

VCF Automation 9.1

für IT und Entwickler

Christian Ferber

Solution Architect Cloud Management, Broadcom

July 2026

VMware Cloud Foundation

Most trusted private cloud platform
for all applications

CONTAINERS
NATIVE

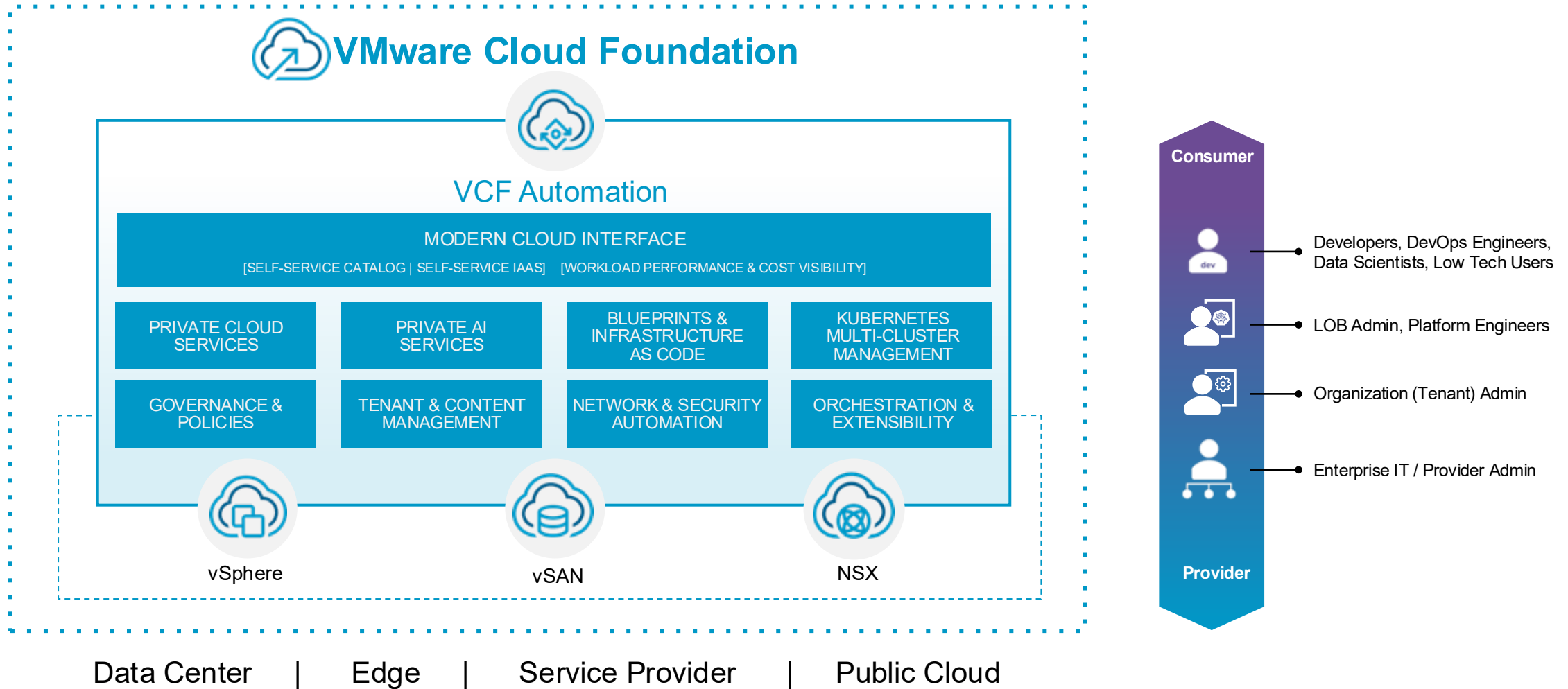


VMS
NATIVE

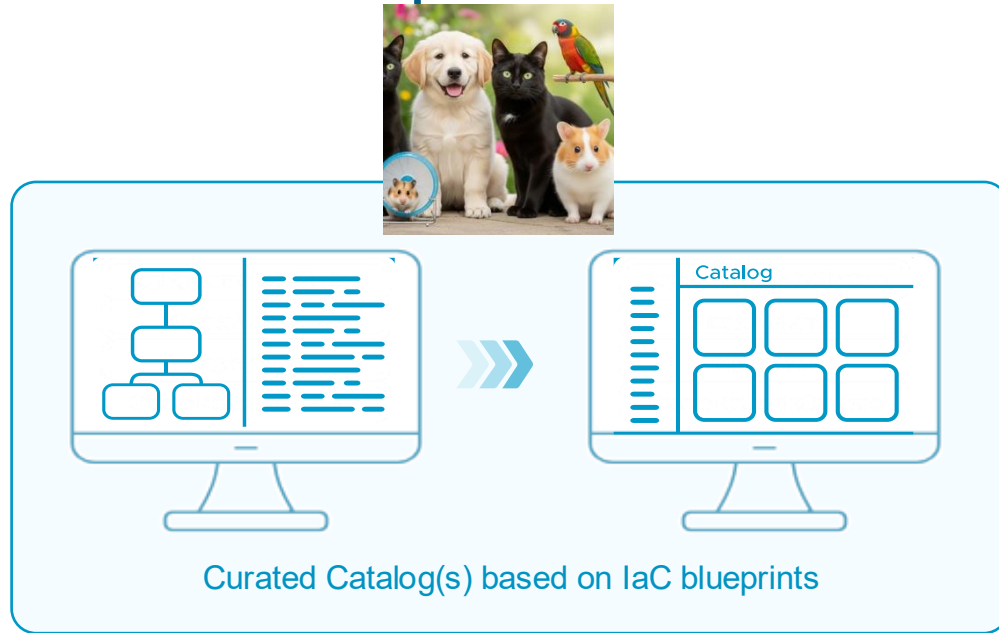


The VCF Automation Solution Overview

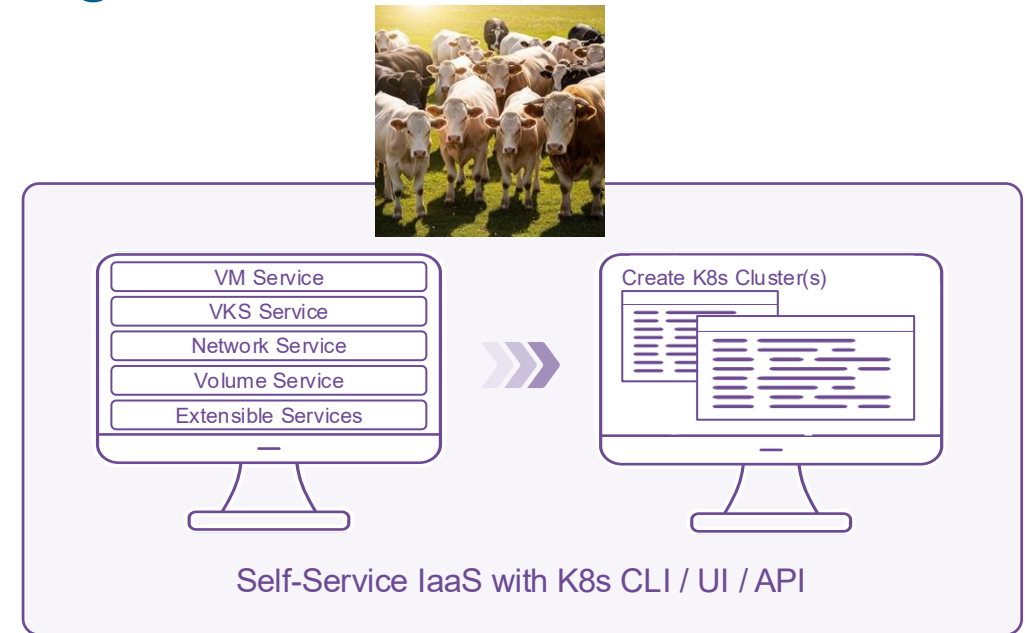
Enable self-service consumption to build, run, and manage AI, K8s, and VM-based workloads



Self Service Options - IaaS and Catalog



- Catalog items are backed by templates that are developed and published to Project teams
- Suitable for users who are Low-Tech



- Create resources like VMs, K8s, Volume, Network, Storage – on-demand
- Suitable for users who are Hi-Tech

More Admin Controlled For
Low-tech Users

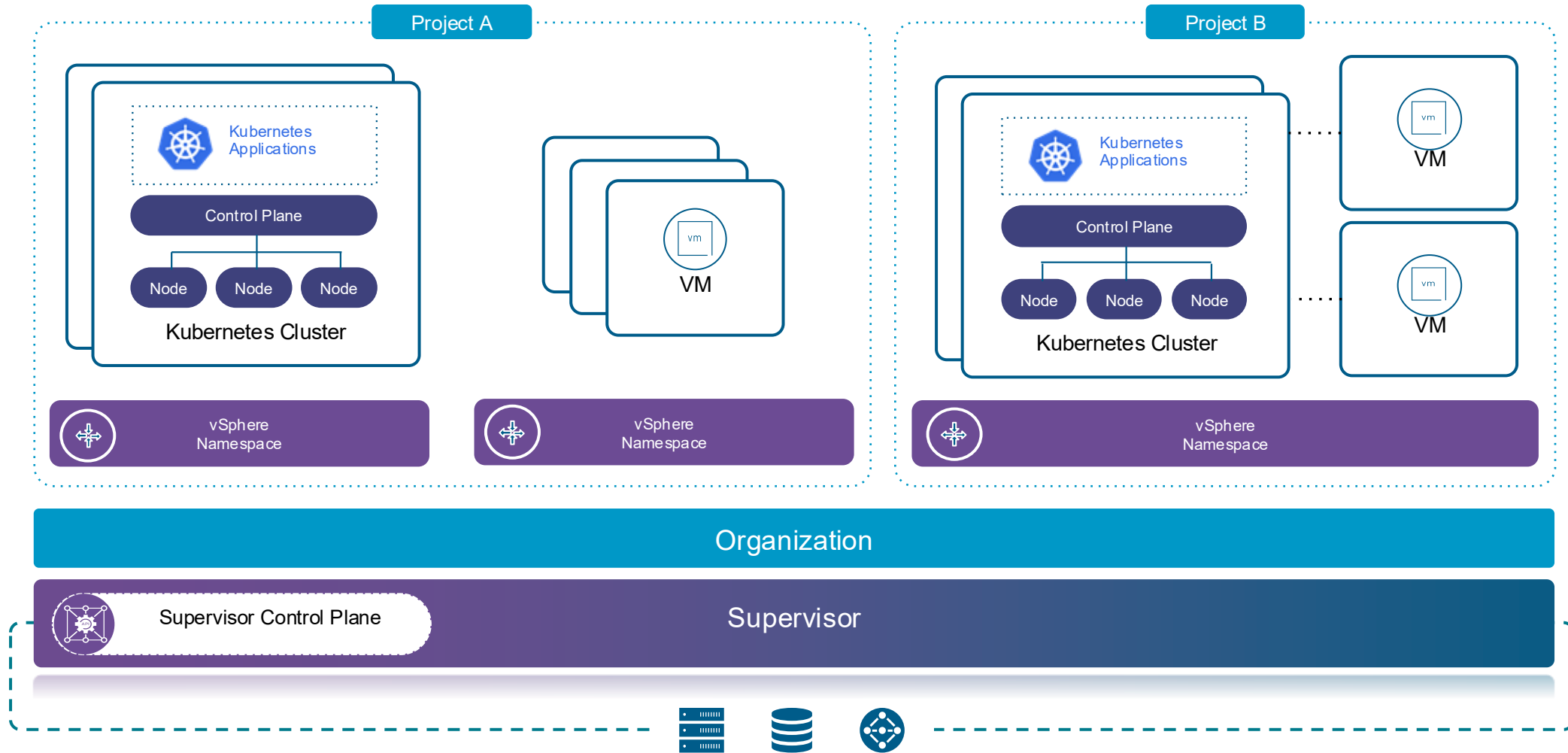
CATALOG SELF-SERVICE

More Developer Controlled For
Hi-tech Users

IAAS SELF-SERVICE

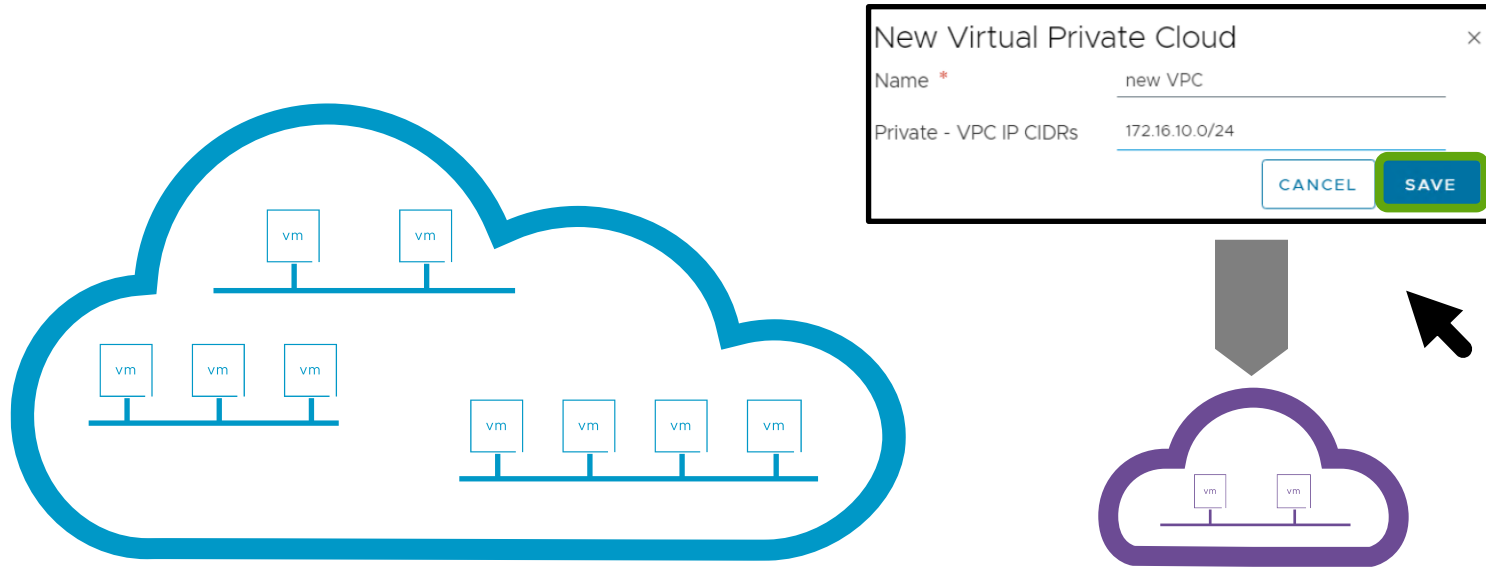
Supervisor Consumption through VCF Automation

Single Organization inside of VCF Automation consuming one Supervisor



VPC Networking for virtual network architecture

Based on NSX



private network – communication within VPC

transit network – communication cross VPC

public network – availability from outside VPC

Catalog-based Consumption

Catalog

Self-Service Experience

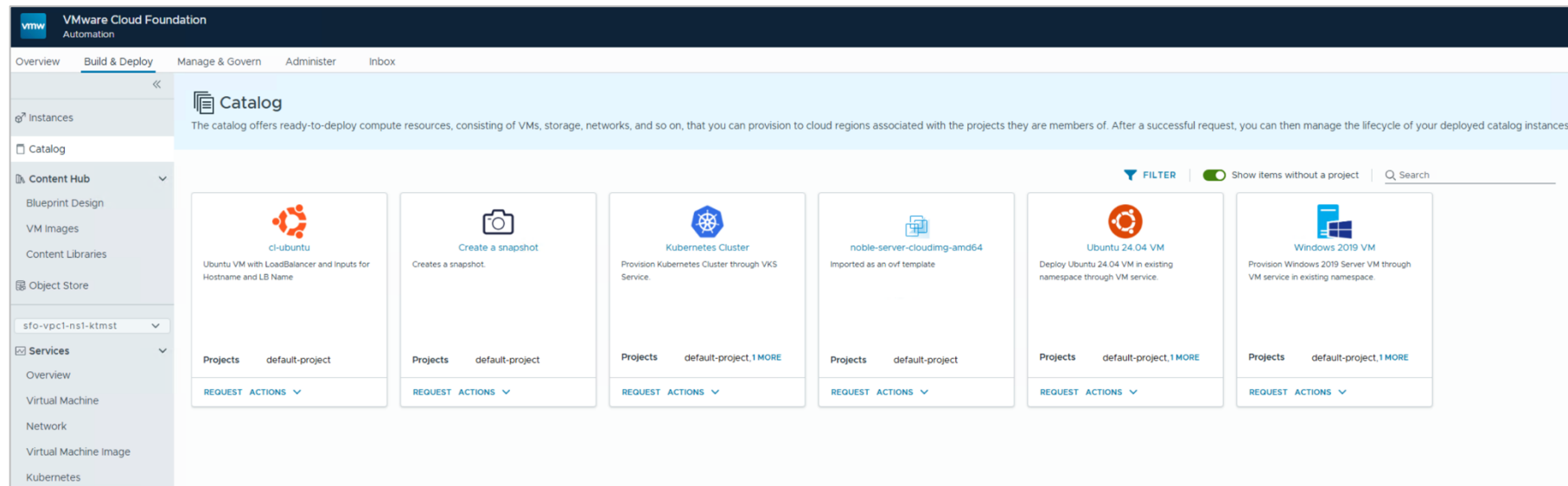


Self-service IaaS for developers, app owners, etc.

Standard deployments with blueprints

Tenant autonomy through role-based access

Central IT governance



Catalog

Customized Requests

VMs, VKS Clusters, Databases,
AI Workloads, Modern
Applications, and Anything-as-a-
service

Blueprints

OVA/OVF Images

Orchestrator Workflows

Policy Controlled



The screenshot shows the VMware Cloud Foundation Automation interface. The top navigation bar includes 'Overview', 'Build & Deploy' (selected), 'Manage & Govern', 'Orchestrate', 'Administer', and 'Inbox'. The left sidebar shows a 'Catalog' menu with options like 'Content Hub', 'Event Subscriptions', 'Object Store', and 'Services'. The main area is titled 'New Request' and shows a form for creating an 'Ubuntu Virtual Machine'. The form fields are as follows:

Field	Value
Project	acme
Deployment Name	Base Ubuntu 22.04
Virtual Machine Name	vm2
Subnet Type	VPC Private Subnet
Virtual Machine Size	Extra Small (2 vCPUs - 2 GiB MEM)
Virtual Machine Image	Ubuntu-22.04-LTS-jammy-server
Target Namespace	acme-ns-1

At the bottom of the form are 'SUBMIT' and 'CANCEL' buttons.

Catalog

Blueprint Design



Customizable blueprints,
drag&drop

Git import

Property groups, custom
resources, resource actions

Custom forms

Version control

IaC YAML Code (Kubernetes
format)

VMware Cloud Foundation Automation

Overview Build & Deploy Manage & Govern Orchestrate Administer Inbox

Ubuntu Virtual Machine Published versions 1 VERSION HISTORY CREATE CUSTOM FORM ALL ACTIONS

Search: e.g. machine

- Supervisor Namespace
- Supervisor Resource
- Virtual Machine
- Virtual Machine Service
- Secret
- Persistent Volume Claim
- VKS Cluster

Namespace 2

- Bootstrap_Sec...
- Virtual_Machine...

Built-in Custom Form

Drag-n-Drop Blueprint Elements

Version Control and Publishing

Nested Blueprint Elements

IaC YAML Code

```
60 stringdata:
61   demouser-passwd: $1$FmYmnb$UIV3KrTd2ZsKgQub3wxLi.
62 Virtual_Machine:
63   type: CCI.Supervisor.Resource
64   dependsOn:
65     - Bootstrap_Secret
66   properties:
67     context: ${resource.Namespace.id}
68   manifest:
69     apiVersion: vmoperator.vmware.com/v1alpha3
70     kind: VirtualMachine
71     metadata:
72       name: ${input.vm_name}
73       labels:
74         vm-selector: ${input.vm_name}
75     spec:
76       className: ${input.vm_size}
77       imageName: ${input.vm_image}
78       storageClass: wcpglobal-storage-profile
79       powerState: PoweredOn
80       bootstrap:
81         cloudInit:
82           cloudConfig:
83             defaultUserEnabled: true
84             ssh_pwauth: true
85             users:
86               - name: demouser
87                 passwd:
88                   name: ${input.vm_name}-bootstrap-secret
89                   key: demouser-passwd
90                   lock_passwd: false
91                   sudo: ALL=(ALL) NOPASSWD:ALL
92       network:
93         interfaces:
94           - name: eth0
95             network:
96               name: ${input.subnet_type}
97               kind: Subnet
```

DEPLOY TEST VERSION CLOSE Last saved 5 days ago

Direct Kubernetes-based Consumption



Direct Kubernetes-based Consumption

Overview



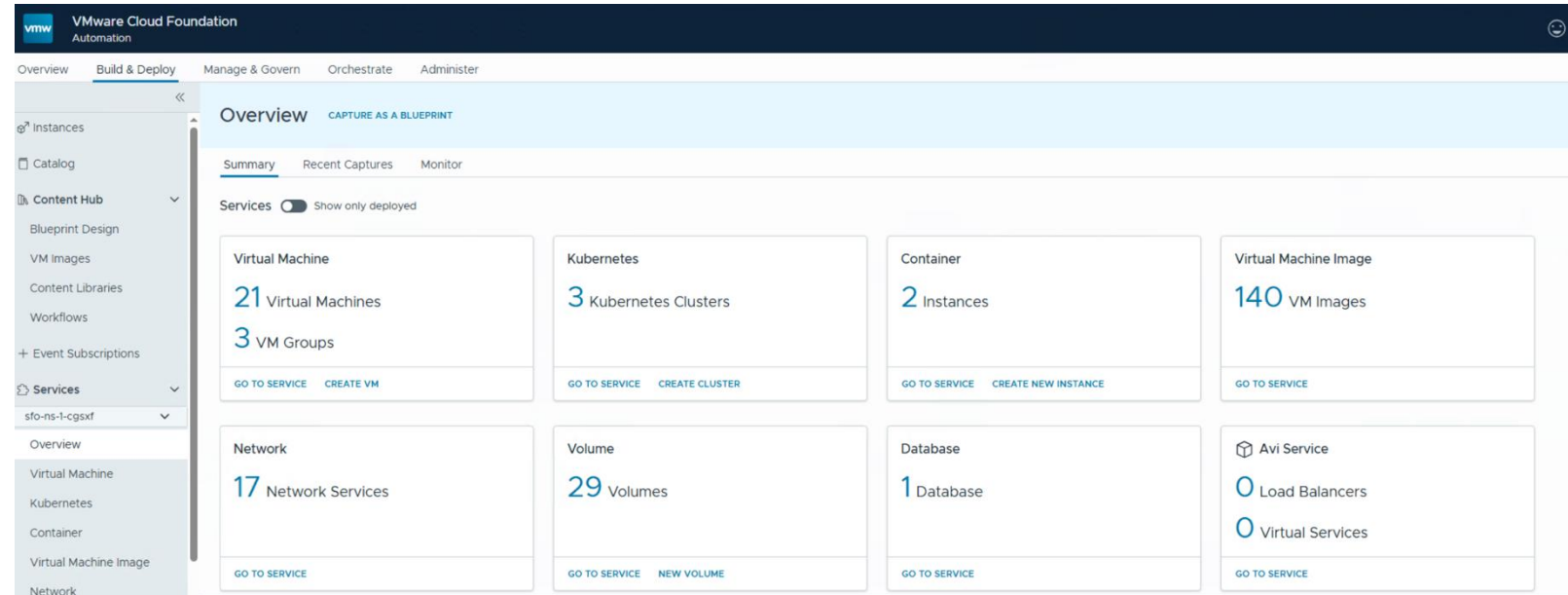
Summary page for infra services offered

Admin and advanced project user roles access Services via the UI, API or VCF command line

Namespace selection

Overview of services provisioned

Detailed provisioning workflow



Services Provisioning

VM Service

Deploy from ISO or OVA formats.

Advanced settings and Network Configurations

Image selection

VM Class, Storage Class, run scripts etc.



The screenshot displays the VMware Cloud Foundation Automation interface. The main panel shows the 'VM Class' configuration page. A table lists available VM classes with columns for Name, CPU, CPU Reservation, Memory, and Memory Reservation. The 'best-effort-xsmall' class is selected. Below the table, the 'Storage Class' is set to 'vcf-wld1-cl01-optimal-datastore-default-policy-raid1' and the 'Power State' is 'Powered On'. A 'Kubernetes Resource YAML' panel on the right shows the generated YAML for the VM. The bottom of the screen shows a progress bar with steps: 1. Overview, 2. Build & Deploy (current), 3. Advanced Settings (Optional), and 4. Network Configuration (Optional).

Name	CPU	CPU Reservation	Memory	Memory Reservation
☒ best-effort-xsmall	2 vCPUs	No Reservation	2 GiB	No Reservation
☐ best-effort-small	2 vCPUs	No Reservation	4 GiB	No Reservation
☐ best-effort-medium	2 vCPUs	No Reservation	8 GiB	No Reservation
☐ best-effort-large	4 vCPUs	No Reservation	16 GiB	No Reservation
☐ best-effort-xlarge	4 vCPUs	No Reservation	32 GiB	No Reservation

```
1 apiVersion: vmoperator.vmware.com/v1alpha3
2 kind: VirtualMachine
3 metadata:
4   name: vm-ojew
5   namespace: development-xc2s9
6   labels:
7     vm-selector: vm-ojew
8 spec:
9   className: best-effort-xsmall
10  # Friendly image name: ubuntu-22.04-lts-jam
11  imageName: vmi-4d31f55e92e236ecd
12  storageClass: vcf-wld1-cl01-optimal-datastore
13  powerState: PoweredOn
14
```

Services Provisioning

Kubernetes Services



Cluster Type selection

Kubernetes-specific data like version, node numbers etc.

Kubernetes YAML for Provisioning Kubernetes clusters

VMware Cloud Foundation Automation

Overview Build & Deploy Manage & Govern Orchestrate Administer Inbox

1. Configuration Type Cluster Type Configuration Type Cluster API Default Configuration

2. Review and Confirm Review all the details before you deploy this cluster

Your cluster will be deployed with the following configuration:

Kind	Cluster
Cluster Name	kubernetes-cluster-qkc5
Cluster Class	builtin-generic-v3.3.0
Kubernetes release	v1.32.0---vmware.6-fips-vkr.2
Control Plane	
Replicas	1
VM Class	best-effort-xsmall
Storage Class	vcf-wld1-cl01-optimal-datastore-default-policy-raid1
OS Image	Photon
Nodepools	
kubernetes-cluster-qkc5-np-771t	
Class	node-pool
Replicas	1
VM Class	best-effort-xsmall
Storage Class	vcf-wld1-cl01-optimal-datastore-default-policy-raid1
OS Image	Photon

Review the configuration and the YAML file generated. Then, click FINISH to start deploying this cluster.

FINISH

Kubernetes Resource YAML

```
1 apiVersion: cluster.x-k8s.io/v1beta1
2 kind: Cluster
3 metadata:
4   name: kubernetes-cluster-qkc5
5   namespace: development-xc2s9
6 spec:
7   clusterNetwork:
8     pods:
9       cidrBlocks:
10        - 192.168.156.0/20
11   services:
12     cidrBlocks:
13       - 10.96.0.0/12
14   serviceDomain: cluster.local
15 topology:
16   class: builtin-generic-v3.3.0
17   version: v1.32.0---vmware.6-fips-vkr.2
18   variables:
19     - name: vmClass
20       value: best-effort-xsmall
21     - name: storageClass
22       value: vcf-wld1-cl01-optimal-datastore-default-policy-raid1
23   controlPlane:
24     replicas: 1
25     metadata:
26       annotations:
27         run.tanzu.vmware.com/resolve-os-image: os-na
28   workers:
29     machineDeployments:
30       - class: node-pool
31         name: kubernetes-cluster-qkc5-np-771t
32         replicas: 1
```

Download all YAMLs to a .zip

Service Provisioning

Containers

The screenshot shows the VMware VCF Automation interface for creating a new vSphere Pod. The left sidebar contains navigation links: Overview, Manage & Govern, Build & Deploy (active), Orchestrate, and Administration. Under Build & Deploy, there are links for Instances, Catalog, Content Hub (with sub-links for Blueprints, VM Images, Content Libraries, and Workflows), Event Subscriptions, and a dropdown menu currently showing 'dev-ns'. The main content area is titled 'New vSphere Pod' and includes a 'BACK' button. The form is divided into sections: 1. vSphere Pod Information, where 'vSphere Pod Name' is set to 'test-pod' and 'Zone' is set to 'Automatic'; Container Image, with instructions on providing a URL and pull secret; Registry Authentication, where 'Authentication required' is checked and 'Select Existing Secret' is chosen; a table of available secrets; and 'Container Image', where the full image path is entered.

Registry	Secret Name
☐ docker.io	docker-secret
☒ harbor.myteam.com	harbor-secret

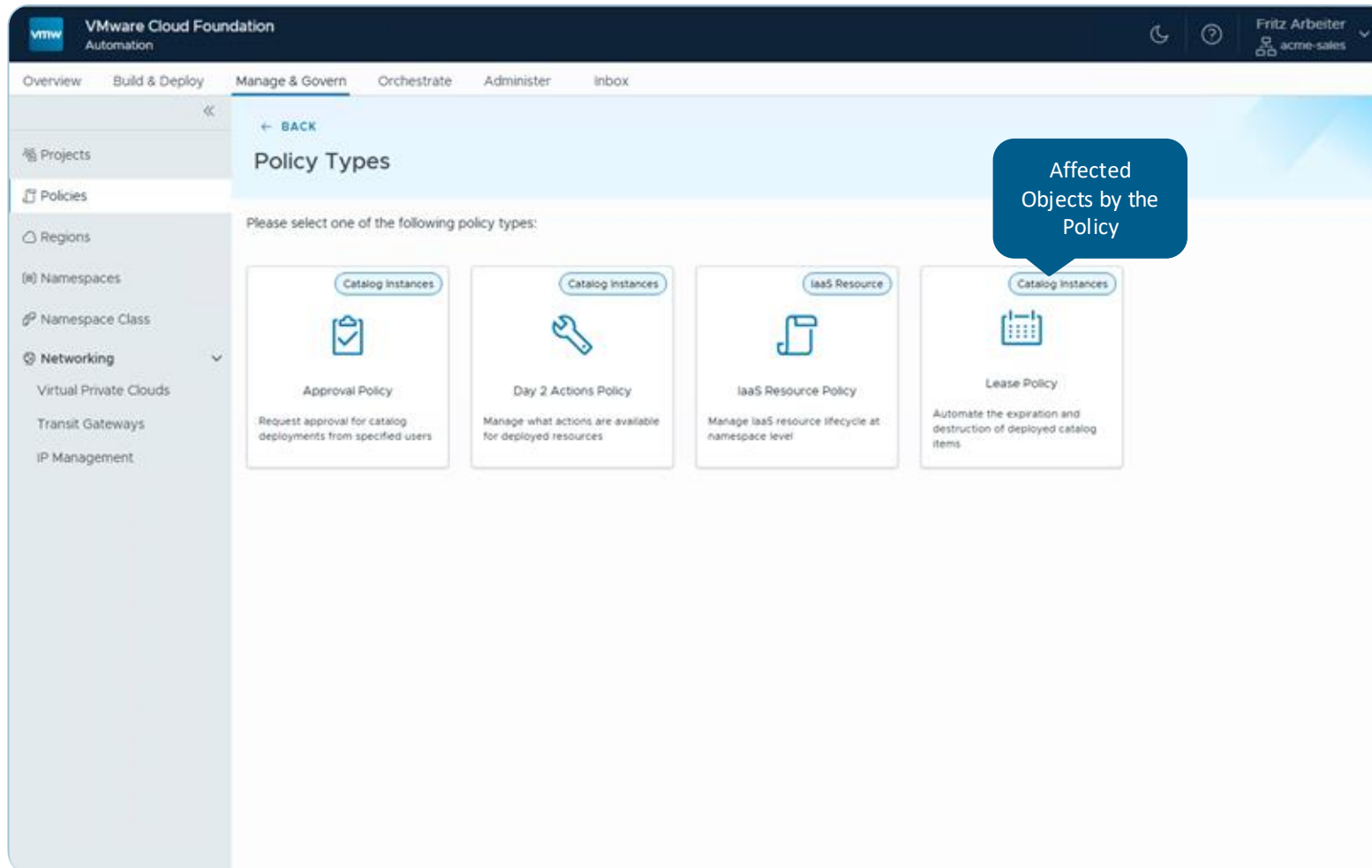
Run containers instantly without standing up or managing a full Kubernetes environment.

Deliver container agility with the security, visibility, and governance of the vSphere platform.

Simplified Container Management: UI-driven provisioning and lifecycle control of Container Instance within vSphere Namespaces.

Resource Management and Architecture

Manage and Govern Policies



Policies provide governance for the organization's infrastructure and operations.

Policies can apply to catalog instances, individual instances, or both, depending on what type of access is granted to users.

Select from the following policy types:

- Catalog Instances Centric (i.e., Deployments):
 - Approval Policy
 - Lease Policy
 - Day 2 Actions Policy
- IaaS Resource Centric:
 - IaaS Resource Policy

vSphere Kubernetes Multi-cluster management

Integrated into VCF Automation (previously Tanzu Mission Control, TMC)



**Automate
manual toil**

1. Fleet-wide multi-cluster management



**Secure your
environments**

2. VKS cluster policy management



**Protect your
data**

3. Data Protection

Lab environments – Capture / deploy full stack

Create reusable sandbox environments with Appstack Formations

The screenshot displays the VMware Cloud Foundation Automation interface. The top navigation bar includes 'Overview', 'Manage & Govern', and 'Build & Deploy'. The left sidebar shows a navigation menu with 'Instances', 'Catalog', 'Content Hub', 'Event Subscriptions', 'Services', and 'Overview' (selected). The main content area is titled 'Capture "Namespace Name"' and shows a 'Blueprint Details' form. The form includes fields for 'Blueprint Name' (App-1), 'Description' (a text area with a placeholder), 'Content Library' (Content Lib 1), 'Capture Customization' (radio buttons for 'Capture as an identical copy' and 'Capture with customizations'), 'Compute and Storage' (checkbox for 'Allow catalog users to pick a different VM and Storage Class for each VM'), and 'Guest OS Settings' (checkbox for 'Configure and enable guest customization options for catalog users'). A 'CONFIGURE' button is visible. Below the form is a table with columns 'VM Name' and 'Configuration'.

VM Name	Configuration
VirtualM-1	Not configured
VirtualM-2	Not configured
VirtualM-3	Not configured

vSphere Namespaces can be captured as consistent blueprints including the infrastructure. (Disks, Networks,...)

Blueprints can be deployed or published to a catalog and redeployed to same or different projects.

Demo



Broadcom Proprietary and Confidential. Copyright © 2026 Broadcom.
All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.





Thank You