VCF TechCon

Powered by VMUG

Kubernetes and Cloud Foundry Defining Your Cloud-Native Strategy

Marc van de Logt

VMUGNL VMware User Group

PQR

Introduction

Marc van de Logt
Technical Architect
https://www.marcvandelogt.com/

- SDDC & Cloud Native
- VMware vExpert *****
- VMware Education Contributor (SME)
- Tanzu Vanguard Member
- CTAB Member
- VMUG Leader

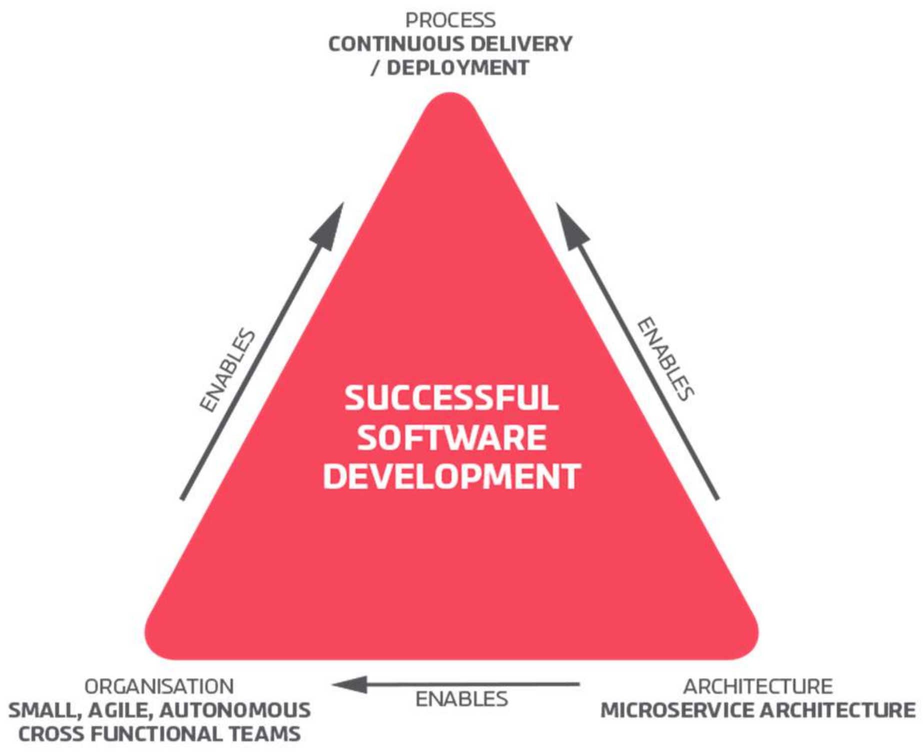
VCF TechCon
Powered by VMUG

Agenda

- Why Modern Apps and Microservices?
- What are Containers?
- Challenges with Modern Apps
- Kubernetes Architecture
- Cloud Foundry Architecture
- vSphere with Kubernetes Service (VKS)
- Tanzu Platform for Cloud Foundry
- Demo

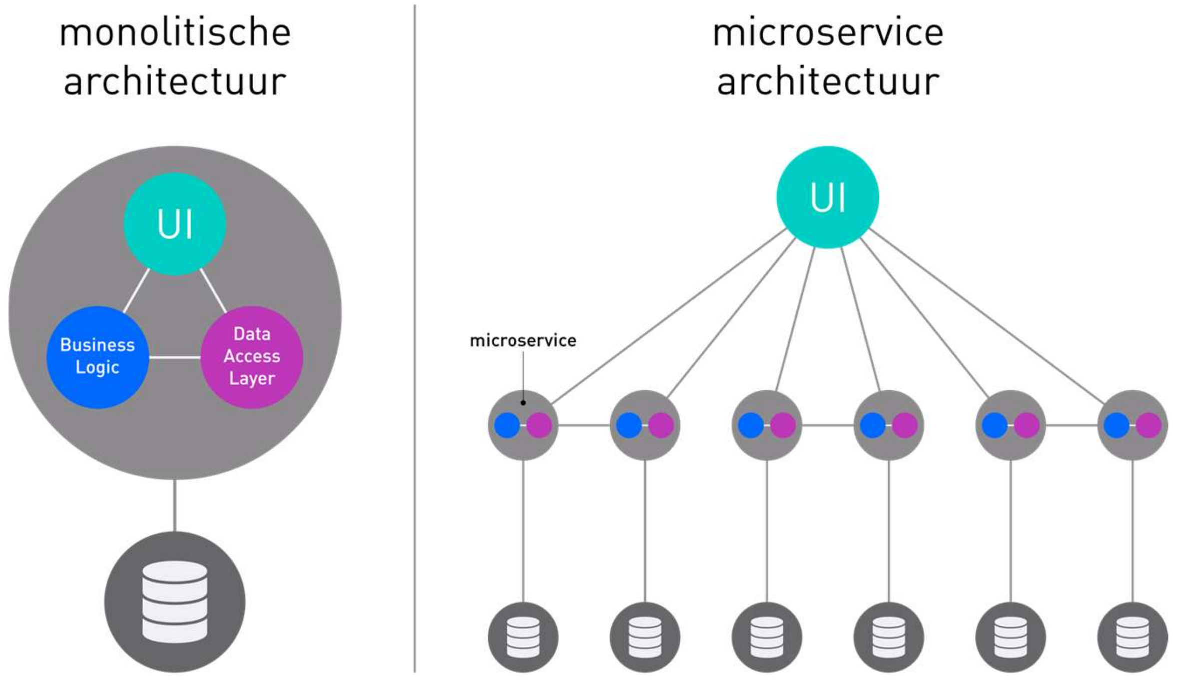
Why Modern Apps and Microservices?

- Consistency from code to production
- Quicker release management
- Abstraction from the infrastructure
- Focus on functionality
- No silo's in the organization
- A faster service for end-users
- Standardization on every platform


PROCESS
CONTINUOUS DELIVERY / DEPLOYMENT
ENABLES
ENABLES
SUCCESSFUL SOFTWARE DEVELOPMENT
ORGANISATION
SMALL, AGILE, AUTONOMOUS CROSS FUNCTIONAL TEAMS
ENABLES
ARCHITECTURE
MICROSERVICE ARCHITECTURE

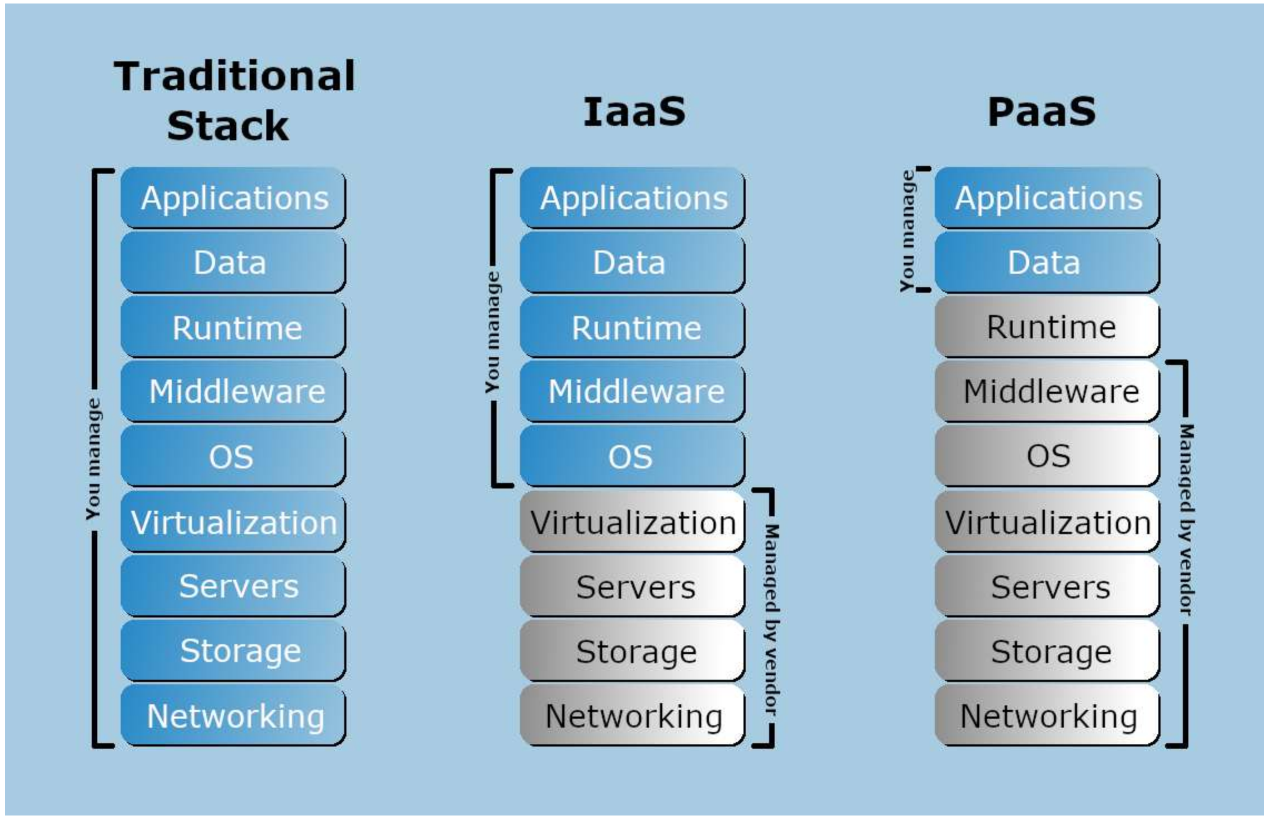
Why Modern Apps and Microservices? (2)

- Traditional Apps:
  - Consists of virtual machines
  - Difficult to scale
  - Long and complex software releases
  - Single code unit
- Modern Apps:
  - Consist of containers
  - Easier to scale
  - Faster software releases
  - Code independent



Why Modern Apps and Microservices? (3)

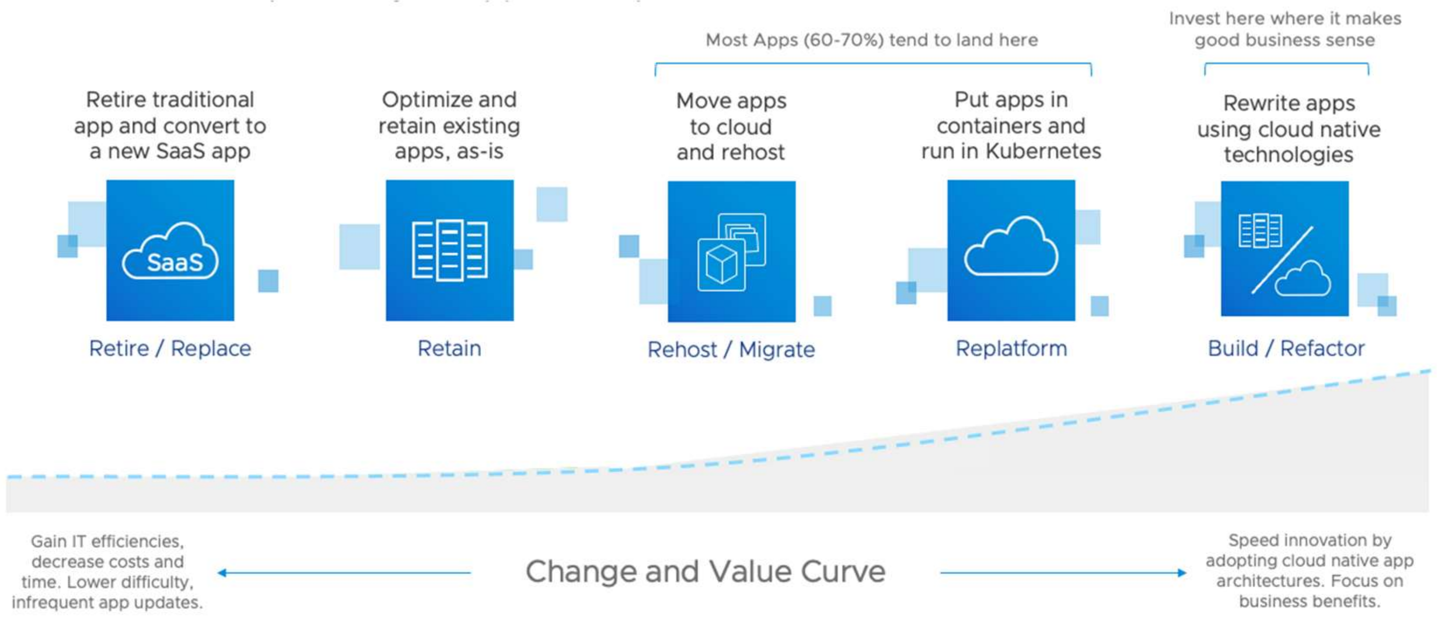
- Traditional Stack:
  - Everything
- IaaS:
  - Applications
  - Data
  - Runtime
  - Middleware
  - OS
- PaaS:
  - Applications
  - Data



Why Modern Apps and Microservices? (4)

The Five Rs:

- Refactor
- Replatform
- Rehost
- Retain
- Retire

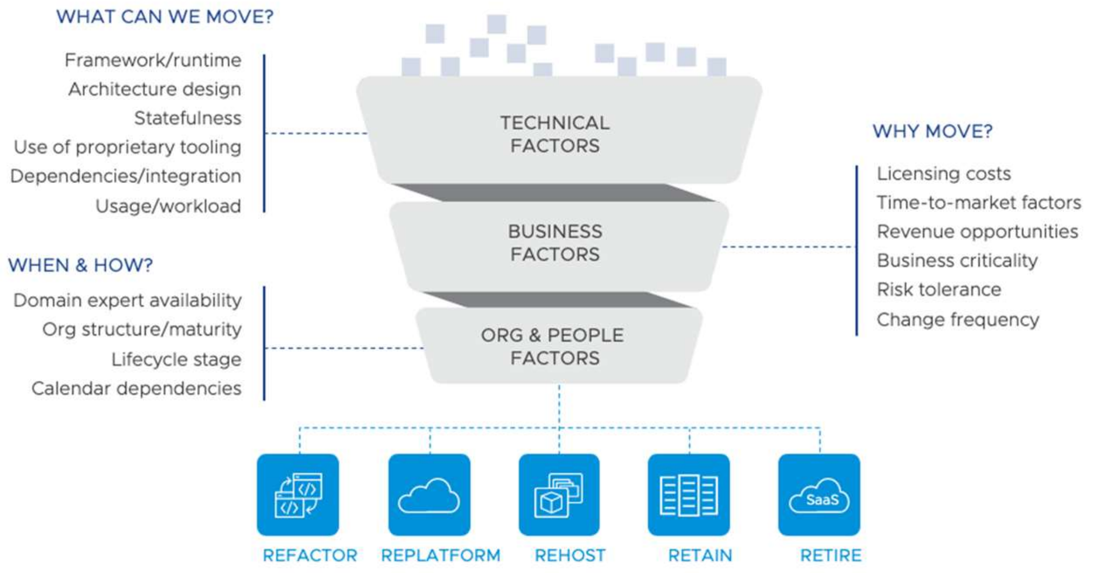


Most Apps (60-70%) tend to land here
Invest here where it makes good business sense
Retire traditional app and convert to a new SaaS app
Optimize and retain existing apps, as-is
Move apps to cloud and rehost
Put apps in containers and run in Kubernetes
Rewrite apps using cloud native technologies
SaaS
Retire / Replace
Retain
Rehost / Migrate
Replatform
Build / Refactor
Gain IT efficiencies, decrease costs and time. Lower difficulty, infrequent app updates.
Change and Value Curve
Speed innovation by adopting cloud native app architectures. Focus on business benefits.

Why Modern Apps and Microservices? (5)

Three main questions:

- Why should we move apps?
- What kind of apps can we move
- When and how do we move apps?



What are Containers?

A standard software unit

An abstraction on the application layer

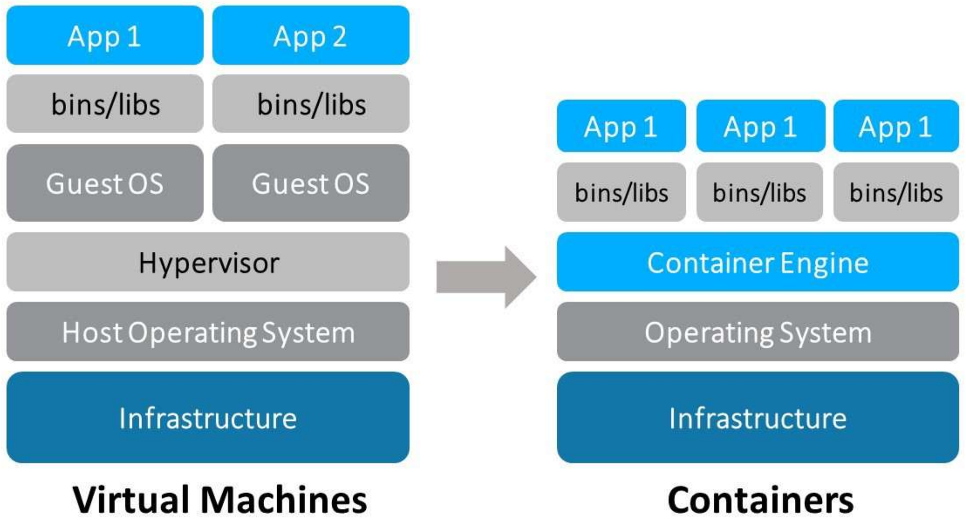
Consists of one or more processes

Code and dependencies together

Runs on a container runtime

Smaller footprint then virtual machines

Docker is often used as a container runtime



Challenges with Modern Apps

Infrastructure Provisioning:

- Compute, network and storage

Day 2 operations:

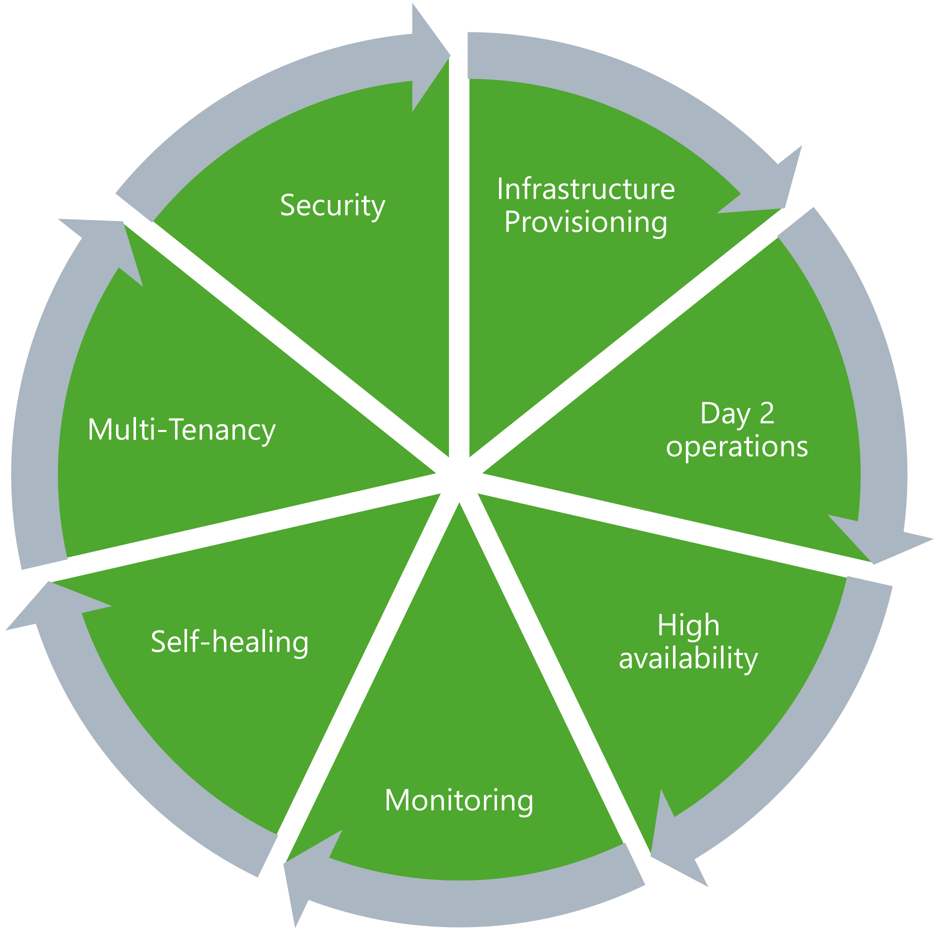
- CVEs, upgrades and automation

High availability:

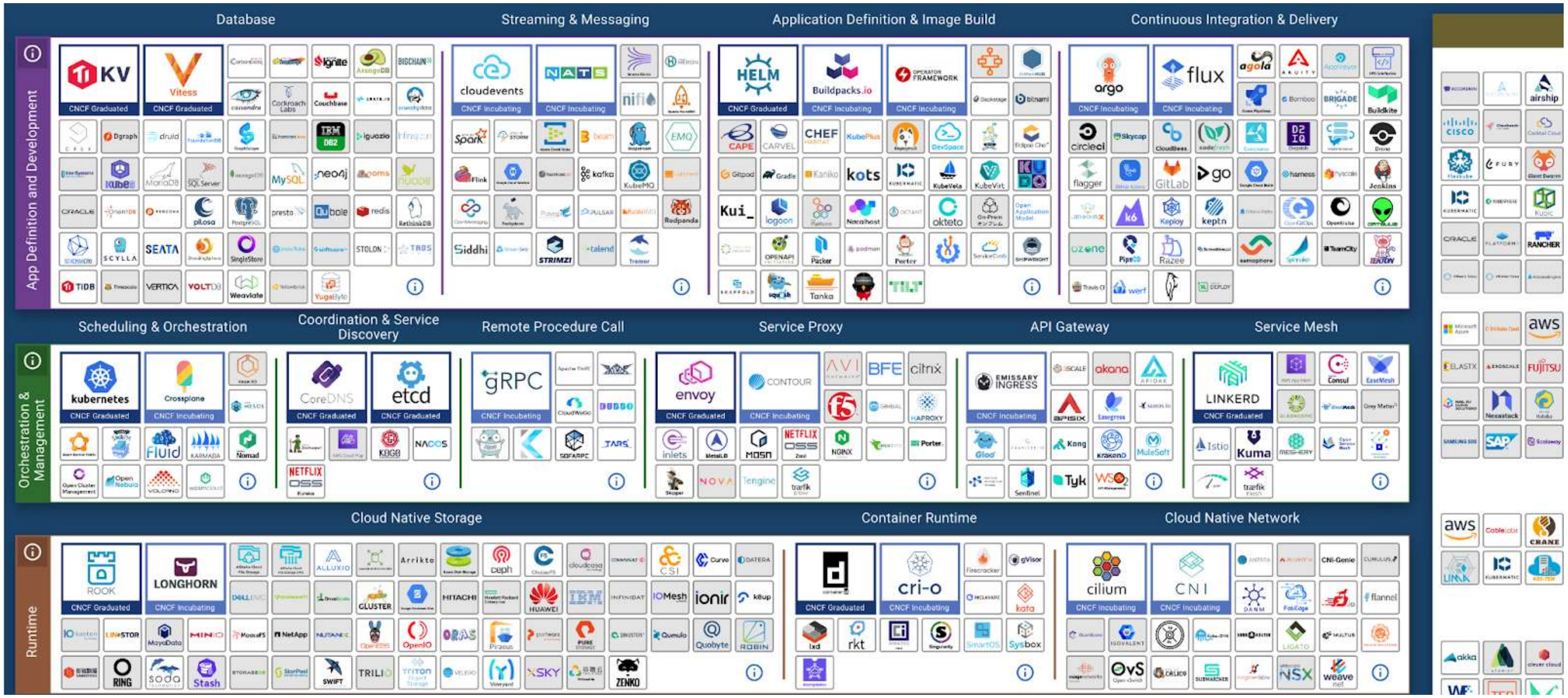
- Monitoring, self-healing and scaling

Security:

- The role of SecOps and multi-tenancy
- A lot of tools and the complexity to manage threats



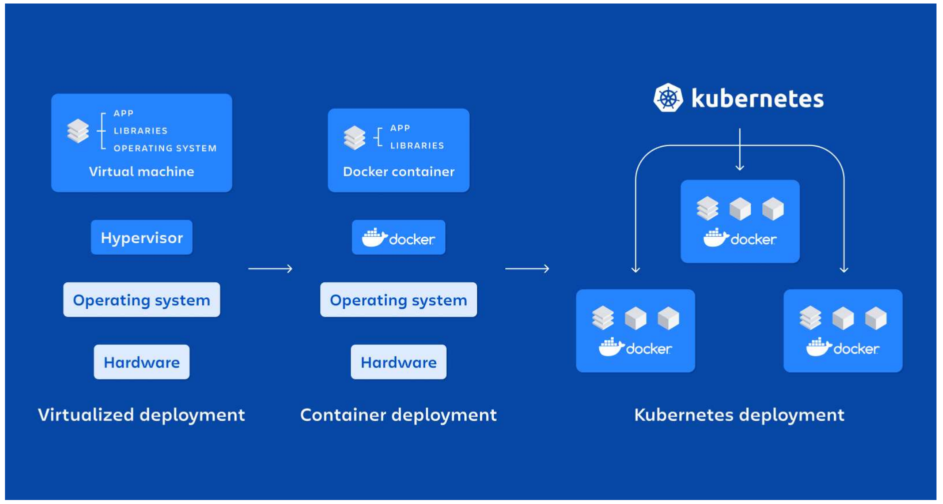
Challenges with Modern Apps (2)



Kubernetes Architecture

What is Kubernetes (k8s) and what does it actually?

- Software for automation of containers
- Simplifies the management of containers
- Functions on every type of infrastructure
- Helps Developers and IT Operations to work together
- Scalability of container applications
- Delivers network, storage and security constructs
- API driven open-source solution



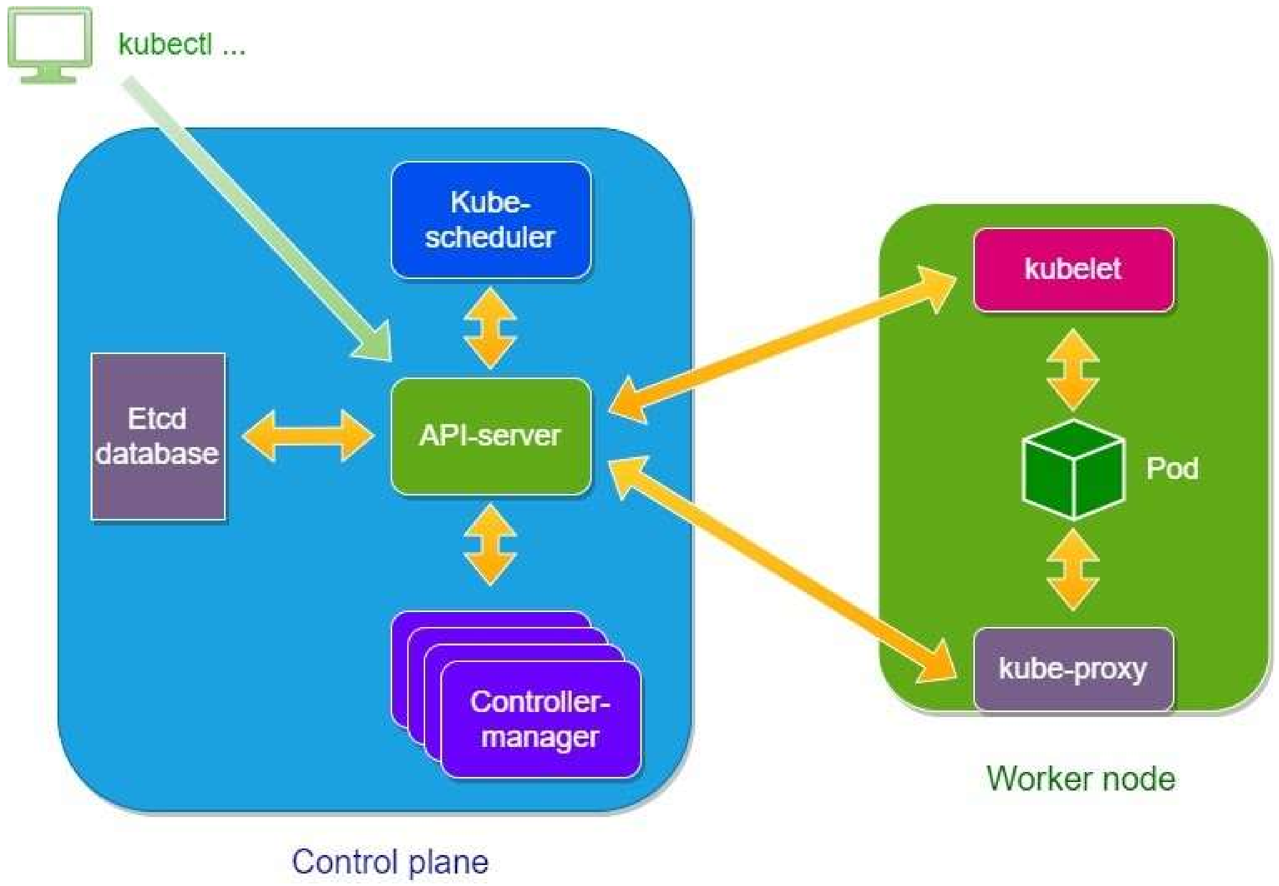
Kubernetes Architecture (2)

Control Plane/Master nodes:

- API-server
- Etcd
- Controller-Manager
- Kube-Scheduler

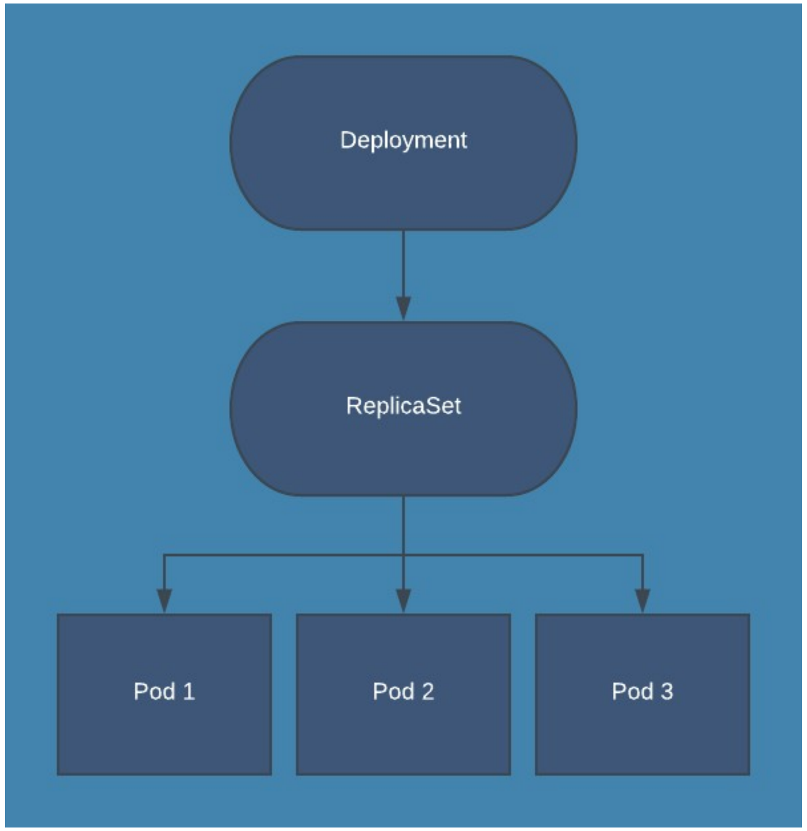
Worker nodes:

- Kubelet
- Kube-proxy
- Pods

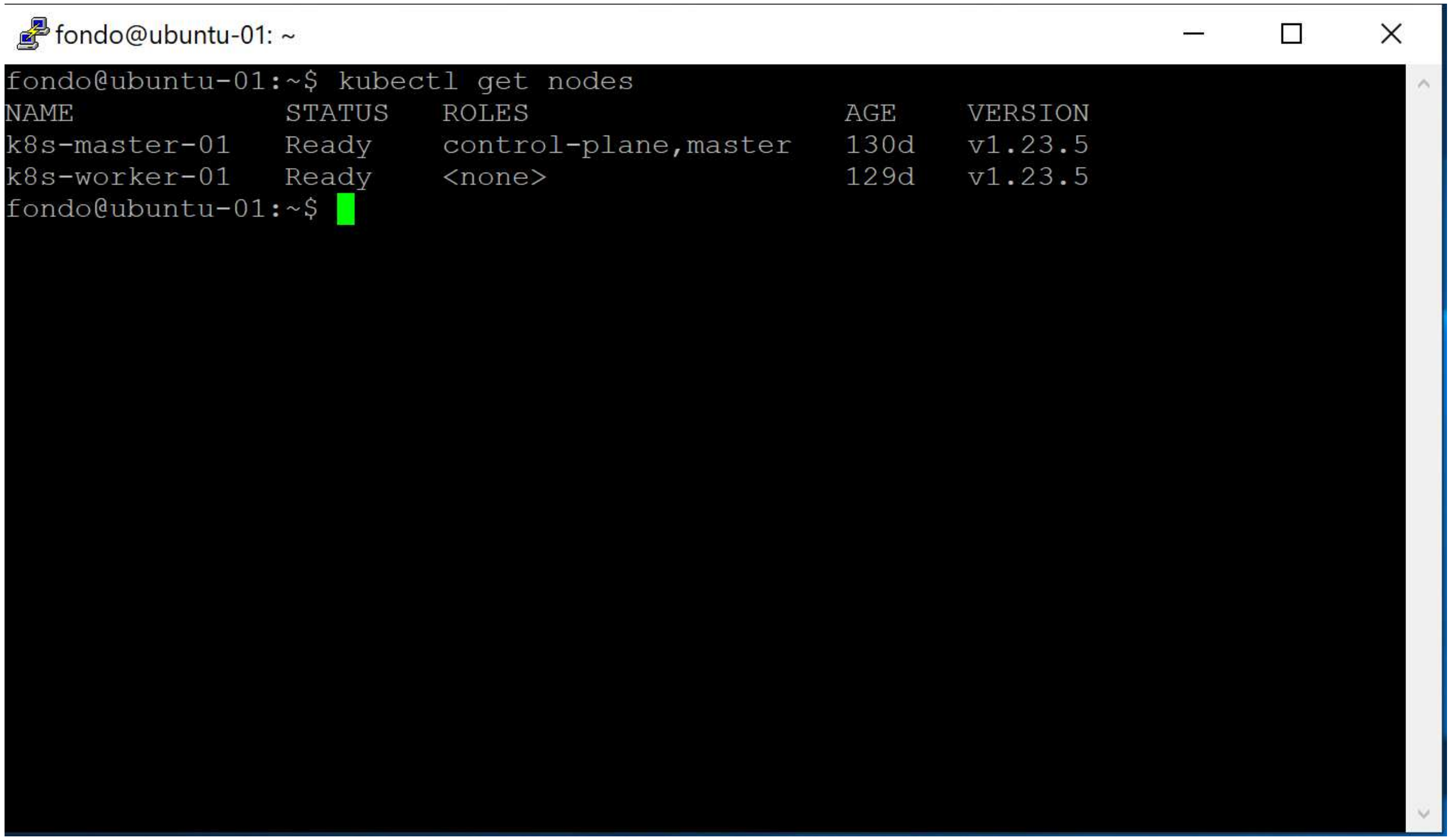


Kubernetes Architecture (3)

- Pods
- ReplicaSets
- Deployments
- Namespaces
- Services
- Ingress
- Persistent Volumes



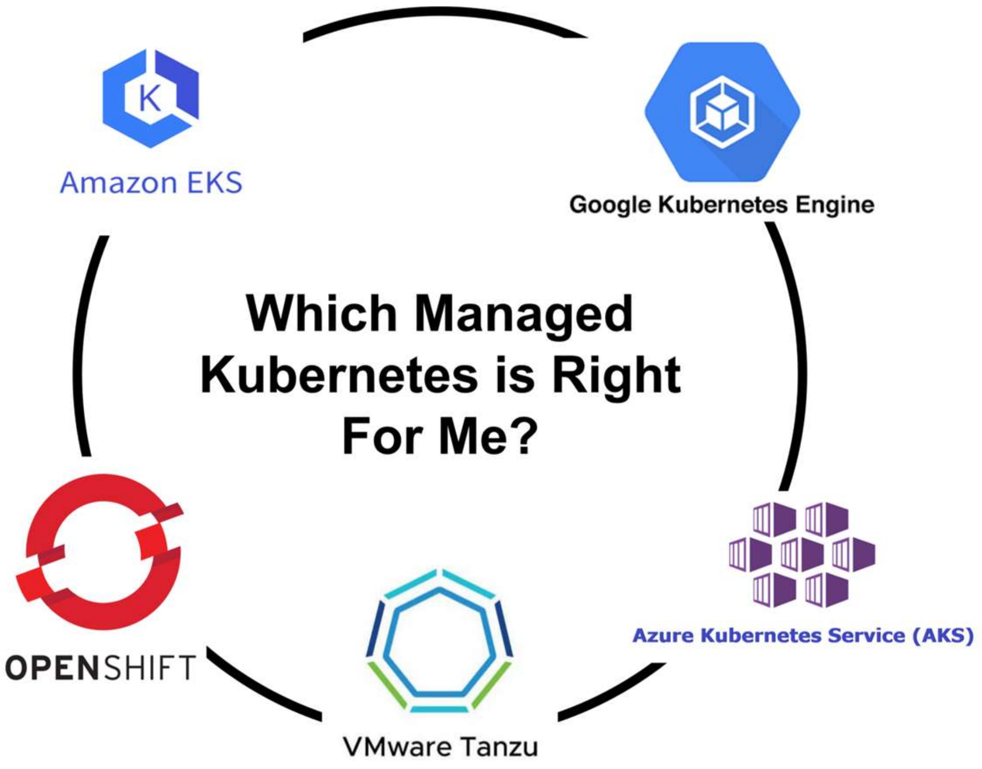
Kubernetes Architecture (4)



```
fondo@ubuntu-01:~$ kubectl get nodes
NAME            STATUS   ROLES                   AGE    VERSION
k8s-master-01   Ready    control-plane,master    130d   v1.23.5
k8s-worker-01   Ready    <none>                  129d   v1.23.5
fondo@ubuntu-01:~$
```

Kubernetes Architecture (5)

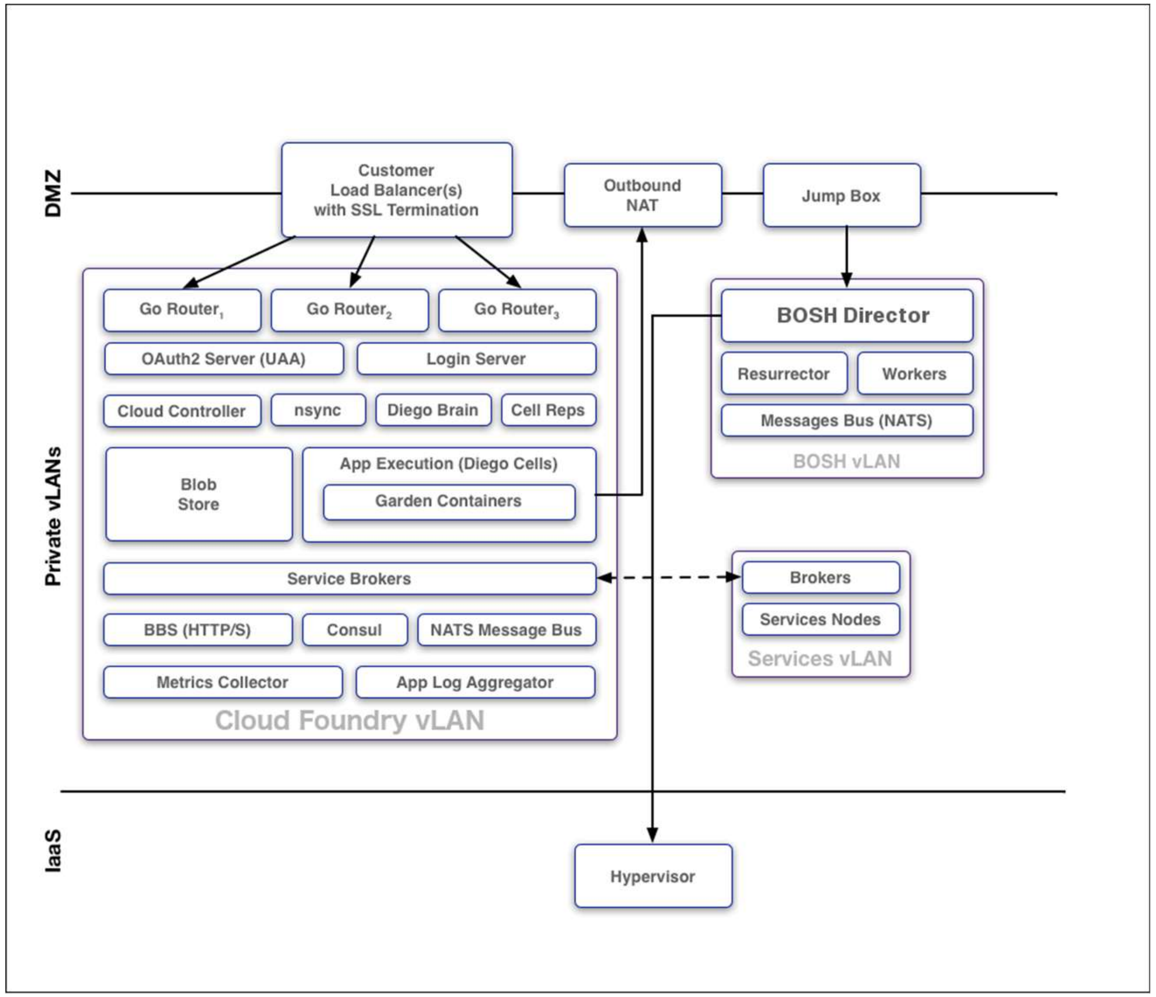
- Managed – Managed by a service provider:
  - Public Cloud: AKS (Azure), EKS (Amazon) or GKE (Google)
  - Private Cloud: VMware Tanzu or Red Hat OpenShift
- Unmanaged – Managed by your own:
  - Kubeadm
  - Minikube


Amazon EKS
Google Kubernetes Engine
Which Managed Kubernetes is Right For Me?
OPENSHIFT
VMware Tanzu
Azure Kubernetes Service (AKS)

Cloud Foundry Architecture

What is Cloud Foundry and what does it actually?

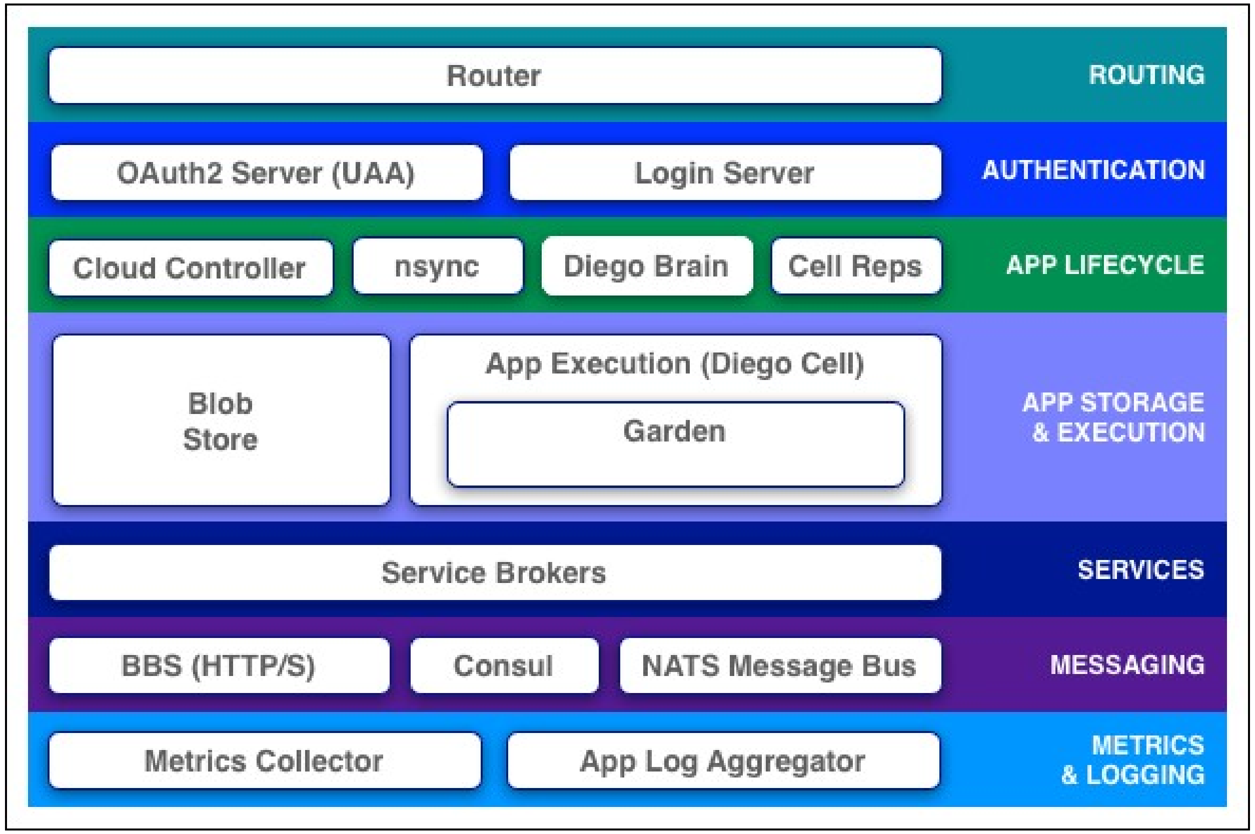
- Open-source Platform-as-a-Service (PaaS)
- No management of VMs and containers directly
- Easy deployment of applications
- Abstracts infrastructure complexity
- Supports multi-cloud deployments



Cloud Foundry Architecture (2)

Cloud Foundry Components:

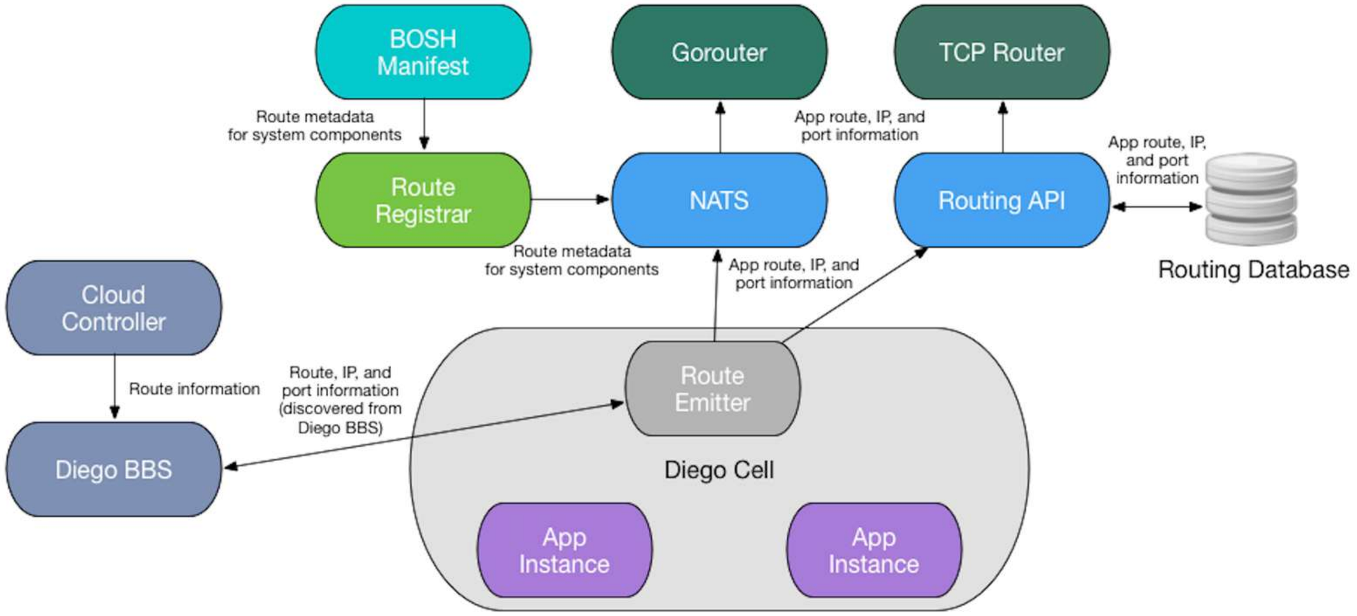
- Routing
- Authentication
- App Lifecycle
- App Storage & Execution
- Services
- Messaging
- Metrics & Logging



Cloud Foundry Architecture (3)

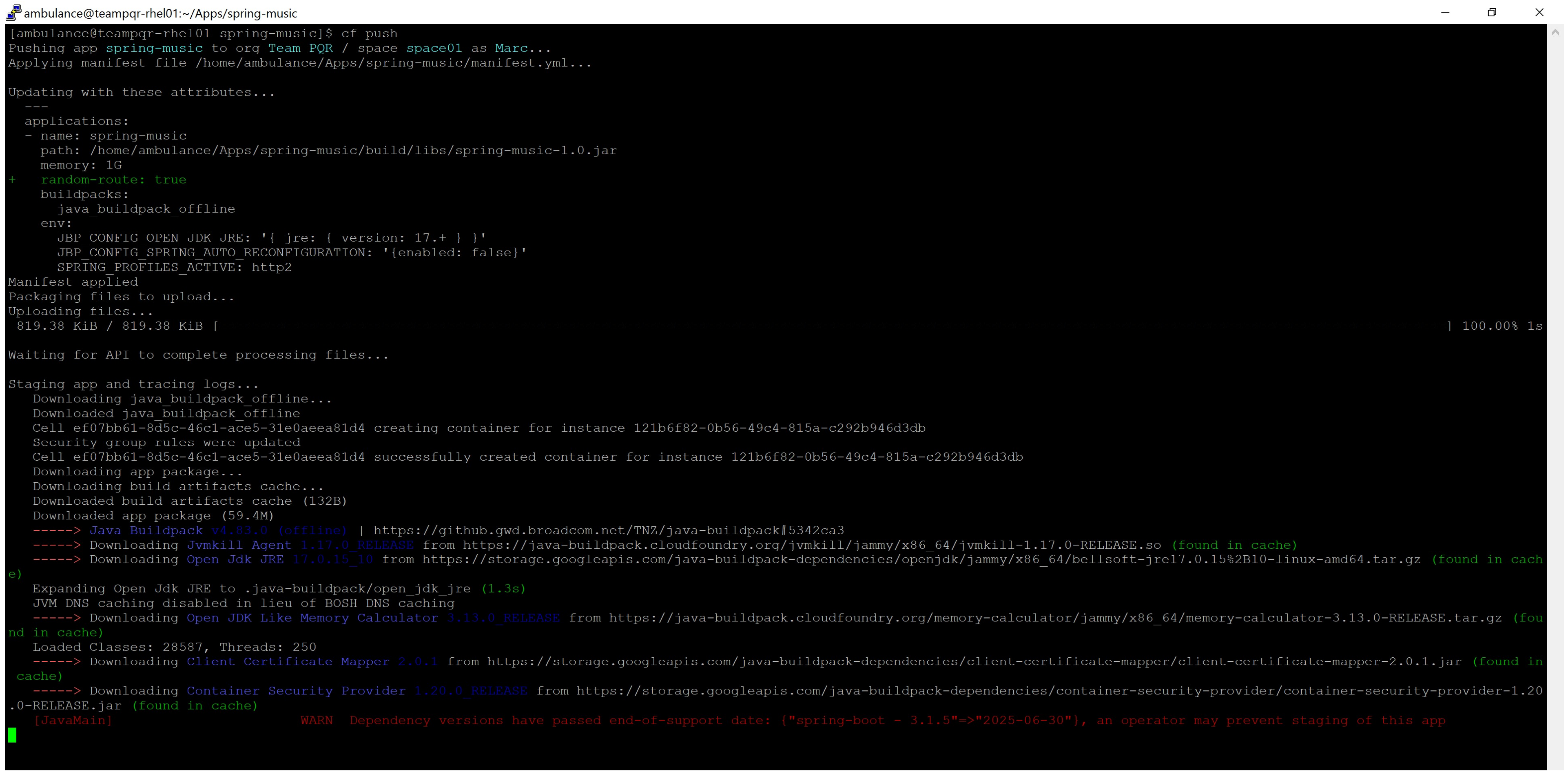
Cloud Foundry Routing Components:

- Cloud Controller
- Diego BBS
- Route Emitter
- Routing API
- Gorouter
- TCP Router
- Routing Database



BOSH Manifest
Gorouter
TCP Router
Route metadata for system components
App route, IP, and port information
App route, IP, and port information
Route Registrar
NATS
Routing API
Route metadata for system components
App route, IP, and port information
Routing Database
Cloud Controller
Route information
Route, IP, and port information (discovered from Diego BBS)
Route Emitter
Diego BBS
Diego Cell
App Instance
App Instance

Cloud Foundry Architecture (4)



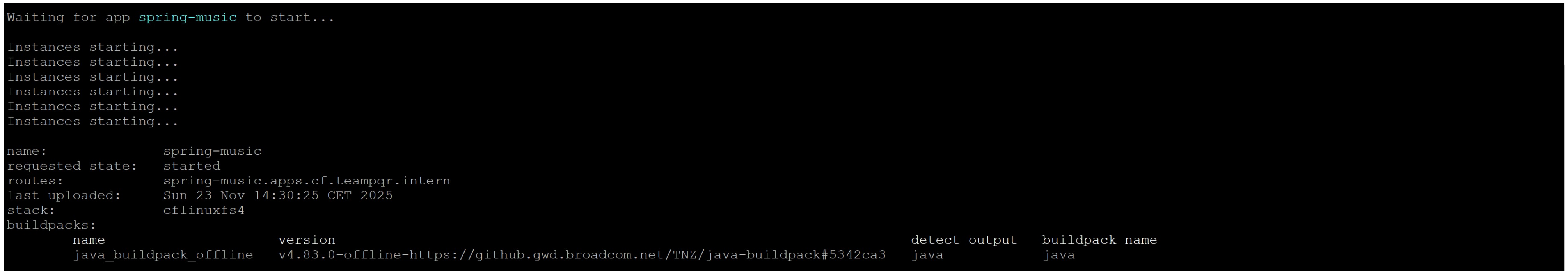
```
[ambulance@teampqr-rhel01 spring-music]$ cf push
Pushing app spring-music to org Team PQR / space space01 as Marc...
Applying manifest file /home/ambulance/Apps/spring-music/manifest.yml...

Updating with these attributes...
  ---
  applications:
  - name: spring-music
    path: /home/ambulance/Apps/spring-music/build/libs/spring-music-1.0.jar
    memory: 1G
+   random-route: true
    buildpacks:
      java_buildpack_offline
    env:
      JBP_CONFIG_OPEN_JDK_JRE: '{ jre: { version: 17.+ } }'
      JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{enabled: false}'
      SPRING_PROFILES_ACTIVE: http2
Manifest applied
Packaging files to upload...
Uploading files...
 819.38 KiB / 819.38 KiB [==============================================================================================================================================] 100.00% 1s

Waiting for API to complete processing files...

Staging app and tracing logs...
   Downloading java_buildpack_offline...
   Downloaded java_buildpack_offline
   Cell ef07bb61-8d5c-46c1-ace5-31e0aeea81d4 creating container for instance 121b6f82-0b56-49c4-815a-c292b946d3db
   Security group rules were updated
   Cell ef07bb61-8d5c-46c1-ace5-31e0aeea81d4 successfully created container for instance 121b6f82-0b56-49c4-815a-c292b946d3db
   Downloading app package...
   Downloading build artifacts cache...
   Downloaded build artifacts cache (132B)
   Downloaded app package (59.4M)
   -----> Java Buildpack v4.83.0 (offline) | https://github.gwd.broadcom.net/TNZ/java-buildpack#5342ca3
   -----> Downloading Jvmkill Agent 1.17.0_RELEASE from https://java-buildpack.cloudfoundry.org/jvmkill/jammy/x86_64/jvmkill-1.17.0-RELEASE.so (found in cache)
   -----> Downloading Open Jdk JRE 17.0.15_10 from https://storage.googleapis.com/java-buildpack-dependencies/openjdk/jammy/x86_64/bellsoft-jre17.0.15%2B10-linux-amd64.tar.gz (found in cach
e)
   Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.3s)
   JVM DNS caching disabled in lieu of BOSH DNS caching
   -----> Downloading Open JDK Like Memory Calculator 3.13.0_RELEASE from https://java-buildpack.cloudfoundry.org/memory-calculator/jammy/x86_64/memory-calculator-3.13.0-RELEASE.tar.gz (fou
nd in cache)
   Loaded Classes: 28587, Threads: 250
   -----> Downloading Client Certificate Mapper 2.0.1 from https://storage.googleapis.com/java-buildpack-dependencies/client-certificate-mapper/client-certificate-mapper-2.0.1.jar (found in
 cache)
   -----> Downloading Container Security Provider 1.20.0_RELEASE from https://storage.googleapis.com/java-buildpack-dependencies/container-security-provider/container-security-provider-1.20
.0-RELEASE.jar (found in cache)
   [JavaMain]                      WARN  Dependency versions have passed end-of-support date: {"spring-boot - 3.1.5"=>"2025-06-30"}, an operator may prevent staging of this app
```

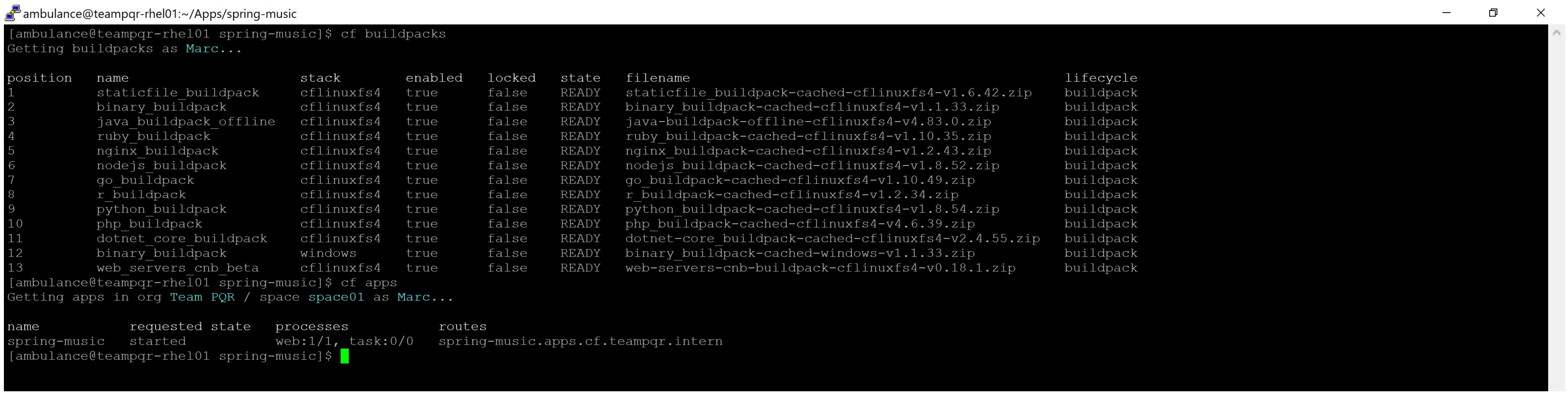
Cloud Foundry Architecture (5)



```
Waiting for app spring-music to start...

Instances starting...
Instances starting...
Instances starting...
Instances starting...
Instances starting...
Instances starting...

name:                spring-music
requested state:     started
routes:              spring-music.apps.cf.teampqr.intern
last uploaded:       Sun 23 Nov 14:30:25 CET 2025
stack:               cflinuxfs4
buildpacks:
        name                    version                                                                  detect output    buildpack name
        java_buildpack_offline   v4.83.0-offline-https://github.gwd.broadcom.net/TNZ/java-buildpack#5342ca3   java             java
```



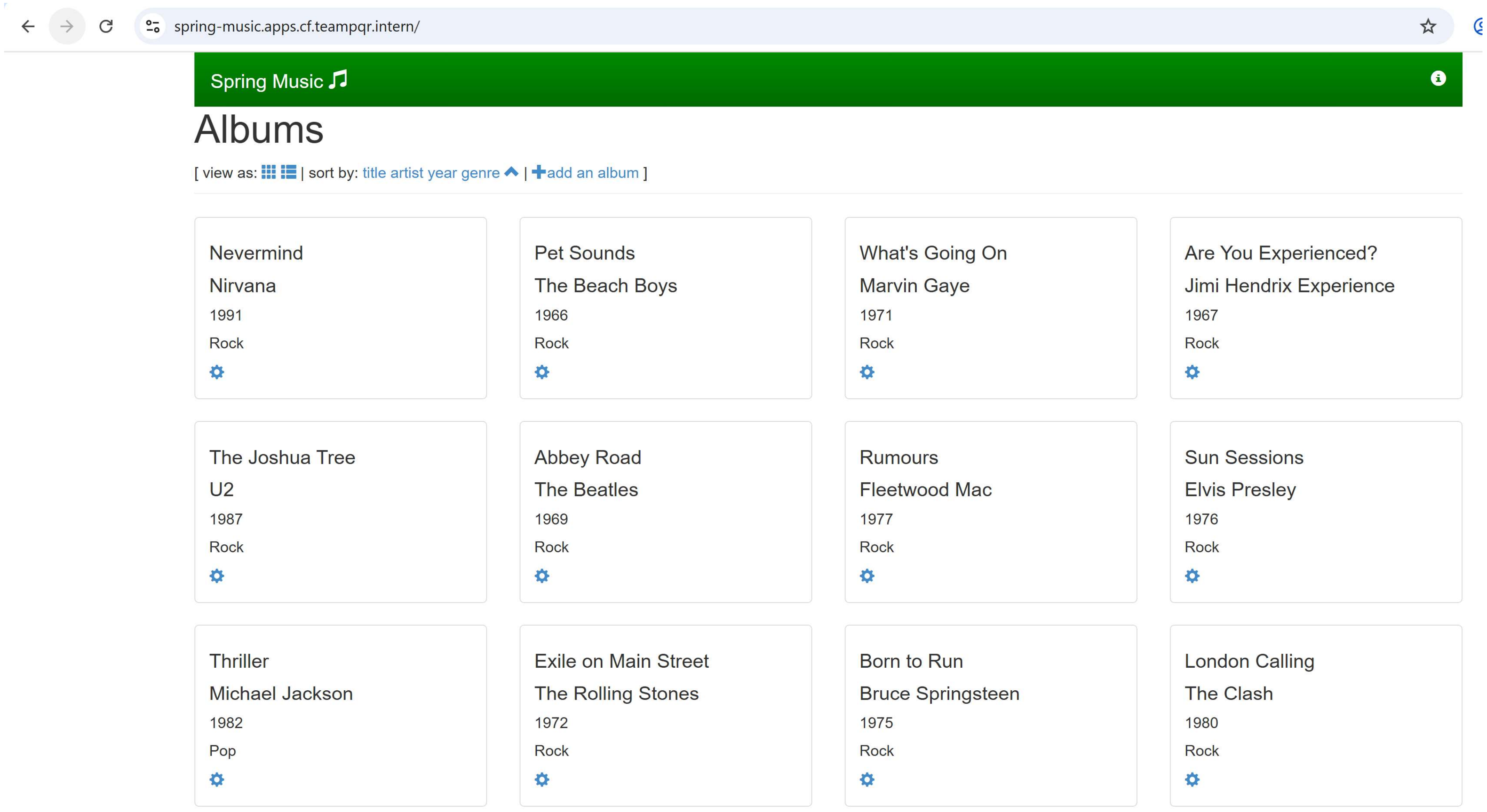
ambulance@teampqr-rhel01:~/Apps/spring-music

```
[ambulance@teampqr-rhel01 spring-music]$ cf buildpacks
Getting buildpacks as Marc...

position   name                     stack        enabled   locked   state   filename                                                 lifecycle
1          staticfile_buildpack     cflinuxfs4   true      false    READY   staticfile_buildpack-cached-cflinuxfs4-v1.6.42.zip       buildpack
2          binary_buildpack         cflinuxfs4   true      false    READY   binary_buildpack-cached-cflinuxfs4-v1.1.33.zip           buildpack
3          java_buildpack_offline   cflinuxfs4   true      false    READY   java-buildpack-offline-cflinuxfs4-v4.83.0.zip            buildpack
4          ruby_buildpack           cflinuxfs4   true      false    READY   ruby_buildpack-cached-cflinuxfs4-v1.10.35.zip            buildpack
5          nginx_buildpack          cflinuxfs4   true      false    READY   nginx_buildpack-cached-cflinuxfs4-v1.2.43.zip            buildpack
6          nodejs_buildpack         cflinuxfs4   true      false    READY   nodejs_buildpack-cached-cflinuxfs4-v1.8.52.zip           buildpack
7          go_buildpack             cflinuxfs4   true      false    READY   go_buildpack-cached-cflinuxfs4-v1.10.49.zip              buildpack
8          r_buildpack              cflinuxfs4   true      false    READY   r_buildpack-cached-cflinuxfs4-v1.2.34.zip                buildpack
9          python_buildpack         cflinuxfs4   true      false    READY   python_buildpack-cached-cflinuxfs4-v1.8.54.zip           buildpack
10         php_buildpack            cflinuxfs4   true      false    READY   php_buildpack-cached-cflinuxfs4-v4.6.39.zip              buildpack
11         dotnet_core_buildpack    cflinuxfs4   true      false    READY   dotnet-core_buildpack-cached-cflinuxfs4-v2.4.55.zip      buildpack
12         binary_buildpack         windows      true      false    READY   binary_buildpack-cached-windows-v1.1.33.zip              buildpack
13         web_servers_cnb_beta     cflinuxfs4   true      false    READY   web-servers-cnb-buildpack-cflinuxfs4-v0.18.1.zip         buildpack
[ambulance@teampqr-rhel01 spring-music]$ cf apps
Getting apps in org Team PQR / space space01 as Marc...

name           requested state   processes           routes
spring-music   started           web:1/1, task:0/0   spring-music.apps.cf.teampqr.intern
[ambulance@teampqr-rhel01 spring-music]$
```

Cloud Foundry Architecture (6)

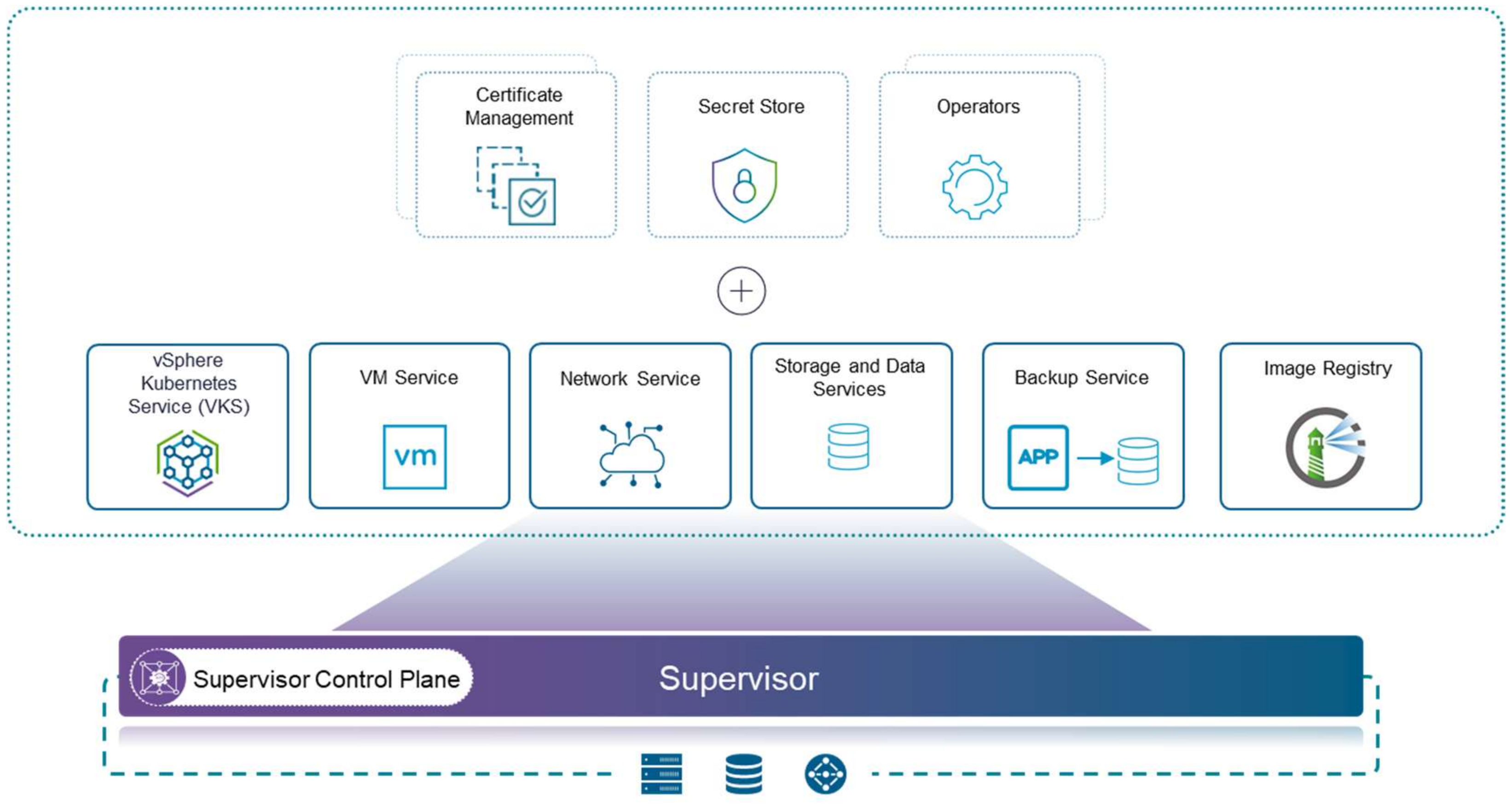

spring-music.apps.cf.teampqr.intern/

Albums

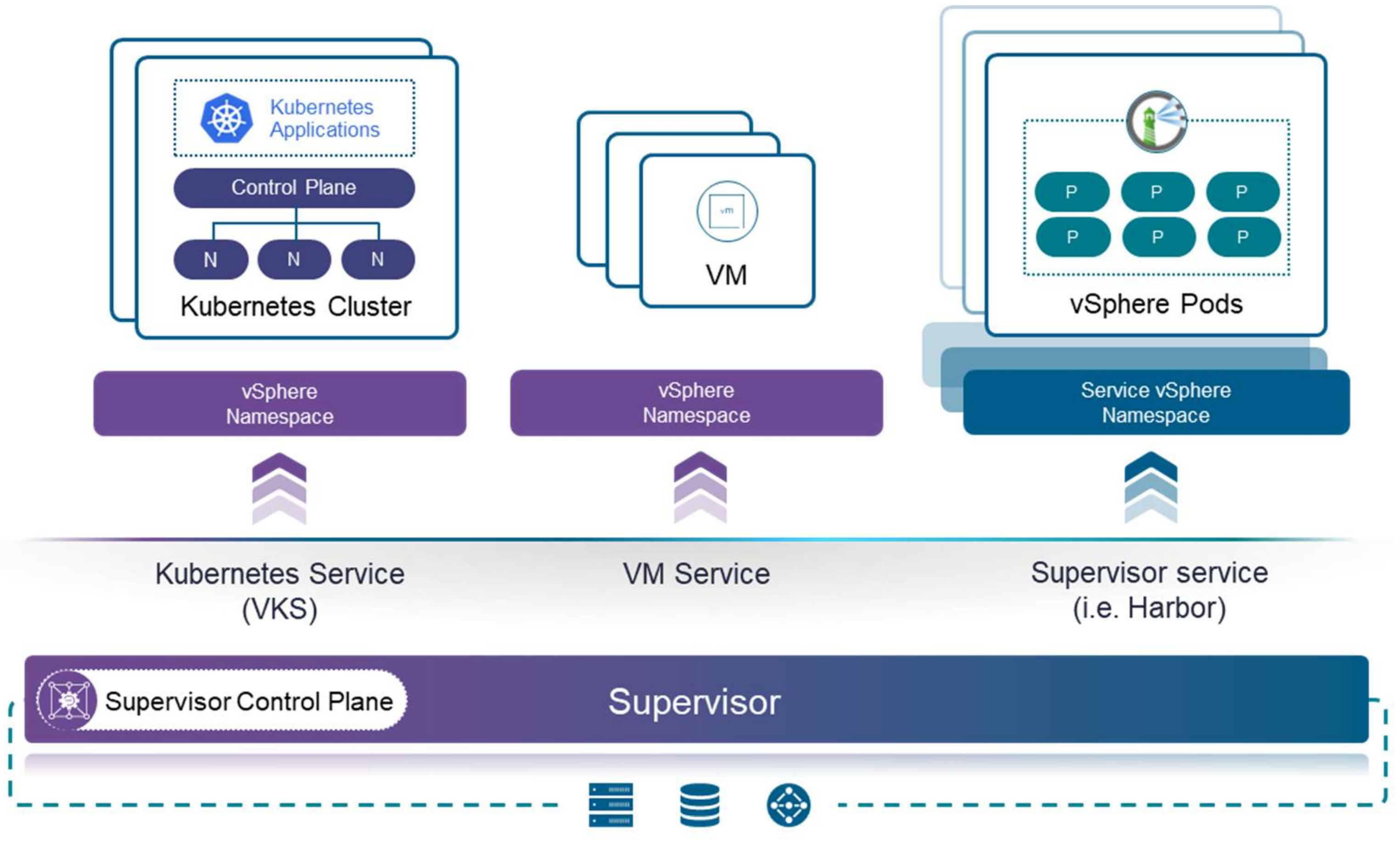
[view as: | sort by: title artist year genre | add an album]

Nevermind Nirvana 1991 Rock	Pet Sounds The Beach Boys 1966 Rock	What's Going On Marvin Gaye 1971 Rock	Are You Experienced? Jimi Hendrix Experience 1967 Rock
The Joshua Tree U2 1987 Rock	Abbey Road The Beatles 1969 Rock	Rumours Fleetwood Mac 1977 Rock	Sun Sessions Elvis Presley 1976 Rock
Thriller Michael Jackson 1982 Pop	Exile on Main Street The Rolling Stones 1972 Rock	Born to Run Bruce Springsteen 1975 Rock	London Calling The Clash 1980 Rock

vSphere with Kubernetes Service

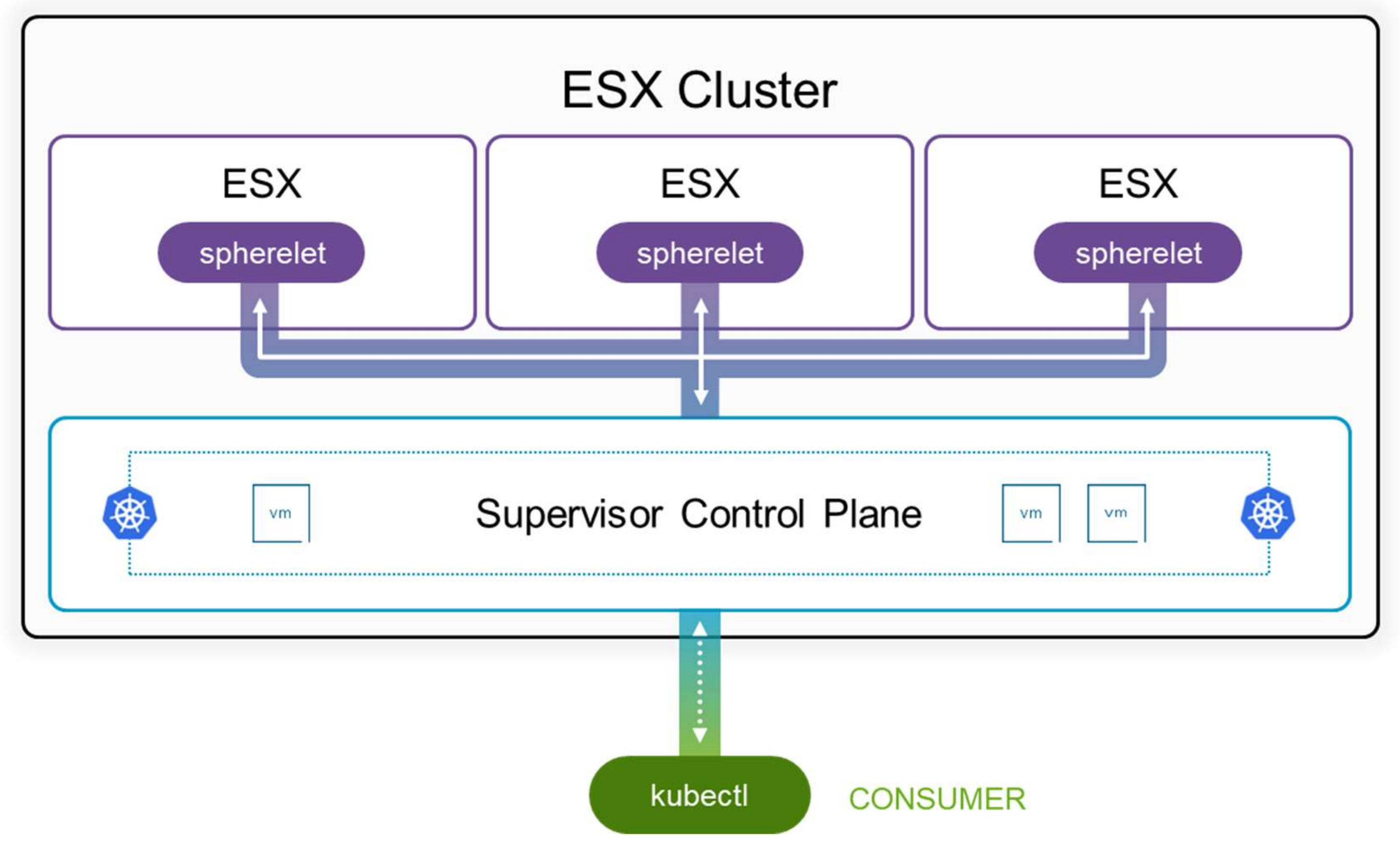


vSphere with Kubernetes Service (2)

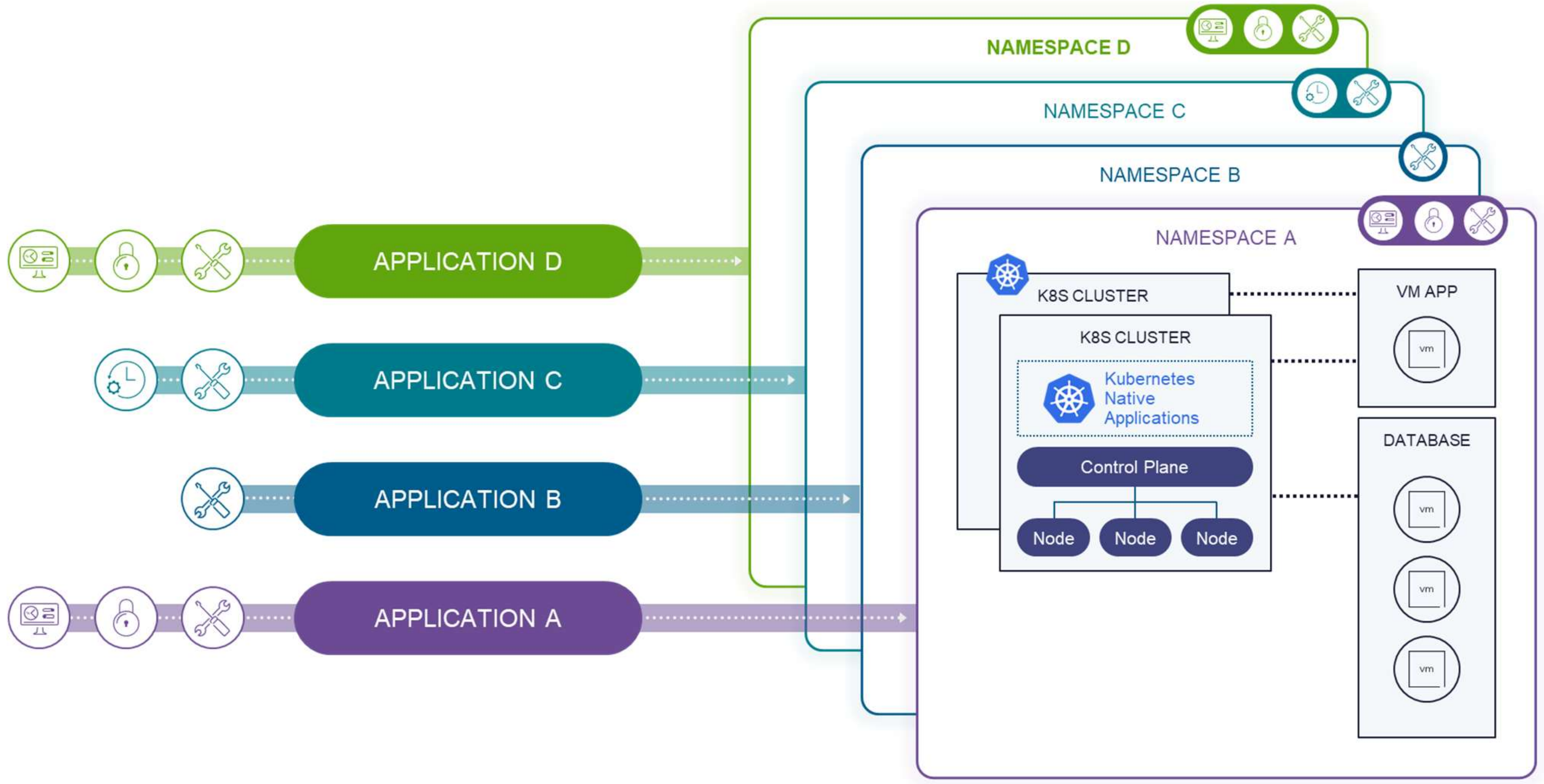


VCF TechCon Powered by VMUG

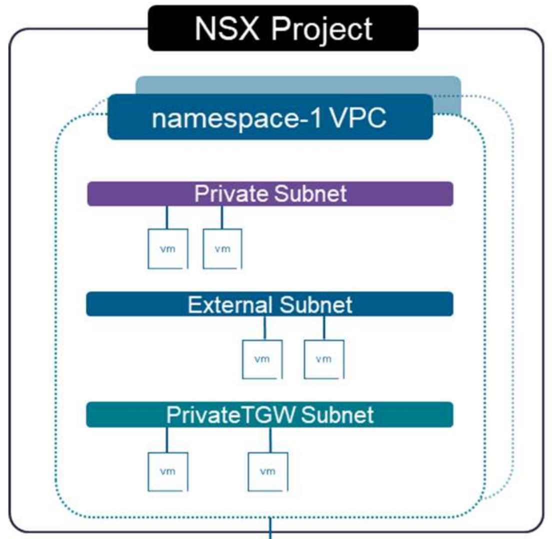
vSphere with Kubernetes Service (3)

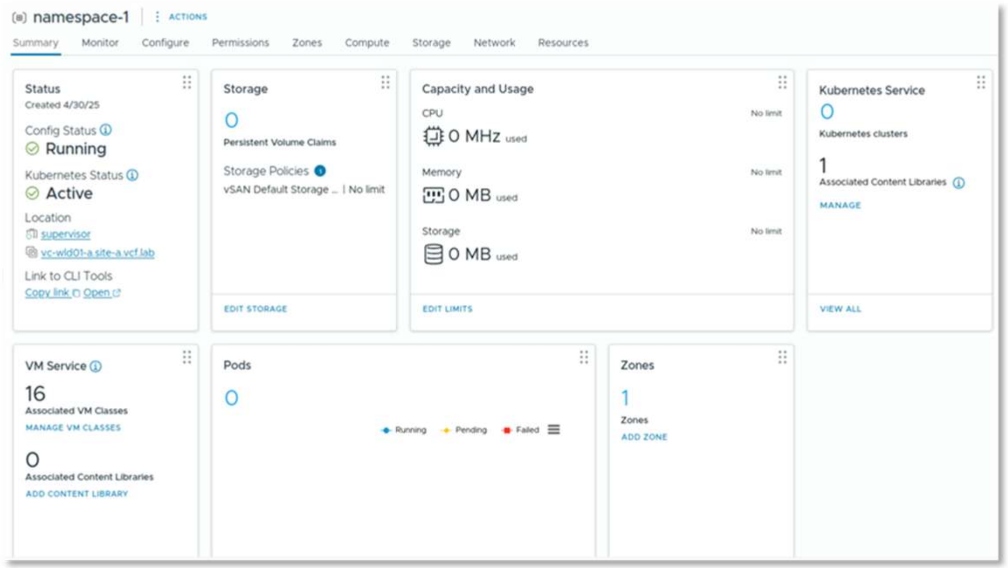


vSphere with Kubernetes Service (4)



vSphere with Kubernetes Service (5)




namespace-1 | ACTIONS

Summary Monitor Configure Permissions Zones Compute Storage Network Resources

Status
Created 4/30/25
Config Status
Running
Kubernetes Status
Active
Location
supervisor
vc-wld01-a.site-a.vcf.lab
Link to CLI Tools
Copy link Open

Storage
0
Persistent Volume Claims
Storage Policies
vSAN Default Storage ... | No limit
EDIT STORAGE

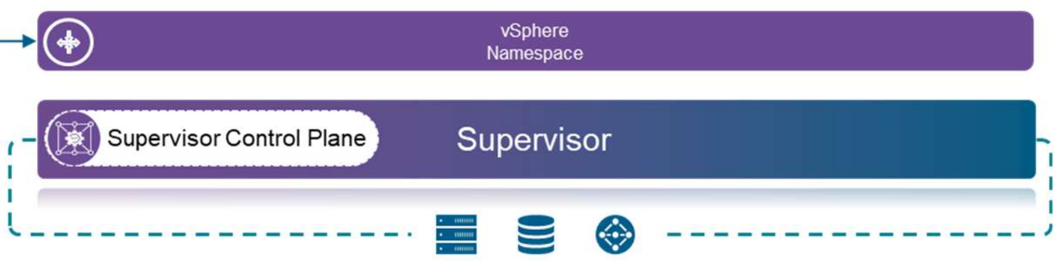
Capacity and Usage
CPU No limit
0 MHz used
Memory No limit
0 MB used
Storage No limit
0 MB used
EDIT LIMITS

Kubernetes Service
0
Kubernetes clusters
1
Associated Content Libraries
MANAGE
VIEW ALL

VM Service
16
Associated VM Classes
MANAGE VM CLASSES
0
Associated Content Libraries
ADD CONTENT LIBRARY

Pods
0
Running Pending Failed

Zones
1
Zones
ADD ZONE



vSphere with Kubernetes Service (6)

HA-Proxy	NSX-ALB	NSX
Open Source	Formerly AVI Vantage	Full virtual networking stack
Deployed as an OVA	Fully supported by VMware	Fully supported by VMware
Simple configuration	Deployed as an OVA	Complex deployment – requires good networking knowledge
Lightweight on resources	Advanced configuration	Significant resource requirements
Product issues directed to HA-Proxy support	Larger resource requirement	Production ready
Good for home-labs and PoCs	Production ready	

VCF TechCon Powered by VMUG

vSphere with Kubernetes Service (7)

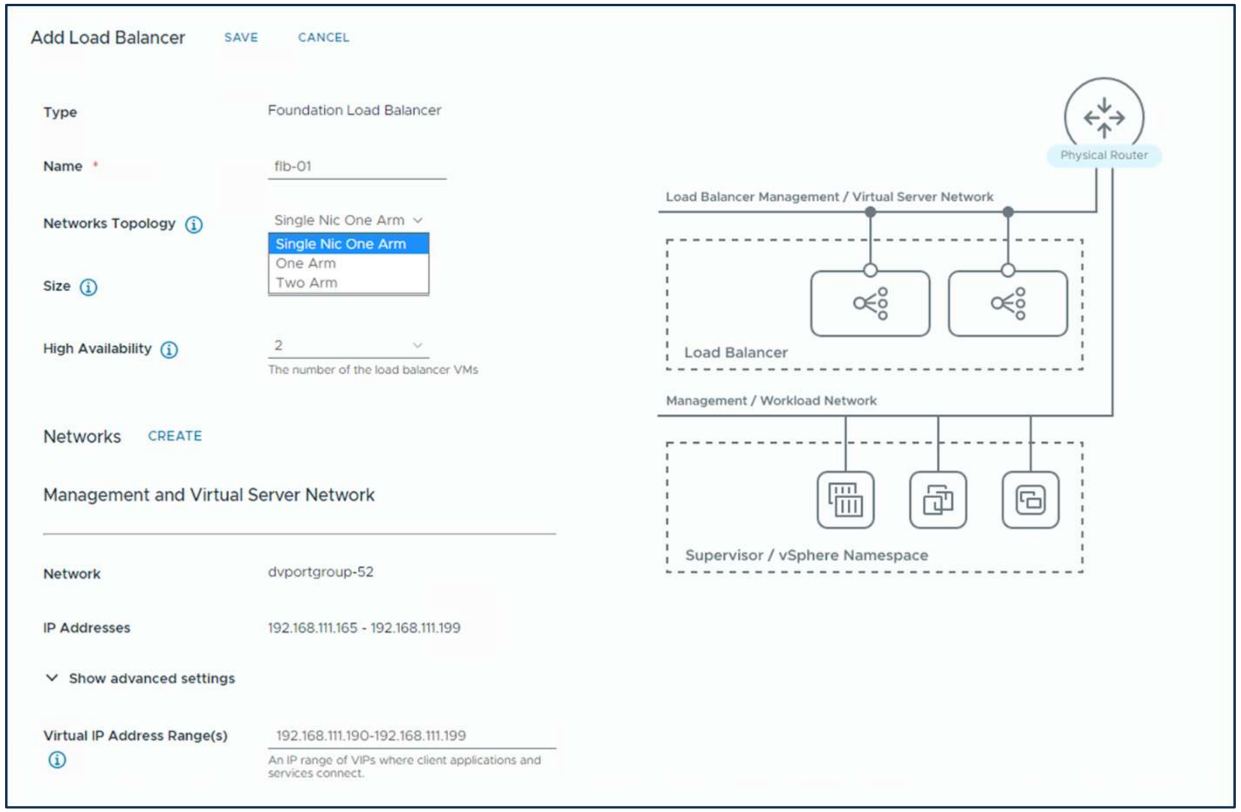
Integrated **Layer 4** Load Balancing for vSphere Supervisor

Supported on **vSphere Networking Stack (VDS)**

3 Networks Topology Options that can evolve

High-availability Option

Deployed by vCenter with minimal user input



Add Load Balancer SAVE CANCEL

Type	Foundation Load Balancer
Name *	flb-01
Networks Topology ⓘ	Single Nic One Arm
Size ⓘ	
High Availability ⓘ	2
	The number of the load balancer VMs

Networks CREATE

Management and Virtual Server Network

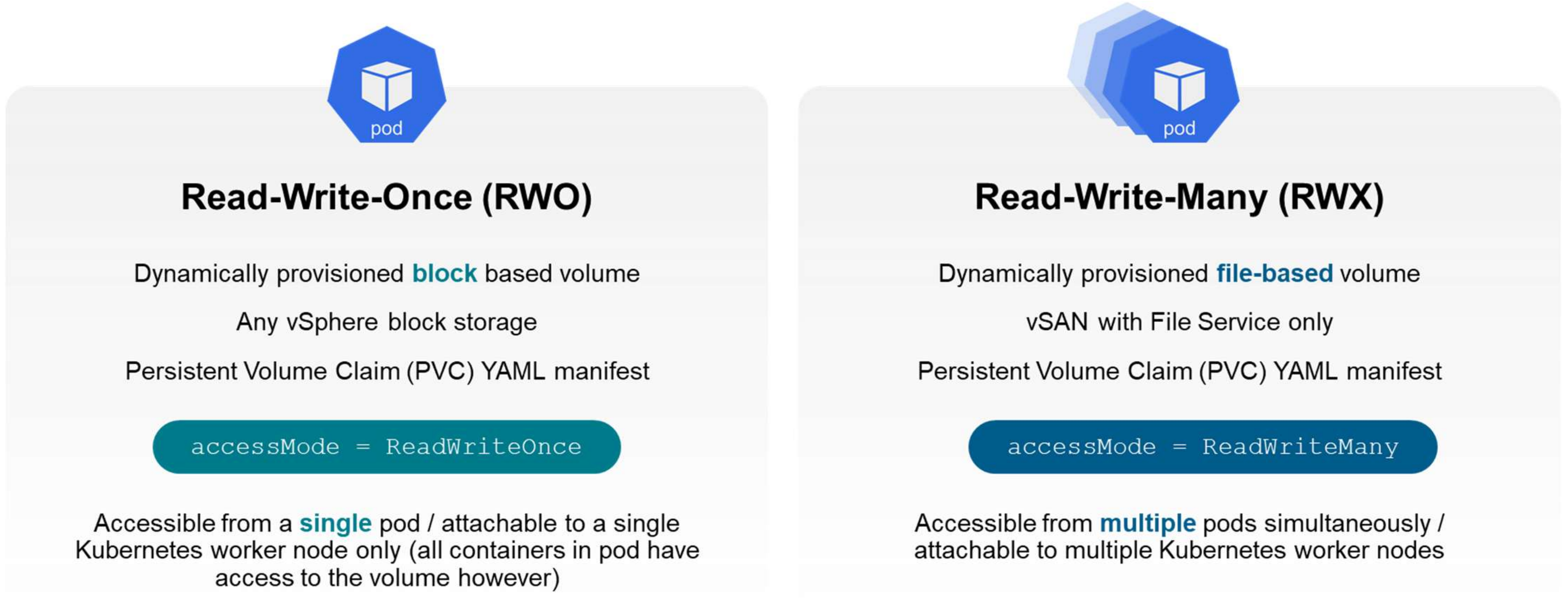
Network	dvportgroup-52
IP Addresses	192.168.111.165 - 192.168.111.199

Show advanced settings

Virtual IP Address Range(s) ⓘ	192.168.111.190-192.168.111.199
	An IP range of VIPs where client applications and services connect.

Single Nic One Arm
One Arm
Two Arm
Physical Router
Load Balancer Management / Virtual Server Network
Load Balancer
Management / Workload Network
Supervisor / vSphere Namespace

vSphere with Kubernetes Service (8)



Read-Write-Once (RWO)

Dynamically provisioned **block** based volume

Any vSphere block storage

Persistent Volume Claim (PVC) YAML manifest

```
accessMode = ReadWriteOnce
```

Accessible from a **single** pod / attachable to a single Kubernetes worker node only (all containers in pod have access to the volume however)

Read-Write-Many (RWX)

Dynamically provisioned **file-based** volume

vSAN with File Service only

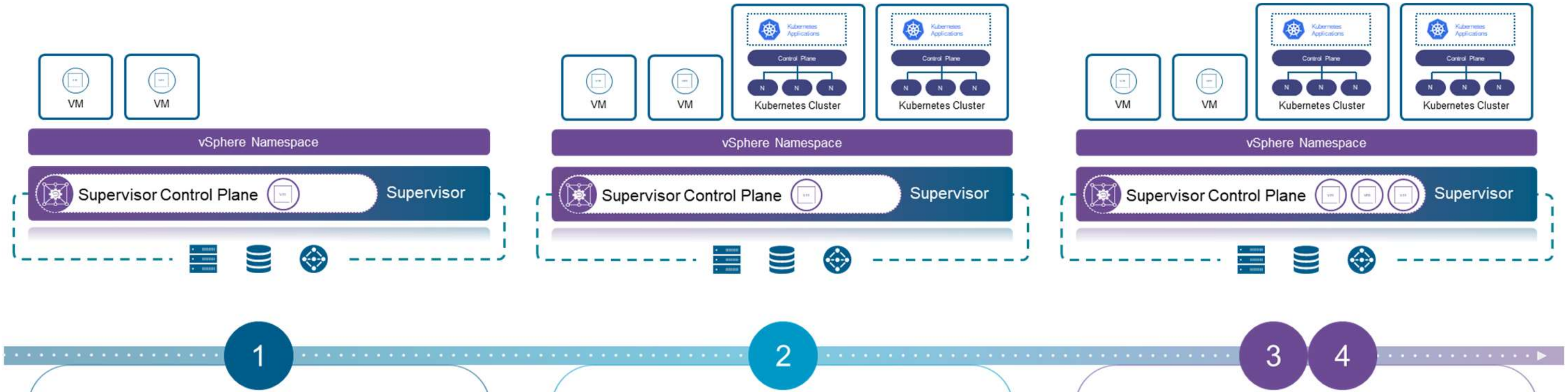
Persistent Volume Claim (PVC) YAML manifest

```
accessMode = ReadWriteMany
```

Accessible from **multiple** pods simultaneously / attachable to multiple Kubernetes worker nodes

VCF TechCon Powered by VMUG

vSphere with Kubernetes Service (9)



1 Enable Easy Supervisor

- a single Control Plane VM
- a single Network
- no Load Balancer

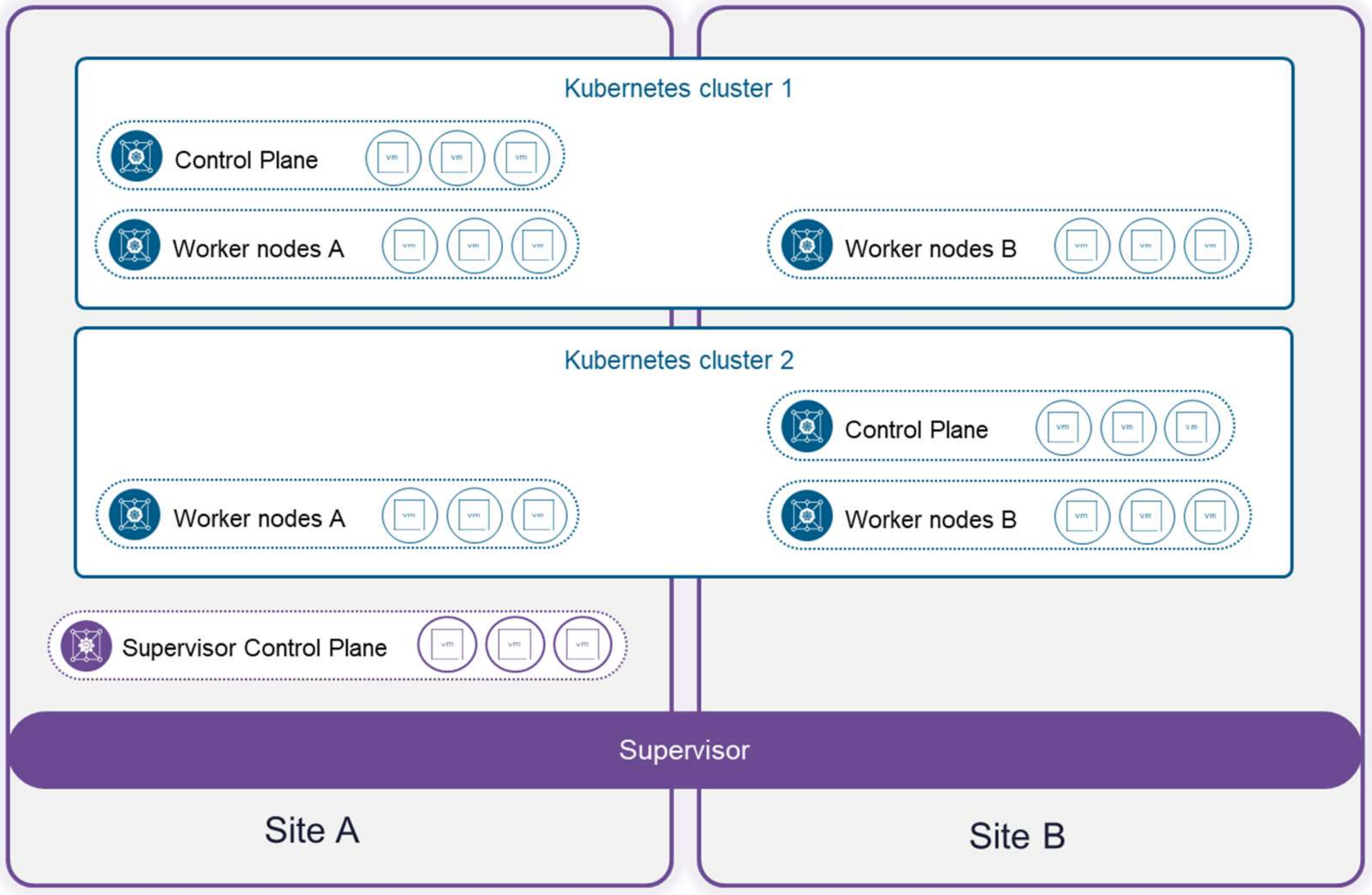
2 Add a Load Balancer

3 4 Scale out Control Plane VMs to provide High Availability

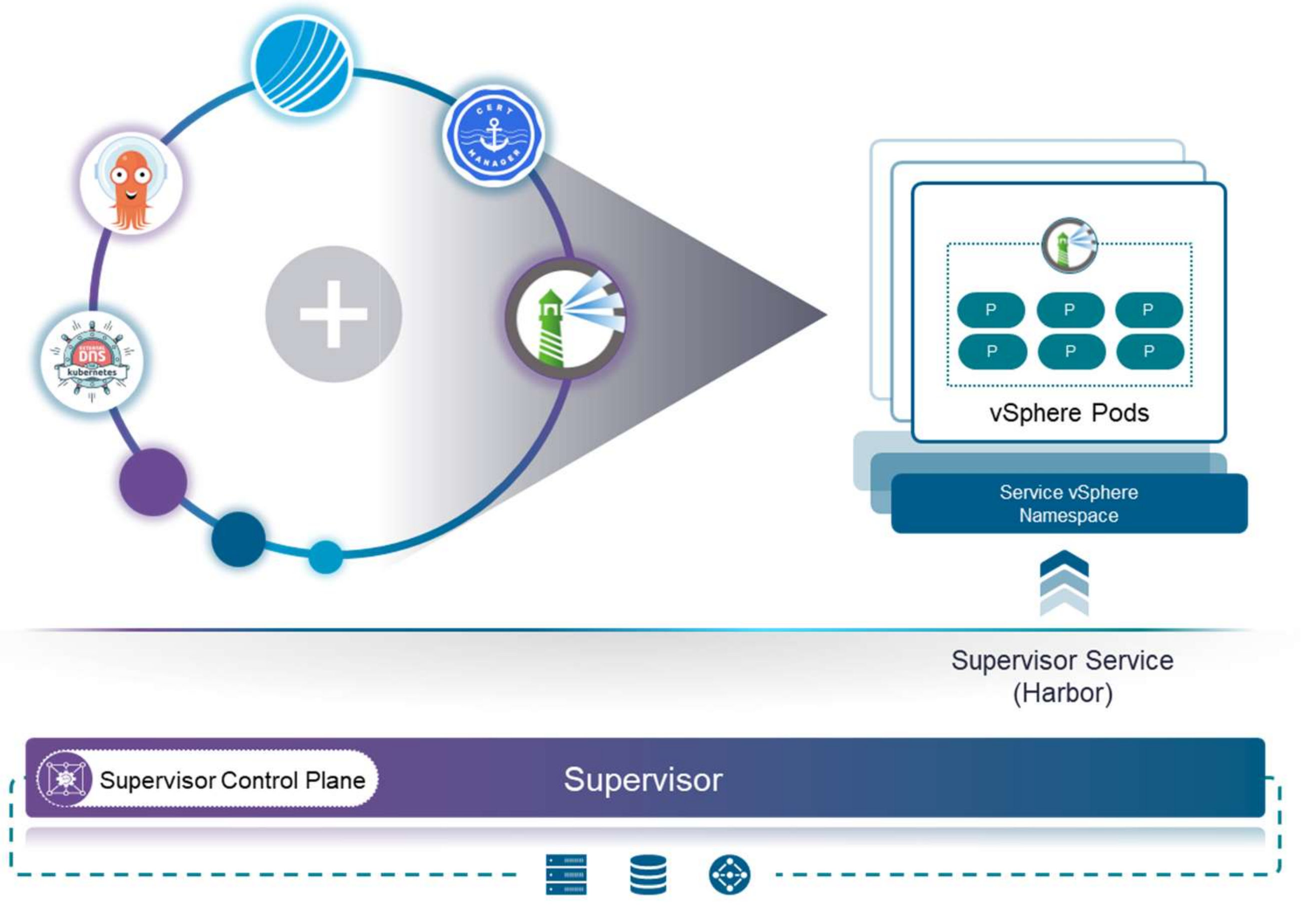
And / Or

Add a Second Network

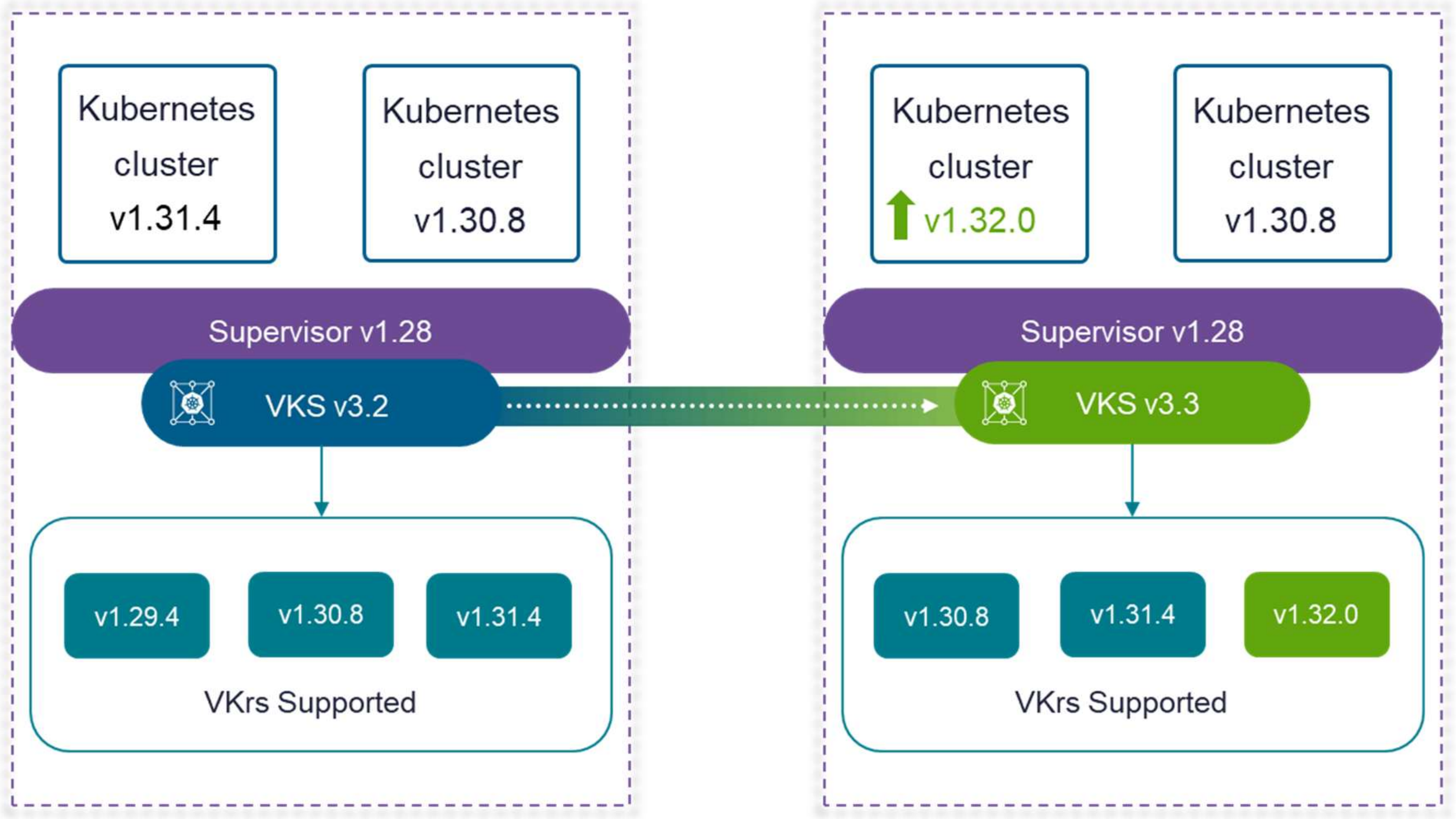
vSphere with Kubernetes Service (10)



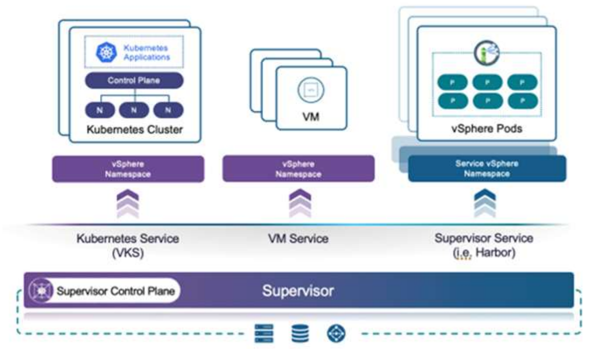
vSphere with Kubernetes Service (11)



vSphere with Kubernetes Service (12)

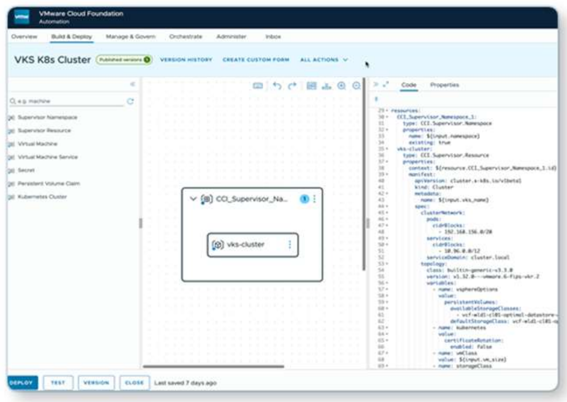


vSphere with Kubernetes Service (13)



Unified Lifecycle Management

Desired state definition, deployment and deletion



VKS K8s Cluster

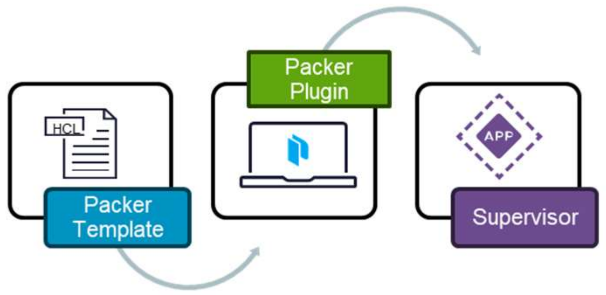
```
resources:
  CCI_Supervisor_Namespace_1:
    type: CCI.Supervisor.Namespace
    properties:
      name: ${input.namespace}
      existing: true
  vks-cluster:
    type: CCI.Supervisor.Resource
    properties:
      context: ${resource.CCI_Supervisor_Namespace_1.id}
      manifest:
        apiVersion: cluster.x-k8s.io/v1beta1
        kind: Cluster
        metadata:
          name: ${input.vks_name}
        spec:
          clusterNetwork:
            pods:
              cidrBlocks:
                - 192.168.156.0/20
            services:
              cidrBlocks:
                - 10.96.0.0/12
            serviceDomain: cluster.local
          topology:
            class: builtin-generic-v3.3.0
            version: v1.32.0---vmware.6-fips-vkr.2
            variables:
              - name: vsphereOptions
                value:
                  persistentVolumes:
                    availableStorageClasses:
                      - vcf-wld1-cl01-optimal-datastore-
                    defaultStorageClass: vcf-wld1-cl01-o
              - name: kubernetes
                value:
                  certificateRotation:
                    enabled: false
              - name: vmClass
                value: ${input.vm_size}
              - name: storageClass
```

Unified Automation and Config Management

Same tools targeting both VMs and Kubernetes

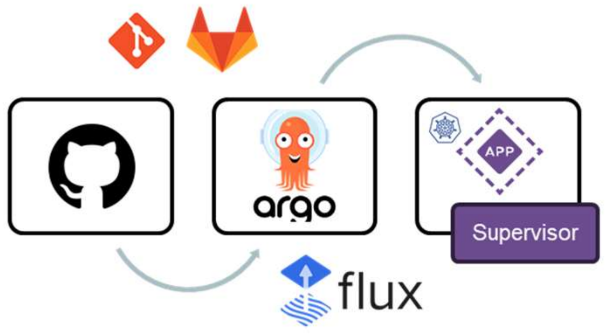
Integrate with Build Tools

Build, customize, and publish VM images and templates

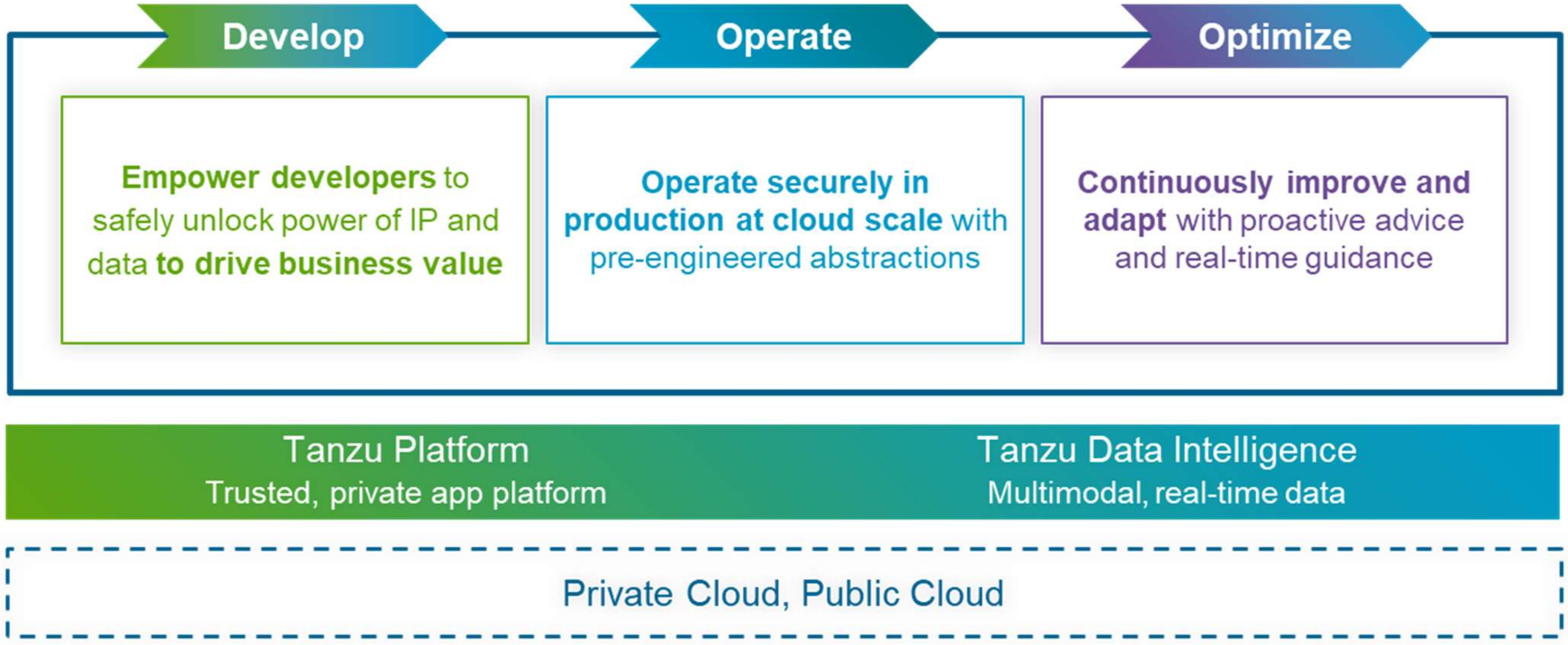


Unified GitOps

Continuous delivery targeting both VMs and Kubernetes

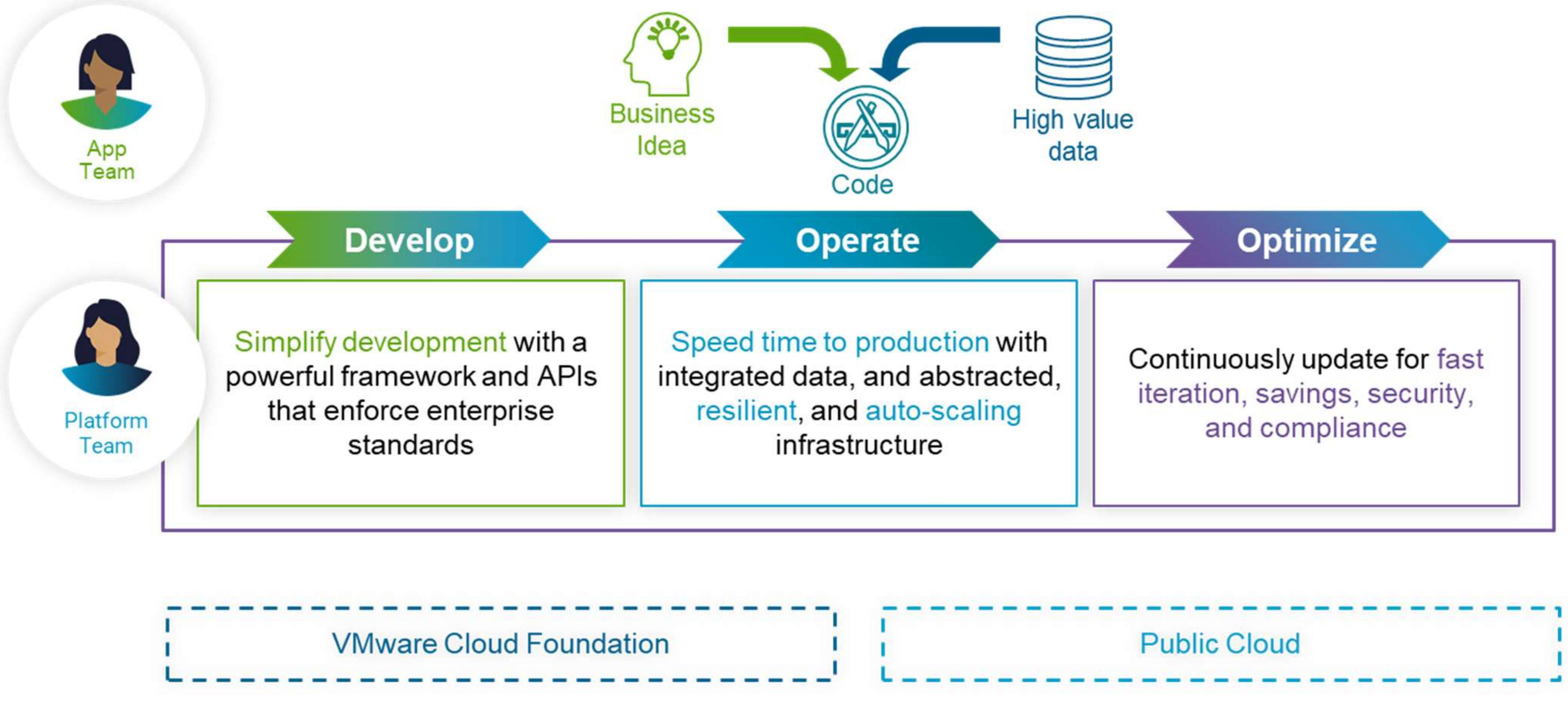


Tanzu Platform for Cloud Foundry

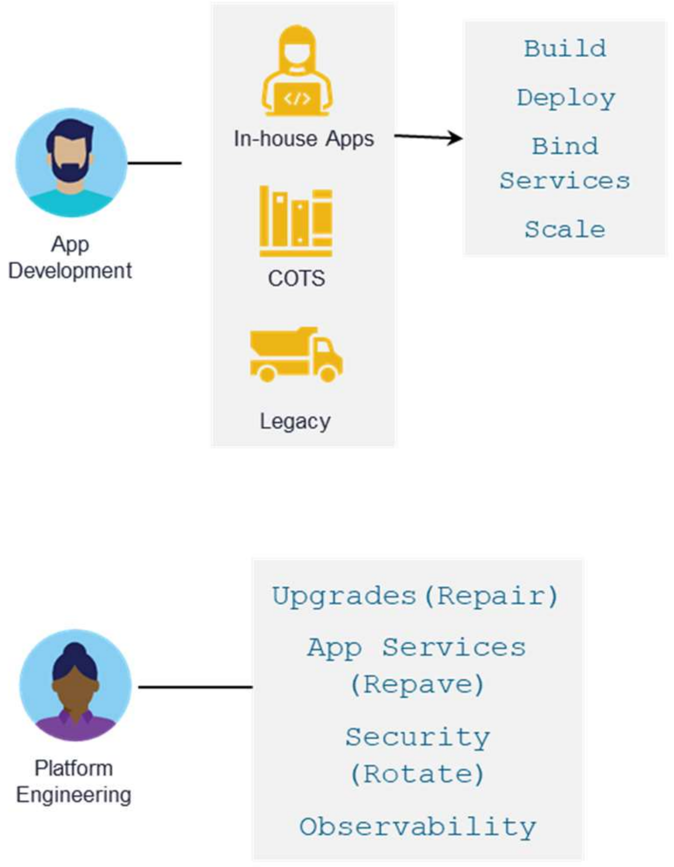


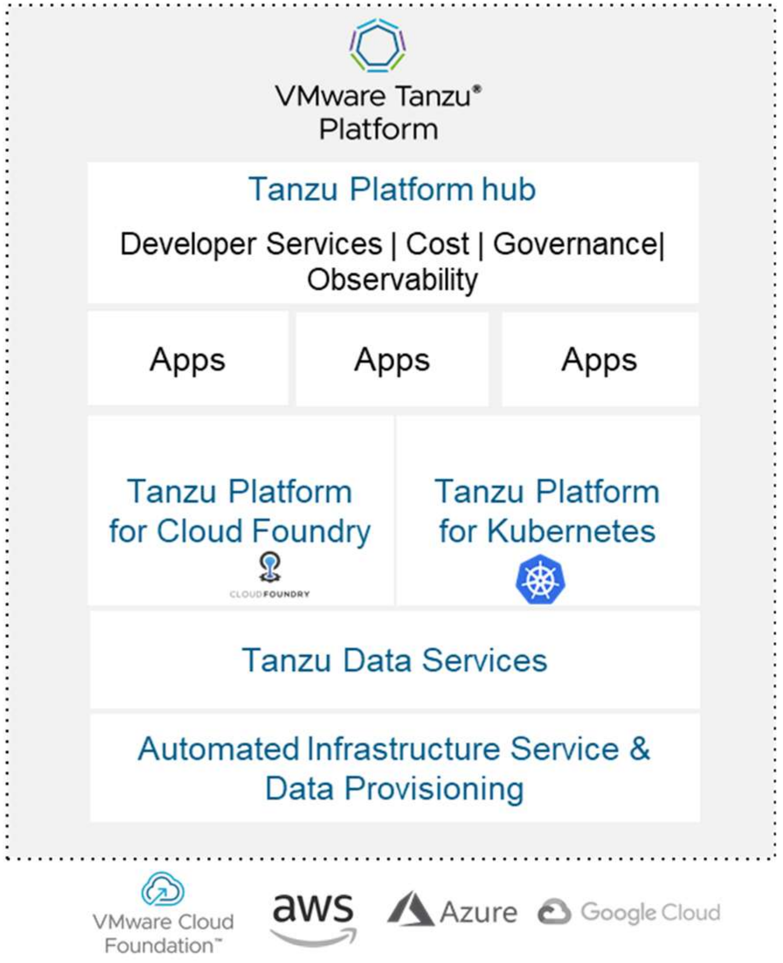
Develop
Operate
Optimize
Empower developers to safely unlock power of IP and data to drive business value
Operate securely in production at cloud scale with pre-engineered abstractions
Continuously improve and adapt with proactive advice and real-time guidance
Tanzu Platform
Trusted, private app platform
Tanzu Data Intelligence
Multimodal, real-time data
Private Cloud, Public Cloud

Tanzu Platform for Cloud Foundry (2)


App Team
Platform Team
Business Idea
Code
High value data
Develop
Operate
Optimize
Simplify development with a powerful framework and APIs that enforce enterprise standards
Speed time to production with integrated data, and abstracted, resilient, and auto-scaling infrastructure
Continuously update for fast iteration, savings, security, and compliance
VMware Cloud Foundation
Public Cloud

Tanzu Platform for Cloud Foundry (3)





VMware Tanzu® Platform
Tanzu Platform hub
Developer Services | Cost | Governance| Observability
Apps
Apps
Apps
Tanzu Platform for Cloud Foundry
CLOUD FOUNDRY
Tanzu Platform for Kubernetes
Tanzu Data Services
Automated Infrastructure Service & Data Provisioning
VMware Cloud Foundation™
aws
Azure
Google Cloud

Tanzu Platform for Cloud Foundry (4)

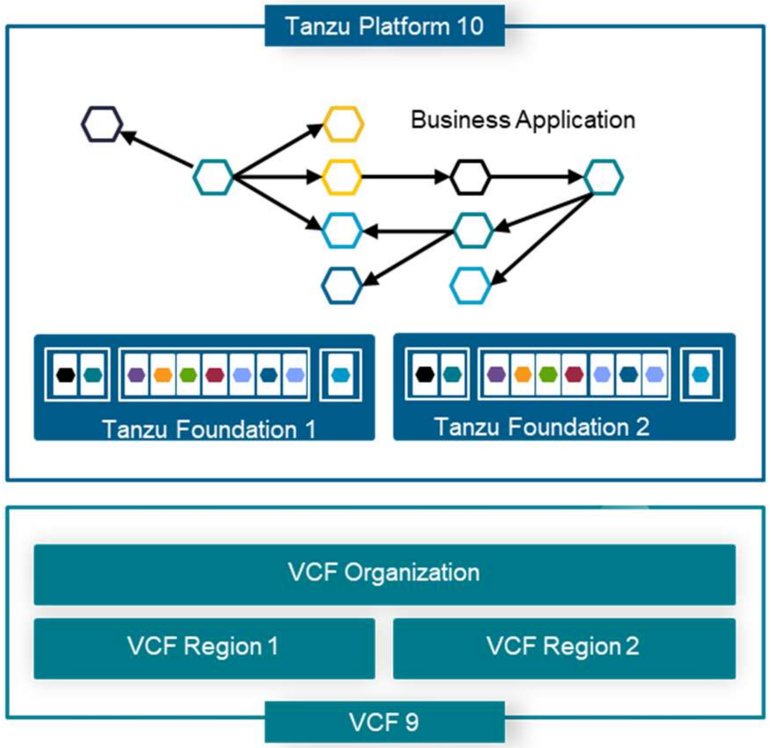
Serverless
Focus on code instead of infrastructure, platform manages lifecycle

High Availability & Resiliency
Resilient apps across compute instances and zones

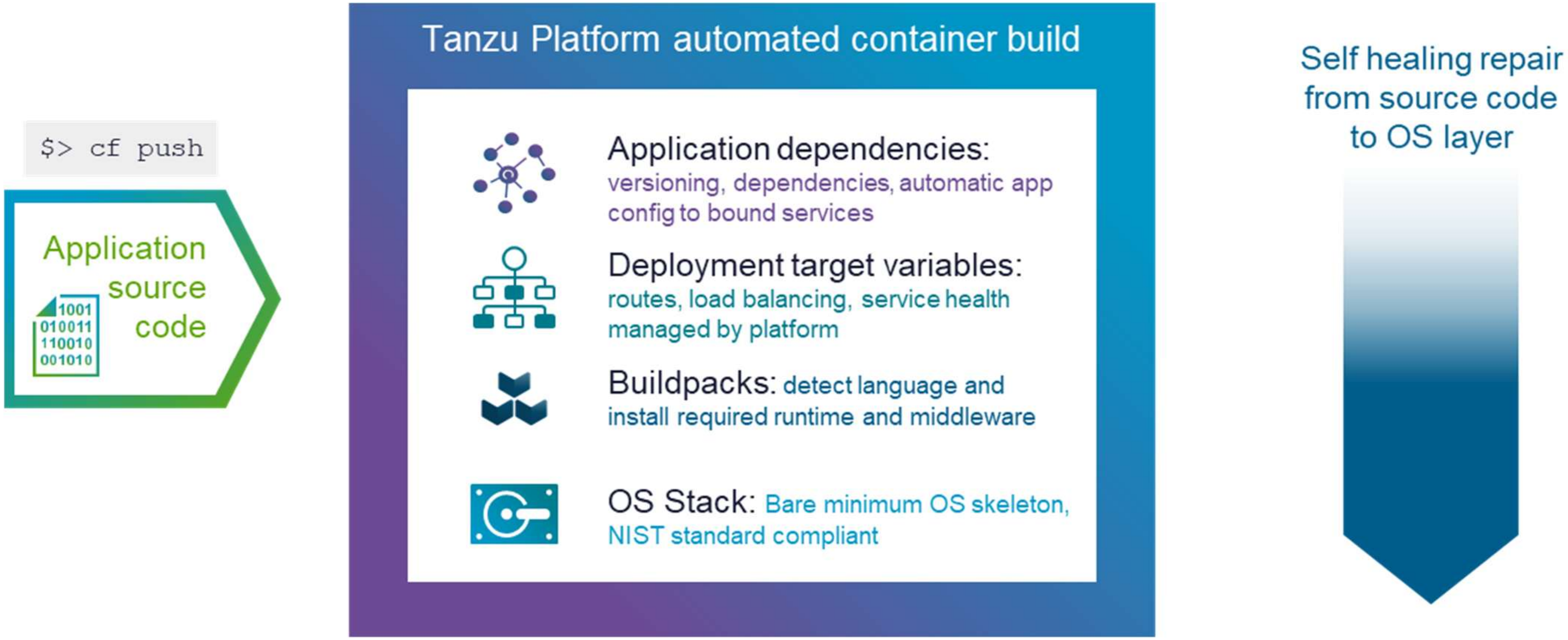
Auto Scaling
Automatically scale app instances based on demand or performance

Continuous Compliance
Automate secure, reliable delivery mechanisms & report dynamically

Trusted and Secure
Ensure zero app downtime with automated vulnerability (CVE) repair, infrastructure repave, and credential rotation



Tanzu Platform for Cloud Foundry (5)



```
$> cf push
```

Application source code
1001
010011
110010
001010
Tanzu Platform automated container build
Self healing repair from source code to OS layer

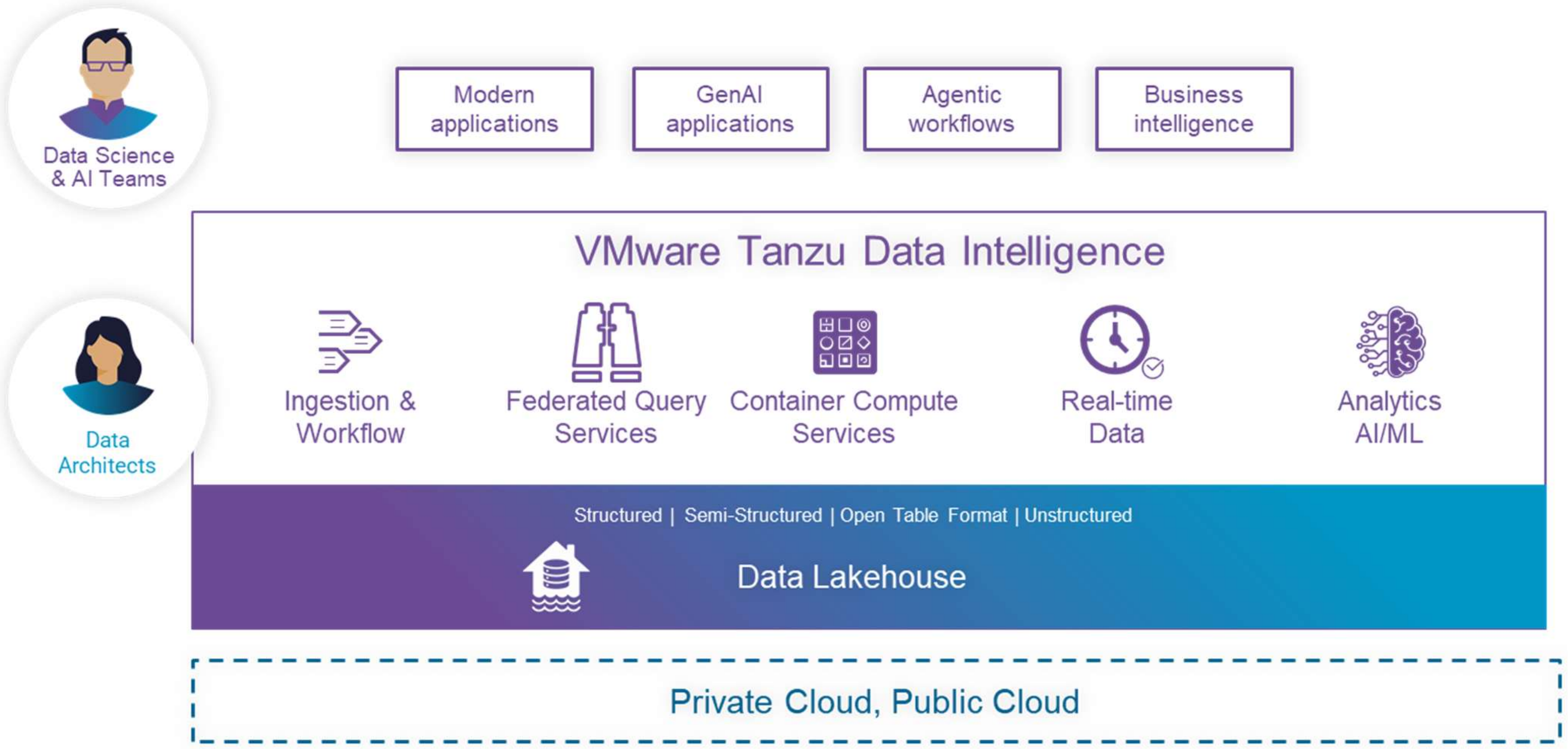
Application dependencies: versioning, dependencies, automatic app config to bound services

Deployment target variables: routes, load balancing, service health managed by platform

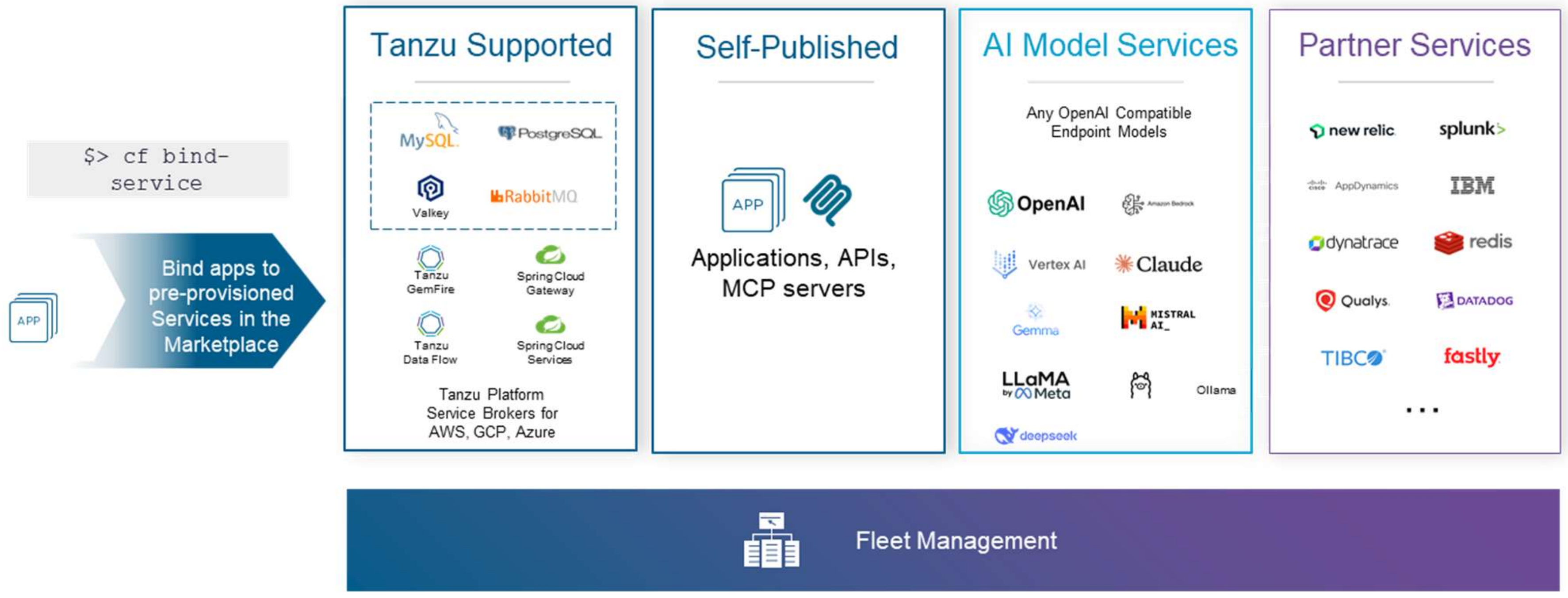
Buildpacks: detect language and install required runtime and middleware

OS Stack: Bare minimum OS skeleton, NIST standard compliant

Tanzu Platform for Cloud Foundry (4)



Tanzu Platform for Cloud Foundry (5)



Demo

VMware Tanzu

VCF TechCon
Powered by VMUG