

Kubernetes for the Virtualization Admin



Michael Cade

Technologist | Global Field CTO
Veeam® Software
@MichaelCade1



Dean Lewis
Veeam Vanguard

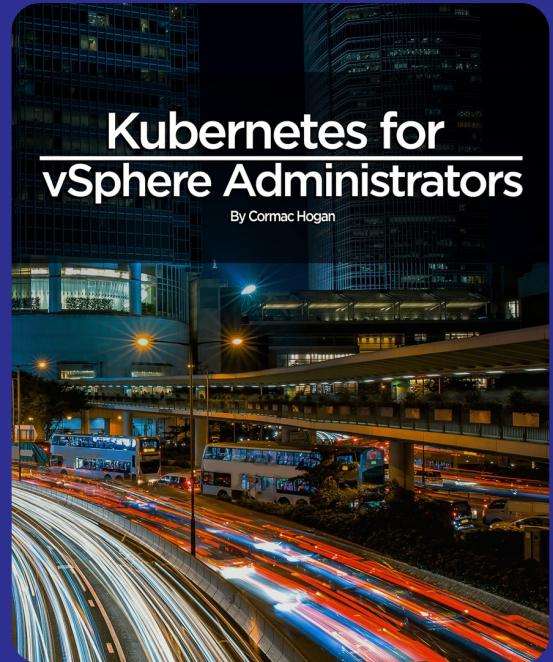


Agenda

- Intro to containers
- Intro to Kubernetes
- Kubernetes on vSphere
- Kubernetes networking with vSphere
- Kubernetes storage with vSphere
- vSphere with Tanzu
- Day two operations

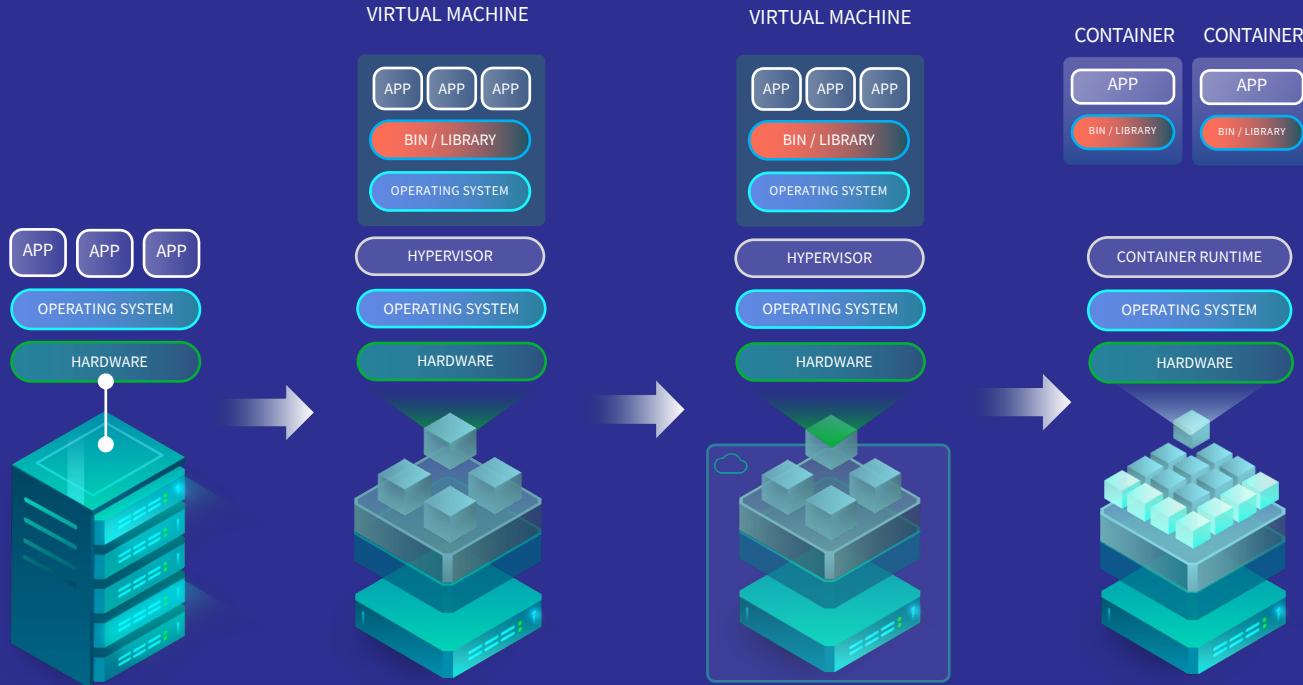


vmware®





Intro to containers

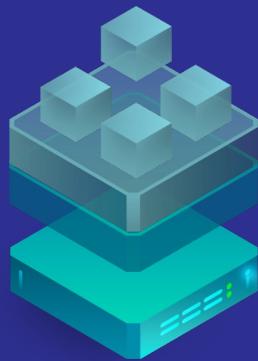


None of these have gone away!

Physical systems



Virtualization



Cloud



Containers

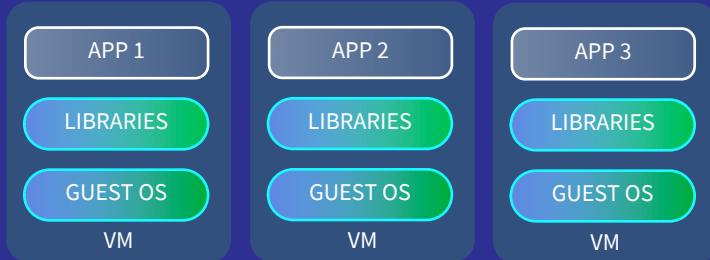


Which one do we choose?

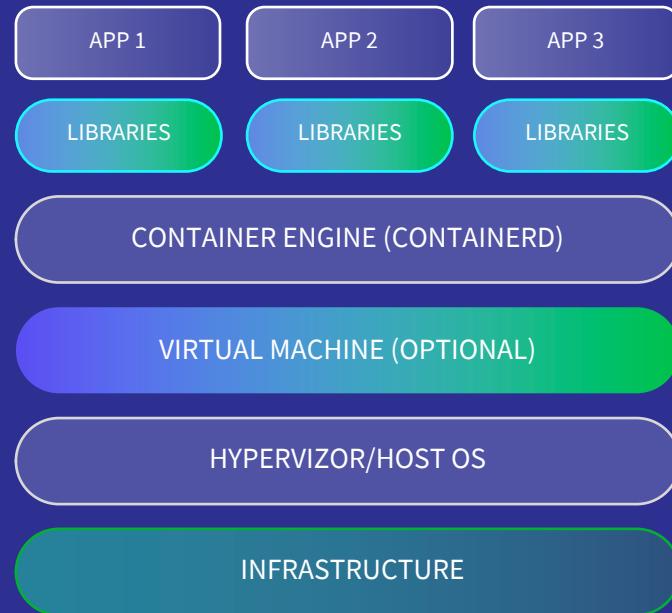


Containers & VMs

Virtualized infrastructure



Containerized infrastructure



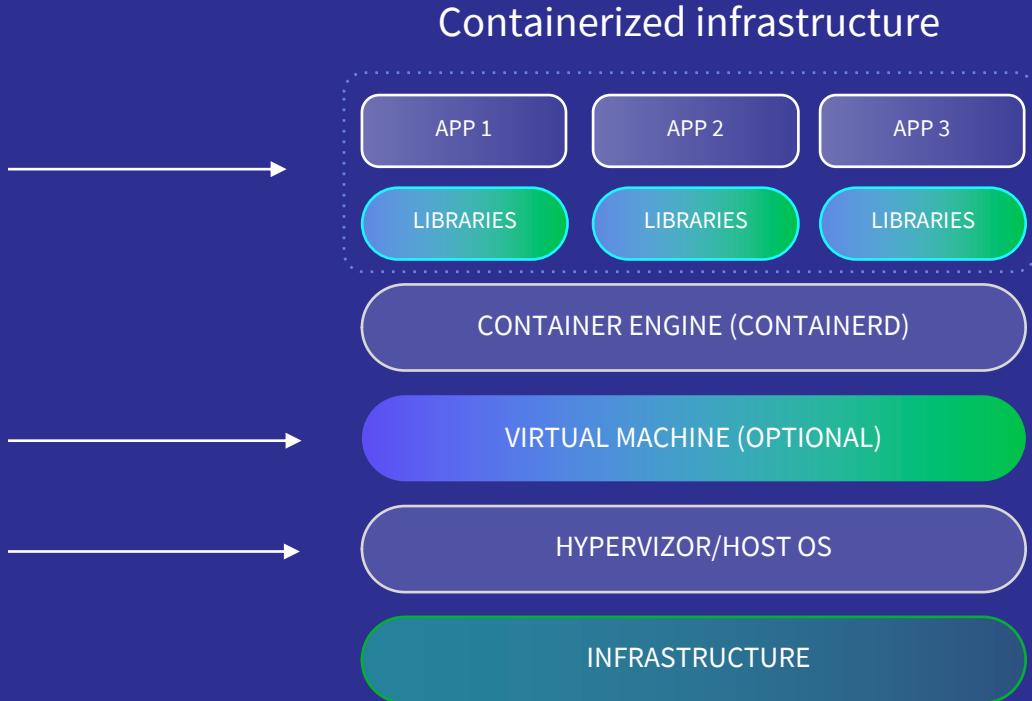
What are containers & containerization?

Just as the use of shipping containers transformed global logistics, the rise of the software container is transforming **software development and deployment**.



What are containers & containerization?

Remember this box as
a **deployment** for later

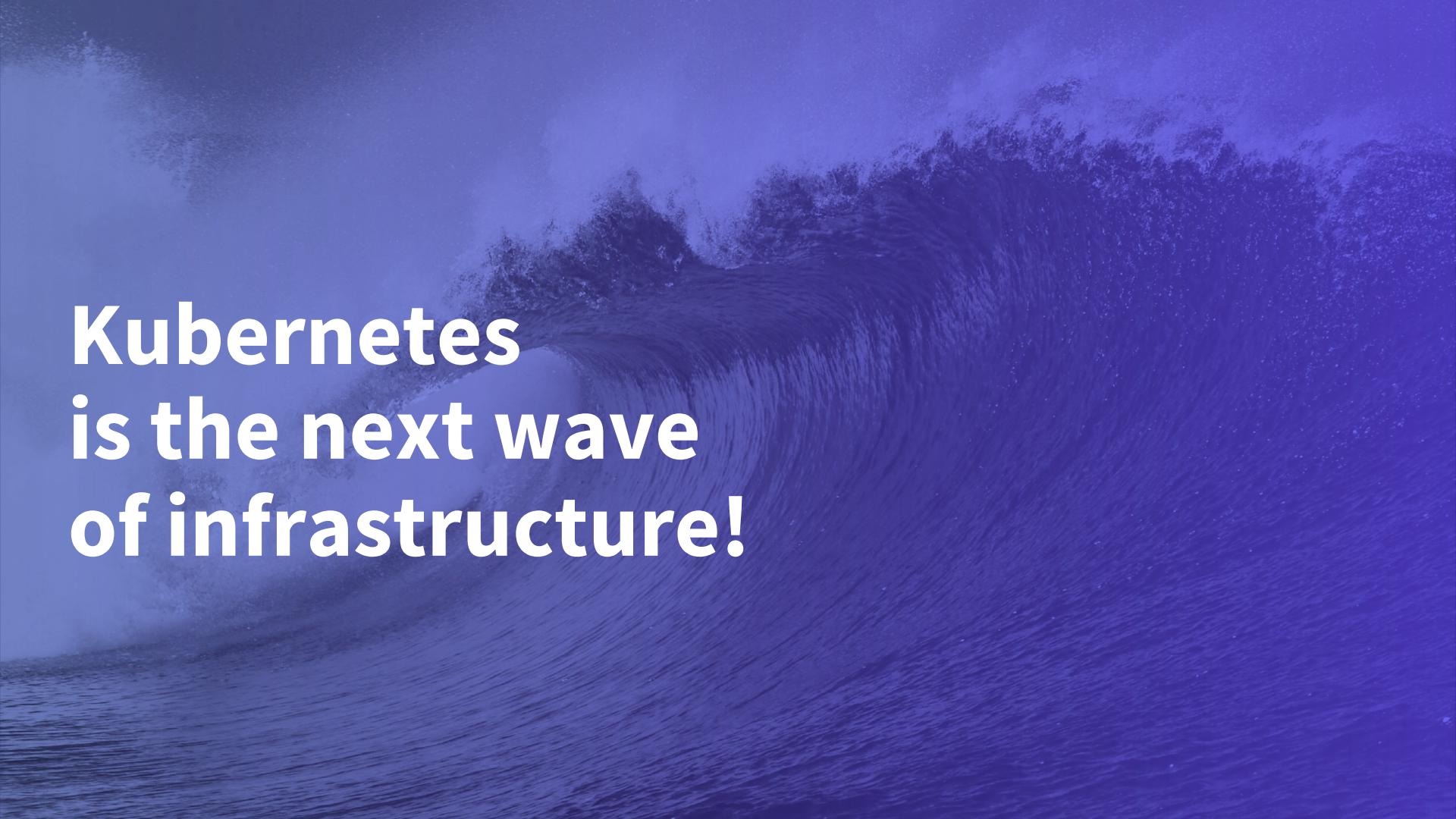


Remember these
as **nodes** for later

veeAMON 2023

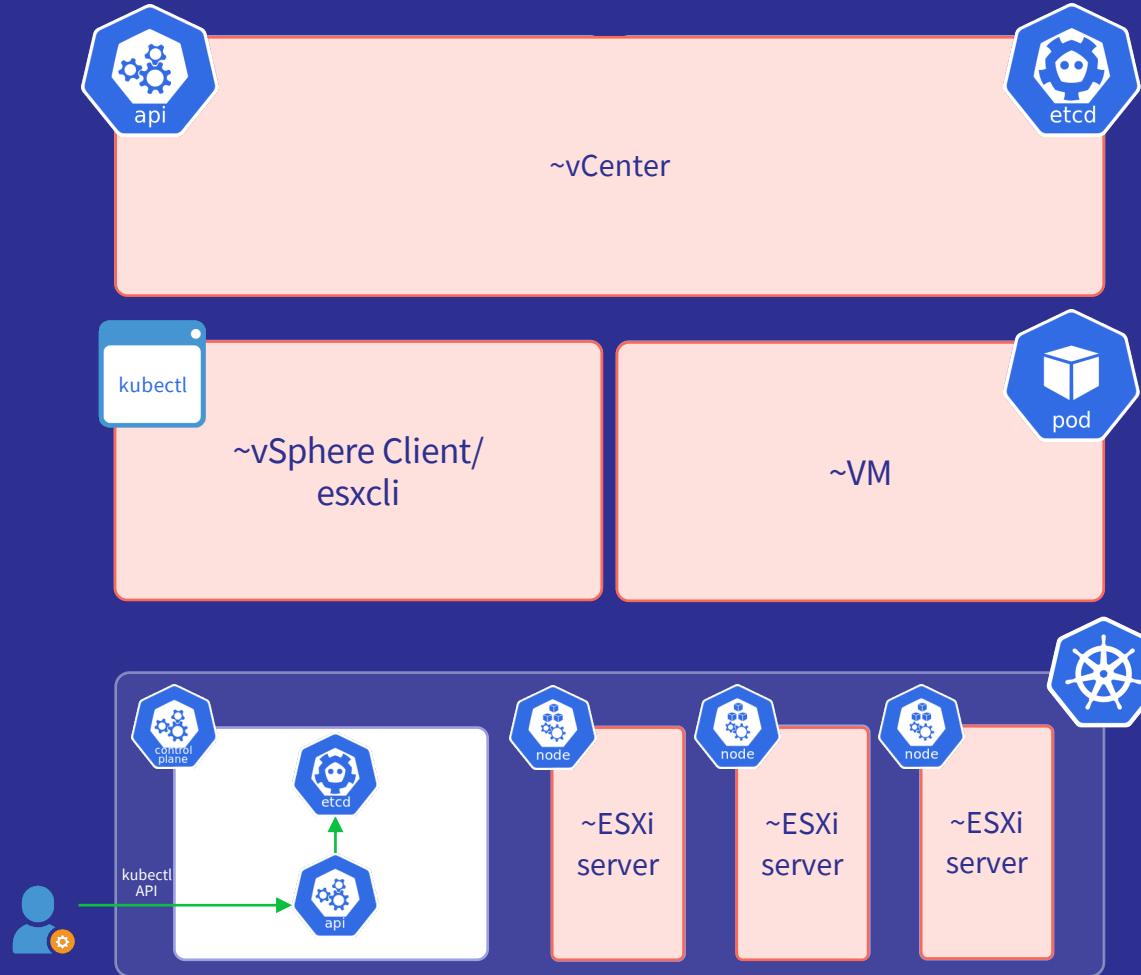


Intro to Kubernetes



**Kubernetes
is the next wave
of infrastructure!**





Kubernetes = container orchestration

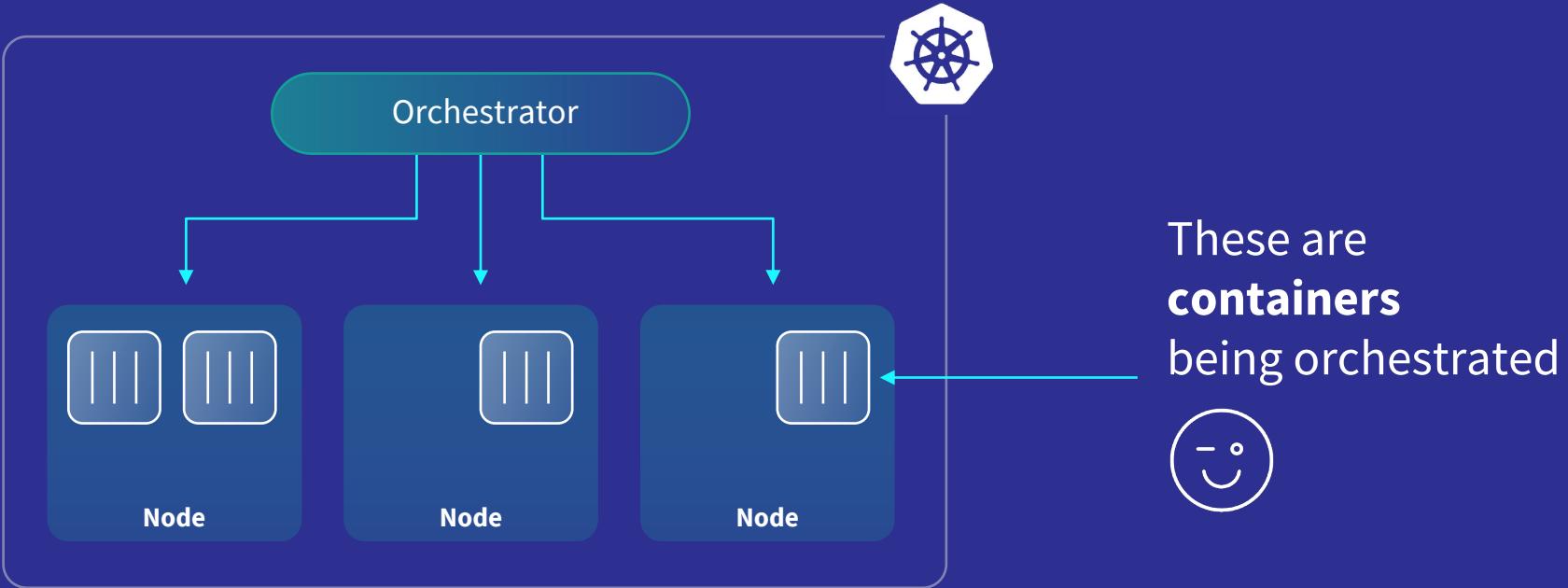
Container orchestration manages the deployment, placement and lifecycle of containers.
It also has many other responsibilities:

- Cluster management federates hosts into one target.
- Schedule management distributes containers across nodes through the scheduler.
- Service discovery knows where containers are located and distributes client requests across them.
- Replication ensures that the right number of nodes and containers are available for the requested workload.
- Health management detects and replaces unhealthy containers and nodes.



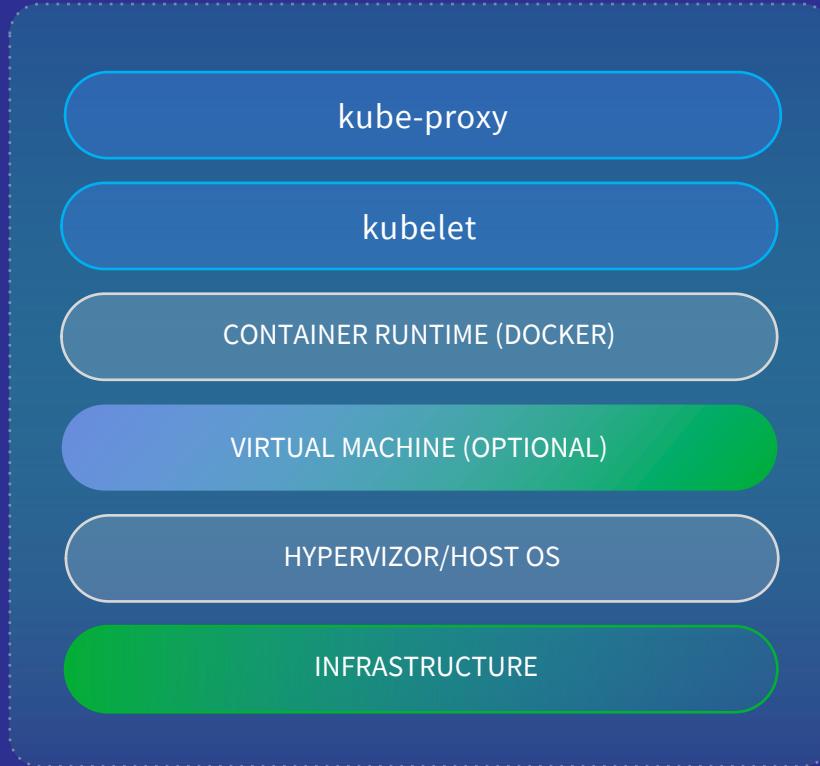
Kubernetes = container orchestration

**Container orchestration manages the deployment, placement and lifecycle of containers.
It also has many other responsibilities:**



The building blocks – Kubernetes

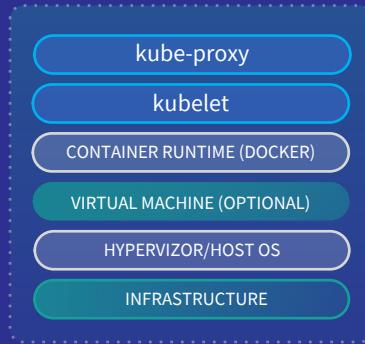
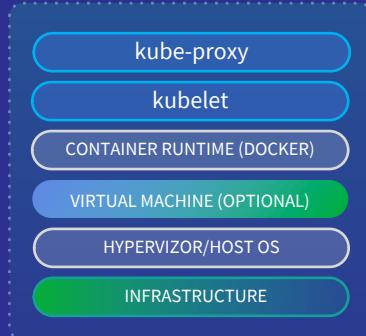
Node



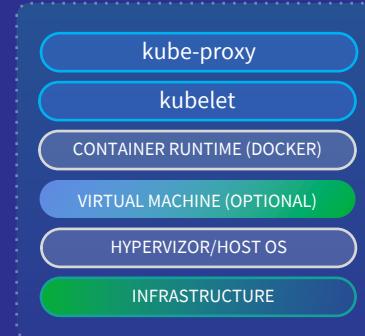
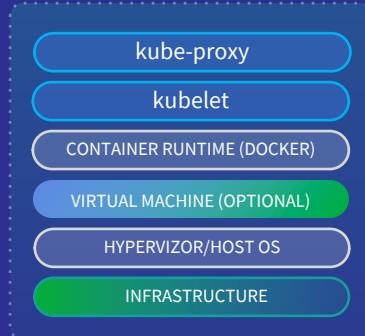
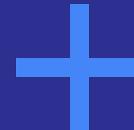
The building blocks – cluster & nodes

Kubernetes cluster

Worker node

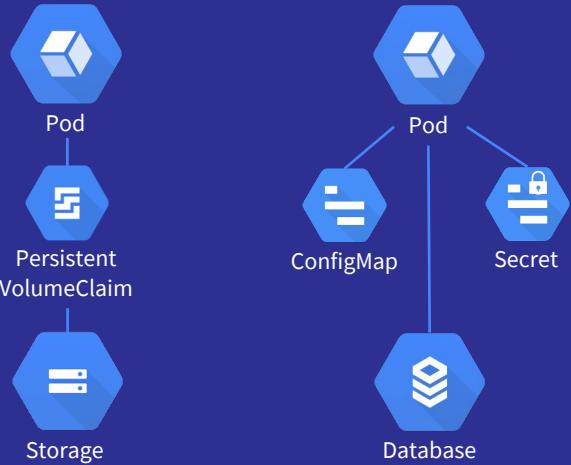


Master node



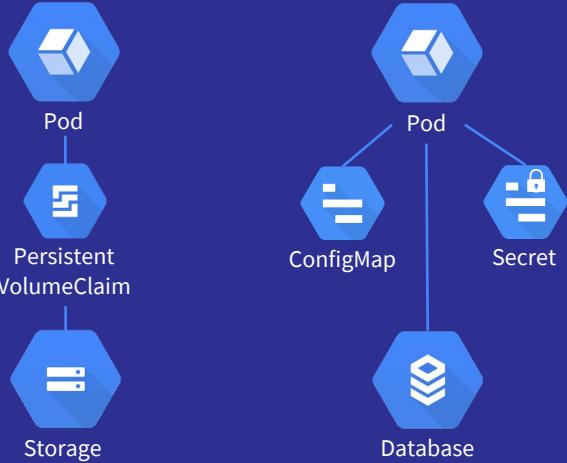
The building blocks – pods

- The **pod** is the basic building block of Kubernetes.
- A **pod** contains a group of one or more containers.
- Generally, each **pod** has one container.
- **Pods** handle volumes, secrets and configuration.
- **Pods** are ephemeral – restarted automatically when they die.



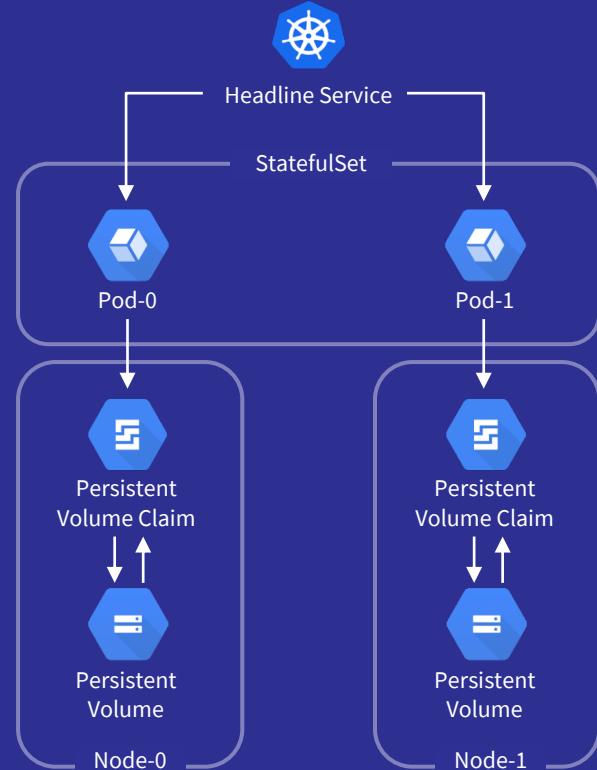
The building blocks – deployments

- You can just decide to run pods but when they die, they die.
- A deployment will enable your pod to run continuously.
- Deployments allow you to update a running app without downtime.
- Deployments also specify a strategy to restart pods when they die.



The building blocks – StatefulSets

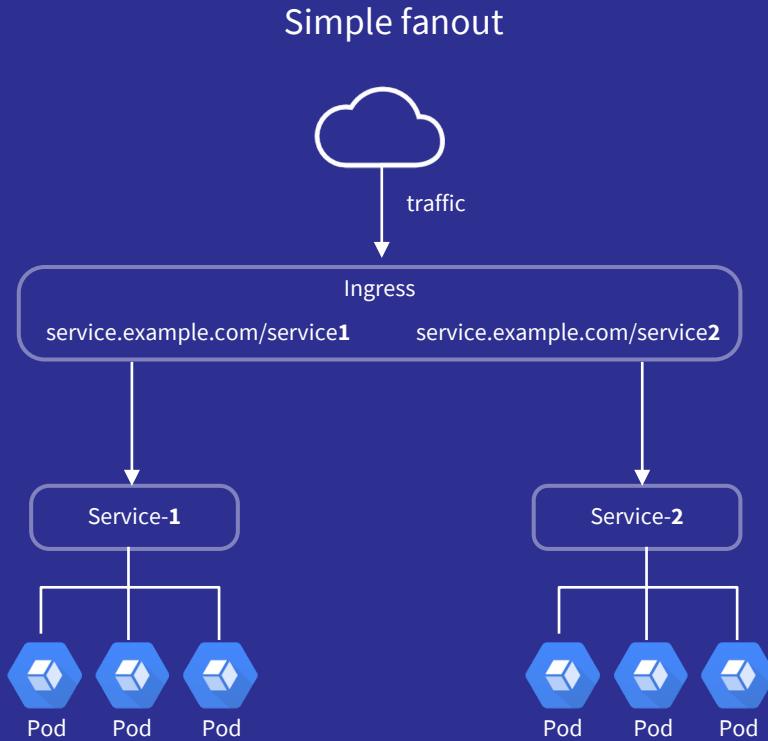
- Does your app require you to keep information about its state?
- A database needs state.
- A StatefulSet's pods are not interchangeable.
- Each pod has a unique, persistent identifier that the controller maintains over any rescheduling.
- Each pod has a unique, persistent identifier that the controller maintains over any rescheduling.



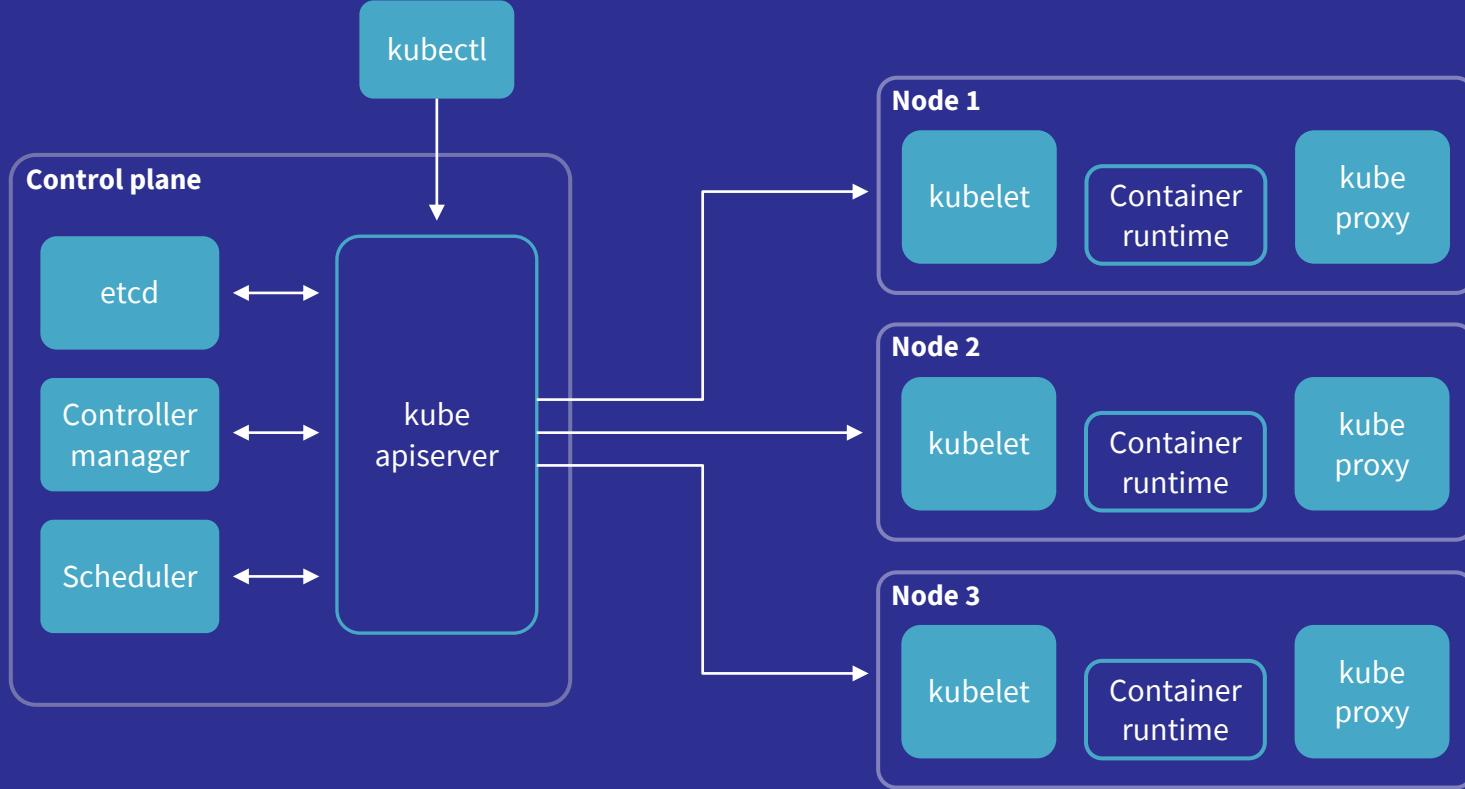
The building blocks – Kubernetes

Services:

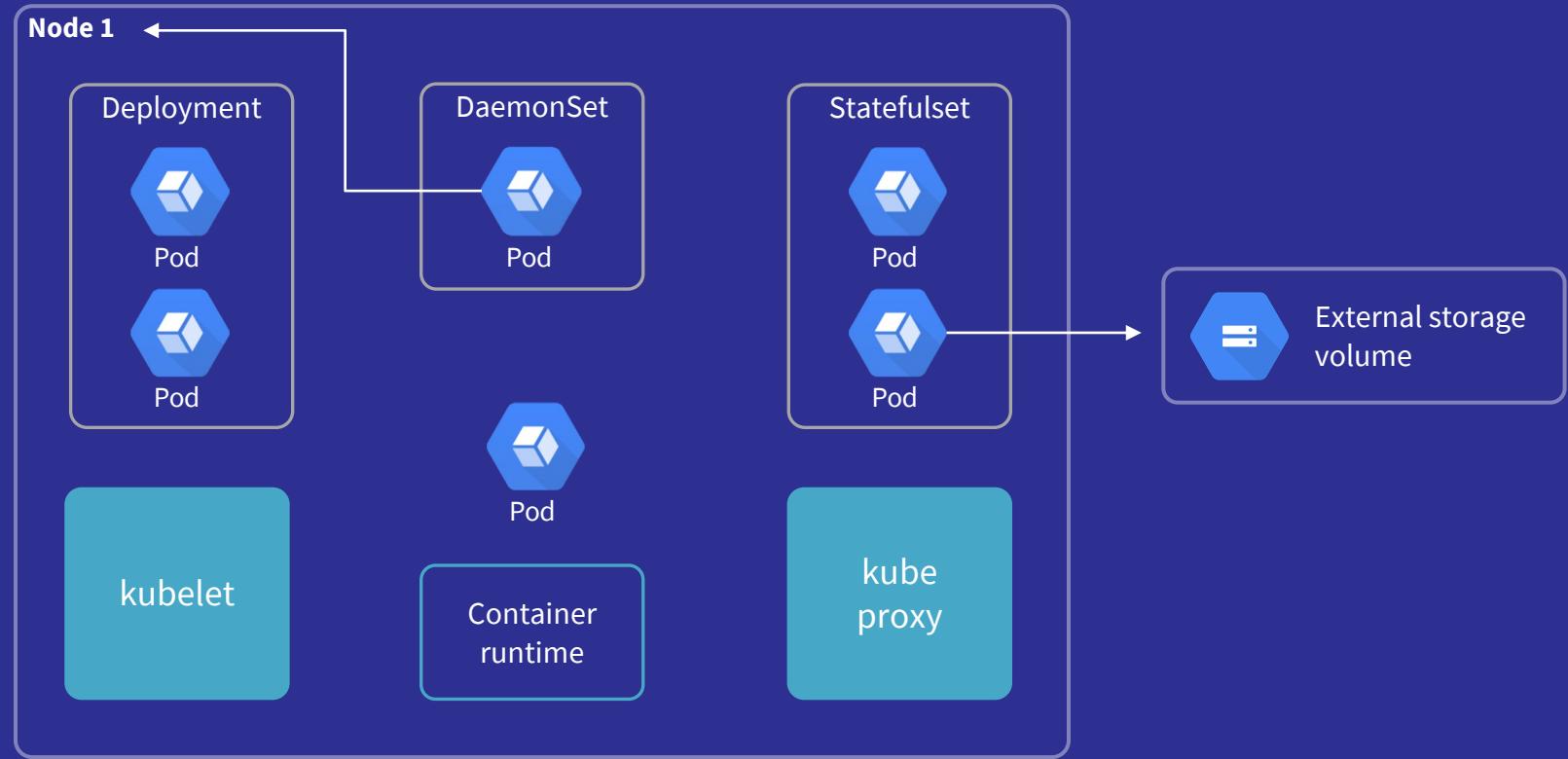
- A single endpoint to access pods.
- A unified way to route traffic to a cluster and eventually to a list of pods.
- By using a service, pods can be brought up and down without affecting anything.



Cluster architecture



Workloads



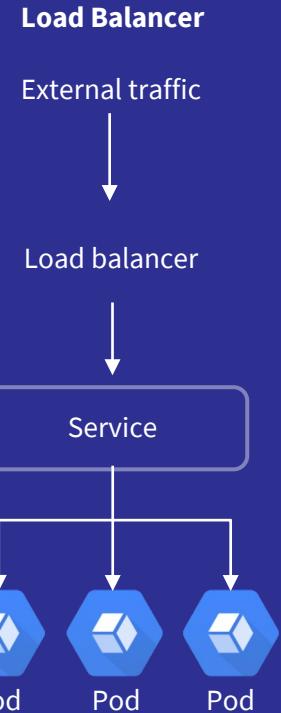
Services

A single endpoint to access pods.

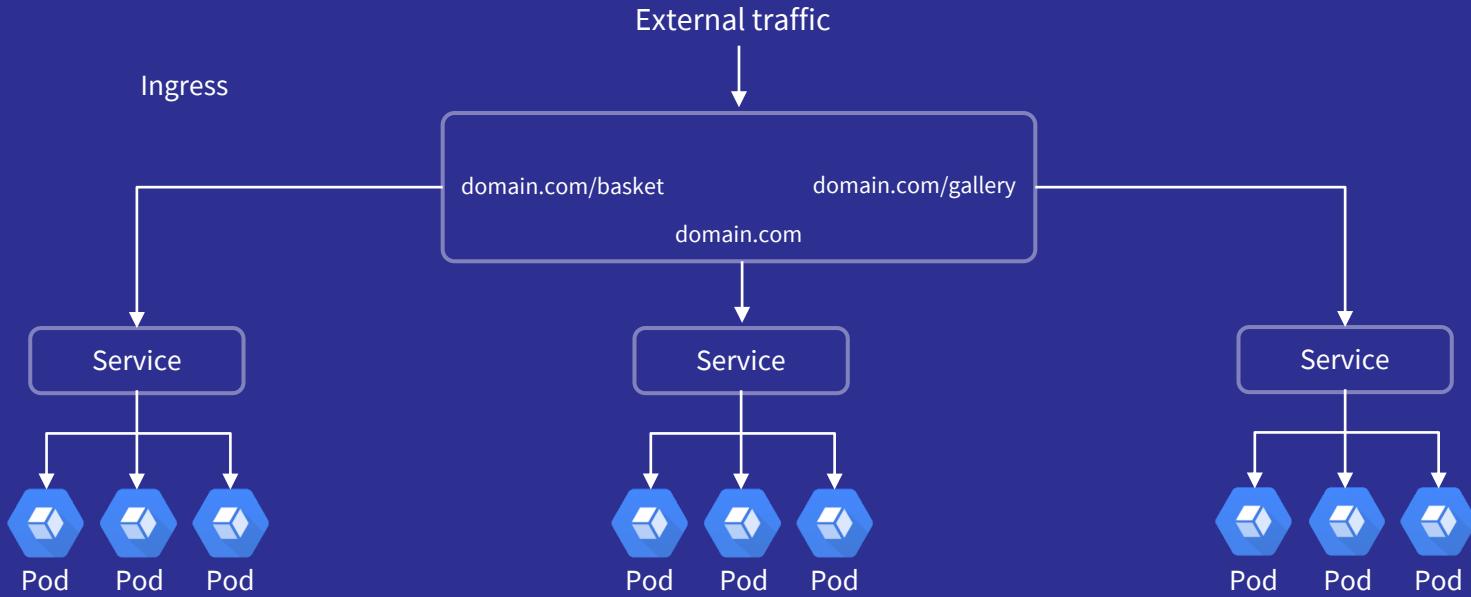
A unified way to route traffic to a cluster and eventually to a list of pods.

By using a service, pods can be brought up and down without affecting anything.

You can also have ClusterIP and NodePort Services.



Services & networking

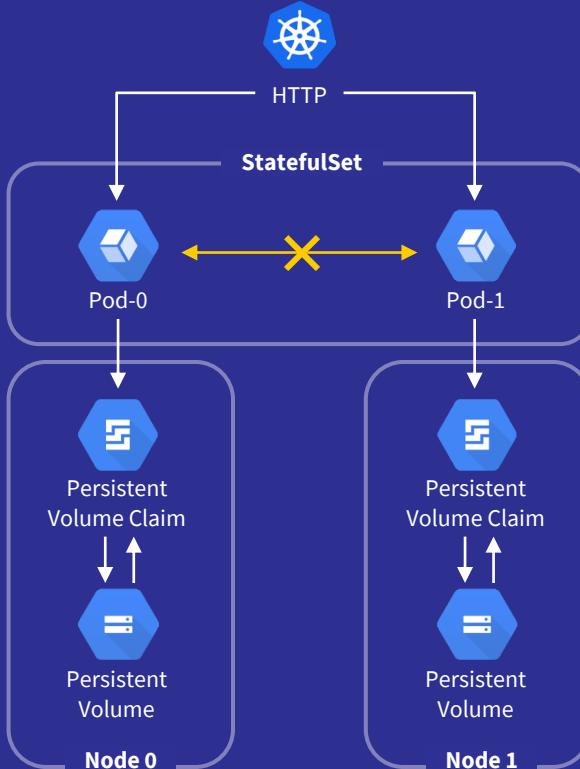


Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource

Networking policies

- Type: network policies are an application-centric construct which allow you to specify how a pod is allowed to communicate with various network "entities" over the network
- By default, if no policies exist in a namespace, then all ingress and egress traffic is allowed to and from pods in that namespace.

TLDR; control traffic flow at the IP address or port level (OSI layer 3 or 4).



veeAMON 2023



Kubernetes on vSphere



Kubernetes networking with vSphere



Kubernetes storage with vSphere

veeAMON2023



vSphere with Tanzu



Day two operations

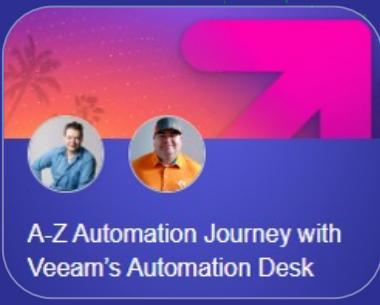
Other Sessions to take a look at:



MIA14 - Diving in: Kasten K10 +
Veeam Backup & Replication



Top Secret: Kasten K10
Announcement



A-Z Automation Journey With
Veeam's Automation Desk



Thank you!