QCon 全球软件开发大会
INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

BEIJING 2018

Kubernetes – Software Platform for the Future

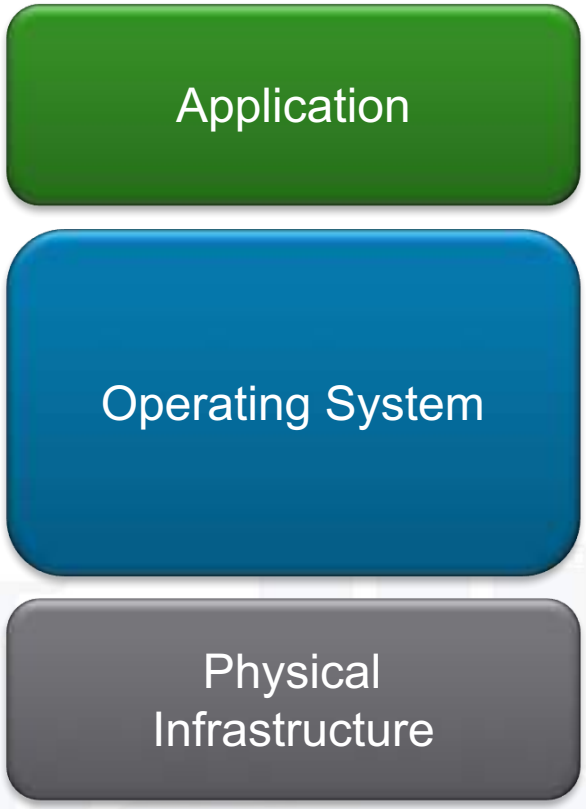
Michael Chen
VMware

主办方 Geekbang 极客邦科技 InfoQ

Developer

- Hello World!
- Containerize Applications
- Micro Services Applications
- Scalable Kubernetes Applications
- Scalable Infrastructure for Applications

Platform



Containers as Enabler

Fast

Boot Environments Rapidly

Lightweight

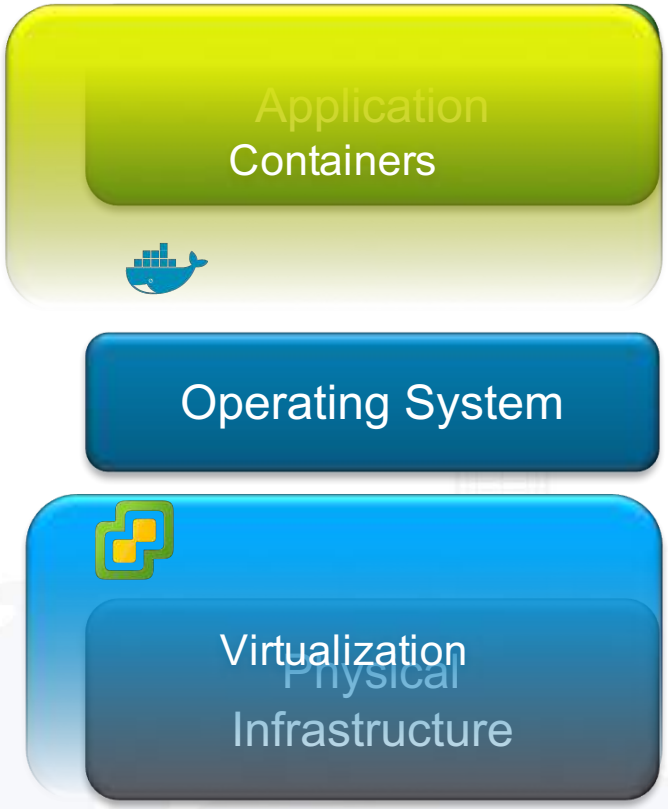
Minimal Resources Needed

Portable

Ability to Move Containers Freely

Containers and VMs - A Practical Comparison

The hypervisor virtualizes the hardware limiting the number of **hardware dependences** that you need to install on the OS

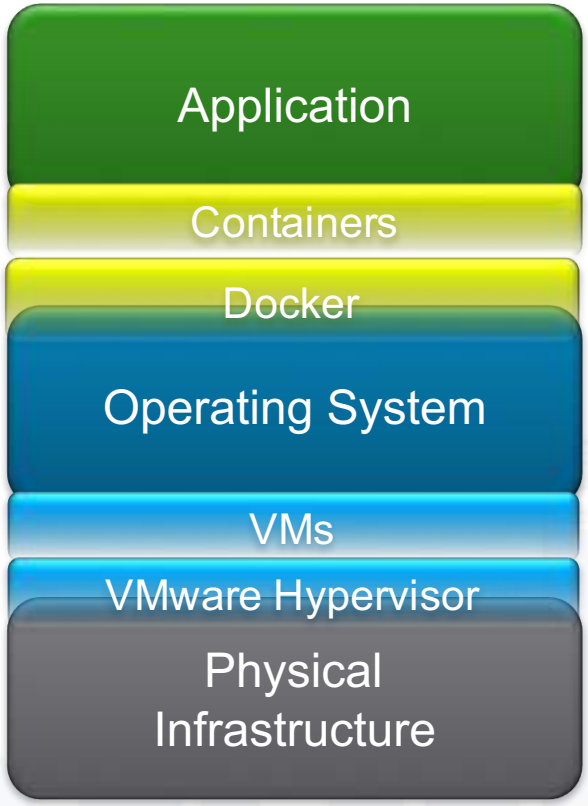


Containers virtualize the operating system limiting the the number of **application dependencies** that you need to install on the OS.

2 technologies with 2 different objectives

Abstract the App From the OS

Abstract the OS from the hardware



Allows you to run multiple applications on the same OS

Allows you to run multiple OS on the same hardware

Containers User Cases

Developer Sandbox

- Ready-to-go development
- Self-service portal

Application Repackaging

- Simplify app maintenance
- Improve developer workflow

Cloud Native

- New application development
- 12-factor apps, PCF

docker

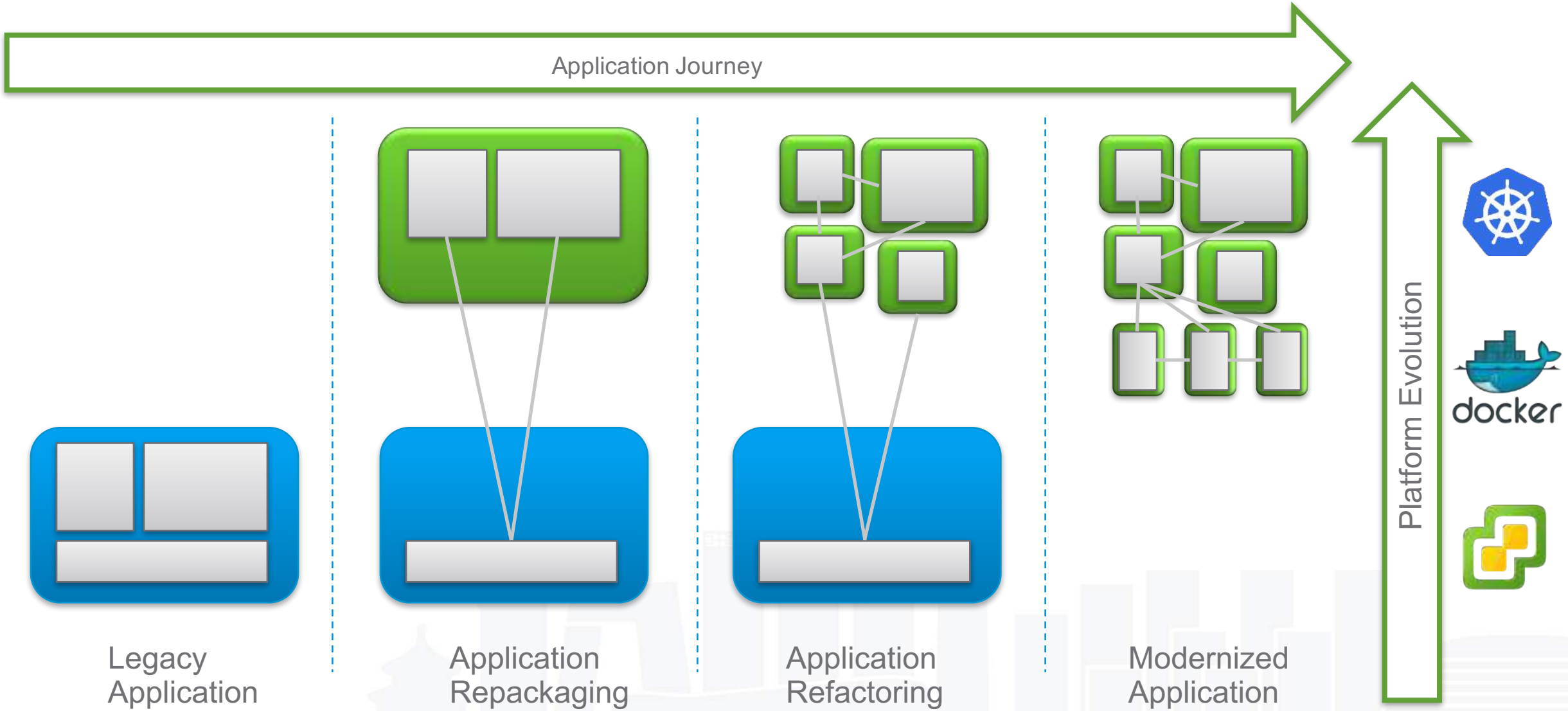
vSphere Integrated Containers

kubernetes

Pivotal Container Service

Pivotal Cloud Foundry

The need for containers and containers orchestrators



Docker and Kubernetes

Kubernetes in 5 min
https://youtu.be/PH-2FfFD2PU

Docker

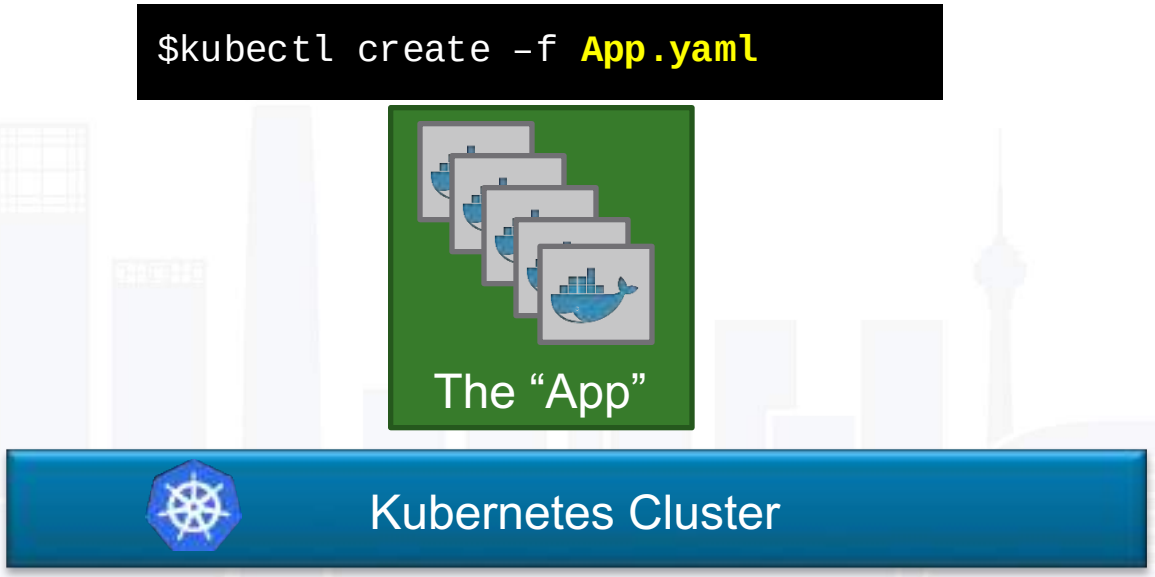
Run One Container at a Time

- Core **docker** functionality provides the tooling to create and run single containers
  - Very manual, no fault tolerance, hard to scale, etc

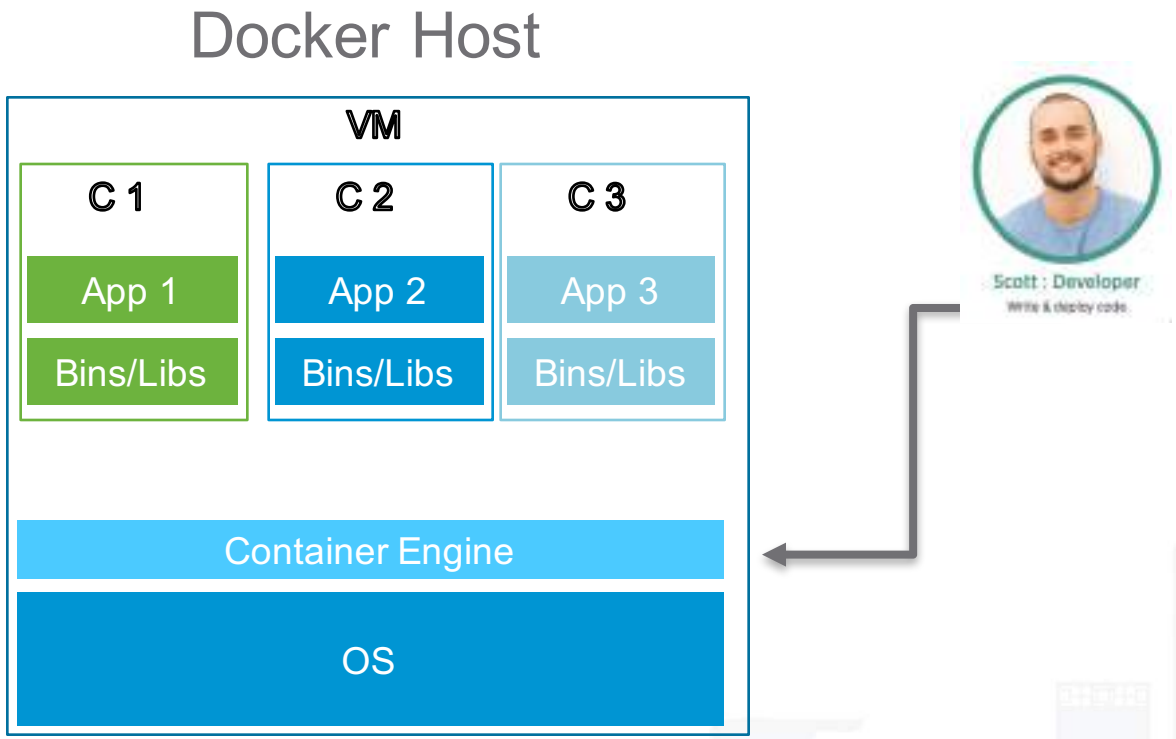
Kubernetes

Orchestrating Multiple Containers

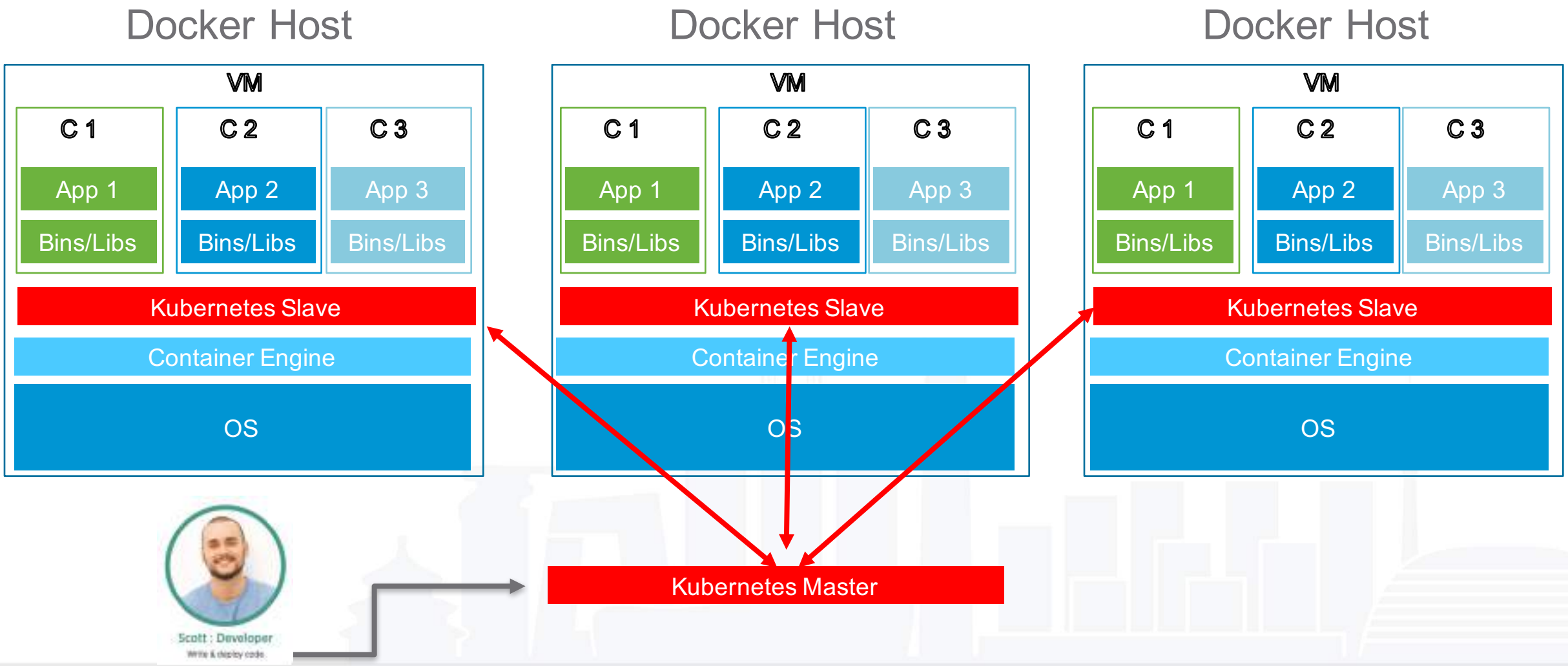
- Scheduling, provisioning, and resource management of multiple containers
  - Docker, Mesos → Kubernetes Support
  - AWS, Azure, Google → Kubernetes Services



What is Kubernetes?



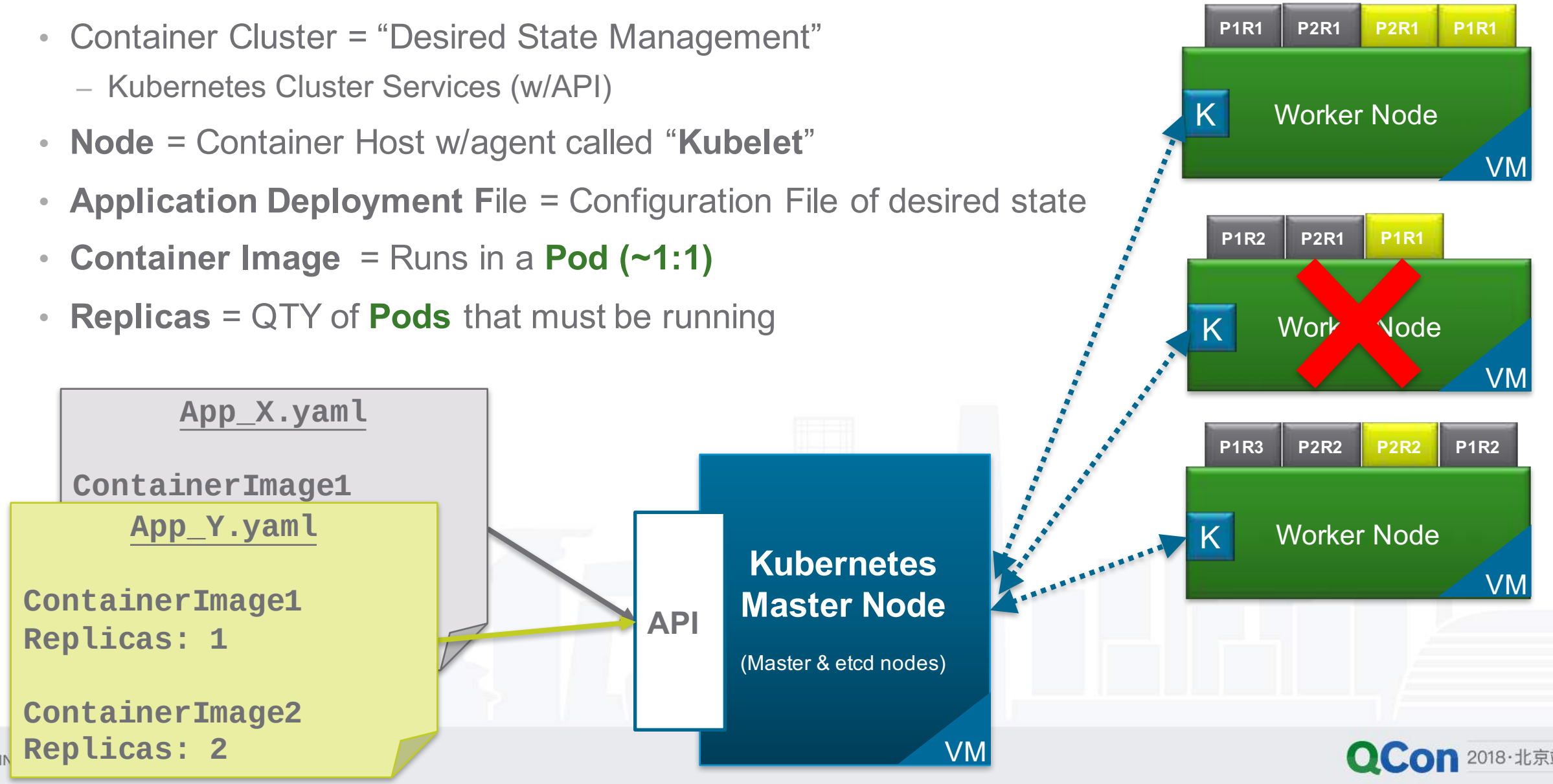
What is Kubernetes?



https://youtu.be/PH-2FfFD2PU

Kubernetes 101 at the Highest Level

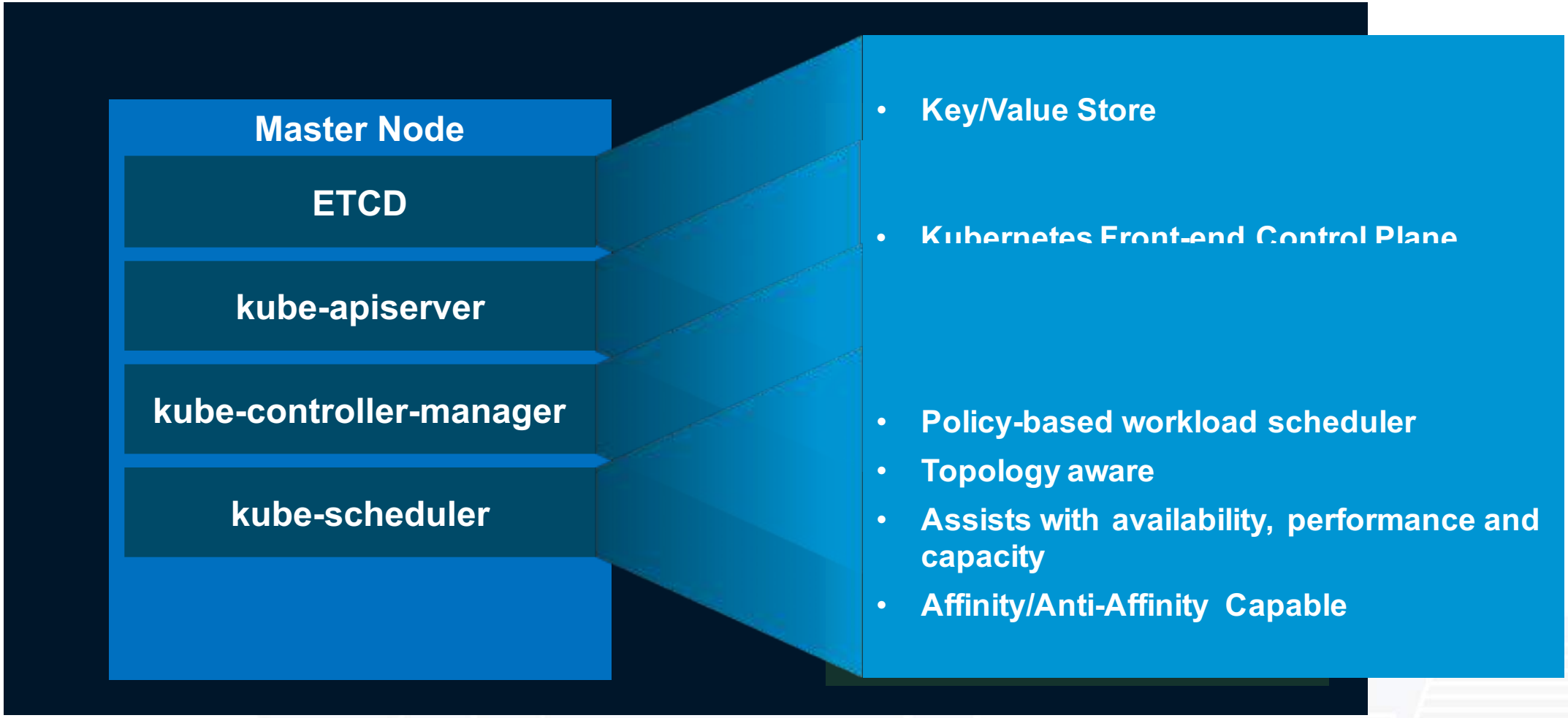
- Container Cluster = “Desired State Management”
  - Kubernetes Cluster Services (w/API)
- **Node** = Container Host w/agent called “**Kubelet**”
- **Application Deployment File** = Configuration File of desired state
- **Container Image** = Runs in a **Pod (~1:1)**
- **Replicas** = QTY of **Pods** that must be running



Kubernetes Trend

The Kubernetes Master Node

Basic Components

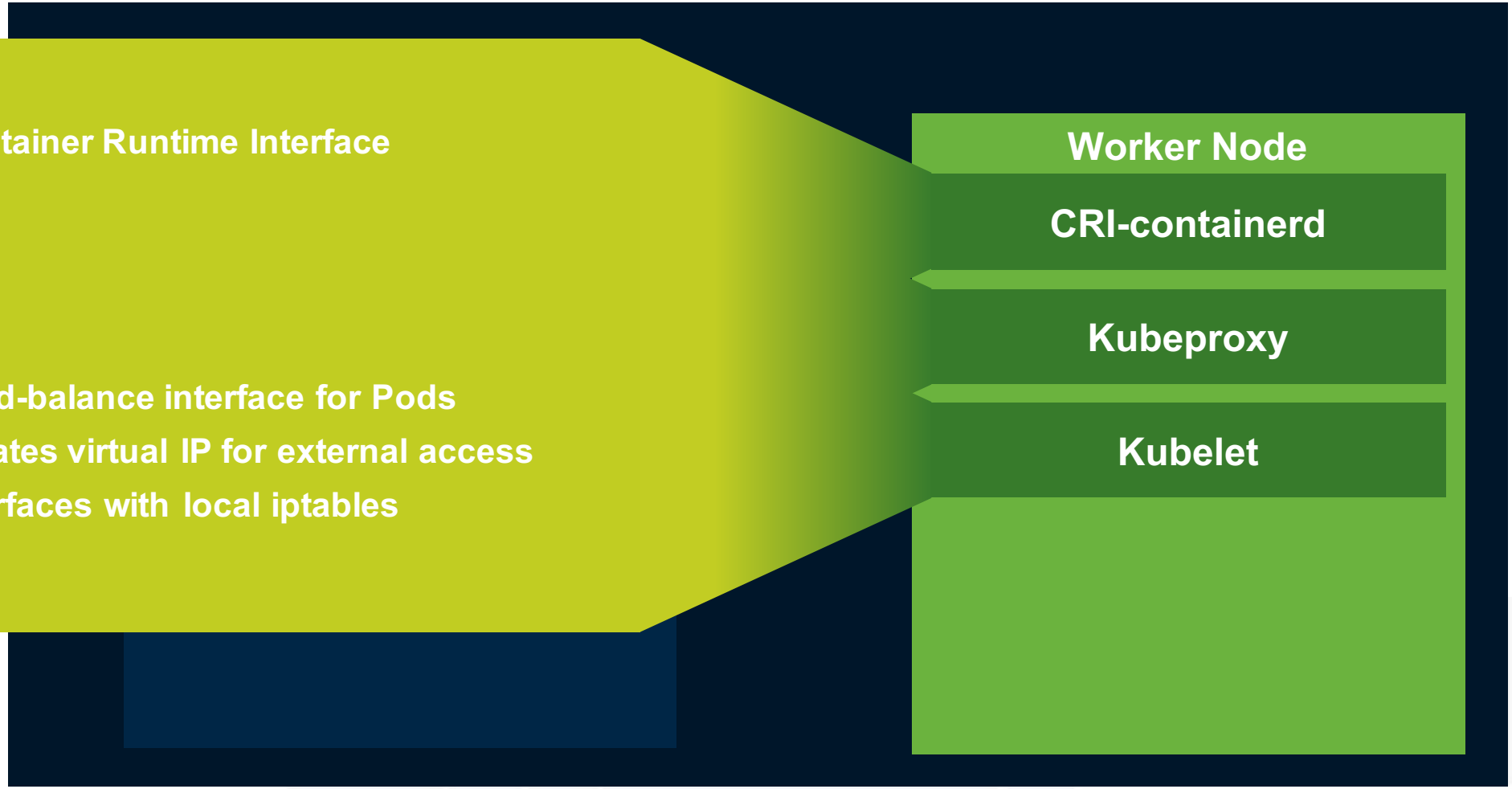


The Kubernetes Worker Node

Basic Components

- Container Runtime Interface

- Load-balance interface for Pods
- Creates virtual IP for external access
- Interfaces with local iptables



The Kubernetes Runtime Components

Component	Description
Pods	A grouping of one or more containers as an atomic unit
Namespaces	A way to organize items in a cluster
Labels, Annotations & Selectors	Tags for component grouping and methods to access them
Service Discovery	An object associated to a label selector to provide a LB and Service DNS
ReplicaSets	A cluster wide Pod manager providing Pod scaling
DaemonSets	A Pod manager to ensure a Pod is scheduled across a Cluster Node set
StatefulSets	Replicated Pods where each Pod gets an indexed hostname
Jobs	A Pod which runs until the process returns a successful termination
Deployments	Manages the release rollout of new versions of Pods
Singletons	A single instance of a Pod which is not replicated or scaled

Developers

Admin

Kubernetes & PKS

Pivotal
Container Service™

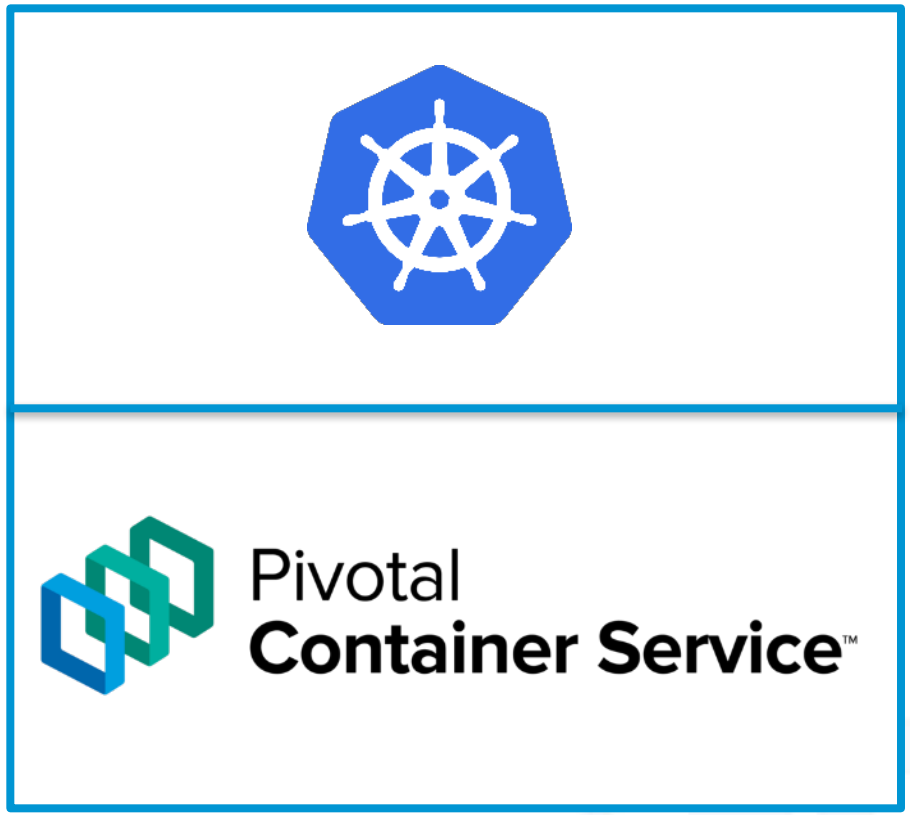
Deploys and Operates
the Kubernetes Clusters

```
$kubectl create -f App.yaml
```

Kubernetes Cluster

The difference between PKS and Kubernetes

Open Source Project – Google/Pivotal/VMware



Container scheduling, scale, resiliency, and Day 2

Desired state of **Application**

Kubernetes cluster scheduling, scale, resiliency, and Day 2

Desired state of **Kubernetes**

VMware PKS Value Proposition

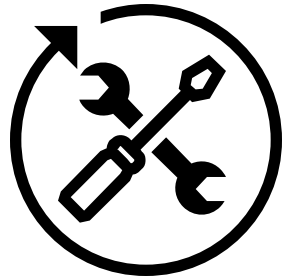
A turnkey solution for enterprises and service providers to provision, operate and manage production grade Kubernetes clusters

Fully supported Kubernetes

PKS uses the latest stable distribution of Kubernetes—with no proprietary extensions and constant compatible with GKE

On-Demand Provisioning

PKS deploys Kubernetes clusters on-demand on vSphere or GCP

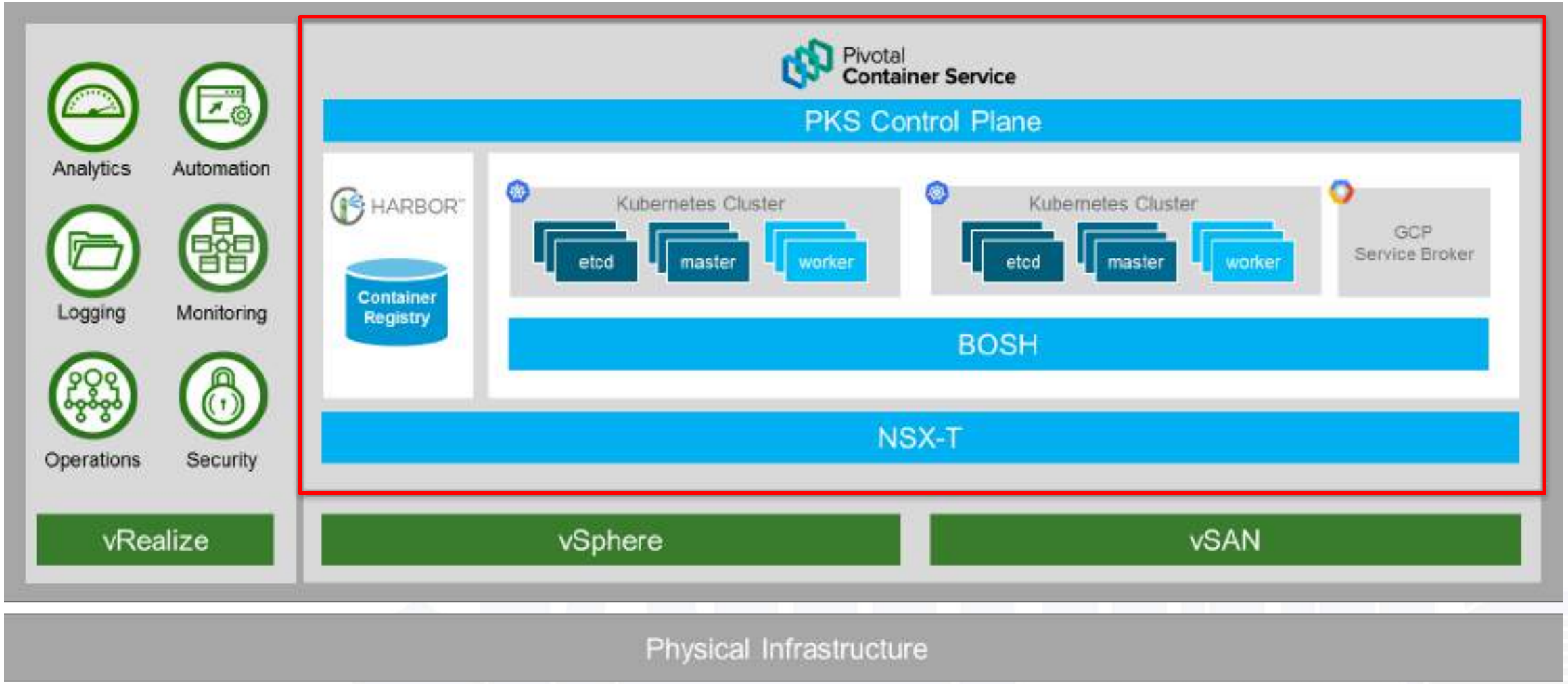


Built for Day 2 Operations

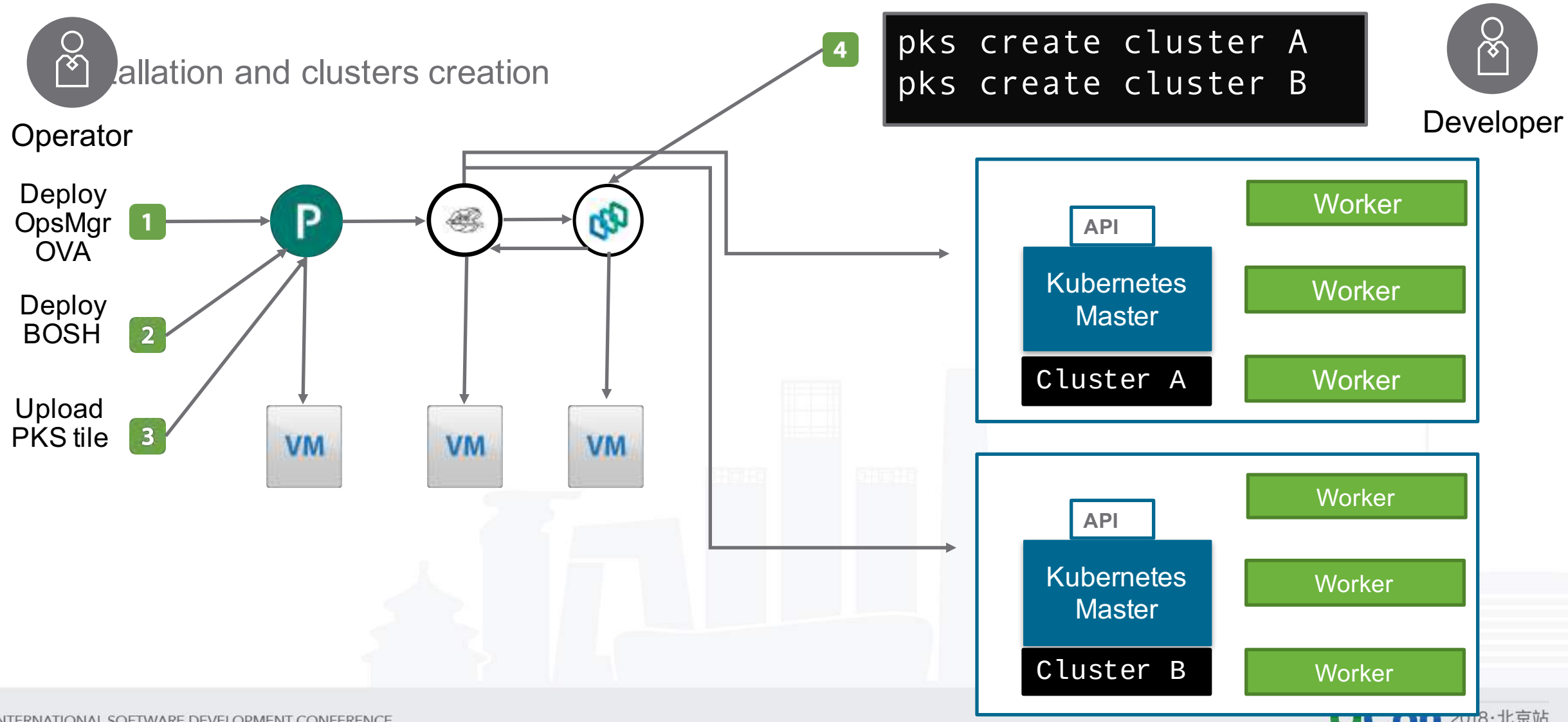
PKS simplifies Day 2 operations with built-in network security—powered by NSX, high availability, logging, monitoring, analytics, and automated health checks.

Container Service for Cloud-Native Apps

Rapidly deliver and operationalize next generation apps



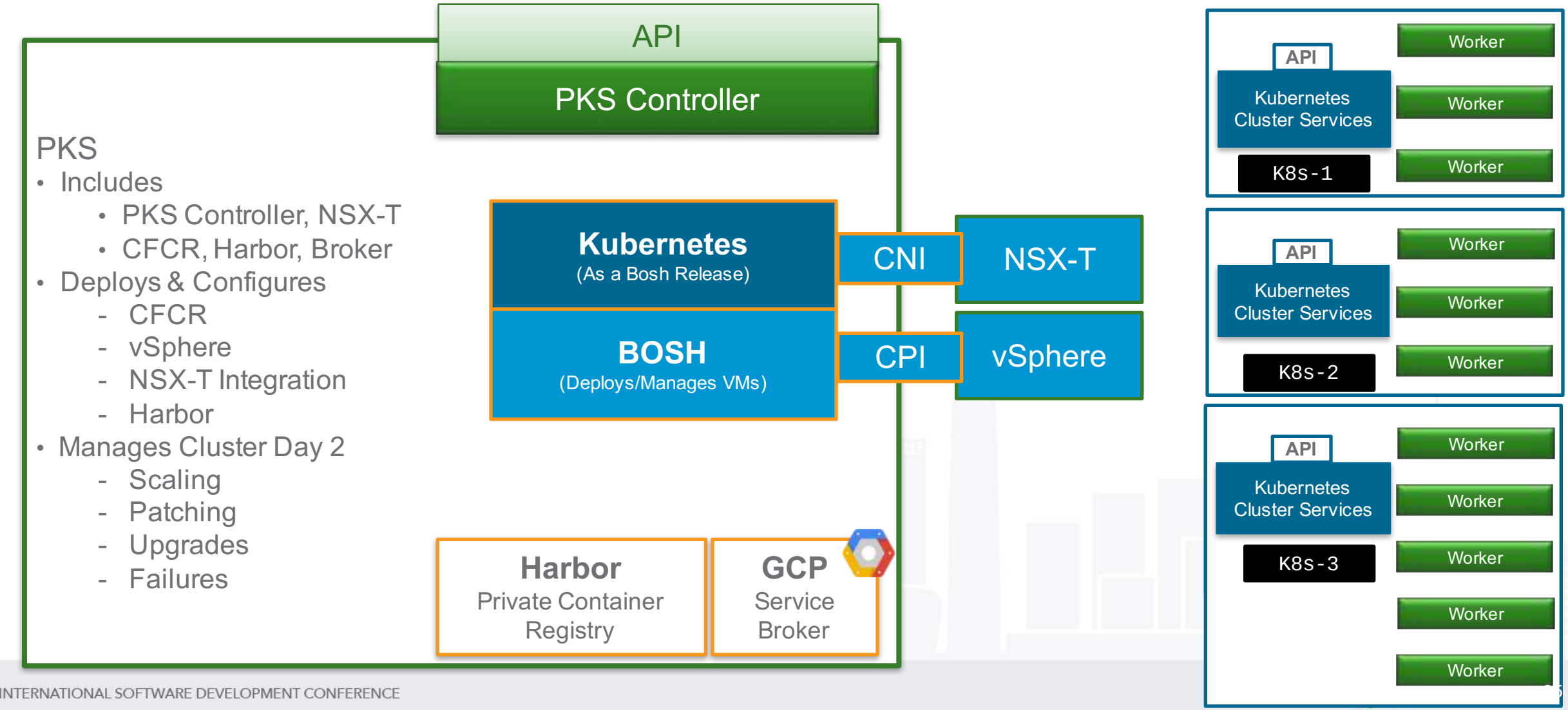
End User Experience



Architecture

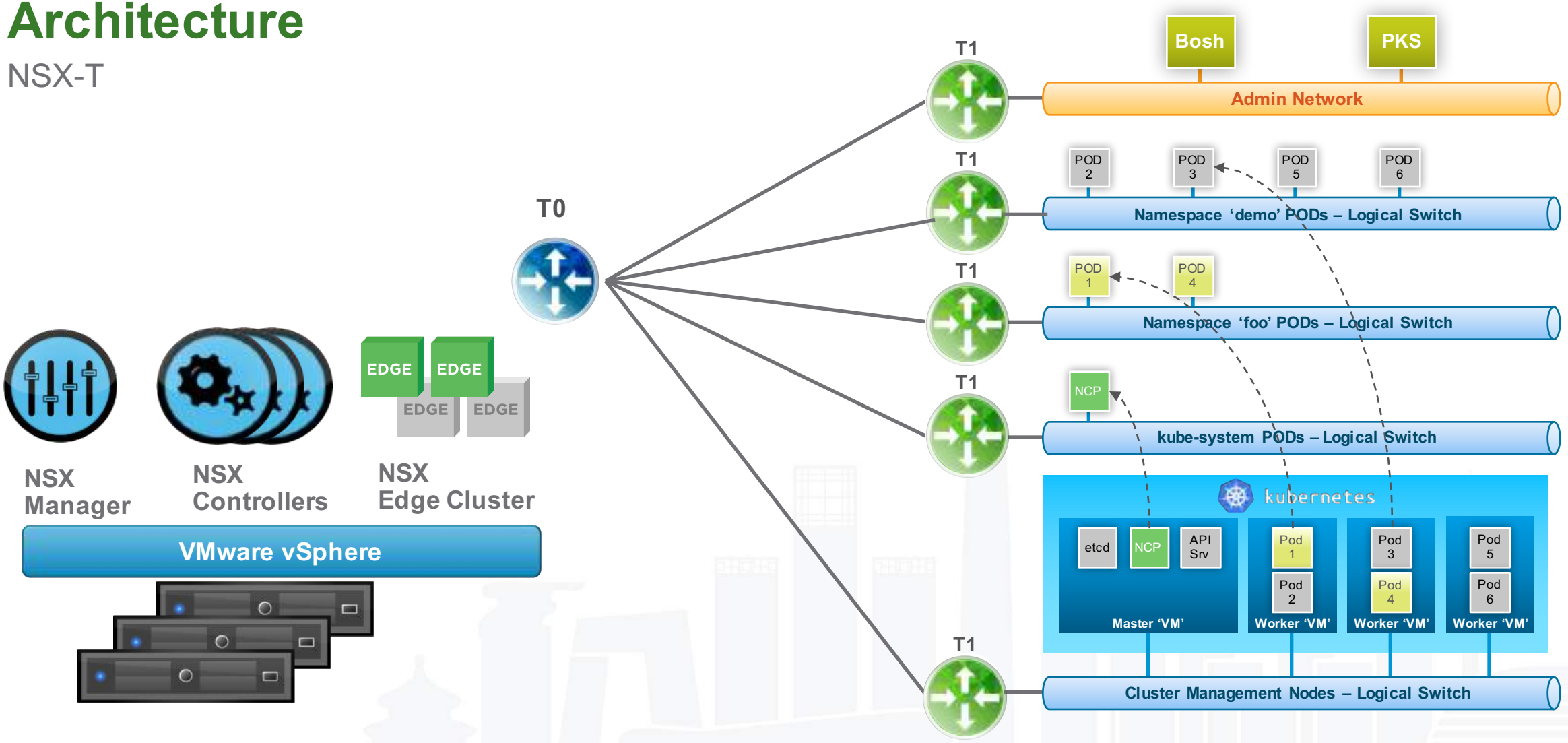
PKS Controller

```
#pks resize K8s-3 n=5
```



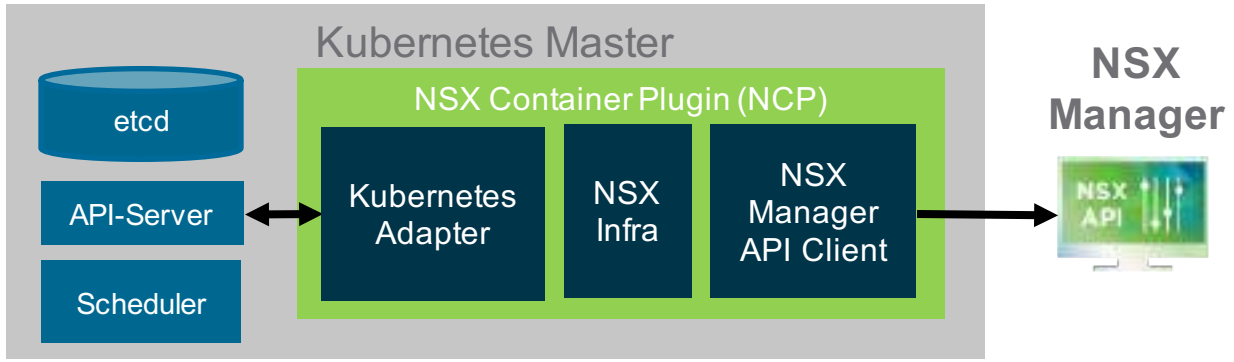
Architecture

NSX-T



Architecture

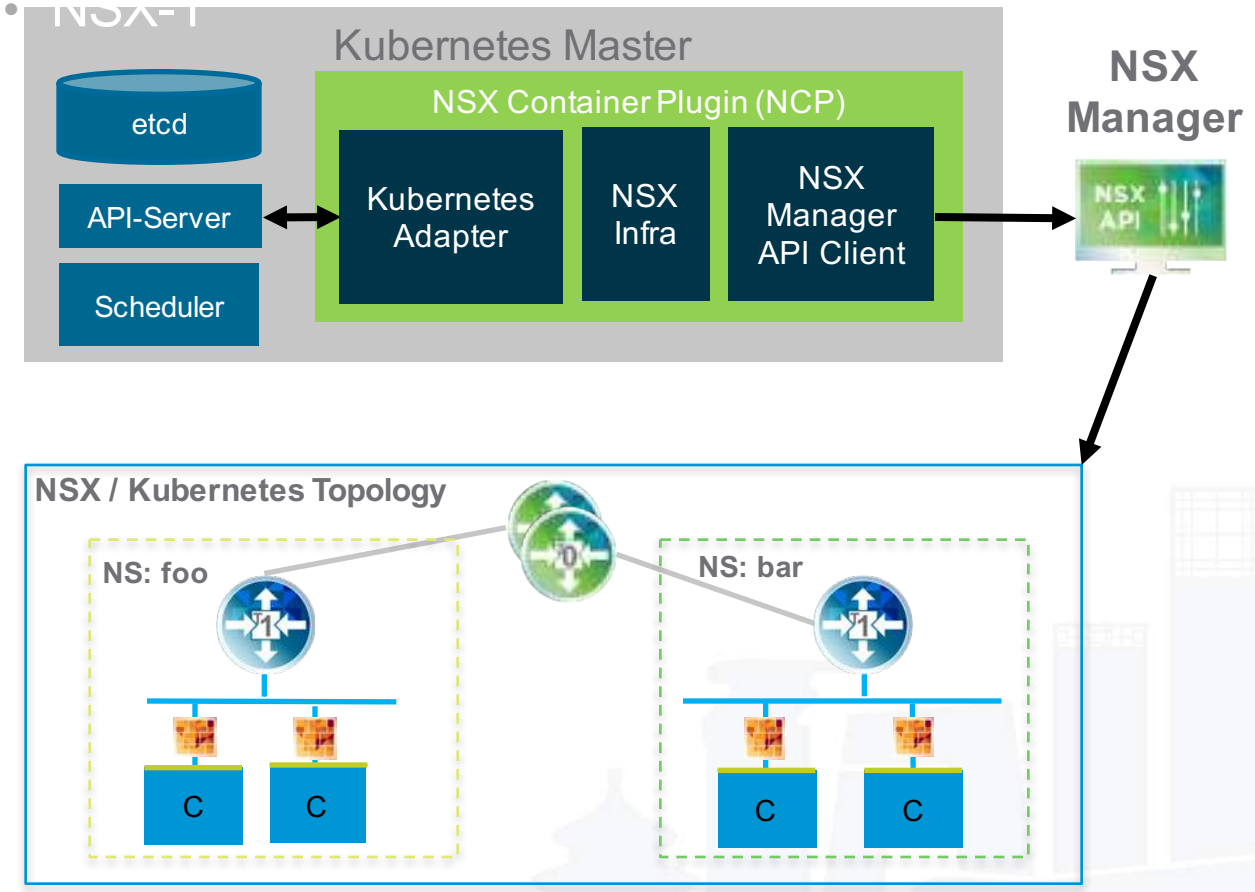
NSX-T



Network Container Plugin (NCP)

- **NSX Container Plugin:** NCP is a software component provided by VMware in form of a container image, runs in K8s as a Pod
- **Kubernetes Adapter:** NCP is built in a modular way, so that individual adapters can be added for different CaaS and PaaS systems
- **NSX Infra layer**: Implements the logic that creates topologies, attaches logical ports, etc. based on triggers from the Adapter layer
- **NSX API Client**: Implements a standardized interface to the NSX API

Architecture

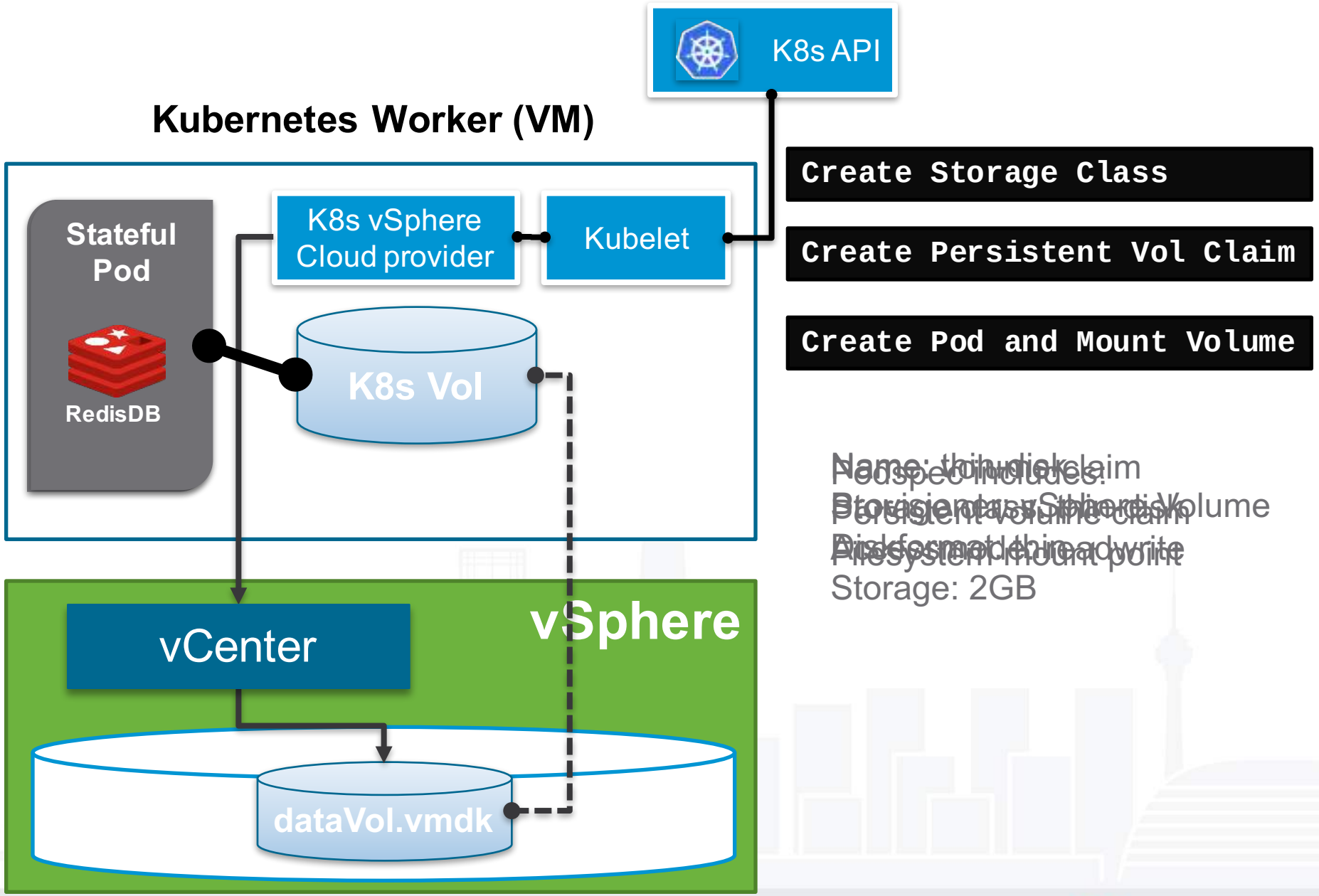


Namespace Creation Workflow

1. NCP creates a 'watch' on K8s API for any Namespace events
2. A user creates a new K8s Namespace
3. The K8s API Server notifies NCP of the change (addition) of Namespaces
4. NCP creates the network topology for the Namespace :
   a) Requests a new subnet from the pre-configured IP block in NSX
   b) Creates a logical switch
   c) Creates a T1 router and attaches it to the pre-configured global T0 router
   d) Creates a router port on the T1 router, attaches it to the LS, and assigns an IP from the new subnet

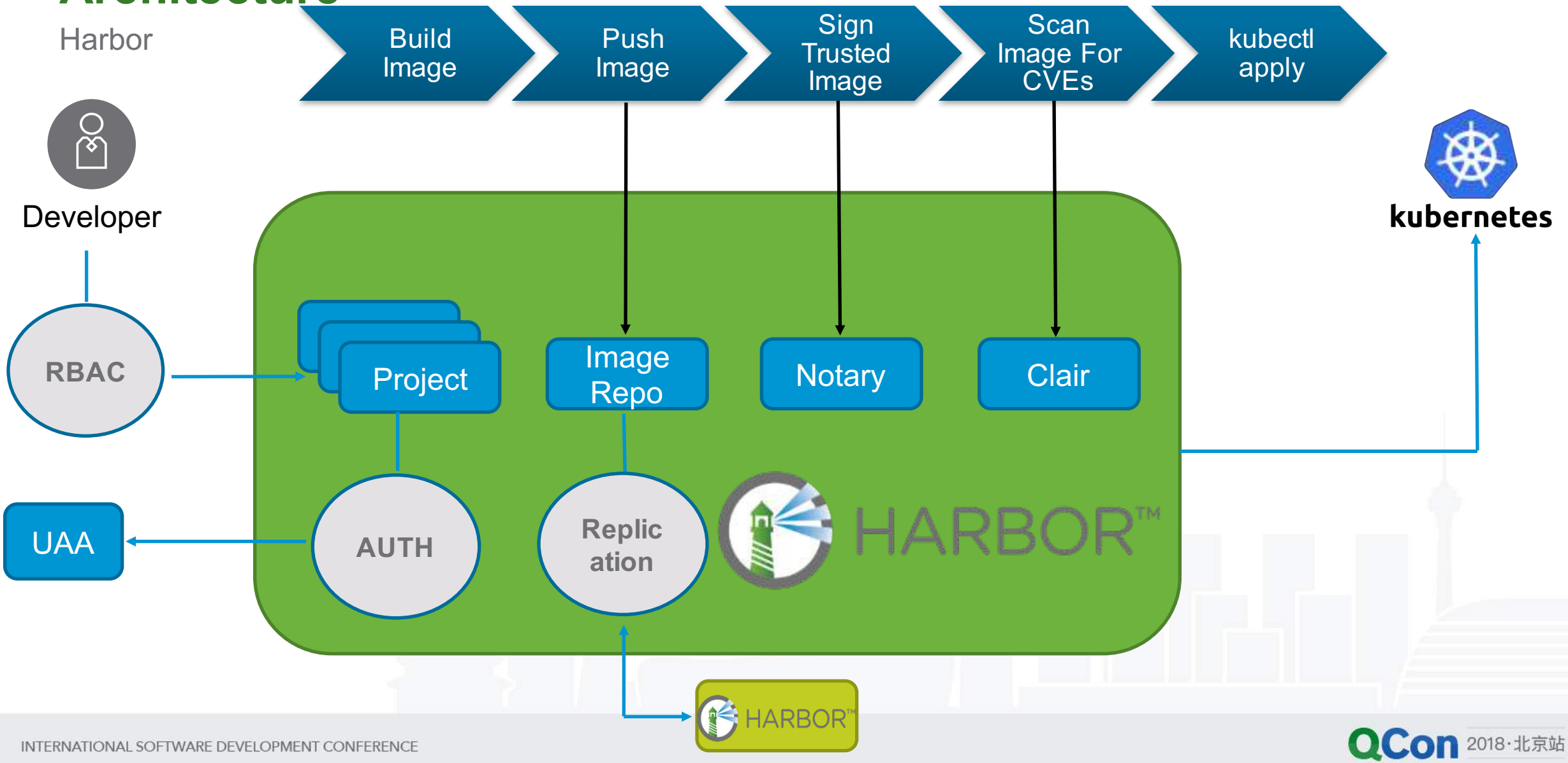
Architecture

Storage

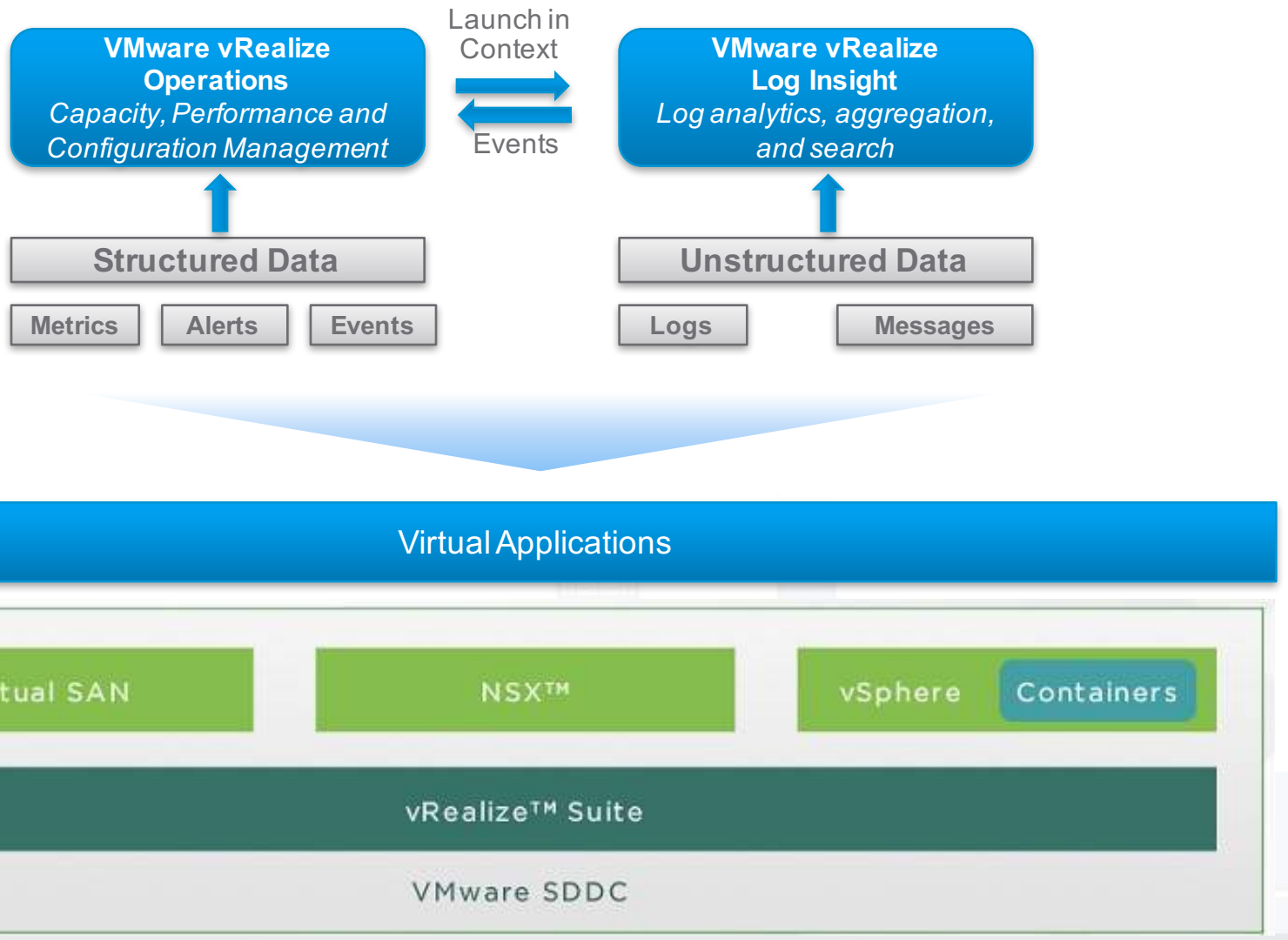


Architecture

Harbor



vRealize Ops, Log Insight For Comprehensive Visibility



Wavefront By VMware
SaaS-Based Metrics Monitoring and Analytics Platform

Thank You!


@cloudnativeapps

vmware.github.io

blogs.vmware.com/cloudnative