



Divided Yet United: Architecting the Multi-Tenant Private AI Cloud

Empowering seamless self-service, strict isolation, and Private AI readiness with VMware Cloud Foundation® 9

Table of contents

Introduction: the multi-tenancy imperative 3

The governance challenge: why multi-tenancy matters..... 3

The provider/consumer model 4

How multi-tenancy works: the concentric model 5

Self-service consumption and multi-tenancy 6

What is new in VCF 9: the Kubernetes-based architecture..... 7

Enterprise multi-tenant models.....11

Advanced use cases 12

How VCF 9 is different 13

Business outcomes and value 14

Conclusion 15

Glossary.....16

Introduction: the multi-tenancy imperative

The private cloud is undergoing a significant transformation since virtualization. Organizations now need to support diverse workloads — AI, Kubernetes/containers, and VMs — on shared infrastructure, while keeping developers, business units, and/or external clients strictly isolated. Acquisitions make this even harder; enterprises need to absorb new workloads while keeping acquired entities fully isolated from the rest of the business. Multi-tenancy has moved from a nice-to-have to a requirement; you need it to build a cloud, not simply to have infrastructure with self-service bolted on.

VMware Cloud Foundation (VCF) 9 provides multi-tenancy capabilities via VMware Cloud Foundation Automation (VCF Automation). These capabilities draw on years of experience from VMware Cloud Director (VCD), the industry-recognized platform for service provider multi-tenancy. But VCF 9 goes further: it delivers multi-tenancy on a Kubernetes-based architecture, aligning isolation, governance, and self-service with the way modern applications are actually built and deployed.

This whitepaper explains why multi-tenancy matters now, how VCF Automation delivers it, and what makes it different. It is written for IT executives, cloud architects, platform engineers, and administrators who are evaluating or adopting a multi-tenant private AI cloud. The focus is on the why and the what, the business problems multi-tenancy solves and the capabilities that solve them, rather than deep implementation details.

The governance challenge: why multi-tenancy matters

Most enterprise IT organizations share infrastructure across multiple teams, business units, and environments. Many rely on dedicating physical infrastructure like VMware vCenter® instances or workload domains to keep groups separate, a pattern that is expensive and does not scale. Without a formal multi-tenancy model, isolation depends on manual processes, and self-service suffers because there is no safe way to give tenants direct access. The results are predictable: configuration drift, security gaps, wasted resources, and slow provisioning.

The cost of manual isolation

Manual enforcement of network isolation, firewalls, and data sovereignty across shared infrastructure is error-prone, raising fears of lateral breaches. When isolation depends on administrators remembering to apply the right rules, a single mistake can expose sensitive workloads to unauthorized access. Many organizations also lack comprehensive visibility into infrastructure utilization across groups, environments, and workload types.

To close these gaps, some customers have built their own custom workflows or portals on top of existing infrastructure. These homegrown approaches can be costly to build and maintain over time, yet teams still resort to spreadsheets to track organizational setup and guardrail definitions across tenants. Without granular, namespace-level evidence of what developers deployed and what it cost the business, justifying infrastructure investments becomes a guessing game.

What changes with multi-tenancy

Multi-tenancy makes isolation and governance core capabilities of the platform rather than manual overhead. With VCF Automation, a VI Admin new to multi-tenancy can quickly set up organizations like a cloud provider, carving up resources, isolating infrastructure, and segmenting networks to create isolated private clouds for each organization. The platform prevents unauthorized access or communication between different groups by design, not by manual rule.

Each tenant operates within its own isolated environment, with access only to allocated services and resources. Governance, quota limits, and operational policies apply automatically, tailored to specific use cases or ownership groups. VCF Automation delivers all of this out-of-the-box, replacing the custom workflows and portals that organizations previously had to build and maintain themselves. The shift is fundamental: the platform enforces boundaries instead of asking administrators to do it.

The provider/consumer model

VCF Automation introduces a clear separation between the provider of cloud services and the consumers of those services. This model mirrors how public cloud providers operate, and it is the foundation of multi-tenancy in VCF 9.

Provider: the cloud operator

The provider, typically an Enterprise IT Admin (Provider Admin), uses the Provider Management Portal to set up organizations (tenants) for business divisions or lines of business (LOBs), allocate resources, configure networking, and manage identity. The provider owns the infrastructure: zones, regions, compute, storage, and network. They define the guardrails within which tenants operate, and monitor utilization, chargeback, and alerts across all tenants through provider dashboards. When a tenant cannot manage its own identity, the provider can do so on its behalf, acting as the tenant's operator for as long as needed.

Consumer: the tenant

Each organization (tenant) is a top-level entity that represents a business division (or a client). Each operates in a secure, isolated environment with access only to allocated services and resources. The Org Admin manages the organization through the Organization Portal, creating projects for different LOBs (or different teams within a LOB) and applying policies. The Org Admin can delegate administrative tasks to project-level admins as needed, keeping governance centralized while distributing day-to-day operations.

Within each organization, LOB Admins and platform engineers support the consumption of resources by application teams and developers. They deploy workloads, manage namespaces and [private cloud services](#), and operate within the guardrails the provider and Org Admin have established. The consumer gets self-service access without needing to understand the underlying infrastructure.

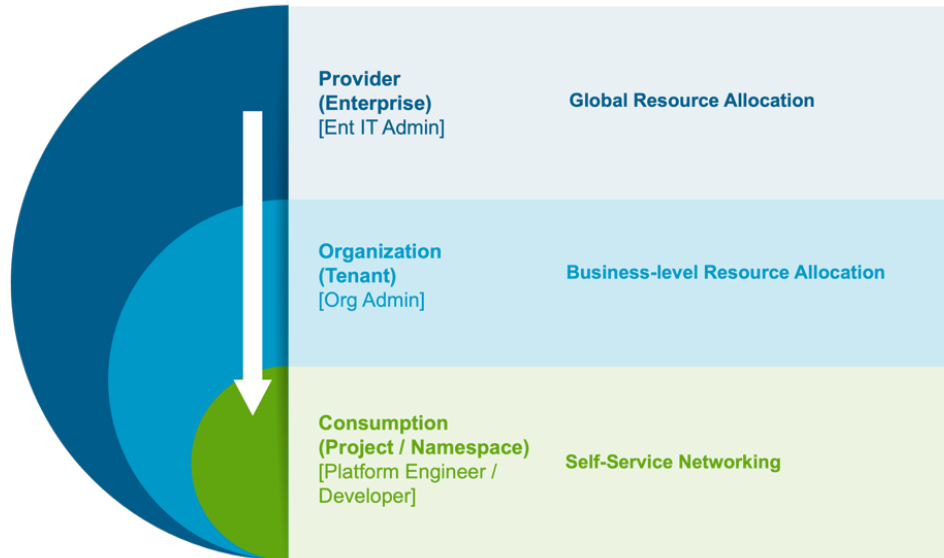
The structure above is illustrative, not prescriptive. VCF Automation supports any depth of hierarchy: a provider admin can manage everything alone, or delegate across multiple layers of org and project admins. With multiple layers of tenancy and separation built in, the platform meets IT where they are in their journey, from the simplest setup to the most complex delegation model.

VCF Automation Role	Typical Real-World Role Example	Responsibilities	Portal
Provider Admin	Enterprise IT Admin, Central Infrastructure Admin	Manage zones, regions, infrastructure, tenant network isolation, system settings	Provider Management Portal
Org Admin	Tenant Admin, Division IT Manager	Create projects, manage namespaces and VPCs, apply policies, manage identity and RBAC, content and catalog	Organization Portal
Project Admin	LOB Admin, Platform Engineer, Application Team Lead	Deploy and manage workloads, consume private cloud services within guardrails, create standardized infrastructure components and environments, manage content and catalog	Organization Portal
Adv Project user / Project user	Developer, DevOps Engineer	Self-service provisioning via catalog, CLI, or Kubernetes APIs; visibility into workload cost	Organization Portal

Table 1: The provider/consumer role hierarchy in VCF Automation multi-tenancy.

How multi-tenancy works: the concentric model

VCF Automation's multi-tenant architecture follows a concentric model with three layers. Enterprise control sits at the outer ring, tenant autonomy and security guardrails in the middle ring, and frictionless developer consumption at the inner ring. Each ring gets exactly the level of control it needs; the provider governs the fabric, tenants govern their own environments, and developers consume resources without touching the infrastructure underneath. Here is how the concentric model applies to networking and security, from the outer ring inward.



The outer ring: enterprise control and fabric management

Enterprise IT Admins handle global resource allocation and the physical and logical fabric. This includes defining structured IP blocks for external connectivity, provisioning physical VLAN extensions to share across organizations, assigning clusters for stateful services, and establishing external connectivity for organizations. The outer ring is where the provider sets the boundaries that every tenant operates within.

The middle ring: tenant autonomy and guardrails

Org Admins manage business-level resource allocation within the provider's boundaries. They build templates to automate VPC setups, manage cross-namespace networking, and enforce security policies. Tenants get architectural control over outbound traffic with multiple dedicated paths for external communication, supporting both centralized Tier-0 and distributed VLAN connections, without waiting on a central networking ticket.

The inner ring: frictionless developer consumption

Because Enterprise IT and Org Admins have established guardrails, platform engineers and developers get frictionless, self-service access. They consume assigned VPCs, self-provision isolated networks, and deploy dedicated namespace subnets on demand, without understanding complex routing protocols. The inner ring is where the self-service promise becomes real.

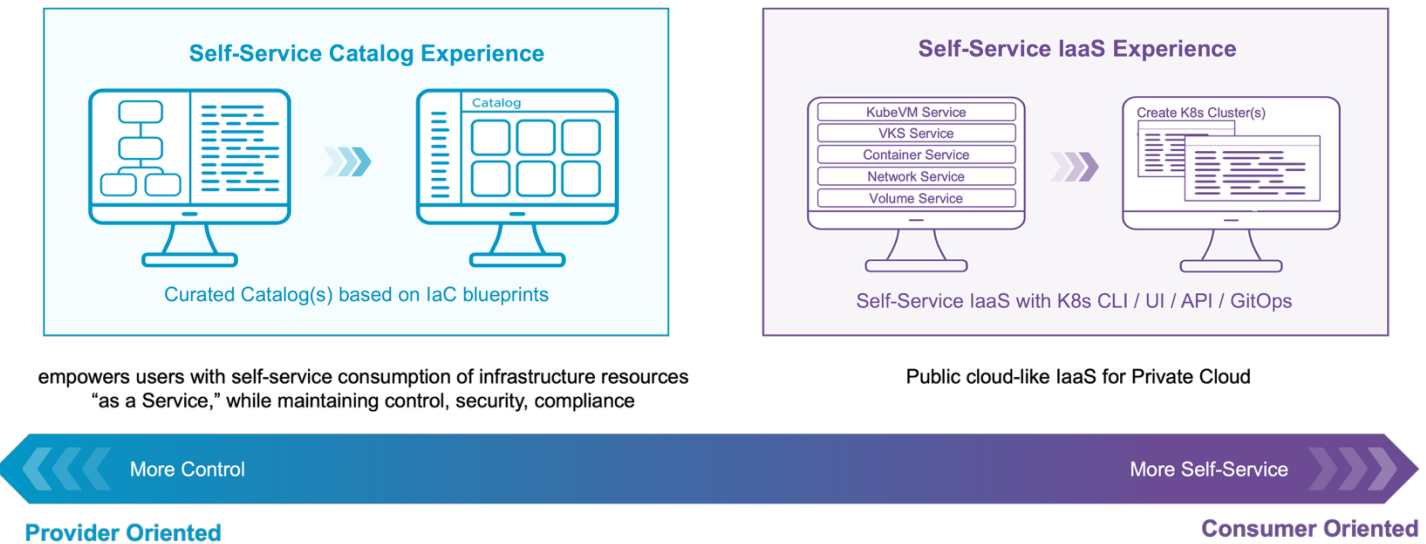
Self-service consumption and multi-tenancy

Multi-tenancy and self-service are inseparable. Without multi-tenancy, self-service lacks the isolation needed to keep teams and workloads separate, creating security and governance risks. With multi-tenancy, each tenant can provision resources within their own isolated, governed environment, without submitting tickets or waiting for IT.

VCF Automation enables IT to deliver a self-service private AI cloud for application teams to build, run, and manage AI, Kubernetes, and VM-based applications. The solution makes it easier to jump start and scale a multi-tenant private AI cloud with IaaS resources available out-of-the-box, helping bring applications to market faster while maintaining control with policy-based governance.

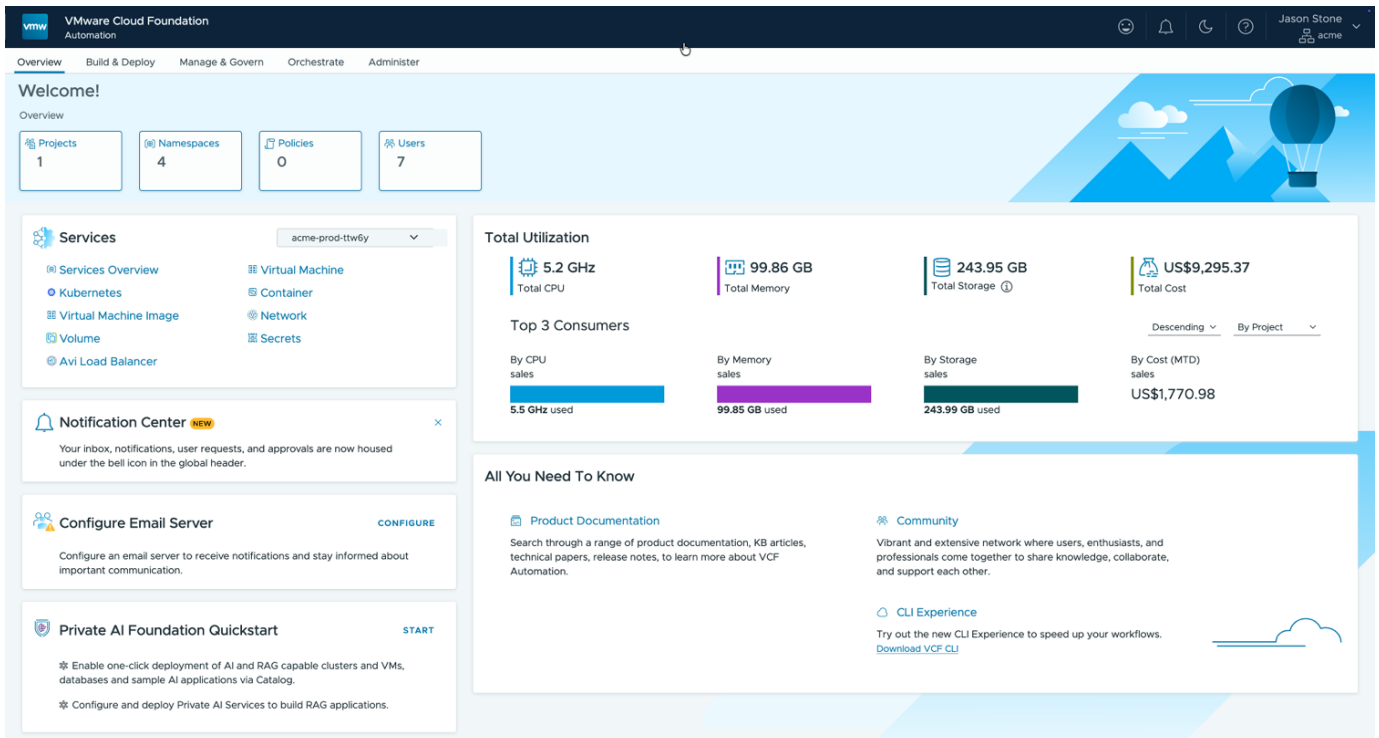
Choice of consumption models

VCF Automation empowers users with a choice of self-service consumption options. The provider (i.e., admin)-oriented self-service catalog is curated with pre-approved, standardized application blueprints suitable for production deployments and users who do not want to manage infrastructure details. The consumer-oriented self-service IaaS exposes private cloud services through UI, CLI, or declarative Kubernetes APIs. DevOps users can also leverage Git repositories and CI/CD tools to continuously build, deploy, and configure infrastructure and applications through GitOps workflows. These paths operate within the same governance guardrails.



FinOps and cost visibility

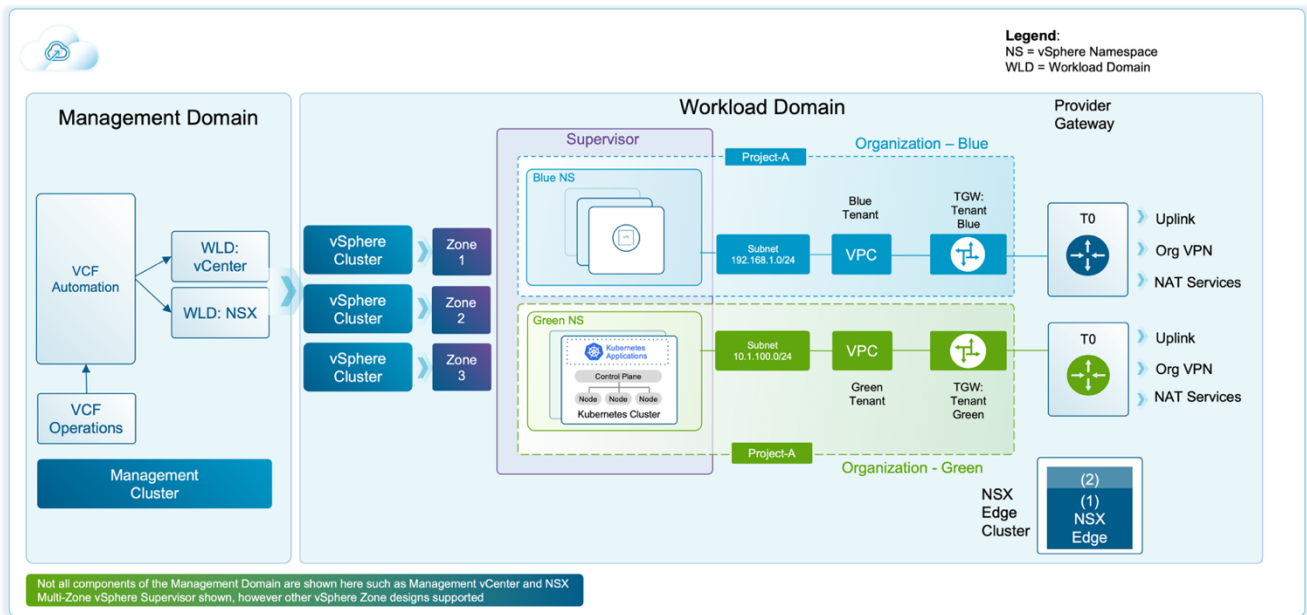
VCF Automation provides users with direct visibility into workload performance and cost. [Embedded FinOps](#) helps reduce total cost of ownership (TCO) by influencing consumption behavior: upfront pricing helps prevent overprovisioning, lease policies minimize wasteful resource usage, and automated reclamation removes inactive 'zombie' VMs. The platform enables tracking of utilization and chargeback across multiple groups, giving IT leaders granular cost accountability at many levels, whether by project, team, or organization, to justify infrastructure investments.



What is new in VCF 9: the Kubernetes-based architecture

The multi-tenancy capabilities in VCF 9 are designed for a fundamentally different architecture than previous solutions. At the core is the VMware vSphere® Supervisor (Supervisor), a Kubernetes-based control plane that runs directly on vSphere clusters. This powers the new 'Org for All Apps' experience and introduces concepts that were not available in VCD.

Example: Multi-Tenant Environment



All Apps Organization

The All Apps organization brings the Kubernetes API model to the forefront for infrastructure consumption in VCF Automation. Through the VCF control plane and Supervisor, infrastructure and services are exposed via Kubernetes APIs and Custom Resource Definitions (CRDs). This gives platform teams and consumers a common declarative API surface for both applications and infrastructure.

This approach opens the door to:

- Kubernetes-style APIs for infrastructure consumption
- GitOps-driven infrastructure workflows
- Cluster API for declarative Kubernetes cluster lifecycle
- VMs, networking, databases, and services exposed as API resources
- Strong multi-tenancy and governance boundaries
- A common API model across applications and infrastructure

Zones

Zones abstract underlying physical infrastructure into logical deployment targets, giving consumers a public-cloud-style availability zone experience. To end users, a zone represents an isolated fault domain; application components, KubeVMs, and VMware vSphere Kubernetes Service (VKS) nodes can be distributed across zones to maximize resilience against localized failures.

A zone is backed by one or more vSphere clusters, completely decoupling physical cluster management from the consumer workflow. When a tenant provisions a namespace, the system automatically handles placement on an underlying cluster within that zone, hiding infrastructure complexity. Zones can share network boundaries, but they strictly do not share or replicate storage pools, preventing storage failures from propagating across zone boundaries.

For topology and sizing, one zone is suited for small, non-critical, or sandbox environments where cross-domain high availability is not required. Three zones is the recommended standard for multi-tenant production deployments, providing a strong consumer experience and control plane high availability. Extended topologies can scale beyond three zones to support specialized requirements, such as dedicated zones for Windows versus Linux node pools or strict compliance boundaries.

Regions

In VCF Automation, a Region aggregates compute, storage, memory, and networking resources from one or more vSphere Supervisors across vCenter instances managed by a single NSX Manager instance.

Regions align directly with your underlying VCF workload domains and networking design boundaries (such as vSphere Zones or availability zones). Depending on your VCF deployment model, a Region can represent a specific physical workload domain, an availability zone, or a logical boundary dedicated to a business unit or performance tier.

Note - Configuration Consistency: All Supervisors within a given Region must maintain homogeneous configurations including matching Kubernetes versions, identical storage class definitions, and a consistent set of IaaS services.

Resources in a region can be dedicated to a single organization or shared across multiple organizations.

Region quota

The platform lets providers carve out quota on shared infrastructure in sophisticated ways: granular compute and storage allocation across multiple organizations on the same physical resources. But multi-tenancy does not require everything to be shared. Providers can assign an entire region to a single organization when IT needs dedicated infrastructure, and the platform meets them where they are. Whether infrastructure is shared or dedicated, admin-level isolation remains; each tenant gets its own portal, identity provider, and branding, so the self-service experience is the same either way.

A region quota defines the compute and storage resources allocated to an organization for a given region. The key aspects include CPU and memory utilization limits (the upper bound of quota), CPU and memory reservations (the guaranteed capacity), VM classes (t-shirt-sized classes of compute), and storage classes with assigned quota, plus infrastructure policies, which are also assigned via the region quota. An organization can receive region quotas from multiple regions, but there can be only one quota per region. In multi-zone regions, you can assign quotas to all or a subset of the zones. VM classes and storage classes are defined at the region quota level and apply to all zones in that quota.

Organizations, projects, and namespaces

VCF Automation establishes a multi-tenant private AI cloud by providing network isolation through Virtual Private Cloud (VPC) constructs, managing Organizations, and applying Projects and Namespace Classes for resource allocation, quota, chargeback, and organizational control. An organization is a top-level entity representing a tenant. For example, a large division within an enterprise, a newly acquired company that needs to stay isolated, or an external client of a service provider. A project is a subdivision of an organization for a LOB or team. Projects provide a next level of lighter tenancy isolation and governance below the organization level, letting LOB Admins manage their own configurations within provider guardrails.

Namespaces (vSphere Namespaces) are resource envelopes that define resource limits for CPU, memory, and storage. Namespace Classes are templates used to create vSphere Namespaces, standardizing resource allocations and governance settings. Each Namespace can be assigned one or more VPCs, which can be shared across multiple Namespaces, enabling application teams to use common networking where required while maintaining isolation.

Virtual Private Clouds (VPCs)

VPCs provide network isolation domains within the shared VCF fabric. VCF Automation automates secure, self-service networking with strict multi-tenant isolation. Org Admins manage transit gateways, build templates to automate VPC setup, create org-level subnets to share across namespaces, and enforce inter- and intra-VPC security via distributed and gateway policies. Consumers get on-demand network isolation and dedicated workload segmentation within organization guardrails.

Networking and security automation

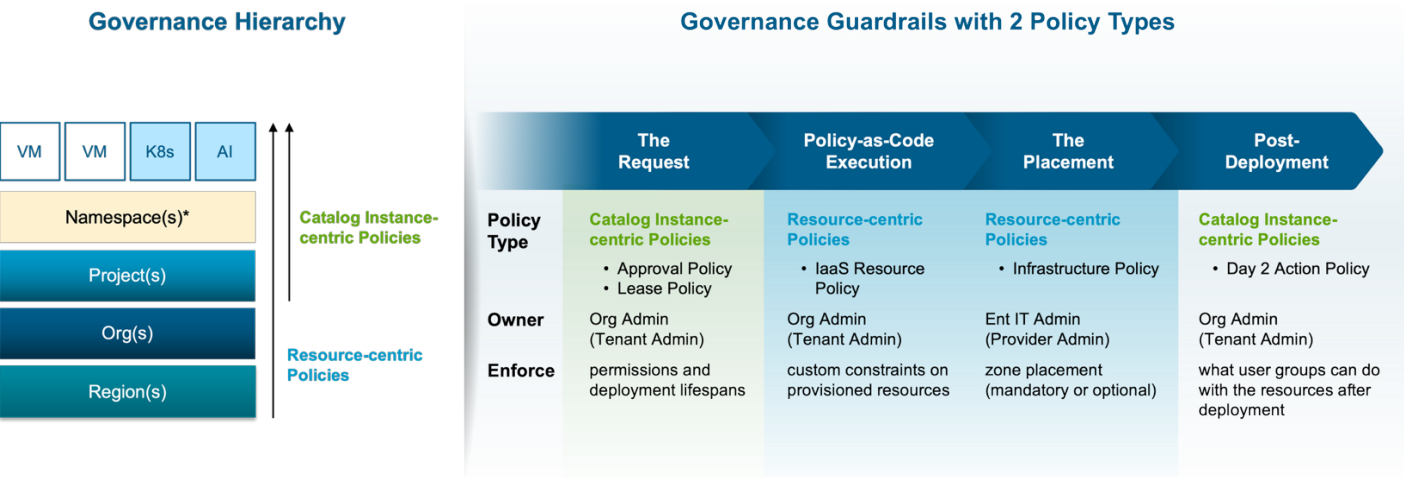
VCF Automation unlocks comprehensive self-service networking for Org Admins. They can independently configure custom NAT rules, static routes, and site-to-site VPNs directly on their gateways without waiting on a central networking ticket.

This autonomy helps maintain security. Powered by VMware® vDefend™, VCF Automation integrates Distributed and Gateway Firewalls driven by built-in Security Profiles. Security rules are processed in a strict, sequential hierarchy, starting from provider rules down to organization and VPC rules. This helps ensure that corporate security mandates are enforced top-down and cannot be overridden by lower-level users.

VCF Automation also introduces full self-service capabilities for VMware® Avi™ Load Balancer, empowering application teams with autonomous, on-demand application delivery while enterprise IT maintains strict control over resource quotas. The result is a public cloud networking experience for application teams, grounded in the hierarchical security and strict multi-tenant isolation the enterprise demands.

Policies

VCF Automation enforces governance through two policy types: catalog instance-centric policies and resource-centric policies. Both apply across different stages of the deployment lifecycle and are managed at either the provider or the Org Admin level.

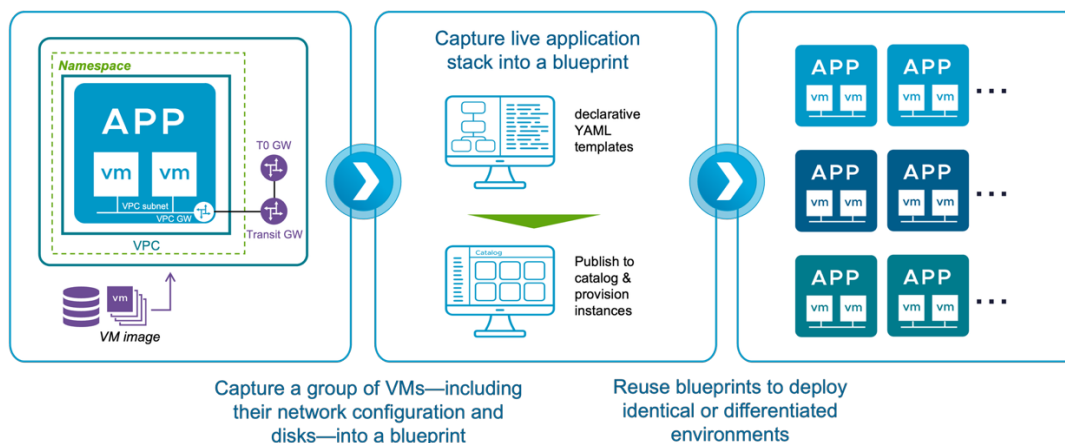


Catalog instance-centric policies apply to catalog requests. Approval policies control who must authorize a deployment before it is provisioned, with pending requests surfacing in the approver's inbox for action. Lease policies define the maximum time a deployment can exist; when it expires, resources are automatically destroyed and reclaimed, reducing manual cleanup. Day 2 action policies control which user groups can perform post-deployment operations such as resizing, snapshotting, or reconfiguring resources.

Resource-centric policies include IaaS resource policies and infrastructure policies. IaaS resource policies govern requests to provision IaaS services like the KubeVM Service, applying granular control over resource types and configurations through YAML-based Policy as Code on Kubernetes Validating Admission Policy. Infrastructure policies, assigned by the provider, define VM placement on specific hosts using vCenter compute policies in the Supervisor context; the system auto-applies mandatory policies, tenants can opt into optional ones, and namespace creation fails if selected zones are incompatible. Providers can add clusters to zones dynamically, but new clusters must match existing policies with at least one tagged host per cluster. Policies cannot be edited after creation, and mandatory policies cannot be assigned to a region quota that already has namespaces.

App stack formation and blueprints

VCF Automation enables platform engineers to capture running sandbox environments as reusable blueprints and redeploy copies of those environments through an app stack formation use case. It helps reduce configuration drift and accelerate deployment of complex, multi-VM application topologies. Blueprints can be designed via visual canvas or YAML, supporting source controlled iterative development with Infrastructure as Code. The Content Hub centralizes blueprints, VM images, and orchestrator workflows across vCenters.



Enterprise multi-tenant models

Many enterprises function as their own cloud provider, serving multiple divisions that require “hard” isolation rather than “soft” (logical) separation through projects. Multi-tenancy gives each division its own tenant with dedicated infrastructure, network segmentation, and independent governance; the same isolation guarantees a service provider offers external clients. The following examples illustrate how enterprises apply this model across different industries and organizational structures.

Broadcom®: an enterprise that operates like a service provider

Broadcom itself is an example of an enterprise that operates like a service provider. With multiple business divisions, Broadcom's IT organization needs to carve up shared infrastructure and allocate resources to each division as if they were separate clients. Multi-tenancy lets Broadcom's IT team create isolated organizations for each division, with dedicated resource pools, network segmentation, and independent policies, all managed through a single platform. Organizations can be added seamlessly as the business grows, supporting growth without disruption.

Holding company: subsidiaries as tenants

A holding company that operates multiple subsidiaries faces a similar challenge. Each subsidiary is an independent business with its own IT requirements, compliance obligations, and data isolation needs. With VCF Automation, the holding company can create an organization for each subsidiary on shared infrastructure. Each subsidiary gets its own isolated environment with independent identity provider integration, dedicated network segments, private compute domains, and separate billing.

Financial services: buy-side and sell-side isolation

Financial services firms often have both buy-side and sell-side business and typically implement strict internal controls like segregation of users, data, and infrastructure to avoid potential conflicts of interest for risk management. VCF Automation's multi-tenancy addresses this directly: the firm can create separate organizations for buy-side and sell-side operations, with infrastructure isolation and segmented networks using VPCs. Isolated resource pools can be created in clusters (for example, GPU silos for AI workloads), ensuring neither side can access the other's resources.

Dev/test/prod environments

A common enterprise use case is environment isolation. Development, testing, and production workloads can be completely isolated to prevent unauthorized changes from impacting live applications. Namespaces provide resource envelopes that make it easy to apply governance, quota limits, and operational policies tailored to each environment, so that a developer's experiment in a dev namespace cannot affect a production namespace on the same physical infrastructure.

Advanced use cases

VPC as a service and sovereign isolation

Enterprise lines of business and sovereign cloud clients need and expect cloud-style networking, complete network isolation, overlapping IP space, and self-managed firewall rules, without waiting days for networking teams to manually configure VLANs or dedicated gateways. In most private data centers, that experience does not exist.

VCF Automation changes that with VCF 9's native VPC architecture. Tenant Admins self-provision isolated VPCs and subnets on demand with direct control over Distributed Firewall policies and NAT configurations within their project boundary. Shared physical infrastructure stays invisible to tenants, who get the public-cloud networking experience on private cloud, and overlapping IP addresses are natively supported across separate projects. Service providers and sovereign cloud operators get high-density, multi-tenant isolation with no cross-tenant visibility or route leaking.

Platform engineering & self-service data services (DBaaS)

Developers often bypass internal IT and go straight to public cloud because traditional private data center portals are VM-centric and ticket-bound. They lack native integration with modern Kubernetes workflows and infrastructure-as-code pipelines, so the private data center feels slow by comparison.

VCF Automation's All Apps Organization turns the private data center into a private cloud powering Internal Developer Platforms, exposing declarative Kubernetes APIs alongside traditional service catalog blueprints. Platform engineers publish curated multi-tier application stacks alongside VCF-provided add-on services, like on-demand stateful Data Services (e.g., PostgreSQL, Redis, or Kafka). Developers self-provision complete application environments, containers, and database instances through their existing CI/CD pipelines, while VCF Automation enforces project quota, namespace isolation, and automated lease-based teardowns. Because every resource uses the same kubectl interface, integration into CI/CD pipelines is uniform, from standard VMs to custom-built operators.

Multi-tenant FinOps & intelligent capacity reclamation

Resource hoarding drives inflated CapEx in multi-tenant private data centers. Tenant teams over-provision CPU, RAM, and expensive vGPU for production and non-production environments alike, leaving idle capacity locked away from other teams, often with no clear cost visibility into who is using what.

VCF Automation paired with VCF Operations enables granular, FinOps across projects, VPCs, and namespaces in a multi-tenant private AI cloud. Provider admins define quota caps, showback / chargeback rate cards, and automated lifecycle policies. This gives LOB managers real-time visibility into unit economics while maximizing hardware utilization across shared infrastructure. Idle resources, such as memory, get reclaimed automatically, so capacity flows to where it is needed instead of sitting unused.

Multi-tenant disaster recovery

Disaster recovery (DR) has long been hard to deliver as a self-service capability in a multi-tenant environment. Traditional DR tools assume a single tenant with full access to the recovery site, so providers face a choice: either centralize all DR operations and force tenants to submit tickets for every protection action, or hand tenants direct access to shared recovery infrastructure and risk one tenant consuming resources another needs. Neither option scales easily.

VCF solves this with the Protection and Recovery for VCF Automation service, which builds DR into the multi-tenant model rather than bolting it on. Providers and Org Admins pre-approve regions, namespace pairings, replication classes, and network mappings, so tenants can self-serve VM-level replication within governed boundaries, without the ability to exhaust shared recovery capacity or declare a disaster without designated failover authority. Tenant isolation persists through recovery: VPC, project, and namespace boundaries are maintained across replication and failover, so a shared recovery site does not remove tenant boundaries. The proxy appliance allows traffic separation between tenant boundaries for additional

tenant isolation. The service supports KubeVMs with RPO as low as one minute through vSphere Replication, and tenants configure protection, and failover directly through VCF Automation, no separate DR console required.

Private AI with multi-tenant model sharing

Cloud service providers and enterprises can deploy private and independent AI environments for tenants. VMware is on a mission to democratize AI and ignite business innovation for all enterprises. VCF Private AI Services, part of VCF subscription, enable enterprises to unlock the business gains of AI while meeting privacy and compliance requirements. VCF enables pooling and sharing GPU resources across the organization so teams can run multiple models on shared hardware instead of dedicating infrastructure to each workload.

Multi-tenant model sharing, now generally available, lets the Model Runtime support secure sharing of AI models between tenants or lines of business through isolated namespaces. This maintains data privacy while helping avoid redundant model deployments that waste GPU allocation and infrastructure resources. Model gallery gives IT and data science teams a single interface that enables governance and security with curated and secure LLMs with integrated RBAC

Industry trends: Kubernetes and AI

The industry is moving decisively toward Kubernetes as the standard for deploying modern applications, and toward AI as the next major workload category. Both trends place new demands on infrastructure: Kubernetes requires declarative, API-driven management, and AI requires GPU sharing, cost governance, and strict data isolation. Traditional multi-tenancy models, built for VM-centric workloads, may not fully meet these demands.

VCF 9 multi-tenancy is designed for this reality. The Kubernetes-based architecture means that governance, isolation, and self-service are expressed through native Kubernetes constructs: namespaces, validating admission policies, and declarative APIs. AI workloads benefit from multi-tenant model sharing, GPU pooling, and namespace-level cost accountability. The platform extends the same network, security, identity, and governance model to every service it adds.

How VCF 9 is different

Functional advantages over alternatives

Compared to VCD, VCF 9 multi-tenancy adds Kubernetes-native governance (policy as code based on Validating Admission Policy), app stack formation, blueprint-based IaC, and a unified content hub. Compared to building a multi-tenant platform from open-source Kubernetes tools, VCF Automation provides an integrated, supported solution with [out-of-the-box IaaS services](#), [embedded FinOps](#), and a single consumption interface across a fleet of VCF environments. VCF Automation abstracts resources across a fleet of VCF environments and provides a centralized, single consumption interface.

Cost advantages

VCF Automation helps reduce TCO through embedded FinOps: upfront pricing influences consumption behavior, lease policies minimize waste, and automated reclamation removes inactive resources. The KubeVM, container, and VKS runtimes are available out-of-the-box, reducing the cost and complexity of a redundant Kubernetes platform. For service providers, the shared workload domain model means a single infrastructure stack serves multiple tenants, reducing hardware costs compared to dedicated infrastructure per tenant.

Capability	VCD	VCF 9: VCF Automation
Architecture	VM-centric	Kubernetes-based (vSphere Supervisor), enabling public cloud-like IaaS Services (KubeVMs, containers, VKS, and other services with any conformant operator)
Tenant isolation	Resource pools, vApp networks	VPCs, namespaces, namespace classes
Kubernetes runtime	Not included (requires separately licensed Container Service Extension)	VKS out-of-the-box
Governance	Automated lease, quota, and VM placement policies (no policy as code or declarative governance framework)	Catalog instance-centric and resource-centric policies, including policy as code (K8s Validating Admission Policy)
Blueprints / IaC	IT-Ops UI driven VCD Terraform Provider, vApp	Modern GitOps driven Low code visual design canvas + YAML (source controlled blueprints), app stack formation
Content management	Catalog, Content Hub	Content Library, Canonical Content Library, Content Hub
Consumption interfaces	Web portal, REST APIs	UI, CLI, declarative K8s APIs, REST APIs
AI workload support	Not native	VCF Private AI Services, Multi-tenant model sharing, GPU pooling

Table 2: VCF Automation multi-tenancy capabilities compared to VCD.

Business outcomes and value

VCF Automation multi-tenancy delivers value across three dimensions: efficiency, flexibility, and cost reduction.

Efficiency

Increased efficiency and productivity come from reducing manual tasks and human intervention, enabling IT teams to focus on strategic initiatives. Quick start wizards and guided workflows provide faster onboarding and scaling of multi-tenant environments. The platform facilitates the evolution of VI Admins into knowledgeable Cloud Admins (Provider and Org Admins), ready to manage multi-tenant environments.

Flexibility

Choice of governance models gives providers the flexibility to adapt their cloud operational model to meet new business requirements as the business evolves. Choice of consumption models gives consumers the flexibility to use the interface that fits their workflow: curated catalog for governed deployments, or developer-oriented IaaS for direct control.

Cost reduction

VCF Automation helps optimize resource utilization and reduce costs through embedded FinOps, automated reclamation, and lease policies. The shared infrastructure model reduces hardware costs for both enterprises and service providers. Granular, namespace-level cost accountability makes it possible to show peak infrastructure utilization and enforce chargeback, changing the conversation when justifying CapEx investments.

Conclusion

Multi-tenancy is the key to governing a modern, multi-tenant, self-service private AI cloud. Without it, organizations are left with manual processes, security risks, and wasted resources. With it, they can carve up shared infrastructure into isolated, governed environments that give each tenant the autonomy of a private cloud while keeping IT in control.

VCF 9 redefines multi-tenancy for the AI and Kubernetes era. Built on a Kubernetes-based architecture, it introduces namespaces, VPCs, policy as code, app stack formation, and a unified consumption model that previous solutions, including VCD, do not offer. It supports both enterprise and service provider models, from Broadcom's own multi-division operations to financial services firms requiring buy-side/sell-side isolation to offering multi-tenant cloud services.

For enterprises new to multi-tenancy, VCF 9 provides guided workflows and quick start wizards that make it easy to get started. For existing VCD customers, it offers a migration path and a new set of capabilities designed for the workloads that matter most today: Kubernetes and AI. The private cloud is no longer just about virtualization. It is about giving every team, every business unit, and every client their own piece of the cloud securely, efficiently, and on their own terms. VCF Automation multi-tenancy makes that possible.

Glossary

- **All Apps organization** - An organization type enabled to run a comprehensive set of built-in cloud services, including VMs, Kubernetes, networking, volumes, and AI workloads. Requires vSphere Supervisor clusters.
- **App stack formation** - A use case that enables the capture of running sandbox environment (a group of VMs) as a reusable blueprint and redeploy copies of that environment.
- **Avi Load Balancer** - A multi-cloud application delivery service that provides autonomous, on-demand load balancing for application teams within VCF Automation. Requires the Avi LB add-on.
- **Blueprint** - A declarative definition of infrastructure and application topology, authored via visual canvas or YAML, used for controlled, repeatable deployments.
- **Content Hub (in VCF Automation)** - A centralized portal for discovering, creating, and synchronizing content libraries, VM images, blueprints, and orchestrator workflows across vCenters and VCF environments.
- **Infrastructure policy** - A policy that defines VM placement on specific hosts, building upon vCenter compute policies to enforce placement and resource rules in the vSphere Supervisor context. Can be mandatory or optional.
- **KubeVM Service (formerly, VM Service)** - The runtime built natively in VCF that provisions and manages Virtual Machines using Kubernetes declarative APIs.
- **Namespace (vSphere Namespace)** - A resource envelope that defines resource limits (CPU, memory, storage) for workloads within an organization. Each namespace can be assigned one or more VPCs.
- **Namespace Class** - A template used to create vSphere Namespaces, defining and standardizing resource allocations and governance settings.
- **Organization** - A top-level entity representing a line of business or tenant. Each organization operates in a secure, isolated environment with access only to allocated services and resources.
- **Policy as Code** - YAML-based custom IaaS resource policies built on native Kubernetes Validating Admission Policy, enabling declarative, version-controllable governance.
- **Project** - A grouping within an organization for a specific LOB or application team. Projects give LOB Admins scoped control over compute, storage, and networking within the guardrails set by the provider and Org Admin.
- **Provider/consumer model** - The separation of roles in VCF Automation: the provider (Enterprise IT Admin) manages shared infrastructure and creates organizations; the consumer (Org Admin, LOB Admin, developer) consumes resources within guardrails.
- **Region** - A combination of compute, storage, memory, and networking resources from one or more Supervisors across one or more vCenter instances managed by the same NSX Manager instance. All Supervisors within a region must have homogeneous configurations.
- **Region quota** - The compute and storage resources allocated to an organization for a given region, including CPU and memory utilization limits, reservations, VM classes, and storage classes.
- **Supervisor (vSphere Supervisor)** - A Kubernetes-based control plane that runs directly on the hypervisor, powering the new multi-tenant, Kubernetes-native capabilities in VCF 9.
- **Validating Admission Policy** - A Kubernetes-native policy enforcement mechanism that inspects and validates API requests before resources are created or modified. VCF Automation uses it as the engine for Policy as Code, allowing Org Admins to write declarative YAML policies that enforce configuration standards, compliance rules, and resource constraints across tenants.
- **VCD (VMware Cloud Director)** - A cloud management platform for service providers that delivers multi-tenant cloud services. VCF 9 multi-tenancy builds on VCD concepts but introduces a Kubernetes-based architecture.

- **VCF Automation** - The automation and consumption component of VMware Cloud Foundation 9 that enables IT to deliver a self-service private AI cloud for application teams to build, run, and manage AI, Kubernetes, and VM-based applications.
- **vDefend (VMware vDefend)** - A security solution integrated with VCF 9.1 that provides Distributed and Gateway Firewalls driven by built-in Security Profiles. Security rules are enforced in a strict top-down hierarchy from provider to organization to VPC level.
- **VKS (VMware vSphere Kubernetes Service)** - The VMware Kubernetes runtime, available out-of-the-box in VCF 9, helping avoid the need for a separate Kubernetes platform.
- **VPC (Virtual Private Cloud)** - A network isolation domain within the shared VCF fabric, providing tenant-level network segmentation and self-service networking capabilities.
- **Zone** - A vSphere cluster that provides compute resources for Supervisors. A Supervisor can span multiple zones. Zones are created and managed in vCenter.

