectivity and management of 2,500 servers will be unavailable when the vCenter is under outage or maintenance.

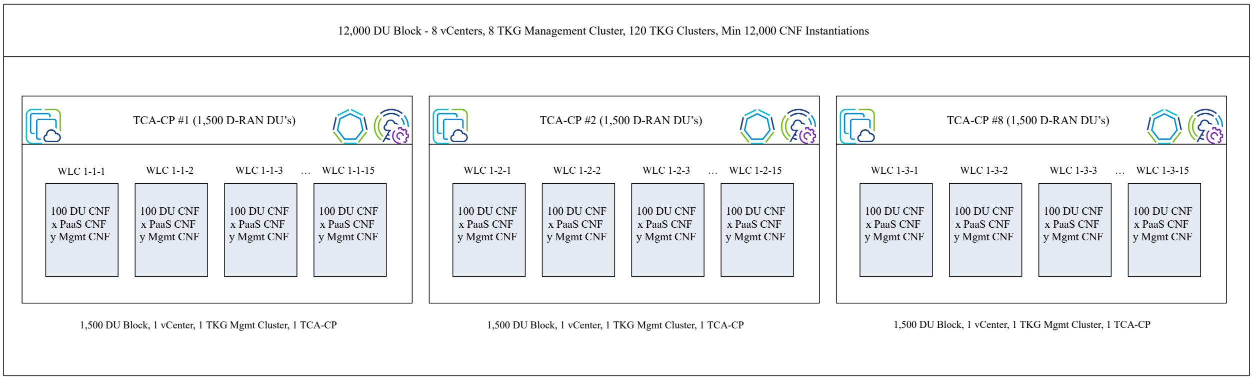
The scale considerations are different for each deployment. Detailed discussions with the CSP and RAN vendors are required to ensure appropriate sizing and scaling considerations or constraints.

The RAN can be divided into logical blocks. A sample structure of these blocks, with different maximums and constraints are as follows:

- Each Tanzu Kubernetes Workload cluster can host a maximum of 100 DU's deployed in a D-RAN deployment mode, with one worker node or DU per ESXi host.
- A maximum of 15 workload clusters per Tanzu Kubernetes Management cluster
- A single Tanzu Kubernetes Management cluster per TCA-CP or vCenter

This structure creates a block of up to 1,500 DU nodes per TCA-CP node, with 1,500 ESXi hosts per vCenter. This larger block can then be scaled out to create multiple blocks of 1,500 DU nodes until a higher scale limit is reached. It can be eight large blocks per TCA Manager, accommodating up to 12,000 DU nodes to a single TCA manager. To increase the size of the RAN deployments, the sizes of the individual scale elements can be adjusted according to the configuration maximums.

Figure 4-40. Sample RAN Dimensioning


12,000 DU Block - 8 vCenters, 8 TKG Management Cluster, 120 TKG Clusters, Min 12,000 CNF Instantiations
TCA-CP #1 (1,500 D-RAN DU's)
WLC 1-1-1
WLC 1-1-2
WLC 1-1-3
...
WLC 1-1-15
100 DU CNF
x PaaS CNF
y Mgmt CNF
1,500 DU Block, 1 vCenter, 1 TKG Mgmt Cluster, 1 TCA-CP
TCA-CP #2 (1,500 D-RAN DU's)
WLC 1-2-1
WLC 1-2-2
WLC 1-2-3
...
WLC 1-2-15
100 DU CNF
x PaaS CNF
y Mgmt CNF
1,500 DU Block, 1 vCenter, 1 TKG Mgmt Cluster, 1 TCA-CP
TCA-CP #8 (1,500 D-RAN DU's)
WLC 1-3-1
WLC 1-3-2
WLC 1-3-3
...
WLC 1-3-15
100 DU CNF
x PaaS CNF
y Mgmt CNF
1,500 DU Block, 1 vCenter, 1 TKG Mgmt Cluster, 1 TCA-CP

Note A single RAN DU cluster needs more than just the DU Network Function to be instantiated. Depending on the overall RAN DU requirements and vendor requirements, additional PaaS and management components may be necessary and must be factored into the CNF scaling and dimensioning.

When planning the dimensioning and scale requirements for RAN deployments, note that each Tanzu Kubernetes workload cluster requires resources for the control-plane nodes. These control-plane nodes can reside in the Domain or Site management cluster or in a vSphere cluster associated with the same vCenter. Different elements of a single Tanzu Kubernetes workload cluster must not be deployed across multiple vCenter Servers.

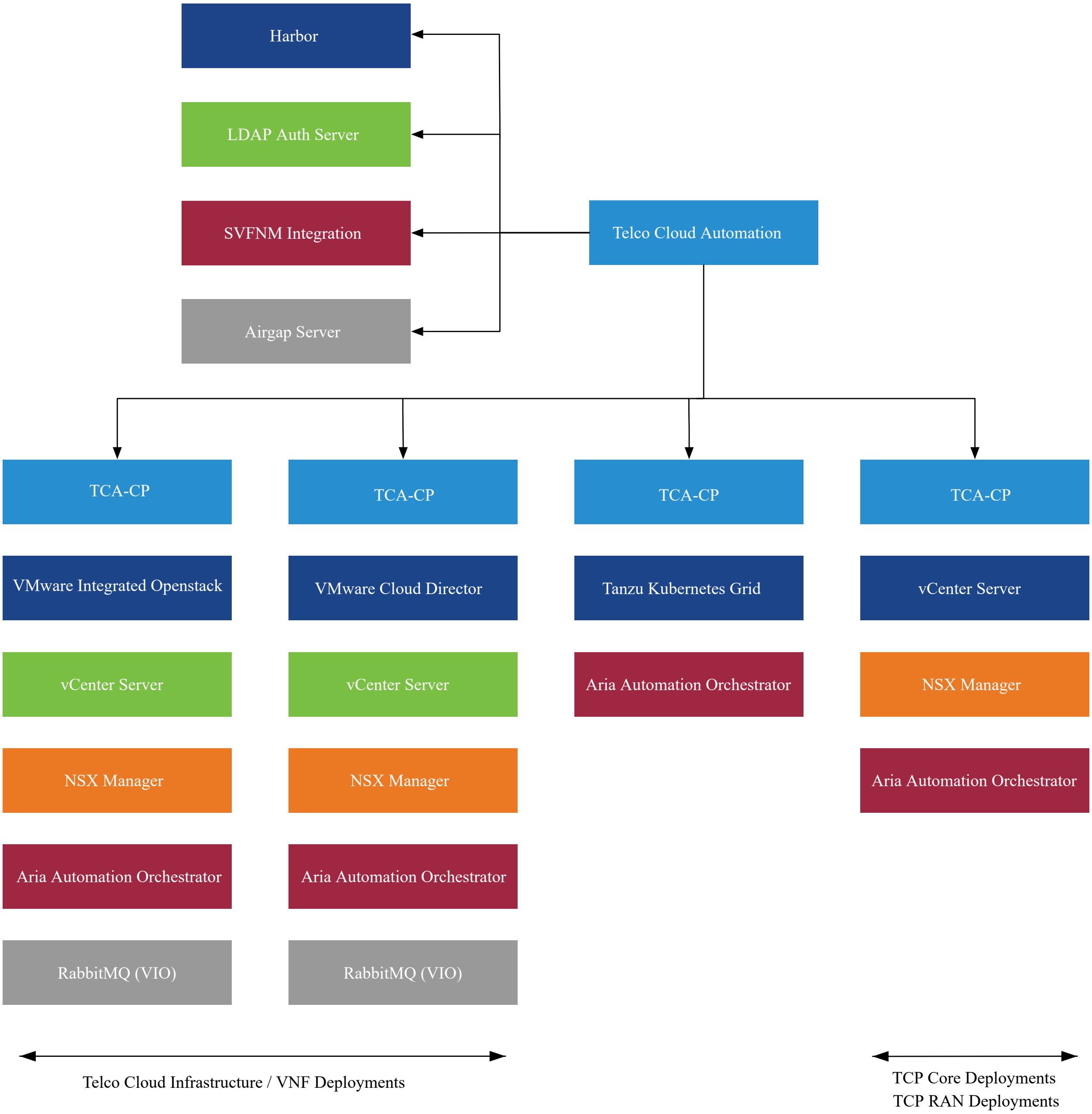
Note When deploying numerous nodes concurrently for RAN, be aware of the per-host and per-datastore limits for cloning. A maximum of eight clone operations can occur per host or per datastore. For a large concurrent deployment, deploy the OVA template per host and automate the deployment to limit the concurrent clone operations to eight.

Telco Cloud Automation Design

This section outlines the design best practices of the Telco Cloud Automation (TCA) components including TCA Manager, TCA-Control Plane, NodeConfig Operator, Container registry, and CNF design.

Telco Cloud Automation distributes VIM and CaaS manager management across a set of distributed Telco Cloud Automation appliances. TCA-CP performs multi-VIM/CaaS registration and synchronizes multi-cloud inventories as shown in the following diagram. In addition, TCA collects faults and performance data from CaaS infrastructure and network functions.

Figure 4-41. Telco Cloud Automation Models



- **TCA Manager**: TCA Manager connects with TCA-CP nodes through site pairing to communicate with the VIM. It posts workflows to the TCA-CP. TCA manager relies on the inventory information captured from TCA-CP to deploy and scale Tanzu Kubernetes clusters.

- **TCA-Control Plane (TCA-CP)**: TCA CP connects to a specific VI (vCenter, Cloud Director, or VMware Integrated OpenStack) and provides the capabilities to deploy VNFs and CNFs to the cloud platform.

- **Tanzu Kubernetes Cluster**: Tanzu Kubernetes cluster bootstrapping environment is abstracted into the TCP-CP node. All the binaries and cluster plans required to bootstrap the Kubernetes clusters are pre-bundled into the TCP-CP appliance. After the base OS image templates are imported into respective vCenter Servers, Tanzu Kubernetes Cluster admins can log into the TCA manager and deploy Kubernetes clusters directly from the TCA manager console.
- **Workflow Orchestration**: By integrating VMware Aria Automation Orchestrator (formerly vRealize Orchestrator) , Telco Cloud Automation provides a workflow orchestration engine that is distributed and easily maintainable. Aria Automation Orchestrator workflows run operations that are not supported natively on TCA Manager. Using Aria Automation Orchestrator, you can create custom workflows or use an existing workflow as a template to design a specific workflow to run on your network function or network service. For example, you can create a workflow to assist CNF deployment or simplify the day-2 lifecycle management of CNF. Aria Automation Orchestrator is registered with TCA-CP.
- **Resource Tagging**: Telco Cloud Automation supports resource tagging. Tags are custom-defined metadata that can be associated with any component. They can be based on hardware attributes or business logic. Tags simplify the grouping of resources or components.

Telco Cloud VM & Node configuration operators

The VMConfig and NodeConfig Operators are essential for configuring the Tanzu Kubernetes clusters based on the Telco workload requirements.

The VMConfig and NodeConfig Operators are Kubernetes operators developed by VMware to handle the Kubernetes node and OS customization. The NodeConfig Operator can be used to customize DPDK binding, Kernel upgrade, OS module installation, and so on. VM-specific operations such as vNIC mapping, Network PortGroup assignment, vCPU pinning, and host memory reservation are handed by the VMConfig Operator.

NodeConfig Operator:

- Node Profile describes the intent that the node-config operator is going to fulfill. Node profile is stored as a Kubernetes ConfigMap.
- NodeConfig Daemon is a DaemonSet running on each node to realize the node profile config passed down to the NodeConfig Daemon as ConfigMap.
- NodeConfig Operator handles the node OS configuration, performance tuning, and OS upgrade. It monitors config update events and forwards events to backend Daemon plug-ins. Each plug-in is responsible for a specific type of event, such as Tuning, Package updates, SR-IOV device management, and so on. After each plug-in processes the update events, node labels are used to filter out a set of nodes to receive the node profile.

VMConfig Operator:

- VMConfig Operator handles VM configurations for Tanzu Kubernetes clusters as the CAPV/CAPI extension. It runs in the Tanzu Kubernetes management cluster.

- VMConfig Operator consists of
  - VM Controller: Monitors VMConfig and CAPI/CAPV events and configures Kubernetes worker nodes on the target workload cluster.
  - ESXInfoController: Responsible for hardware capabilities discovery on an ESXi host.

Telco Cloud Automation is the single source of truth for both VMConfig and NodeConfig operators. Based on the infrastructure requirements defined in the network function catalog TOSCA YAML (CSAR file), Telco Cloud Automation generates a node profile that describes the intended node config the operator is going to realize. The NodeConfig operator runs as Kubernetes DaemonSets on Kubernetes nodes and configures the worker node to realize the desired states specified by Telco Cloud Automation.

Telco Cloud Automation Design Recommendations

Design Recommendation	Design Justification	Design Implication
Integrate the TCA Manager with active directory for more control over user access.	■ TCA-CP SSO integrates with vCenter (not LDAP) ■ LDAP enables centralized and consistent user management.	Requires additional components to manage in the Management cluster.
Deploy a single instance of the TCA manager (of a permissible size) to manage all TCA-CP endpoints.	■ Single point of entry into CaaS ■ Simplifies inventory control, user onboarding, and CNF onboarding.	Large deployments with significant scale may require multiple TCA Managers.
Register the TCA manager with the management vCenter Server.	Management vCenter Server is used for TCA user onboarding if direct AD integration is not configured	None
Deploy a dedicated TCA-CP node to control the vSphere management cluster if any k8s management or workload clusters are required in the management domain.	Required for the deployment of the Tanzu Kubernetes Management cluster.	TCA-CP requires additional CPU and memory in the vSphere management cluster.
Deploy a TCA-CP node for each vCenter Server instance.	■ Each TCA-CP node manages a single vCenter Server. ■ Multiple vCenter Servers in one location require multiple TCA-CP nodes.	■ Each time a new vCenter Server is deployed, a new TCA-CP node is required. ■ To minimize recovery time during TCA-CP failure, each TCA-CP node must be backed up independently, along with the TCA manager.
Deploy TCA manager and TCA-CP on a shared LAN segment used by VIM for management communication.	■ Simplifies connectivity between Telco Cloud Platform management components. ■ TCA manager, TCA-CP, and VIM share the same level of the security trust domain.	None

Design Recommendation	Design Justification	Design Implication
Deploy a three-node Aria Automation Orchestrator cluster.	Ensures high-availability of the Aria Automation Orchestrator cluster for all TCA-CP endpoints.	Aria Automation Orchestrator redundancy requires an external Load Balancer.
Schedule TCA manager and TCA-CP backups at the same time as SDDC infrastructure components to minimize database synchronization issues upon restore. **Note**: Your backup frequency and schedule might vary based on your business needs and operational procedure.	■ Proper backup of all Telco Cloud Automation and SDDC components is crucial to restore the system to its working state in the event of a failure. ■ Time consistent backups taken across all components require less time and effort upon restore.	Backups are scheduled manually. TCA admin must log into each component and configure a backup schedule and frequency.

Telco Cloud Automation - Airgap Design

Isolating your infrastructure from Internet access is often a best practice, but it impacts the default operational mode of VMware Telco Cloud Automation. The Airgap solution eliminates the requirement for internet connectivity.

In the non-air-gapped design, Telco Cloud Automation uses external repositories for Harbor and the PhotonOS packages to implement the VM and Node Config operators, new kernel builds, or additional packages for the worker nodes. Internet access is required to pull these additional components.

The Airgap server is a Photon OS VM that is deployed and configured for use by Telco Cloud Automation. The airgap server is registered as a partner system within the platform and is used in internet-restricted or air-gapped environments.

The airgap server allows the VMware Tanzu Kubernetes Grid clusters to pull the required Kernels, Binaries, and OCI images from a local environment.

Note While the Airgap server removes the requirement for Internet access to build and manage Kubernetes clusters, the Airgap server creation requires Internet access to build and pull all the external images to be stored locally.

The Airgap server can be built on an Internet-accessible zone (direct or through proxy) and then migrated to the Internet-restricted environment and reconfigured before use.

The airgap server operates in two modes:

- **Restricted**: This mode uses a proxy server between the Airgap server and the internet. In this mode, the Airgap server is deployed in the same segment as the Telco Cloud Automation VMs in a one-armed mode design.
- **Air-gapped**: In this mode, the airgap server is created and migrated/moved to the air-gapped environment. The airgap server has no external connectivity requirements. You can upgrade the airgap server by a new Airgap deployment or an upgrade patch.

The Airgap server consists of the following main components along with a set of scripts for easy installation and configuration.

- **NGINX** is used to request files from the local datastore or harbor environment.
- **Harbor** is the container registry that hosts the OCI images required by VMware Telco Cloud Automation and VMware Tanzu Kubernetes Grid.
- **Reposync** synchronizes the air-gapped repository with the upstream repository located on the internet.
- **BOM Files** are used by the VMware Telco Cloud automation platform

Design Recommendation	Design Justification	Design Implication
Where required, leverage the air-gapped solution to eliminate direct Internet connectivity requirements.	▪ Provides a secure environment for the Tanzu Kubernetes Grid deployment as external access is restricted. ▪ Speeds up the Tanzu Kubernetes Grid deployment process by accessing the local infrastructure, without Internet connectivity.	Requires the airgap server to be deployed, maintained, and upgraded over time.

The typical method of deployment for the Airgap server is to have the platform created in an Internet-accessible environment. After the server is created, it can be powered off and relocated into the Telco Cloud environment. After the server is relocated, embedded scripts can be used to reconfigure the airgap server (change IP address, certificates, and so on).

Telco Cloud Automation - Container Registry Design

Telco Cloud workloads consist of CNFs in the form of container images and Helm charts from network vendors. This section outlines the recommendations to deploy and maintain a secure container image registry using VMware Harbor.

Container images and Helm charts must be stored in registries that are always available and easily accessible. Public or cloud-based registries lack critical security compliance features to operate and maintain carrier-class deployments. Harbor addresses these issues by providing trust, compliance, performance, and interoperability.

Note

- Telco Cloud Automation supports HELM charts stored either in the chartmuseum component of Harbor or as OCI charts pushed directly to the repository.
- The chartmuseum feature of Harbor is deprecated and scheduled for removal post Harbor 2.8 releases.

- **Image registry** allows users to pull container images and charts and the admins to publish new container images. Different categories of images are required to support Telco Applications. Some CNF images are required by all Tanzu Kubernetes users, while others are required only by specific Tanzu Kubernetes users. To maintain image consistency, Cloud admins might need to ingest a set of golden CNF public images that all tenants can consume. Kubernetes admins might also require the images not offered by the network vendors and may upload private CNF images or charts.

  To support Telco applications, organize CNF images and helm charts into various projects and assign different access permissions for those projects using RBAC. Integration with a federated identity provider is also important. Federated identity simplifies user management while providing enhanced security.
- **Container image scanning** is important in maintaining and building container images. Irrespective of whether the image is built from the source or from VNF vendors, it is important to discover any known vulnerabilities and mitigate them before the Tanzu Kubernetes cluster deployment. Trivy is the default scanning implementation with Harbor.
- **Image signing** establishes the image trust, ensuring that the image you run in the cluster is the image you intended to run. Notary digitally signs images using keys that allow service providers to securely publish and verify content. Components signing is done before uploading to a harbor project.
- **Container image replication** is essential for backup, disaster recovery, multi-datacenter, and edge scenarios. It allows a Telco operator to ensure image consistency and availability across many data centers based on a predefined policy.

  Images that are not required for production must be bounded to a retention policy, so that obsoleted CNF images do not remain in the registry. To avoid a single tenant from consuming all available storage resources, the resource quota can also be set per tenant project.

Harbor Deployment considerations

Telco Cloud Automation supports multiple harbors to be attached to a Tanzu Kubernetes cluster. However, this architecture allows the CSPs to leverage different harbors for different network functions in a distributed way. This architecture is not designed to offer harbor redundancy.

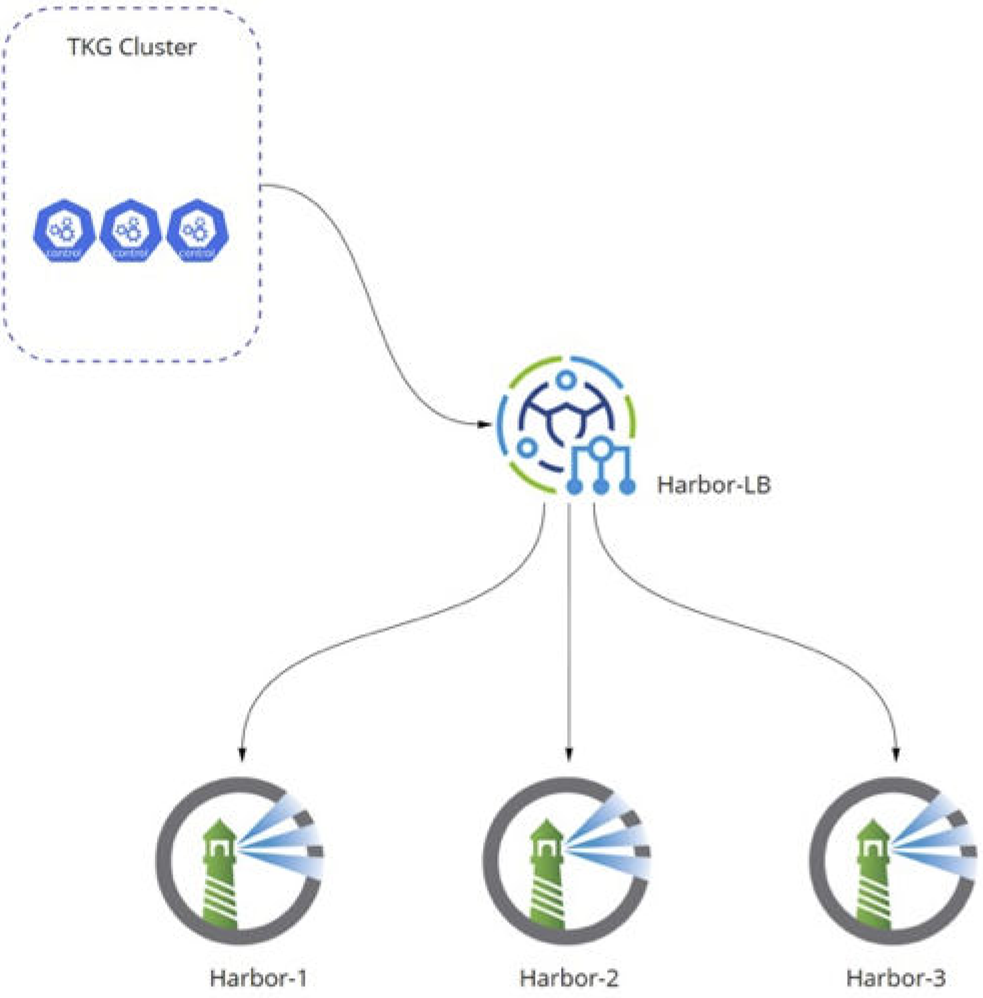
For pulling HELM charts, you can use the following two options to provide a redundant or resilient harbor:

- **Resilient harbor**: In the resilient Harbor method, a resilient harbor is deployed on top of cloud-native k8s infrastructure. It requires additional external components such as external highly available databases. This method is a single harbor deployment that is constrained by locality and other factors.

  Restriction To implement a highly available Harbor as a cloud-native application, external redis and PostgreSQL databases are required. These components are not covered in this reference architecture.

- **Load-balanced Harbor**: This method involves individually load-balanced harbor deployments. The load-balanced Harbor provides a resilient harbor. However, any pools of harbor VMs deployed behind the load-balancer must be in sync. If the images and charts are not properly replicated across all harbors, the load-balancer forwards a request to a harbor that does not have the image in the appropriate project.

Figure 4-42. Load-Balanced Harbor Deployment



The harbor and load-balancer can be configured in various modes, including one-armed load-balancer and routed load-balancer. Examples include simple round-robin with IP stickiness, proximity based, connection count, and so on. In a typical telco cloud deployment, multiple Harbors are deployed within a single region, leveraging a one-armed load-balancer with health checks.

Depending on the size of the Telco Cloud deployments, multiple Harbors can be deployed per region or domain to provide resiliency within a region and to provide a shorter path for image or chart pulls. This reduces traffic on the WAN and speeds up deployments.

When integrating harbors to a Tanzu Kubernetes Cluster through Telco Cloud Automation, only a single account can be used to access the harbor. To ensure that a tenant within Telco Cloud Automation can only access specific projects within the harbor, use the following two methods:

- Onboard multiple copies of the same harbor with different credentials and then assign the appropriate harbor to the TKG cluster.
- Leverage the ability to supply username and password credentials through the TCA UI when onboarding the CNF. This implies that the NF Deploy has the correct URL for the chart location and valid credentials to access the specific project within harbor.

Harbor replication is required to keep the projects across multiple harbors in sync. One of the major requirements for replication is that all harbor instances are running the same version.

Harbor replication consists of setting up endpoints and replication rules. The endpoints control the replication source and destination such as other Harbors, jFrog repositories, and so on. The replication rules specify a push/pull model for distributing projects or components of projects among different harbors.

Harbor Authentication

As part of a common authentication methodology, Harbor supports LDAP integration and RBAC policies.

Harbor can be configured to integrate with LDAP and leverage the same Authentication Directory as used by TCA, thereby achieving commonality across user and group configurations.

Harbor projects can be configured as Private or Public. Additionally, user access to projects can be configured on a per LDAP user or group basis. This allows a granular approach to securing visibility to harbor projects through traditional LDAP group membership.

Container Registry Design Recommendations

Design Recommendation	Design Justification	Design Implication
Integrate Harbor with existing User authentication providers such as OIDC or external LDAP/Active Directory server.	User accounts are managed centrally by an external authentication provider. Central authentication provider simplifies user management and role assignment.	Requires coordination between Harbor administrator and authentication provider.
Use Harbor projects to isolate container images between different users and groups. Within a Harbor project, map the users to roles based on the TCA persona.	Role-based access control ensures that only authorized users can access private container images.	None
Use quotas to control resource usage. Set a default project quota that applies to all projects and use project-level override to increase quota limits upon approval.	Quota system is an efficient way to manage and control system utilization.	None
Enable Harbor image Vulnerability Scanning. Images must be scanned during the container image ingest and daily as part of a scheduled task.	Non-destructive form of testing provides immediate feedback on the health and security of a container image. Proactive way to identify, analyze, and report potential security issues.	None
Centralize the container image ingestion at the core data center and enable Harbor replication between Harbor instances using push mode. Define replication rules such that only the required images are pushed to remote Harbor instances.	Centralized container image ingestion allows better control of image sources and prevents the intake of images with known software defects and security flaws. Image replication push mode can be scheduled to minimize WAN contention. Image replication rule ensures that only the required container images are pushed to remote sites. Better utilization of resources at remote sites.	None
When using a load-balancer harbor, leverage the NSX Advanced Load Balancer to provide the L4 service offering.	Leverages VMware stack for simplified LB deployment	Additional SE deployments might be required to provide the LB services

Design Recommendation	Design Justification	Design Implication
Ensure the LB and harbor nodes are using CA signed certs	Eliminates complexity around container management and certificates	None
If needed create an intake / replication strategy that simplifies the distribution of OCI images and charts to the different harbor deployments.	Ensures a defined process for both intake and image replication	Requires planning to build an intake process from multiple network vendors and a proper replication strategy to ensure that images are available across all required harbor deployments

Telco Cloud Automation - CNF Design

This section outlines the CNF requirements and how CNFs can be onboarded and instantiated across the Telco Cloud.

Helm Charts

Helm is the default package manager in Kubernetes. CNF vendors use Helm to simplify container packaging. With Helm charts, dependencies between CNFs are handled in the formats agreed upon by the upstream community; allowing Telco operators to consume CNF packages in a declarative and easy-to-operate manner. With proper version management, Helm charts also simplify workload updates and inventory control.

Helm repository is a required component in the Telco Cloud Platform. Production CNF Helm charts must be stored centrally and accessible by the Tanzu Kubernetes clusters. To reduce the number of management endpoints, the Helm repository must work seamlessly with container images. A container registry must be capable of supporting both container images and Helm charts.

Note The Chartmuseum feature of Harbor is scheduled for deprecation. Telco Cloud Automation now supports both OCI-based charts and chartmuseum-based charts.

CNF Cloud Service Archive (CSAR) Design

Network Function (NF) Helm charts are uploaded as a catalog offering wrapped around the ETSI-compliant TOSCA YAML (CSAR) descriptor file. The descriptor file includes the structure and composition of the NF and supporting artifacts such as Helm charts version, provider, and set of pre-instantiation jobs.

Telco Cloud Network Functions have a set of prerequisite configurations such as node sizing and base features on the underlying Kubernetes cluster. Telco Cloud Automation also supports Dynamic Infrastructure Provisioning (DIP). These requirements are also defined in the Network Function CSAR. The summary of features supported by the CSAR extension includes:

- Interface configuration and addition, along with DPDK binding
- NUMA Alignment of vCPUs and Virtual Functions
- Latency Sensitivity

- Custom Operating system package installations
- Full GRUB configuration

The following table outlines those extensions in detail:

Component	Type	Description
node_components	Kernel_type	Type of Linux Kernel and version. Based on the Kernel version and type, Telco Cloud Automation downloads and installs the Linux kernel from VMware Photon Linux repo (or airgap server) during Kubernetes node customization.
	kernel_args	Kernel boot parameters are required for CPU isolation. Parameters are free-form text strings. The syntaxes are as follows: Key: the name of the parameter Value: the value corresponding to the key **Note**: The Value field is optional for those Kernel parameters that do not require a value.
	kernel_modules	Kernel Modules are specific to DPDK. When the DPDK host binding is required, the name of the DPDK module and the relevant version are required.
	custom_packages	Custom packages include lxcfs, tuned, and pci-utils. Telco Cloud Automation downloads and installs from VMware Photon Linux repo during node customization.
network	deviceType	Types of network device. For example, vmxnet3.
	resourceName	Resource name refers to the label in the Network Attachment Definition (NAD).
	dpdkBinding	The PCI driver this network device must use. Specify "igb_uio" or "vfio" in case DPDK or any equivalent driver depending on the vendors.
	count	Number of adapters required
caas_components		CaaS components define the CaaS CNI, CSI, and HELM components for the Kubernetes cluster.

CSAR files can be updated to reflect changes in CNF requirements or deployment models. CNF developers can update the CSAR package directly within the TCA designer or leverage an external CICD process to maintain and build newer versions of the CSAR package.

VMware Telco Cloud Automation supports rolling upgrades of network functions. The following options are available for network function lifecycle operations:

- **Upgrade**: The update function updates the entire NF or NS to a new catalog version. This could be used when performing minor updates to a CNF where only a single helm chart component is changed.

  In Telco Cloud Automation 2.3, this model supports adding and removing VDUs (individual HELM charts) from the Network Descriptor.

  Upgrades and updates depend on a newer revision of the CSAR. A new CSAR with the corresponding updates (such as helm charts and release numbers) is supplied. If the Vendor and Product name match, the newer CSARs are available for selection from the catalog during the NF upgrade processes.

- **Upgrade Package**: Upgrade package updates an instantiated NF to the new catalog version, without making any changes to the application.

  The upgrade process links an existing instantiated NF to an updated version of the catalog entry for that NF. The process then allows new workflows that are present in the new catalog to be run. This model can be beneficial in upgrade cases where workflows or migration are necessary before the upgrade.

User-Plane and RAN CNF Workload Considerations

The Telco cloud supports both control plane functions, such as SMF and AMF, and user-plane functions such as the DU, UPF, and so on.

The main considerations for deploying user plane functions include NUMA Alignment, CPU Pinning, and use of SR-IOV.

Telco Cloud Automation supports multiple options for NUMA Alignment and CPU pinning configurations that can be leveraged to meet the requirements of a network function.

- **NUMA Alignment**: This configuration option ensures that NICs, Memory, and CPU are aligned. Also, if this option is used without any other options, it ensures that CPUs are pinned in the format of pCore + Hyperthread and exclusive affinity is granted to these pinned CPUs.

  This implies that a 20vCPU Tanzu Kubernetes Grid VM consumes 10 physical cores and 10 hyperthread. This pinning is static and determined by the VM Operator. This option also reserves 100% of CPU and Memory.

- **Latency Sensitivity**: By setting Latency Sensitivity to HIGH, you can adjust the way the ESXi schedules the VM. In this case, pinning is achieved by ESXi without the need for static pinning.

This implies that a 20vCPU Tanzu Kubernetes Grid VM consumes 20 Physical cores. When LS is set to High, scheduling on the Physical Core sibling hyperthread is prohibited. To achieve this behavior, 100% reservation of CPU must be configured on the VM by the Telco Cloud Automation platform.

Note NUMA Alignment and Latency Sensitivity can be configured at the same time. CPU Pinning is performed based the Latency Sensitivity option, which means the vCPUs are scheduled only on physical cores and its associated hyper-thread are blocked for scheduling. The Latency Sensitivity option also ensures NUMA alignment.

As part of vSphere 8.0, Telco Cloud 3.0 introduces a new feature for SMT threading. This feature allows the CPU pinning to occur in the same way as with the NUMA alignment. However, rather than Telco Cloud Automation statically pinning vCPUs to logical cores, the ESXi scheduler ensures the correct placement and execution of cores.

Note As part of vSphere 8.0, the Virtual Hyperthreading (vHT) function is introduced in VM hardware version 20. This feature allows ESXi to dynamically provision Latency Sensitivity with hyper-threading enabled.

- vHT is an enhancement to the latency sensitivity high feature. With latency sensitivity set to high and vHT activated, specific applications benefit from hyperthreading awareness and achieve performance gains. This model helps prevent cache thrashing.
- Without vHT activated on ESXi, each virtual CPU (vCPU) is equivalent to a single non-hyperthreaded core available to the guest operating system. With vHT activated, each guest vCPU is treated as a single hyperthread of a virtual core (vCore).

For RAN workloads such as DU, the HW support option increases the VM hardware to the latest release available on the target vCenter. This feature ensures that additional real-time scheduling options are available when the RAN DU workload is run.

Tanzu Kubernetes Grid Design

This section outlines the considerations for creating Tanzu Kubernetes Grid clusters for Core and RAN network functions.

Tanzu Kubernetes Cluster Design

Tanzu Kubernetes clusters are deployed in the compute workload domains. The cluster design varies for each function such as Core, RAN, and so on.

The Telco Cloud Platform consumes resources from various compute workload domains. Resource pools within vSphere provide guaranteed resource availability to workloads. Resource pools are elastic; more resources can be added as the resource pool capacity grows.

The target endpoint for a Tanzu Kubernetes Grid is a vSphere resource pool. A resource pool is an abstraction of the resources available to a cluster. By default, each cluster has a root resource pool. Additional resource pools can be created as required.

A resource pool can map to a single vSphere cluster or stretch across multiple vSphere clusters. The stretch feature is not available without the use of a VIM such as VMware Cloud Director. Thus, in the context of Tanzu Kubernetes Grid cluster design, a resource pool is bound to a single cluster, or in the case of RAN cell-sites, a single host.

Note RAN Cell sites can consist of multiple hosts when high availability is a consideration. Most Cell Sites are a single node and redundancy is provided through the Radio network.

In the context of Tanzu Kubernetes Grid, each Kubernetes cluster is mapped to a Resource Pool. A resource pool can be dedicated to a Kubernetes cluster or shared across multiple clusters.

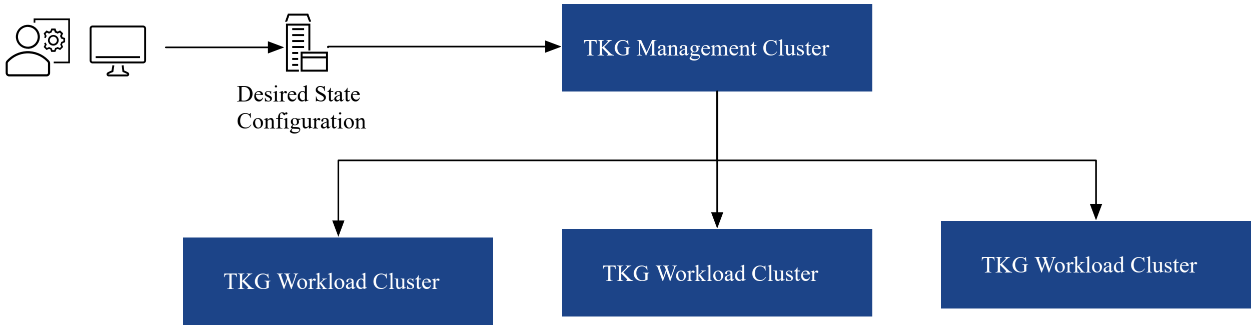
Resource pools allocate committed levels of resources to the worker nodes in a K8s environment. Creating multiple resource pools guarantees a specific level of response for a data-plane workload, while offering a reduced level for non-dataplane workloads.

Note In user plane workloads, the allocation of memory/CPU reservations and limits to a resource pool is separate from the Latency Sensitivity requirements. The reservation of the resource pool must be large enough to accommodate the sum of reservations required by the Tanzu Kubernetes Grid worker nodes.

Tanzu Kubernetes Grid uses two types of clusters:

- **Management cluster**: The management cluster is a Kubernetes cluster that is used to manage and operationalize the deployment and configuration of subsequent workload clusters. Cluster-API for vSphere (CAPV) runs in the management cluster. The management cluster is used to bootstrap the creation of new clusters and configure the shared and in-cluster services such as NFS client mounts within the workload domains.
- **Workload cluster**: The workload cluster is where the actual containerized network functions and PaaS components are deployed. Workload clusters can also be divided into multiple Node pools for more granular management.

Figure 4-43. Tanzu Kubernetes Cluster Types



Cluster & CNF Mapping Design

When creating Tanzu Kubernetes Clusters, note that considerations for Kubernetes in a virtualized environment differs from Kubernetes in a bare-metal environment.

In bare metal environments, the unit of scale is a bare-metal server and the placement of many pods or workloads on each server becomes important. However, in a virtualized environment, the unit of scale is at the individual VM level, which can lead to a more efficient design of Kubernetes clusters.

Within Telco Cloud Automation, the worker nodes in an individual Tanzu Kubernetes Grid cluster can be grouped into logical units called node pools.

Tanzu Kubernetes Grid - Node Pools

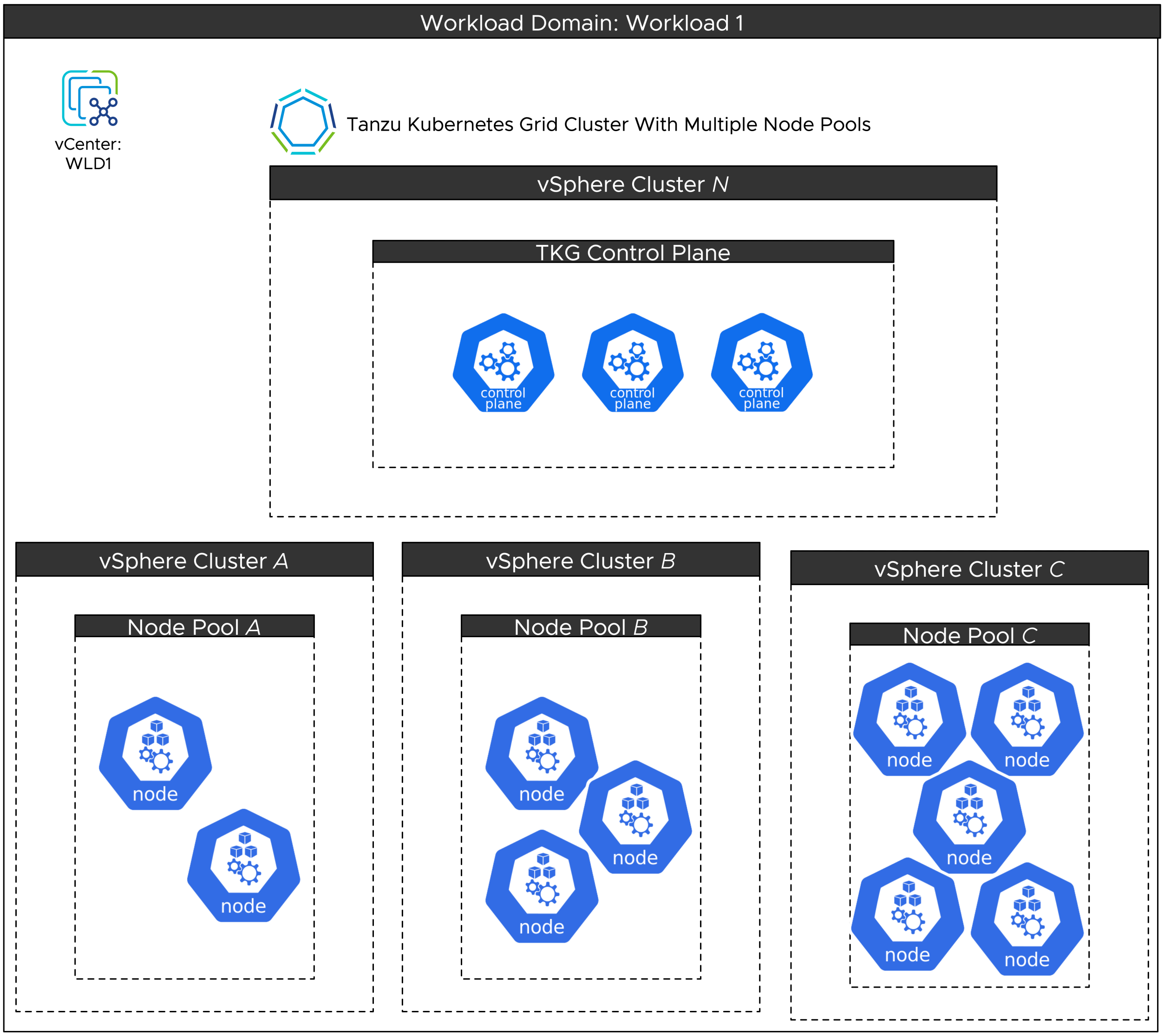
A node pool is a set of worker nodes that have a common configuration set (vCPUs, Memory, Networks, K8s Node Labels) and are thus grouped as a pool.

In the Telco Cloud Automation CaaS deployment, multiple worker node pools can be created in a Kubernetes cluster. Each node pool can have its own unique configuration, along with a replica count.

From a deployment perspective, each node pool must belong to a vSphere cluster or unique host. If required, each node pool in a single Kubernetes cluster can be deployed to different vSphere clusters or hosts for redundancy or load-sharing. In addition, multiple independent Kubernetes clusters can share the same vSphere clusters (resource allocation permitting).

Based on the node pool concept, a Kubernetes cluster deployed to the Telco Cloud can have multiple node pools. Each node-pool within a single vCenter can be deployed to a different cluster or even to an individual ESXi host. However, workers within a single node pool cannot be distributed across multiple vSphere clusters or hosts.

Figure 4-44. Tanzu Kubernetes Cluster with multiple nodepools



If you create a Tanzu Kubernetes cluster where node pools are deployed to different target endpoints (within a single vCenter), it is called a Stretched Cluster. In this scenario, multiple vSphere endpoints are used as target endpoints for different nodepools within a single cluster.

Note When using stretched cluster designs, storage zoning is required if the datastores on each vSphere cluster are unique (as with a vSAN design). This ensures that node-pools within a resource pool on a vSphere Cluster or standalone host can consume the appropriate storage for any persistent volume claims the cluster might need.

The Tanzu Kubernetes Management cluster is typically deployed within the same vCenter as the workload clusters. However, it is not mandatory. The management and workload clusters can also be on different vCenter Servers but it requires multiple TCA-CP nodes, one for the management cluster and another for the workload clusters.

Each worker node in Kubernetes can be assigned a set of node labels. The Kubernetes scheduling engine uses these free-form labels to determine the proper placement of Pods, distribute traffic between services, and so on.

The labels are simple key/value pairs. For example,

```
vendor: vendor
environment: production
function: function_name
```

Note Encoding a naming convention into node labels is not necessary. RegExp is not supported for node selectors.

These labels are leveraged by the Network Function to ensure that pods are scheduled to the correct worker nodes. A node can have multiple labels. If the node selector matches only one of these labels, it can be considered a viable target for scheduling.

Different node pools can have different labels associated. However, the same label set (all Key/Value pairs) cannot be applied to multiple node pools.

Tanzu Kubernetes Cluster Design Options

When you design a Tanzu Kubernetes cluster for workloads with high availability, consider the following:

- Number of failed nodes to be tolerated
- Additional capacity for cluster growth
- Number of nodes available after a failure
- Remaining capacity to reschedule pods of the failed node
- Remaining Pod density after rescheduling
- Individual Worker node sizing
- Life-cycle management considerations (both CaaS and CNF)
- Dynamic Infrastructure Provisioning requirements
- Platform Awareness features (such as NUMA)

By leveraging the Nodepools concept, Kubernetes clusters can be designed and built in various ways. Some of the common design options include:

- A single Kubernetes cluster that hosts multiple CNFs from multiple vendors
  - The single cluster design resembles metal-based deployments, where the VMs consume large amounts of CPU and memory resources. This model has less overhead than other models but brings complexity in terms of scale and lifecycle management.
  - In this model, a Network Function upgrade requires the K8s upgrade and the entire cluster upgrade, leading to application interoperability issues.

  - Different network functions require different network connectivity and different configurations, so providing unique infrastructure requirements to each network function becomes complicated and challenging to manage in a single cluster environment.
- A Kubernetes cluster per Vendor that hosts all the relevant CNFs from a given vendor
  - The per Vendor model solves some of the single cluster challenges. In this model, network functions can be managed across different node pools, and each network function can have different infrastructure requirements.
  - This model faces LCM concerns, especially if newer functions require different versions of Kubernetes.
- A Kubernetes cluster per CNF
  - This model provides the maximum flexibility. In this model, a Tanzu Kubernetes Grid cluster is created for each Network Function, which simplifies the challenges of providing unique infrastructure requirements per network function. This model also removes the common challenge of lifecycle management and provides the most efficient scaling mechanism.

Tanzu Kubernetes Cluster IP Addressing

Tanzu Kubernetes Grid leverages kube-vip to provide a highly available IP address for ingress requests to the Kubernetes API. The kube-VIP address is statically assigned but needs to come from the same subnet as the DHCP scope for the control-plane and worker nodes.

Note When using Tanzu Kubernetes Grid outside of VMware Telco Cloud Automation the NSX Advanced Load Balancer can be used to provide the kube-vip address. However, this feature is not currently implemented in VMware Telco Cloud Automation

When allocating IP addresses space for Tanzu Kubernetes Grid, note the following considerations:

- DHCP address space configuration for all control-plane and worker nodes
- A statically assigned address for the kube-vip address
- Space for scale-out and upgrade of worker nodes.

Ensure that the DHCP addresses allocated to the control-plane nodes are converted to static reservations to avoid any potential changes to the control-node IP addresses.

Tanzu Kubernetes Grid VM Dimensioning Considerations

A node pool is a set of worker nodes that have a common configuration set (vCPUs, Memory, Networks, Labels) and are thus grouped as a pool. The node pool concept simplifies the CNF placement when the CNF has many different components.

When sizing a worker node, consider the number of Pods per node. Most of the kubelet utilization is contributed by ensuring that Kubernetes pods are running and by constant reporting of pod status to the Control node. High pod counts lead to high kubelet utilization. Kubernetes official documentation recommends limiting Pods per node to less than 100.

Other workloads require dedicated CPU, memory, and vNICs to ensure throughput, latency, or jitter. When considering the number of pods per node for high-performance pods, use the hardware resource limits as the design criteria. For example, if a data plane intensive workload requires a dedicated passthrough vNIC, the total number of Pods per worker node is limited by the total number of available vNICs.

To align with metal provisioning of Kubernetes and other models, TKG worker nodes are sized as large VMs, often consuming half of, or even an entire socket. While this design is viable, you can leverage some considerations to create a more optimized infrastructure when dealing with VM-based Kubernetes platforms such as those deployed using Telco Cloud Automation.

Considerations for a sample NF:

	CNF Component	# VCPUs	# RAM	# Replicas
Sample NF	FE	8	32	2
	SIP	8	32	2
	BE	8	32	2
	RCS	8	32	2
	SMx	4	16	3
	DB	8	32	3
	PE	8	32	1

Note This sample NF is only for illustrative purposes and is not intended for any vendor specific NF.

The worker node sizing for this NF require 52 vCPU and 208 GB of RAM, which can create complications from placement/scheduling if NUMA Alignment or Latency Sensitivity are required. This model of creating a single worker node also complicates issues if there are components that require specific infrastructure requirements that other components do not.

When large worker nodes are created, the unit of scale is the size of the worker node template. Thus, if the SMx scale increased from 3 Replicas to 4, a new worker node of 52 CPUs is required (assuming the SMx component uses Pod anti-affinity rules).

An alternate and more flexible approach is to have multiple, smaller worker nodes (distributed across multiple node pools), deployed across a Tanzu Kubernetes stretched cluster.

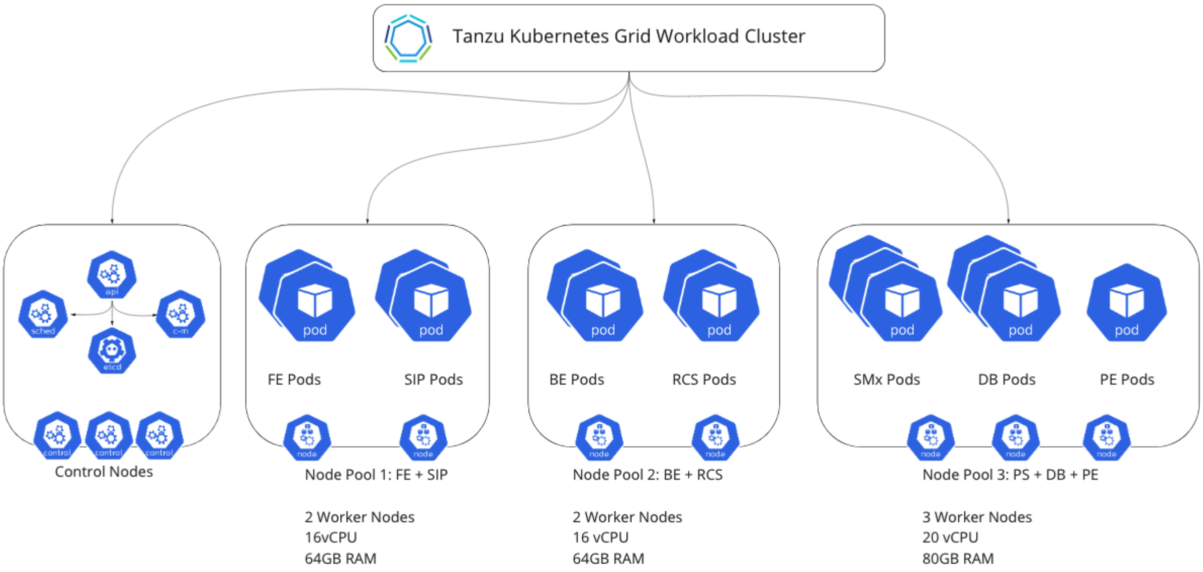
With this architecture, specific co-dependent functions that frequently communicate can be collapsed into smaller worker nodes. For example, FE and SIP are combined into a single node pool, BE and RCS are combined into another node pool, and so on.

The distribution of those components that are co-dependent needs to be determined by the CNF Vendor. Elements such as pod-pod communication may no longer reside within a single node and therefore may incur network latencies, as traffic is sent from hypervisor hosts to different VMs within the environment.

This method has the advantage of keeping the worker nodes smaller in size and thus simpler and easier to scale. It also allows for different configuration at the node-pool level. For example, one component requiring 1 GB huge pages and multiple interfaces/SRIOV.

With no standard design rules for building these clusters, all involved parties must discuss and determine the optimal design. The node selector configuration must be modified in the HELM chart to support the targeted placement of pods.

Figure 4-45. Alternate Cluster Dimensioning



This alternate cluster model creates multiple node pools within a single cluster. The target deployment point for the Network Function is still a single cluster but can be scheduled to the correct worker nodes based on the node pool labeling pods. This model is advantageous over a bare-metal type approach where a single node pool is created with large worker nodes because scale-out can now occur on a per node-pool level. This model makes scaling easier to manage and helps prevent resource waste by dividing worker nodes into smaller, more composable units.

Tanzu Kubernetes Grid Cluster Design Recommendations

Design Recommendation	Design Justification	Design Implication
Map the Tanzu Kubernetes clusters to the vSphere Resource Pool in the compute workload domain.	Enables Resource Guarantee and Resource Isolation	During resource contention, workloads can be starved for resources and can experience performance degradation. **Note**: You must proactively perform monitoring and capacity management and add the capacity before the contention occurs.
Create dedicated DHCP IP subnet pools for the Tanzu Kubernetes cluster management network. Allocate a Static IP for Kubernetes Endpoint API.	Simplifies the IP address assignment to Kubernetes clusters. Use static reservations to reserve IP addresses in the DHCP pool for Kube-Vip address.	DHCP servers must be monitored for availability. Address scopes are not overlapping IP addresses that are being used.
Ensure the DHCP pool has enough address space to manage upgrades and LCM events	Ensure that new nodes can obtain an IP address	Requires the scope to be larger than the actual cluster size. Short-term DHCP leases can be used to offset this; however, this is not recommended.
Place the Kubernetes cluster management interface on a virtual network, which is routable to the management network for vSphere, Harbor, and Airgap mirror.	Provides connectivity to the vSphere infrastructure. Simplifies the network design and reduces the network complexity.	Increases the network address management overhead. Increased security configuration to allow traffic between the resource and management domains.
Leverage a per-CNF model for cluster deployments	Simplifies the lifecycle management and cluster sizing to a per workload basis	Creates additional resource requirements for control-plane nodes. Can use a single cluster if all products are tightly couple and from a single vendor.
Separate control-plane and user-plane functions at both the TKG and vSphere levels	Isolates different workload types Easier resource management and reduced contention	Requires different vSphere clusters for different workload types.

Tanzu Kubernetes Grid Add-on Framework

The cluster add-on framework was introduced as part of VMware Telco Cloud Platform 2.5. The configuration and additional elements of the management and workload clusters are delivered through the add-on framework.

The add-on framework moves some of the cluster configuration options into a modular framework. The modular framework can be used not only for generic cluster elements but to support an increasing number of the Tanzu Kubernetes Grid CLI managed packages.

The add-ons are categorized as follows:

- **Container Networking Interface (CNI) add-ons**: Antrea and Calico. These primary CNI add-ons are selected during the cluster creation.

- **Container Storage Interface (CSI) add-ons**: vSphere-CSI and NFS Client.
- **Monitoring add-ons** : Prometheus and Fluent-bit. These add-ons are used for metric and syslog collection, and they can be added to a workload cluster.
- **Networking add-ons**: Multus, Avi Kubernetes Operator (AKO), and Whereabouts.
- **System add-ons**: System settings (cluster password and generic Syslog configuration), the partner harbor system connectivity, and cert-manager.
- **TCA-Core add-on**: nodeconfig operator. This add-on is deployed automatically as part of Telco Cloud Automation.
- **Tools add-ons**: HELM (v2) and Velero backup frameworks.

Telco Cloud 3.0 also supports native deployment of any of the Tanzu Kubernetes Grid through Telco Cloud Automation. However, the recommended approach is to use the add-on framework.

Prometheus Add-On

Prometheus is a monitoring and alerting platform for Kubernetes. It collects and stores metrics as time-series data. As part of the Prometheus deployment, cadvisor, kube-state-metrics, node exporters, and the Prometheus server components are deployed into the workload cluster.

When deploying Prometheus, an additional Custom Resource (CR) can be applied. The default configuration deploys Prometheus with a service type of clusterip and a PVC of 150 GB for metric retention. The default Prometheus configuration from the Tanzu Kubernetes add-on framework is deployed through the custom resource. The default configuration can be modified as required. For more information about the Prometheus deployment and configuration options, see Prometheus Configuration.

Note Prometheus provides the collected metrics for an upstream platform such as Aria Operations to consume. A recommended integration configuration is provided to capture and translate metrics into formats that Aria Operations can understand.

If you want to modify the integration configuration, contact your local VMware representative.

Fluent-Bit Add-On

Fluent-bit is a lightweight logging processor for Kubernetes. You can deploy fluent-bit through the add-on framework to forward logging information to an external syslog or the Security Information and Event Management (SIEM) platform.

Similar to other add-ons, the fluent-bit deployment uses an additional Custom Resource (CR) for its configuration. Specific fluent-bit configuration is required for the appropriate level of logging at the cluster level.

Note As with Prometheus, a recommended configuration for integrating Fluent-Bit with Aria Operations for Logs is introduced in Telco Cloud 3.0.

For more information about the Fluent-bit configuration options, see Fluent-bit Configuration. For more information or additions related to integrating fluent-bit with Aria Operations for Logs, contact your local VMware representative.

Whereabouts Add-On

Whereabouts is an IP Address Management (IPAM) CNI plugin. It is used with Multus to manage the IP address assignment to secondary pod interfaces in a cluster-wide configuration.

Whereabouts does not require configuration from the add-on framework Custom Resource (CR) screen. After the add-on is deployed, the NF must create a Network Attachment Definition with the IPAM type set to 'whereabouts'. The network definition can then be consumed through the pod or deployment specification.

For more information about Whereabouts consumption, see Multus and Whereabouts deployment.

Cert-Manager Add-On

cert-manager is an x.509 certificate controller for Kubernetes environments. It allows certificates or certificate issuers to be added as objects or resources within the Kubernetes cluster.

Cert-manager supports namespaced (Issuer) or cluster-wide (ClusterIssuers) configurations. Certificates can be self-signed, CA signed, or integrated with external issuers. For more information about Cert-Manager deployment, see Cert-Manager Installation.

Note The default cert-manager deployment does not create any issuers or clusterissuers. Configure the issuers after deploying cert-manager. The configuration varies depending on the customer and application requirements.

Velero Add-On

Velero provides backup capabilities for Kubernetes. Velero administrators can perform backups of Kubernetes namespaces including any PVs, which can be restored upon a failure event.

The backups can be restored to the same cluster or to a new cluster. PVs are typically restored to the same cluster.

To restore the backup to a new cluster, you must first create a clone of the original (failed / deleted) cluster and then use Velero to restore the backup into the new cluster. A remediation option must be run on the NF that was instantiated to the original cluster to remediate the NF against the new cluster.

The remediation reconfigures any Dynamic Infrastructure Provisioning (also known as Late-Binding) that is implemented as part of the NF deployment.

Cloud Native Networking Design

5G CNFs require advanced networking services to support receive and transmit at high speed with low latency. Advanced network capabilities must be achieved without deviating from the default networking abstraction provided by Kubernetes.

The Container Network Interface (CNI) supports the networking constructs within a Kubernetes cluster. Aside from each worker node having a general management IP, there are requirements for pod-pod communications.

The Cloud Native Networking design focuses on supporting multiple NICs in a Pod, where the primary NIC is allocated to Kubernetes management, and attaching additional networks for data forwarding.

The CNI option allows the extension of the traditional single interface model of Kubernetes to support multiple interfaces (also known as MULTUS). This option allows the separation of user-plane and control-plane traffic and provides service isolation by virtue of leveraging multiple external interfaces.

In case of 5G workloads, the main interfaces are managed by the primary CNI. Additional network attachments created using the secondary CNI are bound to SR-IOV or EDP port-groups created within NSX.

TCA lets you choose a CNI that is configured and deployed as part of the cluster creation. Typically, even in control plane CNFs, the secondary interface (MULTUS) is used with network types. This is to provide logical separation at the worker node level for different interfaces. If each interface is required to terminate into its respective VRF or isolate L3 domain (such as signaling and O&M) on the DC Fabric/MPLS/Transport network, use MULTUS to attach multiple interfaces to a worker node.

Interface	5G Control Workloads	5G User-Plane Workloads
Primary CNI (Calico / Antrea)	Required	Required
Secondary CNI (MULTUS / MACVLAN / IPVLAN / SRIOV)	Not required, but usually leveraged for traffic separation	Required to provide high throughput and complex connectivity requirements

When creating additional interfaces to the container, different network types can be configured. The most common are MACVLAN and IPVLAN.

- **MACVLAN Interfaces**: MACVLAN is used to create multiple virtual network interfaces behind the host’s single physical interface. Interfaces of MACVLAN type create a secondary attachment to the kubernetes pod. With MACVLAN, each interface is allocated with a unique MAC address.

- **IPVLAN**: IPVLAN is used in a similar way to create multiple virtual network interfaces to a single physical interface. However, with IPVLAN, all the secondary attachments to the pod have a common MAC address inherited from the physical interface MAC address.

Telco Cloud Automation can add secondary network interfaces (SR-IOV and VMXNET3) through Dynamic infrastructure Provisioning. Regular non-SRIOV or EDP interfaces can be added as part of the cluster creation.

This implies that infrastructure admins are not obliged to design the CAAS secondary networks in advance, during the cluster creation process. The Dynamic Infrastructure Provisioning features of the Telco Cloud allow the Network Function Cloud Service Archive (CSAR) to be customized to add Secondary interfaces to worker nodes. This is achieved through enabling the Multus CNI on those interfaces during the onboarding / instantiation process.

Within the CSAR, you can add a new network adapter of type VMXNET3 or SRIOV, name the Multus Interface, and attach it to the appropriate network resource from the available vSphere resources.

The Telco Cloud platform provides two options for the Primary CNI: Calico or Antrea. The Tanzu Kubernetes Grid Management cluster is always deployed with Antrea as the primary CNI. However, the choice of primary CNI exists when creating the workload cluster.

Note Once the primary CNI choice is made and the cluster is deployed, you cannot change the CNI.

Table 4-18. Primary CNI Deployment Options

Endpoint	Description	Antrea	Calico
Pod Connectivity	Container network interface for pods	Uses Open vSwitch	Uses Linux bridge with BGP
Service: ClusterIP	Default k8s service type accessible from within the cluster	Supported	Supported
Service: NodePort	Allows external access through a port exposed on the worker node	Supported	Supported
Service: LoadBalancer	Leverage a L4 load-balancer to distribute traffic across pods	Provided externally to the CNI, typically through NSX Advanced Load Balancer, HAProxy, MetalLB	
Ingress Service	Routing for inbound pod traffic	Provided externally to the CNI typically through the Avi Kubernetes Operator (provided by the NSX Advanced Load Balancer or Contour)	
Network Policy	Controls ingress and egress traffic	Open vSwitch based	IP tables based
NSX Integration	Connectivity to NSX for administrator defined security policies	Supported	Not supported

Design recommendations for Cloud Native networking depends on various factors. The most common consideration is related to the CNI that is tested by the function vendor.

Note Do not modify the primary CNI configurations. The eBGP function provided by Calico is currently not supported.

It is possible to mix and match across clusters, some clusters can be deployed with Antrea as the Primary CNI and others can use Calico, the primary CNI cannot be mixed across nodepools within the same cluster.

Cloud Native Egress Considerations

The CNF Egress communication requirements are important. The two main considerations for egress networking include:

- **Multus for Egress**: Multus CNI enables the attachment of multiple network interfaces to pods. The Multus CNI plugin is supported with Tanzu Kubernetes Grid. VMware TCA orchestrates the cluster with all required resources to run Multus as an additional CNI.

  The network attachment definition file is used to setup the network attachment for the pod. The CNF vendor must create those files using CNI Custom Resources (CRs) as needed by the application.

- **WorkerNode primary interface**: Pods can share the worker node primary interface. In this case, Kubernetes manages Source Network Address Translation (SNAT) between Pods and Worker Nodes.

  An external security platform such as VMware NSX may also be required if SNAT is required along with multiple VRFs. In this scenario, the NAT rules can be based on the destination networks the packet is heading to. A specific NAT pool is required for each destination traffic type.

  With this option, overlapping networks within the VRF are not supported as the SNAT cannot distinguish between the different destination endpoints.

The recommended egress design is dependent on the overall CNF networking requirements and design. Multus for Egress is a good design consideration as it provides traffic isolation and allows multi-homed pods, this in turn simplifies networking configuration and operations.

Cloud Native Networking Recommendations

Design Recommendation	Design Justification	Design Implication
Work with the function vendor and all parties to determine the preferred Primary CNI for the network function.	Determines if Antrea or Calico has been validated	Impacts the choice of primary CNI.
Leverage MULTUS as the secondary CNI only for functions or architectures that require it	MULTUS is used to provide additional interfaces. Not all applications require multus.	This may impact the network topology of how secondary interfaces connect to the network and how network ingress/egress routing is configured.
Do not change the default configuration of the primary CNI.	This may invalidate support and cause networking issues within the cluster.	The cluster networking capabilities are defined by the deployed CNI versions and the default configuration.

Cloud Native Networking - Load Balancer and Ingress Design

In a K8S environment, external services are exposed with Ingress connected to the Load Balancer.

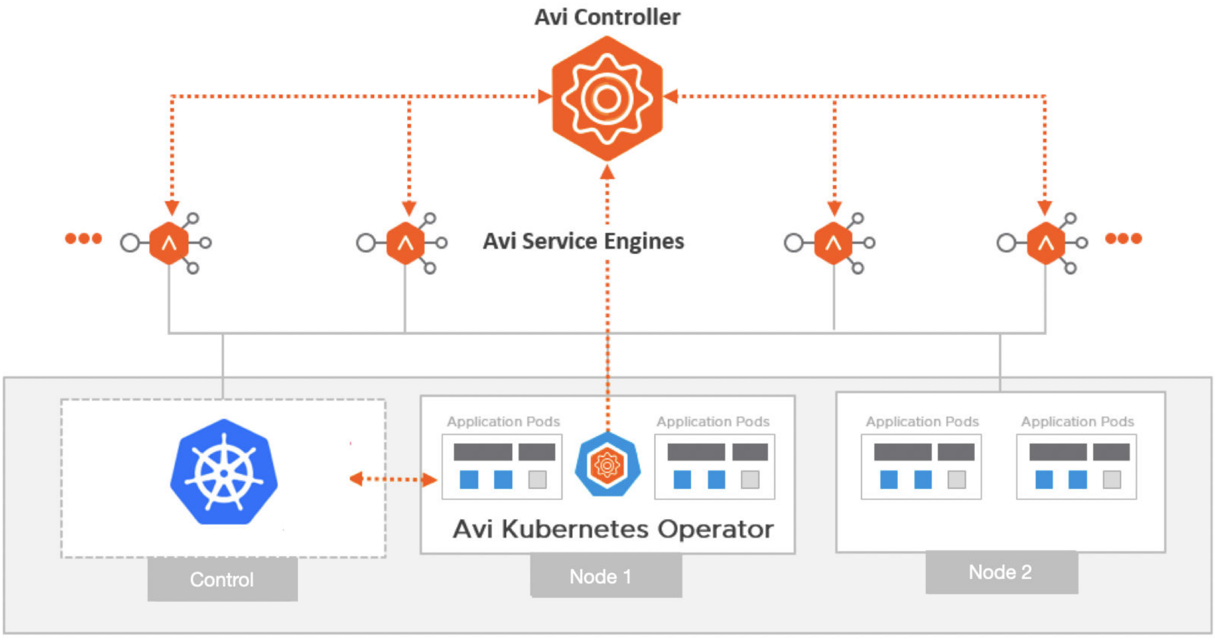
Avi Kubernetes Operator

The Avi Kubernetes Operator (AKO) is a Kubernetes operator that works as an Ingress Controller and performs Avi-specific functions in a Kubernetes environment with the NSX ALB Controller.

AKO is a TKG Cluster Add-On that manages the load balancer functionality for Kubernetes clusters by listening to Ingress, Service type Load Balancer, and Service v2 API objects (GatewayClass and Gateway).

The AKO can be deployed directly from TCA add on or as a CSAR using AKO Helm Chart. The first option provides a fixed version of AKO as it is connected to a TKG release, while the second option provides more flexibility to consume new AKO releases.

Figure 4-46. Avi Kubernetes Operator architecture



AKO is deployed as one or more pods in each Kubernetes cluster and it monitors the Kube API server for any Load balancer service request. AKO requires to have a unique namespace in each Kubernetes cluster.

In addition to AKO, NSX ALB requires Kubernetes objects such as gateways, ingress class, and custom resources in each cluster to integrate NSX ALB with TKG.

Note The Avi Infra setting (Custom Resource) must be created.

The following core parameters are included in the Avi Infra setting custom resource definition:

- **Avi Service Engine group**: Defines the NSX ALB Service Engine group to be used for this cluster
- **Default fallback VIP network**: Defines the fallback Virtual IP network to be used when Virtual IP is not provided by the user; IPv4, IPv6, and dual stack are available.
- **BGP label**: Defines the BGP label associated with specific BGP peer towards the uplink router.

AKO creates the NSX ALB pools with Kubernetes pods as pool members and uses the Kubernetes service's backend port to send the traffic directly. NSX ALB monitors the health of the pod and performs synchronization if changes are triggered.

CNFs can consume NSX ALB as an external load balancer using Labels or Annotations:

- **Annotations**: In this approach, the CNF uses standard Kubernetes service creation to request external load balancer service by adding annotations to the service. This approach can only be used if the load balancer has a single service (protocol/port) requirement.

- **Labels**: In this approach, the CNF uses standard Kubernetes service creation to request external load balancer service by adding labels to the service. This approach can be used both for single service (protocol/port) or multi-service load balancer, where a single VIP is shared across multiple ports/protocols).

Cloud Native Storage Design

The Cloud Native storage design section includes design considerations for stateful workloads that require persistent storage. The vSAN design forms the basis for the Cloud Native Storage design.

In Kubernetes, a Volume is a directory on a disk that is accessible by the containers in a pod. Kubernetes supports many types of volumes. The Cloud Native storage design focuses on the vSAN storage design required to support dynamic volume provisioning. This design does not address different ways to present a volume to a stateful application.

The Telco Cloud Platform vSAN storage design provides the basis for container storage and has the following benefits:

- Optimizes the storage design to meet the diverse needs of applications, services, administrators, and users.
- Strategically aligns business applications and the storage infrastructure to reduce costs, boost performance, improve availability, provide security, and enhance functionality.
- Provides multiple tiers of storage to match application data access to application requirements.
- Designs each tier of storage with different performance, capacity, and availability characteristics. Not every application requires expensive, high-performance, highly available storage, so designing different storage tiers reduces cost.

vSAN storage policies define storage requirements for your storage. Cloud Native persistent storage or volume (PV) inherits performance and availability characteristics from the vSAN storage policy.

Kubernetes admins uses Kubernetes StorageClass objects to describe the storage “classes” available for a Tanzu Kubernetes cluster. Different StorageClasses can map to different vSAN storage policies.

Note

- By default, only a single vSAN storageclass is created on the Tanzu Kubernetes clusters that consume the vSAN default storage policy. Platform owners can create and expose different storage classes with storage policies as required.
- While this design uses vSAN, any supported storage solution that meets the characteristics of this storage design can be used. For best practices, see the vendor documentation.

Cloud Native Storage Access Modes

Cloud Native persistent storage or volume in Kubernetes is mounted with a certain access mode. Three possible access modes are as follows:

Access Mode	CLI Abbreviation	Description
ReadWriteOnce	RWO	The volume can be mounted as read-write by a single node.
ReadOnlyMany	ROX	The volume can be mounted read-only by many nodes.
ReadWriteMany	RWX	The volume can be mounted as read-write by many nodes.

RWO is a common access mode for cloud native Stateful workloads. RWO volumes have a 1:1 relation to a Pod in general.

RWX volumes provide storage shared by multiple Pods with all Pods able to write to it. The difference between RWO and RWX relates to mounting the same filesystem on multiple hosts, which requires support for features such as distributed locking.

With vSAN 7.0 and lower versions, the vSphere Cloud Storage Interface (CSI) based driver provisions only block-based Persistent Volume, which aligns to RWO storage access. In this scenario, a VMDK is created and attached to the TKG VM.

RWX can be accomplished using external NFS, or with vSAN 7.0 and later versions. By enabling the vSAN File Service feature, a single vSAN cluster can be used for both RWO and RWX support. A Container File Volume (CFV) is created within vSAN File Services when a RWX volume is created using the vSphere CSI driver (that is backed by a vSAN datastore).

RAN Timing Considerations

This sections describes PTP considerations for Telco Cloud RAN deployments.

PTP Design Considerations

Precision Time Protocol (PTP) delivers time synchronization in various Telco applications and environments. It is defined in the IEEE 1588-2008 standard. PTP helps issuing accurate time and frequency over telecommunication mobile networks. Precise timekeeping is a key attribute for telco applications. It allows these applications to accurately construct the precise sequence of events that occurred or occur in real time. So, each DU in the Telco Cloud RAN must be time-synchronized.

For more information about the clocking models (LLS-C1 through LLS-C4), see the Telco Cloud - RAN Domains section.

PTP is leveraged in the RAN to provide consistent clocking between the DU and the RU. This can be configured in various ways depending on the RAN clocking requirements for LLS-C3. The following configurations can be automated through Telco Cloud Automation.

- **PCI-Pass Through PTP**: In this model, a dedicated port from the Network Interface Card is used to provide clocking from the cell site router into the DU network function.
- **VF-based PTP**: In this model, an SR-IOV Virtual Function (VF) is used to provide clocking between the cell site router and the DU network function. The VF-based model does not require a dedicated port. Instead, the same physical port used for the fronthaul traffic is used for the PTP clocking.

LLS-C1 is also supported. This model depends on the exact hardware deployed at the cell site. Some NICs support an on-board GNSS, which is connected to the external Global-Positioning-System antenna. In this mode, the cell site router is not involved in the clocking path. The RUs are connected directly to the physical server and PTP is provided directly from the DU to the RU.

Note As highlighted, support for LLS-C1 is dependent on the vendor hardware deployed at the cell site. Some Network Interface Cards have onboard GNSS capabilities, while other Network Interface cards do not have the onboard GNSS and require additional external components for clocking.

PTP Notification

The O-RAN working group 6 (Cloudification and Orchestration Workgroup) includes an O-Cloud Notification API specification for event consumers. Telco Cloud Platform RAN incorporates a model to support this specification.

The notification model provides a way for a consumer (CNF) to subscribe to status and event updates from the O-Cloud environment.

The CNF can subscribe to notifications through a sidecar co-located in the same pod as the DU Worker or through a local REST API. The CNF can query, register, and receive notifications through a callback API specified during the CNF registration.

The initial instantiation of the Cloud API notification framework provides notification for PTP status events. Rather than each RAN NF vendor incorporating multiple mechanisms to support multiple timing implementations, this notification framework provides a REST API for the NF vendor to subscribe and receive PTP synchronization events. The Cloud API Notification framework monitors the PTP status (PTP Sync Status and PTP Lock Status) and delivers notifications through a message bus to consumer applications.

Design Recommendation	Design Justification	Design Implication
If the consumer CNF (DU) requires event notifications, deploy and integrate the sidecar with the CNF.	Provides a way for the notification (PTP status changes) to be communicated by the O-Cloud to event consumers.	Requires integration between the CNF (DU) and the sidecar applications.
If the consumer CNF (DU) requires event notifications, the O-Cloud API CSAR must also be deployed.	Provides a mechanism for the O-Cloud to monitor PTP and communicate to the sidecar.	Increases the overall CNF count deployed The increased CNF count must be considered when designing to scale.
Leverage a clocking solution based on the RAN requirements for LLS-C1 or LLS-C3.	Ensure proper clocking is configured between DU and RU	LLS-C3 is easier to implement, but some deployments might require LLS-C1 which in turn requires additional hardware support.

Telco Cloud Operations Design

The telco cloud includes optional components such as Aria Operations and Aria Operations for Logs that form the operations tier in the Telco Cloud. This section provides guidance on the main design elements such as sizing, networking, and diagnostics.

Aria Operations

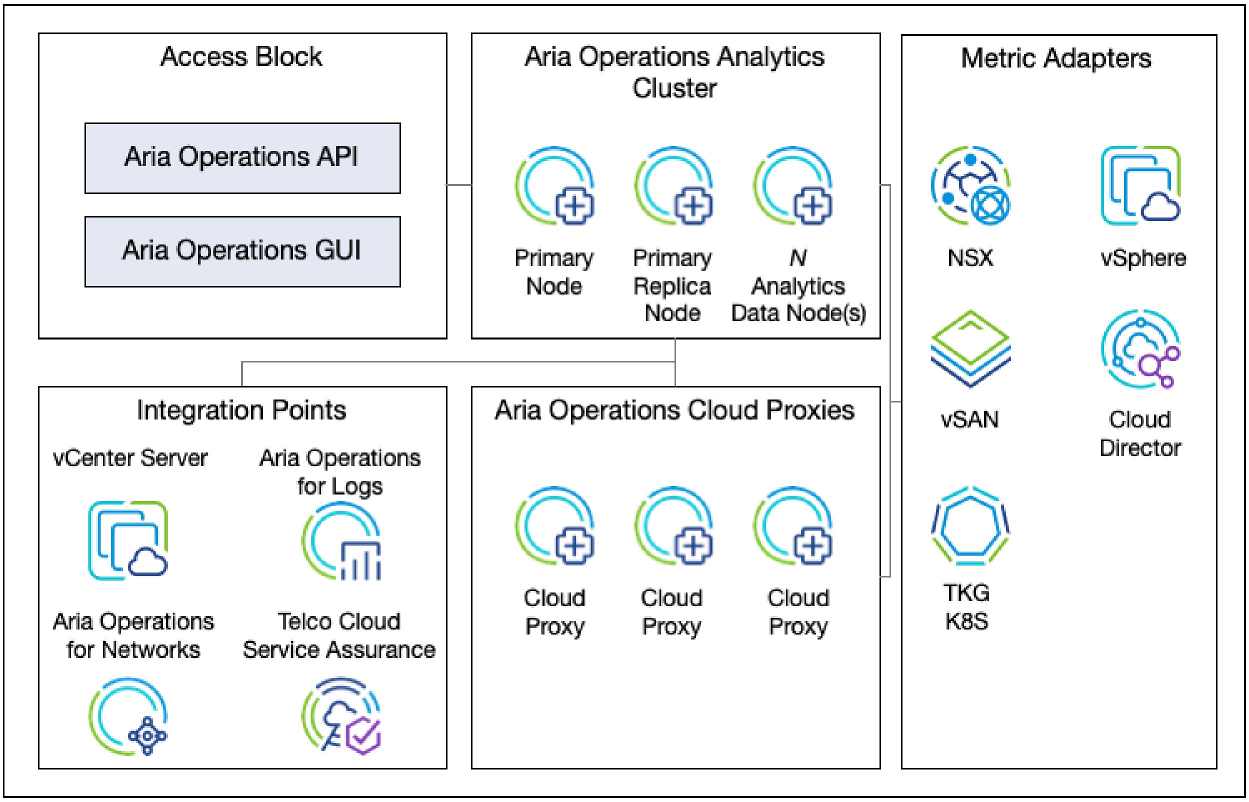
Aria Operations communicates with all management components to collect metrics that are presented through various dashboards and views. Aria Operations can collect metrics from various VMware and non-VMware products, including vSphere, vSAN, NSX, and Kubernetes clusters.

Aria Operations - Logical Design

Aria Operations is a single instance of a multi-node analytics cluster that is deployed in the management cluster. A primary node and primary replica node are created for an Aria Operations HA deployment. More analytic nodes can be added to scale the deployment.

In addition to analytic nodes, cloud proxies (formerly called remote collectors) can be deployed. These cloud proxies allow the distributed deployment of Aria Operations.

Figure 4-47. Logical Design components for Aria Operations



The analytics cluster of the Aria Operations deployment contains the nodes that analyze and store data from the monitored components. The deployment configuration (number and sizing of nodes) for the analytics cluster must be sized to meet the requirements for monitoring based on the number of VMs, objects, and metrics.

Aria Operations integrates with other platforms in the Monitoring / Observability framework of the Telco Cloud to exchange data with platforms. One example is the in-place capabilities of Aria Operations for Logs that allows syslog messages pertaining to a specific event to be viewed seamlessly from Aria Operations.

Aria for operations is a highly scalable platform. The analytics cluster must be sized based on the size of the environment. For more information about dimensioning guidelines for Aria Operations, see Aria Operations Sizing Guidelines.

This design uses medium-size nodes for the analytics cluster and standard-size nodes for the cloud proxies. To collect the required number of metrics, add a virtual disk of 1 TB (or as required) on each analytics cluster node.

The latency requirements between analytics nodes is 5ms. All analytics nodes must be deployed on the same segment and within a single datacenter. In addition, wherever possible, the deployment of Aria Operations nodes must be sized to fit into a single NUMA to provide better performance.

Note In addition to the Highly Available (HA) mode, Aria Operations also supports a Continuous Availability mode. This mode requires additional functionality such as Fault-Domains in the management domain. The continuous availability mode is not covered in this reference architecture guide.

You can use the self-monitoring capability of Aria Operations to receive alerts about operational issues. Aria Operations displays the following administrative alerts:

- **System alert**: Indicates a failed component of the Aria Operations application.
- **Environment alert**: Indicates that Aria Operations stopped receiving data from one or more resources. This alert might indicate a problem with system resources or network infrastructure.
- **Log Insight log event**: Indicates that the infrastructure on which Aria Operations is running has low-level issues. You can also use the log events for root cause analysis.
- **Custom dashboard**: Aria Operations shows dashboard for data center monitoring, capacity trends, and single pane of glass overview.

When Aria operations is deployed in a highly available design, it requires an external load balancer to provide a single point of entry for users and applications leveraging the Aria operations API.

To leverage the NSX Advanced Load Balancer (AVI), an instantiation of the controllers and service engines needs to be deployed in the Management cluster. This allows for load-balancing to the analytics cluster running in the main management cluster.

When creating the load balancer, ensure that the certificate covers all nodes (analytics nodes, cloud proxies, and load balancer) of the entire Aria Operations deployment.

Aria Operations - Distributed Design

Aria Operations supports the deployment of Cloud Proxies. Cloud proxies are additional nodes that distribute the ingestion of metrics from the adapters into the analytics cluster. The connection from the cloud proxy node to the analytics cluster is a one-way connection.

The deployment of the cloud proxies in pairs allows the formation of collector groups. The collection of a specific instance of an Aria Operations adapter can be assigned to a collector group, this allows for highly available collection of metrics even in the case of a service impact to one of the cloud proxies.

In the multi-site design, the cloud proxies must be deployed in the multi-site management domain. This enables a distributed collection of metrics across the Telco Cloud with a centralized management view.

A single cloud-proxy or collector group collects metrics from different adapters. Cloud proxies can be deployed in a standard or large form factor, depending on the number of metrics and objects the cloud proxy needs to collect.

Important There is a latency requirement for less than 200ms between the cloud-proxies and the Analytics nodes

Aria Operations - Scaling

Aria Operations can be scaled to support up to 16 Large Analytic nodes (or 12 Extra-Large nodes). Each node can be deployed in a Small, Medium, Large, or Extra-Large size. The CPU, Memory, and Disk requirements increase depending on the overall size of the Analytics node.

Aria Operations supports scaling up the nodes from medium to large and scaling out the number of supported nodes from 2 to 3 or more. Ensure that all the nodes are scaled up before scaling out.

Storage can also be added independently of scale-out or scale-up operations. When storage is added, ensure that the same additional storage amount is added to each node in the cluster.

The maximum number of metrics that can be supported depends on the node size. A Large node collects up to approximately 20,000 objects and 4 million metrics. A large cloud proxy collects up to 32,000 objects and 6.5 million metrics

Aria Operations - Management Packs

Aria Operations adapters and management packs come in two specific configurations:

- Normal adapters require a one-way communication to the monitored resources
- Hybrid adapters require two-way communication to the monitored resources

The Telco Cloud deployment for Aria Operations focuses on Normal adapters. The main adapters are vSphere, NSX, Cloud Director, Aria Operations for Logs and Kubernetes management packs.

Depending on the overall environment and the deployment of the Telco Cloud, the following adapters must be deployed:

- vSphere collects metrics from vCenter and ESXi hosts.
- vSAN collects metrics from vSAN datastores deployed throughout the telco cloud.
- NSX collects metrics from the NSX Manager and edge nodes.
- Aria Operations for logs collects and integrates Aria Operations with Aria Operations for logs.

If a storage other than vSAN is used, use the Management Pack for Storage devices to collect storage specific metrics.

Each adapter or management pack supports multiple instantiations. There is a single vSphere management pack, although each vCenter endpoint has its own instantiation. When creating each instantiation, the collection of the relevant metrics must be assigned to the correct cloud-proxy or collector group.

Aria Operations - K8s & Prometheus Integration

Aria operations can collect metrics from K8s deployments such as those provided by Tanzu Kubernetes Grid.

Prometheus is commonly used to provide access to metrics from a K8s cluster. However, metrics collected by Prometheus are typically presented in the form of a counter. Counter metrics are presented by a value that always increases.

To ensure that these metrics are interpreted properly by Aria Operations, the metrics must be rated over a period of time and presented to Aria Operations through a specific Prometheus configuration or through Aria Operations configurations.

When using VMware Telco Cloud Automation to deploy the Prometheus add-on to the Tanzu Kubernetes Grid cluster, a reference configuration can be used. The reference configuration includes modified recording rules that instruct prometheus to create additional metrics for consumption by Aria Operations. In addition, by using a custom kubernetes mapping within Aria Operations, these metrics can be assigned to the correct placement in the metric tree for easy viewing.

Aria Operations Design Recommendations

Table 4-19. Recommended Sizing for Aria Operations

Attribute	Specification
Appliance Size	Medium
Number of vCPUs	8
Memory	32GB
Disk Space	As required based on dimensioning

Design Recommendation	Design Justification	Design Implication
Deploy Aria Operations as a cluster of three nodes: ■ 1 Primary node ■ 1 Primary Replica node ■ 1 additional analytics node	■ Provides the scale capacity that is required to monitor up to 25,000 objects or 7 million metrics. ■ Supports scale-up with additional data nodes.	All the nodes must be sized identically.
Deploy two remote cloud proxies for the management domain	Reduces the load on the analytics cluster from collecting application metrics and provides availability for metric collection	Requires additional resources to create the cloud proxies
Deploy two cloud proxies for each management domain if using the multi-site model.	Places the collector closer to the source of the metrics	Requires additional resources to create the cloud proxies
Create collector groups for each management domain	Allows the adapter instances to be assigned to collector groups for higher availability	Requires additional configuration.

Design Recommendation	Design Justification	Design Implication
Deploy each node in the analytics cluster as a medium-size appliance.	Provides the scale required to monitor the Telco Cloud	ESXi hosts in the management cluster must have physical CPUs with a minimum of 8 cores per socket. Aria Operations uses a total of 24 vCPUs and 96 GB of memory in the management cluster.
Scale up all the existing analytics nodes (if required) before scaling out.	Ensures that the analytics cluster has enough capacity to meet the VM object and metric growth.	The capacity of the physical ESXi hosts must be sufficient to accommodate VMs that require the additional requirements without bridging NUMA node boundaries.
Use anti-affinity rules to ensure that the Analytics nodes (and cloud-proxies) are scheduled on different hosts.	Ensures high availability, a single node does not impact the overall Aria Operations platform	Requires enough hosts to be able to distribute the Aria Operations nodes. Host failure scenarios must be considered.
Deploy the standard-size cloud-proxies. Always deploy in pairs to allow for the creation of a collector group. If remote locations are planned to be large, use Large cloud proxies.	Creates a distributed and highly available environment for metric collection	You must provide 4 vCPUs and 8 GB of memory in the management cluster.
Add a virtual disk of 1 TB for each analytics cluster node.	Provides enough storage for the expected number of objects.	You must add the 1 TB disk manually while the VM for the analytics node is powered OFF.
Configure Aria Operations for SMTP outbound alerts.	Enables administrators and operators to receive email alerts from Aria Operations.	Aria Operations must have access to an external SMTP server.
Ensure that the management cluster is not heavily oversubscribed.	Oversubscribed management domains can impact the optimal performance of Aria Operations.	Requires active monitoring of CPU Ready, Co-Stop, and other metrics to ensure that Aria operations is not throttled.
Integrate Aria Operations with Active Directory users and groups.	Provides fine-grained role and privilege based access for varying user roles across the organization.	Requires access to Active Directory
Create service accounts for use with third-party integration. Align permissions with customer security policies.	■ Restricts access to the environment ■ Allows Aria Operations to function with a minimal set of features.	Custom configuration on Aria Operations or the integration points might be required.
Replace all the default certificates with CA signed certificates.	Ensures that all communication is properly encrypted and proper certificate management processes are followed.	A single certificate is required for all the nodes including the load balancer. As the Aria Operations deployment scales-out, a new certificate is required to accommodate the new node.

Design Recommendation	Design Justification	Design Implication
Deploy and configure connectivity through the following management packs or integration: ■ vCenter ■ NSX ■ Cloud Director ■ vSAN ■ Kubernetes ■ Aria Automation Orchestrator ■ Aria Operations for Logs	Ensures that metrics are collected from all endpoints of the Telco Cloud.	Requires the configuration and allocation to collector groups or cloud proxies for all integration points.
Configure Aria Operations to send logs to Aria Operations for Logs.	Enables logging events to be monitored by Aria Operations for Logs.	None
If monitoring Tanzu Kubernetes Grid clusters, ensure the deployment of Prometheus through Telco Cloud Automation	Allows reference configuration to be applied to ensure proper metrics collection	Requires deployment of add-ons through VMware Telco Cloud Automation Requires the deployment of v2 Clusters
To protect Prometheus metrics, use Avi to access the Prometheus server through TLS and AD integration.	Prevents metrics being presented over insecure HTTP endpoints. Allows NSX Advanced Load Balancer to be configured to leverage AD integration for authentication, if required.	Requires the NSX Advanced Load Balancer operator to be deployed into the cluster Requires service edges to be deployed in proximity to the K8s clusters

Note Aria Operations must be configured to use minimum number of nodes. Use fewer, but larger nodes before scaling out and adding more analytic nodes. When adding resources to the Analytics nodes, ensure that you follow the considerations around resources and NUMA.

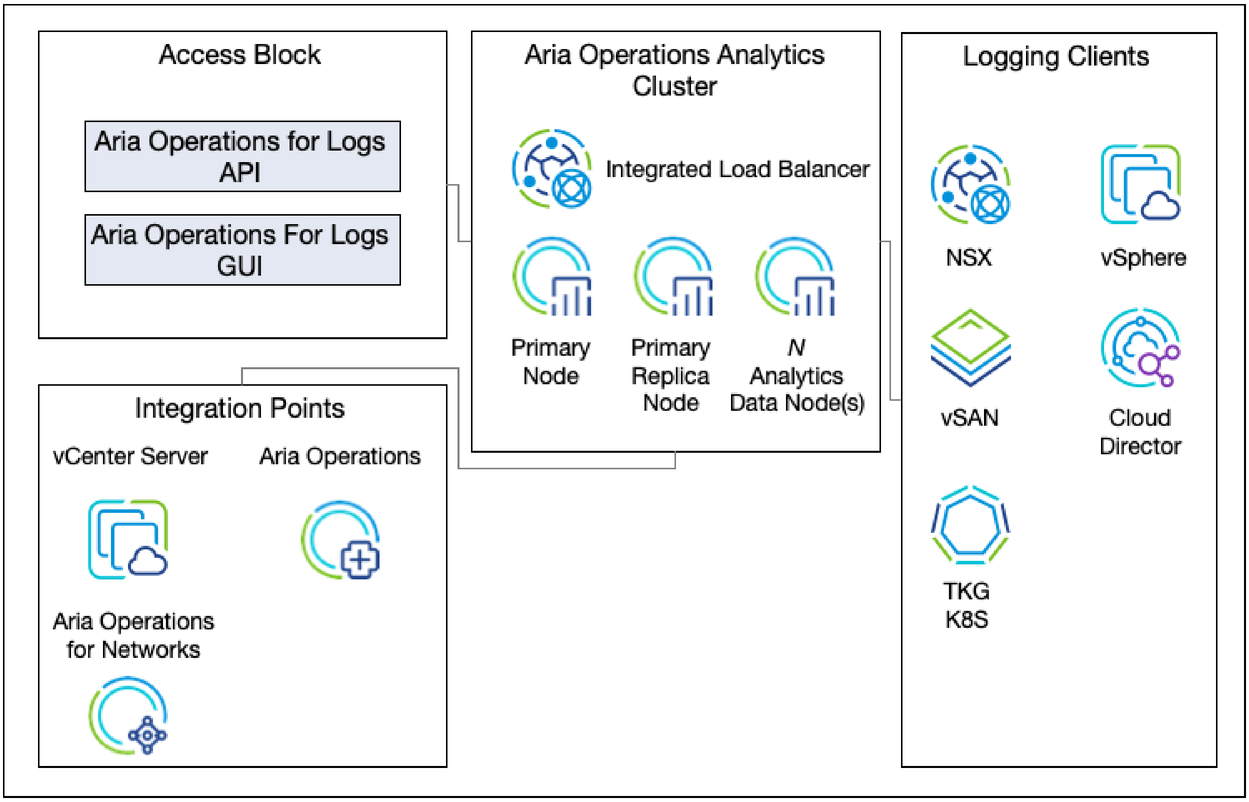
Aria Operations for Logs

The Aria Operations for Logs design enables real-time logging for all components in the telco cloud. The Aria Operations for Logs cluster consists of one primary node and two or more secondary nodes behind a load balancer.

Aria Operations for Logs - Logical Design

The deployment of Aria Operations for Logs is a single instance of a multi-node logging cluster that is deployed in the management cluster. Initial three-node cluster must be created for a highly available Aria Operations for Logs deployment. Additional nodes can be added to scale the deployment.

Figure 4-48. Logical Design Components for Aria Operations for Logs



The Aria Operations for Logs deployment contains the nodes that analyze and store data from the logging clients. The deployment configuration (number and sizing of nodes) for the Aria Operations for Logs cluster must be sized to meet the requirements for log ingestion rate.

Aria Operations for Logs integrates with other platforms in the Monitoring or Observability framework of the Telco Cloud to exchange data with platforms. For example, the in-place capabilities of Aria Operations for Logs allows syslog messages pertaining to a specific event to be viewed seamlessly from within Aria Operations.

The Integrated Load Balancer (ILB) must be used on the Aria Operations for Logs cluster so that all log sources can address the cluster by a load-balanced address, by using the ILB. When a scale-out or node failure occurs, you do not need to reconfigure log sources with a new destination address. The ILB also guarantees that the Aria Operations for Logs cluster accepts all incoming ingestion traffic.

The ILB address is required for users to connect to Aria Operations for Logs using either the Web UI or API and for clients to ingest logs using syslog or the Ingestion API.

Note Multiple ingress IP addresses can be allocated to the Aria Operations for Logs cluster. Each unique entry can implement ingress tagging for all log messages. Ingress tagging provides a high-level distinction among different elements of the Telco Cloud, such as RAN, 5G Core, and so on. Up to 60 Ingress IP addresses can be created per cluster.

Aria Operations for Logs - Distributed Design

Aria Operations for Logs does not support the same distributed design of Aria Operations. The concept of cloud proxies does not exist for Aria Operations for Logs.

The distributed model of Aria Operations for Logs is to create separate instances and use them as forwarders to a centralized Aria Operations for Logs cluster.

In the multi-site design, if desired separate smaller instances of Aria Operations for Logs must be deployed in the multi-site management domain, this will allow for a distributed collection of logs across the Telco Cloud with a centralized management view.

The distributed model allows for increased Ingress IP addresses. Multiple ingress addresses can be created per deployment. This can be useful to apply ingress tagging metadata to logs to indicate a site, region, or other data so that logs can be navigated easily from the centralized management domain.

Aria Operations for Logs - Scaling

Aria Operations for Logs can be scaled to support up to 18 nodes (1 Primary and 17 Workers). Each node can be deployed in a Small, Medium, or Large form factor. The CPU, Memory, and Disk requirements increase depending on the overall size of the node.

Aria Operations for Logs also supports scaling up the nodes from medium to large and scaling out the number of supported nodes from 3 to 4 or more.

Storage can also be added independently of scale-out or scale-up operations. When a storage is added, ensure the same additional storage is added to each node in the cluster. A maximum of 4TB storage can be added to each node. The storage can be 2x2TB disks or 4x1TB disks. A single disk cannot be larger than 2TB.

The maximum number of logs that can be supported depends on the node size. A Large node collects up to approximately 1,50,000 events per second from up to 750 syslog sources.

Aria Operations for Logs supports the following alerts that trigger notifications about its health and the monitored solutions:

- **System Alerts**: Aria Operations for Logs generates notifications when an important system event occurs. For example, when the disk space is almost exhausted and Aria Operations for Logs must start deleting or archiving old log files.
- **Content Pack Alerts**: Content packs contain default alerts that can be configured to send notifications. These alerts are specific to the content pack and are deactivated by default.
- **User-Defined Alerts**: Administrators and users can define alerts based on the data ingested by Aria Operations for Logs.

Note Each ESXi host sends up to 10 messages per second with an average message size of 170 bytes/message, which is equivalent to 150 MB per day for each host.

Aria Operations for Logs - K8s & FluentBit and Integrations

In addition to the built-in log collection facilities provided by the vSphere components and VM appliances such as TCA and TCA-CP, logs need to be collected from Kubernetes-based components that run as containers in the worker nodes.

FluentBit is a commonly used kubernetes logging component to capture logs from a K8s cluster. Fluent-bit also collects the logs from both the Kubernetes pods and VM processes on each worker node within the cluster, adding the required metadata and performing routing to the desired Aria Operations for Logs endpoint.

The stdout/stderr stream of messages from all containers is also captured by kubelet and stored in files on the worker node. You can use FluentBit to collect and forward these logs to one or more endpoints such as the regional Aria Operations for Logs cluster. Additionally, application logs can be forwarded to an application-specific logging stack provided by the application vendor, this allows centralized access to application logs without compromising sensitive infrastructure component data.

When using Telco Cloud Automation to deploy the FluentBit add-on to the Tanzu Kubernetes cluster, a reference configuration can be used. The reference configuration includes modified filters, inputs, and outputs for consumption by Aria Operations for Logs. The reference configuration ensures that the cluster name is added to the logging messages, simplifying the capability to search for logs from a specific cluster.

Aria Operation for Logs can act as a log forwarder. By using combinations of tagging, Aria Log Forwarding and Fluentbit output targets, specific logs can be sent to external logging platforms (such as a SIEM). This capability enables separation and isolation of infrastructure, application data, and security events.

Aria Operations for Logs Recommendations

Table 4-20. Recommended Sizing for Aria Operations for Logs

Attribute	Specification
Appliance Size	Medium (75 GB/ Day Logs, 400 events / second)
Number of vCPUs	8
Memory	16 GB
Disk Space	As required based on dimensioning

Design Recommendation	Design Justification	Design Implication
Deploy Aria Operations for Logs in a cluster configuration of three nodes with an integrated load balancer: - one primary node - two worker nodes	Provides high availability The integrated load balancer: - Prevents a single point of failure - Simplifies the Aria Operations for Logs deployment and subsequent integration - Simplifies the Aria Operation for Logs scale-out operations, reducing the need to reconfigure existing logging sources	- Deploy a minimum of three medium nodes - Size each node identically - If the capacity of your Aria Operations for Logs must expand, add identical capacity to each node.
Deploy Aria Operations for Logs with nodes of at least medium size.	Accommodates the number of expected syslog and connections from the following sources: - Management and Compute vCenter Servers - Management and Compute ESXi hosts - NSX Components - Aria Operations Components - Telco Cloud Automation and Tanzu Kubernetes Grid clusters	If you configure Aria Operations for Logs to monitor additional syslog sources, increase the size of the nodes.
Enable alerting over SMTP	Administrators and operators can receive email alerts from Aria Operations for Logs	Requires access to an external SMTP server.
Forward alerts to Aria Operations.	Provides monitoring and alerting information that is pushed from Aria Operations for Logs to Aria Operations for centralized administration.	None
Leverage fluent-bit on the Tanzu Kubernetes clusters to forward syslog information to Aria Operations for Logs.	Provides a central logging infrastructure for all the core Tanzu Kubernetes clusters	None
Integrate Aria Operations for Logs with Active Directory users and groups.	Provides fine-grained role and privilege based access for varying user roles across the organization.	Requires access to Active Directory
Create service accounts for use with third-party integrations Align permissions with customer security policies	Restricts access to the environment and allows Aria Operations for logs to functions with a minimal set of features.	May require custom configuration on Aria Operations for Log or the integration points
Replace all the default certificates with CA signed certificates.	Ensures that all communication is properly encrypted and proper certificate management processes are followed	A single certificate is required for all the nodes including the load balancer. As the Aria Operations for logs deployment scales-out, a new certificate is required to accommodate the new node.

Design Recommendation	Design Justification	Design Implication
Deploy and configure connectivity through the following management packs or integrations: ■ vCenter ■ NSX ■ Cloud Director ■ Aria Automation Orchestrator	Ensures that metrics are collected from all endpoints of the Telco Cloud	Requires the configuration and allocation to collector groups or cloud proxies for all integration points.
Wherever possible, leverage logging over TCP or TLS for reliable and secure transmissions.	Ensures reliable and secure transmission	May require additional configuration on the appliances and additional firewall port openings to support TCP connections.
When monitoring Tanzu Kubernetes Grid clusters, ensure that FluentBit is deployed through Telco Cloud Automation	Allows reference configuration to be applied to ensure proper logging collection, including logging using TLS	Requires deployment of add-ons through VMware Telco Cloud Automation Requires the deployment of v2 Clusters
When using multiple Aria Operations for Logs in a multi-site environment, ensure that CFAPI is used between clusters.	Maintains the original syslog message to ensure that correct source location is correlated	None

Aria Operations for Networks

Aria Operations for Networks communicates with management components to collect metrics, topology data, and connectivity that are presented through various dashboards and views. Aria Operations for Networks collects metrics from various VMware and non-VMware products, including vSphere, NSX, Kubernetes, and physical network components.

Aria Operations for Networks - Logical Design

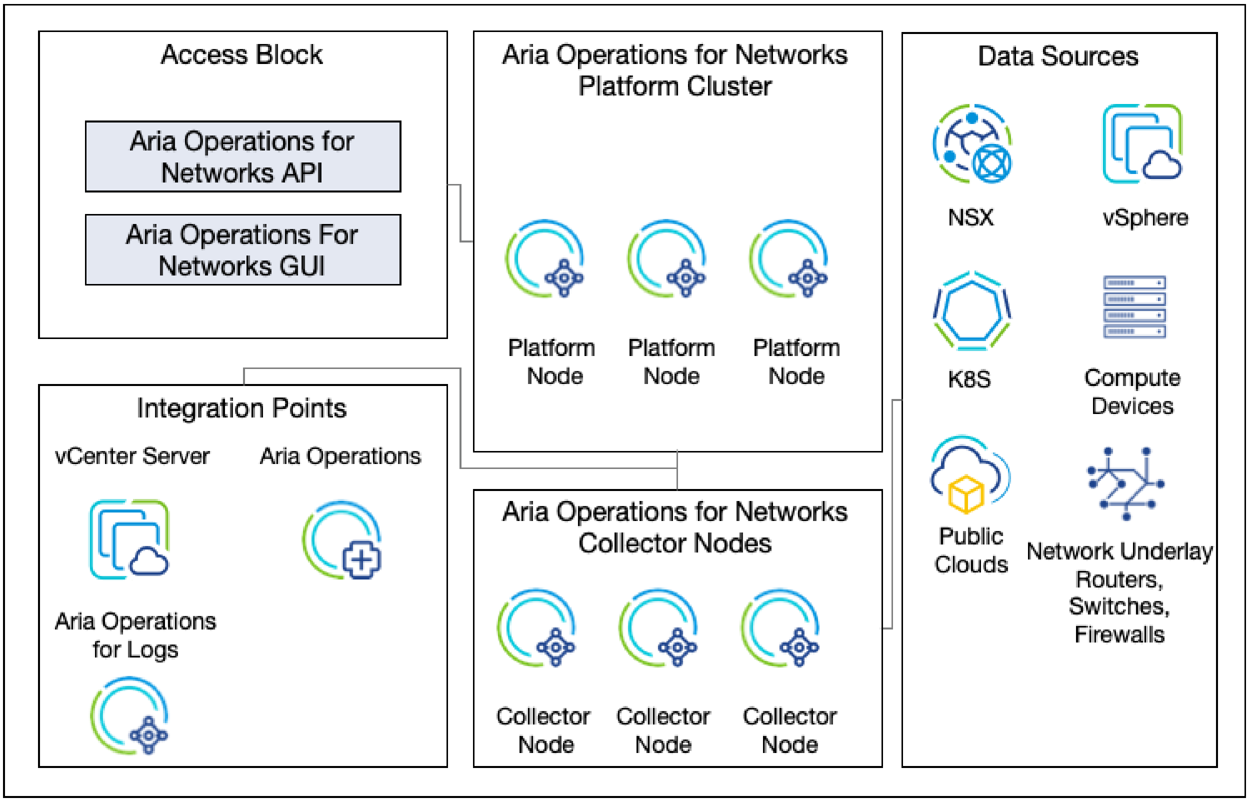
Aria Operations for Networks is deployed as a cluster called the Aria Operations for Networks Platform Cluster. This cluster processes the collected data and presents it using a dashboard. Aria Operations for Networks also uses collector nodes to collect data from the data sources, such as vCenter Server , NSX Manager, and NSX Advance Load Balancer, and send the data to the Platform Cluster for processing.

The platform cluster capacity is based on the deployment size of each node and the number of nodes composing the cluster. For more information about how to decide the platform brick (node) size and number of platform bricks (nodes), see the Aria Operations for Networks documentation.

The collector capacity is based on the deployment size. The amount of data sources that you can add to a specific collector is based on the documented capacity of the collector size (accounting for the number of VMs and the number of flows).

Note When using Aria Operations for Networks, it provides an option to enable IPFIX export on vCenter switches. While this provides additional information for communication between the elements of the Telco Cloud, IPFIX exporting can add latency issues. As such, IPFIX is not recommended for deployment on Workload Switches as this can impact the performance of user-plane functions.

Figure 4-49. Logical Design Components for Aria Operations for Networks



Aria Operations for Networks - Scaling

Aria operations for Networks scales in two dimensions: the platform and the collectors. The platform can only be clustered if using appliances of at least the large size. Medium deployments cannot be scaled into a cluster without first scaling up to large.

The scaling of a platform to a cluster model does not provide additional availability to the platform. Scaling must be used when maximums for a single node exceed.

The overall dimensioning for collector scaling is based on various factors including:

- Number of vCenter Servers
- Number of VMs
- Number of flows to track

To determine when to size the platform and collector nodes, see the Aria Operations for Network Scaling deployment guide. When collecting over 10,000 VMs and 4 million active flows, the platform must be upgraded from a single node to a clustered deployment. Similar scale-up and scale-out requirements exist for the Collectors.

Aria Operation for Network Design Recommendations

Table 4-21. Recommended Sizing for Aria Operations for Networks

Attribute	Specification
Appliance Size - Platform	Large
Number of vCPUs (based on CPU speed)	2.1 Ghz - 15 vCPUs, 2.3 Ghz - 14 vCPUs, 2.6 Ghz - 12 vCPUs
Memory	48 GB
Disk Space	1 TB
Appliance Size - Collector	Large
Number of vCPUs (based on CPU speed)	2.1 Ghz - 10 vCPUs, 2.3 Ghz - 9 vCPUs, 2.6 Ghz - 8vCPU
Memory	16 GB
Disk Space	200 GB

Design Recommendation	Design Justification	Design Implication
Deploy a single, large Aria Operations for Network platform instance.	A cluster is required based on scale. A single large platform node allows expansion into a cluster while providing initial monitoring capabilities The platform can be scaled into a cluster based on the future growth or feature.	Each platform node must have 100% reservation for CPU and RAM.
Deploy at least one Collector Deployment for every Workload Domain.	Enables data collection locally on every Workload Domain	Each Collector node must have 100% reservation for CPU and RAM. Dimensioning of Collector must be checked during Workload Domain creation/expansions. Collector nodes are not highly available, but they must be protected by vSphere HA.
Platform and Collector nodes must be scaled as the network scales to accommodate the additional load.	Aria Operations for Networks is based on components that can be scaled out or scaled up as required.	Additional resources must be available.
Aria Operation for Networks can be connected to the vCenter, NSX, NSX Advanced Load Balancer, and Log Insight data sources.	Provides network visibility of the virtual networking.	Requires valid service accounts for Aria for Operations components.

Bare Metal Automation

VMware Bare Metal Automaton (BMA) is a platform that offers a wide range of capabilities for business process automation. VMware Bare Metal Automation is a low-code workflow engine to automate the deployment of ESXi on bare metal servers. Additional capabilities include the server BIOS configuration, firmware and bios version management, and custom workflows to integrate with existing CI/CD pipelines post execution.

VMware Bare Metal Automation - Logical Design

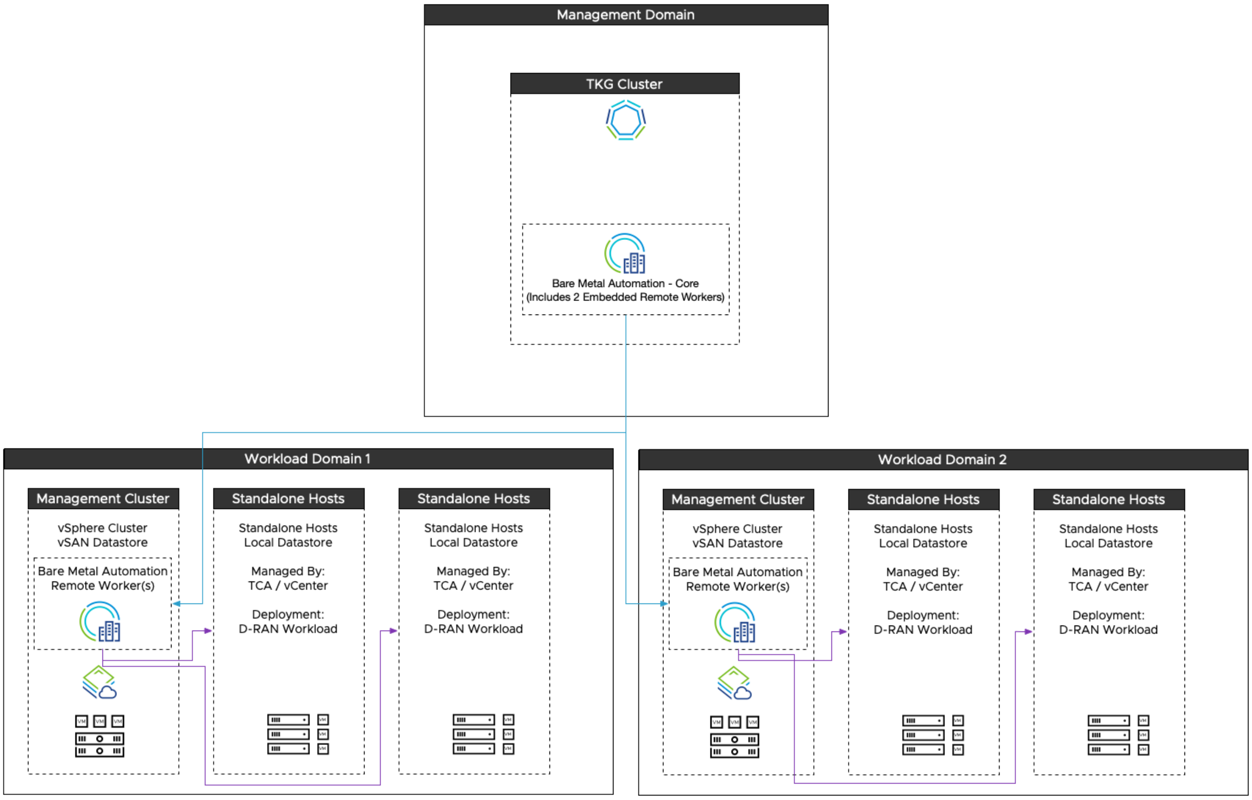
VMware Bare Metal Automation (BMA) is deployed as a cloud-native application, which requires an existing Kubernetes cluster to host the BMA components.

BMA is split into two architectural components:

- **BMA Core**: The core component is the primary interface. It provides dashboards, plug-in repositories, and code editing. The real-time execution engine can exist within the core deployment or separately as a remote worker.
- **BMA Remote workers**: The remote worker component is also a cloud-native application. It requires a Kubernetes cluster to host the components of the remote worker. Firewalls for HTTPs must be opened between the remote workers and the BMA core.

ISO server is an additional component that is used to create the ESXi ISO images for remote mounting to the servers. For optimal deployment, ISO servers must be deployed at each location where the remote workers are deployed for maximum scale and performance.

Figure 4-50. Logical Design Components for VMware Bare Metal Automation



The logical design of VMware Bare Metal Automation includes the Core component deployed at the central management domain and two remote workers to execute bare metal deployment workflows.

To scale the deployment, remote workers can be deployed. In this diagram, the remote workers are deployed at the multi-site management domain. In reality, these remote workers can exist throughout the network, as close to the standalone hosts as possible for efficient communication.

The Kubernetes service for VMware Bare Metal Automation is of type Load Balancer. An external load balancer is required for the Core services.

VMware Bare Metal Automation - Scaling

VMware Bare Metal Automation can be scaled by adding remote workers to the BMA deployment.

While it is possible to create larger worker nodes for the core and remote worker pods to execute on, overall that element of scale depends on factors including network latency and the overall workload of the hosts.

Note For more information about tuning and scaling up the BMA core, contact your local VMware representative.

Worker groups can be created to aggregate the remote workers. Multiple workers can be added to a worker group, when using groups BMA will round-robin the requests, each BMA worker can execute up to 8 parallel tasks.

Note When adding remote workers to a group, permissions for that group do not exist. You must add the appropriate role and user permissions in VMware Bare Metal Automation.

Currently, up to 50 BMA remote workers can be connected to a BMA core instance. Thus, when devising a placement plan for the remote workers in a large deployment, the distribution of remote workers to worker groups can be as required. Ensure that the 50 remote worker limit is not exceeded. The recommended latency between the BMA core and BMA remote workers must be less than 200 ms.

Bare Metal Automation Design Recommendations

Table 4-22. Recommended Sizing for Bare Metal Automation

Attribute	Specification
BMA Core	Kubernetes cluster with at least two worker nodes
Number of vCPUs	4 vCPUs
Memory	16 GB
Disk Space	50 GB
Remote Worker	Kubernetes cluster with at least one worker node (more if multiple workers are needed)
Number of vCPUs	1 vCPU
Memory	4 GB
Disk Space	50 GB

Design Recommendation	Design Justification	Design Implication
Deploy a BMA core instance (along with an ISO server) into the central management domain.	Places the core BMA platform in the management cluster. Used as the main API or UI interface for BMA along with two remote workers in the core. Can be used for standalone hosts in proximity with the central management domain or for non-multisite designs.	Requires the creation of a TKG cluster in the management domain. Can be created with VMware Telco Cloud Automation, but requires a TCA-CP node for the management vCenter
Deploy at least one remote worker (and ISO server) for every Workload Domain.	Allows the execution of workflows and preparation to be more distributed	Requires a TKG cluster for the remote worker to execute on and a web server for the ISO component in each distributed domain

Design Recommendation	Design Justification	Design Implication
Create groups for each set of remote workers.	Allows load-sharing and distribution of tasks	Requires the creation of worker groups in the BMA core configuration.
Integrate VMware Bare Metal Automation with LDAP directory services.	Allows for centralized control of user management	Requires manual configuration to integrate with customer LDAP environment
Configure logging from Pliant to the Aria Operations for Logs platform. Use TCP or TLS for secure logging.	Allows log messages to be sent to a centralized platform	Requires certificates if using TLS based logging and firewall ports to be opened between BMA components and the Aria Operations for Logs cluster.
Use NSX Advanced Load Balancer and AVI Kubernetes Operator (AKO) to provide the Kubernetes load-balancer service.	Allows a supported load-balancer to expose the VMware Bare Metal Automation services.	Requires AKO to be deployed to the Tanzu Kubernetes cluster