WHITE PAPER–MARCH 2020

Networking and Security in VMware vSphere with Kubernetes

vmware®

Contents

Introduction

Enterprises both large and small constantly undergo application transformation to stay ahead of their competition. As part of this, utilizing containers and microservices emerged as the dominant software development pattern for application modernization. Kubernetes was developed to help manage deployment and availability of containerized applications, especially with automation and container orchestration.

Kubernetes is an open-source project originally developed for lifecycle management for hundreds of thousands of containers at Google. It is an open-source project governed by the Cloud Native Computing Foundation (CNCF). Since Google made Kubernetes open source in 2015, it has become the de facto container orchestration platform with a rich, growing ecosystem of surrounding services, support and tools.

The 2019 CNCF survey shows that 78 percent of respondents currently use Kubernetes in production. This is up from the 58 percent using it in 2018. In addition, 18 percent of respondents run more than 5,000 containers in production.[1]

Kubernetes architecture and components

When you first launch Kubernetes, you get a Kubernetes cluster, which contains many components. In this white paper, we will discuss some of those components that are relevant to deploying VMware vSphere® with Kubernetes. For additional information related to Kubernetes components, you can visit the Kubernetes documentation.

Nodes

There are two primary node types used in Kubernetes: a master and a worker. A master node manages a set of worker nodes (workload runtimes) and is composed of the following components:

- Kube-apiserver acts as the front end to the cluster. All external communication to the cluster is conducted via the apiserver.
- Kube-controller-manager runs a set of controllers for the running cluster. The controller-manager provides governance across the cluster.
- Kube-scheduler schedules activities to the worker nodes based on events occurring on the etcd. The kube-scheduler selects the optimal node for newly created pods or unscheduled pods.
- Etcd is the cluster state database, used for storing and replicating the Kubernetes cluster state.

Pod

A pod is a group of one or more containers. Pods are managed by the kubelet running on each node. The kubelet watches all podspecs assigned to it and handles the task throughout its lifecycle by comparing the actual pod state to the desired state defined in the podspec.

1. Cloud Native Computing Foundation. "CNCF Survey 2019." March 2020.

Kubelet

A kubelet is an agent on each Kubernetes node that ensures they are running in a pod. The kubelet watches all podspecs assigned to it and handles the task throughout its lifecycle by comparing the actual pod state to the desired state defined in the podspec.

Figure 1 demonstrates how these components are connected and work together.

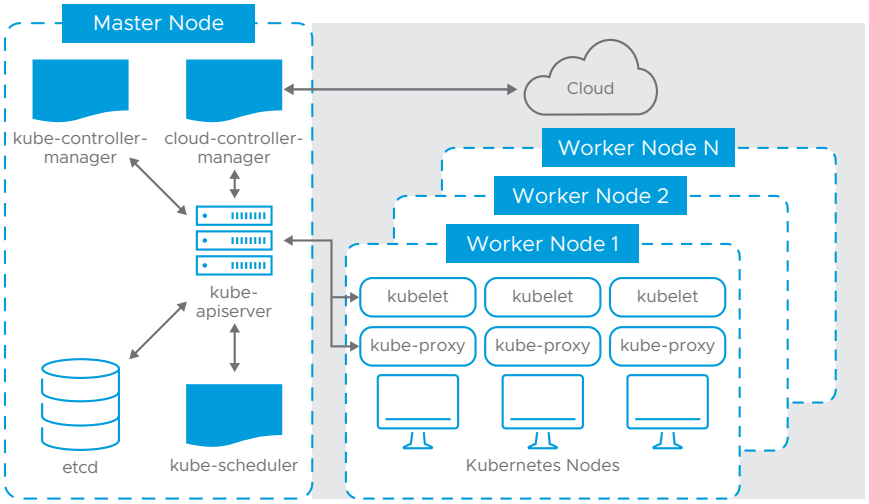

FIGURE 1: Kubernetes architecture and components (Kubernetes docs).

What is VMware vSphere with Kubernetes?

VMware vSphere with Kubernetes is simply a re-architecture of vSphere that utilizes Kubernetes as the control plane. This solution is uniquely suited for running traditional workloads and modern, cloud native applications. It allows developers to use the familiar Kubernetes API to manage cloud resources—such as virtual machines (VMs), disks and networks—while enabling virtualization admins to manage the entire application rather than deal with individual VMs.

Despite its advantages, there are some reasons companies and individuals do not implement Kubernetes and resist doing so. The top three barriers to adopting Kubernetes are cultural change (43 percent), security (40 percent) and complexity (38 percent), according to the 2019 CNCF survey.[1] vSphere with Kubernetes addresses all three of these concerns.

vSphere with Kubernetes represents the best of both worlds and provides users with everything they need. Virtual infrastructure (VI) admins can provide clusters on demand to support developers while using their current vSphere management tools—such as vSphere Client, vSphere PowerCLI™ and APIs—to manage VMs. To a developer, vSphere with Kubernetes looks and acts like a standard Kubernetes cluster, and they can use standard Kubernetes APIs to define necessary resources, such as storage and networking. With this approach, there is no need to learn or use vSphere APIs, clients or infrastructure.

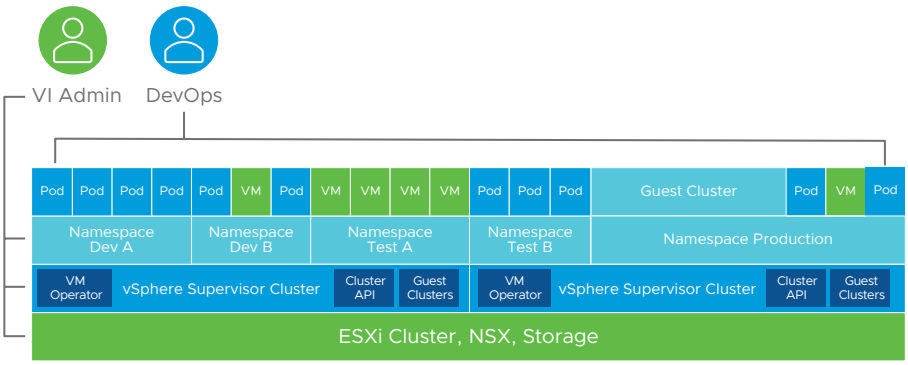

FIGURE 2: vSphere with Kubernetes represents the best of both worlds for VI admins and DevOps.

With vSphere with Kubernetes, there are two types of Kubernetes clusters that can run natively:

- vSphere Supervisor Cluster™ – A Kubernetes control plane for vSphere
- VMware Tanzu™ Kubernetes Grid™ cluster (also called a guest cluster) – A standards-based cluster that is fully conformant with upstream Kubernetes

vSphere Supervisor Cluster

The vSphere Supervisor Cluster is a special type of Kubernetes cluster that uses VMware ESXi™ hosts as worker nodes instead of Linux nodes. It uses a custom kubelet (called a Spherelet) that is integrated directly into the ESXi hypervisor. The Spherelet is responsible for all aspects of the pod lifecycle and configuration, and all interaction between pods. It also ensures that any pod dependencies—such as images, networking and volumes—are available and correctly configured. In addition, the Spherelet periodically reports back about the pod health running in its domain, so the Kubernetes scheduler and K8s API master can execute corrective action, if needed.

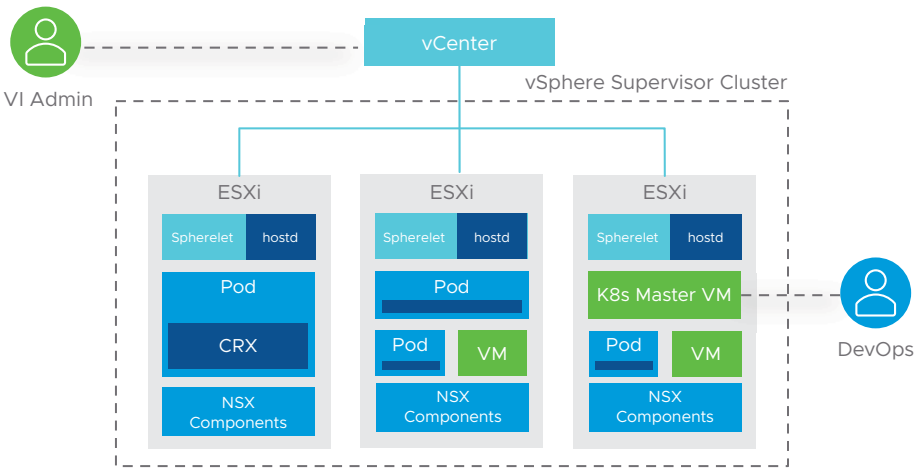

FIGURE 3: VMware NSX is designed into vSphere with Kubernetes. NSX components include the NSX Container Plug-in, vSphere Distributed Switch™ and NSX Agent.

Tanzu Kubernetes Grid cluster (or guest cluster)

The vSphere Supervisor Cluster serves as the control plane for vSphere with Kubernetes. While it uses Kubernetes, it is not itself a conformant Kubernetes cluster. On the other hand, the Tanzu Kubernetes Grid cluster was developed to offer fully conformant Kubernetes clusters for developers to consume as a service using standard Kubernetes commands.

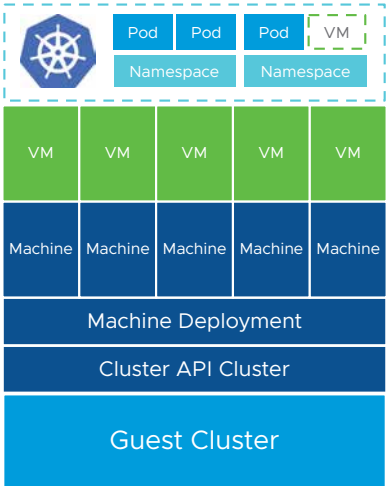

FIGURE 4: Components of a Tanzu Kubernetes Grid cluster (or guest cluster).

Namespaces in vSphere with Kubernetes

A namespace is used for management in environments that contain many users across multiple teams or projects. Namespaces serve as a solution to divide cluster resources and separate permissions among users. An organization can create four namespaces for different environments (such as dev, test and production) and for different types of applications (such as front-end, DB and analytics).

VMware NSX networking

Software-defined networking (SDN) is a core infrastructure primitive that touches all aspects of the fabric (hypervisors, VMs, containers, guest clusters, etc.). This allows for uniform security policy, controls and network abstraction, enabling seamless connectivity and built-in security.

VMware NSX® provides full-stack networking and security for vSphere with Kubernetes and VMware Cloud Foundation™. From micro-segmentation and load balancing to service mesh, NSX enables enterprises to connect and protect their microservices and workloads running in containers and VMs. With NSX, enterprises extend consistent policies across app environments and simplify Day 0 to Day 2 operations while delivering the agility and performance required to run modern apps.

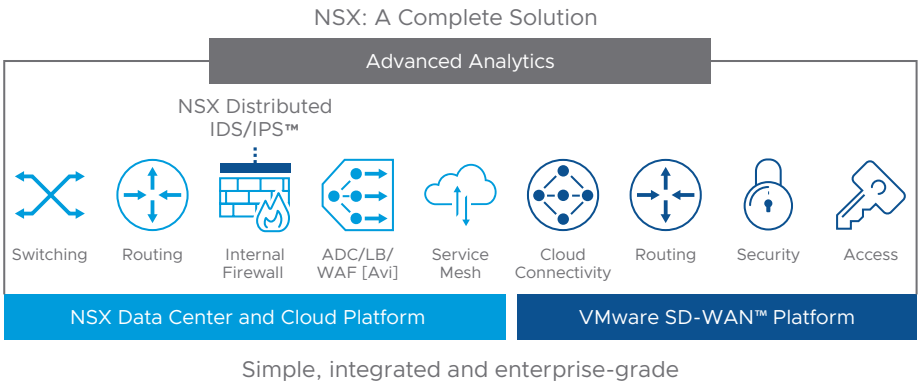

FIGURE 5: The NSX portfolio is the full-stack networking and security solution for VMs, containers and bare metal.

Why use NSX for VMs, containers and bare metal?

Networking for vSphere with Kubernetes remains consistent with an organization's existing networking. With native pods, VMs and containers, NSX is the only solution that can provide the full-stack networking for VMs, containers and bare metal with full visibility.

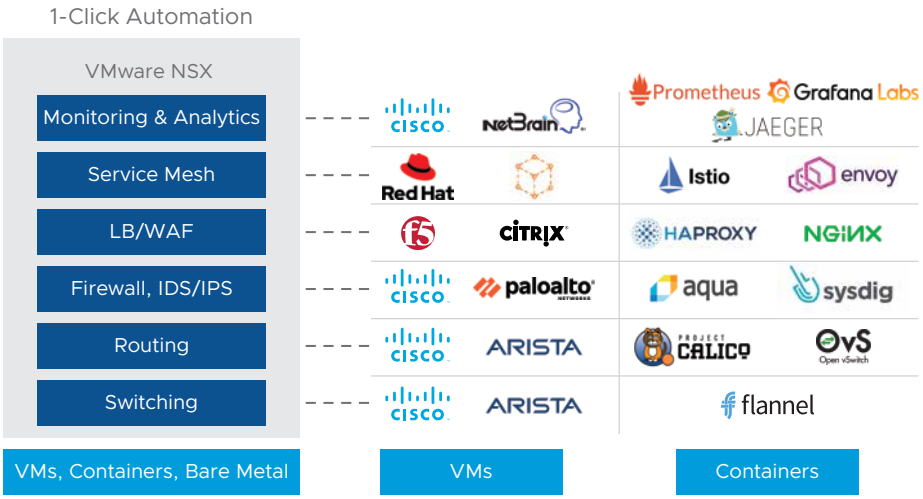

FIGURE 6: NSX provides full-stack networking from L2 to L7 for VMs, containers and bare metal, unlike DIY solutions that require several vendors.

Why NSX for vSphere with Kubernetes?

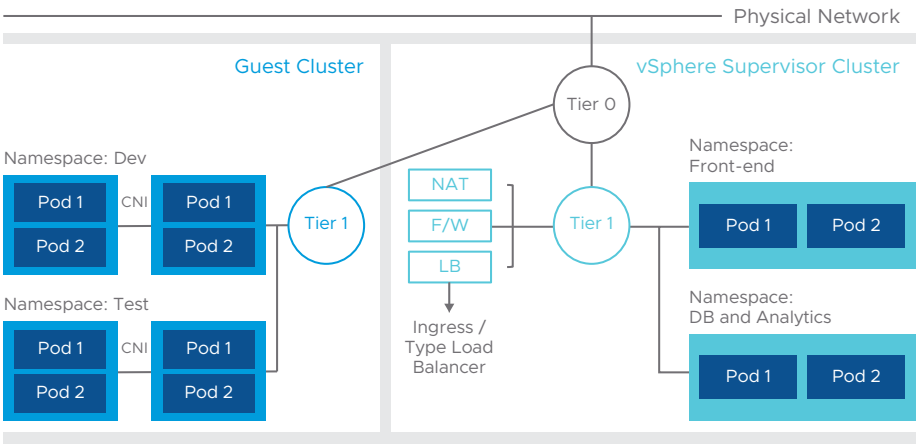

FIGURE 7: NSX provides full-stack networking capabilities for vSphere with Kubernetes.

NSX is designed into vSphere with Kubernetes, providing full-stack networking and security capabilities. It also provides connectivity between the vSphere Supervisor Cluster and the physical network.

For east-west traffic in the vSphere Supervisor Cluster, NSX provides distributed switching, routing and distributed load balancing for the different pods within a given namespace to communicate. For north-south traffic, NSX provides NAT, firewalling at the container level, ingress and type load balancers for public applications used with the outside world. All the networking and security functions can be controlled by the native Kubernetes network policy, allowing DevOps teams that use Kubernetes commands to manage their own networks.

In the guest cluster, connecting to the physical network and publishing applications is accomplished using the NSX Load Balancer. For pod-to-pod communication, vSphere with Kubernetes offers a choice of certified container networking interfaces (CNIs), including third-party networking plug-ins, Project Antrea and NSX Container Plug-in.

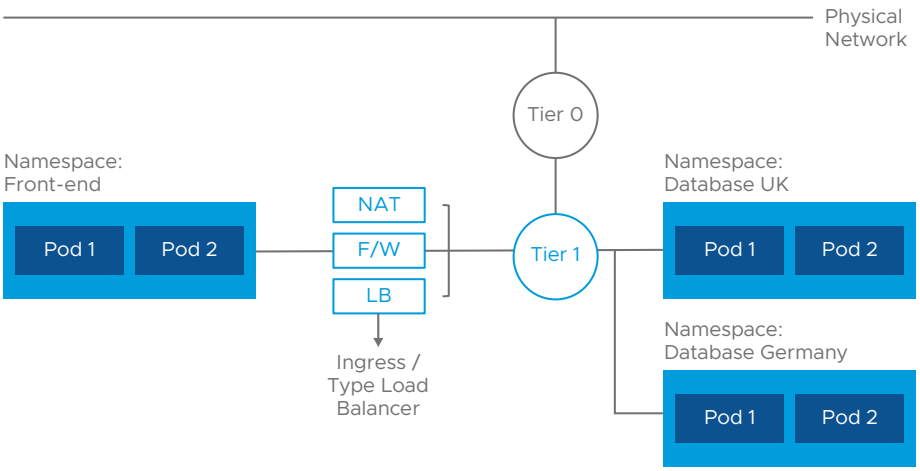

FIGURE 8: NSX provides full-stack networking and security services to vSphere with Kubernetes.

Unlike other container network solutions, NSX provides granular policies, such as SNAT, per namespace. This is especially useful for GDPR and compliance use cases.

How it works

NSX creates a logical segment for each namespace through the Tier 1 router. Firewall policies can then be applied to the container level. Traditional firewall appliances and load balancers do not allow for visibility into the container level. However, NSX has complete visibility into the containers and can apply firewall policies directly on the containers. While other load balancers are tied to a particular cluster, NSX provides fully distributed Layer 4 load balancing services to Kubernetes.

More importantly, NSX can provide more granular policies, such as SNAT, per namespace. This is useful for GDPR and compliance use cases where NSX can restrict access to geo-specific databases, isolated by their namespaces, and prevent access by front-end applications.

Enhanced for NSX-T 3.0

The NSX Distributed Load Balancer, enhanced in VMware NSX-T™ 3.0, provides the ability for services in the same namespace to communicate with each other.

Advantages for the full-stack engineer

Organizations are constantly working to accelerate service delivery, streamline operations and enter new markets. Running NSX and vSphere with Kubernetes helps these organizations implement full-stack networking and security for their traditional and modern workloads using existing tooling and governance models. Developers gain instant access to the tools they need to build and run applications on their choice of computer platforms and across multi-cloud solutions. When networking and security are defined and consumed in software, admins and developers can work at the same speed and drive toward common business objectives.

Glossary

vSphere with Kubernetes	A re-architecture of vSphere that utilizes Kubernetes as its control plane. vSphere with Kubernetes allows developers to use the familiar Kubernetes API to manage cloud resources (such as VMs, disks and networks) while enabling VI admins to manage the entire application rather than deal with individual VMs.
vSphere Supervisor Cluster	A type of Kubernetes cluster that uses ESXi hosts as worker nodes instead of Linux nodes. It uses a custom kubelet (called a Spherelet). Think of the vSphere Supervisor Cluster as a Kubernetes-native control plane over vSphere that enables Kubernetes as a service and VM as a service.
Spherelet	A custom management agent, contained in vSphere with Kubernetes, that is embedded directly into ESXi. Much like the kubelet that acts as the management node, the Spherelet enables the ESXi hypervisor to act as a native Kubernetes node participating in a Kubernetes cluster. This ensures that the native pods are always running and healthy.
Container runtime (CRX)	For ESXi to run LinuxOS, container runtime has been added to ESXi. It is presented to Kubernetes as an ESXi PodVM.
vSphere native pod	An enhanced VM used to run a lightweight Linux kernel and a small CRX.

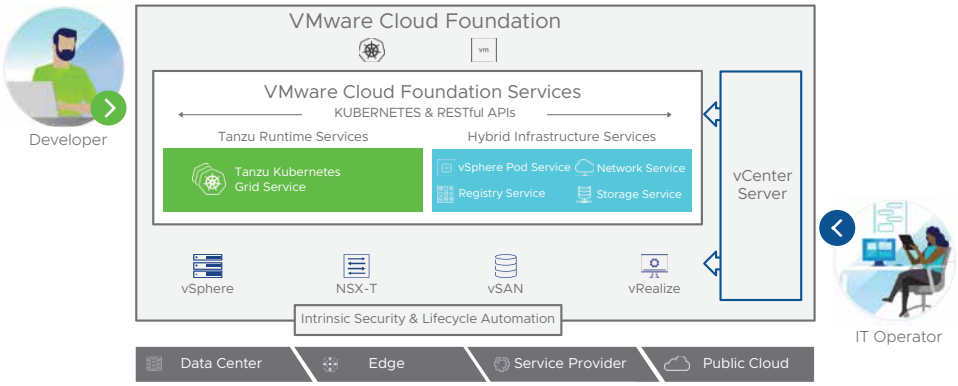

FIGURE 9: VMware Cloud Foundation includes SDDC, Tanzu runtime services and hybrid infrastructure services.

VMware Cloud Foundation Services	Tanzu runtime services and hybrid infrastructure services delivered through a set of Kubernetes and REST APIs.
Tanzu Kubernetes Grid™ service for vSphere	This is part of the Tanzu runtime services. Tanzu Kubernetes Grid service for vSphere allows developers to manage consistent, compliant and conformant Kubernetes clusters. The guest cluster service creates clusters on demand, and exposes networking and storage services through the CNI and container storage interface (CSI) plug-ins.
Hybrid infrastructure services	Consists of the following services: • vSphere Pod Service – Allows developers to run containers natively and securely on vSphere without having to manage VMs or Kubernetes clusters. • Registry Service – Allows developers to store, manage and secure Docker and OCI images using Project Harbor. Project Harbor is an open-source container image registry that secures images with role-based access control, scans images for vulnerabilities and signs images as trusted. • Network Service – Allows developers to define virtual routers, load balancers and firewall rules for use with their applications. • Storage Service – Allows VMware vCenter Server® storage policies and devices to be consumed as Kubernetes storage classes. They can also be used as persistent disks for use with containers, Kubernetes and VMs.

LEARN MORE

For information relating to VMware vSphere with Kubernetes, visit *vmware.com/products/vsphere*.

For information relating to VMware NSX, visit *vmware.com/products/nsx*.

Appendix: Kubernetes networking model 101

While Kubernetes does not provide a default networking implementation, it does provide a model for third-party tools to implement, also known as a CNI.

How pods communicate with each other

Each pod contains a unique IP in a flat address space inside the Kubernetes cluster, meaning direct pod-to-pod communication is possible without any type of proxy or address translation.

How pods communicate with services

Kubernetes services allow grouping pods under a common access policy. For example, a group of pods can be load balanced. In that case, the load balancing services are assigned a virtual IP, which pods outside can communicate with.

Incoming traffic from the outside world

Nodes inside a Kubernetes cluster are firewalled from the internet by default, and service IPs are only reachable within the cluster network. There are two approaches here: The first approach is to route requests using the external IP, and the second is to offer an ingress API.

To allow for incoming traffic, a service can be mapped to one or more external IPs. Incoming requests at the external IP are routed to the node. The node knows which services are mapped to that external IP and which pods are part of the service. After this, the request is routed to the appropriate pod.

To support more complex policies, Kubernetes provides the ingress API, offering externally reachable URLs, traffic load balancing, SSL termination and name-based virtual hosting. An ingress is a collection of rules that allow inbound connection to the service. An ingress controller, usually a load balancer, is responsible for fulfilling the ingress.

An ingress controller allocates an external IP to satisfy the rules defined by the ingress and forwards all requests arriving at that external IP to the service mapped in the ingress specification.

DNS for services and pods

Kubernetes provides its own DNS service to resolve domain names inside the cluster, so pods can communicate with each other. This is implemented by deploying a regular Kubernetes service that does not name the resolution inside the cluster. It then configures individual containers to contact the DNS service to resolve domain names.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com Item No: 495450aq-wp-netwkng-sec-vsphr-k8s-a4-Final3 3/20