Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6, 6557-6578
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i6.3409

Managing cloud-native applications using vSphere with Tanzu and Tanzu Kubernetes grid

Vedran Dakić[1*], Adriano Bubnjek[2]
[1,2]Department of system engineering and cybersecurity, Algebra University; vedran.dakic@algebra.hr (V.D.) abubnjek@algebra.hr (A.B.).

Abstract: This study examines VMware cloud-native platforms, emphasizing their orchestration and automated management functionalities for container technologies, which are essential in modern IT infrastructure. The study seeks to comprehensively examine VMware Tanzu's architecture, technical elements, and efficacy in facilitating containerized applications. Case studies were used to gain comprehensive insights into brownfield and greenfield deployment strategies, illustrating practical implementation challenges and advantages for vSphere with Tanzu and Tanzu Kubernetes Grid. These studies emphasize scenarios from gradually incorporating Kubernetes into current VMware environments to developing optimized infrastructures from the ground up. Significant findings encompass strategies for scalable IT infrastructure, innovative applications of persistent storage via VMware Cloud Native Storage, and the function of Tanzu Kubernetes Grid in multi-cloud deployment contexts. The study concludes that Tanzu platforms successfully integrate traditional virtualization with contemporary containerization, providing robust scalability, flexibility, and efficient resource-use solutions. The practical implications emphasize how organizations can utilize these insights to seamlessly integrate Kubernetes workloads, optimize investments in VMware ecosystems, and respond to evolving IT requirements.
Keywords: *Cloud-native, Containerization, Kubernetes, Orchestration, Tanzu, Virtualization, VMware.*

1. Introduction

This paper examines VMware cloud-native platforms for the orchestration and automated management of container technologies, a swiftly advancing field in information technology. A significant focus is placed on platform architecture and technical performance, illustrated through images and code detailing configuration steps. The tools prioritize console-based and graphical interface solutions rather than conventional command-line techniques. Nonetheless, configuring Tanzu platforms is a complex undertaking.

VMware's offerings, including Tanzu, are proprietary and expensive. Extensive literature on these technologies is limited, with professional texts available in restricted quantities. One of the cited books, published just last year, emphasizes the originality of these solutions. VMware produces comprehensive technical documentation primarily aimed at vSphere administrators knowledgeable in vSphere and NSX platforms. Official VMware training courses offer an alternative learning pathway, though at a significant expense, prioritizing technical implementation rather than theoretical understanding.

There are three main scientific contributions of this paper:

1. Strategy for scalable and adaptive IT infrastructure using vSphere for Tanzu and Tanzu Kubernetes Grid - The paper provides a detailed analysis for integrating vSphere with Tanzu and Tanzu Kubernetes Grid with existing vSphere environments or from zero, emphasizing the architecture and technical components needed for container orchestration and automation.
2. Analysis of Implementation of Persistent Storage in VMware CNS - By integrating VMware's Cloud Native Storage (CNS) with Kubernetes, the study demonstrates a creative approach to

* Correspondence: vedran.dakic@algebra.hr

enabling persistent storage for containerized applications. It explains the configuration of CSI drivers, StorageClass manifests, and PersistentVolumeClaims (PVCs), highlighting efficient resource management strategies within hybrid infrastructure environments. This offers significant insights into enhancing Kubernetes storage capabilities in enterprise-level IT solutions.

The rest of this paper is organized as follows: first, the related works section discusses the current state of the research field, followed by five sections related to the technology used throughout the paper – vSphere with Tanzu and Tanzu Kubernetes Grid, VMware Cloud Native Storage, Tanzu Mission Control and NSX. The rest of the paper contains three case studies related to the vSphere with Tanzu integration, followed by a section about potential future research areas and a conclusion.

2. Related Works

The advancement of IT infrastructure has increased dependence on virtualization and containerization technologies for resource optimization and scalability. Virtualization, a well-established technique, abstracts physical hardware into virtual machines (VMs), providing resource isolation and flexibility. Nonetheless, virtualization entails considerable resource overhead due to each virtual machine's requirement for dedicated operating systems. Containerization arose as a viable alternative, providing lightweight solutions via shared operating systems. Containers utilize Linux functionalities like cgroups and namespaces for resource management and isolation, enhancing efficiency and portability.

Comparative analyses have emphasized the performance trade-offs between virtualization and containerization. Containers surpass virtual machines' resource efficiency and initialization speed, rendering them optimal for microservices architectures. Nonetheless, virtual machines offer enhanced isolation and security, which is essential in specific situations [2,3]. Research demonstrates that while containers improve operational agility, their shared kernel architecture may introduce vulnerabilities if the host kernel is compromised [4].

Cloud-native architectures emphasize scalability, adaptability, and swift deployment. These principles are encapsulated in container orchestration platforms such as Kubernetes, which oversee container lifecycles, scaling, and networking. Kubernetes has become the preeminent platform, facilitating microservices deployment and enabling self-healing via automated resource allocation [5]. Research highlights Kubernetes' significance in unifying cloud-native applications with hybrid and multi-cloud environments [6].

Runtime performance significantly influences the efficacy of container orchestration systems. Default placement strategies in Kubernetes frequently do not enhance performance, necessitating the development of sophisticated algorithms that consider runtime metrics [7]. Performance-aware deployment strategies can improve efficiency in containerized environments by a factor of two [8].

The implementation of virtualization and containerization has progressed markedly, especially in hybrid cloud settings. VMware Tanzu demonstrates the seamless integration of these technologies by utilizing VMware vSphere to facilitate Kubernetes workloads. Tanzu facilitates resource isolation through vSphere namespaces and provides persistent storage for containerized applications, as evidenced by industry case studies [9].

Energy efficiency is an essential factor. Research indicates that container-based solutions markedly decrease energy consumption in cloud data centers relative to conventional virtualization methods [10]. Techniques such as resource consolidation and optimized traffic steering enhance efficiency, making containerized architectures a sustainable choice [11].

Despite their benefits, containers need help with security and resource isolation. Nested virtualization, in which containers operate within virtual machines, offers enhanced isolation but adds complexity. Advanced techniques such as shadow paging and lightweight hypervisors seek to reconcile security with performance [12,13].

The future of cloud-native architectures depends on integrating emerging technologies like WebAssembly and edge computing. These advancements are expected to expand containerized applications beyond conventional cloud data centers, promoting innovation in 5G and IoT [14].

3. VMware vSphere with Tanzu

VMware vSphere with Tanzu is a sophisticated platform that allows vSphere administrators to convert their environments into infrastructures compatible with Kubernetes. This functionality enables developers and administrators to create and manage Kubernetes workload entities on current ESXi hosts, including pods and clusters. Integrating vSphere with Tanzu enables administrators to monitor and manage these environments utilizing familiar tools such as the vSphere web interface. This transformation signifies the changing responsibilities of vSphere administrators, progressively integrated with DevOps and DevSecOps methodologies.

The vSphere with Tanzu architecture comprises several essential components. The vSphere Namespace, comparable to Kubernetes Namespaces, guarantees resource isolation and fair allocation of resources such as CPU, memory, and storage within Kubernetes clusters. This functionality facilitates multi-tenancy, an essential criterion for contemporary cloud computing. The Namespace delineates the virtual machine (VM) images to create TKG (Tanzu Kubernetes Grid) clusters. Importantly, it allows Kubernetes pods and VMs to coexist on the same node, eliminating the need for dedicated clusters. The architecture of vSphere with Tanzu solution is shown in Figure 1:

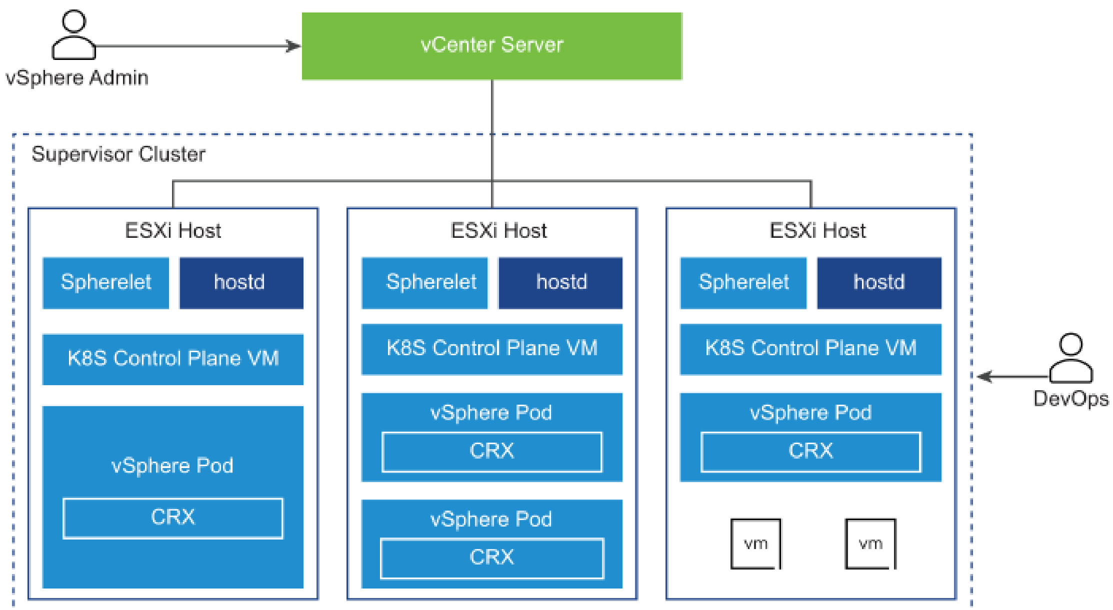

Figure 1.
vSphere with Tanzu Architecture (https://docs.vmware.com/en/VMware-vSphere/7.0/vmware-vsphere-with-tanzu/GUID-3E4E6039-BD24-4C40-8575-5AA0EECBBBEC.html).

The architecture presented in Figure 1 clearly shows what this platform is meant to be used for - it's for smaller to mid-size businesses that want to leverage the capabilities of Kubernetes on top of their existing VMware vSphere-based infrastructure. This means that Kubernetes-based features must be integrated in a user-friendly and accessible way. This was done by integrating with vSphere Client - the central management UI for environments that use vSphere as a virtualization solution. This solution can be integrated with NSX, but it requires a lot of time and effort - it's not as integrated as the TKG solution, especially when configured via VMware Cloud Foundation.

The Supervisor Cluster is the primary control plane, analogous to Kubernetes control nodes. It comprises fundamental Kubernetes components, including etcd, API servers, and various VMware-specific extensions such as CNI (Container Network Interface) and CSI (Container Storage Interface). The Supervisor Cluster, configured through the vSphere console, utilizes vSphere functionalities, including Distributed Resource Scheduler (DRS) and High Availability (HA) for node establishment and durability. The ESXi hosts operate as worker nodes, with the Spherelet Agent—a VMware modification of Kubernetes' kubelet—overseeing container lifecycles, network configurations, and storage mounts.

A distinctive aspect of the vSphere with Tanzu is the vSphere Pod, a specialized Kubernetes pod that functions on ESXi hosts. These pods utilize CRX (Container Runtime for ESXi), offering a lightweight, virtual machine-like instance. CRX instances are administered by the Spherelet Agent and provide two modes: managed (visible in the vSphere console) and unmanaged (functioning like a standard application or daemon). Nonetheless, conventional VMware functionalities such as vMotion, utilized for virtual machine migrations, are inaccessible for pods. Kubernetes features such as Deployments and StatefulSets guarantee availability.

The Tanzu Kubernetes Grid (TKG) Service is fundamental to this architecture, facilitating the creation and management of TKG clusters. TKG is entirely compatible with open-source Kubernetes, facilitating seamless functionality across platforms, including VMware Cloud on AWS, Azure VMware Solution, and Google Cloud VMware Engine. The Supervisor Cluster frequently functions as the management cluster for TKG, enabling direct control over infrastructure.

vSphere with Tanzu encompasses tools designed to streamline operations. The Namespace Service enables developers and platform administrators to create namespaces, alleviating administrative burdens independently. Configurable via a wizard in the vSphere console, this service allows resource allocation and access control without requiring administrator intervention. The vSphere VM Service facilitates VM creation through Kubernetes YAML manifests, providing predefined and customizable VM classes for TKG clusters.

VMware vSphere with Tanzu incorporates Kubernetes capabilities into vSphere environments, facilitating the efficient management of virtual machines and Kubernetes workloads. The platform integrates traditional VMware strengths with Kubernetes capabilities, providing a robust hybrid cloud-native solution for contemporary IT requirements. This paper further explores the practical implementation of these features in existing infrastructures.

4. Tanzu Kubernetes Grid

Tanzu Kubernetes Grid is VMware's comprehensive Kubernetes platform designed to simplify the deployment, operation, and management of Kubernetes clusters across multiple environments, including on-premises, public clouds, and hybrid setups. TKG provides a consistent, enterprise-grade Kubernetes runtime, ensuring organizations can deploy and scale containerized applications while maintaining control and flexibility. Built on open-source Kubernetes, TKG integrates deeply with VMware's ecosystem, including vSphere, NSX, and vSAN, to offer a seamless experience for IT administrators and developers.

A standout feature of TKG is its multi-cloud capability, allowing organizations to create and manage Kubernetes clusters across diverse infrastructure providers, such as AWS, Azure, and Google Cloud, as well as on-premises vSphere environments. This flexibility is critical for businesses aiming to modernize their applications while leveraging existing investments in infrastructure. TKG includes Tanzu Kubernetes Grid Integrated Edition (TKGI) for enhanced capabilities like workload balancing, persistent storage integration, and network policy enforcement. The basic architecture of TKG is shown in Figure 2:

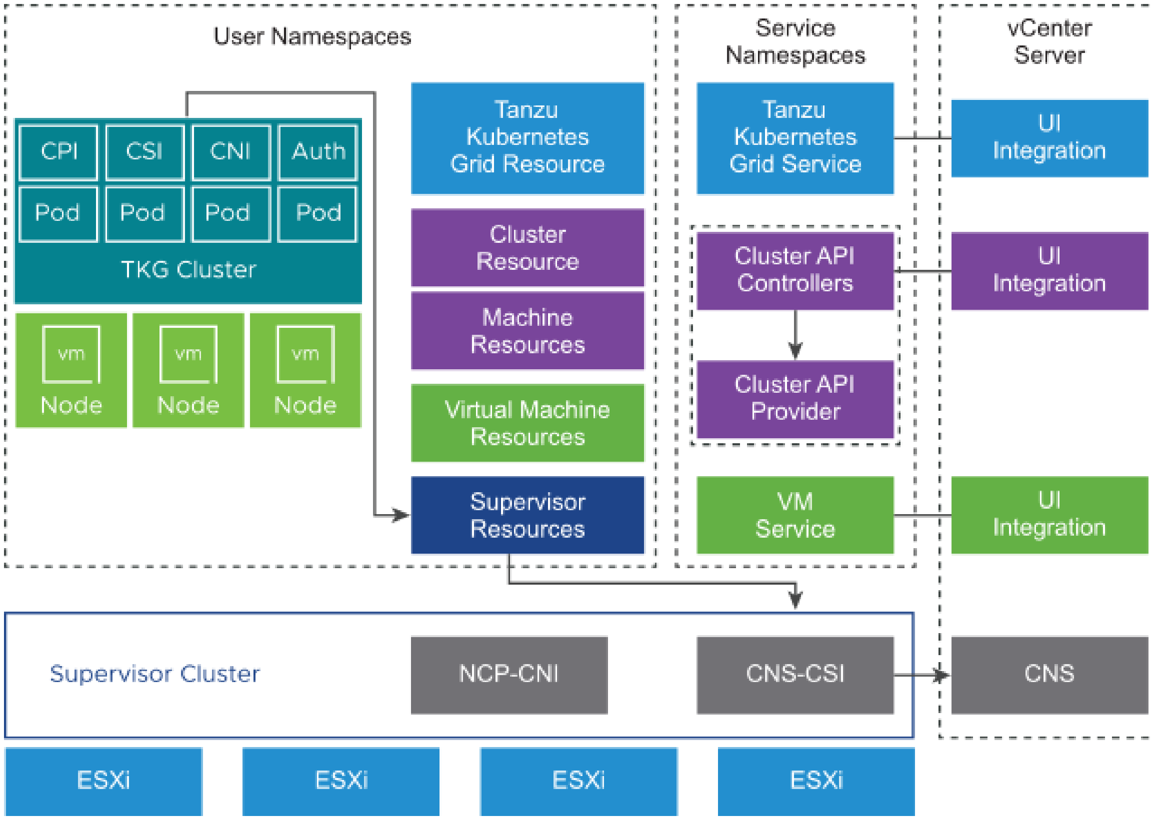

Figure 2.
Tanzu Kubernetes Grid Architecture (https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-with-tanzu-concepts-planning/GUID-4D0D375F-C001-4F1D-AAB1-1789C5577A94.html).

TKG's architecture is designed to streamline Kubernetes deployment by automating many aspects of cluster lifecycle management. This includes provisioning, scaling, updating, and monitoring clusters. Tools like the Tanzu CLI offer simplified commands for creating and managing clusters, while the Tanzu Mission Control platform provides centralized governance, policy enforcement, and monitoring across all clusters. These features empower IT teams to adopt Kubernetes at scale without extensive manual configuration or specialized expertise.

Furthermore, the architecture shown in Figure 2 suggests that TKG-based setups are being used for much more elaborate, scaled-out, larger environments. TKG is a solution for organizations requiring scalable, multi-cloud deployments equipped with advanced networking, persistent storage, and centralized management. TKG facilitates hybrid and multi-cloud environments, providing substantial flexibility for intricate, large-scale operations. Conversely, vSphere with Tanzu serves smaller organizations or those desiring streamlined Kubernetes integration, which is suitable for environments that emphasize simplicity and resource efficiency without necessitating enterprise-level functionalities.

One of TKG's significant strengths is its tight integration with VMware's infrastructure solutions. For example, the platform uses VMware vSphere to simplify node provisioning and resource management, NSX for advanced networking capabilities, and VMware Cloud Native Storage (CNS) for dynamic, policy-based storage provisioning. This integration allows enterprises to leverage their VMware environment while gaining Kubernetes' agility and scalability.

TKG supports a modular architecture, enabling organizations to incorporate additional components like Harbor for container image management, Fluentd for logging, and Prometheus and Grafana for monitoring and observability. It also supports Kubernetes-native tools, providing flexibility for

ISSN: 2576-8484
Vol. 8, No. 6: 6557-6578, 2024
DOI: 10.55214/25768484.v8i6.3409

developers who prefer open-source options for continuous integration and deployment (CI/CD) pipelines.

The platform addresses critical enterprise needs, including security, scalability, and reliability. It offers built-in features such as role-based access control (RBAC), secure ingress/egress traffic management, and automated backup and disaster recovery options. Furthermore, TKG's scalability ensures that organizations can accommodate increasing workloads and user demands, making it a reliable choice for enterprises operating on a scale.

By incorporating VMware's robust virtualization and cloud-native technologies, TKG provides a unified platform bridging the gap between traditional and modern containerization. This makes it an ideal choice for organizations undergoing digital transformation, enabling them to adopt Kubernetes without discarding their VMware investments. TKG's flexibility, robust integrations, and enterprise-grade features make it a cornerstone of VMware's Tanzu portfolio, helping businesses thrive in the cloud-native era.

5. Differences Between Vsphere with Tanzu and TKG

Although both vSphere with Tanzu and Tanzu Kubernetes Grid are essential elements of VMware's Tanzu portfolio, they address distinct use cases and deployment contexts.

vSphere with Tanzu incorporates Kubernetes functionalities directly into VMware vSphere settings, enabling administrators to execute Kubernetes workloads in conjunction with conventional virtual machines on the same infrastructure. It employs vSphere namespaces for resource distribution and offers tools such as the Supervisor Cluster, which converts vSphere servers into Kubernetes nodes. This strategy is optimal for enterprises aiming to enhance their vSphere investments within the Kubernetes ecosystem.

Tanzu Kubernetes Grid provides a more extensive Kubernetes solution that transcends vSphere. TKG offers a uniform Kubernetes runtime across multi-cloud settings, allowing enterprises to deploy clusters on public clouds such as AWS and Azure and on-premises infrastructure. In contrast to vSphere with Tanzu, TKG emphasizes flexibility and multi-cloud interoperability, rendering it appropriate for organizations overseeing varied environments.

The primary distinction is in scope: vSphere with Tanzu is vSphere-centric and closely integrated with VMware's virtualization framework. TKG is a more expansive Kubernetes platform for multi-cloud operations and large-scale Kubernetes cluster management.

6. VMware Cloud Native Storage

Cloud Native Storage (CNS) is a VMware vSphere functionality intended to incorporate containerized workloads with vSphere's comprehensive storage framework effortlessly. The demand for persistent and dependable storage solutions has become critical with the growing adoption of container orchestration technologies such as Kubernetes. CNS resolves this issue by facilitating the dynamic provisioning of storage on vSphere using Kubernetes in an automated and scalable fashion. It also offers administrators improved visibility and management functionalities through the vSphere Client.

The CNS fundamentally consists of two main elements: the vSphere Container Storage Interface (CSI) plugin and the CNS control plane integrated into the vCenter Server. The CSI plugin is an intermediary between Kubernetes and vSphere, enabling volume provisioning, attachment, detachment, and lifecycle management of persistent volumes. This integration allows Kubernetes to utilize vSphere's storage primitives, such as standard volumes, persistent volumes, and dynamic provisioning, thereby assuring that stateful containerized applications may retain data during restarts and outages.

The CNS control plane in the vCenter Server introduces container volumes, managing their lifespan independently from virtual machines. This separation guarantees container volumes possess unique identities within vSphere, facilitating precise management and monitoring. Administrators can employ Storage Policy-Based Management (SPBM) to establish storage policies corresponding to application specifications. These policies can be immediately implemented on container volumes, guaranteeing compliance with designated performance and availability standards.

A notable benefit of CNS is its compatibility with several vSphere storage backends, such as vSAN, Virtual Volumes (vVols), VMFS, and NFS. This flexibility allows enterprises to utilize their storage investments while providing a uniform platform for virtual machines and containerized applications. With the introduction of vSphere 7.0 and the CSI 2.0 driver, vVols were supported as a storage mechanism for CNS, facilitating enhanced storage management and scalability.

The integration of CNS and vSphere provides numerous significant advantages:

- Automated Storage Provisioning: CNS enables Kubernetes to request and provision storage dynamically, minimizing the need for manual intervention and lowering the risk of errors.
- Improved Visibility: The vSphere Client enables managers to obtain information regarding container volumes, encompassing their health, adherence to storage policies, and connections to foundational storage objects.
- Consolidated Management: CNS facilitates virtual machine and container storage administration on a singular platform, optimizing operations and diminishing complexity.
- Scalability: Considered to accommodate container systems' elevated churn rates, CNS guarantees that storage operations maintain efficiency and performance as workloads expand.

VMware's Cloud Native Storage effectively connects classic virtualization with modern containerized applications, offering a robust, scalable, and integrated storage solution that satisfies the requirements of current IT settings.

7. Comparison with Competitive Storage Architectures

Implementing containerized apps and Kubernetes has necessitated the demand for resilient, scalable, and adaptable storage solutions to handle stateful workloads. VMware's Cloud Native Storage (CNS) seamlessly integrates with the vSphere ecosystem, delivering powerful storage functionalities to Kubernetes. Nevertheless, CNS is not the exclusive alternative available in the market. Alternative solutions such as OpenShift Container Storage (OCS), Portworx by Pure Storage, and Ceph (via Rook) address the storage requirements of Kubernetes systems. Each solution possesses distinct advantages and challenges, rendering them appropriate for various situations based on an organization's architecture, needs, and goals.

OpenShift Container Storage (OCS) is Red Hat's Kubernetes storage solution, intricately connected with the OpenShift platform. OCS, constructed on technologies like Ceph and NooBaa, offers a comprehensive range of functionality, including dynamic provisioning, snapshots, and multi-cloud object storage. The seamless interface with OpenShift provides a substantial advantage, facilitating straightforward deployment and control from the OpenShift console. Nonetheless, its dependence on Ceph, albeit resilient, may introduce a degree of complexity regarding installation and upkeep, especially for businesses needing more prior familiarity with Ceph. Moreover, licensing expenses for OCS may be a factor for enterprises already invested in alternative platforms.

Portworx by Pure Storage is a cloud-native storage solution recognized for its platform-agnostic characteristics, providing interoperability with various Kubernetes releases and underlying infrastructures. Portworx delivers enterprise-level functionalities, including snapshots, encryption, disaster recovery, and multi-cloud compatibility. Its adaptability makes it an optimal selection for enterprises operating in hybrid or multi-cloud environments. Even with this, Portworx's subscription-based license may be too expensive for smaller enterprises or those with constrained storage needs. Moreover, although very proficient, its sophisticated features may necessitate a more challenging learning curve for teams unacquainted with its ecosystem.

Ceph, through the Rook operator, is a widely utilized open-source storage system that connects with Kubernetes to deliver highly scalable and distributed storage capabilities. Ceph's ability to provide block, object, and file storage inside a unified platform is one of its distinguishing characteristics. The Rook operator simplifies Ceph management within Kubernetes, facilitating dynamic provisioning and seamless integration. Ceph, as an open-source solution, has the benefits of cost-effectiveness and extensive customization. Nonetheless, its operational intricacy may be disadvantageous, especially for smaller entities needing more specialized storage knowledge. Optimizing performance and diagnosing issues in Ceph can be arduous, particularly in extensive installations.

A summarized table of comparison between these competitive products is presented in Table 1:

Table 1.
Comparison of VMware CNS with other Kubernetes storage solutions.

Feature/Capability	VMware cloud native storage (CNS)	OpenShift container storage (OCS)	Portworx by pure storage	Ceph (via Rook)
Integration with platform	Native integration with vSphere	Integrated with OpenShift	Platform-agnostic	Platform-agnostic
Supported storage backends	vSAN, vVols, VMFS, NFS	Ceph, NooBaa, Red Hat Gluster	Various cloud and on-premises storage	Ceph
Dynamic provisioning	Yes	Yes	Yes	Yes
Storage policy management	SPBM integration	StorageClass definitions	StorageClass definitions	StorageClass definitions
Data services	Snapshots, Cloning	Snapshots, Cloning, Tiering	Snapshots, Encryption, DR	Snapshots, Cloning
Scalability	High, leveraging vSphere's capabilities	High, designed for OpenShift	High, designed for large-scale deployments	High, scalable architecture
Management interface	vSphere Client	OpenShift Console	Portworx Central	Kubernetes Dashboard
Licensing model	Part of vSphere licensing	Requires separate subscription	Subscription-based	Open-source

VMware Cloud Native Storage (CNS) is distinguished by its inherent integration with the vSphere environment, rendering it an optimal selection for enterprises already committed to VMware's infrastructure. CNS utilizes vSphere's robust storage infrastructure, encompassing vSAN, vVols, VMFS, and NFS, guaranteeing compatibility with current storage assets. Features such as Storage Policy Based Management (SPBM), detailed monitoring, and integrated management via the vSphere Client render CNS a formidable solution for companies pursuing efficiency and performance. Nonetheless, CNS's reliance on VMware's ecosystem may provide a constraint for enterprises with heterogeneous infrastructure or those functioning beyond the VMware environment.

8. Tanzu Mission Control Platform

Tanzu Mission Control (TMC) is an essential platform within VMware's Tanzu portfolio, aimed at streamlining the administration of Kubernetes clusters and containerized applications across various cloud environments, including public cloud providers like AWS, Azure, and GCP, as well as private infrastructures. TMC is predominantly available as a SaaS within VMware Cloud, though it can also be implemented on-premises. It enables lifecycle management activities, including creating, deleting, upgrading, and scaling Kubernetes clusters. TMC offers centralized oversight, allowing administrators to assess the status of clusters, namespaces, nodes, and workloads from a unified console. This visibility guarantees prompt responses to problems such as resource overutilization or node malfunctions. TMC also encompasses tools for compliance verification, security administration, and backup and recovery processes.

The architecture of TMC is centered on hierarchical resource organization. At the highest level are organizational entities associated with business units or geographical regions. Below are cluster groups that consolidate Kubernetes clusters, providing an infrastructural viewpoint and workspaces that organize namespaces for application-centric management. This framework facilitates the application of access policies, as policies designated to a group are automatically disseminated to its resources. TMC additionally facilitates API-driven cluster management via command-line interfaces, augmenting its adaptability for diverse operational requirements. The architecture of TMC can be seen in Figure 3:

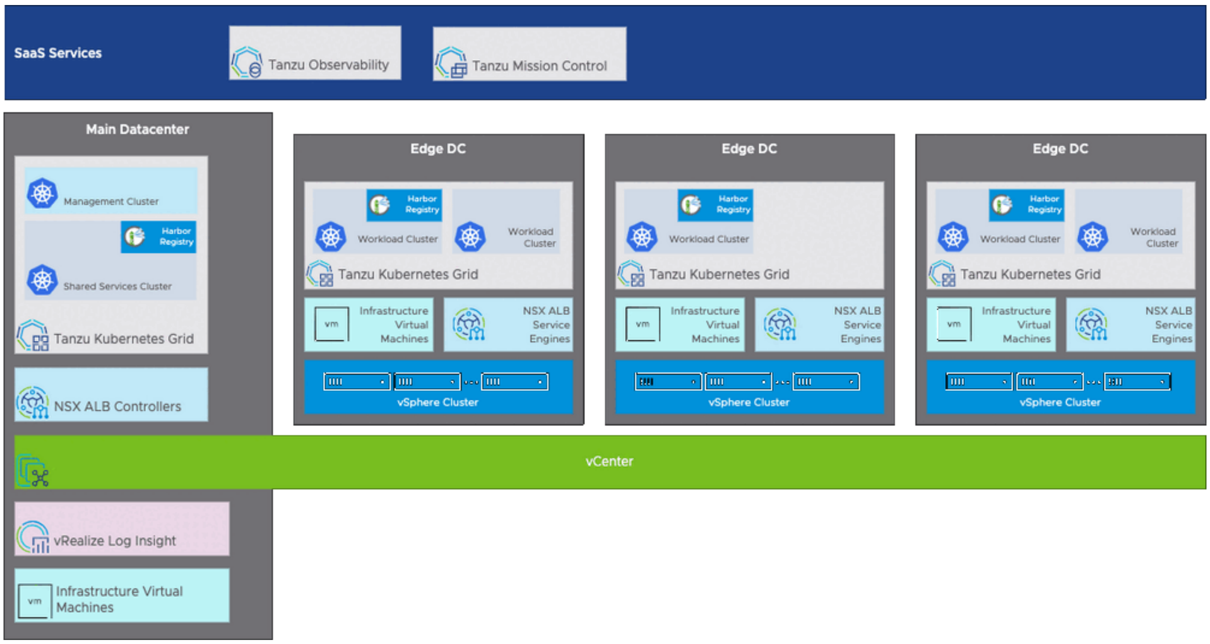

Figure 3.
Tanzu Mission Control architecture concepts (https://docs.vmware.com/en/VMware-Tanzu-for-Kubernetes-Operations/1.0/edge-reference-architecture/GUID-reference-designs-tkg-edge.html).

TMC integrates effortlessly with current vSphere configurations in VMware environments. A standard configuration comprises vSphere clusters that host multiple virtual machines, including those for Supervisor clusters, TKG clusters, and NSX Advanced Load Balancer (ALB) instances. NSX ALB provides automated load balancing for Kubernetes clusters, utilizing components such as the controller and service engines to manage virtual services and route traffic between nodes and applications. This integration facilitates effective management and scalability of Kubernetes resources within vSphere environments.

Day 2 operations focus on maintaining, monitoring, and scaling Kubernetes clusters post-deployment. These tasks encompass node scaling, log monitoring, backup, recovery, and cluster upgrades. Kubernetes offers three scaling methodologies: automatic, horizontal, and vertical. Automatic scaling modifies node quantities according to resource demand or underutilization, whereas horizontal scaling necessitates manual modifications through CLI tools, YAML manifests, or management consoles. Vertical scaling adjusts the resource distribution of nodes, necessitating predetermined VM classes for effective operation. TMC facilitates operational scaling by allowing modifications via its console or API.

Cluster upgrades are essential for ensuring compatibility with new features and preserving security. The upgrade process commences with updating management clusters, followed by workload clusters. For TKG clusters, this entails downloading updated virtual machine templates and implementing them using tools such as Tanzu CLI or management consoles. TMC offers an intuitive interface for initiating upgrades and optimizing the process while ensuring cluster availability.

Monitoring is essential for managing Kubernetes clusters. VMware's vRealize Log Insight tool provides extensive logging and analytical functionalities. It integrates effortlessly with TKGI and vSphere environments, collecting logs from clusters, nodes, and applications. Establishing Log Insight requires deploying its virtual appliance and configuring integration parameters, including SSL certificates and endpoint specifications. Manual configurations, including installing Fluentd and ConfigMap objects, may be necessary for non-VMware platforms or Kubernetes distributions.

TMC enhances its capabilities by incorporating supplementary monitoring and metrics tools, providing administrators with the necessary insights to optimize resource utilization and application performance. TMC centralizes operations across cloud environments, offering a unified framework for managing Kubernetes clusters and facilitating the scalability and flexibility crucial for contemporary cloud-native applications. Through its robust architecture, lifecycle management capabilities, and monitoring tools, TMC effectively addresses the complexities of multi-cloud Kubernetes operations.

9. NSX Integration

The NSX virtualization platform is a comprehensive VMware solution that expands the software-defined networking (SDN) framework to facilitate efficient, flexible, and automated management of network resources. This discussion centers on integrating NSX with Tanzu technology, emphasizing architecture, fundamental concepts, and network security features.

10. NSX Architecture

VMware NSX is a software-defined networking (SDN) platform engineered to provide sophisticated network virtualization and security for contemporary data centers, cloud infrastructures, and edge sites. NSX's architecture abstracts network infrastructure into a versatile, scalable software layer, allowing enterprises to manage their networks dynamically and programmatically, improving agility, automation, and security.

The NSX design comprises three fundamental layers: Management, Control, and Data Planes. Each executes specific functions to facilitate uninterrupted network operations:

a. The Management Plane offers a centralized interface for configuring, monitoring, and administering the whole NSX infrastructure. The NSX Manager is the central hub for all configuration tasks, including defining virtual network topologies, security rules, and operational settings. Administrators engage with this platform using the NSX Manager UI, REST APIs, or integration with external technologies. The Management Plane retains configurations and transmits them to the Control Plane.
b. The Control Plane manages network intelligence and the dynamic configuration of network topologies. It sustains a comprehensive network perspective by calculating routes, disseminating policies, and administering logical switches and routers. The NSX Controller, an essential element of this layer, guarantees consistency by disseminating configurations and state information to the Data Plane. This layer functions in a distributed manner, guaranteeing scalability and resilience.
c. The Data Plane is the component where actual network traffic transits and regulations are implemented. It comprises NSX-enabled transport nodes, typically hypervisors (e.g., ESXi hosts), routers, and switches. NSX executes network functions at this layer, including distributed routing, load balancing, firewalling, and NAT, utilizing a distributed design to reduce bottlenecks and latency. Transport nodes utilize virtual switch technology for packet forwarding management, exemplified by the NSX vSwitch.

The basic architecture of the NSX platform is shown in Figure 4:

ISSN: 2576-8484
Vol. 8, No. 6: 6557-6578, 2024
DOI: 10.55214/25768484.v8i6.3409

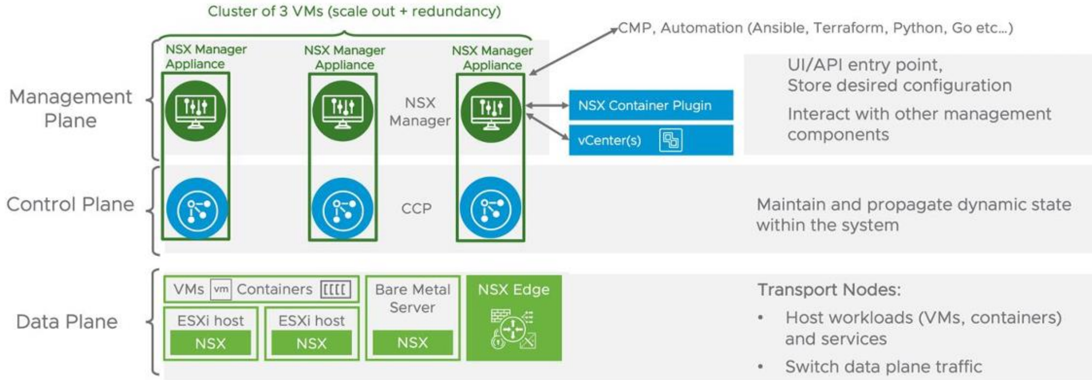

Figure 4.
NSX architecture (VMware NSX Reference Design Guide, https://community.broadcom.com/viewdocument/nsx-reference-design-guide-42-v10?CommunityKey=b76535ef-c5a2-474d-8270-3e83685f020e&tab=librarydocuments).

The NSX architecture is fundamental to VMware's Tanzu environments, offering a software-defined networking framework that abstracts physical network infrastructure into a versatile, programmable layer. In Tanzu environments, NSX integrates effortlessly to facilitate Kubernetes workloads in conjunction with conventional virtual machines. NSX facilitates workload isolation and secure traffic management through logical switches for Layer 2 segmentation and logical routers for inter-network communication. The distributed firewall guarantees micro-segmentation by implementing specific security policies for east-west traffic within the environment. Furthermore, NSX's load-balancing functionalities enhance application performance, ensuring high availability and resilience. The NSX Container Plugin (NCP) functions as Kubernetes's Container Network Interface (CNI), automating network provisioning for Tanzu Kubernetes Grid clusters, allocating IP addresses, and efficiently routing services. The profound integration of NSX with Tanzu environments establishes a cohesive platform for administering containerized and virtualized workloads, guaranteeing scalability, robust security, and operational adaptability.

The platform facilitates "north-south" traffic (between NSX and external networks) and "east-west" traffic (within the NSX domain). Logical segmentation enables precise control over network traffic.

11. NSX Logical Elements

NSX transforms physical network infrastructure into logical entities, encompassing:

a. Logical Switches are virtual network segments replicating actual Layer 2 switches, facilitating isolated traffic flows among virtual machines and applications.
b. Logical Routers: Layer 3 devices facilitate routing between logical switches and external networks, enabling dispersed routing for enhanced performance and scalability.
c. Distributed Firewall (DFW): A hypervisor-integrated firewall that precisely regulates east-west traffic, facilitating micro-segmentation and zero-trust security protocols.
d. Load Balancers: These services allocate traffic effectively among application instances to enhance performance and availability.

NSX logical elements—logical switches, routers, the Distributed Firewall, and load balancers—revolutionize networking by abstracting physical infrastructure into scalable and adaptable components. They facilitate effective traffic management, detailed security, and robust availability, rendering NSX essential for contemporary IT ecosystems.

12. Security Capabilities

Security is fundamental to NSX architecture, featuring characteristics such as:

a. Micro-segmentation: This technique enables fine-grained traffic control at the workload level by separating apps and imposing security policies directly on virtual machines. It reduces the attack surface and enhances east-west traffic security.
b. The Distributed Firewall (DFW) is integrated at the hypervisor level, enforcing Layer 2-7 policies across workloads and providing low-latency security without the need for traffic redirection. It facilitates application-level controls, encompassing URL filtering and deep packet inspection.
c. Intrusion Detection/Prevention System (IDS/IPS): This system detects and addresses attacks in real-time by analyzing traffic for recognized vulnerabilities, atypical behaviors, and zero-day exploits.
d. Service Insertion: This process incorporates external security solutions, such as antivirus or sophisticated threat detection systems, into the NSX environment for improved protection.
e. Endpoint Protection: Safeguards virtual workloads by integrating them with endpoint security services, providing thorough defense at the origin.
f. Security Groups and Tags: Automates policy enforcement through the dynamic aggregation of workloads based on attributes, assuring uniform security as environments expand.
g. Traffic Monitoring and Traceflow: Evaluates traffic patterns to identify abnormalities and resolve network faults, facilitating proactive security management.

NSX enables organizations to manage networks with agility, expand economically, and improve security by detaching networking services from physical hardware, thereby becoming an essential element of contemporary IT architecture.

13. Integration with Kubernetes Utilizing CNI Solutions

Within Kubernetes, NSX integrates through the NSX Container Plugin (NCP), a CNI (Container Network Interface) driver. NCP automates the creation of NSX logical network topologies, assigns IP addresses, implements security policies, and provides Kubernetes services like load balancing. Other supported CNIs include Antrea, which relies on Open vSwitch for network configuration, and Calico, which is known for its enhanced security features and efficient traffic routing.

Antrea enables container communication within and between nodes utilizing protocols such as GRE and VXLAN. Conversely, Calico facilitates native routing that is devoid of encapsulation, thereby minimizing overhead. NCP distinguishes itself using NSX's sophisticated networking features, including distributed firewalls and network address translation (NAT). Nonetheless, its intricacy frequently results in the preference for more straightforward CNI solutions in various settings.

14. Case Studies – vSphere with Tanzu and Tanzu Kubernetes Grid

This section examines three unique methodologies for implementing vSphere with Tanzu and Tanzu Kubernetes Grid in VMware-based environments, emphasizing each strategy's practical problems and advantages. The initial case study analyzes brownfield deployment, wherein an existing VMware infrastructure is augmented to accommodate containerized apps, optimizing current investments while integrating Kubernetes. The second emphasizes greenfield deployment, meticulously crafted to effortlessly accommodate virtual machines and contemporary container workloads. These scenarios offer insights into the design of efficient, scalable, and future-proof IT systems.

14.1. Case Study 1: Brownfield Deployment of vSphere with Tanzu

Upgrading IT infrastructure has become essential for organizations adapting to fast-emerging technology and business models. For enterprises already adopting VMware vSphere for virtualization, introducing container-based workloads frequently marks the next stage in digital transformation. A brownfield deployment entails upgrading an existing VMware environment to accommodate contemporary application architectures while maintaining uninterrupted operations. This strategy is especially appealing for enterprises seeking to adopt Kubernetes and containerized apps while preserving their current investments in virtualized infrastructure.

This case study examines a mid-sized organization that predominantly utilizes VMware technologies to virtualize its essential applications, including ERP systems, web servers, and internal productivity tools. These apps operate on numerous virtual machines administered using vSphere, offering the firm a resilient and scalable platform. The company aims to implement container-based microservices to assist its software development teams and expedite the time-to-market for new features. This shift involves careful integration of Kubernetes into its vSphere ecosystem, leveraging VMware's vSphere with Tanzu to build a unified environment where containers and virtual machines coexist harmoniously.

The organization's principal objectives encompass preserving operational continuity, reducing disruption to mission-critical systems, and enabling its IT teams to navigate the new environment without necessitating extensive retraining. The company aims to optimize the value of its current investments in VMware infrastructure, including vSAN for storage and NSX for networking. These objectives provide a brownfield deployment of vSphere with Tanzu a rational selection.

This deployment presents distinct problems and opportunities. The present VMware environment must be carefully analyzed to verify that it has sufficient capacity and is compatible with the requirements of vSphere with Tanzu. Networking and security policies must be modified to accommodate Kubernetes workloads while preserving the isolation and performance of current virtual machines. Furthermore, IT personnel must acquire expertise in administering Kubernetes with their conventional virtualization responsibilities, resulting in a significant yet surmountable learning curve. This brownfield implementation can be a model for integrating contemporary application platforms into existing virtualized environments.

Steps in the process:

1. Current infrastructure evaluation
   a. Perform a thorough evaluation of the existing VMware infrastructure, encompassing vSphere, networking, storage, and computational capacity.
   b. Assess the capacity of current resources to accommodate Kubernetes workloads without degrading the performance of existing virtual machines.
2. Networking Integration
   a. Employ VMware NSX to establish a cohesive networking layer for virtual machines and Kubernetes workloads.
   b. Formulate network policies to segregate workloads, guaranteeing security and compliance.
3. Activating vSphere with Tanzu
   a. Enable vSphere with Tanzu on the current infrastructure, utilizing the vSphere Distributed Resource Scheduler (DRS) and High Availability (HA) clusters for workload distribution and reliability.
   b. Implement Tanzu Kubernetes Grid on the vSphere platform for container orchestration.
4. Workload re-balancing
   a. Commence with non-essential workloads to assess the integration and performance of vSphere with Tanzu.
   b. Systematically transition containerized workloads to Kubernetes while maintaining seamless compatibility with current virtual machines.

There are some advantages to this approach:

1. Current investments can be leveraged: There is no requirement for significant initial expenditures on new gear or software licenses.
2. Easier transition: Co-locating virtual machines and containers reduces interference with current operations.
3. Familiar ecosystem: Administrators may oversee Kubernetes and VM workloads using the vSphere Client, thereby minimizing the learning curve.

In terms of disadvantages:

1. Resource contention risks: Virtual machines and Kubernetes workloads may compete for resources if the infrastructure is inadequately scaled.
2. Integrating NSX for Kubernetes may complicate setups needing more prior NSX experience.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6557-6578, 2024
DOI: 10.55214/25768484.v8i6.3409

3. Legacy limitations: Outdated hardware or software versions may restrict the functionalities of vSphere with Tanzu.

Let's now discuss the opposite approach, the greenfield deployment.

14.2. Case Study 2: Greenfield Deployment of vSphere with Tanzu

When initiating the design of an IT infrastructure from the ground up, organizations possess a distinctive chance to establish an environment tailored for both present requirements and future expansion. In contrast to brownfield deployments, greenfield projects allow the implementation of advanced technology and optimal processes unencumbered by pre-existing systems. This method is particularly advantageous for startups or organizations undertaking extensive digital transformation projects, where scalability, adaptability, and contemporary application architectures are crucial.

In this case study, a software firm intends to develop its IT infrastructure to serve a mix of traditional corporate applications and containerized workloads. The firm aims to establish a platform to support virtual machines for ERP, CRM, various business applications, and Kubernetes-based environments for microservices and DevOps pipelines. To realize this objective, the company chose VMware's vSphere with Tanzu, a comprehensive solution that enables Kubernetes and virtual machines to work side-by-side on a single infrastructure.

The objectives for greenfield deployment encompass creating a scalable and highly available platform, optimizing resource consumption, and using contemporary software-defined networking and storage solutions. Beginning from scratch, the organization can customize its environment to maximize the functionalities of vSphere with Tanzu, integrating VMware NSX for sophisticated networking and VMware vSAN for policy-based storage. This strategy reduces compromises and guarantees the infrastructure is optimal for present workloads and expected future expansion.

Nonetheless, greenfield deployments present considerable problems. In contrast to brownfield scenarios, where IT professionals can utilize their familiarity with existing systems, greenfield projects necessitate the establishment of all components from the ground up. This comprises gear acquisition, network design, software installation, and staff training. Furthermore, the firm must meticulously oversee its budget and schedules, as greenfield deployments typically entail more extraordinary initial expenses and extended implementation durations relative to brownfield integrations.

The firm can leverage a contemporary, efficient, and future-oriented IT infrastructure by adopting a greenfield deployment plan. This case study analyzes the design, implementation, and operational aspects of developing an infrastructure that exploits the capabilities of vSphere with Tanzu while assuring seamless coexistence of virtual machines and Kubernetes workloads.

Steps in the process:

1. Infrastructure development strategy
   a. Construct the physical infrastructure utilizing high-performance servers, networking apparatus, and storage systems to fulfill the demands of both virtual machines and Kubernetes.
   b. Choose VMware vSAN for storage to streamline management and facilitate storage policy-based provisioning.
2. Network Architecture
   a. Deploy NSX from inception to facilitate software-defined networking for containerized and virtualized workloads.
   b. Define network segmentation, load balancing, and ingress/egress policies specific to Kubernetes and virtual machine traffic.
3. Implementation of vSphere and Tanzu:
   a. Install the most recent version of vSphere with Tanzu, utilizing its inherent functionalities to establish Supervisor Clusters.
   b. Configure TKG to operate concurrently with conventional virtual machines, guaranteeing a scalable and adaptable implementation.
4. Operational Preparedness:
   a. Educate personnel on administering vSphere with Tanzu to address Kubernetes-specific needs.

Deploy monitoring and logging tools like VMware vRealize Suite to provide insights into workload and infrastructure performance.

There are some advantages to this approach:

1. Optimized for Scalability: Infrastructure is developed from conception to accommodate VMs and Kubernetes workloads.

2. Streamlined Networking: NSX is integrated from inception, eliminating the necessity for retrofitting.

3. Future-Ready Design: The environment is engineered to facilitate the seamless integration of advanced features and updates.

Also, there are some disadvantages:

1. Higher Initial Costs: Requires significant upfront investment in hardware, software, and training.
2. Increased Learning Curve: Teams may encounter difficulties acclimating to new tools and practices.
3. Prolonged Setup Duration: Deployment may require additional time due to the necessity for comprehensive configuration.

The case studies of vSphere integration with Tanzu in brownfield and greenfield deployments demonstrate two distinct methodologies for updating IT infrastructure. The brownfield deployment illustrates how enterprises with established VMware systems may augment their capabilities by integrating Kubernetes with virtual machines, optimizing existing investments, and reducing disruption. This technique may encounter obstacles, including resource congestion, compatibility issues with legacy systems, and the complexity of integrating current networking solutions, such as NSX, into existing infrastructure. Notwithstanding these obstacles, it remains a cost-efficient and pragmatic alternative for organizations already committed to VMware.

By comparison, greenfield deployments demonstrate the potential to establish a contemporary, scalable, and adaptable environment from inception. By architecting the infrastructure to accommodate both virtual machines and containers, greenfield deployments circumvent legacy limitations and enhance performance and scalability. Nonetheless, they necessitate considerable initial investment, time, and skill for implementation, rendering them more appropriate for firms with defined design objectives and adequate resources.

These approaches typically depend on an organization's initial position and priorities. A brownfield deployment is advantageous for individuals with an established VMware ecosystem aiming for a gradual upgrade. Conversely, greenfield deployments provide a future-oriented solution for novel environments, albeit at an elevated expense. Each option possesses distinct advantages, and the decision ultimately hinges on the equilibrium among operational efficiency, financial constraints, and long-term scalability.

14.3. Case Study 3: Tanzu Kubernetes Grid Greenfield Deployment

A mid-sized technology company aimed to upgrade its IT infrastructure to create a flexible, scalable platform that accommodates containerized applications and conventional workloads. The organization chose a greenfield deployment of VMware Tanzu Kubernetes Grid due to the absence of legacy systems. The aim was to create a Kubernetes environment seamlessly integrated with VMware's ecosystem, delivering enterprise-level functionalities for container orchestration, networking, and storage. The company utilized VMware NSX for software-defined networking and vSAN for dynamic storage provisioning, designing its infrastructure to enhance resource utilization and streamline management. The modular architecture of TKG and its compatibility with hybrid and multi-cloud environments were crucial to their strategy, guaranteeing the platform's adaptability to future business requirements. The greenfield approach enabled the implementation of contemporary DevOps practices, automation of CI/CD pipelines, and optimization of operational workflows from the beginning, thereby positioning the company for sustained success.

1. Infrastructure deployment strategy

a) Deployment of computational resources: VMware vSAN for storage, providing policy-driven provisioning and scalability
b) NSX deployment to implement software-defined networking: workload isolation and secure traffic management.

c) Implementing TKG clusters: The design emphasizes resource distribution between containers and virtual machines to facilitate application workloads and system services.

The design prioritizes redundancy, high availability, and scalability to accommodate changing requirements.

2. Network Architecture: NSX was pivotal to the networking strategy, offering a virtualized network layer that abstracted the physical infrastructure.

a) Logical switches segregated workloads into dedicated Layer 2 segments, whereas logical routers facilitated effective inter-network communication.
b) Distributed firewalls enable micro-segmentation, facilitating the application of precise security policies to east-west traffic within the data center.
c) The load balancing features of NSX were utilized to enhance application performance and guarantee high availability.
d) The NSX Container Plugin can incorporate Kubernetes, optimize networking for TKG clusters, and automate IP assignment and service routing.

3. TKG implementation: TKG utilized the Tanzu CLI to streamline cluster provisioning and administration.

a) A Supervisor Cluster functioned as the primary control plane, facilitating efficient management of all Kubernetes clusters.
b) Each cluster was established with role-based access control (RBAC) and namespaces to guarantee secure multi-tenancy.
c) Persistent storage was established utilizing VMware vSAN, whereas monitoring and observability were augmented with Prometheus and Grafana.
d) CI/CD pipelines were implemented utilizing Jenkins and Tanzu Build Service to expedite the application development lifecycle.

4. Operational Readiness: To guarantee preparedness, IT teams trained on VMware Tanzu tools, Kubernetes principles, and contemporary DevOps methodologies.

a) Monitoring solutions, such as VMware vRealize Log Insight and Velero for backups, were implemented to facilitate day-2 operations.
b) The automated scaling and resource optimization features were evaluated to confirm performance across workloads.
c) Comprehensive runbooks and disaster recovery protocols were established to mitigate potential disruptions.

There are some advantages to using TKG over vSphere for Tanzu:

1. Multi-Cloud Flexibility: TKG facilitates effortless Kubernetes deployment across hybrid and multi-cloud settings, encompassing AWS, Azure, and on-premises infrastructure, providing enterprises exceptional deployment flexibility.
2. Tight Integration with the VMware Ecosystem: TKG operates cohesively with VMware solutions such as NSX for sophisticated networking, vSAN for storage, and vSphere for resource management, optimizing operations, and augmenting functionality.
3. Enterprise-Level Security: Capabilities, including role-based access control (RBAC), micro-segmentation through NSX, and compatibility with external security tools, provide comprehensive safeguarding for Kubernetes workloads.
4. Centralized Management: Solutions like Tanzu Mission Control offer a unified interface for overseeing multiple clusters, implementing policies, and monitoring workloads across various environments.
5. Scalability and Performance: TKG facilitates dynamic workload and infrastructure scaling, guaranteeing high availability and performance in response to increasing enterprise demands.

Some disadvantages of this approach would be:

1. Substantial Initial Expenditures: The implementation necessitates considerable investment in hardware, software licenses, and training, potentially dissuading smaller enterprises.
2. Significant Learning Curve: IT teams require specialized expertise in Kubernetes and VMware-specific tools, resulting in heightened training duration and complexity.

3. Complex Deployment: The initial configuration, particularly in greenfield deployments, requires substantial expertise and can be protracted.
4. Dependence on the VMware Ecosystem: TKG's reliance on VMware products may result in vendor lock-in, constraining flexibility for organizations utilizing or transitioning to non-VMware platforms.
5. Resource Contention: Operating Kubernetes with conventional workloads on shared infrastructure necessitates meticulous resource allocation to prevent performance bottlenecks.

The greenfield implementation of Tanzu Kubernetes Grid exemplifies a progressive strategy for constructing a contemporary IT infrastructure designed for containerized applications and conventional workloads. Utilizing VMware's ecosystem, the organization effectively created a scalable, secure, and adaptable environment to meet future requirements. NSX facilitated sophisticated network virtualization, ensuring workload isolation, automated routing, and meticulous security via micro-segmentation. vSAN optimized storage administration, guaranteeing persistent and adaptive storage for Kubernetes workloads. The integration of TKG with VMware tools facilitated the establishment of a cohesive management strategy, streamlining cluster deployment and operations while promoting contemporary DevOps methodologies.

Notwithstanding its benefits, the greenfield approach presented difficulties, such as substantial initial expenses and a significant learning curve. Nonetheless, these were counterbalanced by the enduring advantages of a future-oriented, highly optimized infrastructure. This case study illustrates how TKG facilitates organizations' large-scale adoption of Kubernetes, utilizing VMware's established technologies to provide a robust platform for digital transformation. TKG delivers adaptability, security, and enterprise-level reliability for organizations undertaking comparable modernization initiatives essential for success in the current competitive environment.

15. Future Works

There are at least five exciting avenues for future research related to the topic of this paper. They are as follows:

1. Performance Optimization in Multi-Cloud Environments: Investigate methods to optimize the performance of VMware Tanzu deployments across hybrid and multi-cloud environments. This entails assessing latency, resource allocation, and workload distribution strategies to enhance runtime efficiency across various infrastructure configurations.
2. Security Improvements for Integrated Virtualization and Containerization: Investigate sophisticated security frameworks designed for VMware Tanzu's integration of vSphere, NSX, and Kubernetes. Research may concentrate on creating advanced intrusion detection and prevention systems or improved role-based access control methods for Tanzu's multi-tenant environments.
3. Comparative Analysis of Cost Efficiency: Perform a comprehensive cost analysis comparing VMware Tanzu with alternative Kubernetes distributions (e.g., OpenShift, Azure Kubernetes Service, Amazon Elastic Kubernetes Service) on total cost of ownership, encompassing licensing hardware prerequisites and operational expenditures across several enterprise scenarios.
4. User-Centric Enhancements in Management Interfaces: Develop and test more intuitive graphical management tools for VMware Tanzu, focusing on reducing the steep learning curve associated with its deployment and operation, especially for non-specialist administrators.
5. Energy Efficiency and Sustainability in Cloud-Native Implementations: Analyze VMware Tanzu's energy usage trends relative to competing alternatives. Propose novel algorithms or methodologies to enhance Tanzu deployments' energy efficiency, supporting contemporary IT infrastructure's sustainability objectives.

The five proposed research areas represent critical opportunities to enhance VMware Tanzu's position as a leading cloud-native platform while addressing broader challenges in the evolving IT landscape. With the rising prevalence of hybrid and multi-cloud architectures, optimizing performance in multi-cloud environments is essential for efficient resource utilization and seamless workload distribution, thereby enhancing Tanzu's value proposition for enterprises utilizing varied infrastructure configurations. With escalating security concerns stemming from the intricate nature of integrated virtualization and containerization, developing specialized security frameworks for VMware Tanzu is crucial to protect enterprise data in multi-tenant environments, especially as Kubernetes and cloud-native architectures expand. Simultaneously, comparative cost efficiency analyses can highlight Tanzu's financial viability against competitors like OpenShift, Azure Kubernetes Service, and Amazon Elastic Kubernetes Service, aiding enterprises in making informed investment decisions aligned with their economic constraints and strategic objectives. User-centric management interfaces can enhance accessibility to Tanzu's capabilities, enabling IT professionals to effectively manage complex systems and mitigate the steep learning curve associated with cloud-native solutions. Furthermore, investigations into energy efficiency can position Tanzu as a sustainable IT solution, minimizing its carbon footprint to conform with global climate objectives while providing cost savings for consumers.

16. Conclusion

This paper emphasizes VMware Tanzu as a crucial solution for integrating traditional virtualization with contemporary containerization, providing scalable, flexible, and enterprise-grade functionalities. The research highlights Tanzu's capacity to facilitate Kubernetes adoption across various environments by analyzing its architecture, persistent storage integration, and deployment strategies. Through its powerful tools like Tanzu Kubernetes Grid and NSX, VMware empowers organizations to enhance IT investments and effectively address changing cloud-native requirements.

Table 2 compares the most used Kubernetes platforms and their features and differences. In the table, according to our previous experience, a list of criteria was compiled based on which *OpenShift*, *VMware Tanzu*, and *Kubernetes* can be evaluated:

Table 2.
Comparison of VMware Tanzu, OpenShift, and Kubernetes distribution.

Criteria	VMware Tanzu	OpenShift	Kubernetes
Integration with own products	High level of possibilities	Intermediate level of capabilities	Low level of capabilities
Integration with other third-party products	High level of possibilities	High level of possibilities	High level of possibilities
Own image container registry	Exists	Exists	There is no
Own CI/CD *pipeline*	Exists	Exists	There is no
User experience and interface	Positive	Positive	Moderately
Scaling	High level of possibilities	High level of possibilities	High level of possibilities
Own network solutions	Exist	Exist	They do not exist
The complexity of the installation of the platform	Medium	Medium	High
The complexity of installing applications	Medium	Low/Medium	Medium

As for the first criterion, integrations with their products, Tanzu and OpenShift certainly have an advantage over Kubernetes, which is logical to assume since their primary purpose is to deliver a platform based on Kubernetes architecture with the addition of ready-made tools and solutions. As for Tanzu, it has already been mentioned that TKG supports integration with other VMware products such as vSphere, vSAN, NSX, vRealize, and others. On the other hand, OpenShift has fewer integration options since it does not even have such a product portfolio. Still, it does support Red Hat products such as RHEL, Ansible, Jaeger, and others. As the table above states, Kubernetes does not have its products but uses a modular approach to integrate them into its ecosystem.

Kubernetes excels at integrating with third-party products and solutions. Tanzu and OpenShift also support many products, such as Helm, Knative, Prometheus, Grafana, ELK, Jenkins, Vault, etc. Some are implemented during operation, and some are just mentioned. Furthermore, OpenShift and Tanzu have their container storage registry Image where it is during operation; VMware's solutions use Harbor. On the other hand, OpenShift uses Quay, which can be easily installed using a console and OperatorHub. Although Kubernetes does not have its container registry, it is possible to integrate Harbor or Quay. It is also important to note that their solutions in Tanzu and OpenShift do not exclude the possibility of integration, such as OpenShift and Harbor or Tanzu and Quay.

The fourth criterion is the availability of its own CI/CD solution for application delivery. Again, Kubernetes does not have its own, but it can be created using tools such as Jenkins X. Tanzu creates CI/CD pipelines using Tanzu Build Service. In addition, it has a dedicated platform. Tanzu Application Platform, in which Tanzu Build Service is one of the components that provide a platform for creating and launching Kubernetes' cloud-native applications. On the other hand, OpenShift uses your solution, OpenShift Pipelines, which is the counterpart to Tanzu Build Service.

Furthermore, when it comes to user interface criteria, since OpenShift and Tanzu are commercial products, they certainly have an advantage over Kubernetes, where sometimes long-time users and administrators of Kubernetes platforms have never even seen a Kubernetes console even though it exists. When discussing the OpenShift console, Tanzu offers the same possibility by using vSphere or the TKGI Management console for cluster management. Still, Kubernetes resources in Tanzu are created using the command shell.

There are various options for cluster scaling and pods for all three distributions. The only difference is in the scaling methodology, i.e., the ability to scale clusters and floors from graphical interfaces, as is the case with Tanzu using TKGI Management or TMC console.

For the following criterion, availability of network solutions, Kubernetes is again behind OpenShift and Tanzu, offering advanced network features that are out-of-the-box, such as load balancers and network security policies and integration with SDN solutions. Specifically, OpenShift uses its own OpenShift SDN, which is based on Open vSwitch, and Tanzu, in addition to the NSX platform, can be used by Calico, Flannel, and Antrea. In Kubernetes, all these features can be achieved using CNI implementations or SDN solutions such as Calico and Flannel, but they must be installed manually.

Furthermore, when it comes to the complexity of platform installation, Kubernetes is far more complex and demanding than Tanzu and OpenShift because its installation is done entirely from the command shell, most often using kubeadm or kubespray tools. On the other hand, the installation of Tanzu is done via consoles and configuration wizards, and there is no difference when it comes to OpenShift. However, although their installation is relatively simple, the prerequisites that need to be set and met before that are not, and in the case of Tanzu, it also requires knowledge of other VMware products such as vSphere and NSX.

After comparing the most prominent on-prem Kubernetes distributions, which can be installed in the cloud today, two purebred Kubernetes distributions or services on cloud computing servers are compared. The following table compares the Kubernetes services used during operation: AKS on Azure and EKS on AWS. The table will differ from the previous one because of significant differences, such as having and not having your network solution or registry for containers. Still, the differences are in the specific names of the services that allow it in the cloud. Therefore, evident differences will be compared in the Table 3:

Table 3.
Compare AKS and EKS Kubernetes cloud services.

Criteria	AKS	EKS
The complexity of the installation	Installation is initiated and configured using the Azure portal from a single configuration wizard.	The installation is initiated and configured using the AWS console. However, the setup of the Control Plane and Worker nodes is separate, and IAM roles must be created.
Integration with your services	Ability to integrate with Azure Container Registry, Azure Monitor, DevOps, Azure Cosmos DB, etc.	Ability to integrate with Amazon Elastic Container Registry, Amazon CloudWatch, AWS CodePipeline, Amazon DynamoDB, etc.
Integration with other third-party products	Integrations with various tools, such as Jenkins, Calico, GitHub, Istio, HA-Proxy, Harbor, etc., are possible.	Integrations with various tools, such as Jenkins, Calico, GitHub, GitLab, Isti, HA-Proxy, Splunk, etc., are possible.
CI/CD pipeline	Ability to integrate with your own CI/CD service Azure DevOps	Possibility of integration with your own CI/CD service AWS CodePipeline
Network Solutions	They use their network solutions, such as Azure Virtual Network, Azure Load Balancer, Azure Application Gateway, and third-party solutions, such as Calico and Weave Net.	They use network solutions, such as Amazon VPC, Amazon ELB, and Amazon Route 53, and third-party solutions, such as Calico and Cilium.
Surveillance services	It uses its monitoring service, Azure Monitor, and third-party solutions such as Prometheus and Fluentd.	It uses its monitoring service, Amazon CloudWatch, and third-party solutions such as Prometheus and Fluentd.
Scaling	Supports automatic horizontal and vertical scaling of floors and nodes	Supports automatic horizontal and vertical scaling of floors and nodes
Payment model	Based on the number of Kubernetes nodes and the time during which they are running, as well as the infrastructure on which they are located	Based on the number of EC2 instances, i.e., Worker nodes that are running and on additional AWS services required for EKS

The configuration of the control plane and worker nodes is separated when creating an EKS cluster. It is also necessary to create IAM roles, which are different from an AKS installation where everything can be configured in a single configuration wizard. In addition, the differences in the names of other AWS and Azure services used by the cluster for the same purpose, Kubernetes The clouds are the same. Both payment models are also very similar and are paid nodes, that is, virtual machines run by virtual machines, which depend on the region in which they are running and the size of the virtual machine. Node and the time during which the Node is running. The payment model also includes accompanying, sometimes indispensable Azure AWS services that serve a Kubernetes cluster as a Load Balancer, network infrastructure, storage system, etc. So, VMware is aware of its apparent advantages (vSphere, NSX, vSAN, vRealize, etc.) to other competitors, uses them very well, and composes them into its interpretation of Kubernetes. It should also be noted that other VMware products (for example, VMware Cloud Foundation or VCF) use a version of Kubernetes that entirely separates the management/control plane from the worker nodes, which should also be the preferred solution for risk avoidance reasons.

Ultimately, the selection among Kubernetes, Tanzu, Tanzu Kubernetes Grid, AWS Elastic Kubernetes Service, and Azure Kubernetes Service is contingent upon your organization's specific requirements, proficiency, and infrastructure strategy. Kubernetes, the premier open-source container orchestration platform, provides unparalleled flexibility and control for sophisticated users. Nonetheless, it necessitates profound proficiency in command-line utilities, scripting, and cluster management, rendering it more suitable for organizations possessing adept DevOps teams capable of navigating its intricacies.

Tanzu and Tanzu Kubernetes Grid excel in bridging this gap, especially for organizations seeking a more intuitive interface. Tanzu provides a user interface-driven management experience that renders Kubernetes accessible to a broader audience, enabling users lacking extensive command-line interface knowledge to manage deployments efficiently. Furthermore, Tanzu's comprehensive lifecycle management features facilitate updates, scaling, and maintenance in hybrid and multi-cloud settings, minimizing operational overhead while ensuring enterprise-level performance and adaptability.

Simultaneously, managed services such as AWS EKS and Azure AKS provide ease of use and scalability, seamlessly integrated within their respective cloud ecosystems. They are superior options for cloud-native applications, yet they may not offer the extensive multi-cloud and on-premises integration available through Tanzu.

Tanzu ultimately balances functionality and usability, rendering Kubernetes accessible to wider teams while delivering advanced functionalities for organizations concentrating on unified, lifecycle-managed Kubernetes clusters. Tanzu is an appealing solution for organizations prioritizing accessibility while maintaining enterprise functionality.

References

[1] M., A. Dinkar, S. C. Mouli, S. B., and A. A. Deshpande, "Comparison of Containerization and Virtualization in Cloud Architectures," 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, pp. 1–5, Jul. 09, 2021. doi: 10.1109/conecct52877.2021.9622668.

[2] J. Zhang, X. Lu, and D. K. Panda, "Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters," 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, pp. 1777–1784, May 2016. doi: 10.1109/ipdpsw.2016.178.

[3] A. Dab, I. Fajjari, M. Rohon, C. Auboin, and A. Diquelou, "An Efficient Traffic Steering for Cloud-Native Service Function Chaining," 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). IEEE, pp. 71–78, Feb. 2020. doi: 10.1109/icin48450.2020.9059340.

[4] G.I.Radchenko, A.B.A. Alaasam, A.N.Tchernykh, "Comparative Analysis of Virtualization Methods in Big Data Processing," Supercomputing Frontiers and Innovations, vol. 6, no. 1, FSAEIHE South Ural State University (National Research University), pp. 32-33, Mar. 2019. doi: 10.14529/jsfi190107.

[5] R. Vaño, I. Lacalle, P. Sowiński, R. S-Julián, and C. E. Palau, "Cloud-Native Workload Orchestration at the Edge: A Deployment Review and Future Directions," Sensors, vol. 23, no. 4, pp. 2215, Feb. 16, 2023. doi: 10.3390/s23042215.

[6] F. Leymann, U. Breitenbücher, S. Wagner, and J. Wettinger, "Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation," Communications in Computer and Information Science, Springer International Publishing, pp. 16–40, 2017. doi: 10.1007/978-3-319-62594-2_2.

[7] E. F. Boza, C. L. Abad, S. P. Narayanan, B. Balasubramanian, and M. Jang, "A Case for Performance-Aware Deployment of Containers," Proceedings of the 5th International Workshop on Container Technologies and Container Clouds. ACM, pp. 25–30, Dec. 09, 2019. doi: 10.1145/3366615.3368355.

[8] S. Hirai, T. Tojo, S. Seto, and S. Yasukawa, "Automated Provisioning of Cloud-Native Network Functions in Multi-Cloud Environments," 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, pp. 1–3, Jun. 2020. doi: 10.1109/netsoft48620.2020.9165343.

[9] A. Bouridah, I. Fajjari, N. Aitsaadi, and H. Belhadef, "Optimized Scalable SFC Traffic Steering Scheme for Cloud Native based Applications," 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC). IEEE, pp. 1–6, Jan. 09, 2021. doi: 10.1109/ccnc49032.2021.9369583.

[10] A. Tan, H. Ma, and Y. Mei, "Novel Genetic Algorithm with Dual Chromosome Representation for Resource Allocation in Container-Based Clouds," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, pp. 452–456, Jul. 2019. doi: 10.1109/cloud.2019.00078.
[11] L. Wang, J. Wu, Y. Gao, and J. Zhang, "Deep Reinforcement Learning Based Resource Allocation for Cloud Native Wireless Network," arXiv, May 2023. doi: 10.48550/ARXIV.2305.06249.
[12] H. Huang et al., "PVM: Efficient Shadow Paging for Deploying Secure Containers in Cloud-native Environment," Proceedings of the 29th Symposium on Operating Systems Principles. ACM, pp. 515–530, Oct. 23, 2023. doi: 10.1145/3600006.3613158.
[13] G. Ambrosino, G. B. Fioccola, R. Canonico, and G. Ventre, "Container Mapping and its Impact on Performance in Containerized Cloud Environments," 2020 IEEE International Conference on Service Oriented Systems Engineering (SOSE). IEEE, pp. 57–64, Aug. 2020. doi: 10.1109/sose49046.2020.00014.
[14] A. A. Khan and M. Zakarya, "Energy, performance and cost efficient cloud datacentres: A survey," Computer Science Review, vol. 40, Elsevier BV, p. 100390, May 2021. doi: 10.1016/j.cosrev.2021.100390.