Pivotal

Pivotal Container Service Overview and Deep Dive

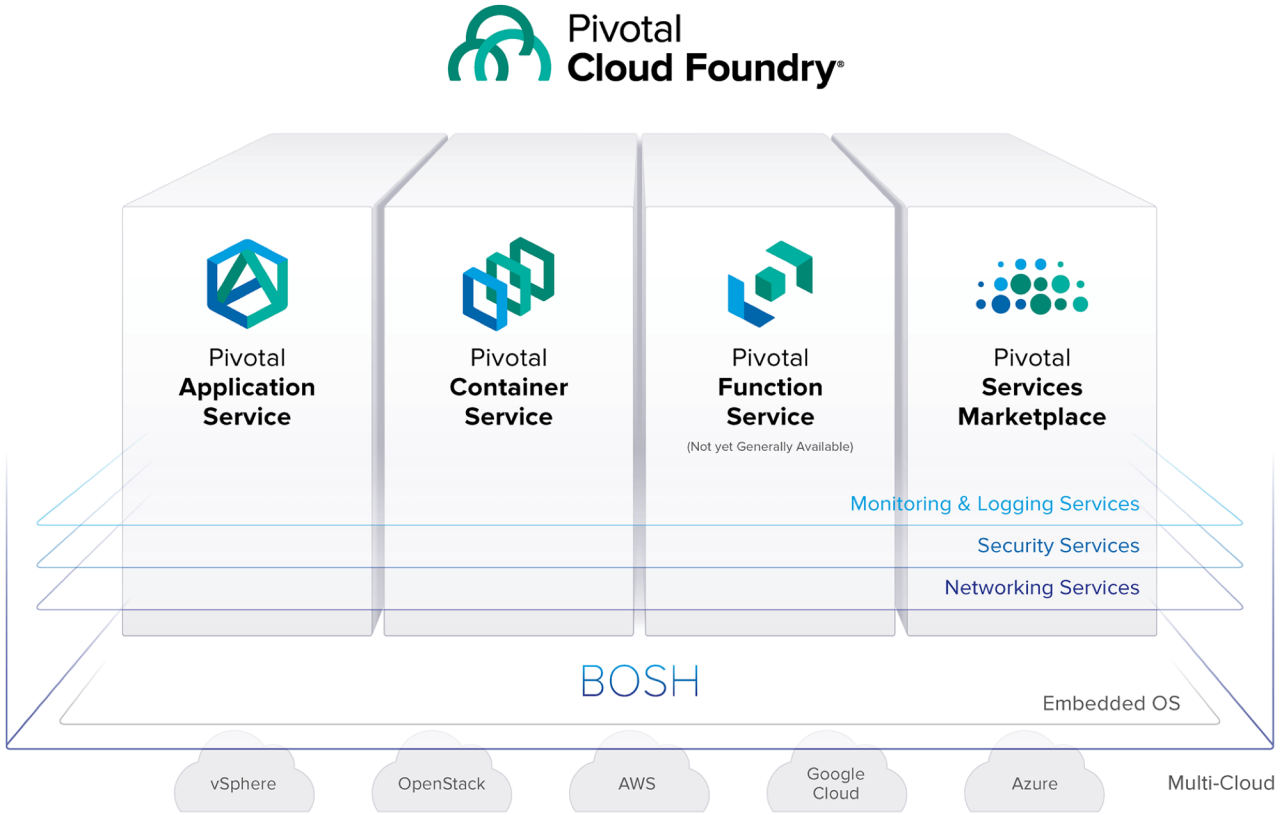
Jaime Aguilar
Platform Architect, Pivotal
jaguilar@pivotal.io
@jaimegag

May 2019

Agenda

- Why PKS?
- What is PKS
- Deep Dive
- Reference Architectures
- Workloads

Pivotal


Pivotal
Cloud Foundry
Pivotal
Application
Service
Pivotal
Container
Service
Pivotal
Function
Service
(Not yet Generally Available)
Pivotal
Services
Marketplace
Monitoring & Logging Services
Security Services
Networking Services
BOSH
Embedded OS
vSphere
OpenStack
AWS
Google
Cloud
Azure
Multi-Cloud

Main Capabilities

Why PKS?

Platform Team Delivering Real Value

Developer Productivity

- Accelerate feedback loops by improving delivery velocity
- Focus on applications, not infrastructure
- Give developers the tools and frameworks to build resilient apps

Operational Efficiency

- Employ 500:1 developer to operator ratio
- Perform zero-downtime upgrades
- Runs the same way on every public/private cloud

Comprehensive Security

- Adopt a defense-in-depth approach
- Continuously update platforms to limit threat impact
- Apply the 3 R's → repair, repave, rotate

High Availability

- Run platforms that stay online under all circumstances
- Scale up and down, in and out, through automation
- Deploy multi-cloud resilience patterns

What Workloads are Realizing these Benefits?

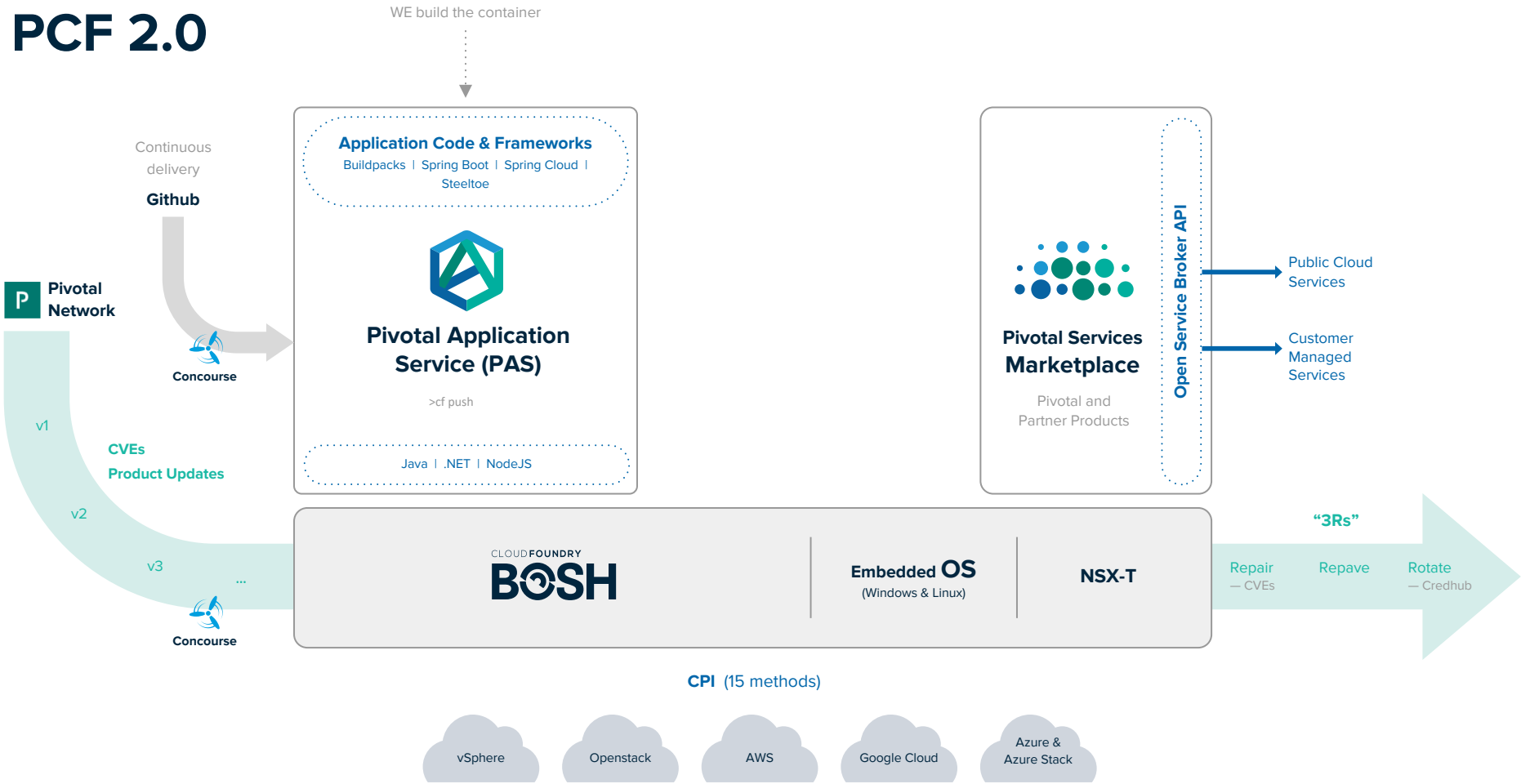
MICROSERVICES

Some MONOLITHIC APPLICATIONS

Some DATA SERVICES

.NET APPLICATIONS

PCF 2.0



Pivotal

Pivotal Application Service (PAS): A Runtime for Apps

Increase speed and deploy code to production thousands of times per month. Use PAS to run Java, .NET, and Node apps.

Best runtime for Spring and Spring Boot — Spring's microservice patterns—and Spring Boot's executable jars—are ready-made for PAS.

Turnkey microservices operations and security — Spring Cloud Services brings microservices best practices to PAS. It includes Config Server, Service Registry, and Circuit Breaker Dashboard.

A native Windows and .NET experience — Use PAS to run new apps built with .NET Core. Run your legacy .NET Framework apps on PAS too, using the .NET Hosted Web Core buildpack. Push applications to containers running on Windows Server 2016.

Built for apps — PAS has everything to need to run apps. Buildpacks manage runtime dependencies; metrics, logging, and scaling are done for you. Multitenancy, and blue/green deployment patterns are built-in. Extend apps with a rich service catalog.

Container-ready — PAS supports the OCI format for Docker images. Run platform-built and developer-built containers.

Can we realize these benefits for other workloads too?

MICROSERVICES

Some MONOLITHIC APPLICATIONS

Some DATA SERVICES

.NET APPLICATIONS

CONTAINERS

COTS

More MONOLITHIC APPLICATIONS

Stateful or MICROSERVICES Clusters

Can we realize these benefits for other workloads too?

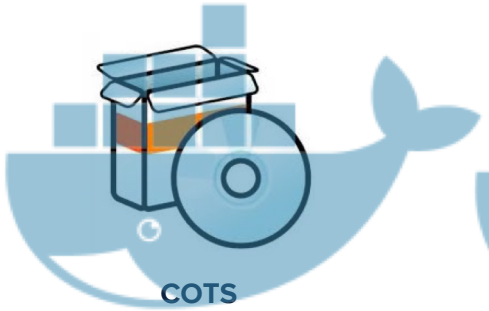
MICROSERVICES

Some MONOLITHIC APPLICATIONS

Some DATA SERVICES

.NET APPLICATIONS

CONTAINERS

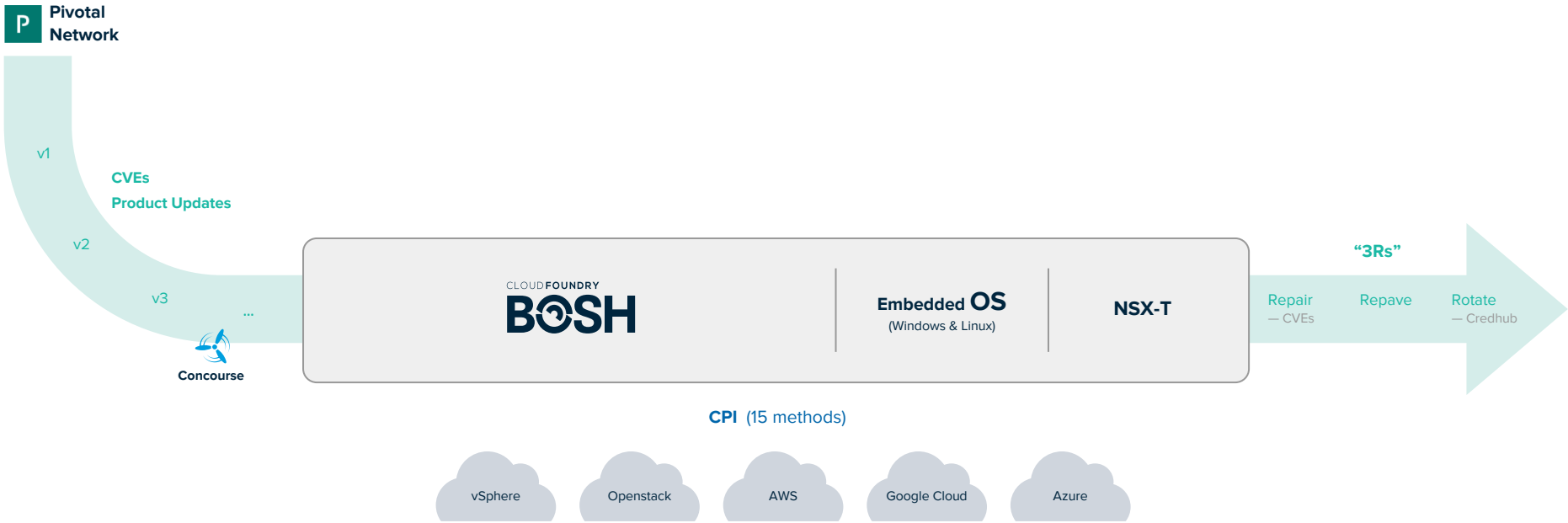


COTS

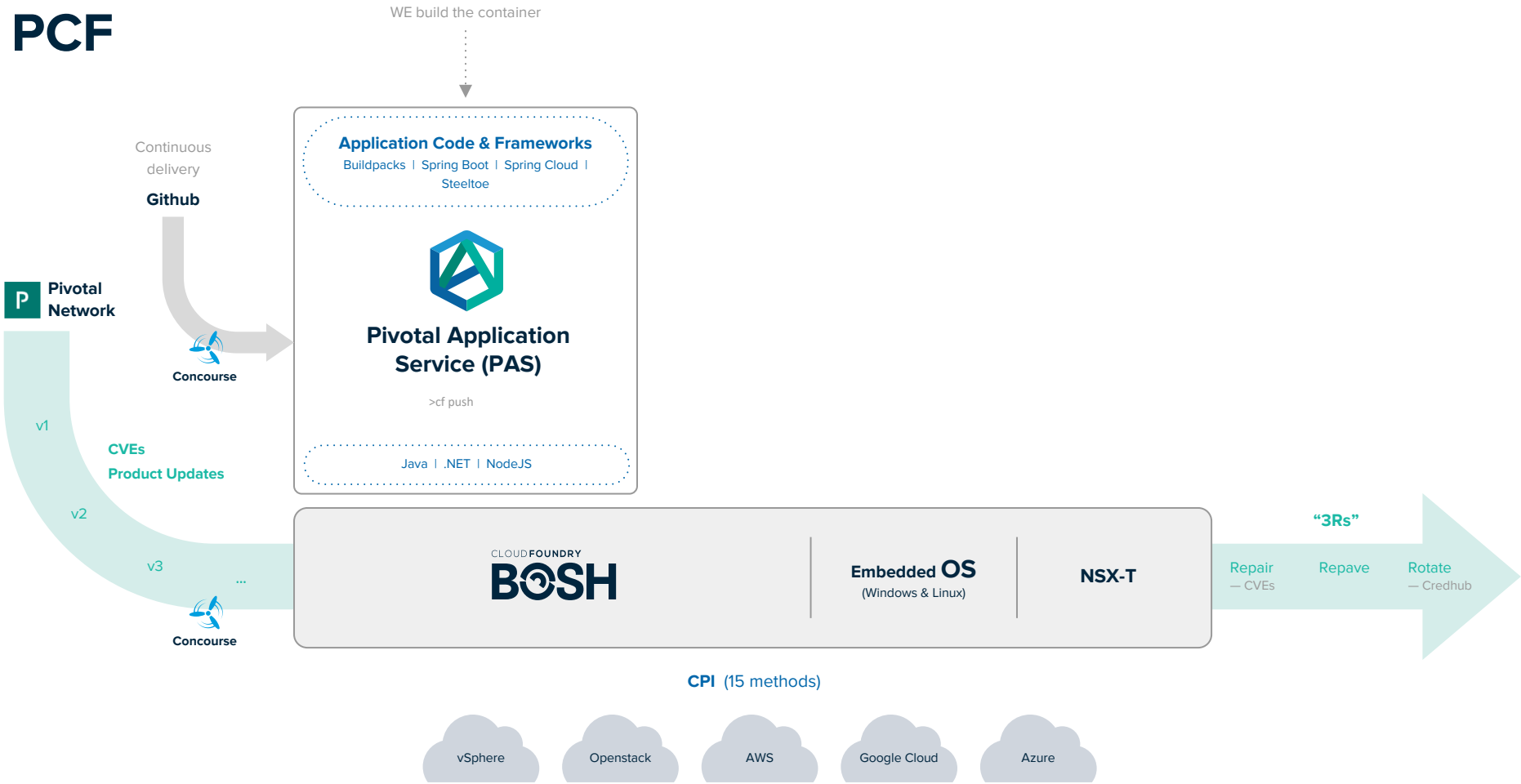
More MONOLITHIC APPLICATIONS

Stateful or MICROSERVICES Clusters

PCF

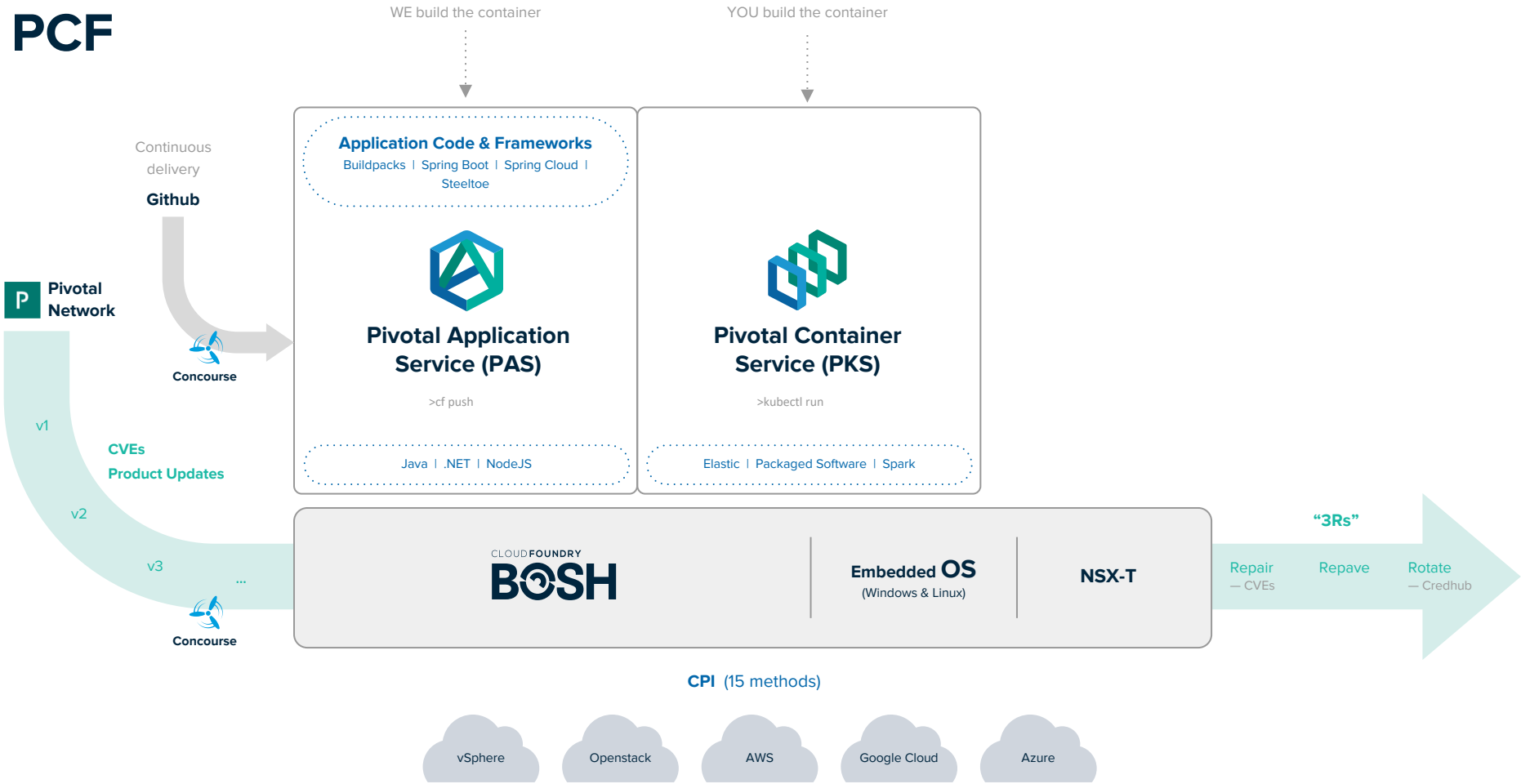

Pivotal Network
v1
CVEs
Product Updates
v2
v3
...
Concourse
CLOUDFOUNDRY
BOSH
Embedded OS
(Windows & Linux)
NSX-T
"3Rs"
Repair
— CVEs
Repave
Rotate
— Credhub
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure

PCF


WE build the container
Continuous delivery
Github
Application Code & Frameworks
Buildpacks | Spring Boot | Spring Cloud | Steeltoe
Pivotal Application Service (PAS)
>cf push
Java | .NET | NodeJS
Pivotal Network
Concourse
v1
CVEs
Product Updates
v2
v3
...
Concourse
CLOUDFOUNDRY BOSH
Embedded OS
(Windows & Linux)
NSX-T
"3Rs"
Repair
— CVEs
Repave
Rotate
— Credhub
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure

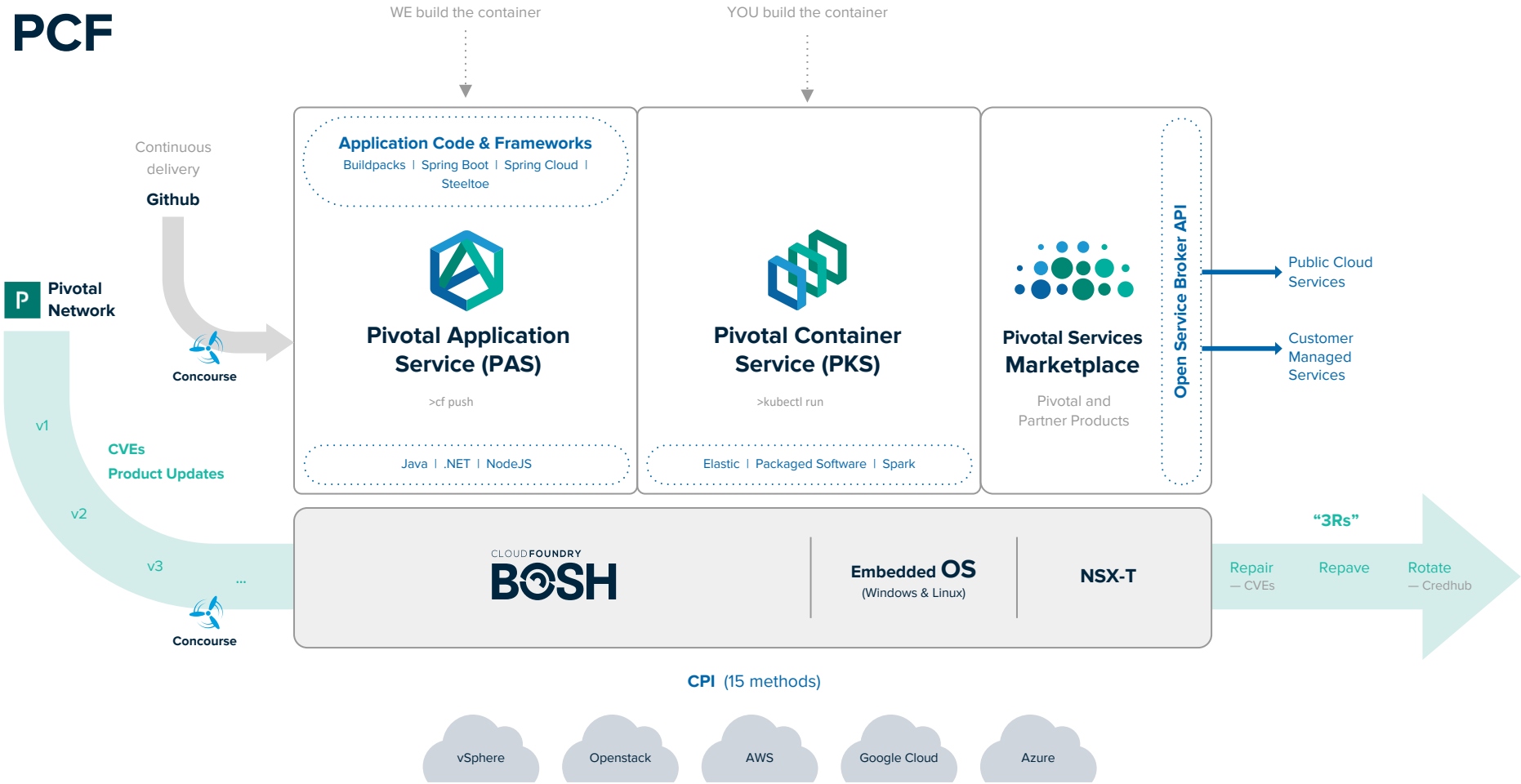
Pivotal

PCF


WE build the container
YOU build the container
Continuous delivery
Github
Application Code & Frameworks
Buildpacks | Spring Boot | Spring Cloud | Steeltoe
Pivotal Network
Pivotal Application Service (PAS)
>cf push
Pivotal Container Service (PKS)
>kubectl run
Concourse
v1
CVEs
Product Updates
v2
v3
...
Java | .NET | NodeJS
Elastic | Packaged Software | Spark
CLOUDFOUNDRY BOSH
Embedded OS (Windows & Linux)
NSX-T
"3Rs"
Repair — CVEs
Repave
Rotate — Credhub
Concourse
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure

Pivotal

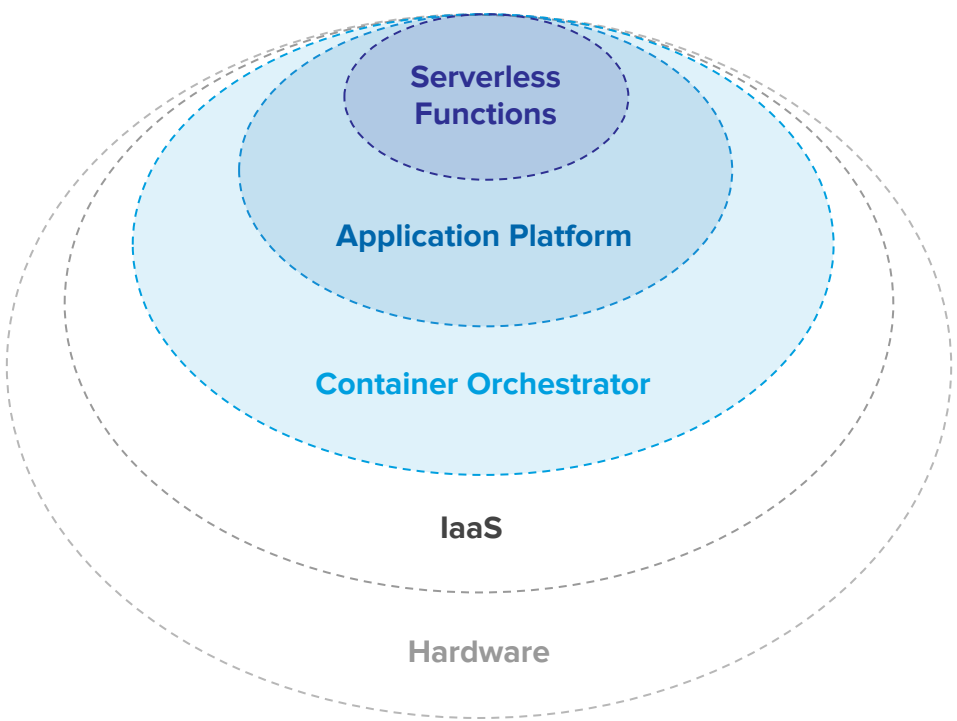
PCF


WE build the container
YOU build the container
Continuous delivery
Github
Application Code & Frameworks
Buildpacks | Spring Boot | Spring Cloud | Steeltoe
Pivotal Application Service (PAS)
>cf push
Java | .NET | NodeJS
Pivotal Container Service (PKS)
>kubectl run
Elastic | Packaged Software | Spark
Pivotal Services Marketplace
Pivotal and Partner Products
Open Service Broker API
Public Cloud Services
Customer Managed Services
Pivotal Network
Concourse
v1
CVEs
Product Updates
v2
v3
...
Concourse
CLOUDFOUNDRY BOSH
Embedded OS
(Windows & Linux)
NSX-T
"3Rs"
Repair
— CVEs
Repave
Rotate
— Credhub
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure

Pivotal

Choose the right tool for the job

Higher flexibility and less enforcement of standards



Lower development complexity and higher operational efficiency

Strategic goal (portfolio optimization): Push as many workloads as technically feasible to the top of the platform hierarchy

Kubernetes

Kubernetes is an open-source platform designed to automate deploying, scaling, and operating **application containers**.

With Kubernetes, you are able to quickly and efficiently respond to customer demand:

- Deploy your applications quickly and predictably.
- Scale your applications on the fly.
- Roll out new features seamlessly.
- Limit hardware usage to required resources only.
- Manage your applications like cattle instead of pets.

https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

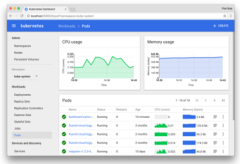
Kubernetes is a Runtime for Containerized Workloads

App User

Dev / Apps

IT / Ops

```
> kubectl
```



Kubernetes Dashboard

kubernetes

CONTAINERS

COTS

MONOLITHIC APPLICATIONS

MICROSERVICES

Compute

Storage

Networking

Pivotal

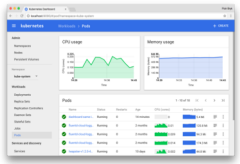
...but Kubernetes alone is not enough for enterprises

App User

Load Balancing / Routing

Dev / Apps

```
> kubectl
```



Kubernetes Dashboard

Container Image Registry

App Monitoring

App Logging

kubernetes

CONTAINERS

COTS

MONOLITHIC APPLICATIONS

MICROSERVICES

External Data Services

Cluster Provisioning

IT / Ops

Command Line / API

Management GUI

Monitoring GUI

OS Updates

K8S Updates

Log & Monitor

Backup & Restore

OS Images

K8S Images

Recover & Restart

Provision & Scale

Compute

Storage

Networking

Pivotal

...but Kubernetes alone is not enough for enterprises

Pivotal

Pivotal Container Service (PKS) provides what's missing

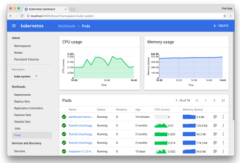
Pivotal
vmware
Google

App User

vmware NSX

Dev / Apps

> kubectl



Kubernetes Dashboard

HARBOR™

WAVEFRONT

vRealize LogInsight

fluentbit

kubernetes

CONTAINERS COTS MONOLITHIC APPLICATIONS MICROSERVICES

OPEN SERVICE BROKER API

PKS Control Plane

IT / Ops

> pks

Operations Manager

vRealize Operations*

CLOUD FOUNDRY BOSH™

Compute

Storage

vmware NSX

Pivotal

on any Cloud

Pivotal

vmware

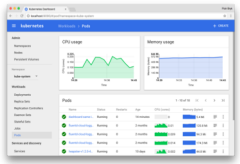
Google

App User

vmware NSX

Dev / Apps

> kubectl



Kubernetes Dashboard

HARBOR™

WAVEFRONT

vRealize LogInsight

fluentbit

kubernetes

CONTAINERS

COTS

MONOLITHIC APPLICATIONS

MICROSERVICES

OPEN SERVICE BROKER API

PKS Control Plane

IT / Ops

> pks

Operations Manager

vRealize Operations*

CLOUD FOUNDRY BOSH™

flannel

vmware NSX

vmware vSphere

Azure

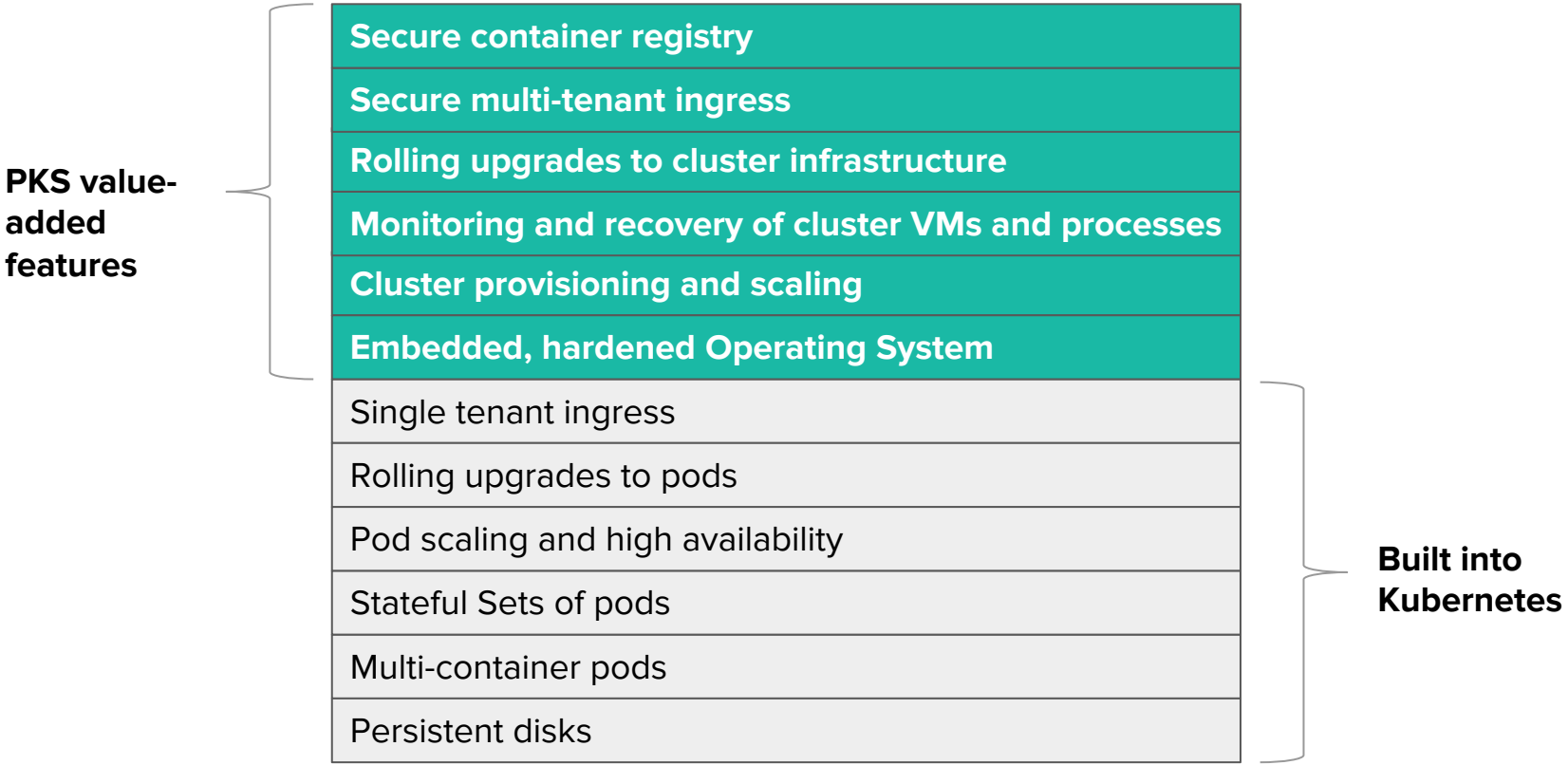
Google Cloud Platform

aws

openstack.

Pivotal

What PKS adds to Kubernetes



“On a good day our DIY k8s solution could provision a cluster in 45 minutes. On a bad day it would fail to deploy ~30 times” - IT Operator

Three Questions

Upgrades

How many times does K8s release per year?

Answer: Major releases quarterly; minor releases/patches frequently

PIVOTAL'S OPINION

- Automated upgrades on-demand with BOSH
- Automatic patching with Concourse pipelines
- Self-healing nodes on failure

Multi-tenancy

How many clusters are you going to need?

Answer: More than one required to provide true multi-tenancy

PIVOTAL'S OPINION

- Self-service, on-demand provisioning of clusters
- Pre-defined T-shirt size clusters
- Scale clusters up and down

Network

Is your network ready? (How mature is your SDN?)

Answer: Probably not; if you require tickets/manual process for network or firewall rules, you aren't ready

PIVOTAL'S OPINION

- Microsegmentation with NSX-T
- Automated IP allocation and load balancer provisioning
- Monitoring & troubleshooting with familiar VMware tooling

Pivotal

Main Capabilities

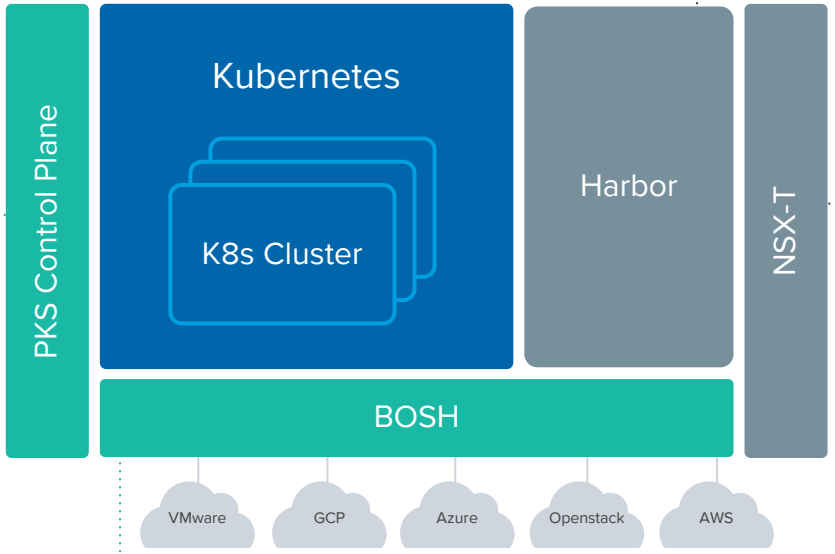
What is PKS?

Pivotal Container Service™

Enterprise-Grade Kubernetes

Built with open-source Kubernetes
Constant compatibility with the latest stable release Kubernetes —no proprietary extensions.

PKS Control Plane
Use the PKS CLI and API to create, operate, and scale your clusters.



Harbor
An enterprise-class container registry. Includes vulnerability scanning, identity management, and more.

NSX-T
Network management, security, and load balancing out-of-the-box with VMware NSX-T. Multi-cloud, multi-hypervisor.

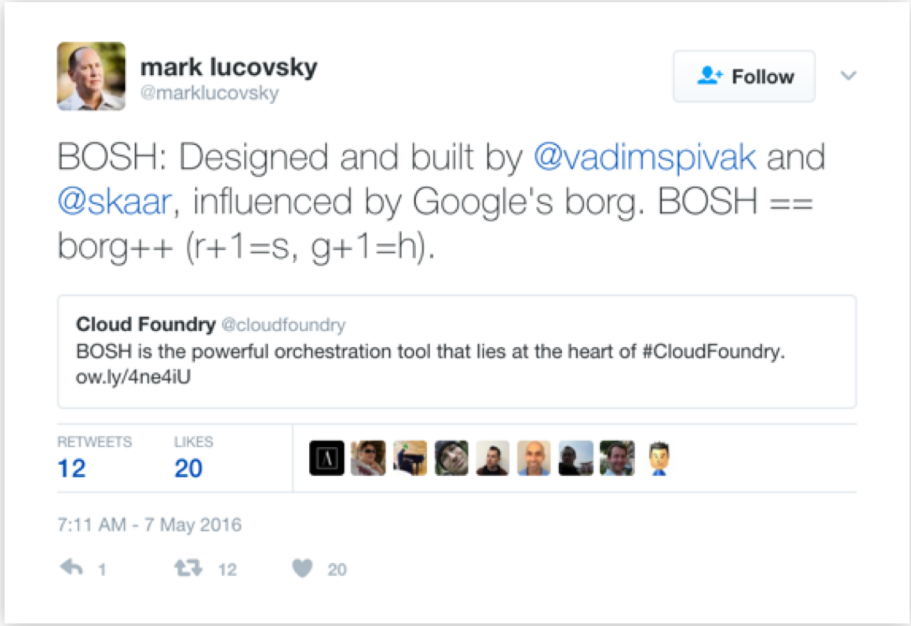
BOSH
Reliable and consistent operational experience for any cloud.

Main Capability #1

BOSH

BOSH

BOSH is an open source tool for release engineering, deployment, lifecycle management, and monitoring of distributed systems.



BOSH


Package


Provision


Deploy


Monitor


Upgrade

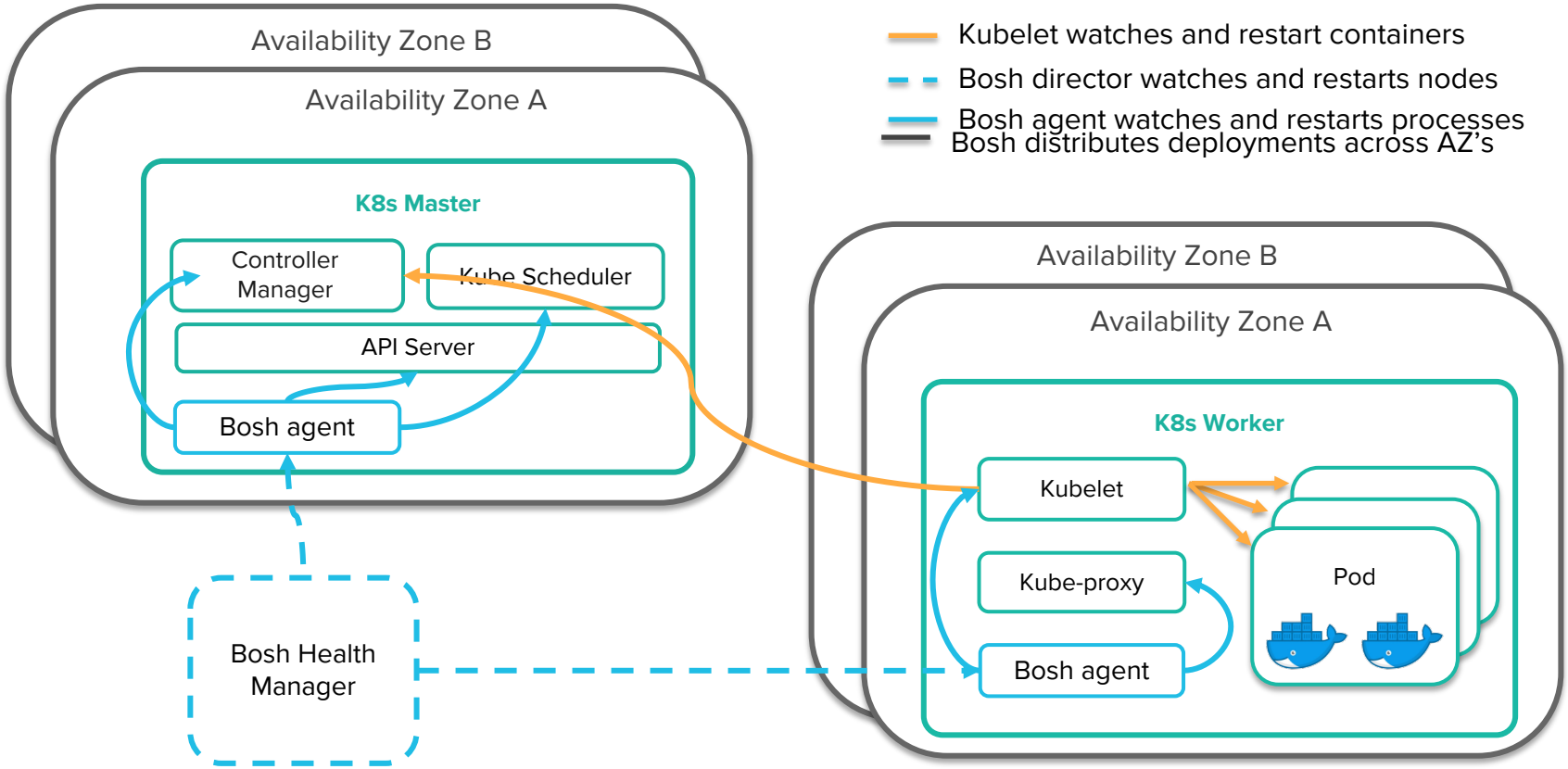
- ➔ Packaging w/ embedded OS
- ➔ Server provisioning on any IaaS
- ➔ Software deployment across AZs

- ➔ Health monitoring (server & processes)
- ➔ Self-healing w/ Resurrector
- ➔ Storage management
- ➔ Rolling upgrades with canaries
- ➔ Easy scaling of clusters
- ➔ Repeatability and Consistency

Pivotal


PKS does for your Kubernetes
what
Kubernetes does for your apps
Pivotal

PKS Health Management

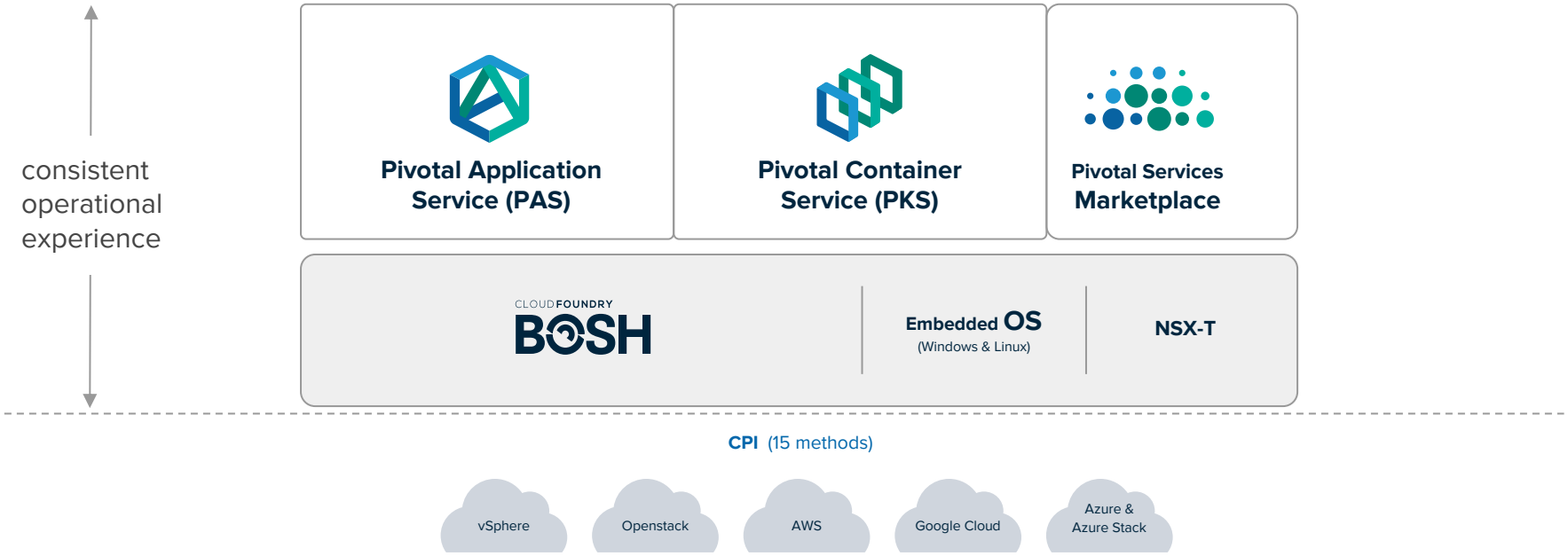


Pivotal

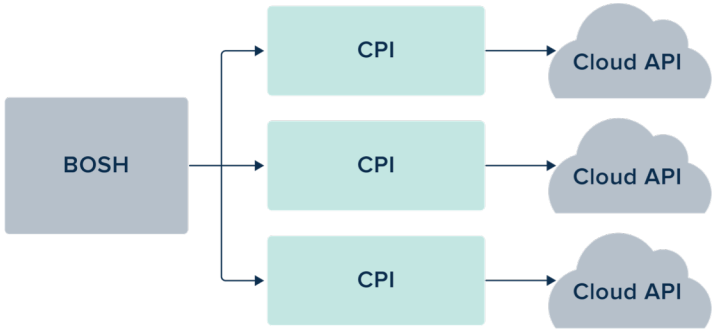
Main Capability #2

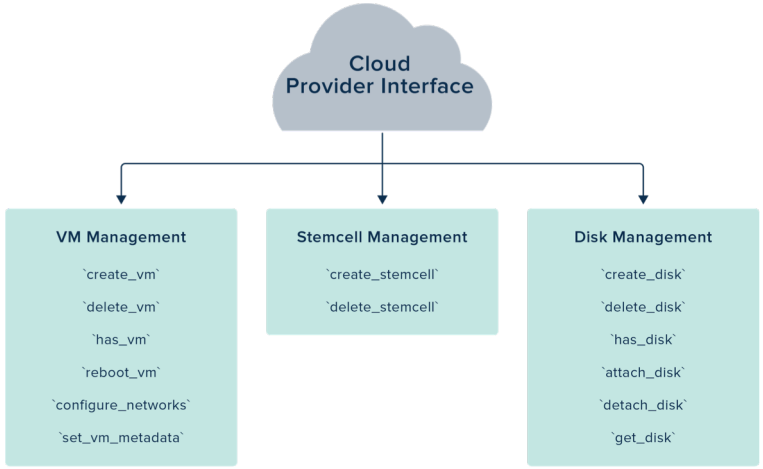
Multi Cloud

Multicloud


consistent
operational
experience
Pivotal Application
Service (PAS)
Pivotal Container
Service (PKS)
Pivotal Services
Marketplace
CLOUDFOUNDRY
BOSH
Embedded OS
(Windows & Linux)
NSX-T
CPI (15 methods)
vSphere
Openstack
AWS
Google Cloud
Azure &
Azure Stack

Multi-Cloud with BOSH + CPI





vmware® openstack® Azure aws Google Cloud Platform

Main Capability #3

Multi Cluster

Tenancy

Multi-tenancy models remain weak in Kubernetes alone

We provide solutions for this today

Two models supported

No other "on prem" solution has this!!!

Multi-tenant clusters

- Leverage Kubernetes namespaces

Limitations with Kubernetes alone

- Noisy neighbors (workloads can affect other tenants)
- Share the same network
- Share DNS
- Shared Configuration
- …

We add

- Network microsegmentation with NSX-T
  - Eliminating "Share the same network"

Multi (Single-tenant) clusters

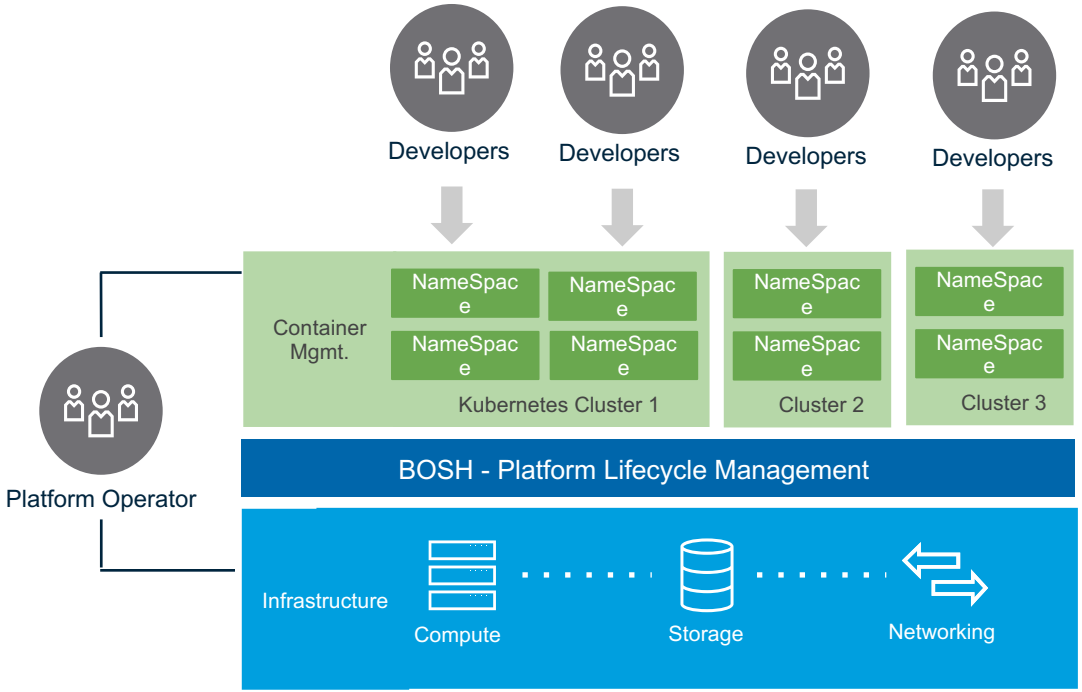
It is having an API for creation and management that enables this!!!

- Every tenant gets their own cluster

Addresses limitations

- Single tenant worker VMs (depend on the hypervisor to ensure host is properly shared)
- Every cluster has own network segment
- Every cluster has own DNS
- Every cluster has own configuration
- …

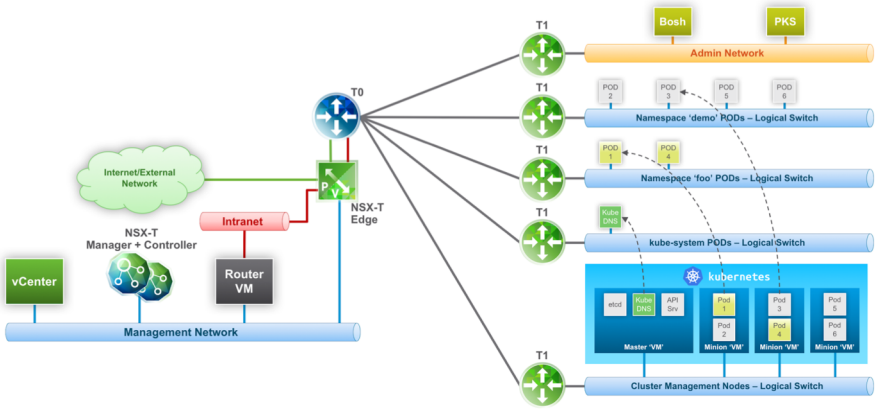
Flexible Multi-Tenancy



Main Capability #4

NSX-T

vmware NSX

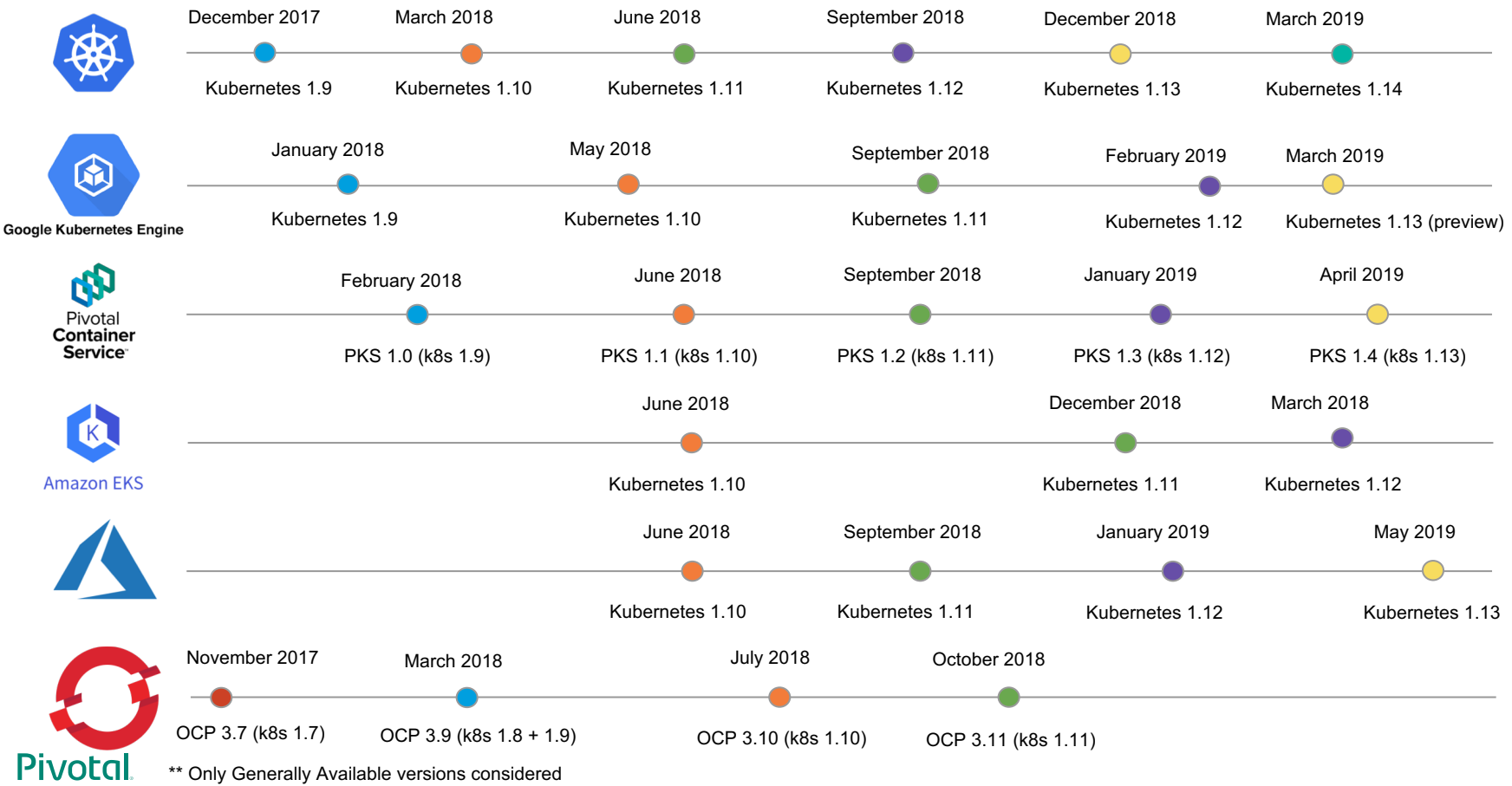


- Unified VM to Container Networking
- On-demand network virtualization
- Microsegmentation
- Full Network Visibility
- Enterprise Support
- Pod-Level Container Networking
- Load Balancing
- Network Security policies
- Tenant - level isolation
- Unique logical switch per K8s namespace

Pivotal

And...

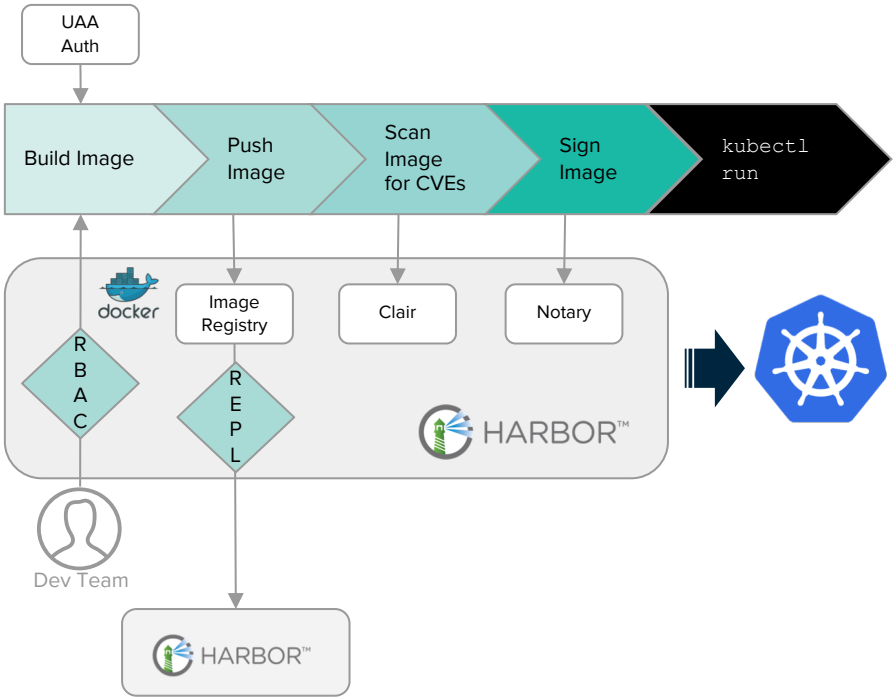
Keeping the pace with k8s / giving Devs the latest



** Only Generally Available versions considered

HARBOR™

An enterprise-class registry server for Docker images



- Role-Based Access Control (RBAC)
- LDAP/AD Integration
- Image Vulnerability Scanning (Clair)
- Notary Image Signing
- Policy-Based Image Replication
- Graphical User Portal & RESTful API
- Image Deletion & Garbage Collection
- Auditing

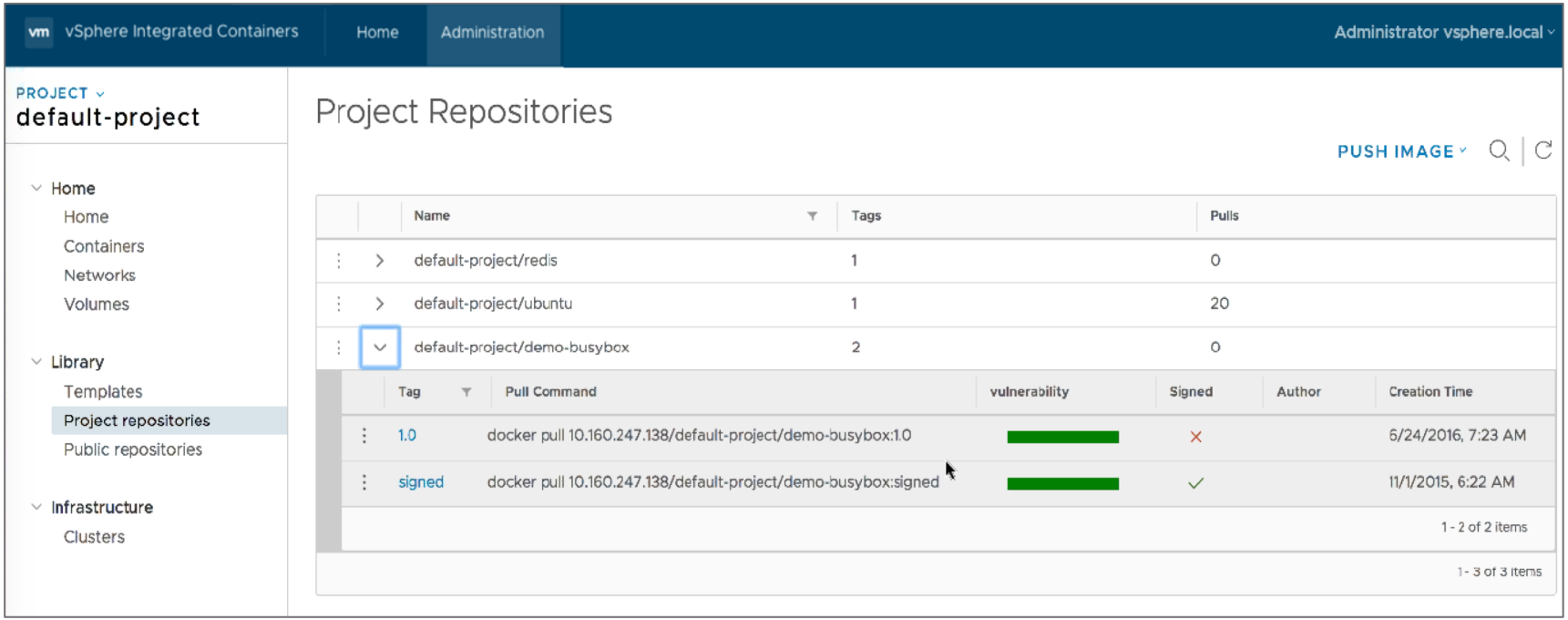
Pivotal

Container Registry Vulnerability scanning

Vulnerability	Severity	Package	Current version	Fixed in version
CVE-2016-1252	high	apt	1.0.1ubuntu2.6	1.0.1ubuntu2.17
CVE-2016-5011	low	util-linux	2.20.1-5.1ubuntu20.3	
Description: The parse_dos_extended function in partitions/dos.c in the libblkid library in util-linux allows physically proximate attackers to cause a denial of service (memory consumption) via a crafted MSDOS partition table with an extended partition boot record at zero offset.				
CVE-2014-9114	low	util-linux	2.20.1-5.1ubuntu20.3	
CVE-2013-0157	low	util-linux	2.20.1-5.1ubuntu20.3	
CVE-2017-6350	low	vim	2:7.4.052-1ubuntu3	
CVE-2017-5953	low	vim	2:7.4.052-1ubuntu3	
CVE-2017-6349	low	vim	2:7.4.052-1ubuntu3	
CVE-2016-1248	medium	vim	2:7.4.052-1ubuntu3	2:7.4.052-1ubuntu3.1
CVE-2017-9525	low	cron	3.0pl1-124ubuntu2	
CVE-2017-10685	medium	ncurses	5.9+20140118-1ubuntu1	
CVE-2016-7543	medium	bash	4.3-7ubuntu1.5	4.3-7ubuntu1.7

Pivotal

Container Registry Image Signing

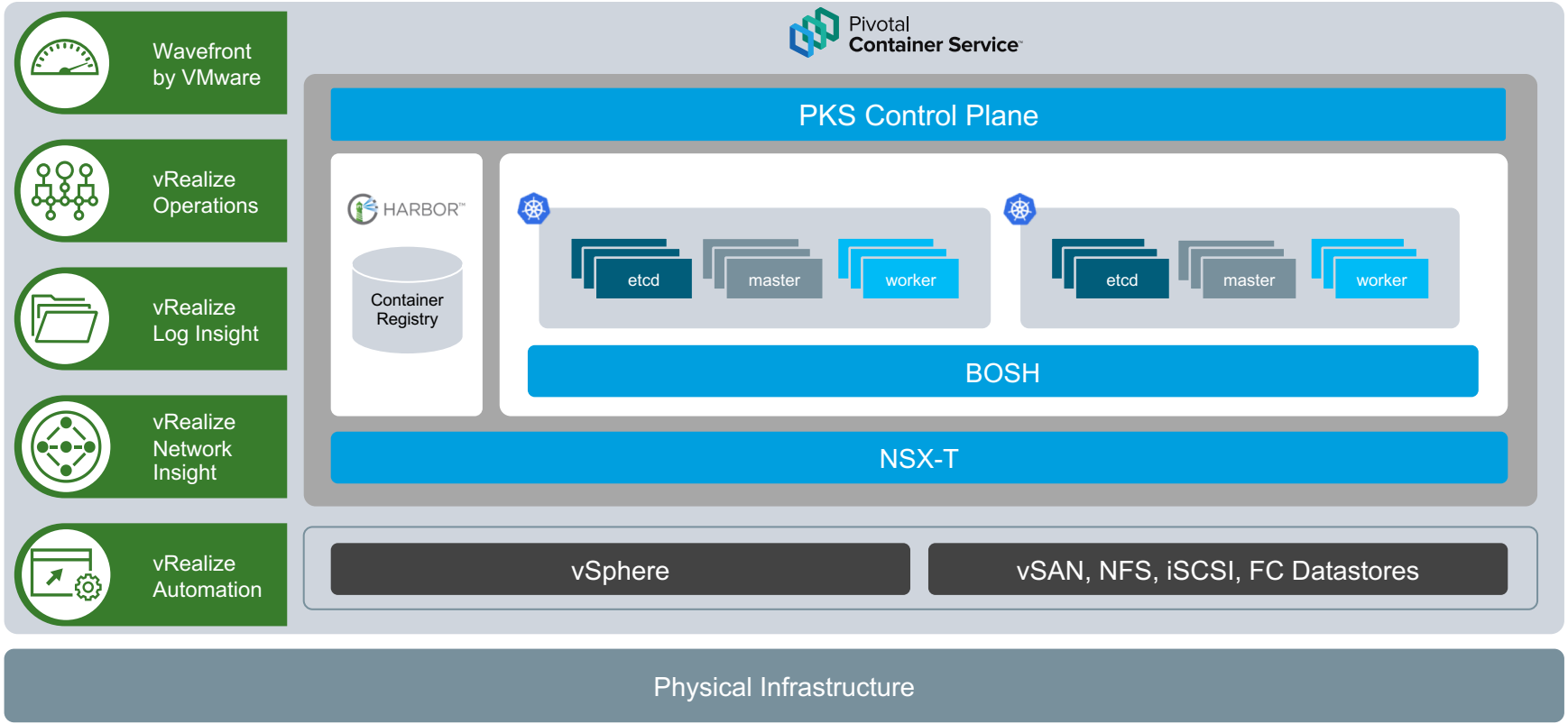



```
root@jt-dev:/var/log/harbor/2017-08-09# export DOCKER_CONTENT_TRUST=0
root@jt-dev:/var/log/harbor/2017-08-09# docker pull 10.160.247.138/default-project/demo-busybox:1.0
Error response from daemon: unknown: The image is not signed in Notary.
root@jt-dev:/var/log/harbor/2017-08-09#
```

Pivotal

Pivotal Container Service™

Integrations w/ VMware



Pivotal

Deep Dive

A Day in Life with PKS

Platform Operator (Alana)

- Install/Manage PKS
- Configure cluster plans
- Apply a patch / update
- Onboard Cluster Owner via RBAC
- Operate Bosh

Cluster Owner (Cody)

- Create a cluster
- Scale a cluster
- Create Network Policy
- Onboard App Dev via RBAC
- ...

App Dev (Naomi)

- Deploy an app
- Expose app with service type: LoadBalancer
- Expose app with Ingress
- ...

BOSH

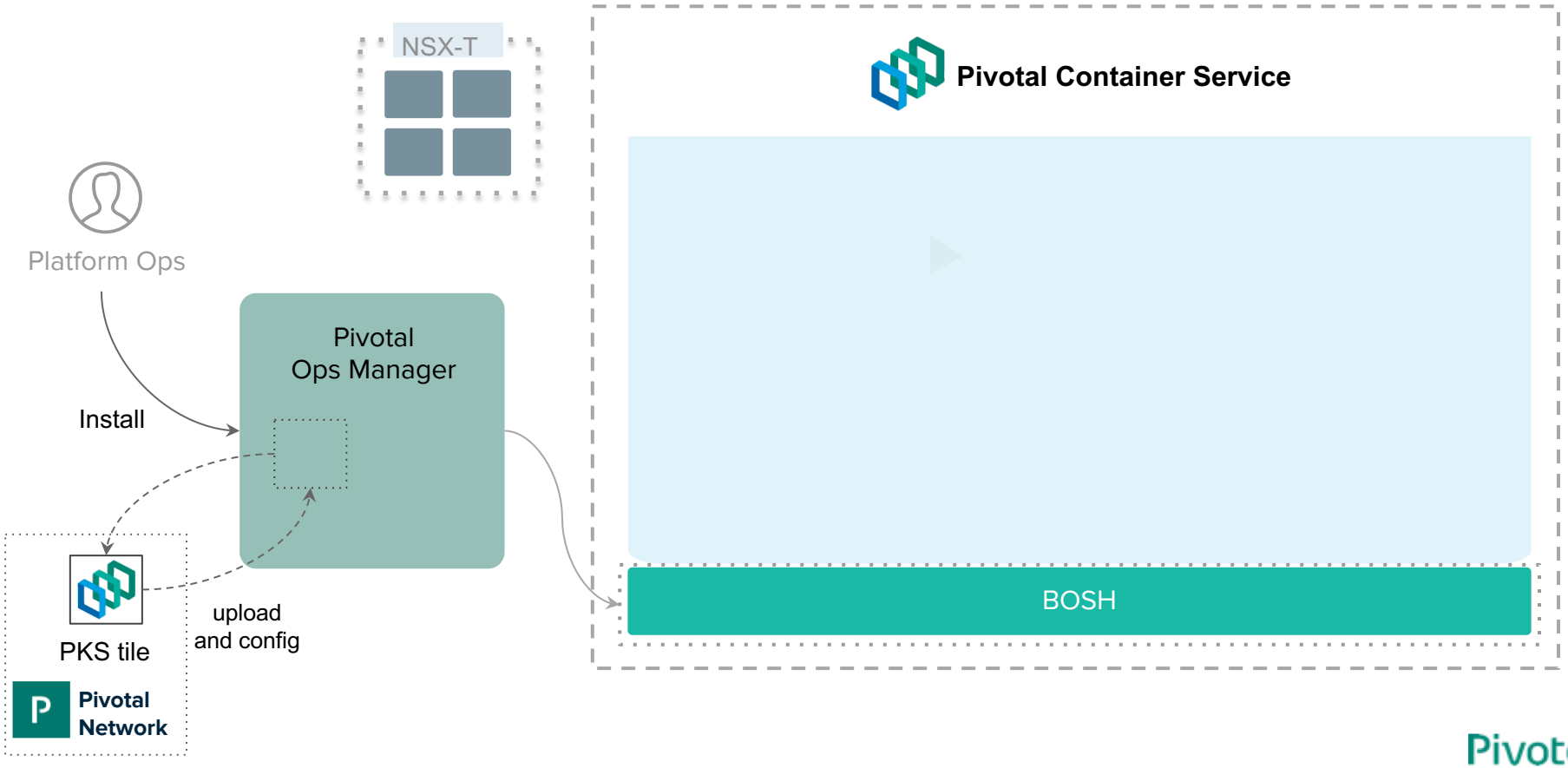
vmware NSX

Automation

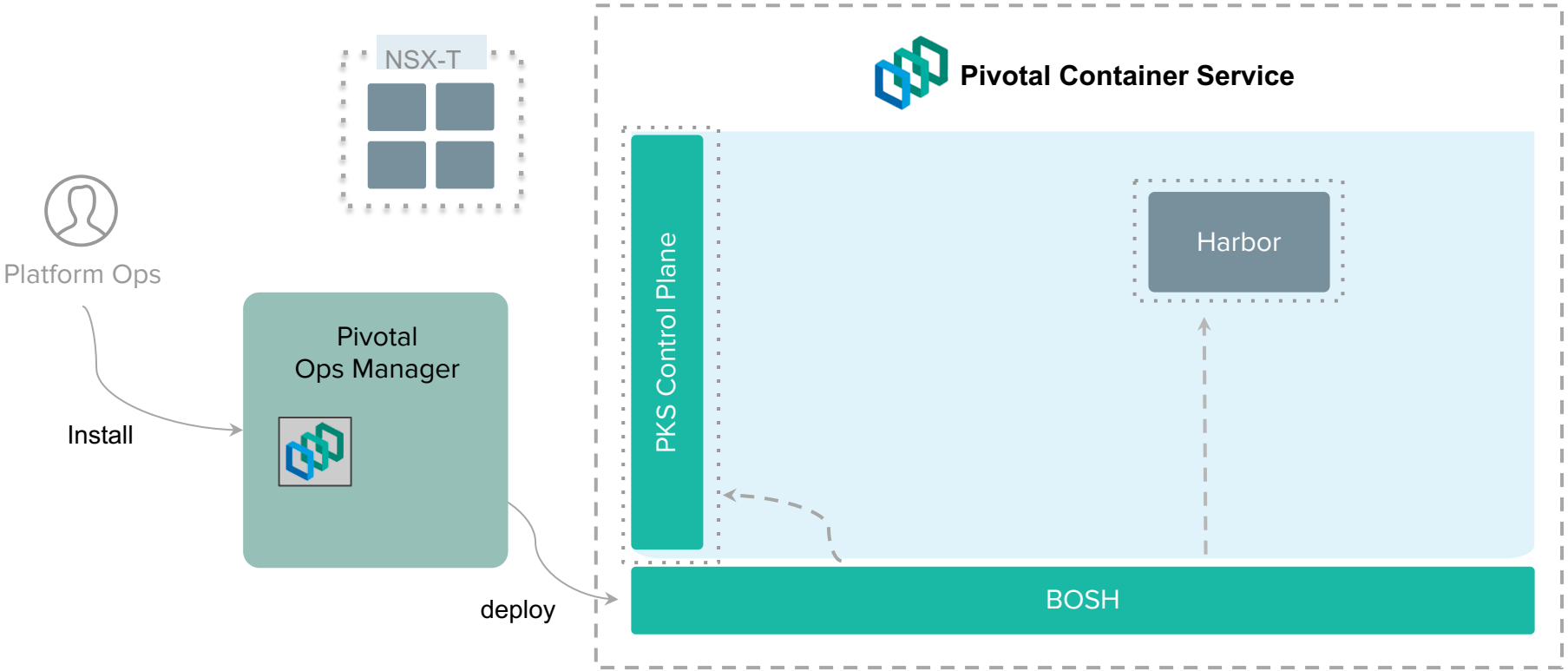
- Health Management (server & process)
- Network Automation

Pivotal

Installing PKS

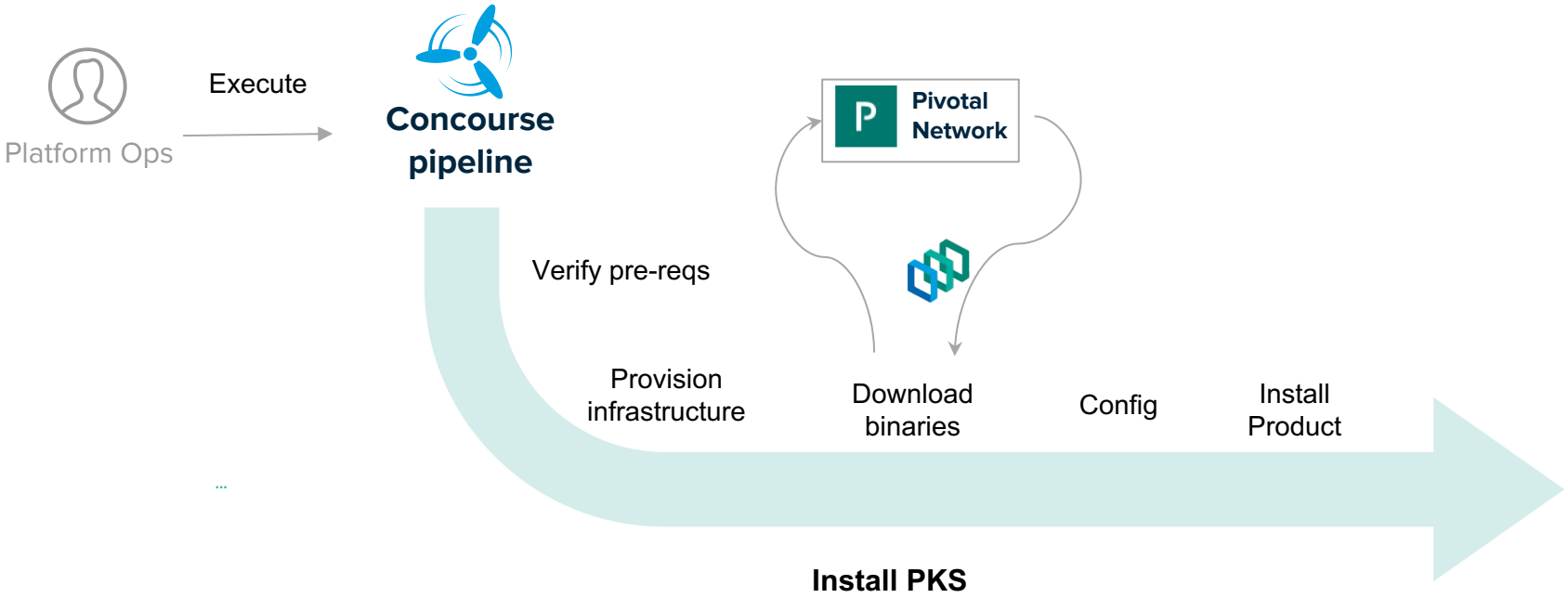


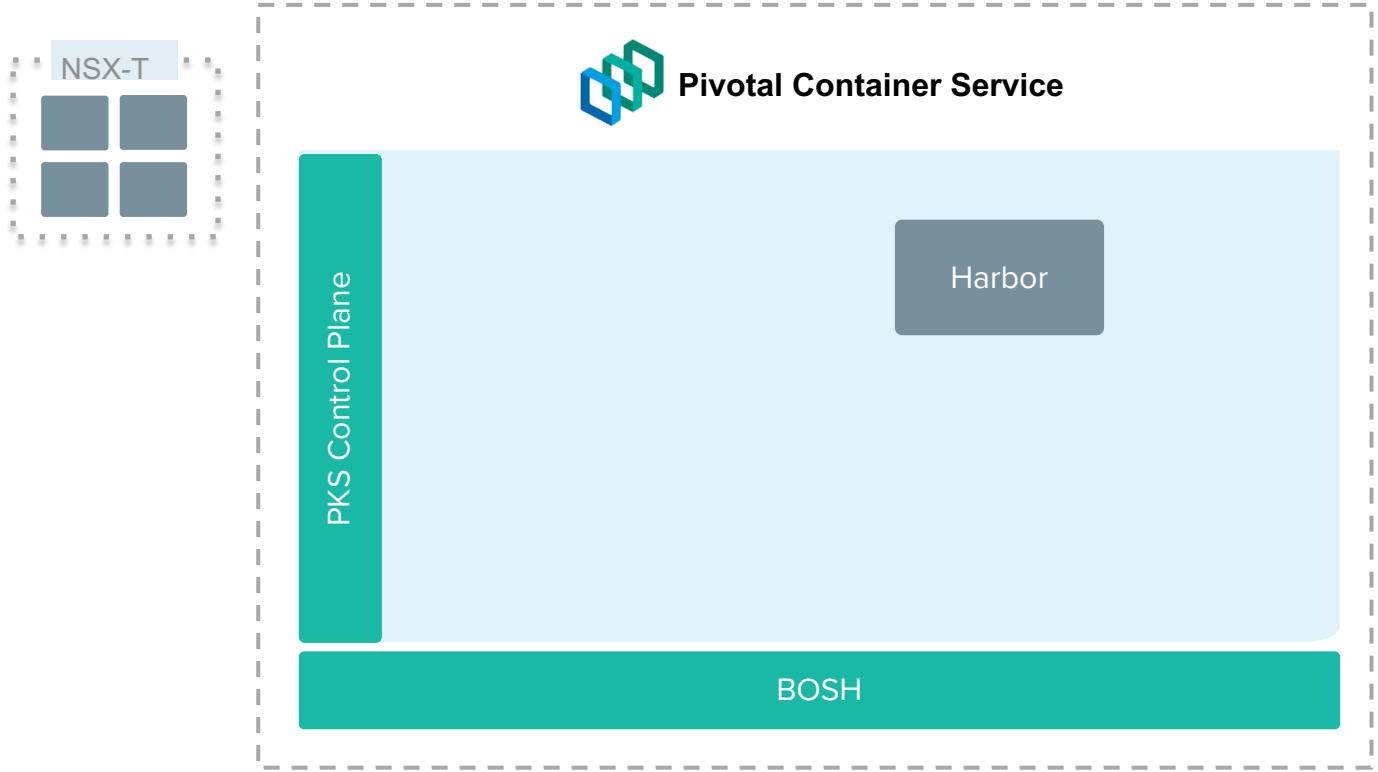
Installing PKS



Pivotal

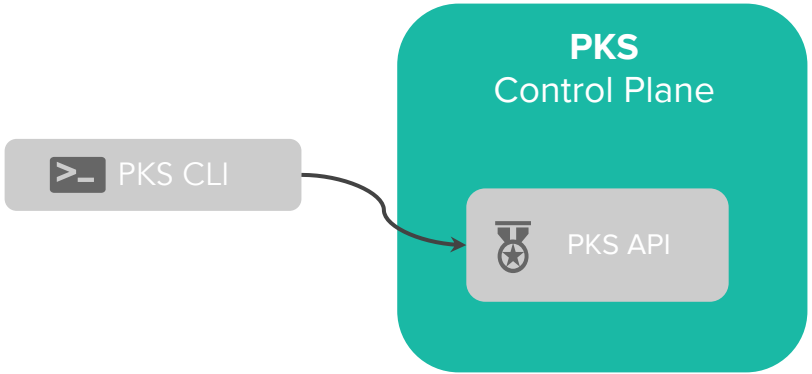
… or


Platform Ops
Execute
Concourse pipeline
Pivotal Network
Verify pre-reqs
Provision infrastructure
Download binaries
Config
Install Product
Install PKS

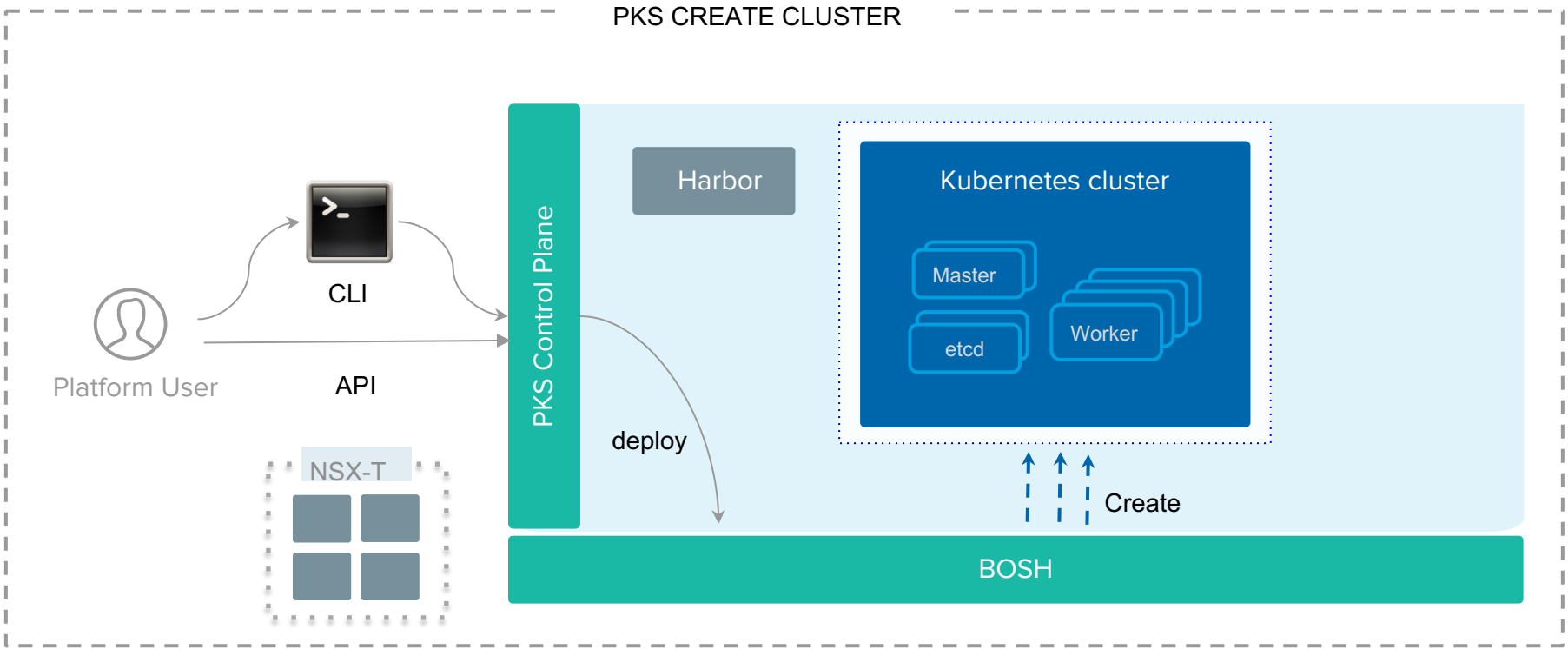

NSX-T
Pivotal Container Service
PKS Control Plane
Harbor
BOSH

PKS User Interaction

- The PKS Management VM runs the PKS API together with the Broker, UAA and a MySQL DB.
- The PKS API orchestrates the initial kubernetes cluster deployments and scaling of those clusters.
- A single PKS VM can manage hundreds of Kubernetes cluster.
- The PKS CLI is a single binary that can be installed on a Mac, Windows, or Linux to drive the PKS API.

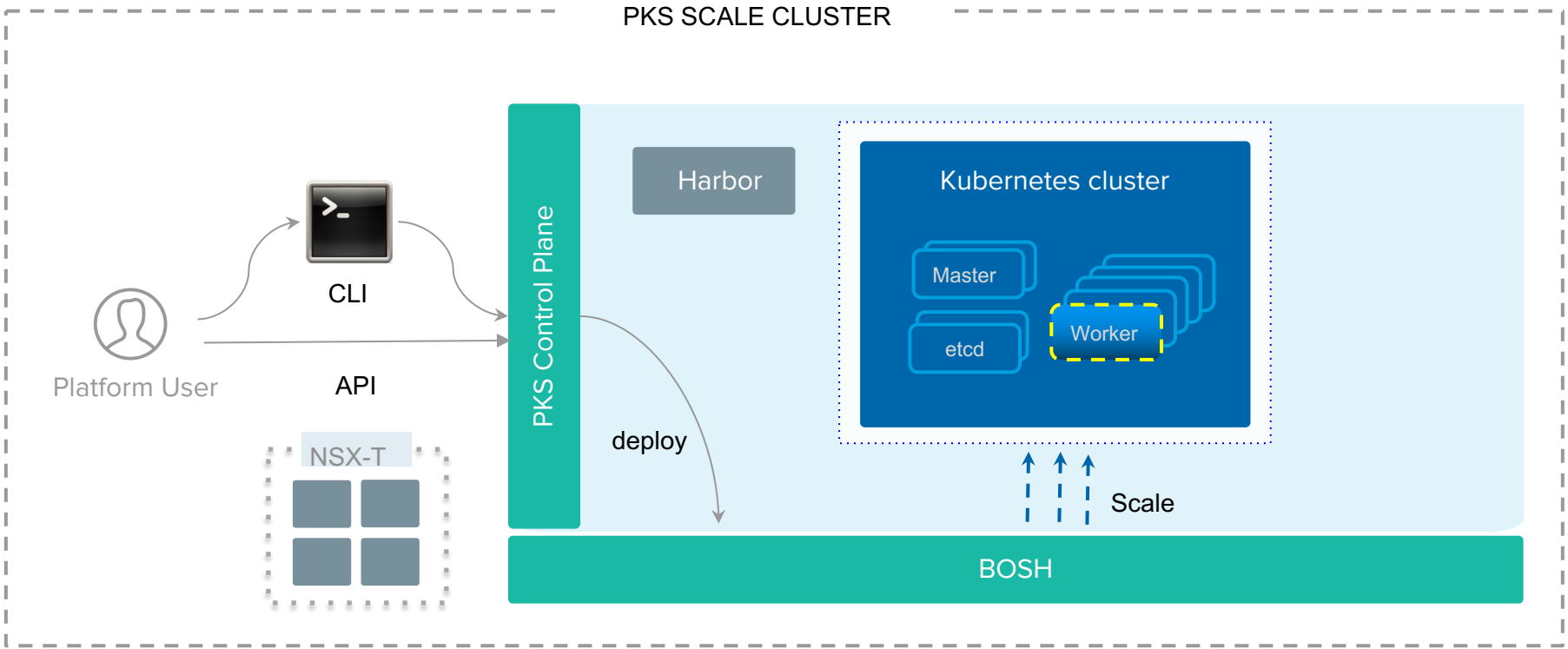


Creating a new K8s Cluster

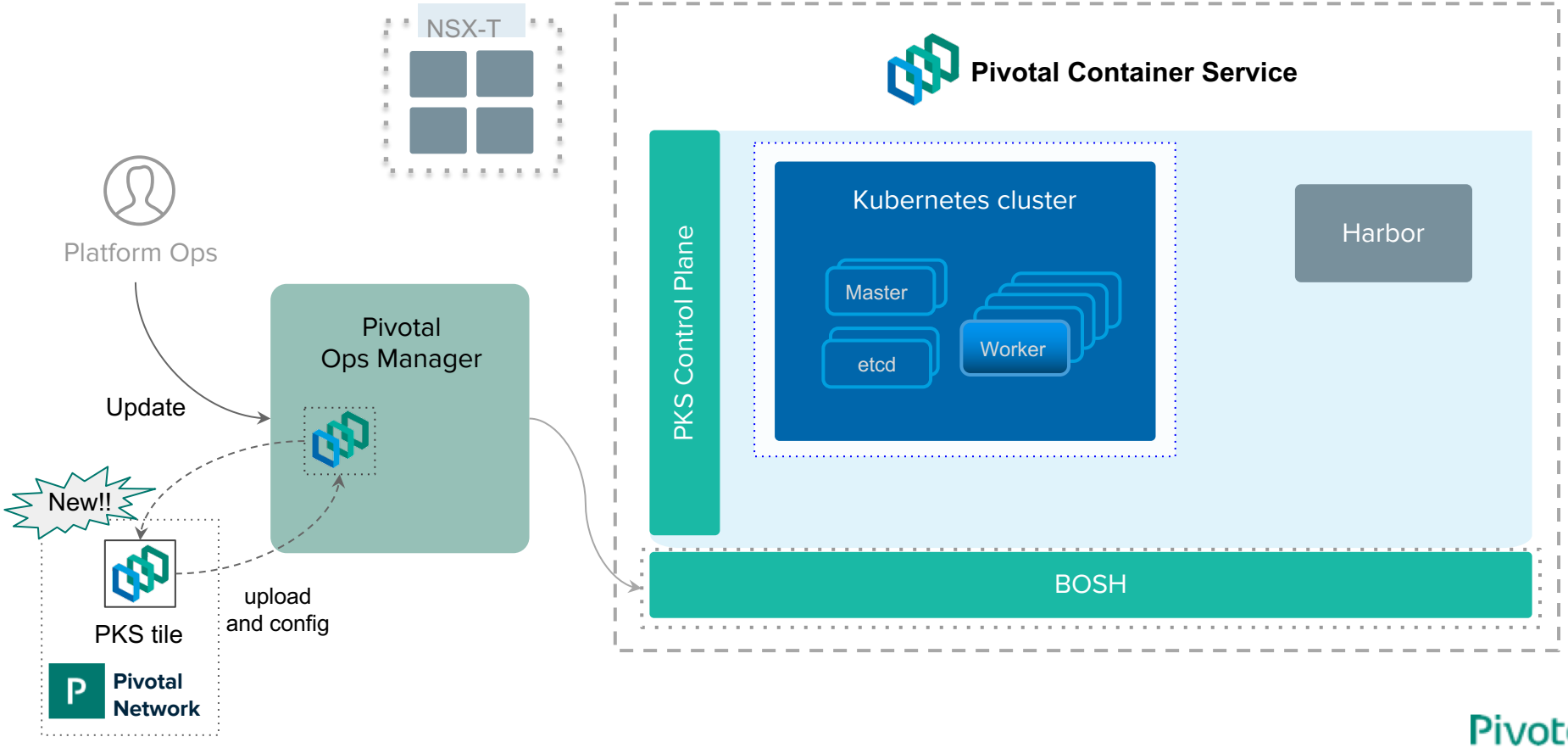


Pivotal

Scaling a Kubernetes Cluster

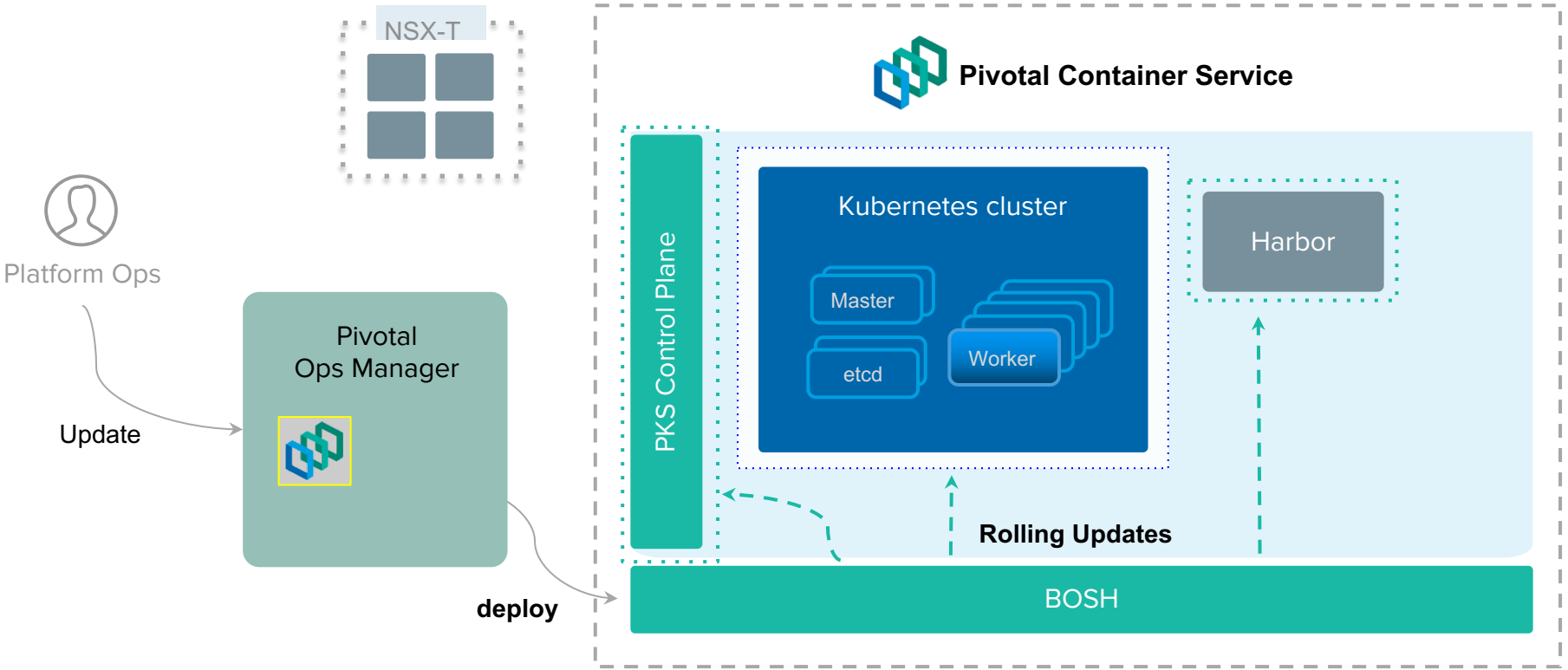


Platform Ops updates PKS



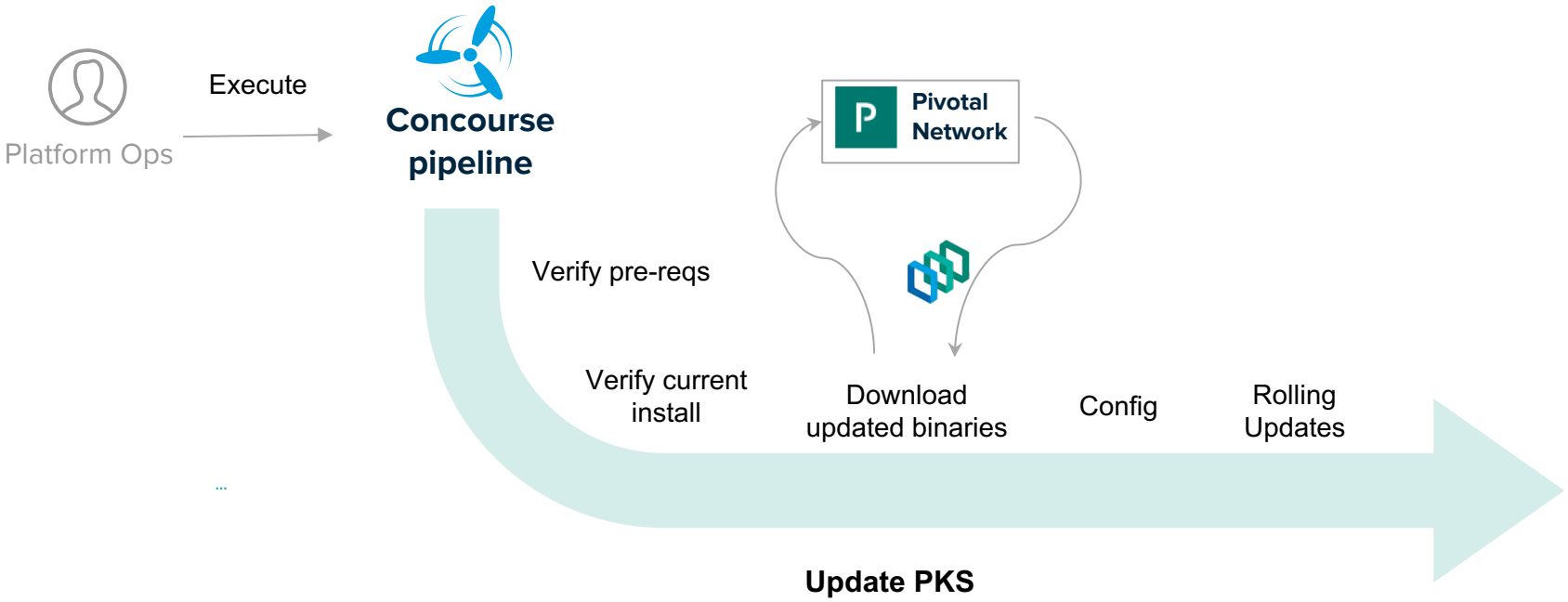
Pivotal

Platform Ops updates PKS



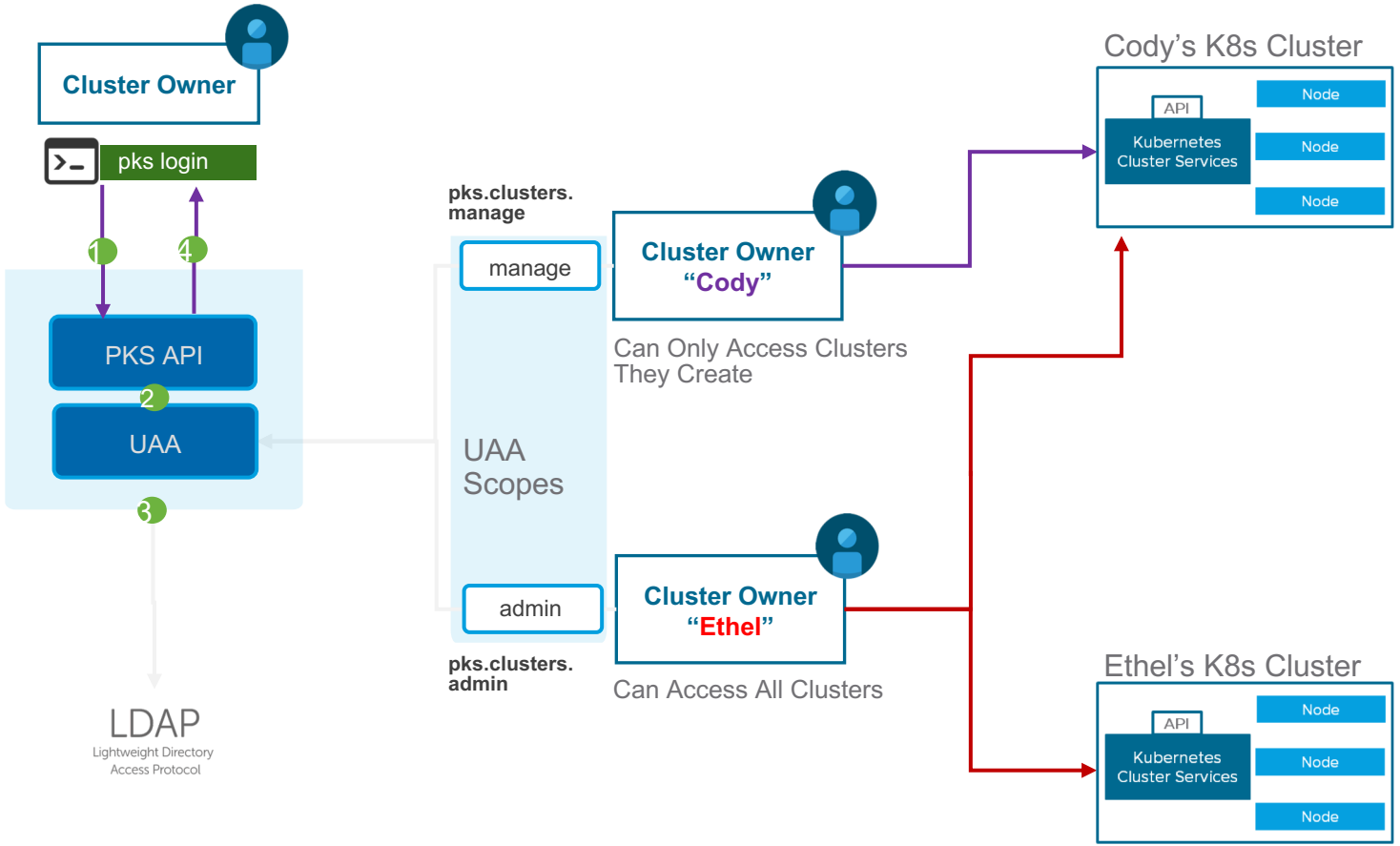
Pivotal

… or

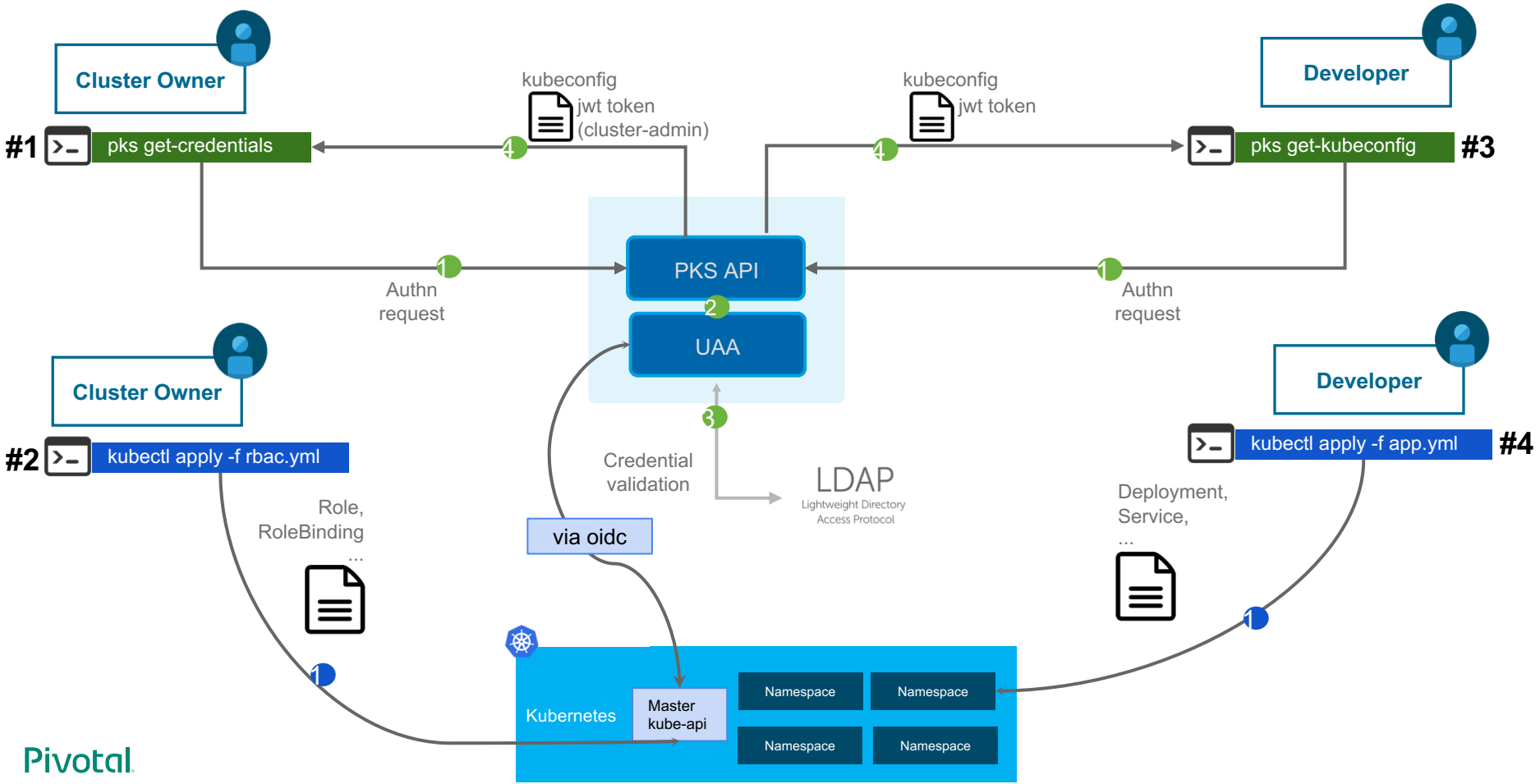

Platform Ops
Execute
Concourse pipeline
Pivotal Network
Verify pre-reqs
Verify current install
Download updated binaries
Config
Rolling Updates
Update PKS

PKS Control Plane - Centralized Access Management

- Identity Backed by Enterprise Standards (LDAP/AD)
- PKS RBAC Controls who can create, scale, and update K8s clusters on demand



K8s Cluster - Access Authentication/Authorization

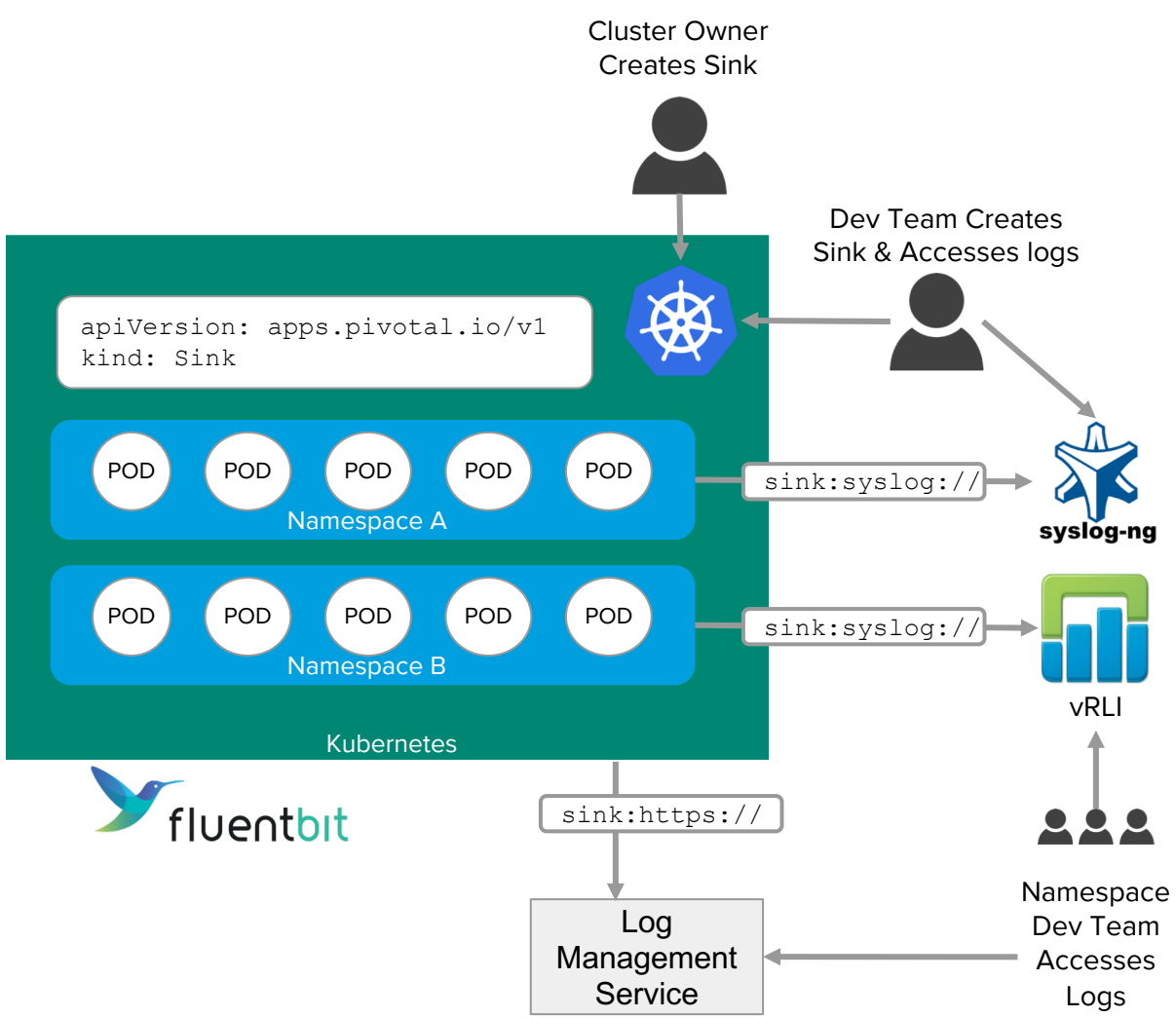


Pivotal

Dev Productivity: Log Sinks

Application Logging by Cluster or Namespace

- Syslog endpoints
- Webhook endpoints

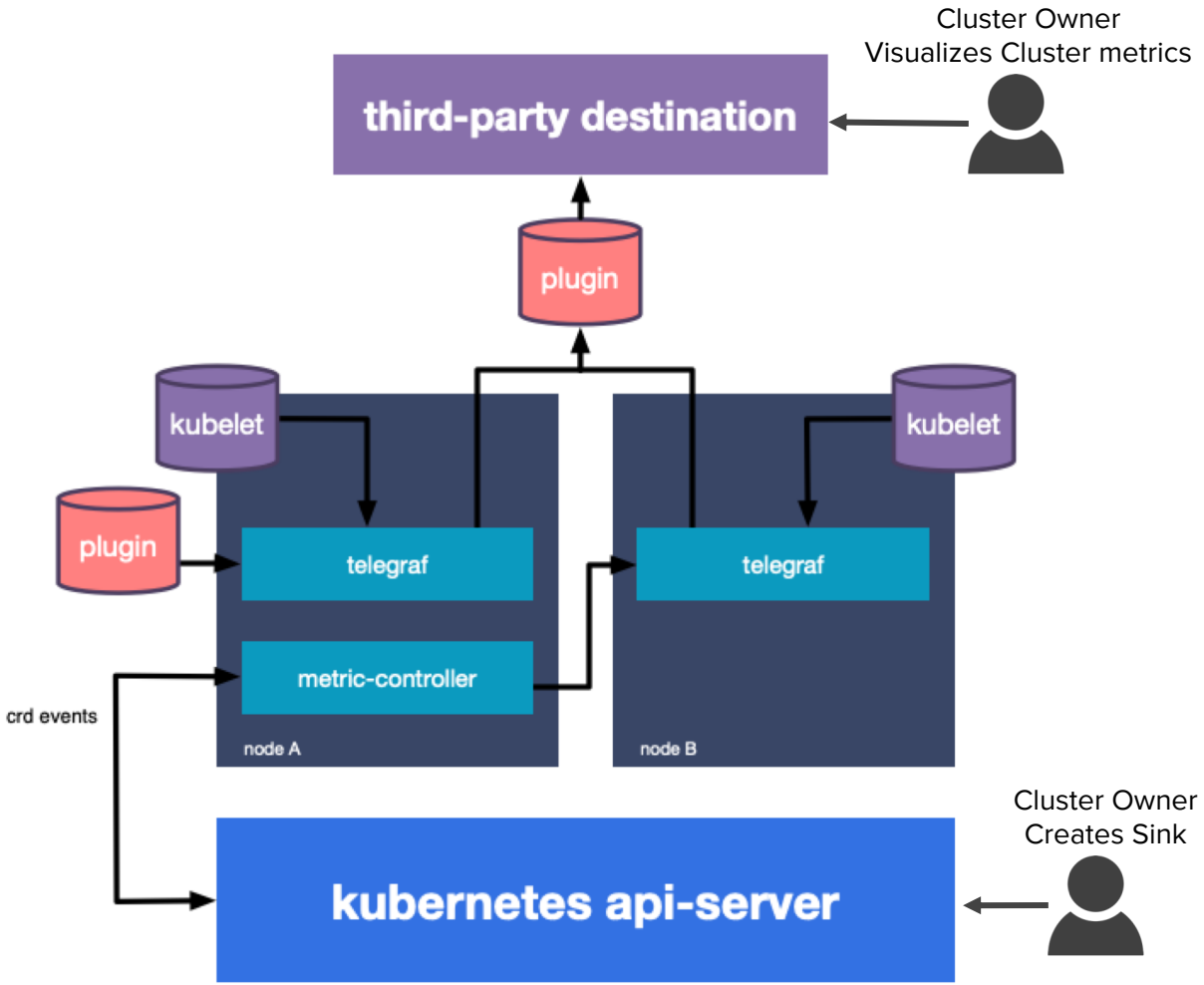


Pivotal

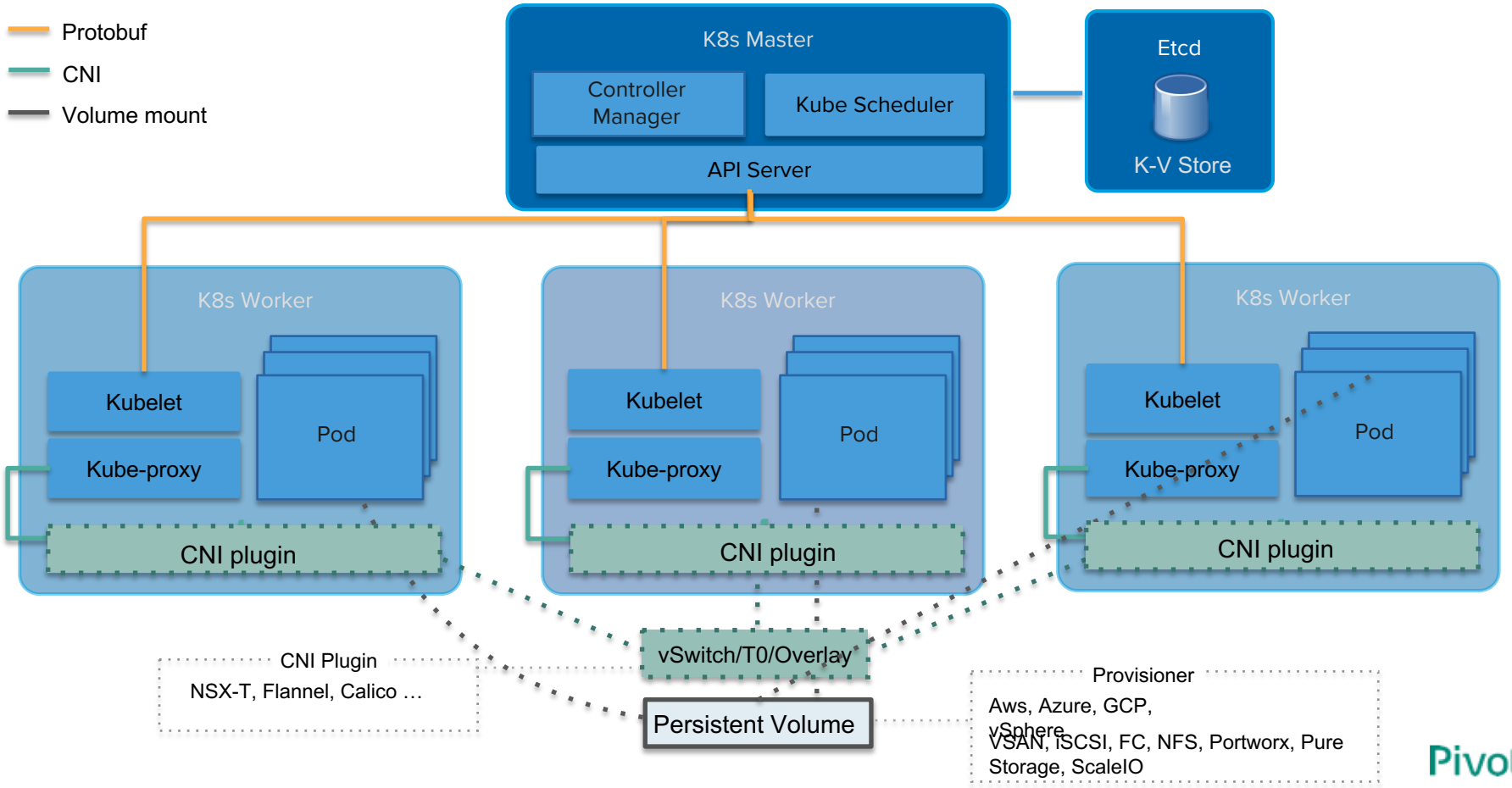
Operations and Observability: Metrics Sinks

Metric Sink collects and writes metrics from a cluster to specified outputs

Supports Telegraf input/output plugins

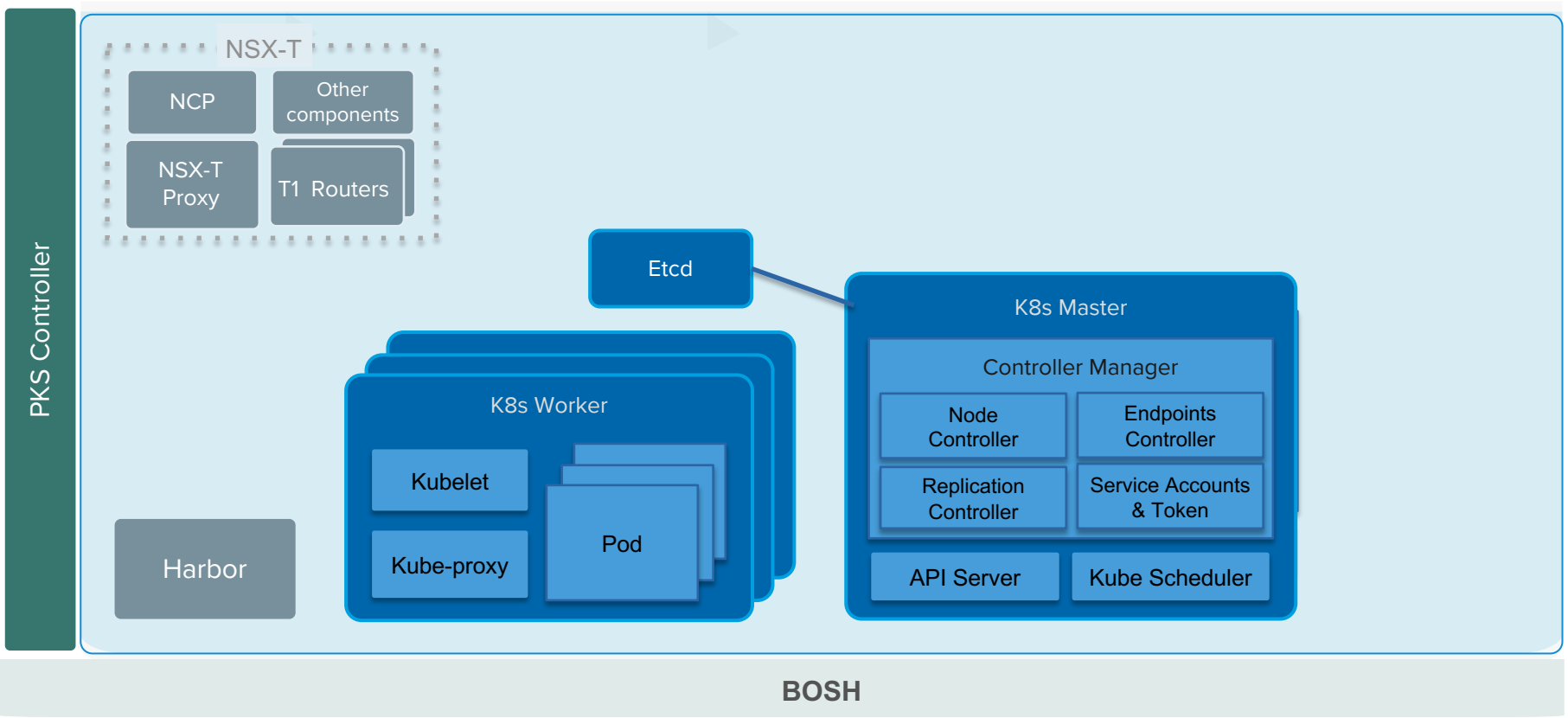


Kubernetes high-level components



Pivotal

Pivotal Container Service

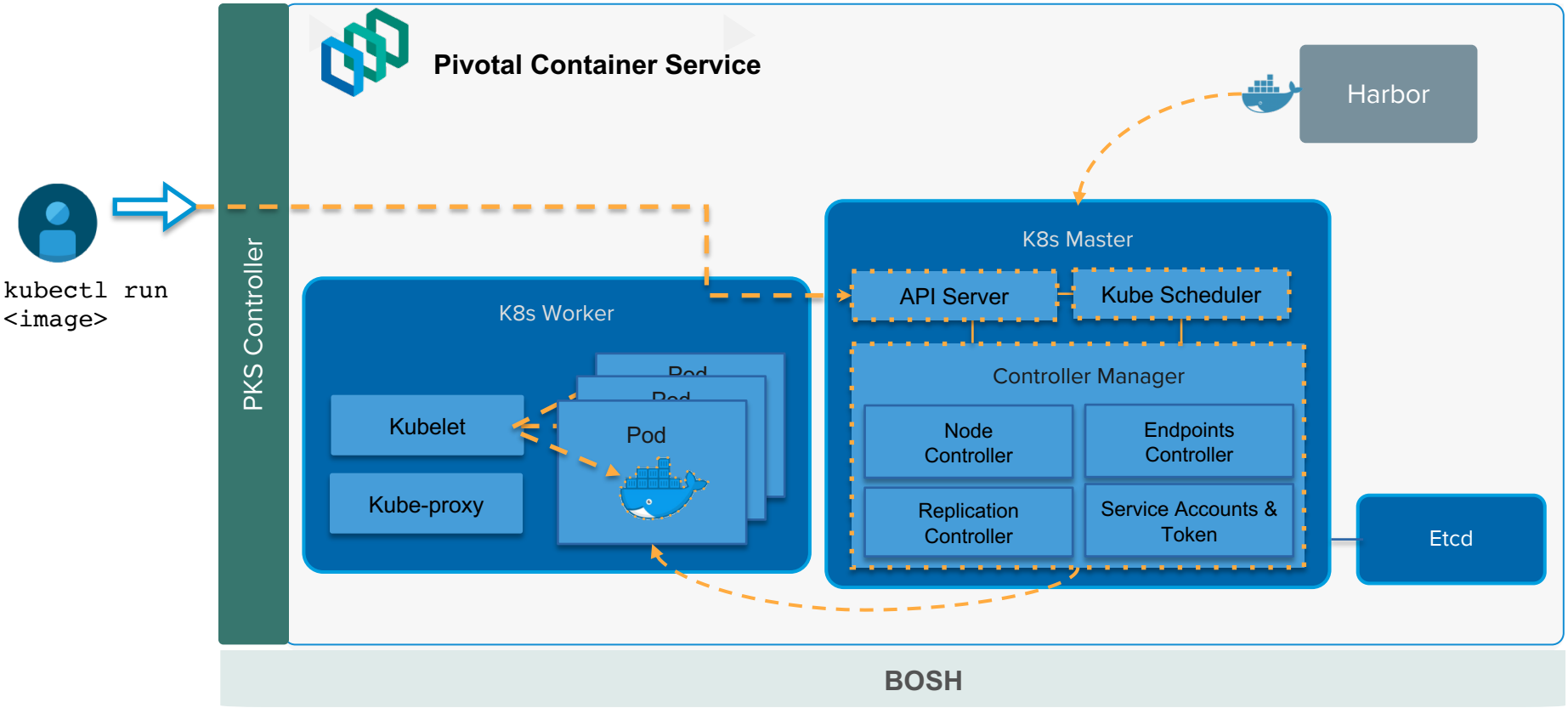


Included component, but optional usage

OSS Kubo

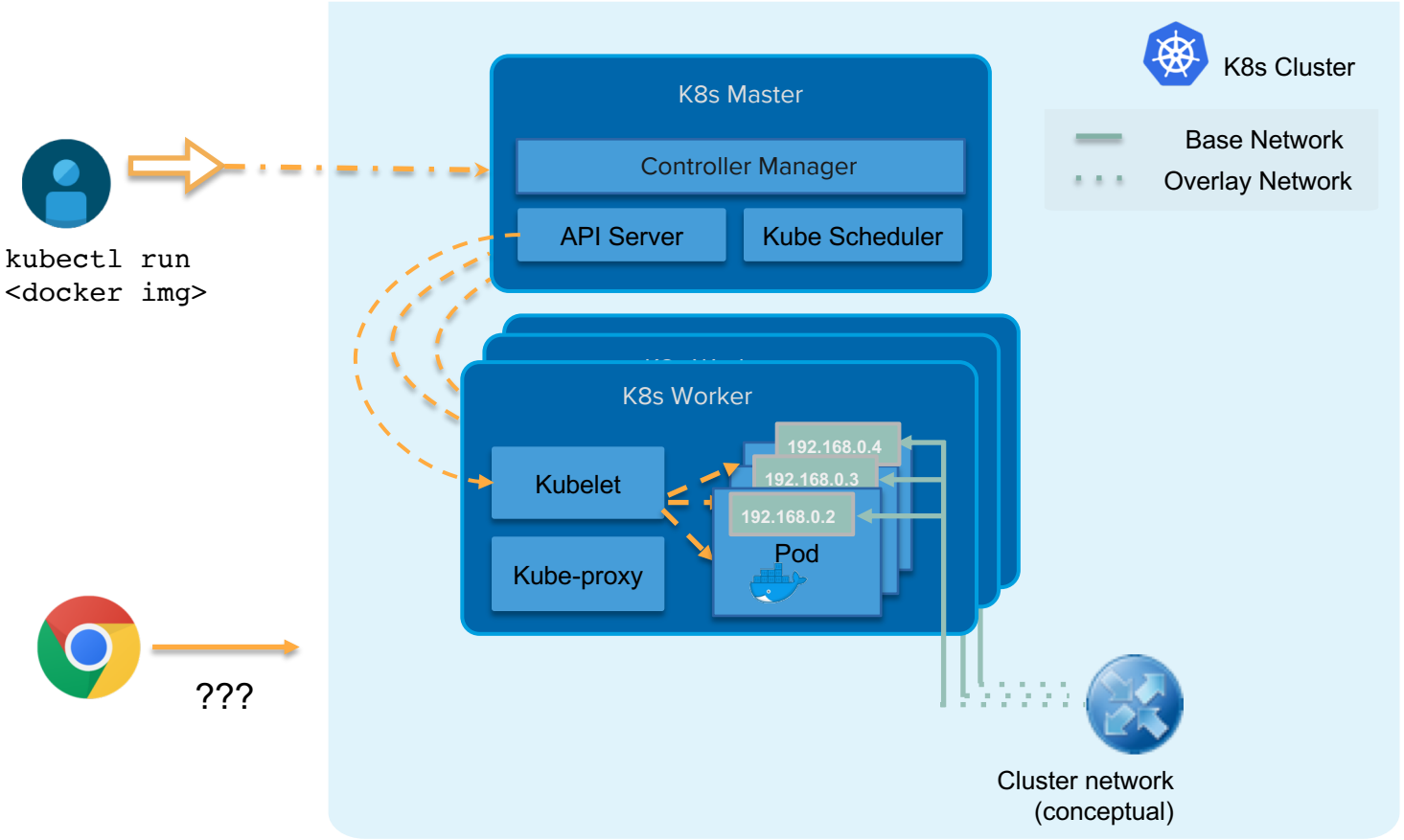
Pivotal

Pushing a Container

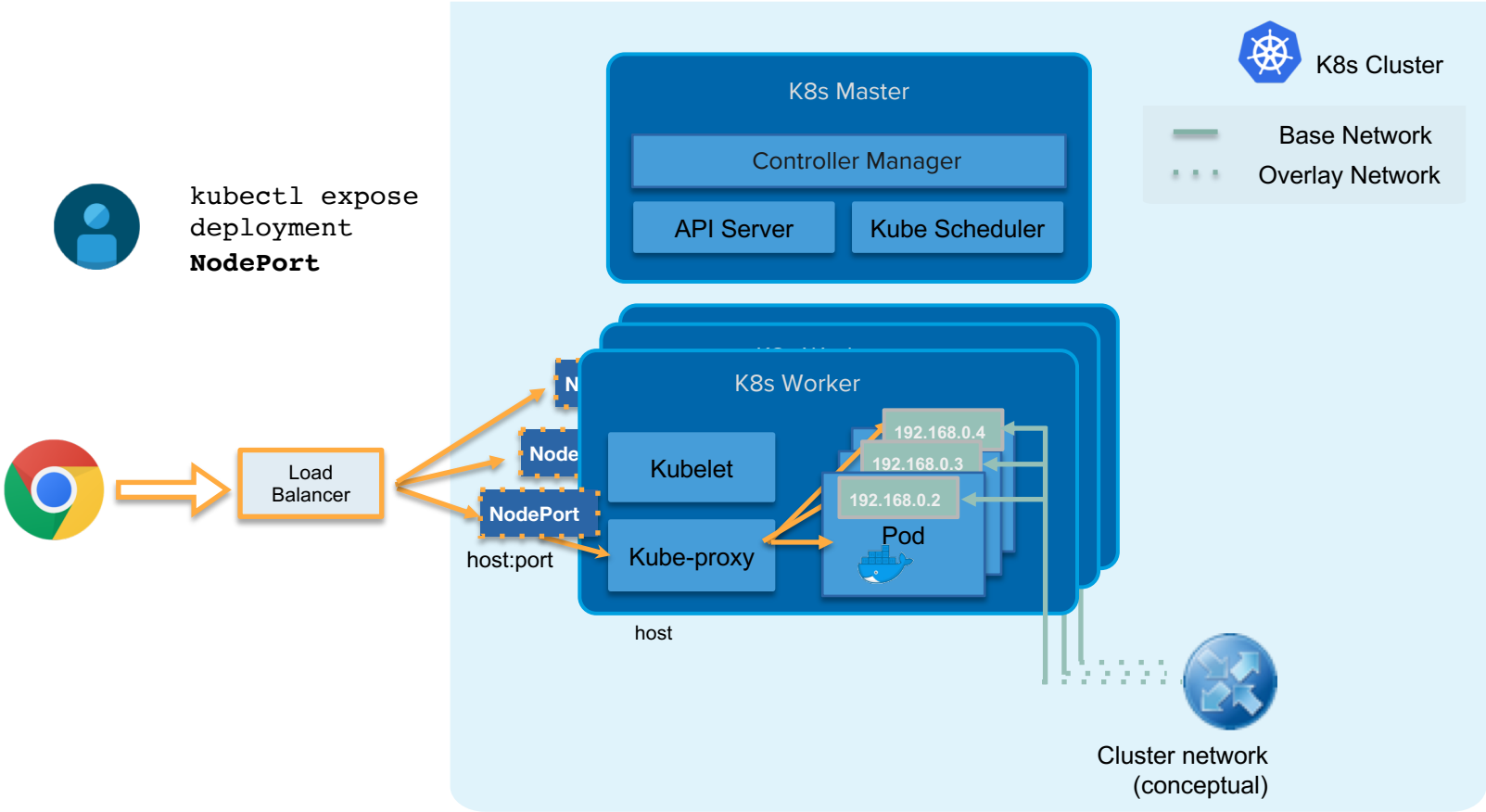


Pivotal

Services and Routing

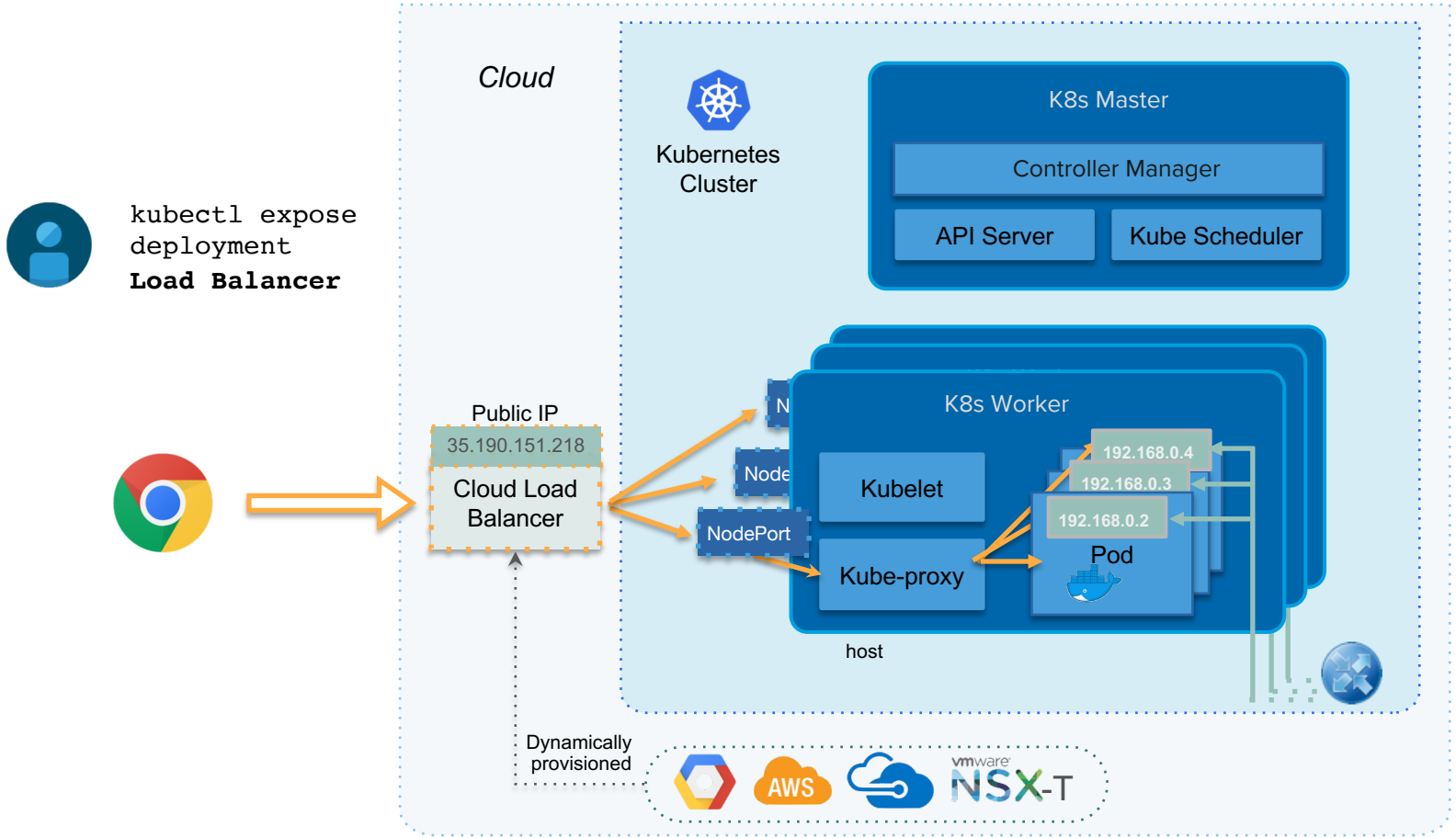


Services: NodePort



Pivotal

Services: Load Balancer



Pivotal

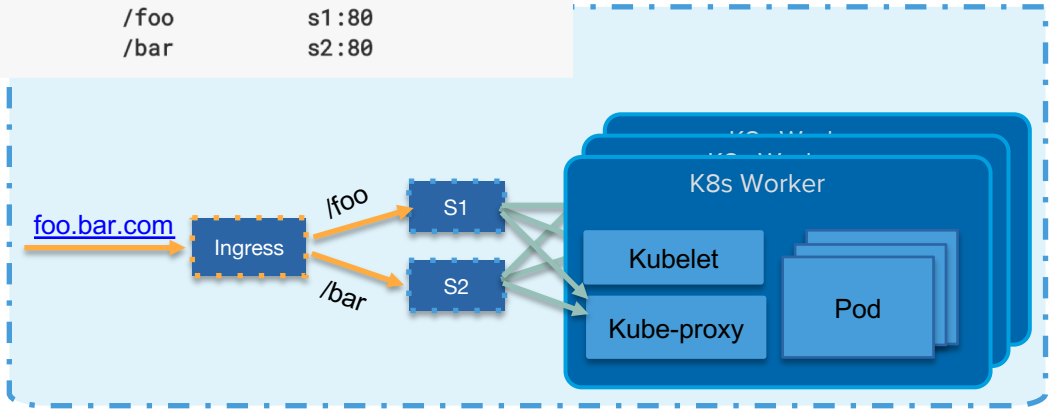
Services: Ingress

An API object that manages external access to the services in a cluster, typically HTTP.

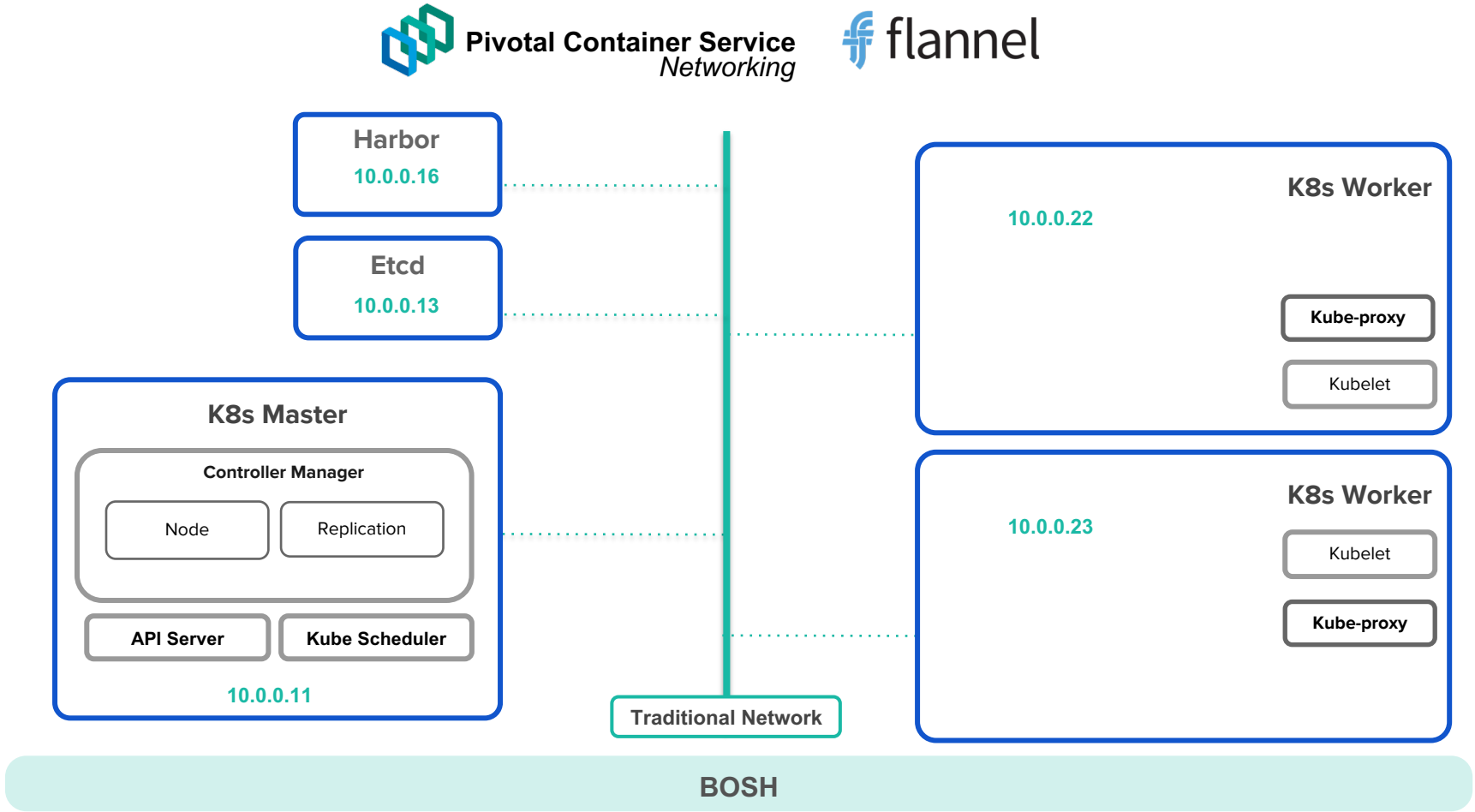
Ingress can provide **load balancing**, **SSL termination and name-based virtual hosting**.

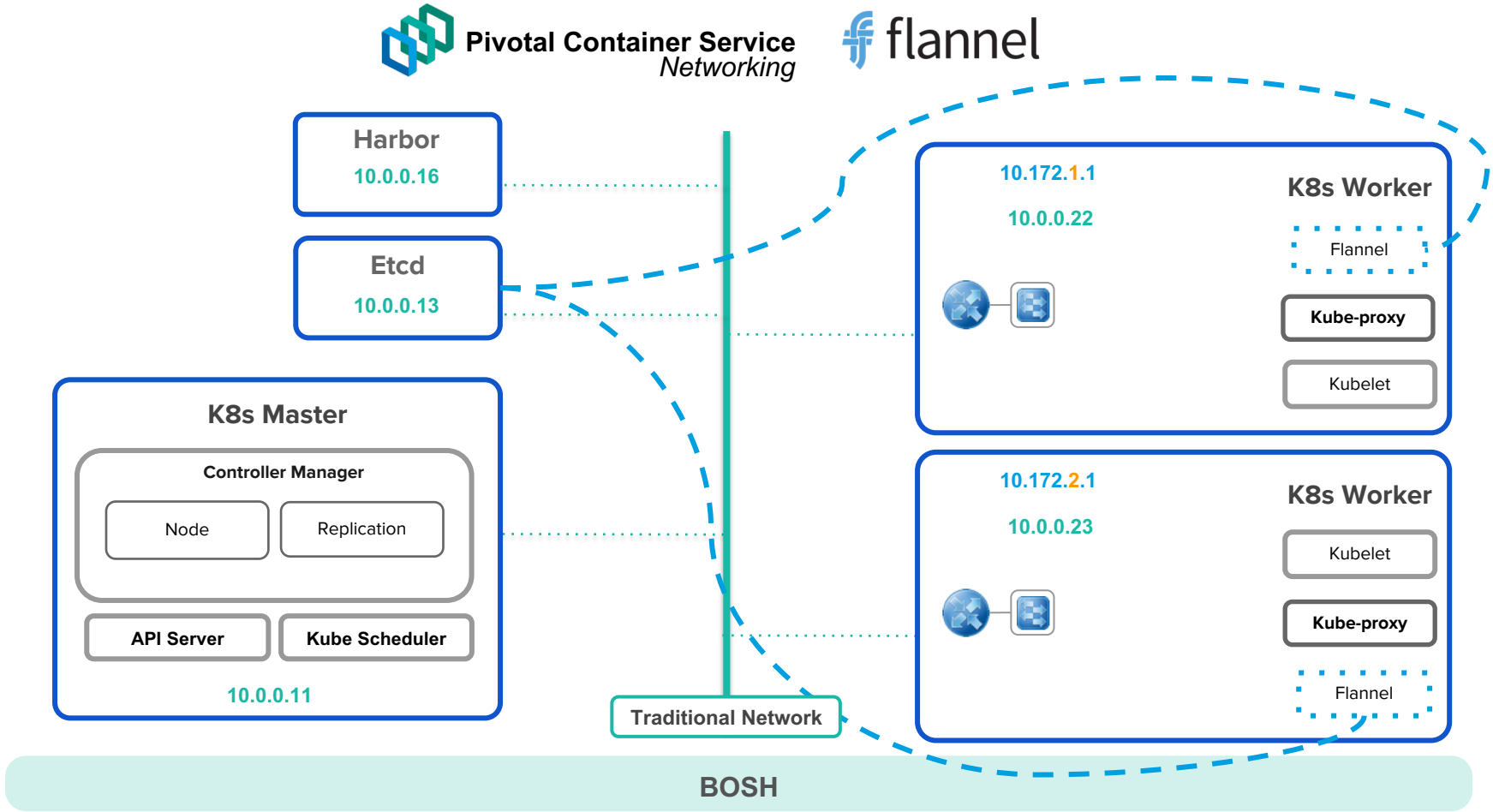
```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
  annotations:
    ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: s1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: s2
          servicePort: 80
```


```
$ kubectl get ing
NAME      RULE          BACKEND   ADDRESS
test      -
          foo.bar.com
          /foo          s1:80
          /bar          s2:80
```

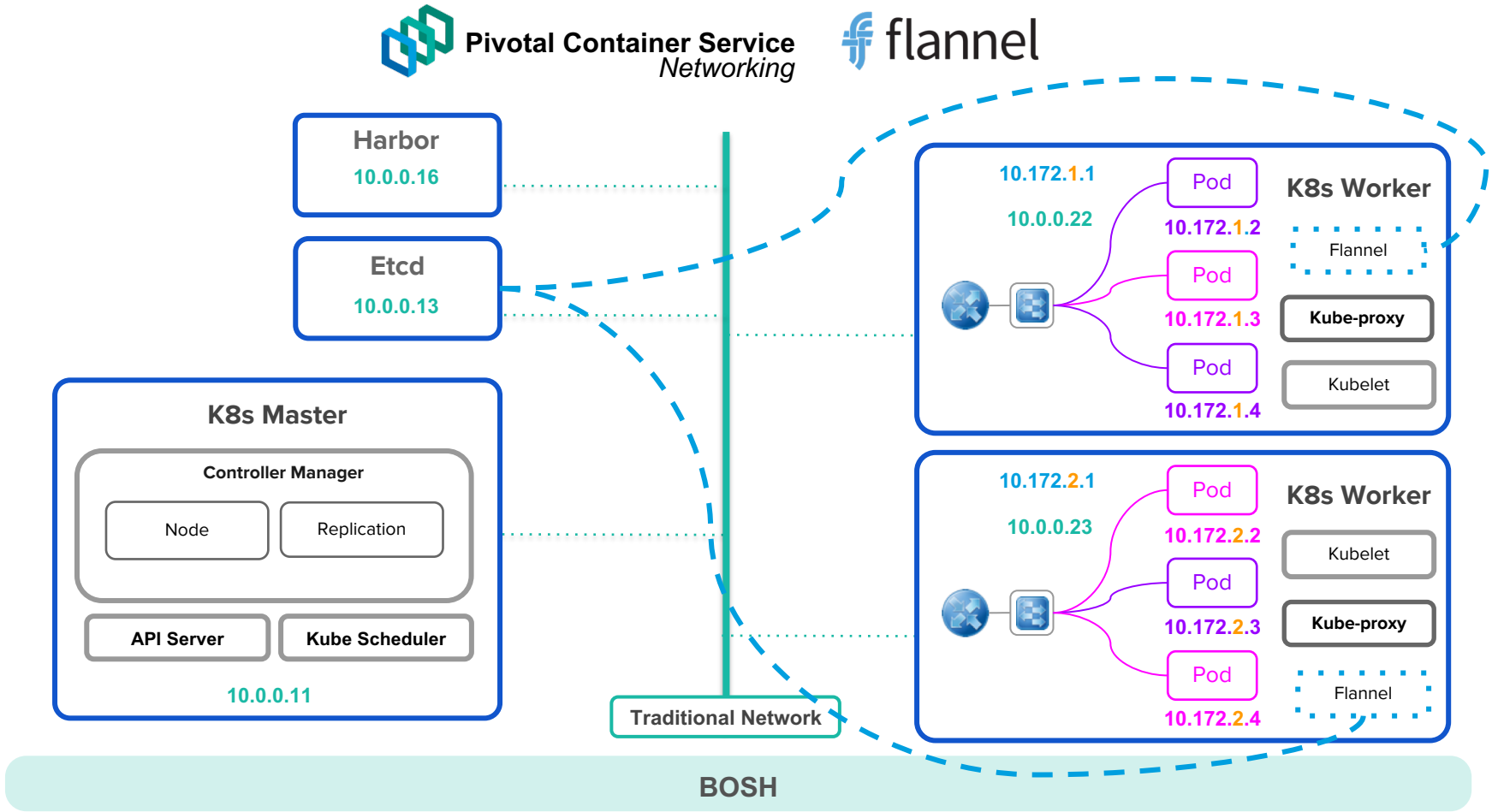


Pivotal

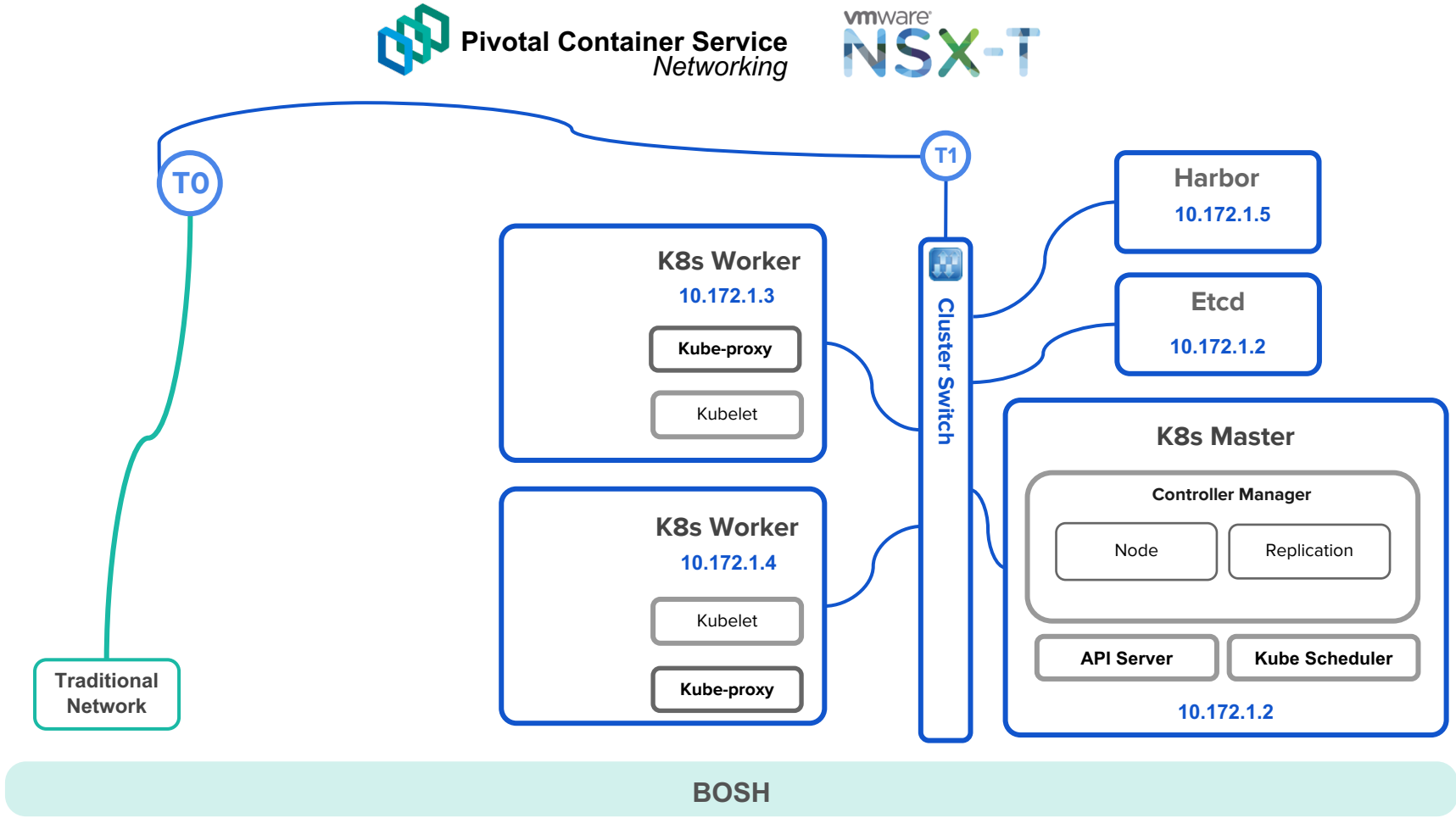

Pivotal Container Service
Networking
flannel
Harbor
10.0.0.16
Etcd
10.0.0.13
K8s Master
Controller Manager
Node
Replication
API Server
Kube Scheduler
10.0.0.11
Traditional Network
K8s Worker
10.0.0.22
Kube-proxy
Kubelet
K8s Worker
10.0.0.23
Kubelet
Kube-proxy
BOSH


Pivotal Container Service
Networking
flannel
Harbor
10.0.0.16
Etcd
10.0.0.13
K8s Master
Controller Manager
Node
Replication
API Server
Kube Scheduler
10.0.0.11
Traditional Network
10.172.1.1
10.0.0.22
K8s Worker
Flannel
Kube-proxy
Kubelet
10.172.2.1
10.0.0.23
K8s Worker
Kubelet
Kube-proxy
Flannel
BOSH

Pivotal


Pivotal Container Service
Networking
flannel
Harbor
10.0.0.16
Etcd
10.0.0.13
K8s Master
Controller Manager
Node
Replication
API Server
Kube Scheduler
10.0.0.11
Traditional Network
10.172.1.1
10.0.0.22
Pod
10.172.1.2
Pod
10.172.1.3
Pod
10.172.1.4
K8s Worker
Flannel
Kube-proxy
Kubelet
10.172.2.1
10.0.0.23
Pod
10.172.2.2
Pod
10.172.2.3
Pod
10.172.2.4
K8s Worker
Kubelet
Kube-proxy
Flannel
BOSH

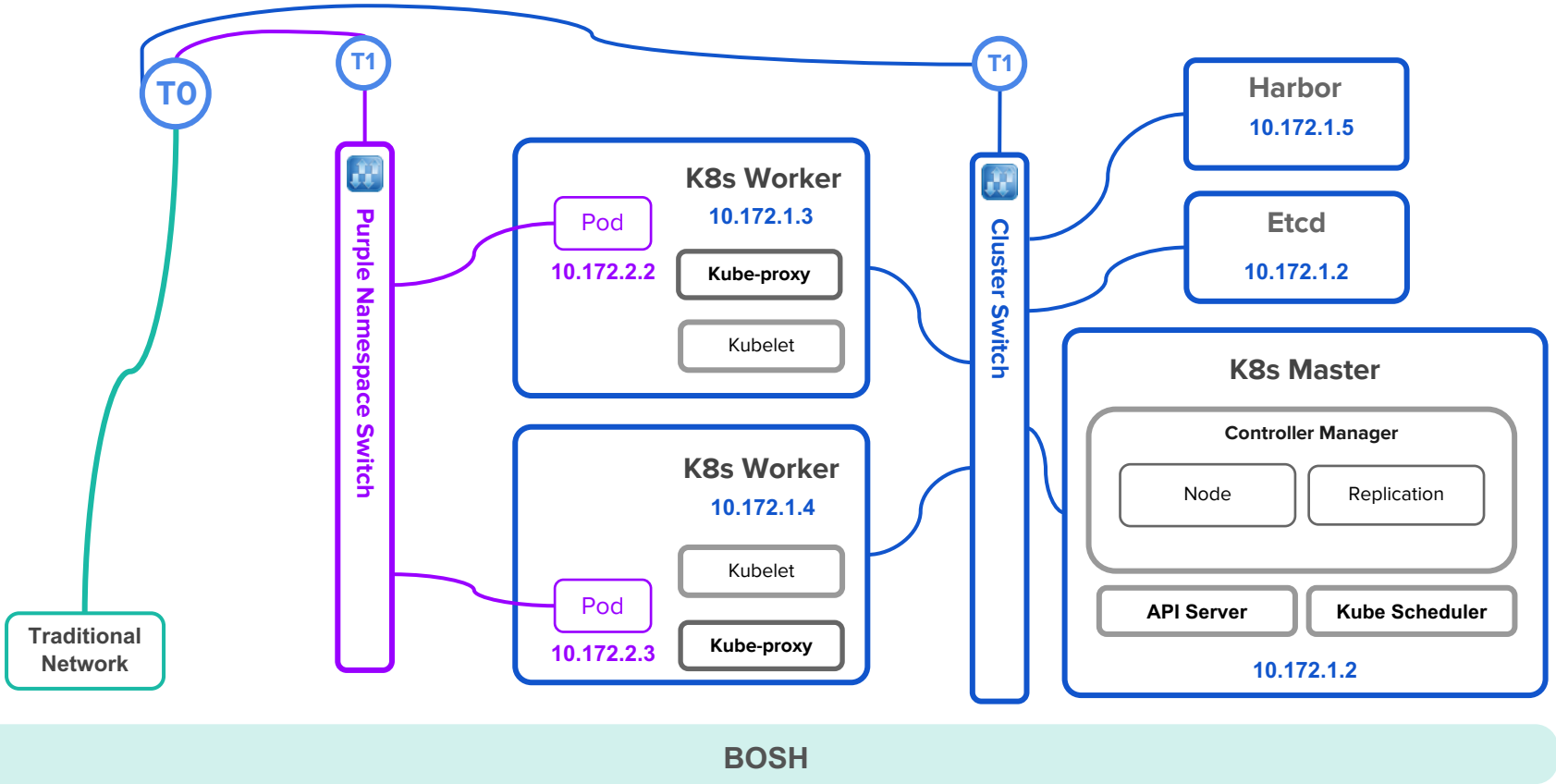
Pivotal


Pivotal Container Service
Networking
vmware
NSX-T
T0
T1
Traditional Network
K8s Worker
10.172.1.3
Kube-proxy
Kubelet
K8s Worker
10.172.1.4
Kubelet
Kube-proxy
Cluster Switch
Harbor
10.172.1.5
Etcd
10.172.1.2
K8s Master
Controller Manager
Node
Replication
API Server
Kube Scheduler
10.172.1.2
BOSH

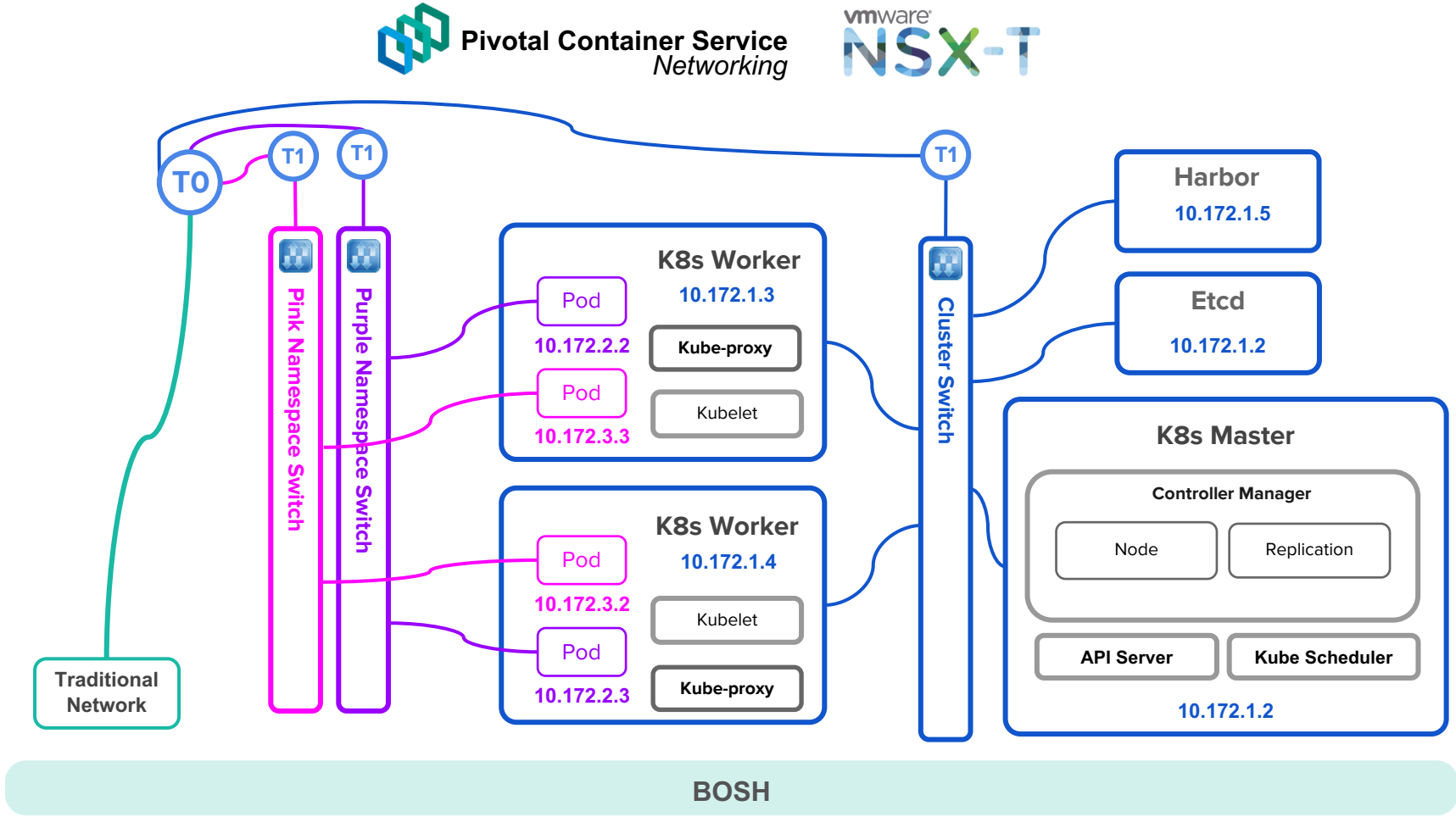
Pivotal

Pivotal Container Service
Networking

vmware NSX-T



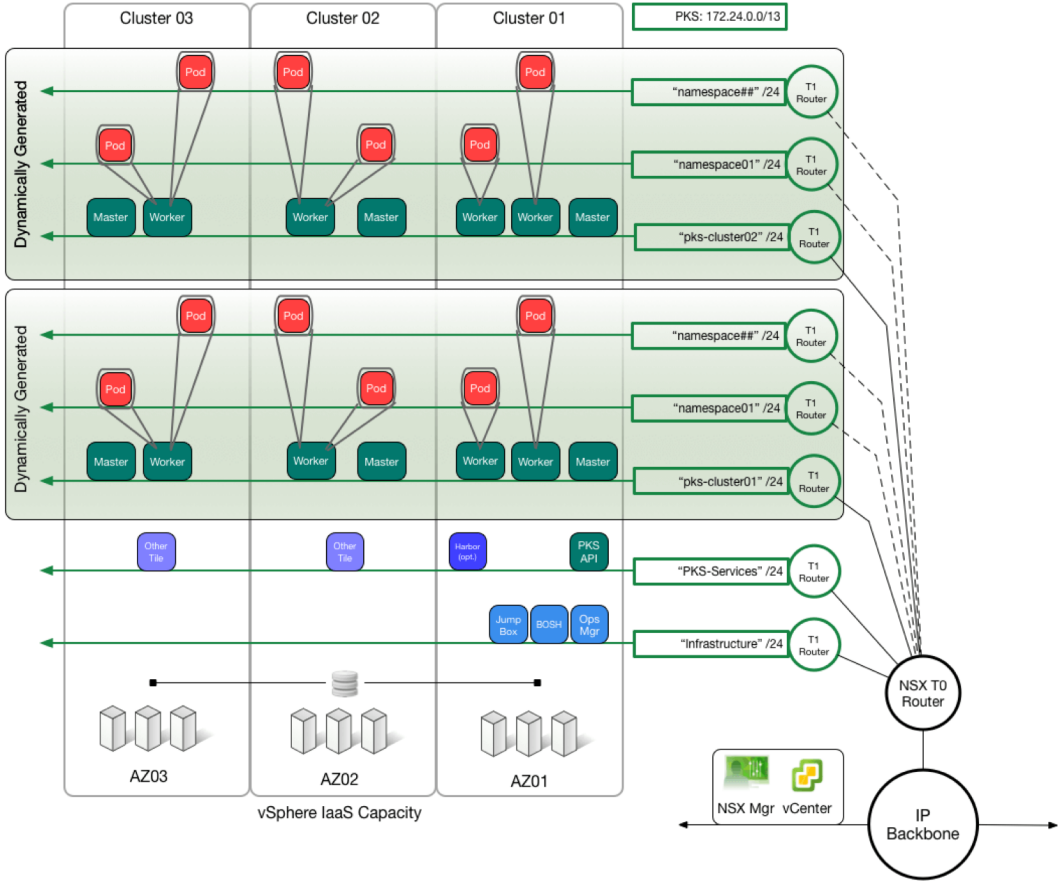
Pivotal


Pivotal Container Service
Networking
vmware
NSX-T
T0
T1
T1
T1
Pink Namespace Switch
Purple Namespace Switch
Cluster Switch
Traditional Network
K8s Worker
10.172.1.3
Pod
10.172.2.2
Pod
10.172.3.3
Kube-proxy
Kubelet
K8s Worker
10.172.1.4
Pod
10.172.3.2
Pod
10.172.2.3
Kubelet
Kube-proxy
Harbor
10.172.1.5
Etcd
10.172.1.2
K8s Master
Controller Manager
Node
Replication
API Server
Kube Scheduler
10.172.1.2
BOSH

Pivotal

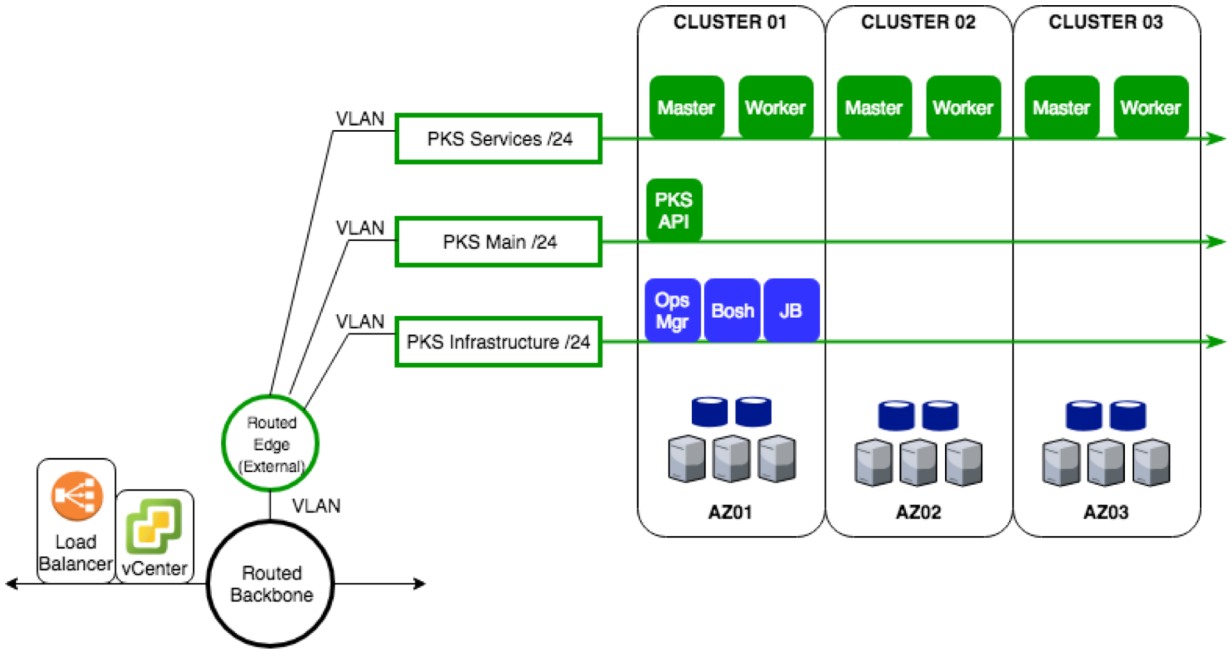
Reference Architectures

Reference Architecture vSphere w/ NSX-T

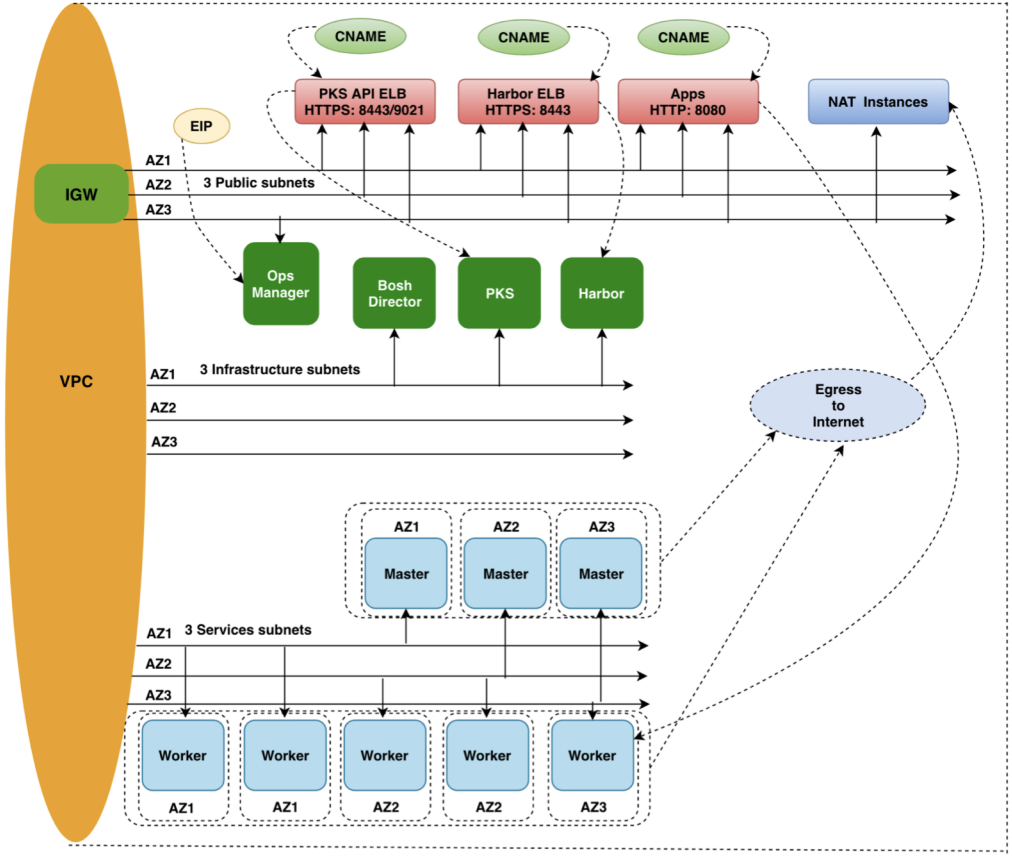


Pivotal

Reference Architecture vSphere w/o NSX-T



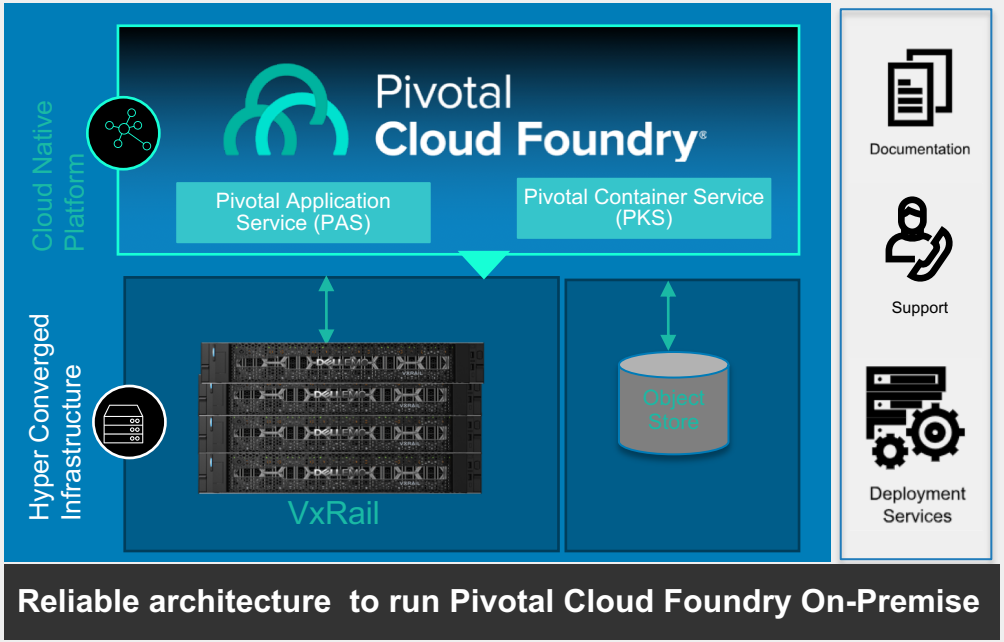
Reference Architecture AWS



Pivotal

Pivotal Ready Architecture

- Pivotal Cloud Foundry tested and **validated** on top of VxRail.
- Hyper-converged Infrastructure which provides robustness, ability to **scale** out and **ease of** lifecycle management
- Documentation and tools enables the **design** and **deploy** of the architecture **in a resilient and reliable way.**
- Product Support.
- Optional **service** can be leveraged by customers to deploy and manage the lifecycle of the Platform.



Reliable architecture to run Pivotal Cloud Foundry On-Premise

Pivotal vmware DELL EMC

Deeper Dive

Workload Placement

What workloads should run on PKS?

Stateful workloads

Commercial (packaged) software

Software distributed via Helm chart

Apps using non-standard port behavior

Legacy, zero-factor, apps

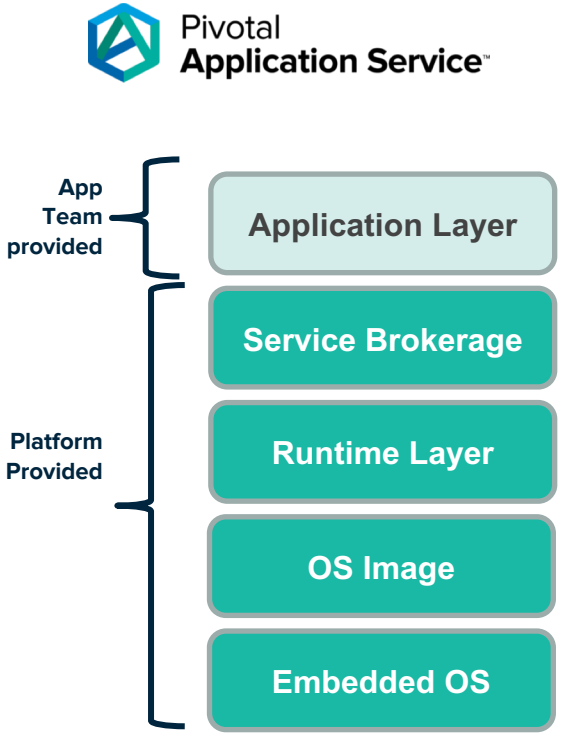
MongoDB, PostgreSQL, Cassandra, Spark, Elastic Search

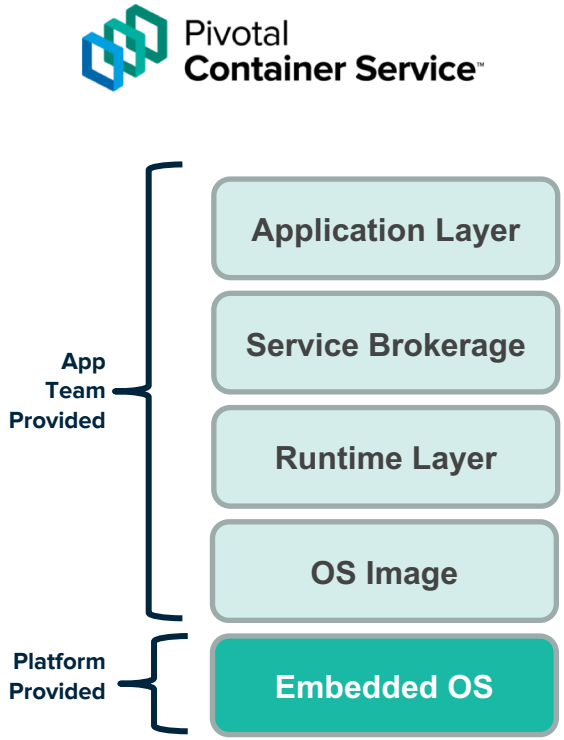
Microsoft SQL Server

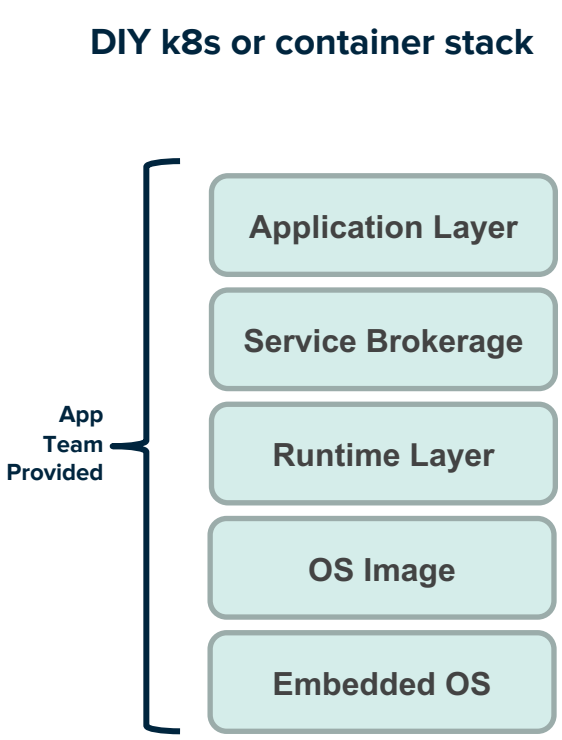
Apache Kafka

Custom-built C++ app listening on 3 ports

Decide what you want to “own” from security perspective







Comparing Spring Boot app deployment processes

Spring Boot app deployed to **PKS**

- Compile Spring Boot app
- Choose base Docker image
- Author Dockerfile w/ app entrypoint
- Build Docker image
- Upload image to container registry
- Decide on JVM tuning parameters to use when starting pods
- Create kubernetes deployment config
- Use CI/CD tool or kubectl to apply kubernetes configuration and deploy pods
- Create service to expose pod for users
- Profit!

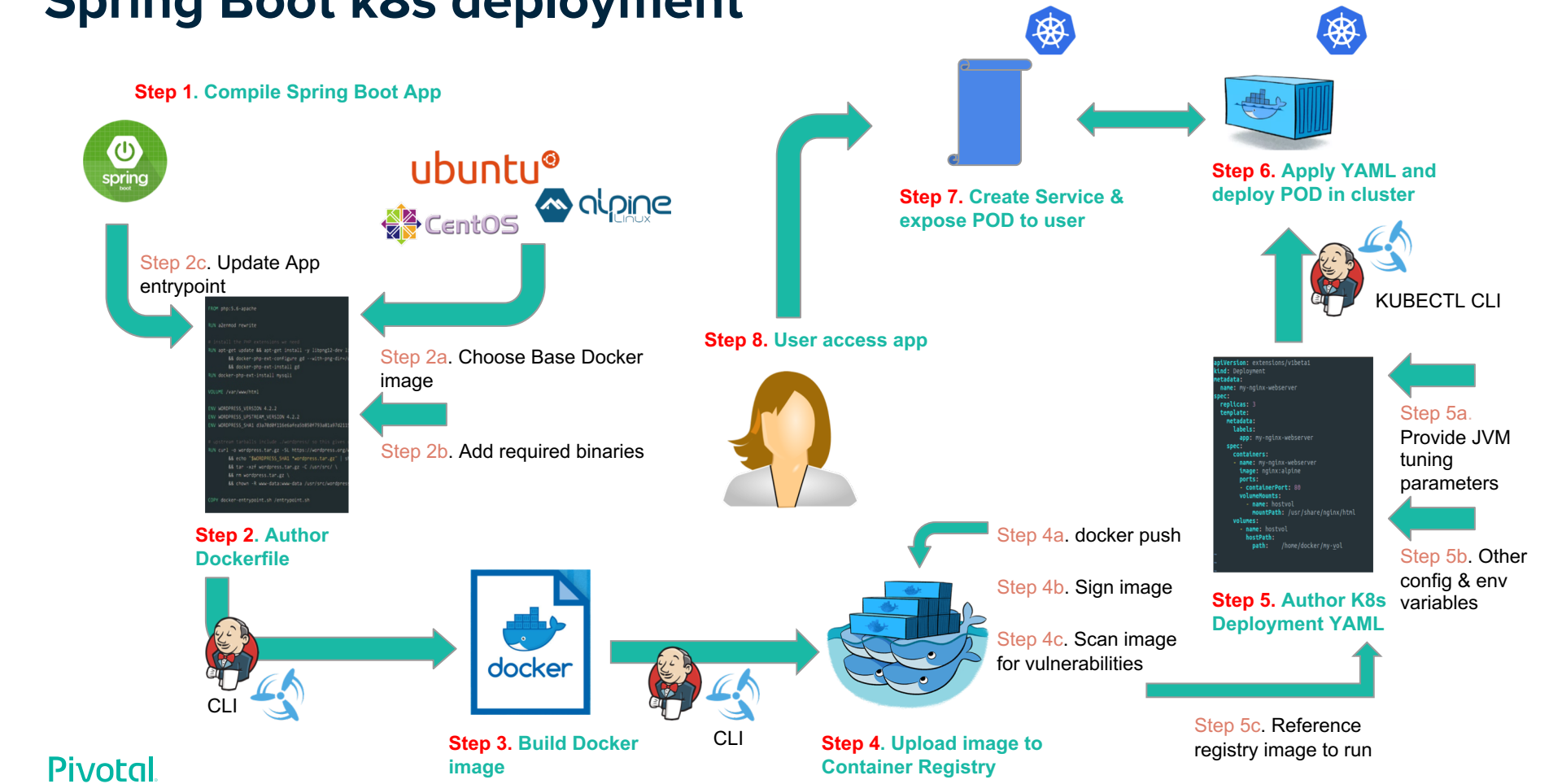
Snowflake container images
Governance enforced by corporate process

Spring Boot app deployed to **PAS**

- Compile Spring Boot app
- Create manifest.yml to describe the app
- Use CI/CD tool or `cf push` to deploy
- Profit!

Standardized container images
Governance enforced by the platform

Spring Boot k8s deployment



Pivotal

Spring Boot PKS deployment

Step 1. Compile Spring Boot App

Step 2c. Update App entrypoint

Step 2a. Choose Base Docker image

Step 2b. Add required binaries

Step 2. Author Dockerfile

CLI

Step 3. Build Docker image

CLI

Step 4. Upload image to Harbor

Step 5. Author K8s Deployment YAML

Step 5a. Provide JVM tuning parameters

Step 5b. Other config & env variables

Step 5c. Reference registry image to run

Latest k8s version
Latest k8s features

KUBECTL CLI

Step 6. Apply YAML and deploy POD in cluster

Step 7. Create Service & expose POD to user

Step 8. User access app

Pivotal

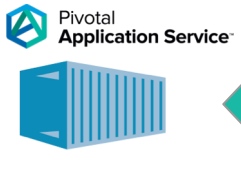
Spring Boot PAS deployment

Step 1. Compile App

Step 2a. Update App details

Step 2. Author Manifest file

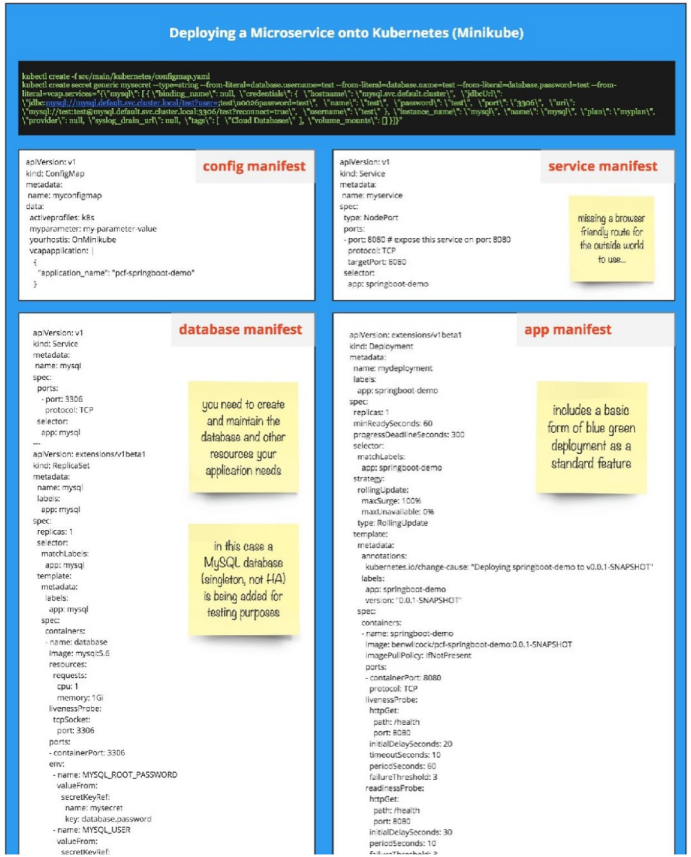
Step 3. User access app



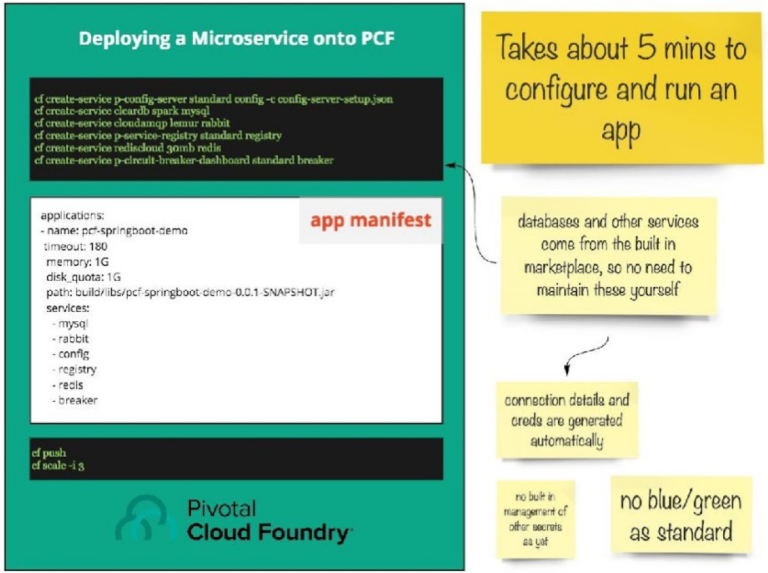
cf push

Pivotal

Comparing Spring Boot app deployment processes



Takes about an 2hrs to configure and run an app



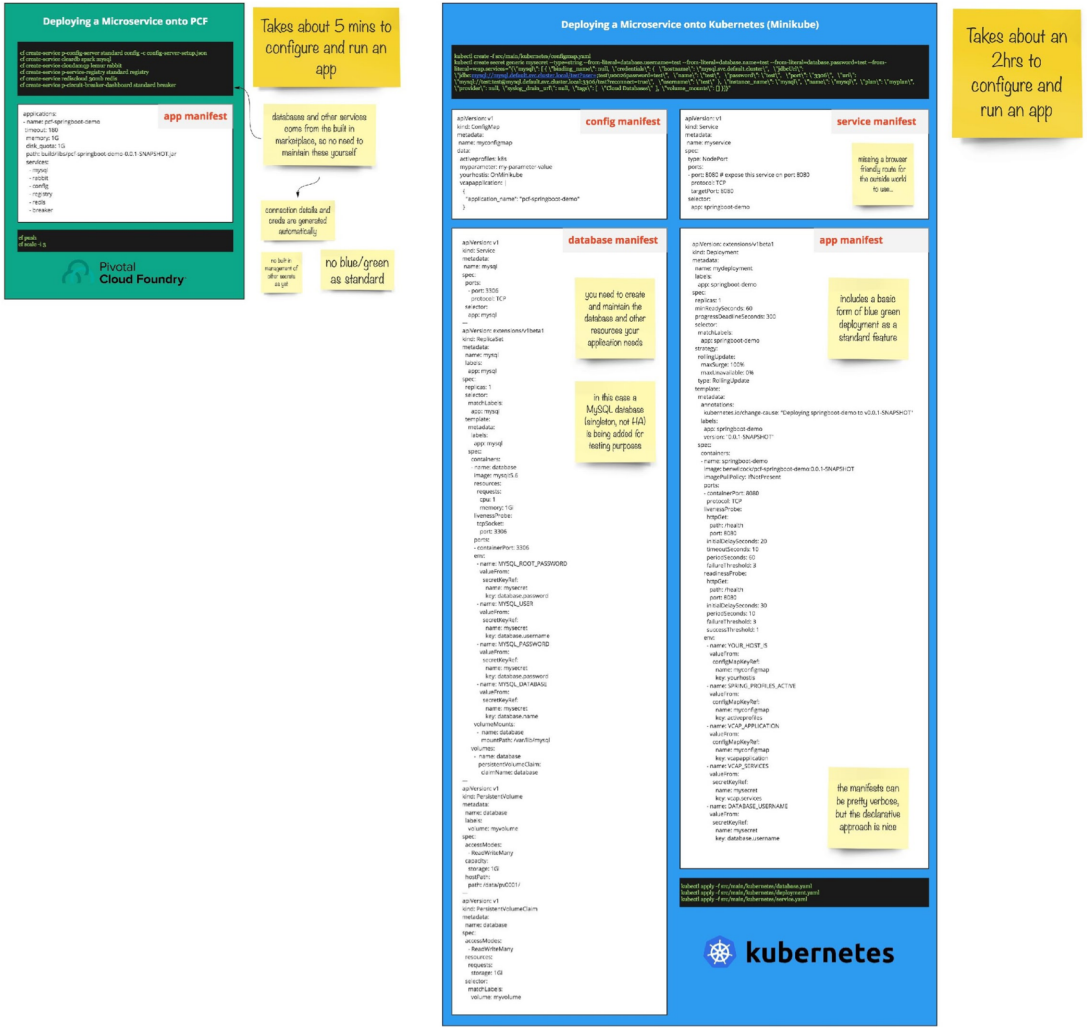
Takes about 5 mins to configure and run an app

databases and other services come from the built in marketplace, so no need to maintain these yourself

connection details and creds are generated automatically

no built in management of other secrets as yet

no blue/green as standard


Deploying a Microservice onto PCF
app manifest
Pivotal Cloud Foundry
Takes about 5 mins to configure and run an app
databases and other services come from the built in marketplace, so no need to maintain these yourself
connection details and creds are generated automatically
no built-in management of offer secrets as yet
no blue/green as standard
Deploying a Microservice onto Kubernetes (Minikube)
config manifest
service manifest
missing a browser friendly route for the outside world to use...
database manifest
you need to create and maintain the database and other resources your application needs
in this case a MySQL database (singleton, not HA) is being added for testing purposes
app manifest
includes a basic form of blue green deployment as a standard feature
the manifests can be pretty verbose, but the declarative approach is nice
kubernetes
Takes about an 2hrs to configure and run an app

Pivotal

Comparing PKS to PAS

Secure container registry (Harbor)	Secure container registry (Harbor or Blob Store)
Secure multi-tenant ingress	Secure multi-tenant ingress (Go Router)
Rolling upgrades to cluster infrastructure	Rolling upgrades to cluster infrastructure
Monitoring and recovery of cluster VMs and processes	Monitoring and recovery of cluster VMs and processes
Cluster provisioning & scaling	Cluster provisioning and scaling
Embedded, hardened Operating System	Embedded, hardened Operating System
Single tenant ingress	Single tenant ingress (Go Router + Isolation Segments)
Rolling upgrades to pods	Rolling upgrades to containers (blue/green deploy)
Pod scaling and high availability	Container scaling and high availability
Stateful Sets of pods	-
Multi-container pods	-
Persistent disks	Persistent disks (Volume Services)
	Native .NET and Windows support
	Built-in managed services for RabbitMQ, MySQL, PCC
	Spring Cloud Services
	Multi-tenancy through Orgs, Spaces, and Quotas
	IdM components for OAuth, SAML, LDAP
	Marketplace and secure service bindings
	Built in load balancing with flexible route bindings
	Metrics and aggregated logging
	Droplets automatically created by buildpacks
	Interceptable routes with Route Services

Transforming How The World Builds Software