

Kubernetes Fundamentals

Tommy Falgout
Kendall Roden



Introduction



Tommy Falgout

2 years at Microsoft

20+ years as Unix/Linux developer

Cloud Solution Architect

@lastcoolname <https://lastcoolnameleft.com>

thfalgou@microsoft.com



Kendall Roden

3 years at Microsoft

Premier Developer Consultant

@KendallRoden

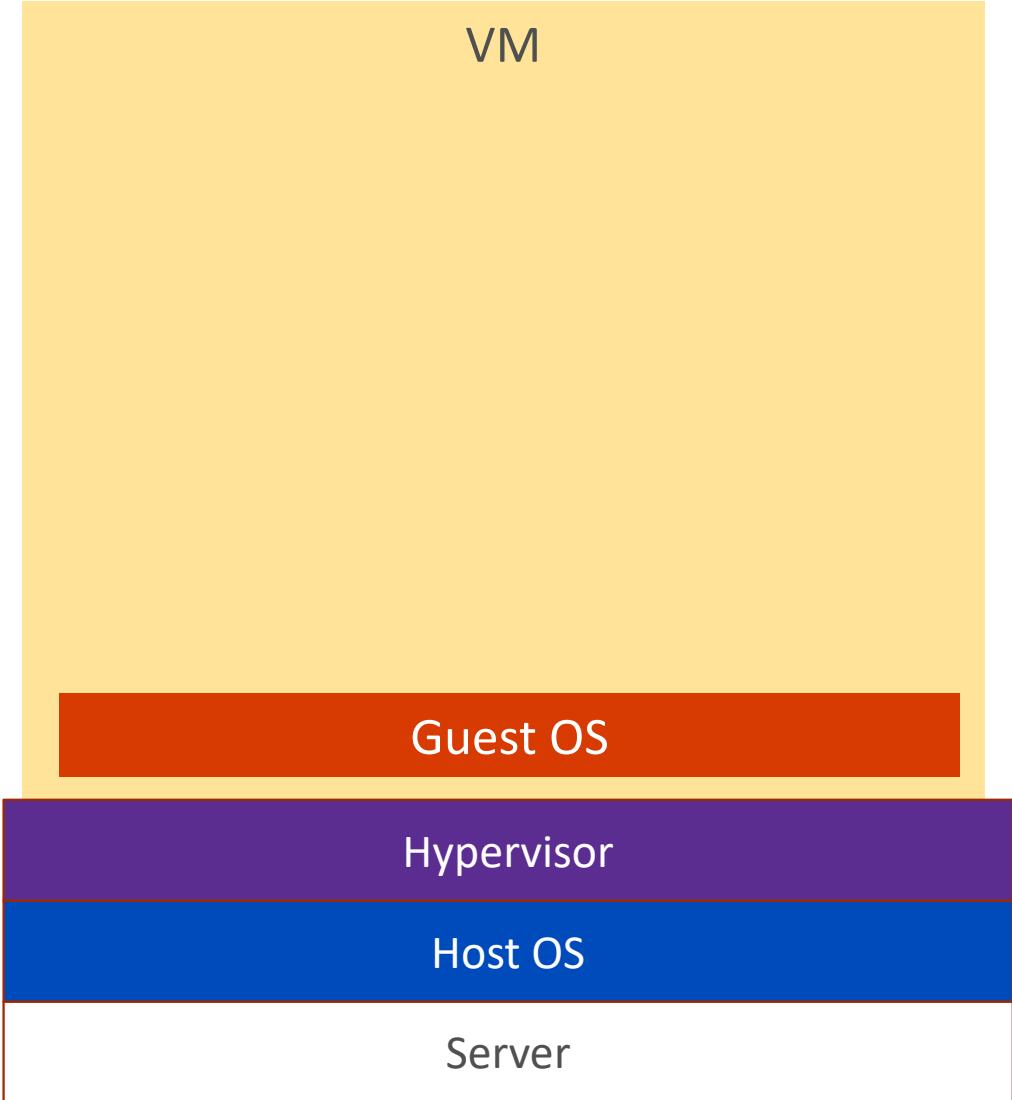
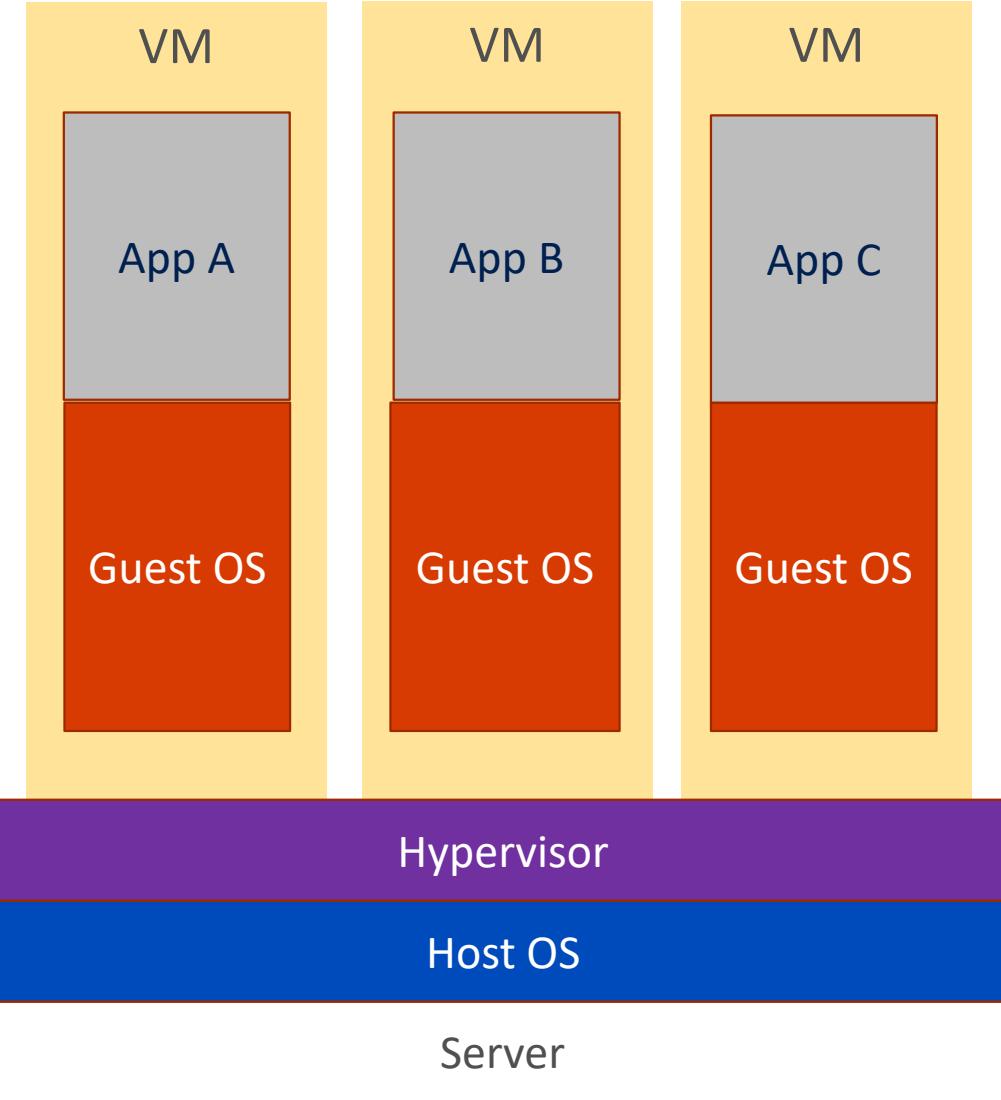
keroden@microsoft.com

Session learning objectives

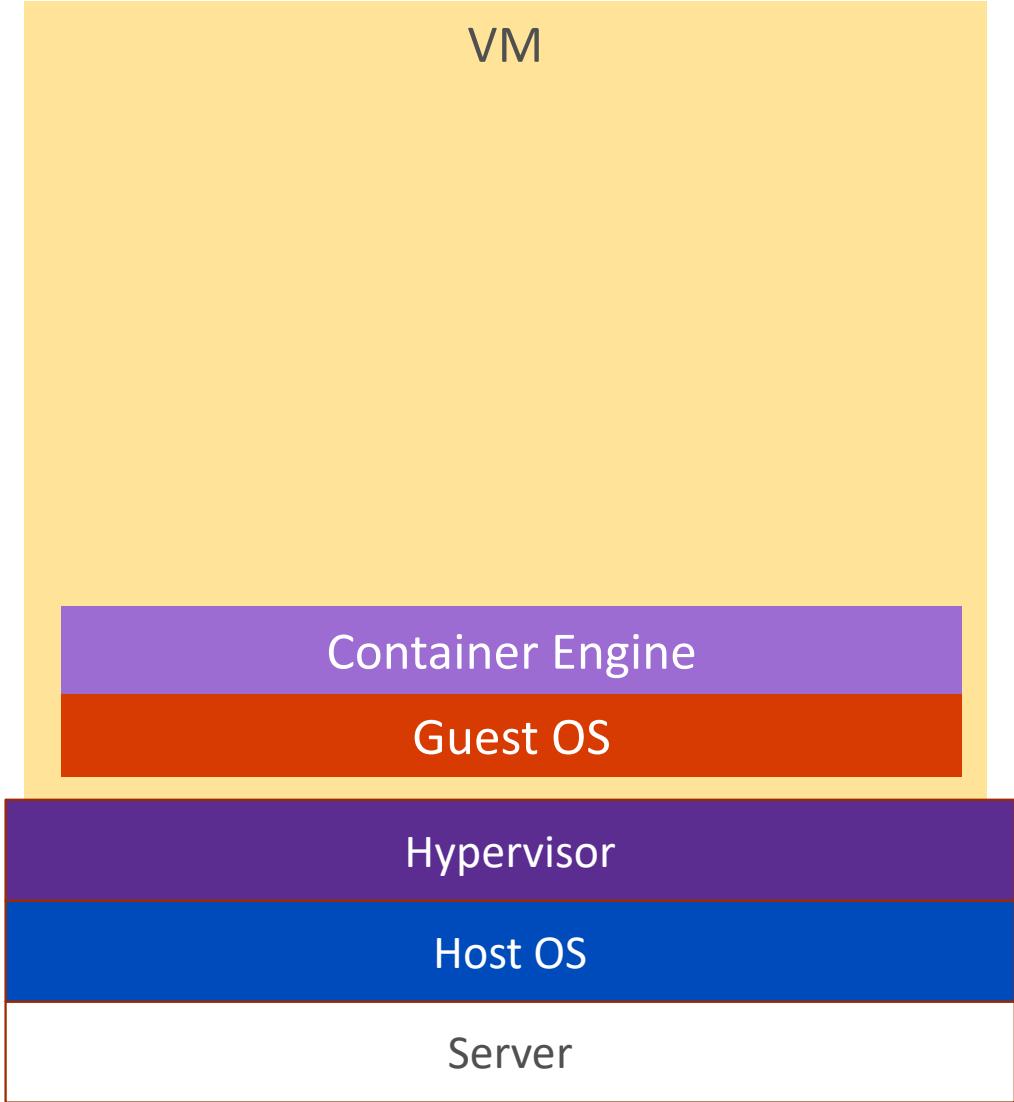
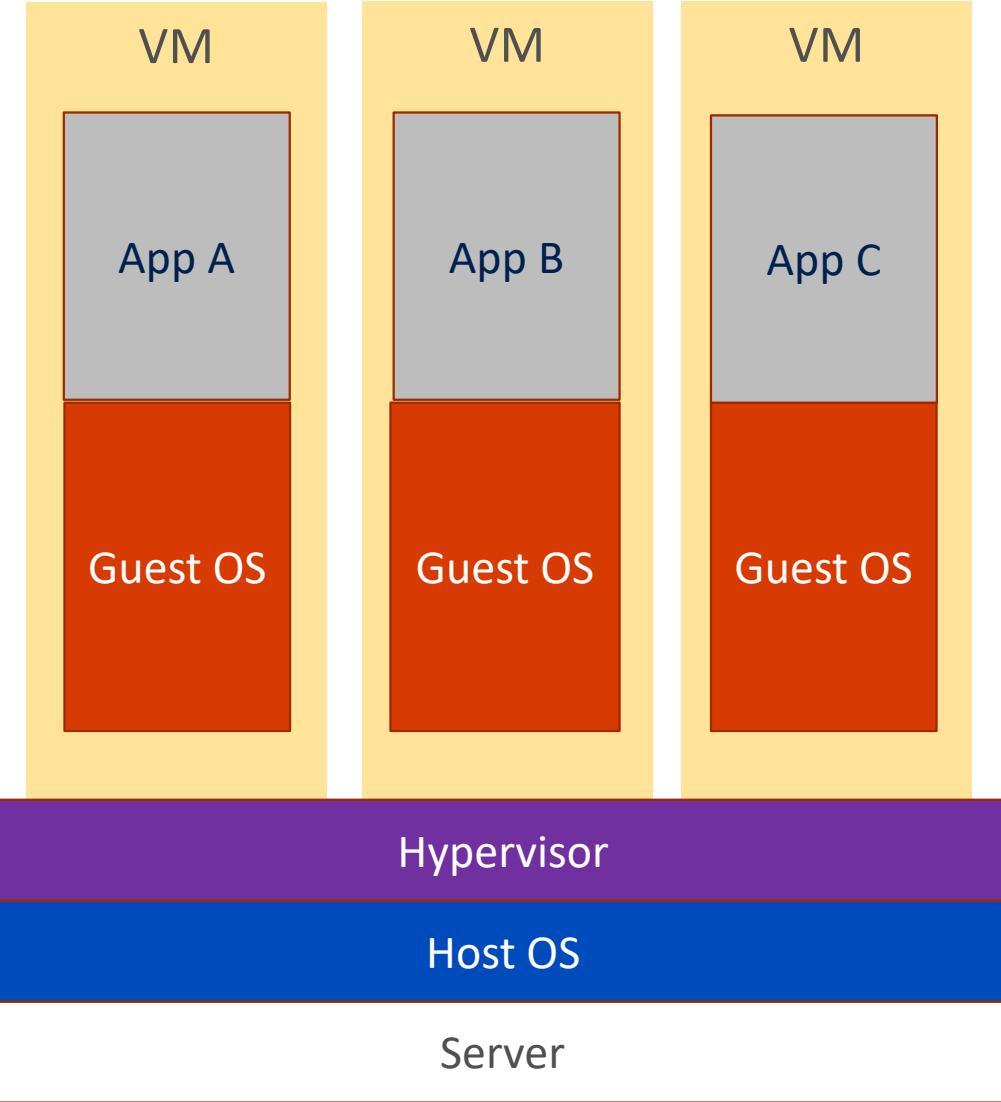
At the end of this session, you should be better able to...

- Explain how Kubernetes works and how it can improve the scalability and maintainability of production deployments
- Understand how managed Kubernetes offerings such as AKS can simplify the implementation of Kubernetes

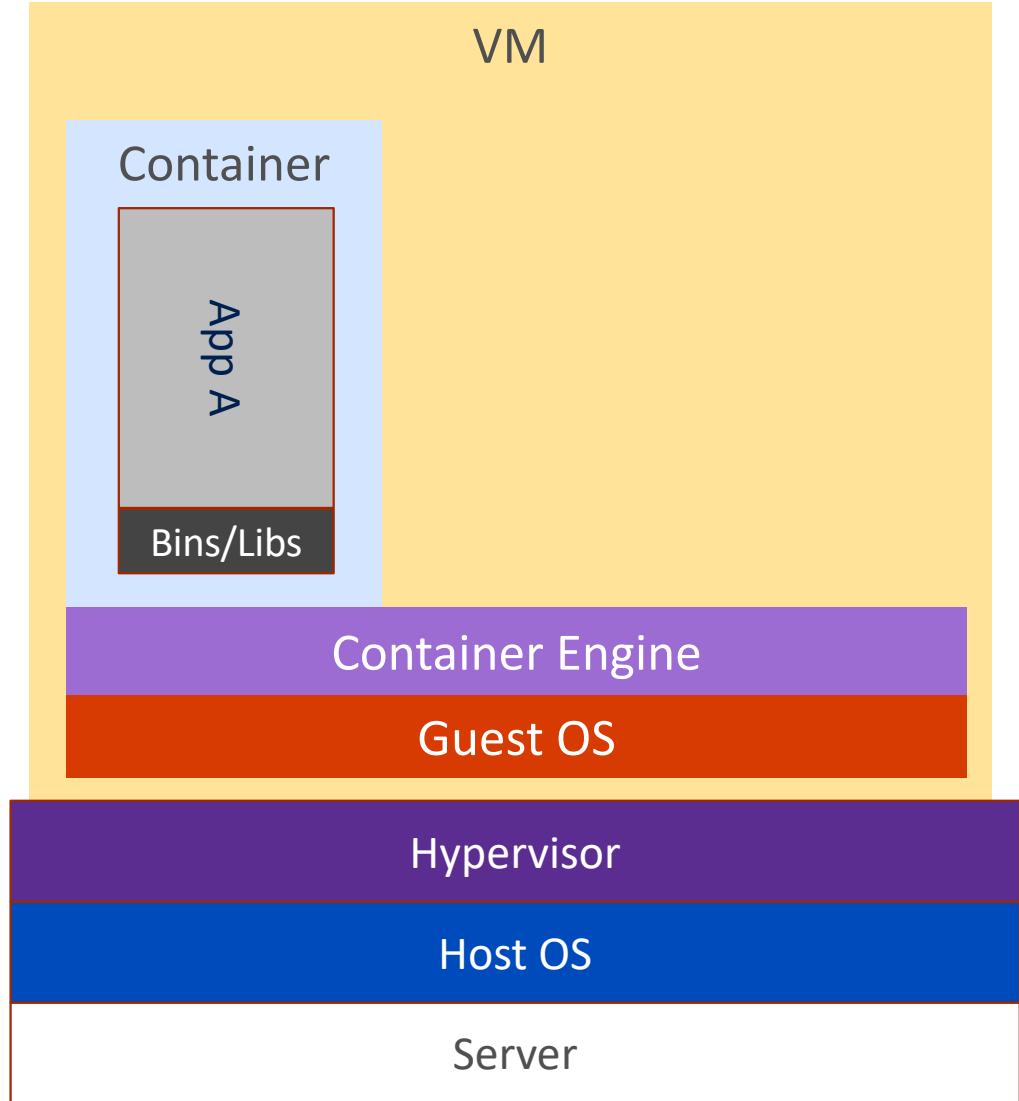
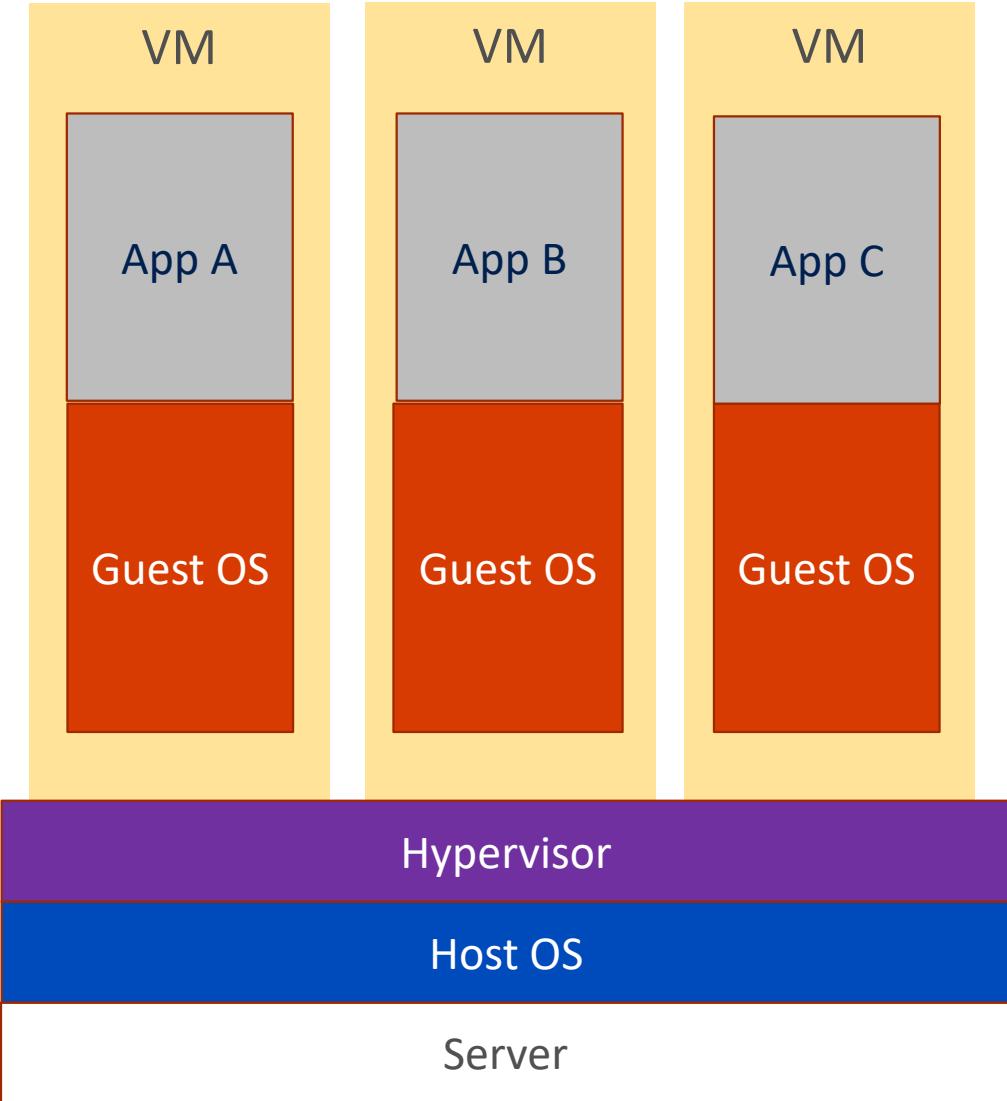
What is a Container?



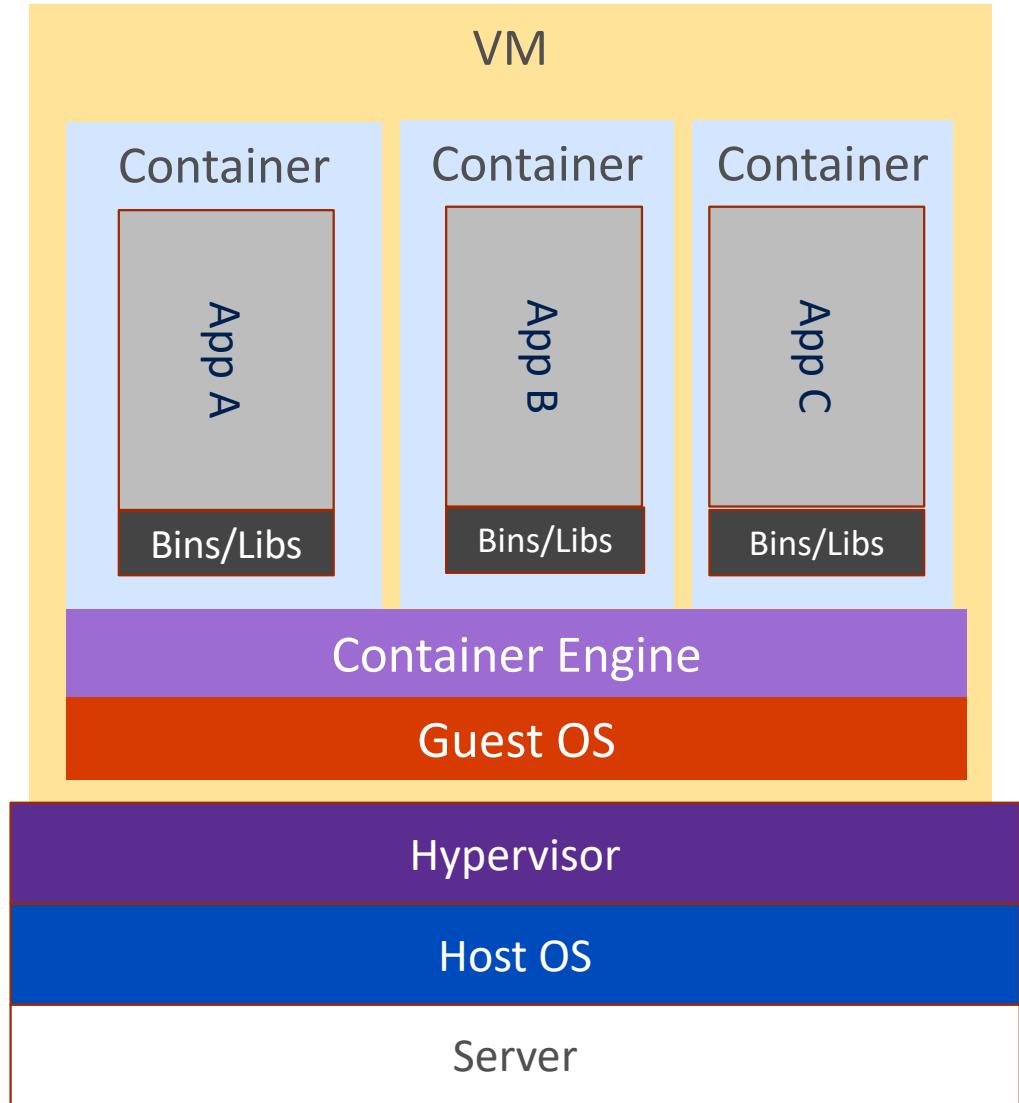
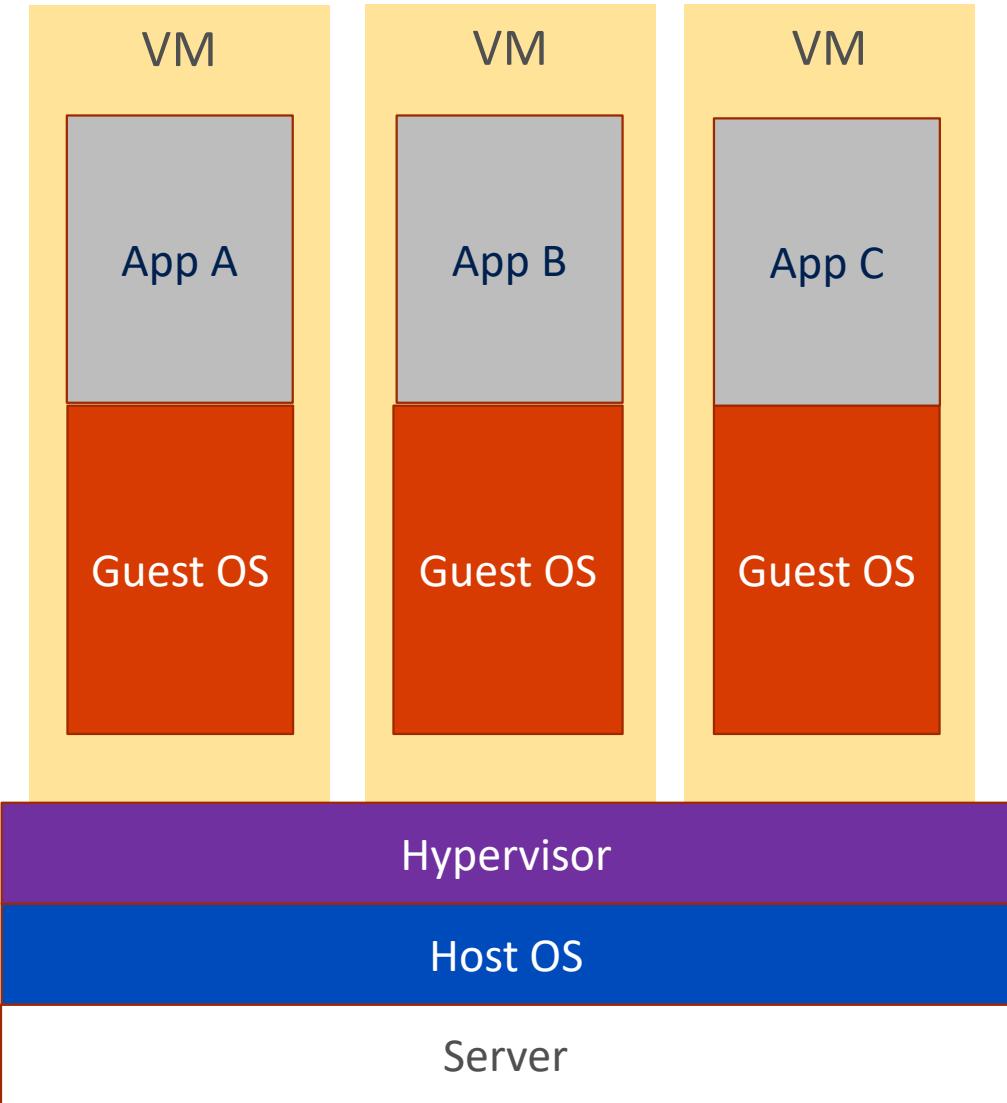
What is a Container?



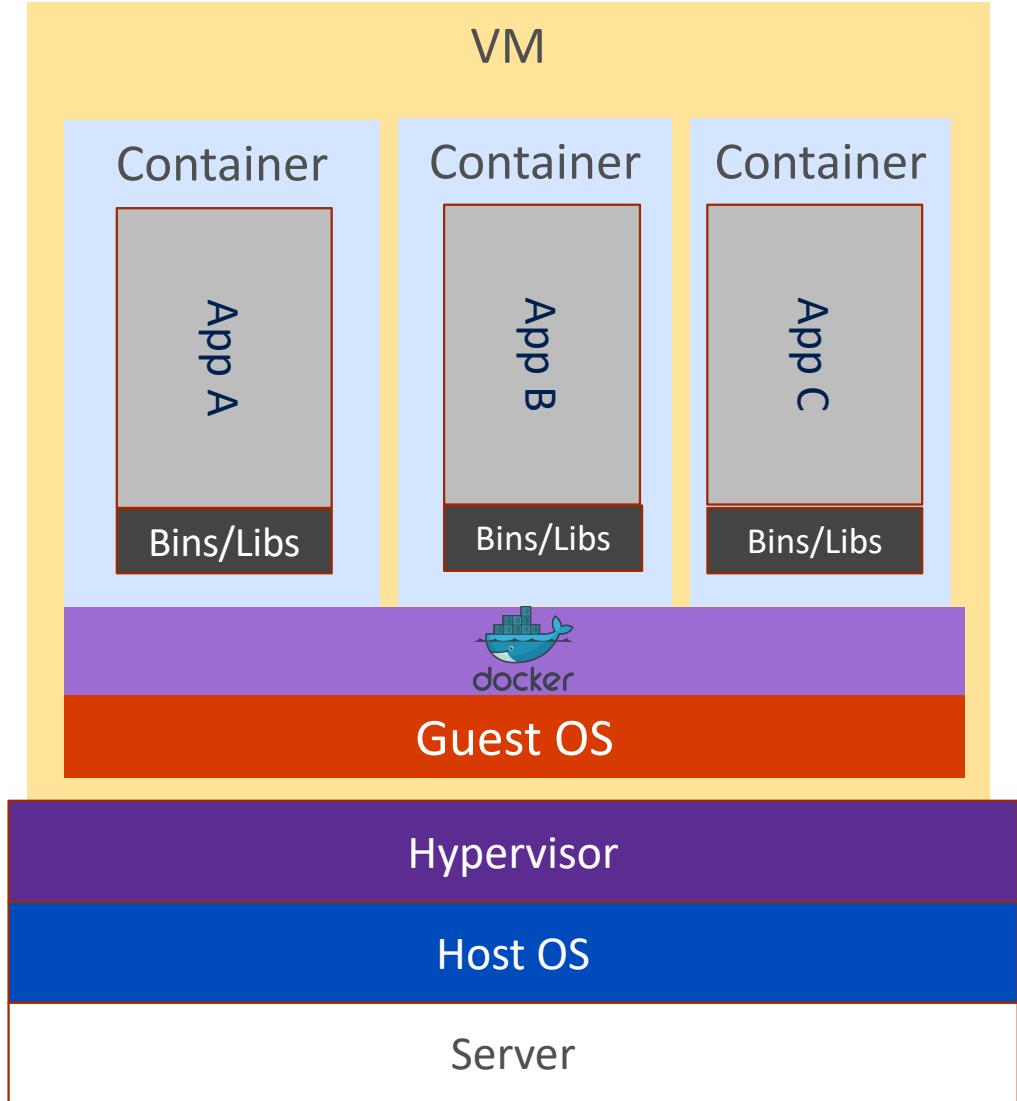
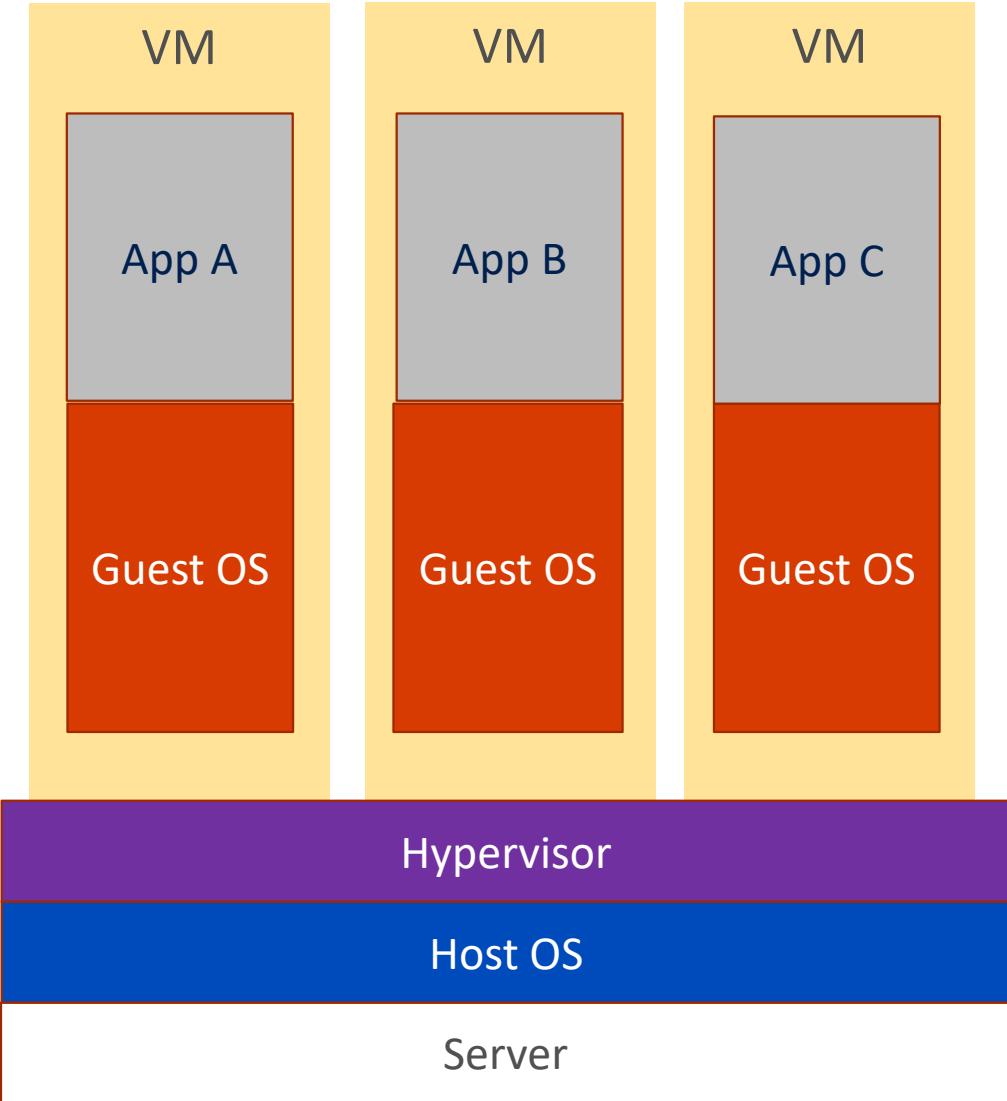
What is a Container?



What is a Container?



What is a Container?



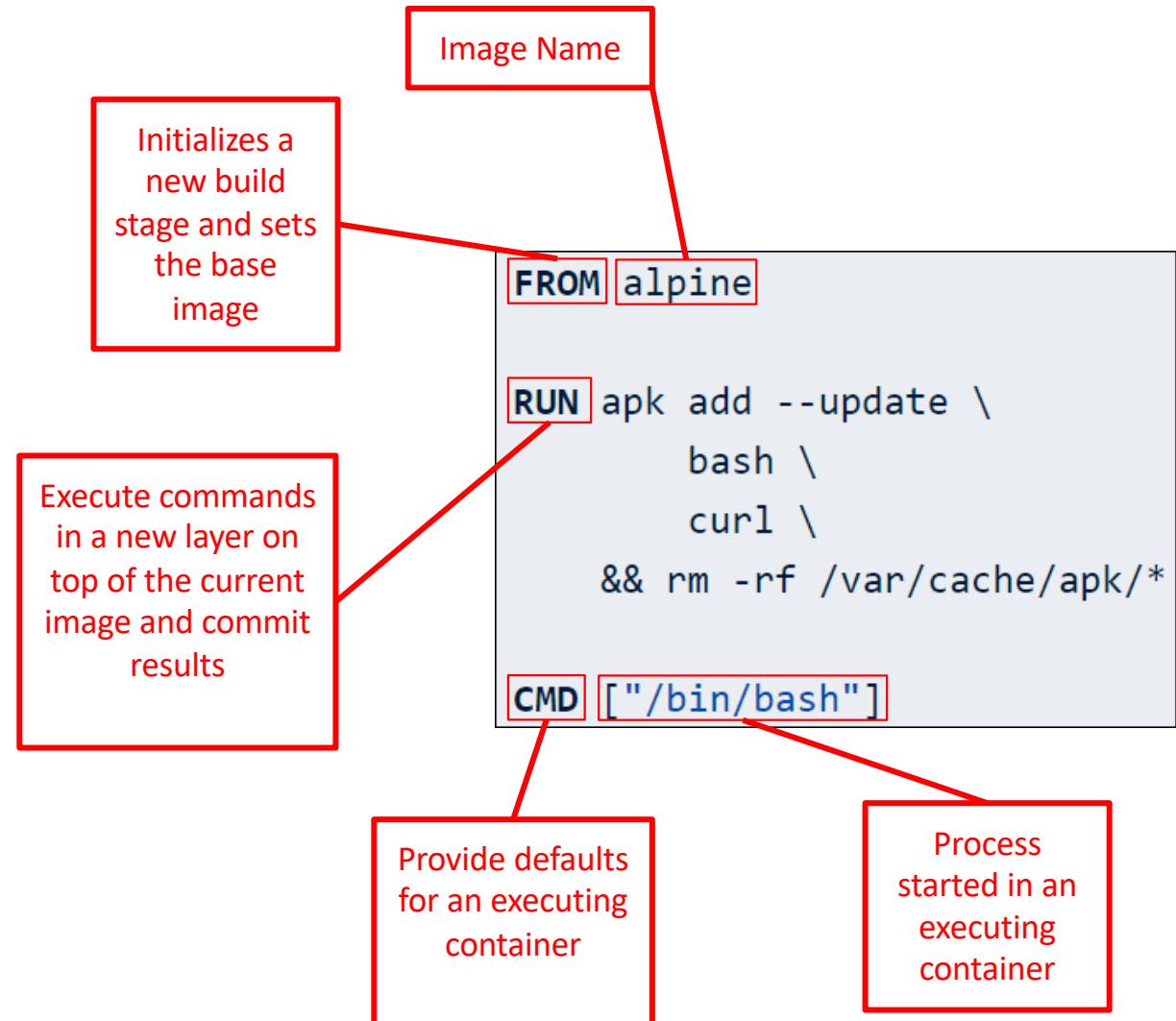
How do we create containers?

We start with the Dockerfile

Captures the steps to build a container image

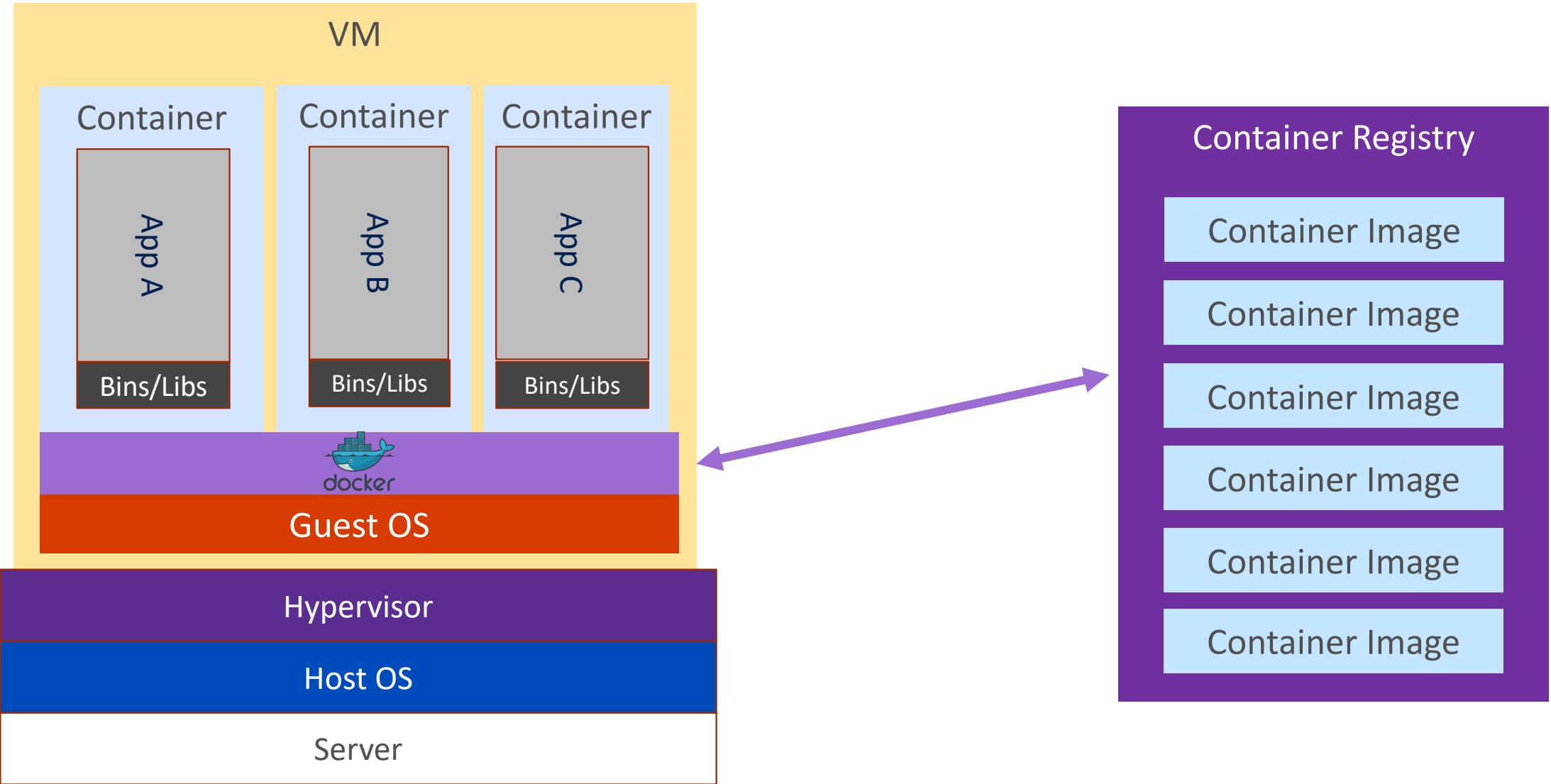
Version-able asset in your DevOps flows

Configuration-as-Code

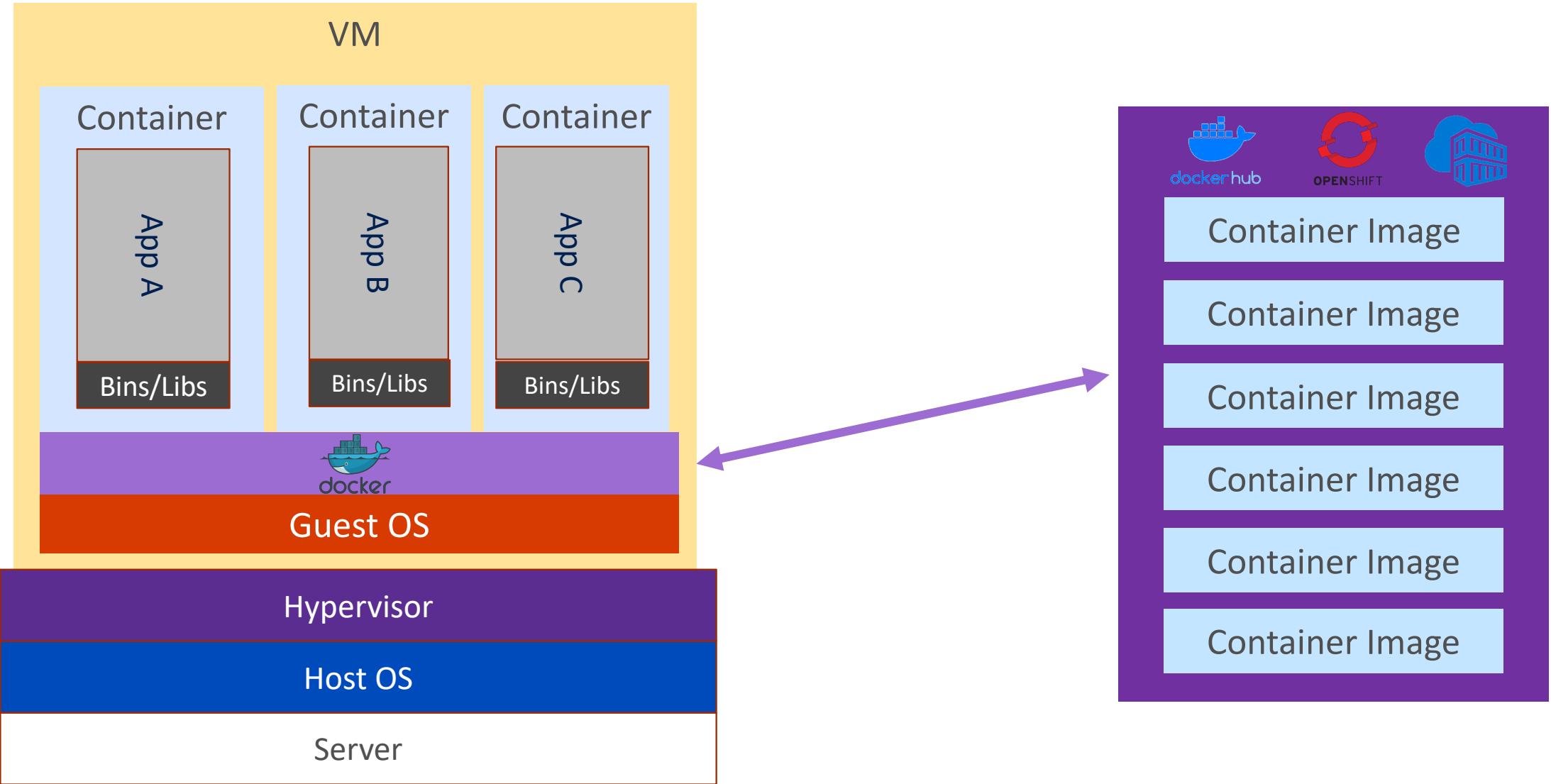


Now we need a container
registry...

What is a Container Registry?



What is a Container Registry?

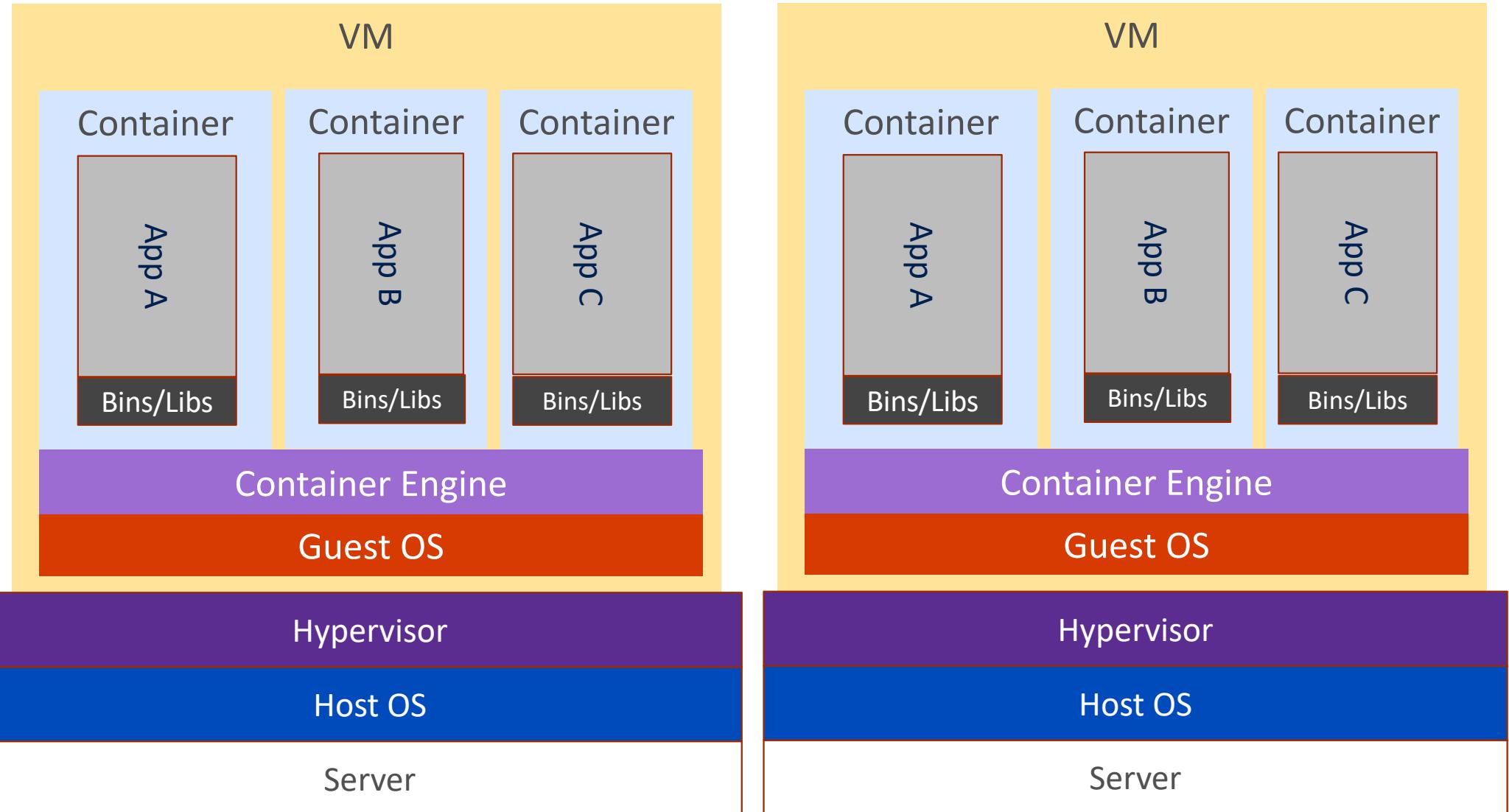


Ok, we've got containers.

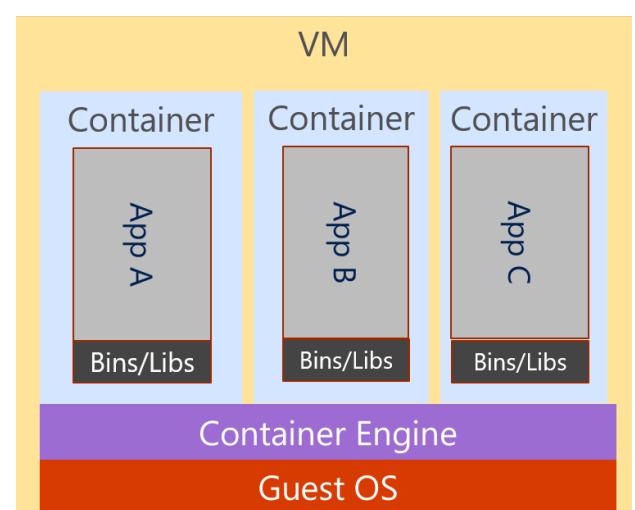
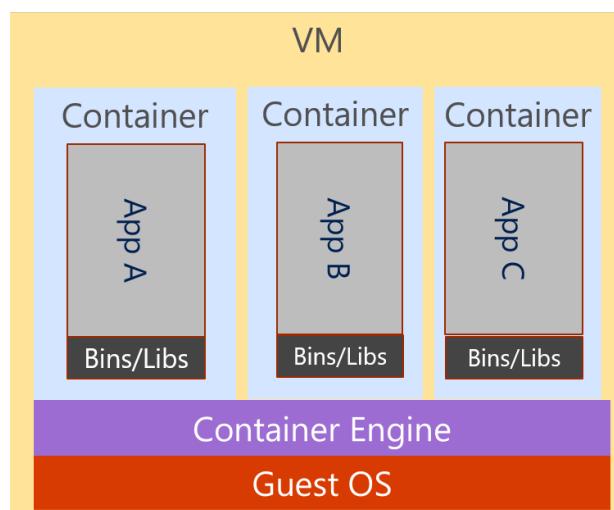
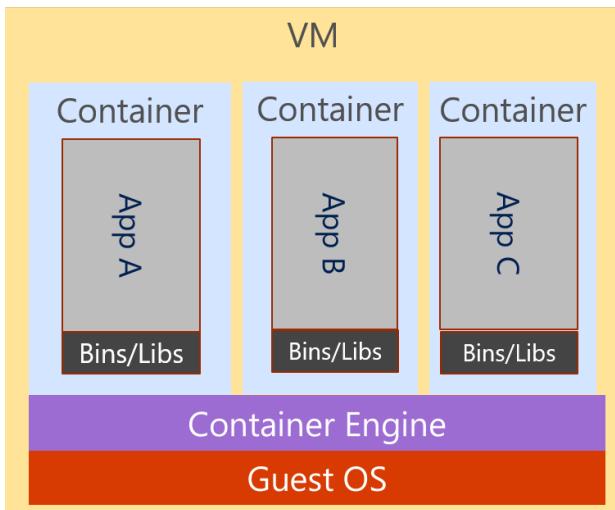
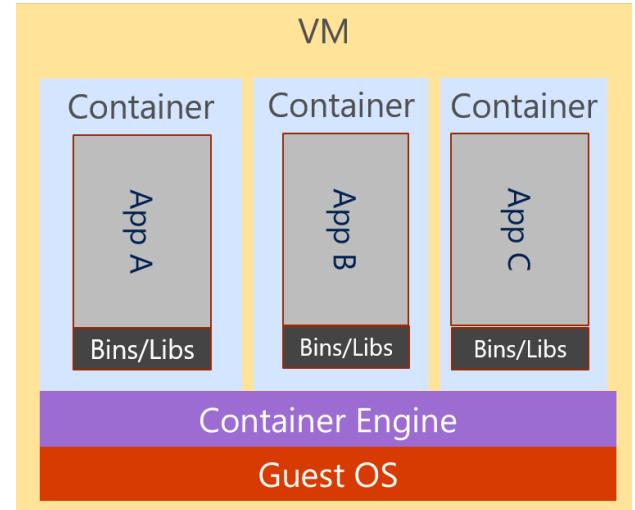
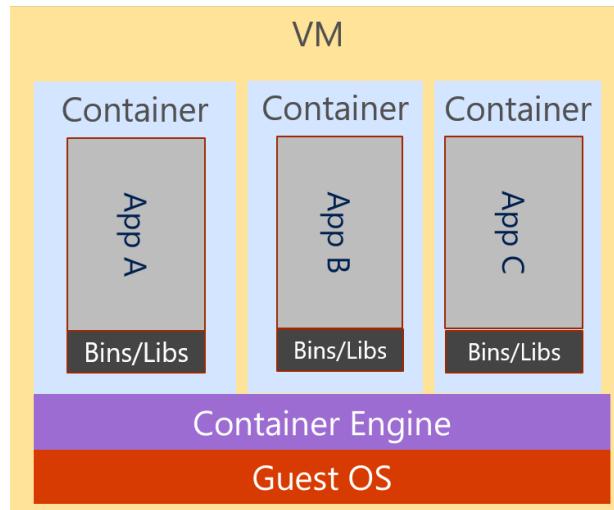
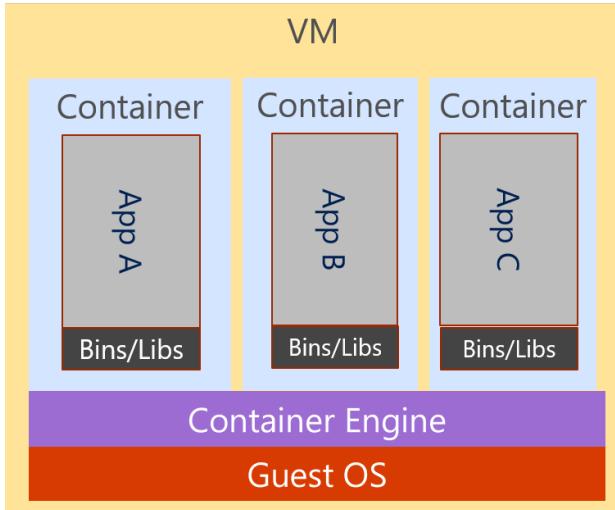
And a registry.

Now what?

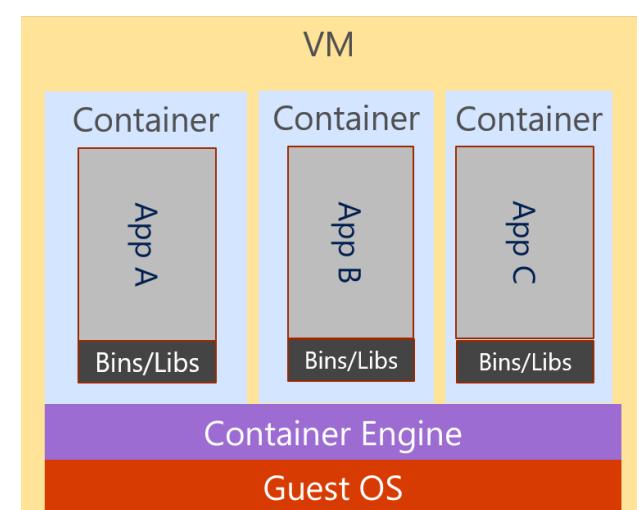
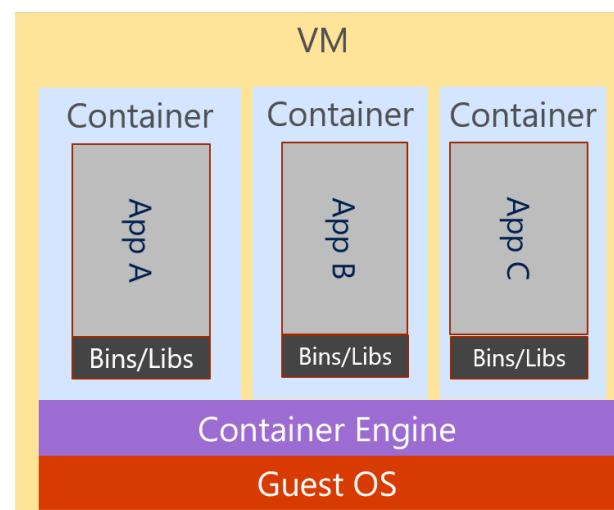
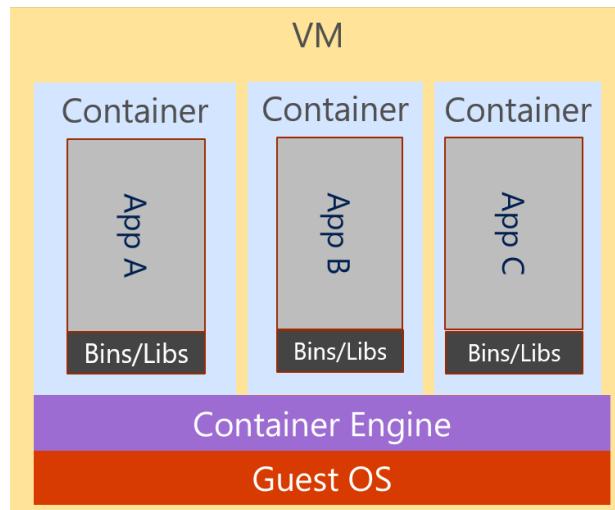
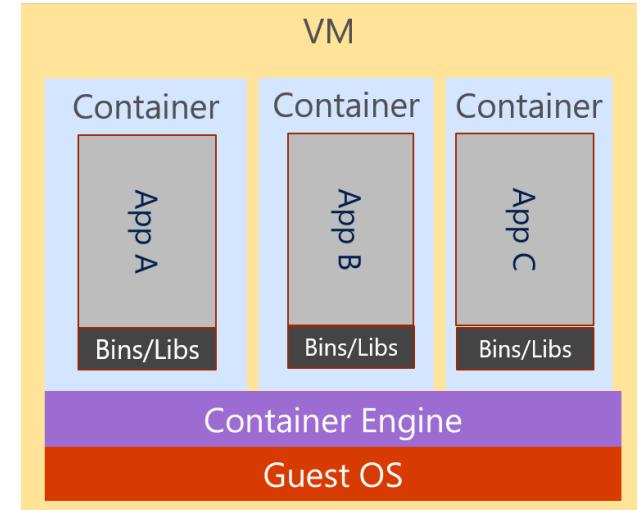
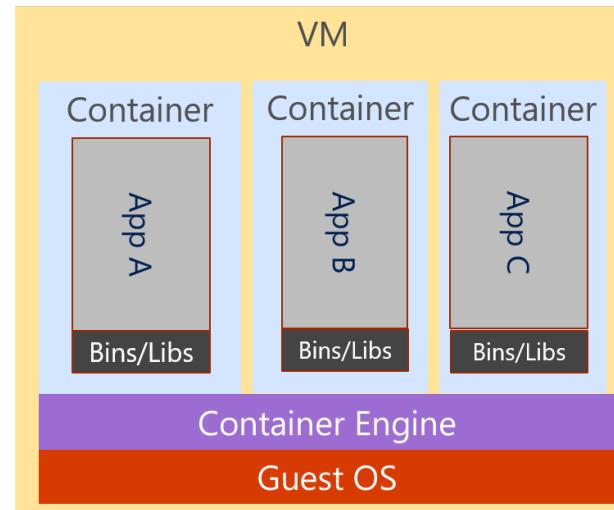
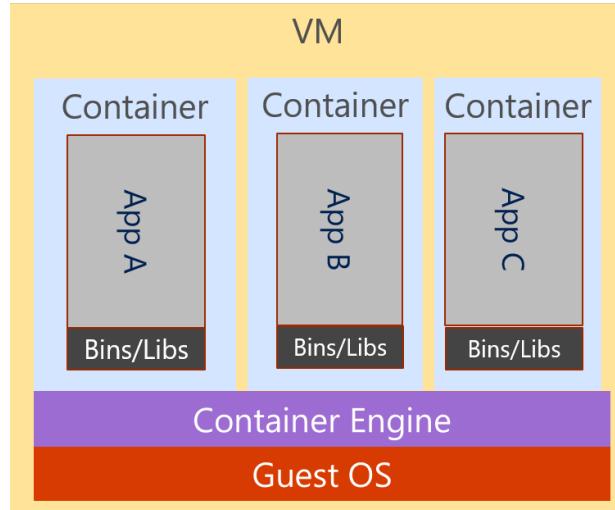
We are going to want more VMs with containers!



And more... and more!



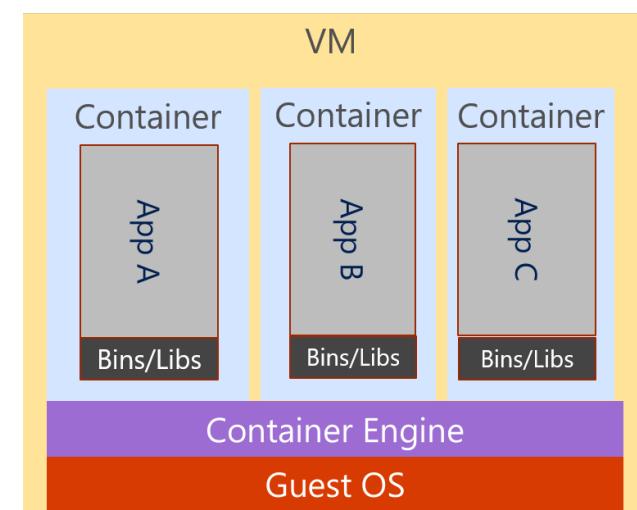
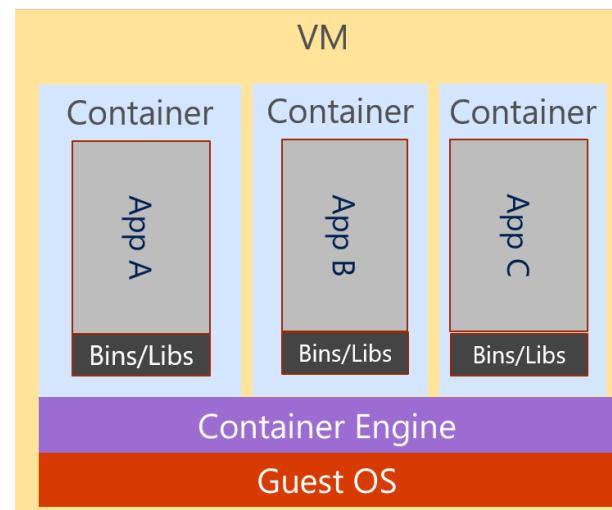
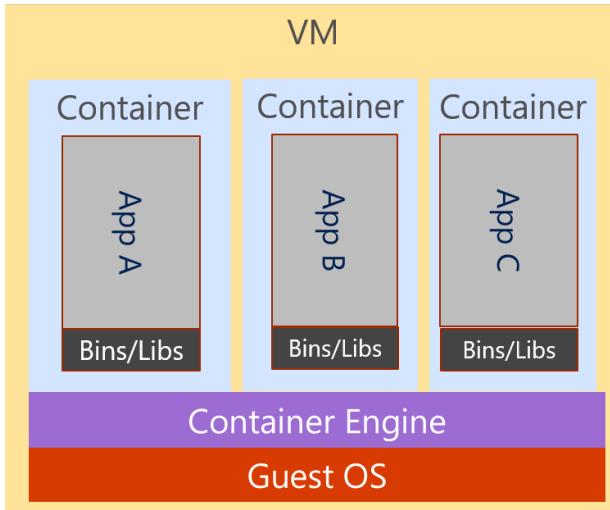
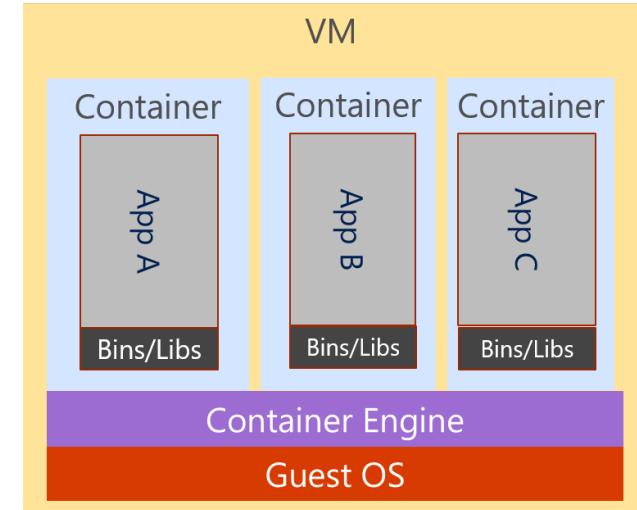
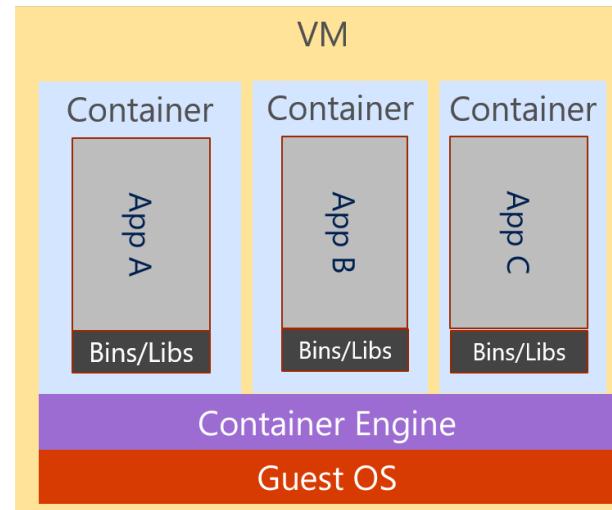
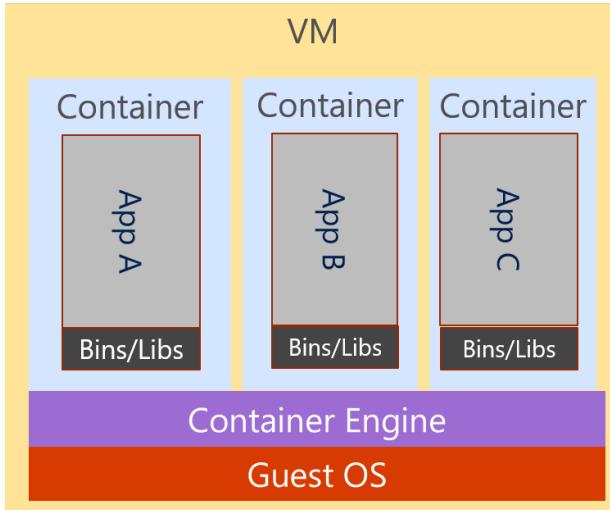
And more... and more... CHAOS!!!



Introducing the container orchestrator!

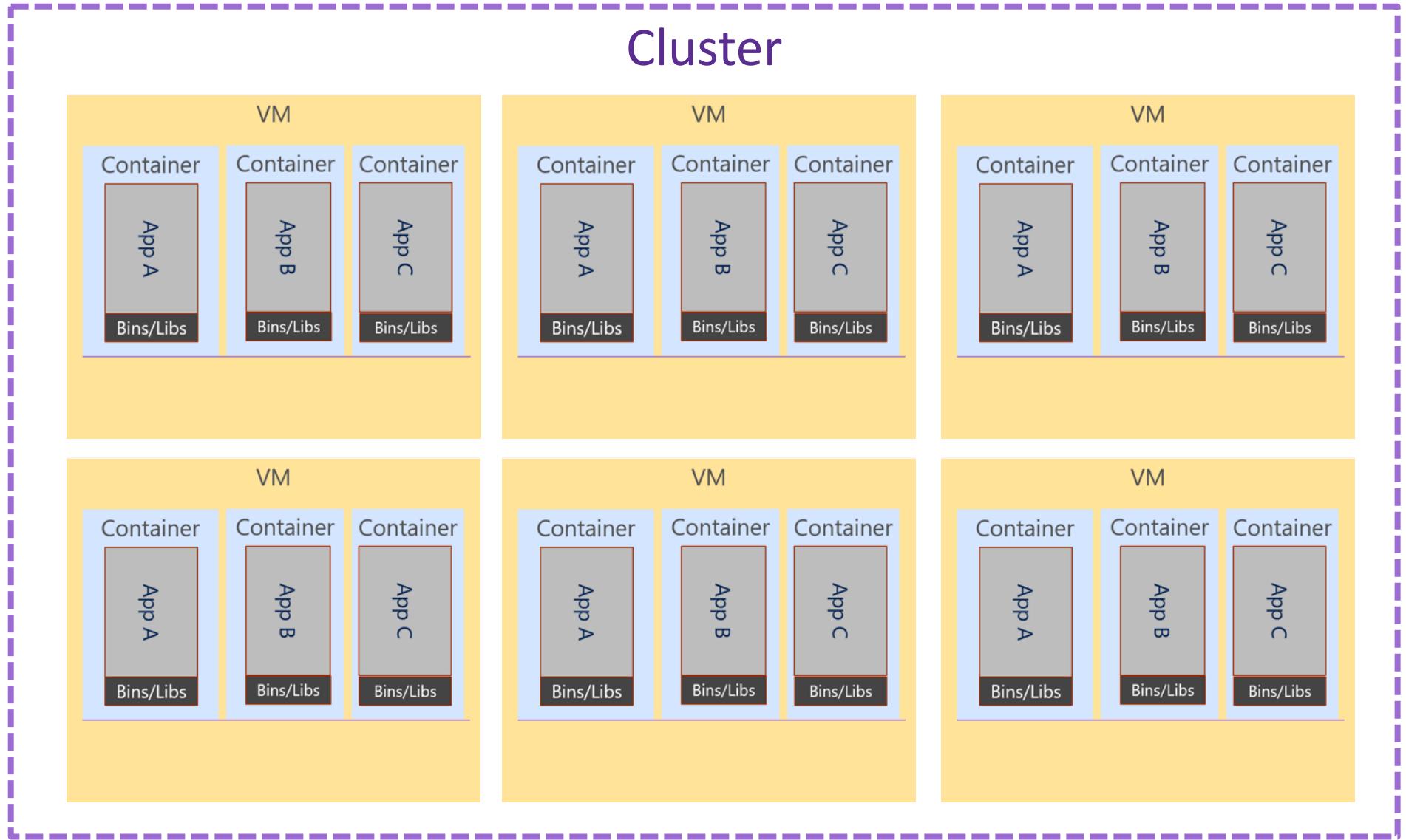
what is a Container Orchestrator?

Container Orchestrator

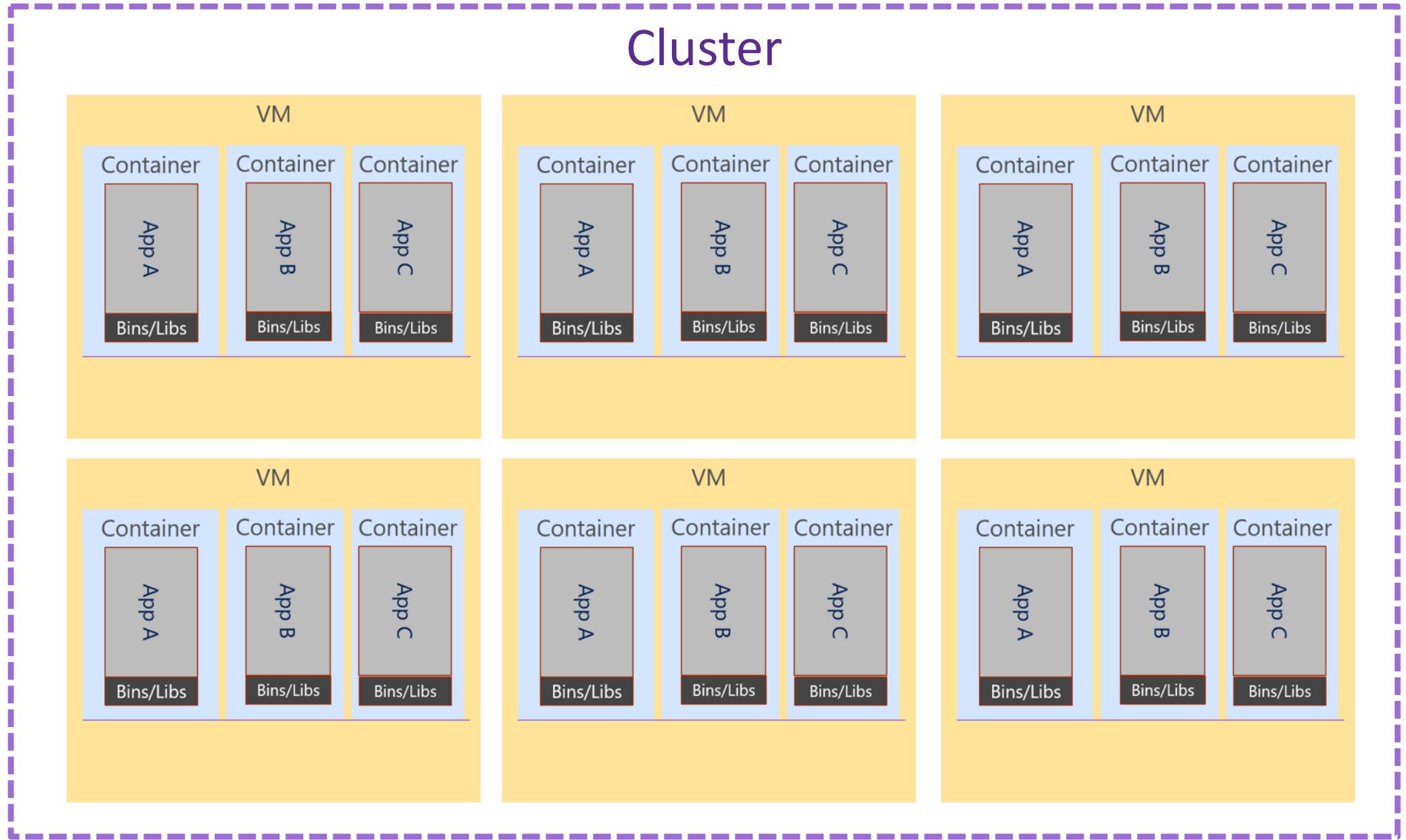


What is a Container Orchestrator?

Container Orchestrator



What is a Container Orchestrator?



Container Orchestrator Leaders



What is Kubernetes?

"Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."

Kubernetes comes from the Greek word **κυβερνήτης**, which means *helmsman* or *ship pilot*, ie: the captain of a container ship.



Kubernetes has a Master & Cluster of “Nodes”



Azure Kubernetes Service (AKS)

AKS reduces the complexity and operational overhead of managing Kubernetes by offloading much of that responsibility to Azure.

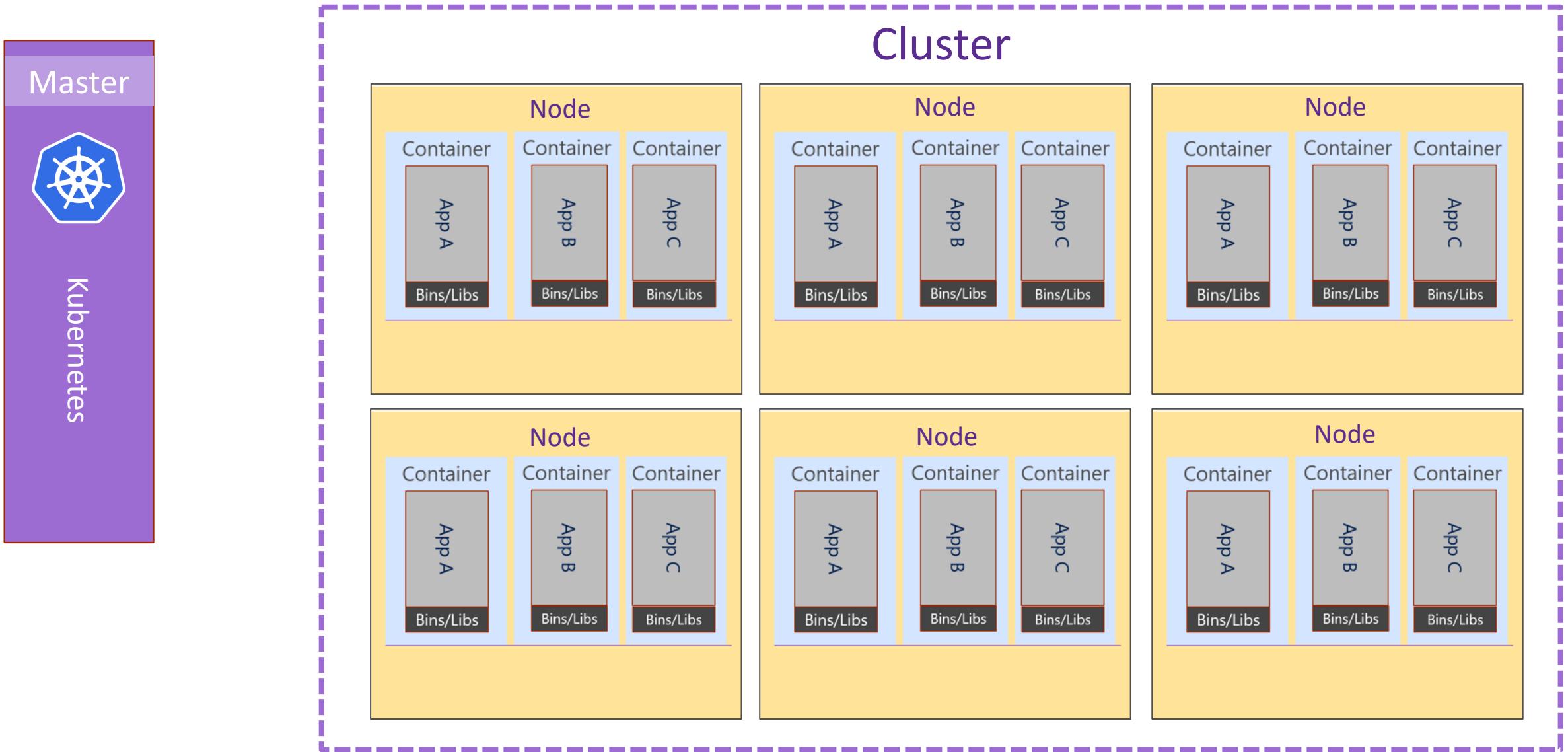


You only pay for the worker nodes within your clusters, not for the master nodes

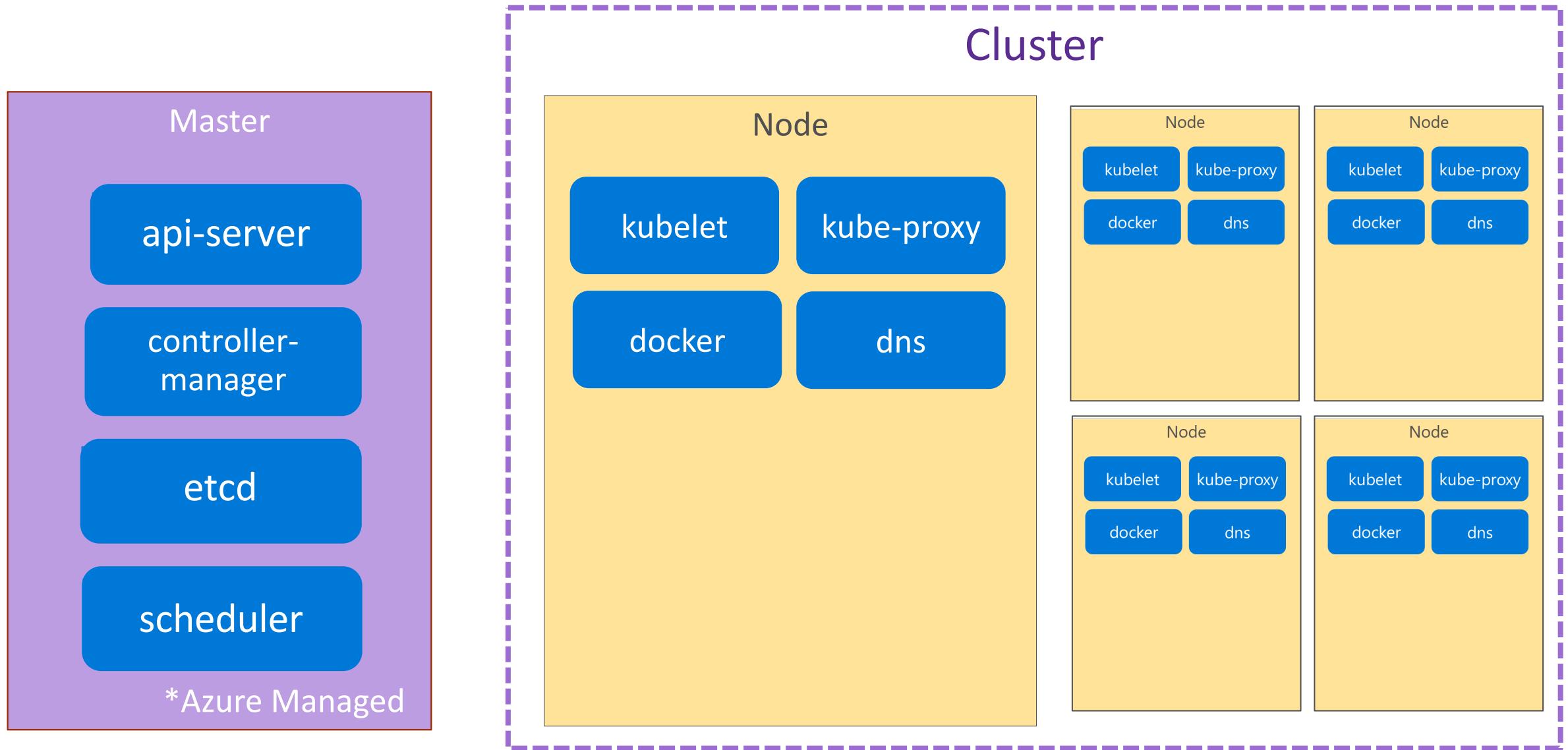
Kubernetes at 10,000 ft



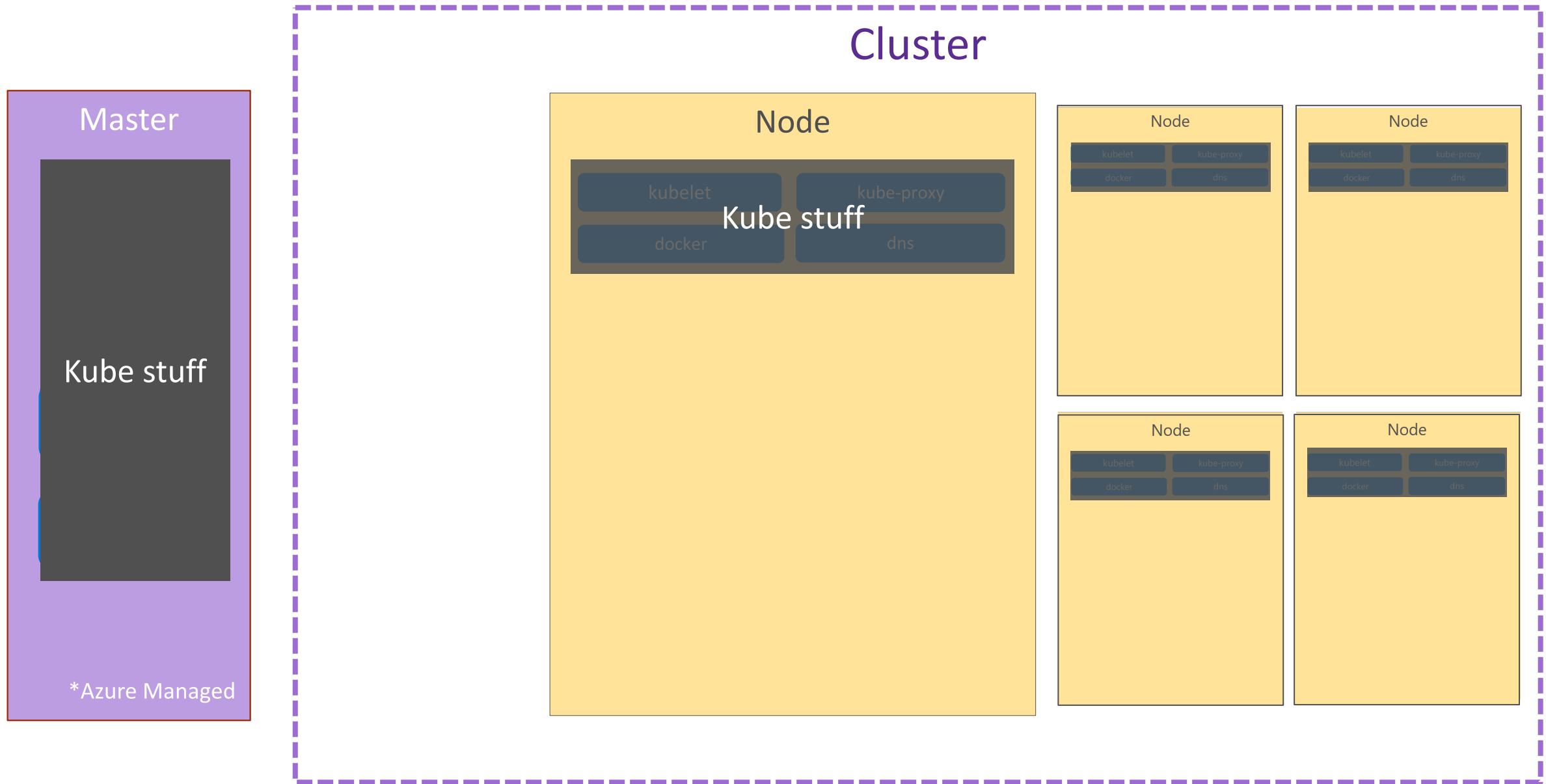
Kubernetes @ 10,000ft



Kubernetes Architecture Components

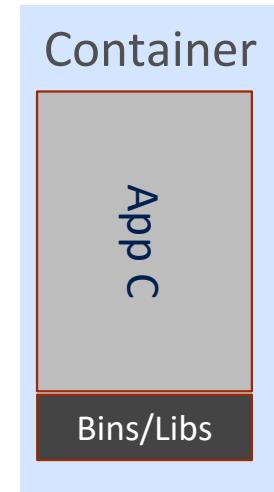
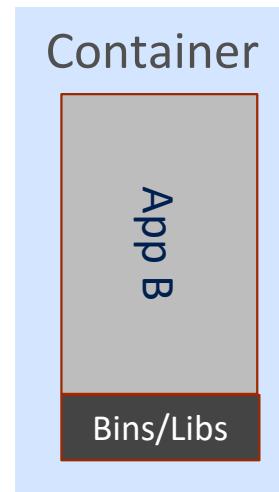
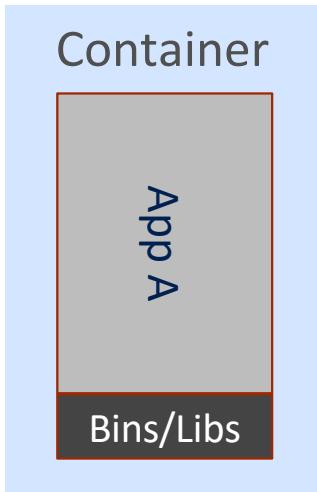


Kubernetes Architecture Components



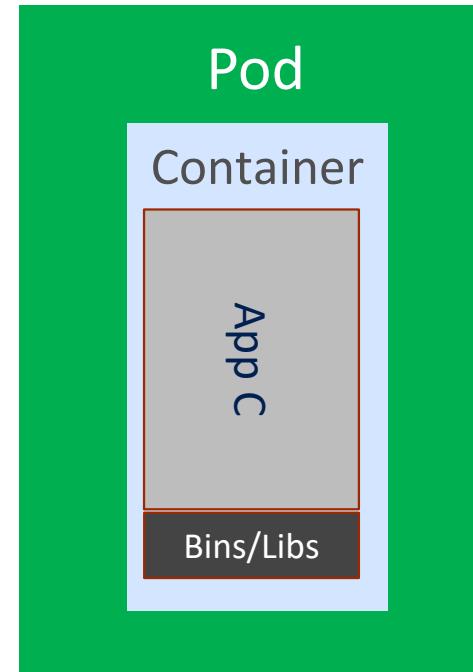
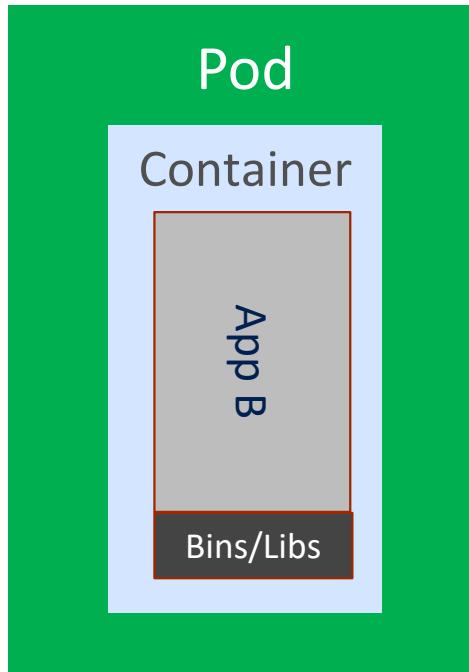
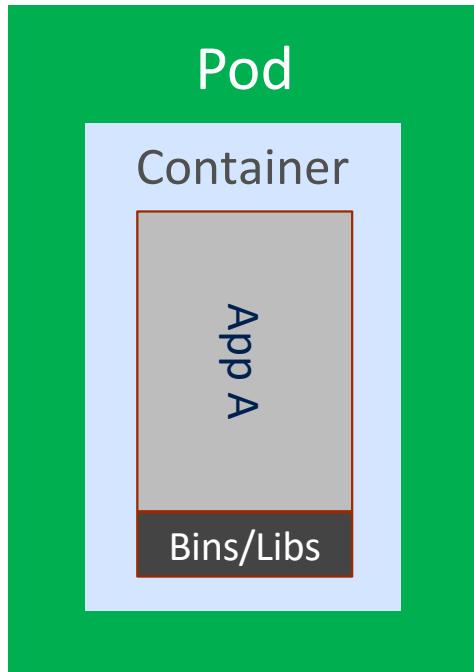
Kubernetes Architecture Components

Where do the Containers go?



Kubernetes Architecture Components

Introducing.... Pods!



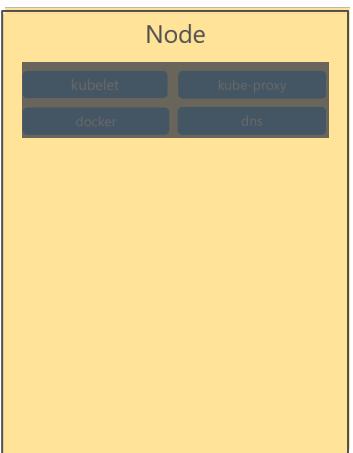
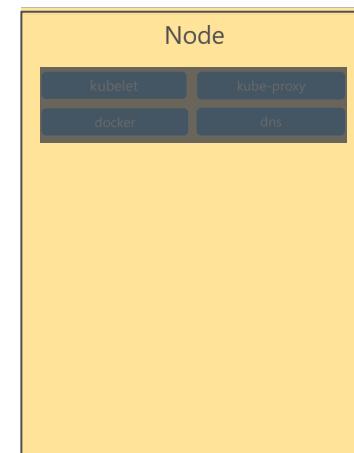
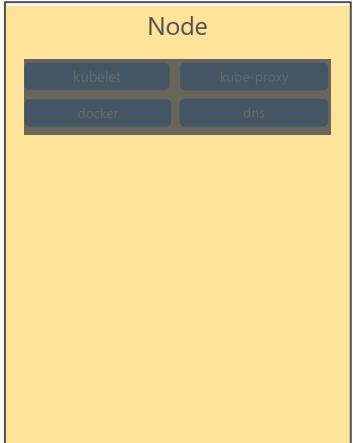
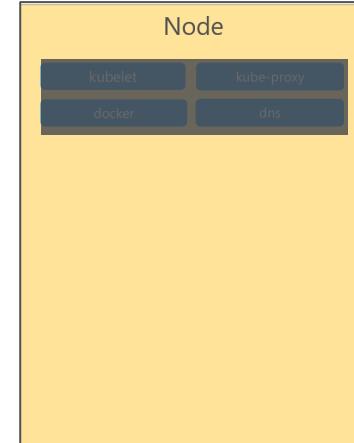
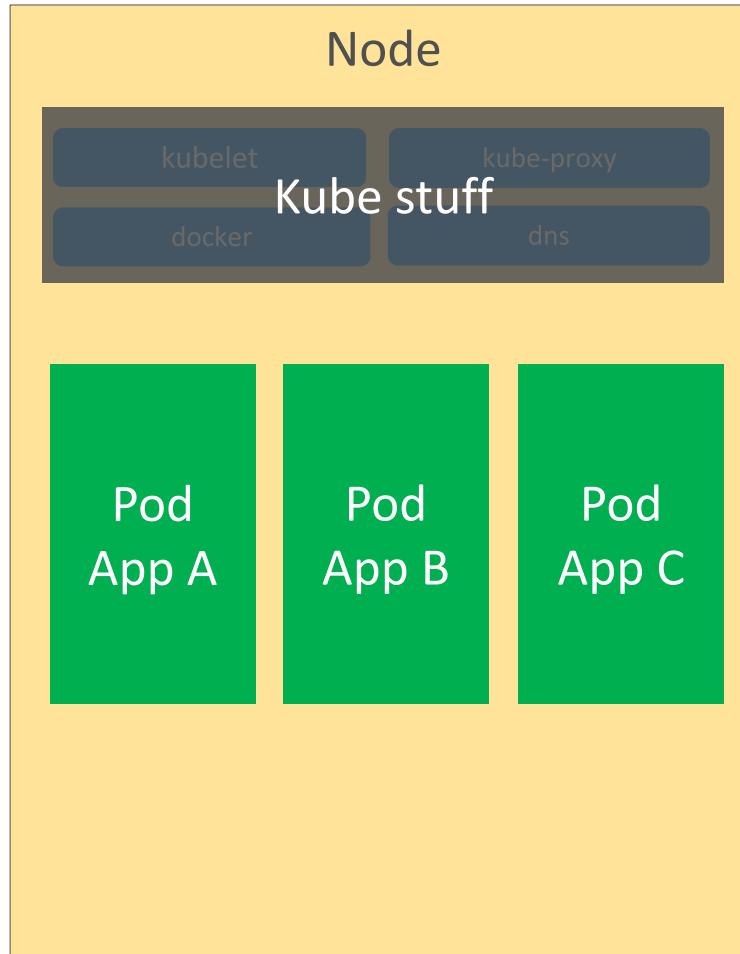
Containers go inside Pods!

Kubernetes Architecture Components

Pods



Cluster

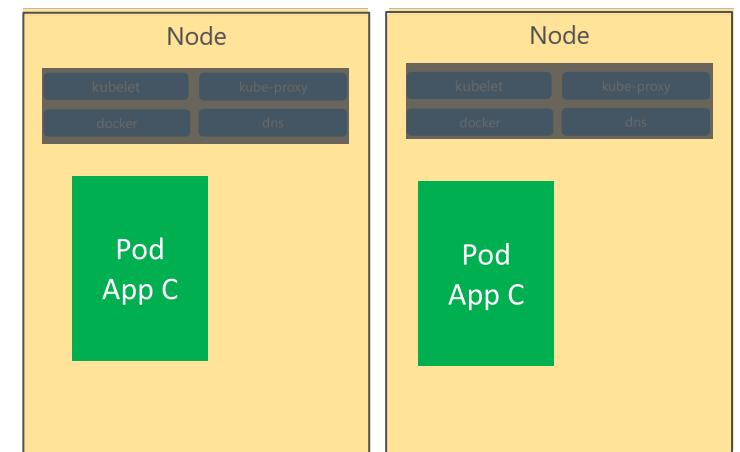
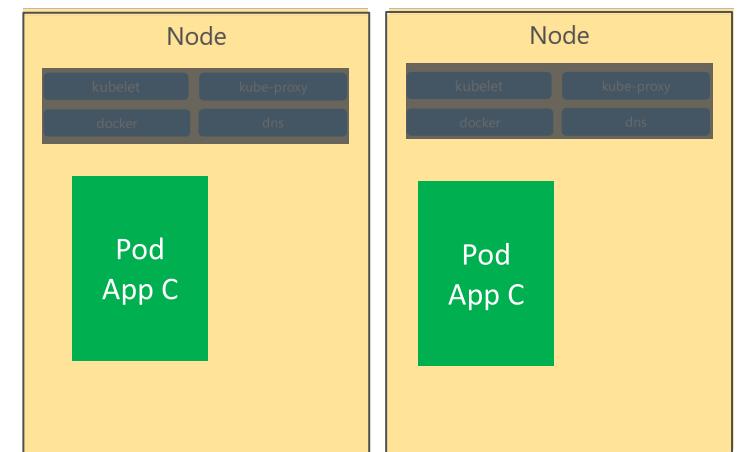
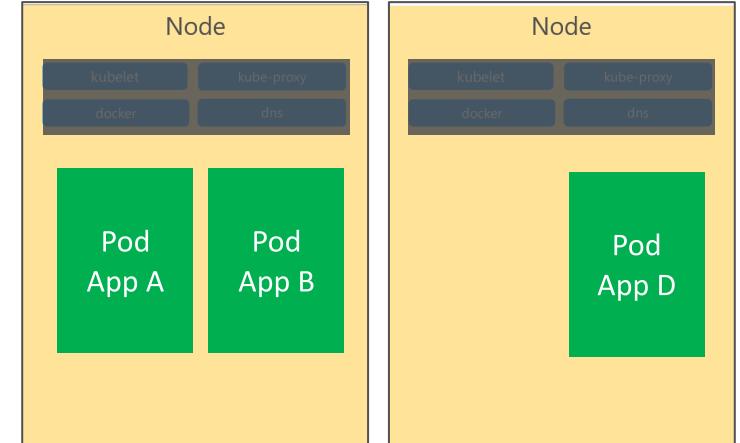
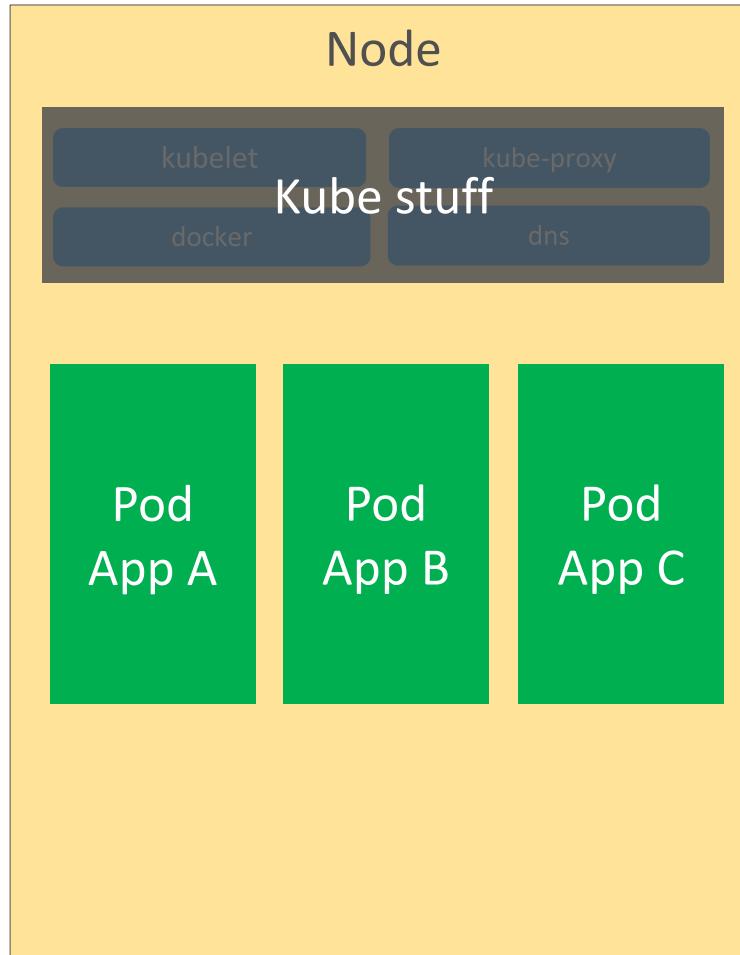


Kubernetes Architecture Components

Pods

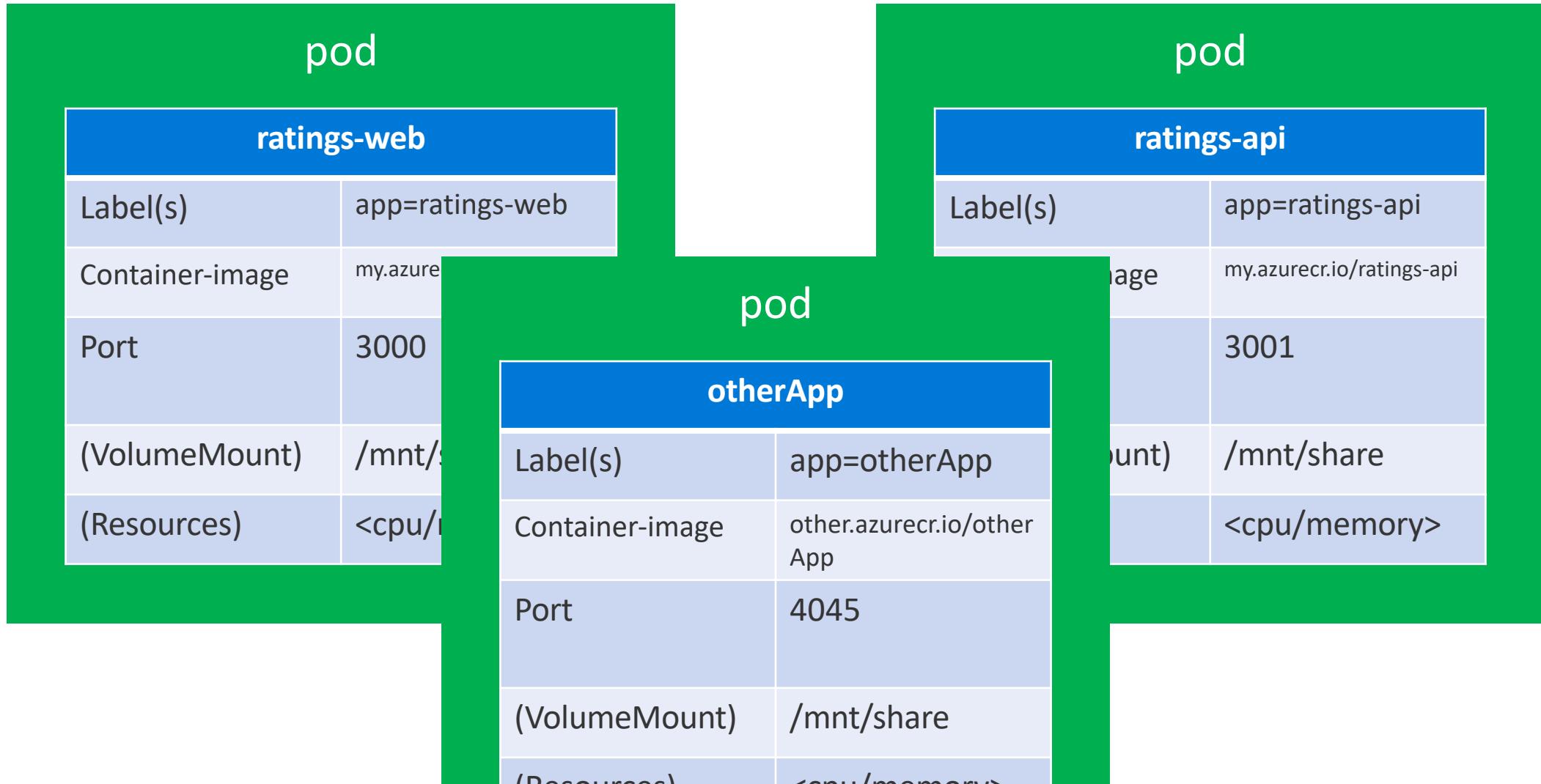


Cluster



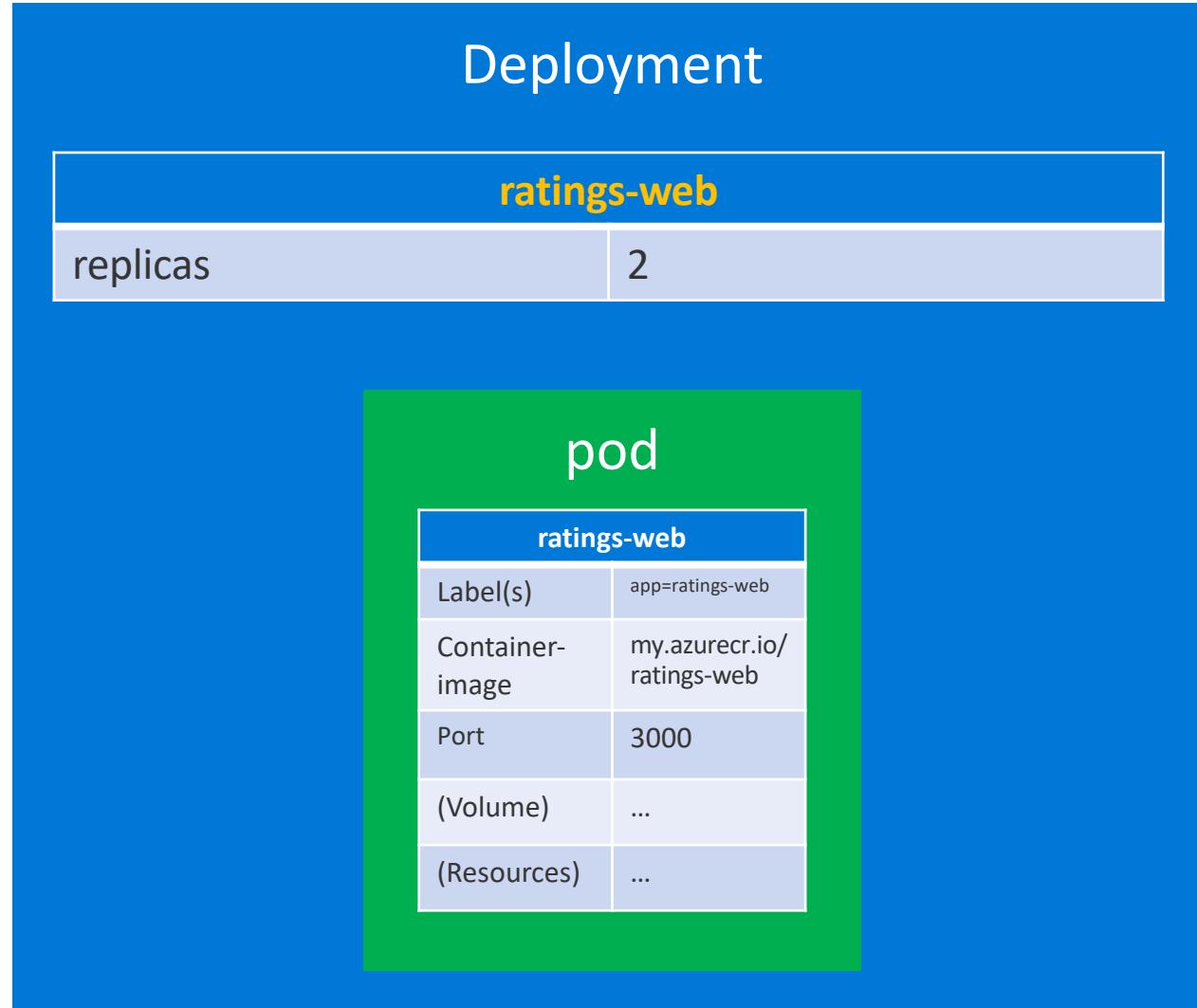
Pods – What are they?

Metadata that describes how the containers are configured.

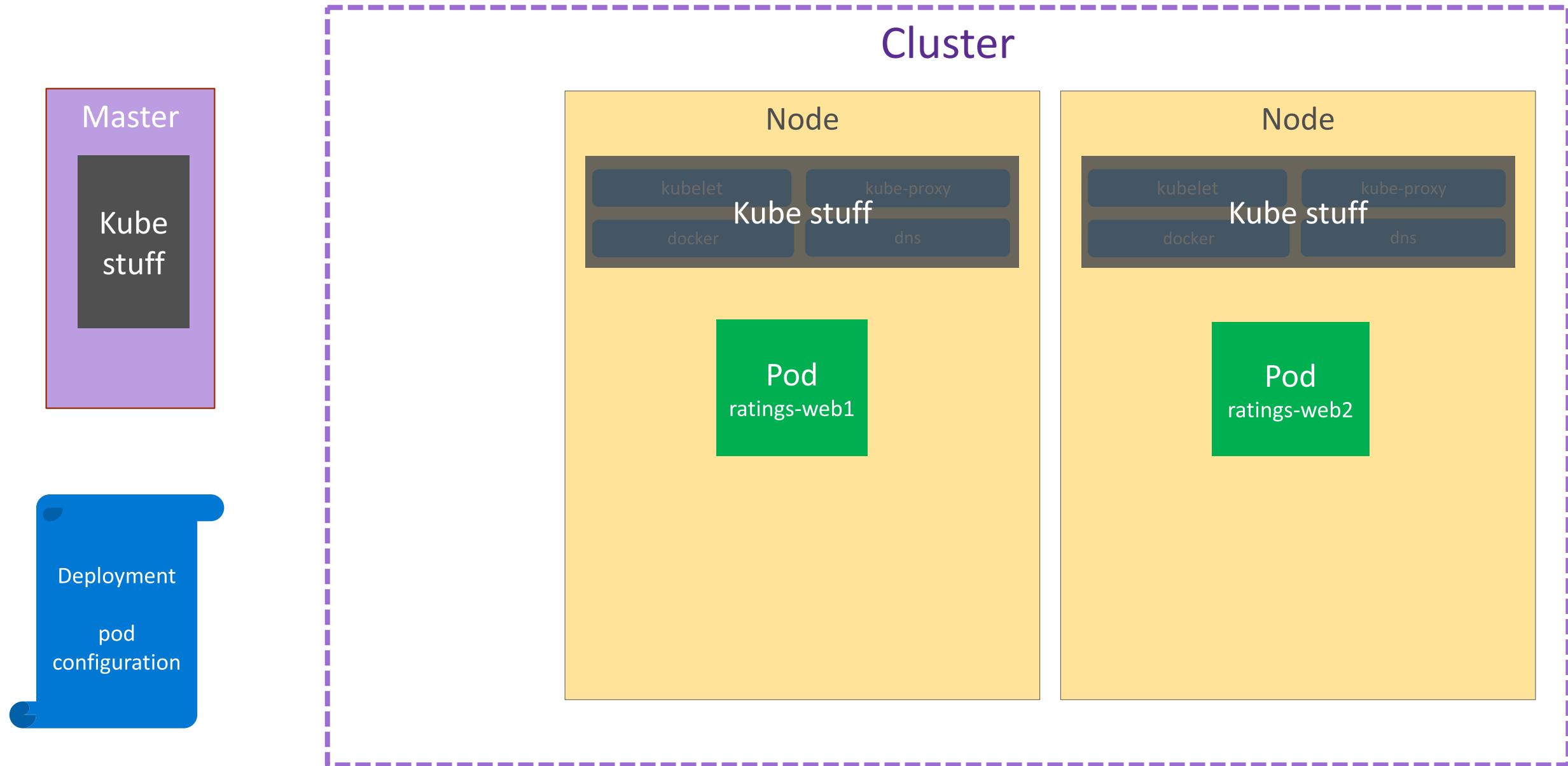


Deployments

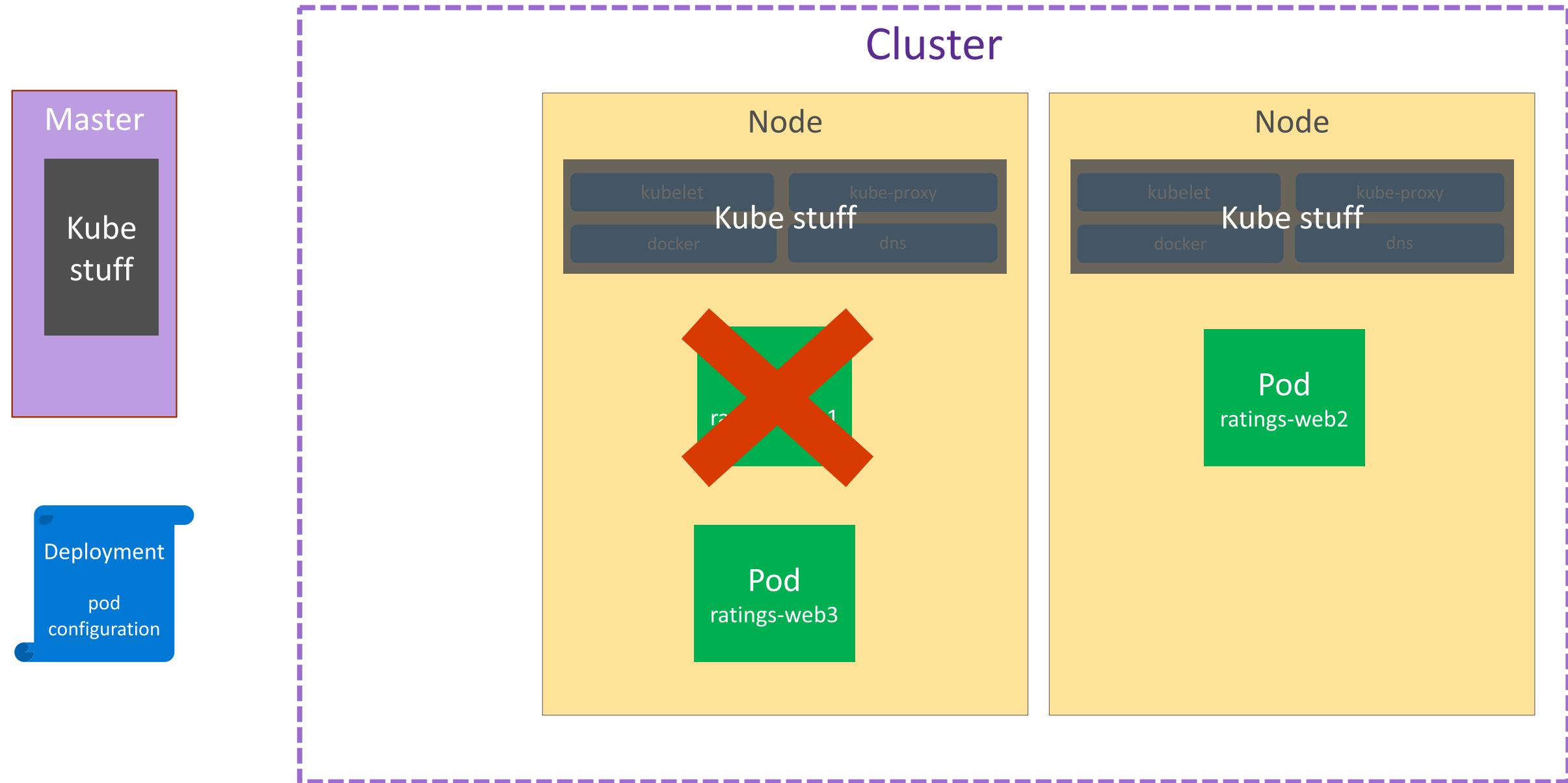
Metadata describes how to deploy a pod



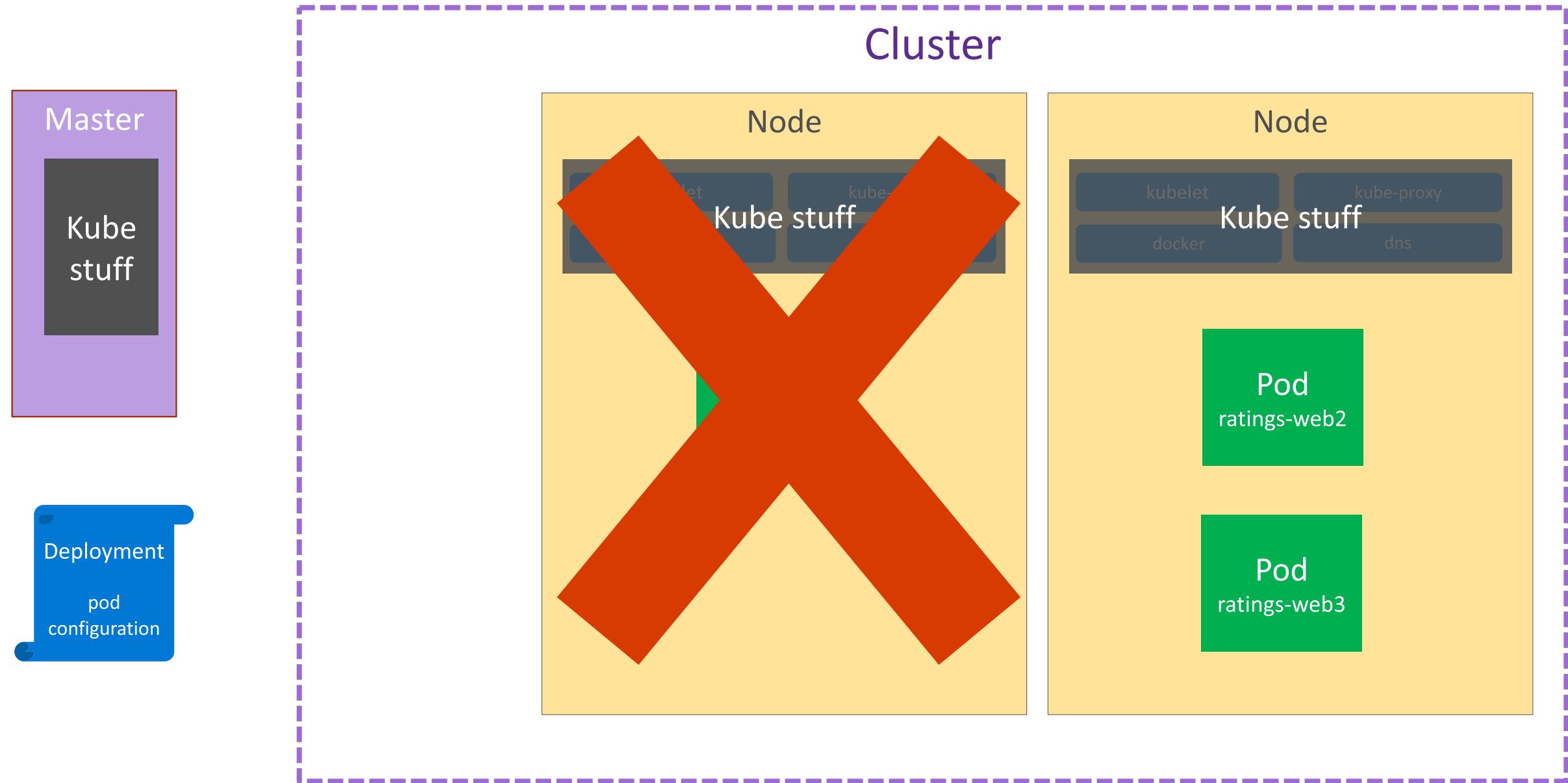
Kubernetes Deployment In Action



Kubernetes Desired State In Action – Example 1

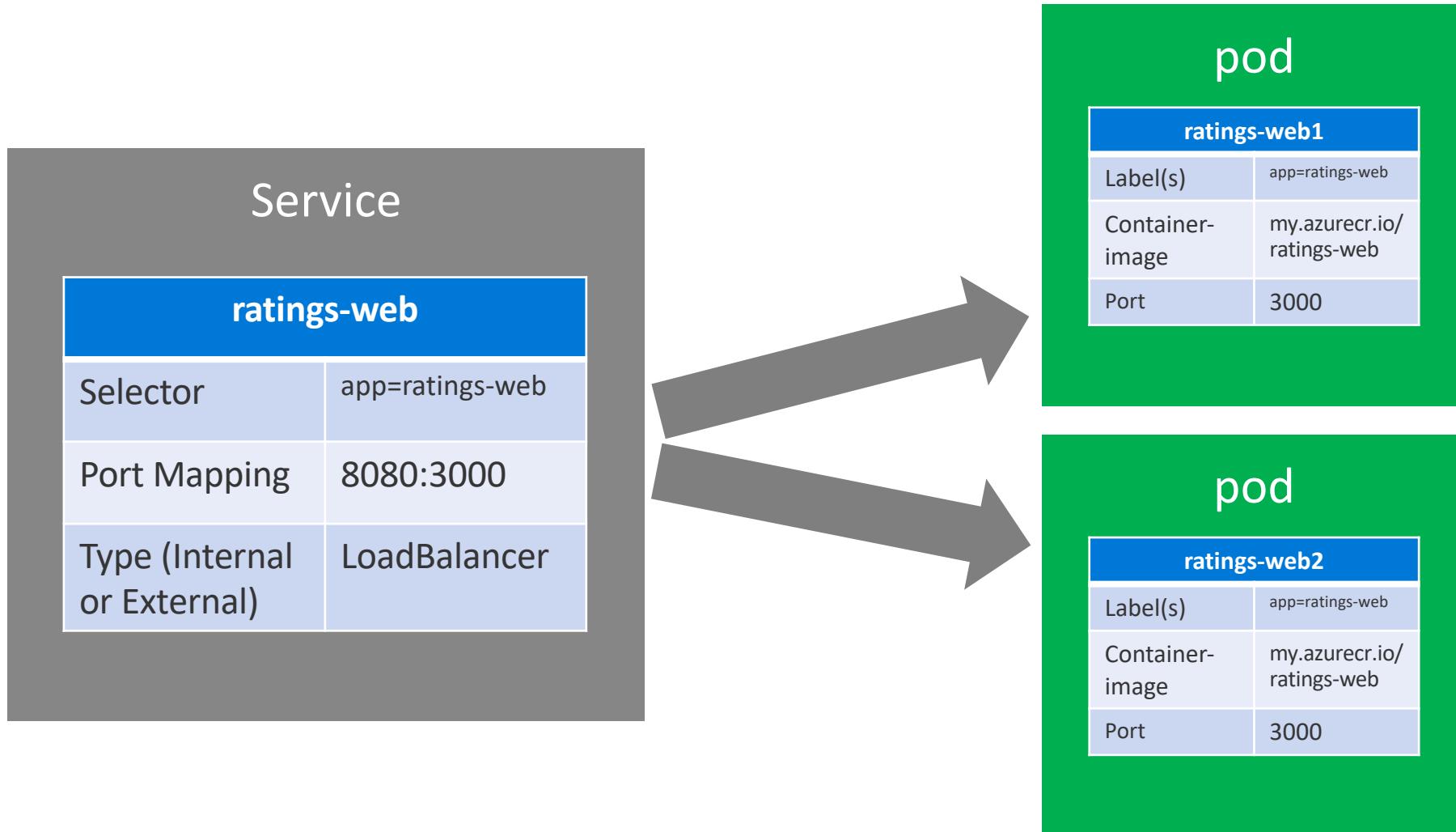


Kubernetes Desired State In Action – Example 2



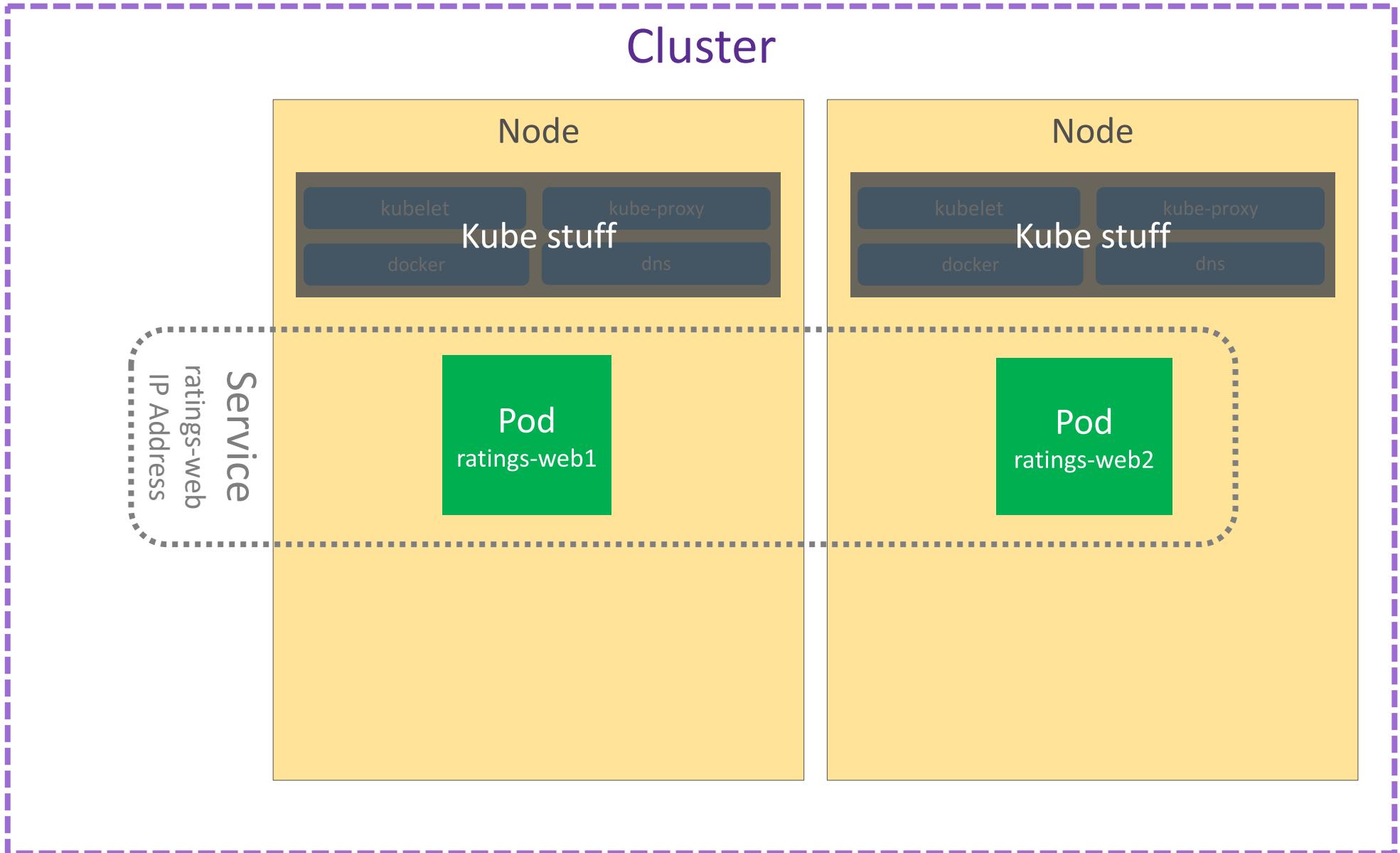
Services

~~What data does the dashboard~~ What data describes how to reach the pods



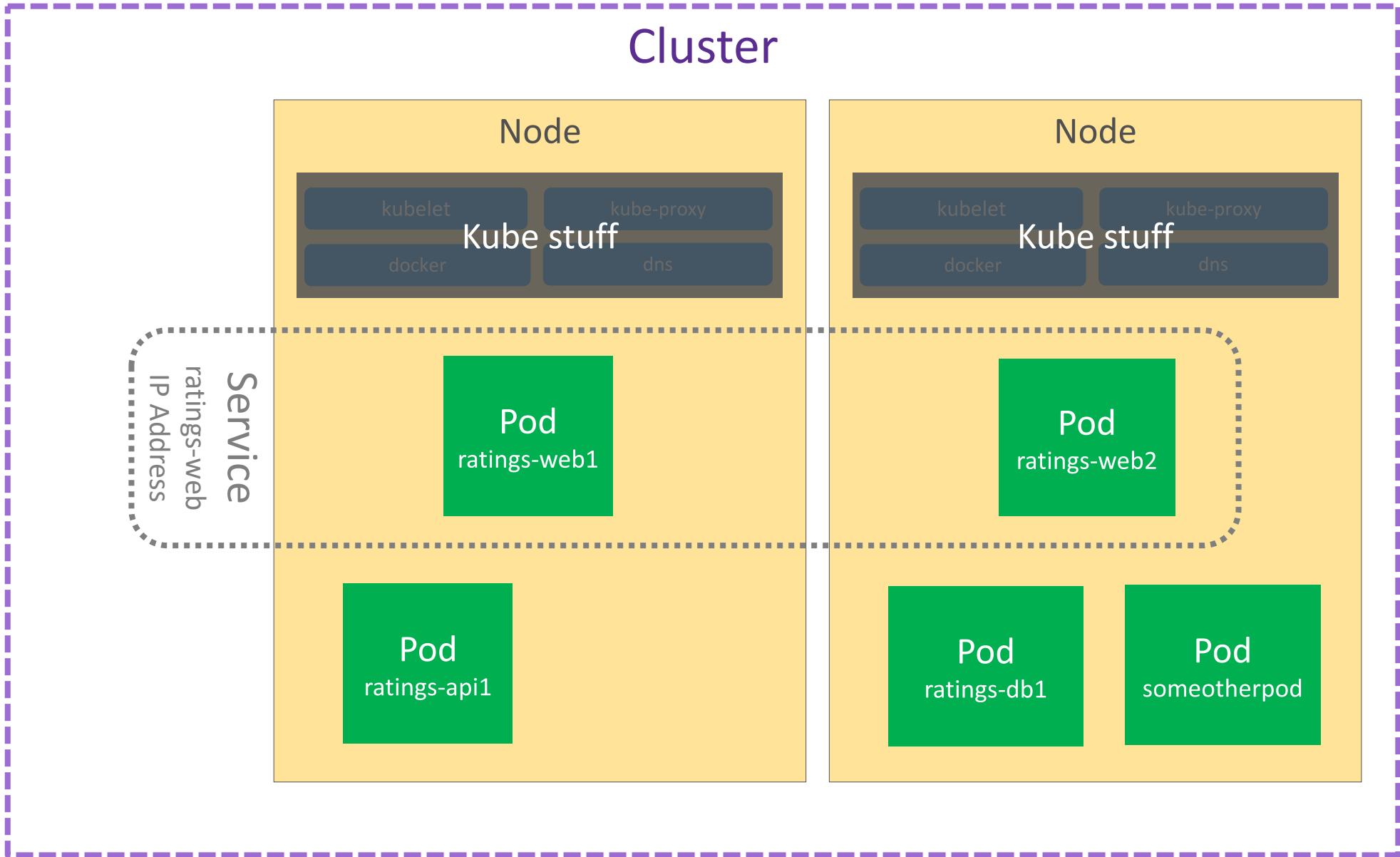
Kubernetes Service Deployment In Action

Services



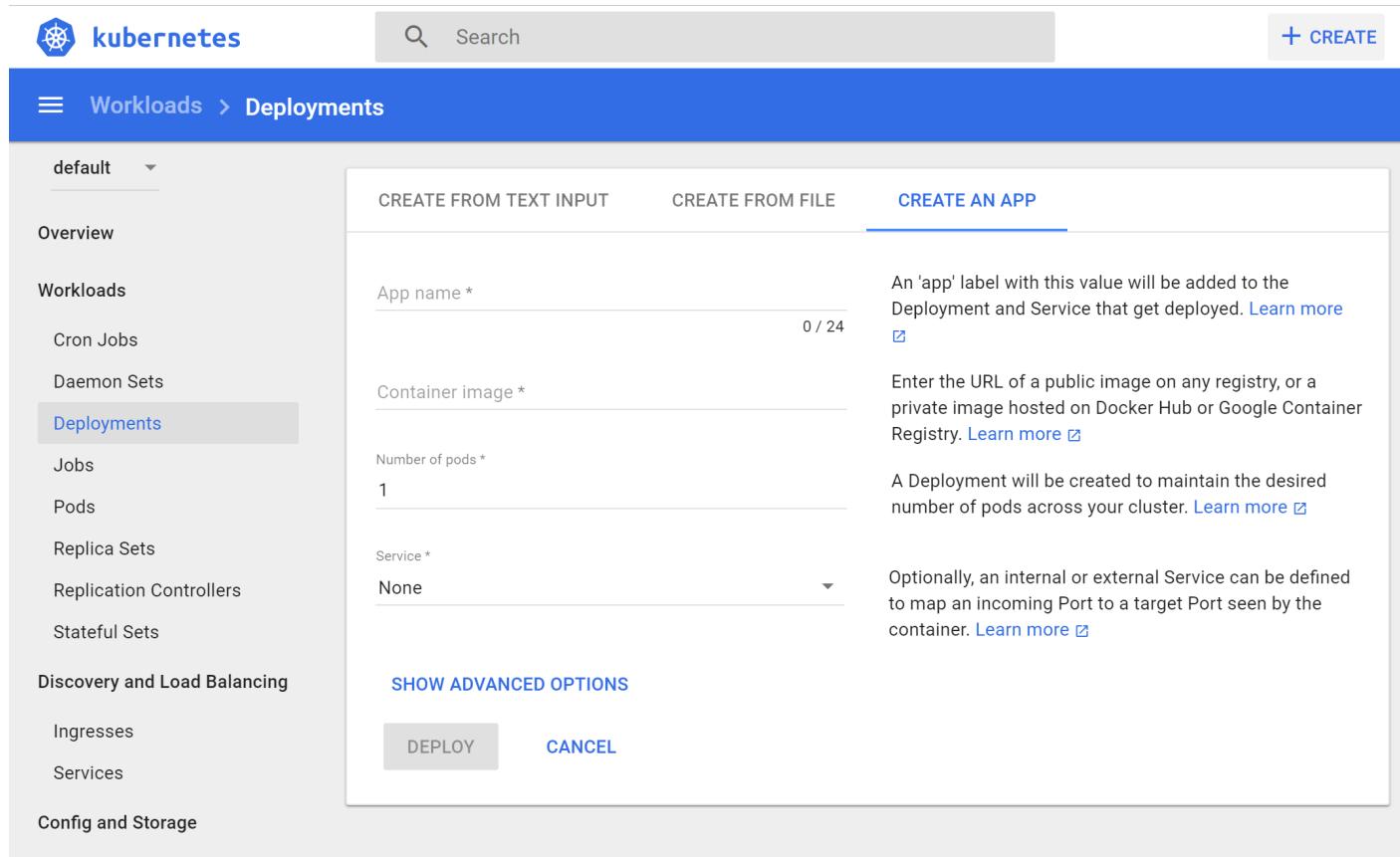
Kubernetes Service Deployment In Action

Services



Deployments in Kubernetes

Manual via Kubernetes Portal



The screenshot shows the Kubernetes Portal's 'Workloads > Deployments' page. On the left, a sidebar lists various workloads like Workloads, Cron Jobs, Daemon Sets, and Deployments (which is selected). The main area has tabs for 'CREATE FROM TEXT INPUT', 'CREATE FROM FILE', and 'CREATE AN APP'. The 'CREATE AN APP' tab is active. It contains fields for 'App name *' (with a note about adding an 'app' label), 'Container image *' (with a note about public or private images on Docker Hub or Google Container Registry), 'Number of pods *' (set to 1), and 'Service *' (set to 'None'). Below these are 'SHOW ADVANCED OPTIONS', 'DEPLOY' (disabled), and 'CANCEL' buttons.

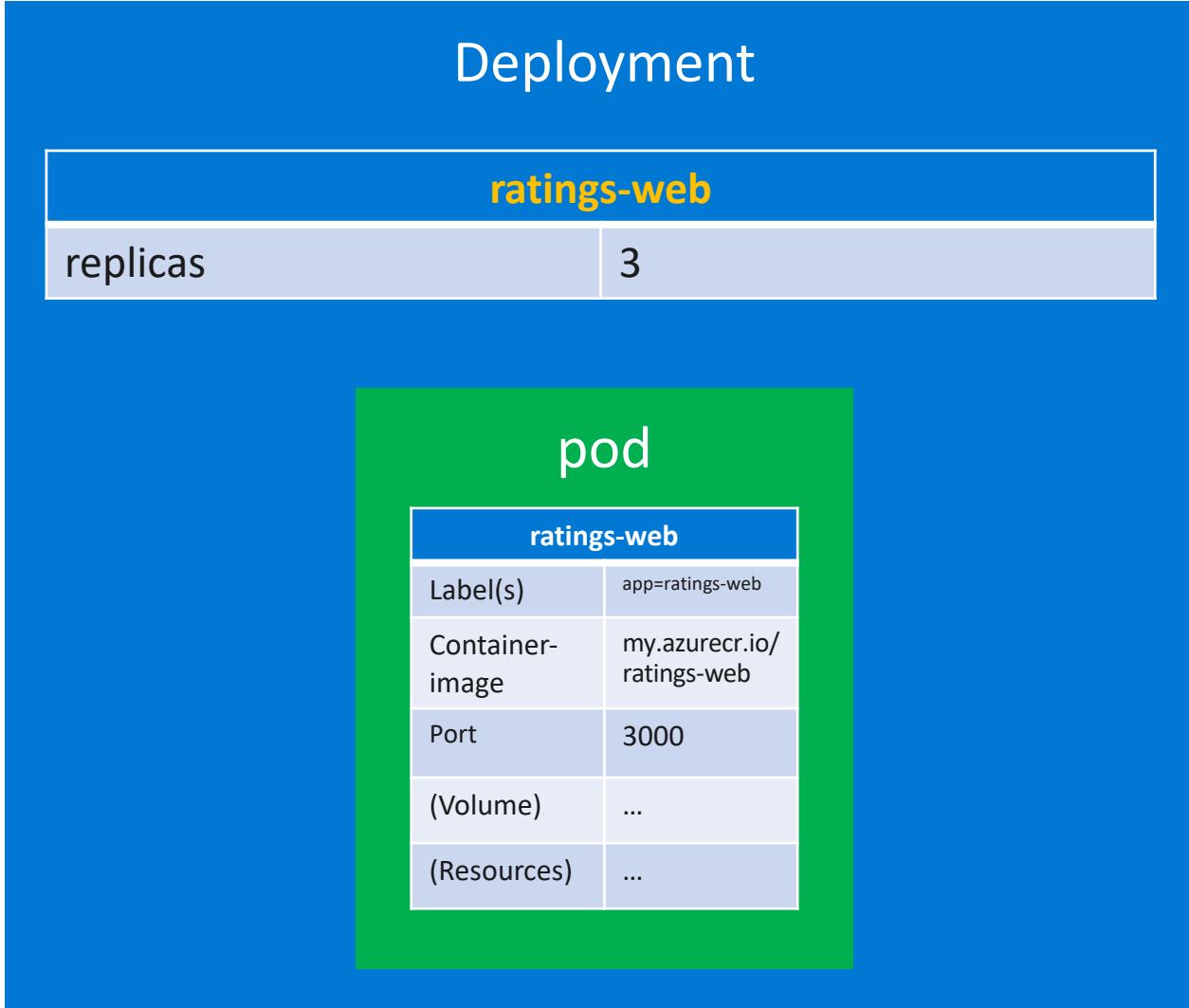
Infra-as-Code a YAML file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: content-web
  labels:
    app: content-web
spec:
  replicas: 1
  selector:
    matchLabels:
      app: content-web
  template:
    metadata:
      labels:
        app: content-web
    spec:
      containers:
        - name: content-web
          image: dta2018hack/content-web
          resources:
            requests:
              cpu: 500m
              memory: 128Mi
          ports:
            - containerPort: 3000
          env:
            - name: CONTENT_API_URL
              value: "http://content-api:3001"
```

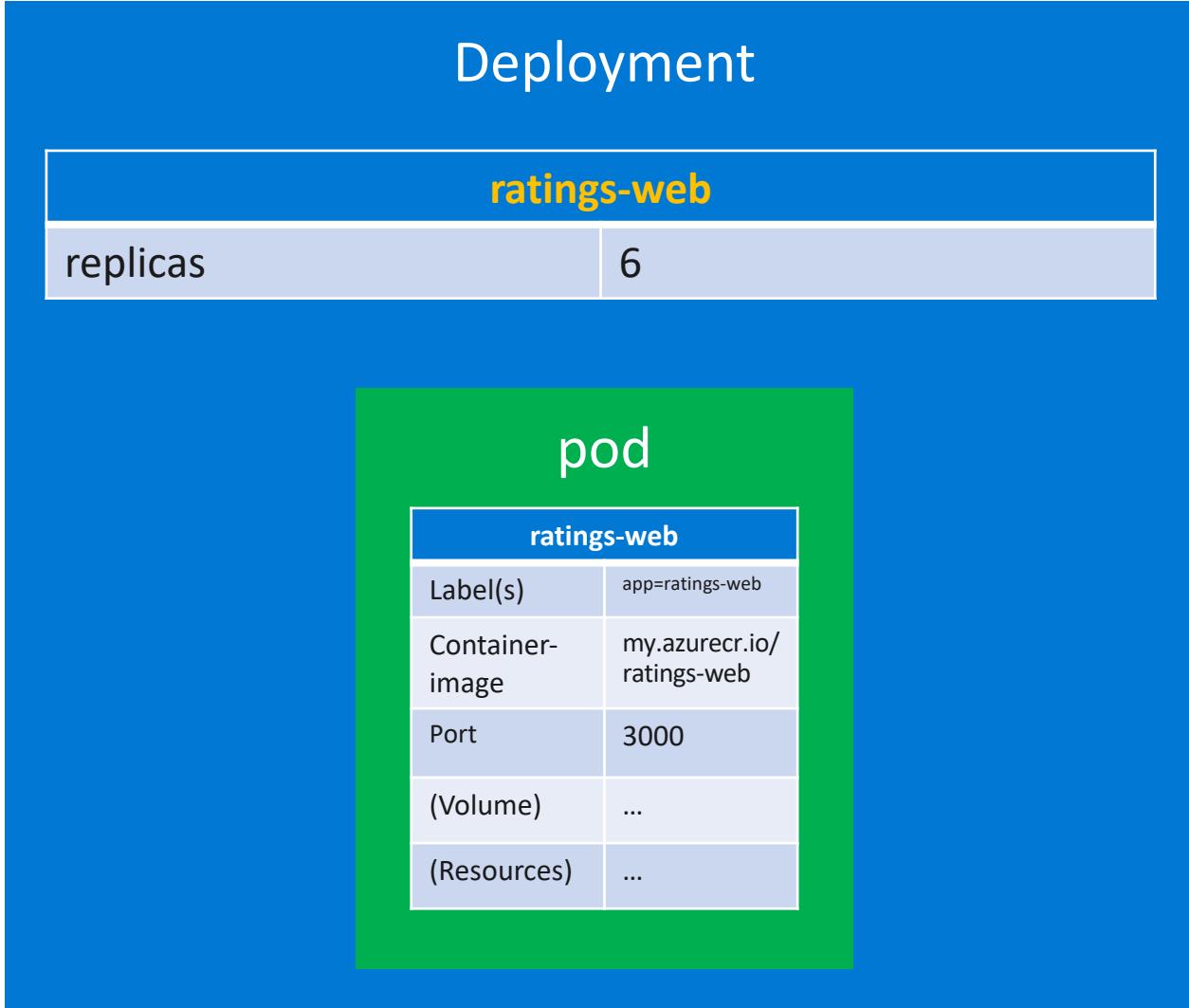
Scaling & High Availability



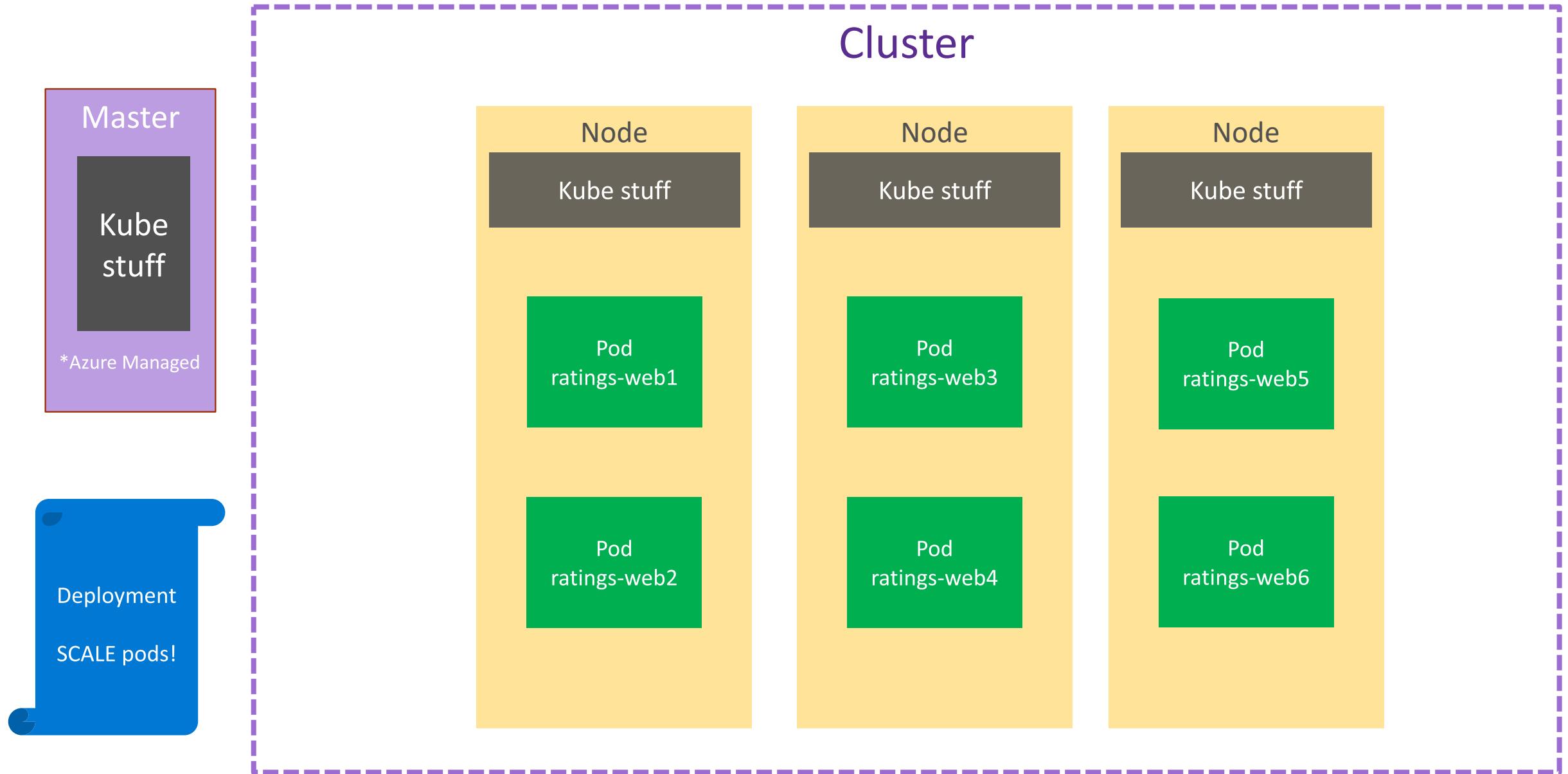
Scaling Pods



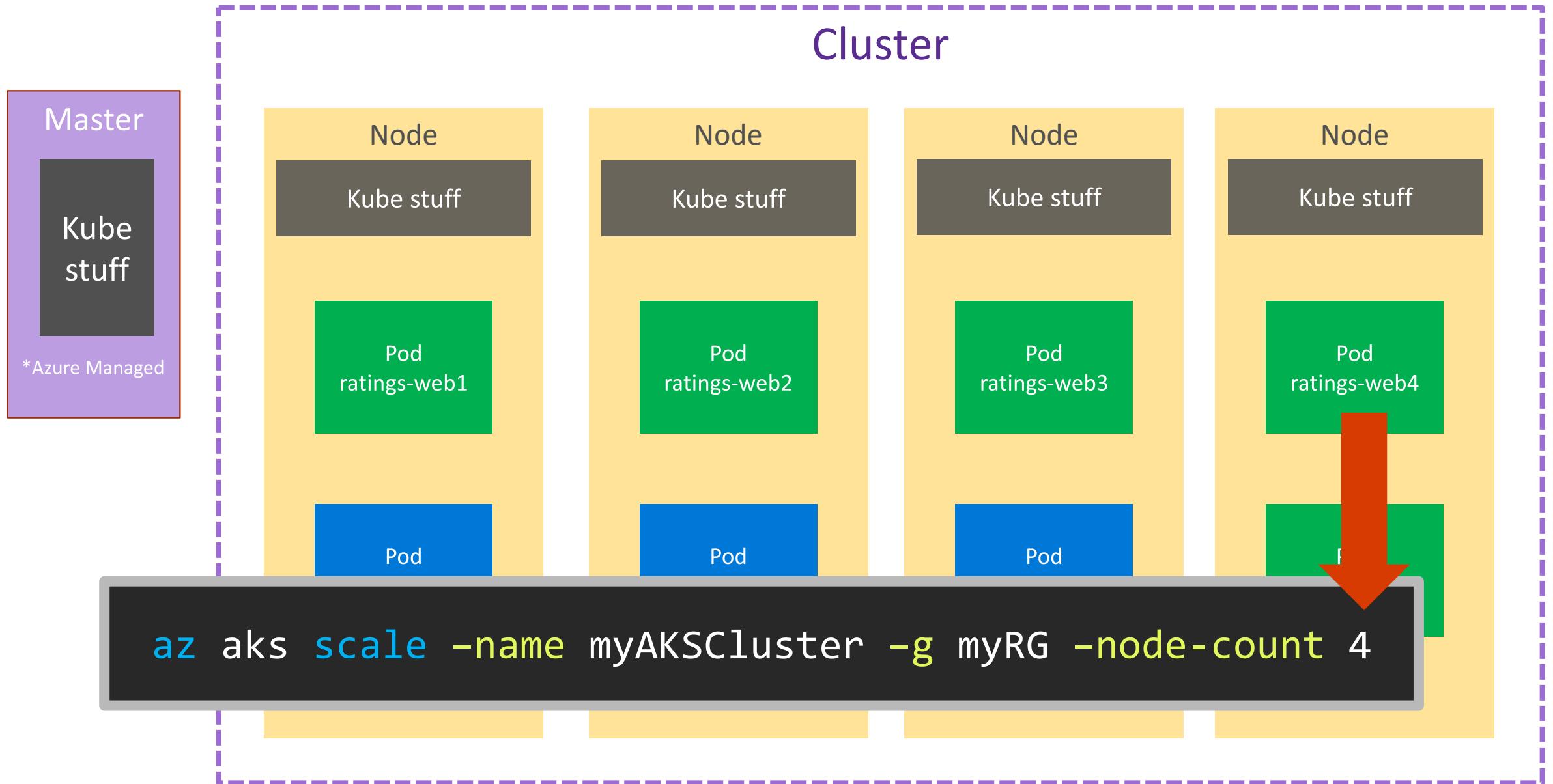
Scaling Pods



Scaling Pods

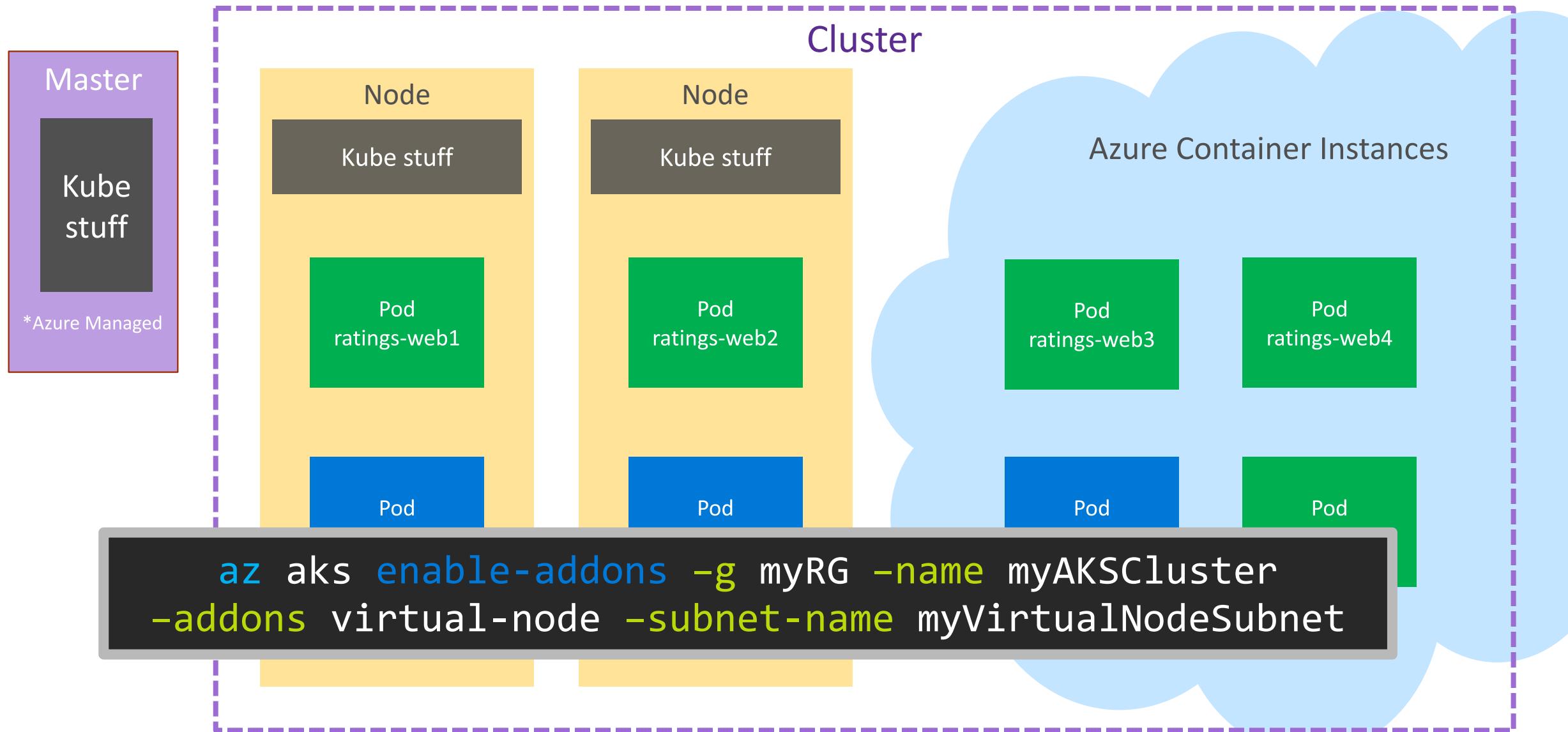


Scaling Nodes



Virtual Nodes

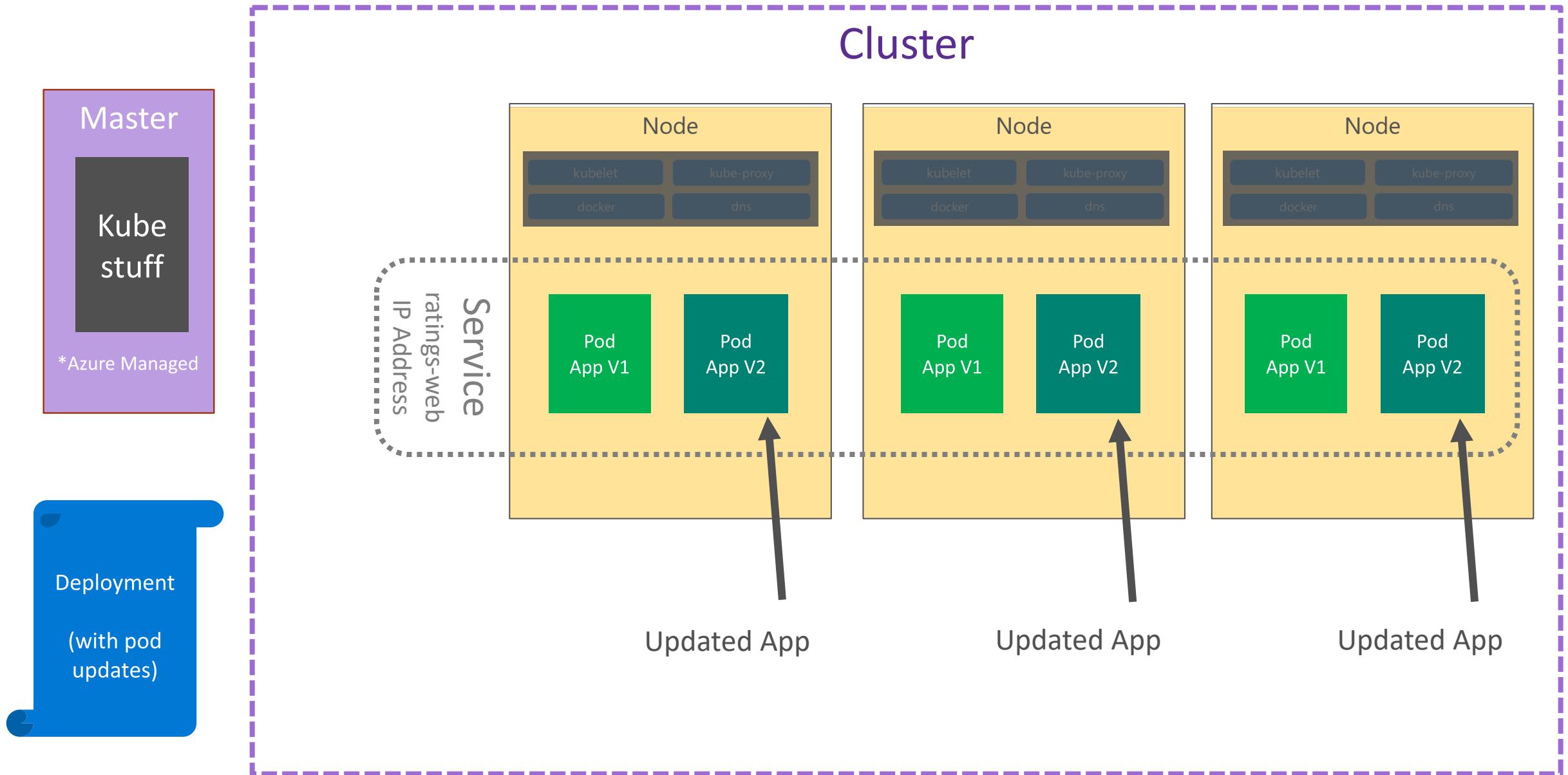
With Azure Container Instances



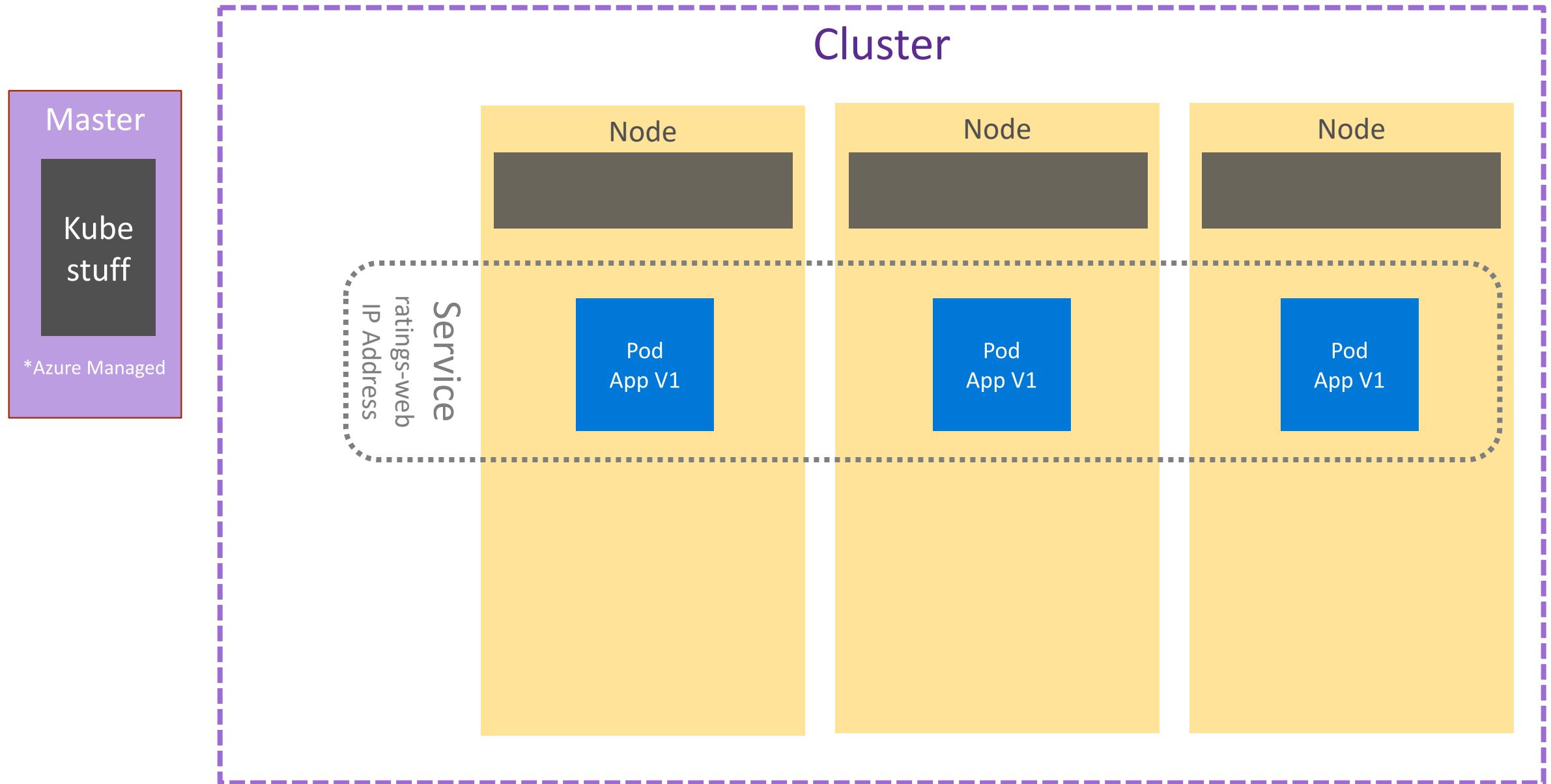
Pod Updates



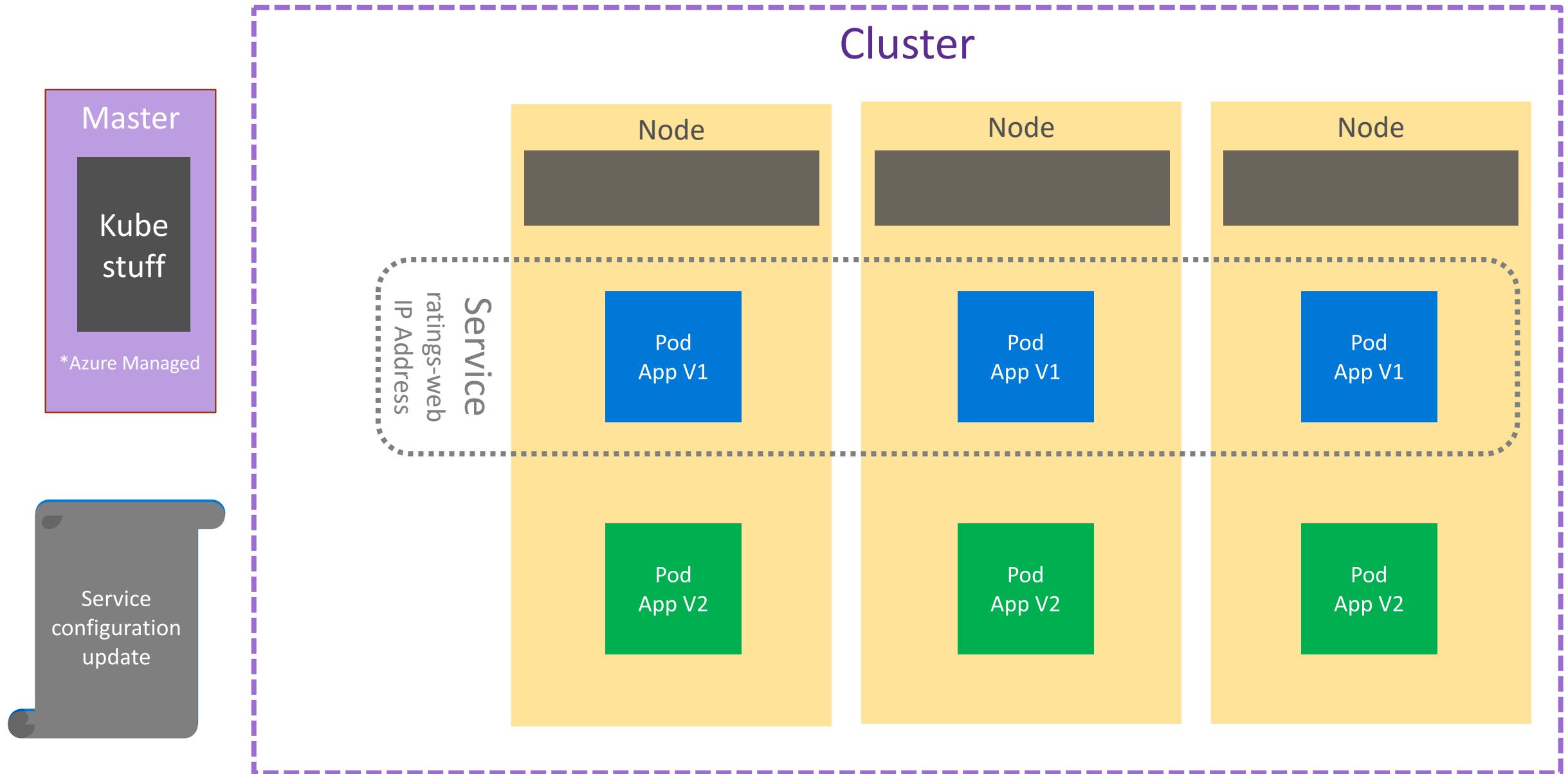
Rolling updates



Blue – the initial deployment



Green – the new deployment



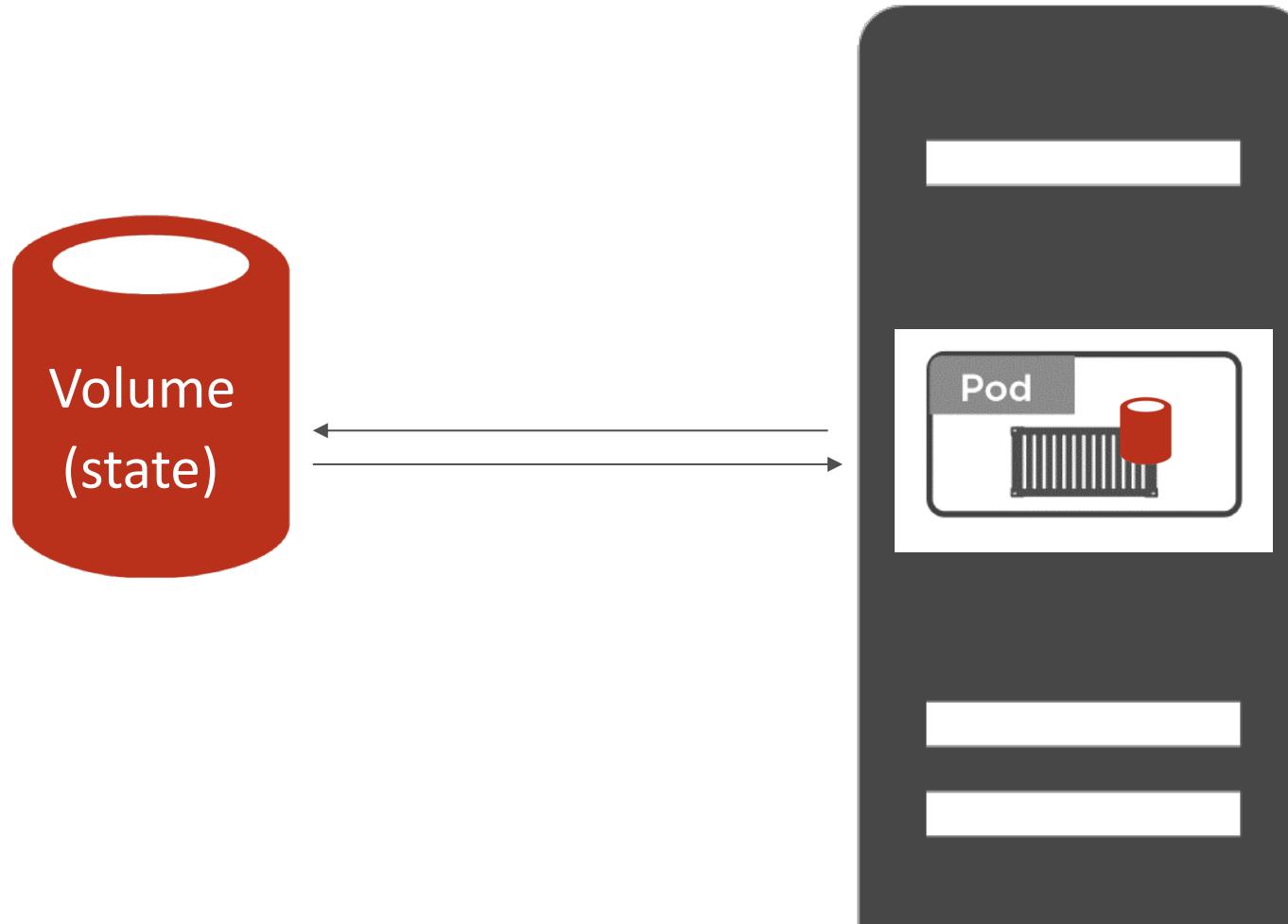
Storage





Volumes in Kubernetes

Why do we need volumes ?



On-disk files in a Container
are
ephemeral

Storage Classes in Kubernetes



Cluster Admin

```
C:\Users\sowmyans>kubectl get sc
NAME          PROVISIONER
default (default)  kubernetes.io/azure-disk
managed-premium   kubernetes.io/azure-disk

```



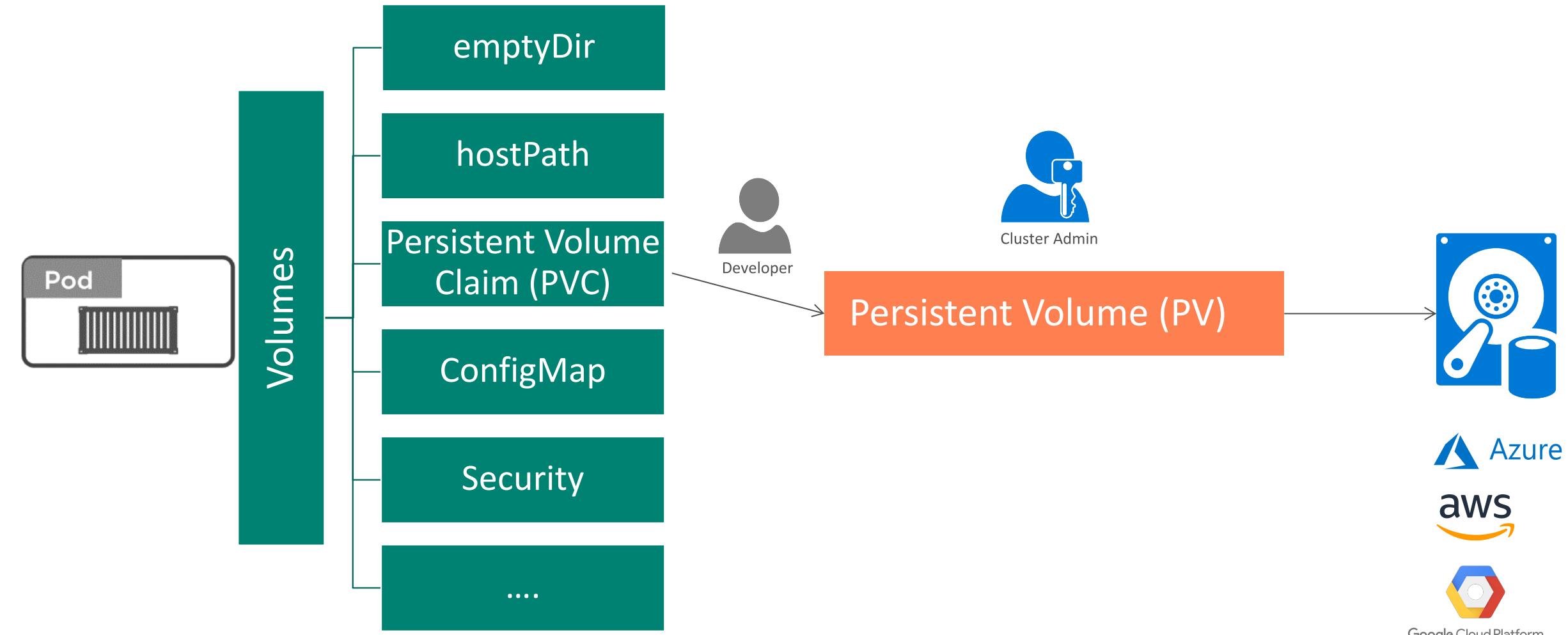
Google Cloud Platform





Volumes in Kubernetes

PV vs PVC





Volumes in Kubernetes

Dynamic Storage in AKS – Azure Disk

```
C:\Users\sowmyans>kubectl get sc
NAME          PROVISIONER          AGE
default (default)  kubernetes.io/azure-disk  9d
managed-premium   kubernetes.io/azure-disk  9d
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: azure-managed-disk
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: managed-premium
  resources:
    requests:
      storage: 5Gi
```

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: myfrontend
      image: nginx
      volumeMounts:
        - mountPath: "/mnt/azure"
          name: volume
  volumes:
    - name: volume
      persistentVolumeClaim:
        claimName: azure-managed-disk
```

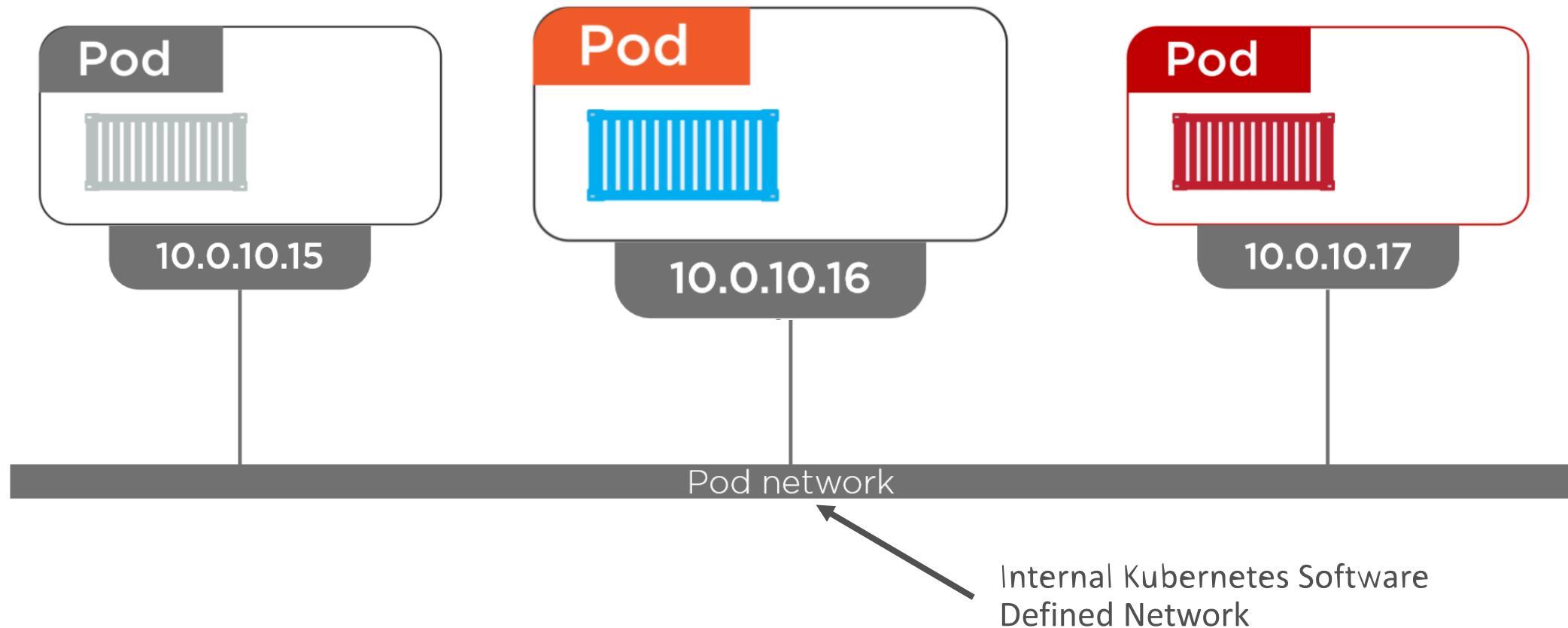
Networking





Networking – Pod

Inter-Pod Communication



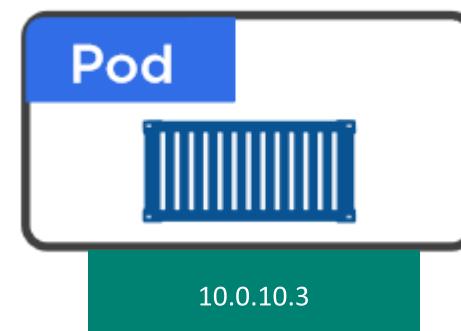
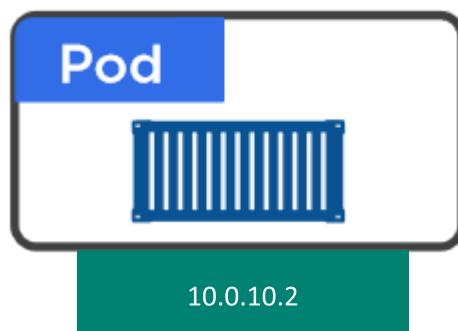
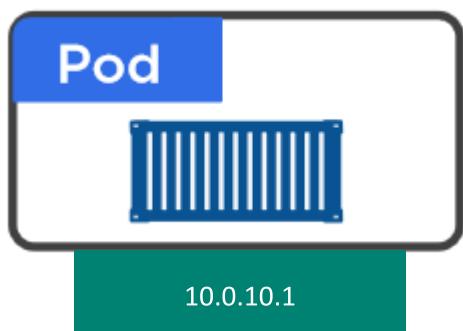
Networking – Service



Why Services?



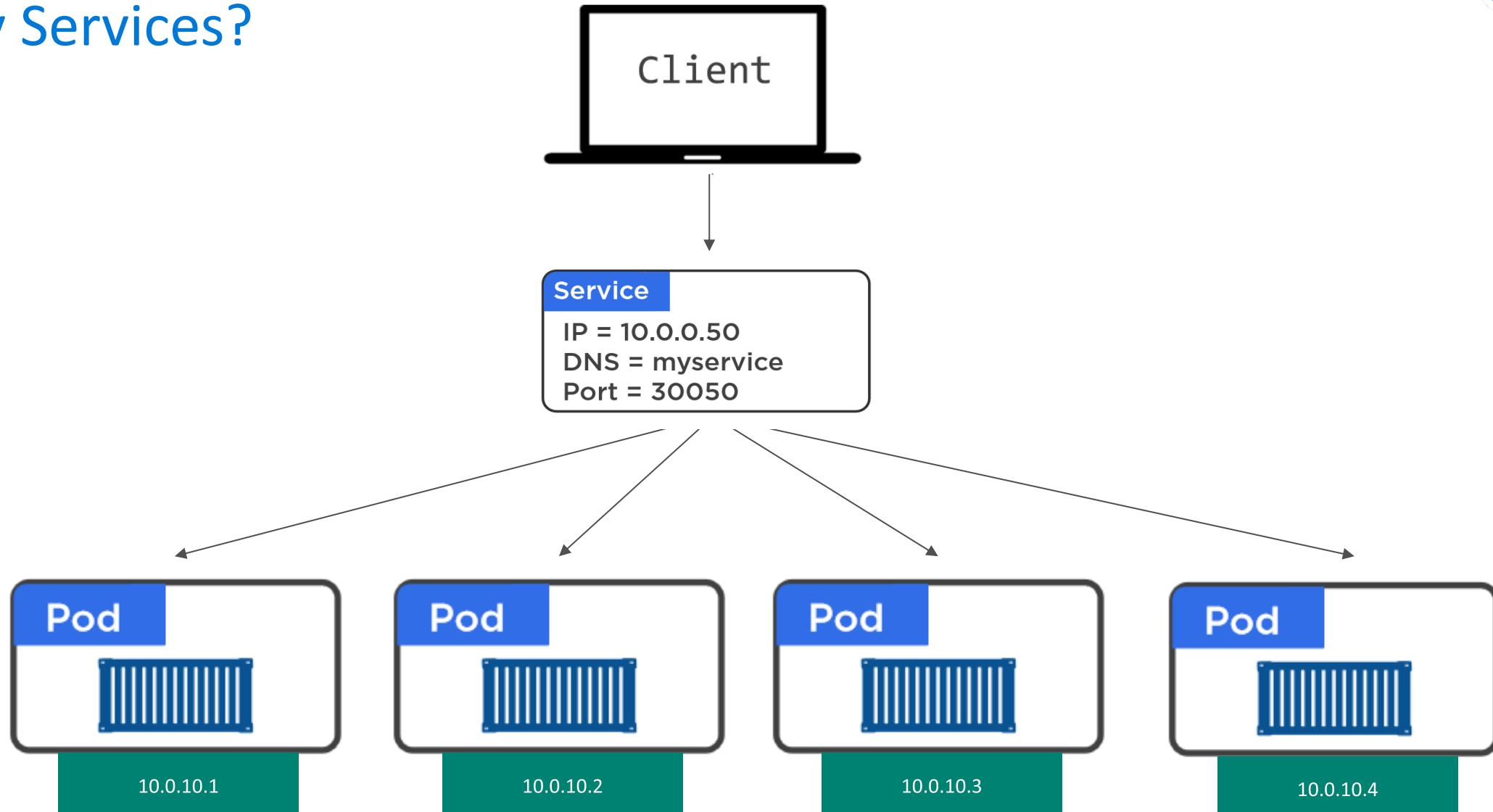
Where's
10.0.10.4?!





Networking – Service

Why Services?





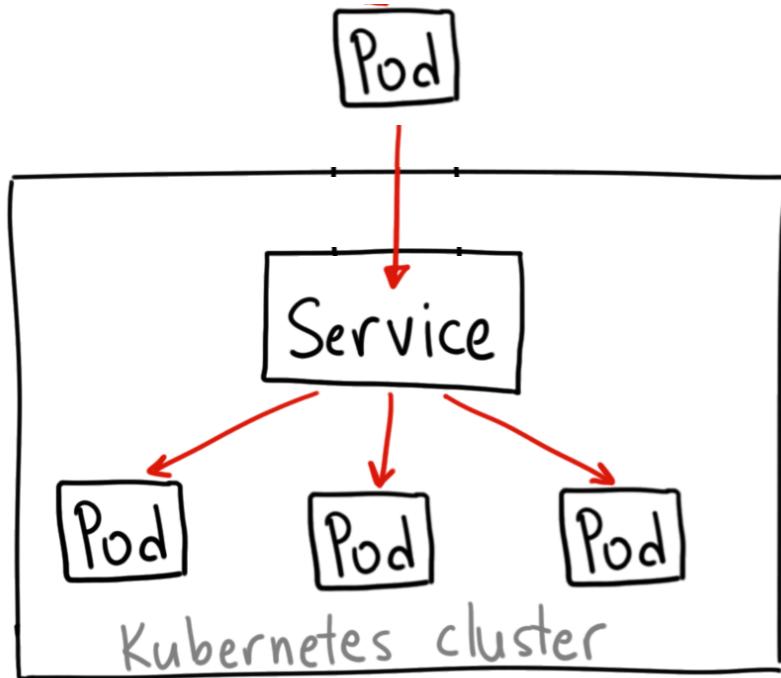
Different service types

- ClusterIP (internal)
- LoadBalancer (external)
- NodePort (external)



Networking – Service : ClusterIP

How does ClusterIP Work

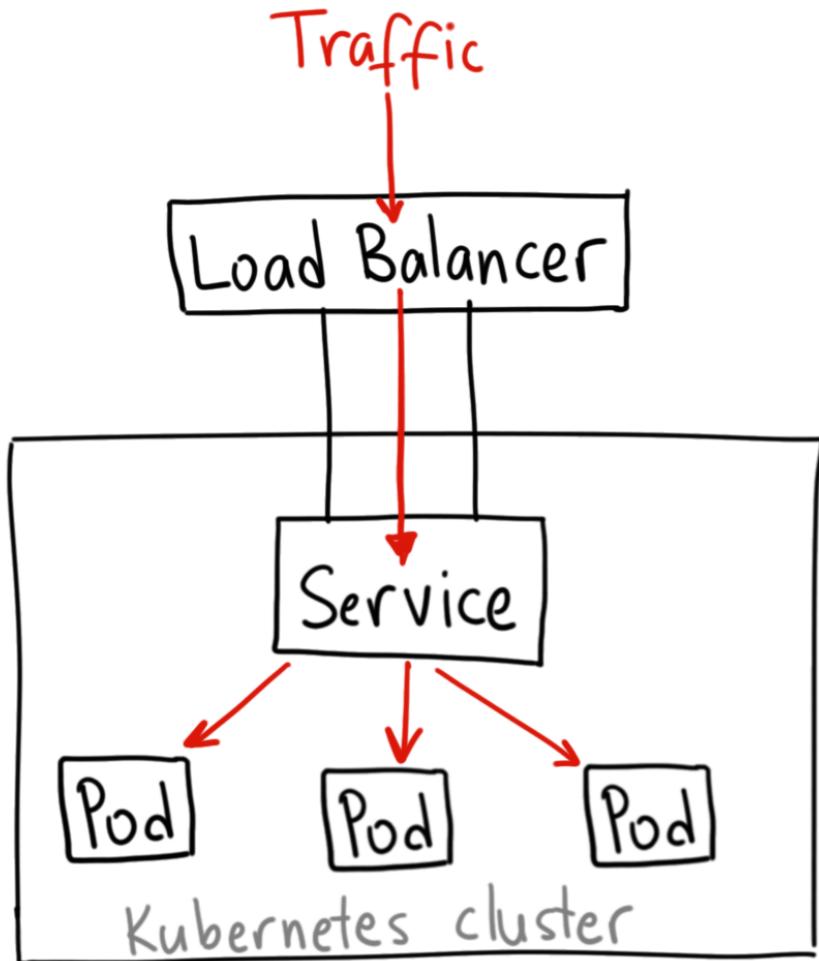


```
1  apiVersion: v1
2  kind: Service
3  metadata:
4  |   name: my-internal-service
5  spec:
6  |   selector:
7  |   |   app: my-app
8  |   type: ClusterIP
9  ports:
10 |   - name: http
11 |   |   port: 80
12 |   |   targetPort: 80
13 |   |   protocol: TCP
```



Networking – Service : LoadBalancer

How does LoadBalancer Work

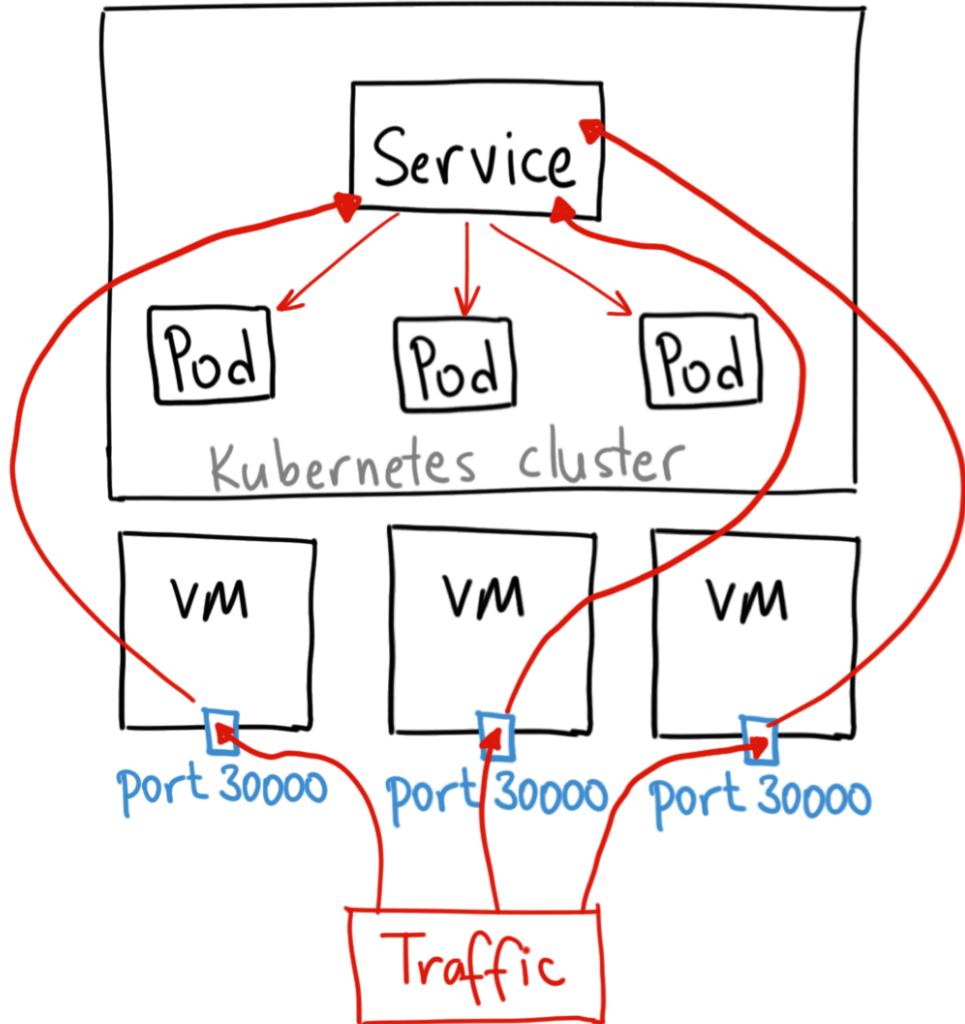


```
1  apiVersion: v1
2  kind: Service
3  metadata:
4  |  labels:
5  |    app: web
6  |    name: web
7  spec:
8  |  ports:
9  |    - name: web-traffic
10 |      port: 80
11 |      protocol: TCP
12 |      targetPort: 3000
13 |  selector:
14 |    app: web
15 |  sessionAffinity: None
16 |  type: LoadBalancer
```



Networking – Service : NodePort

How does NodePort Work

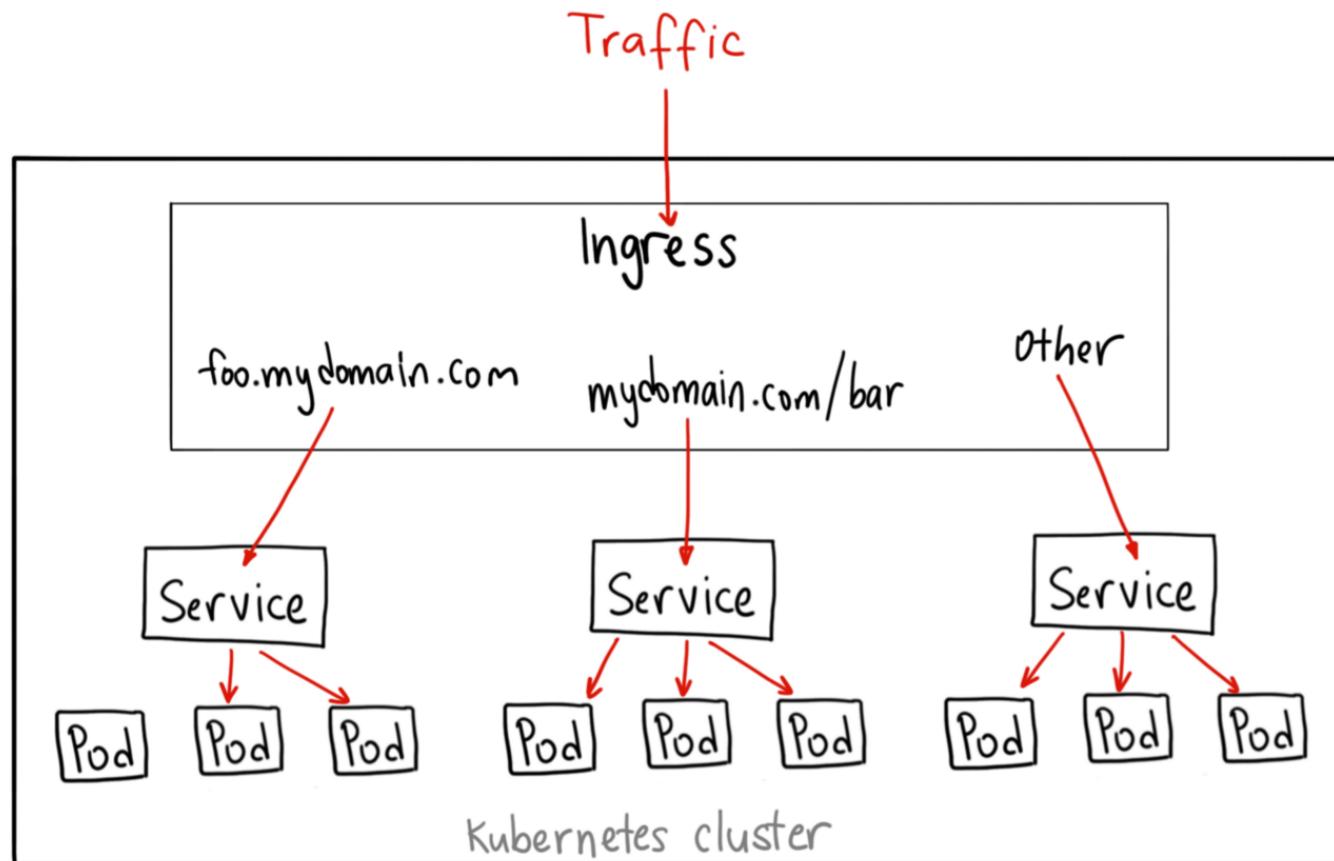


```
1  apiVersion: v1
2  kind: Service
3  metadata:
4  |   name: my-nodeport-service
5  spec:
6  |   selector:
7  |   |   app: my-app
8  |   type: NodePort
9  ports:
10 |   - name: http
11 |   |   port: 80
12 |   |   targetPort: 80
13 |   |   nodePort: 30036
14 |   |   protocol: TCP
```



Networking – Ingress

How does Ingress Controller Work



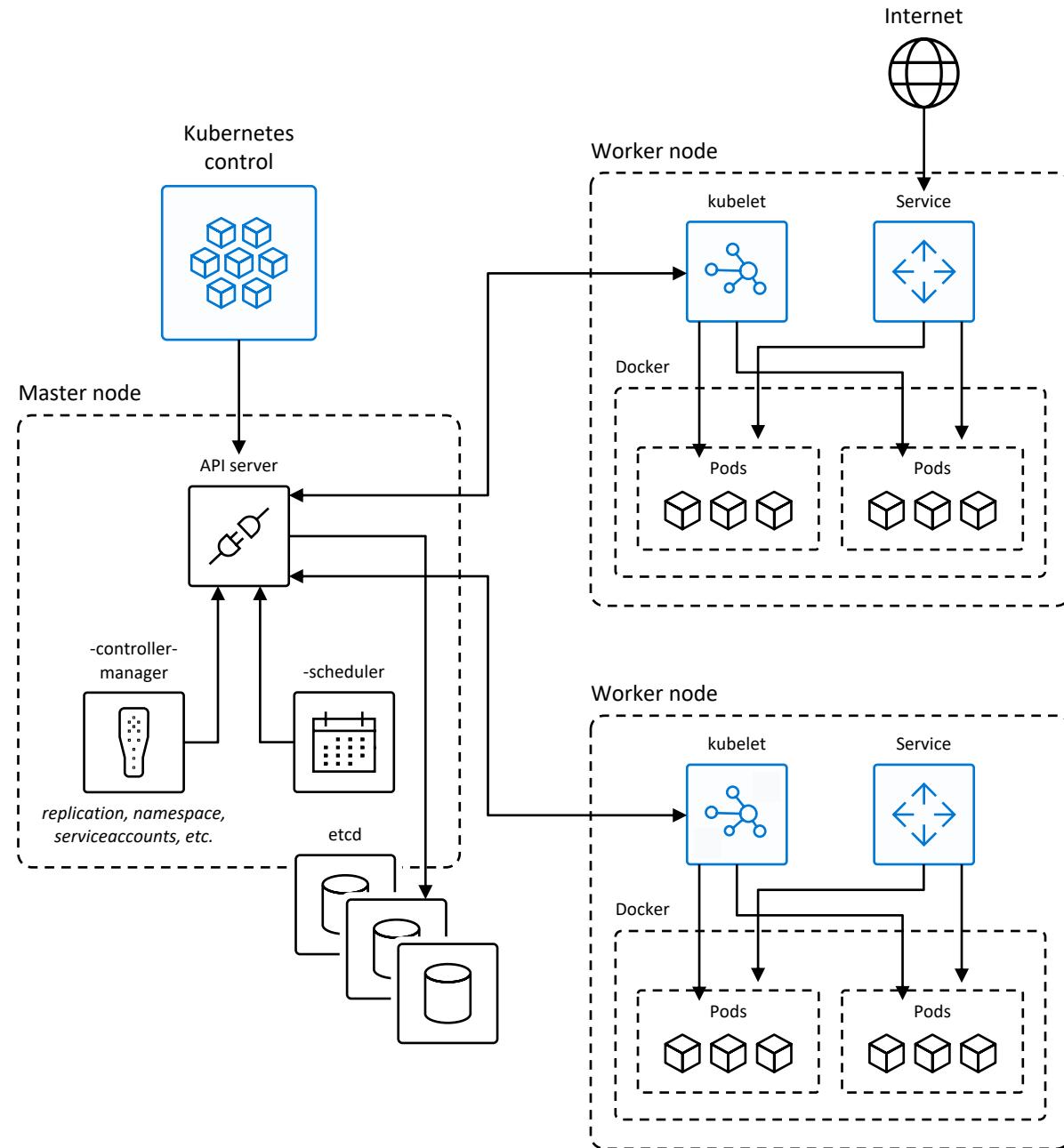
```
1  apiVersion: extensions/v1beta1
2  kind: Ingress
3  metadata:
4    name: my-ingress
5  spec:
6    backend:
7      serviceName: other
8      servicePort: 8080
9    rules:
10   - host: foo.mydomain.com
11     http:
12       paths:
13         - backend:
14           serviceName: foo
15           servicePort: 8080
16   - host: mydomain.com
17     http:
18       paths:
19         - path: /bar/*
20           backend:
21             serviceName: bar
22             servicePort: 8080
```

Let's review...



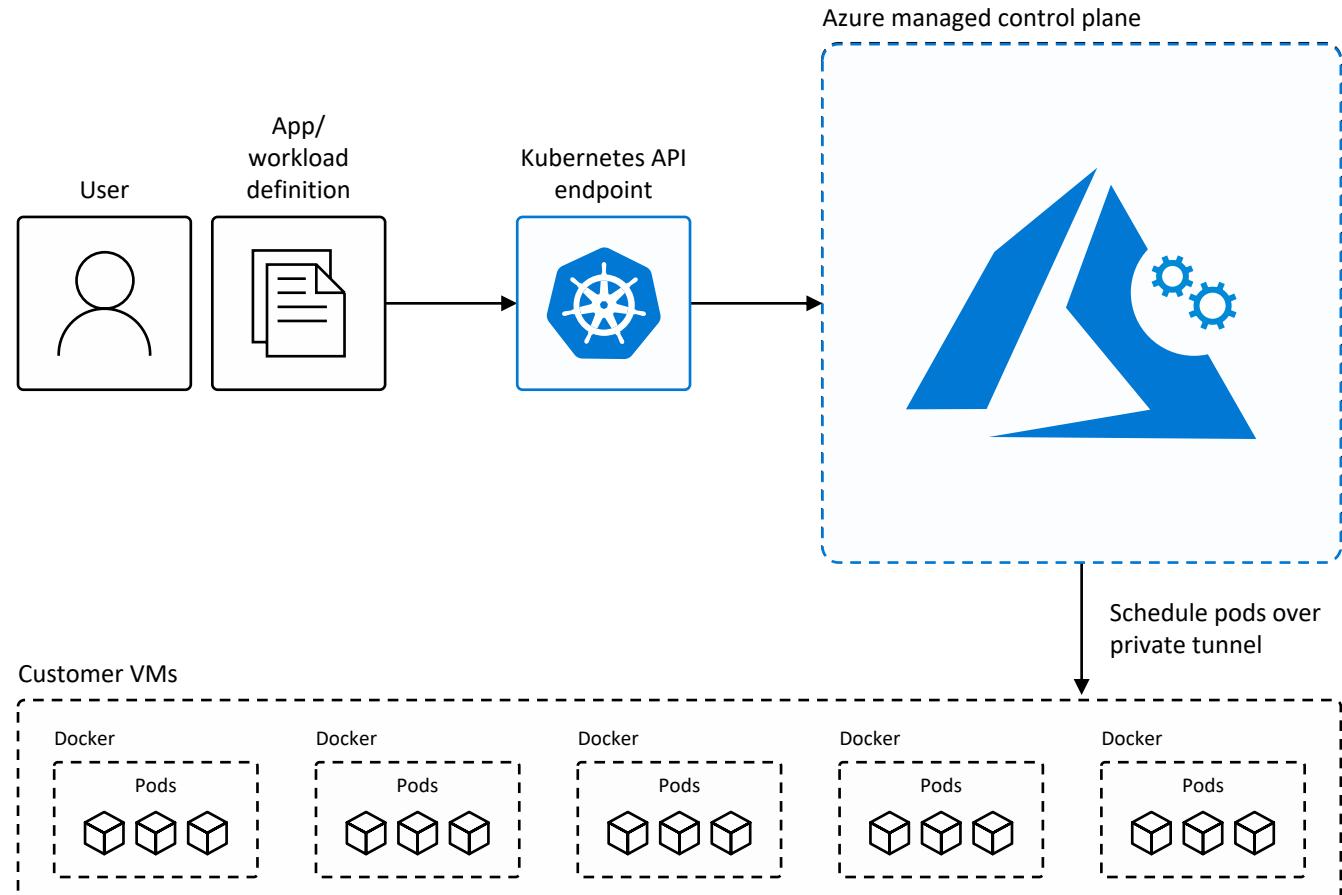
Kubernetes 101

1. Kubernetes users communicate with API server and apply desired state
2. Master nodes actively enforce desired state on worker nodes
3. Worker nodes support communication between containers
4. Worker nodes support communication from the Internet



How managed Kubernetes on Azure works

- Automated upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge



What if we don't use a managed service?



Task	The old way	With Azure
Create a cluster	<p>Provision network and VMs</p> <p>Install dozens of system components including etcd</p> <p>Create and install certificates</p> <p>Register agent nodes with control plane</p>	<code>az aks create</code>
Upgrade a cluster	<p>Upgrade your master nodes</p> <p>Cordon/drain and upgrade worker nodes individually</p>	<code>az aks upgrade</code>
Scale a cluster	<p>Provision new VMs</p> <p>Install system components</p> <p>Register nodes with API server</p>	<code>az aks scale</code>

And, we're done...



Just kidding!
You're NEVER done with Kubernetes!



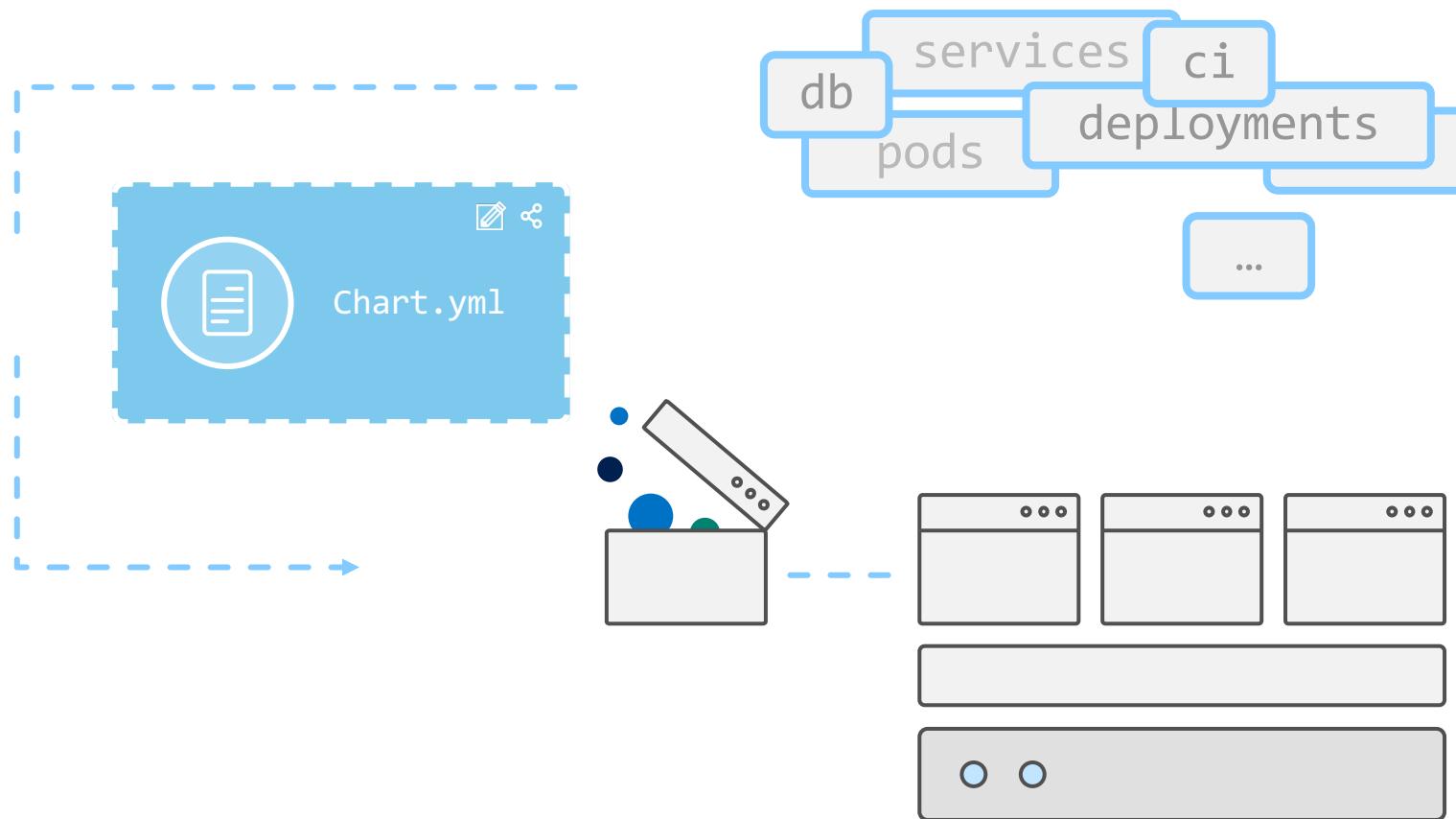
What's Next?

Operationalizing Kubernetes



Helm

Helm Charts helps you define, install, and upgrade even
the most complex Kubernetes application



Helm

The best way to find, share, and use software built
for Kubernetes



Manage complexity

Charts can describe complex apps; provide repeatable app installs, and serve as a single point of authority



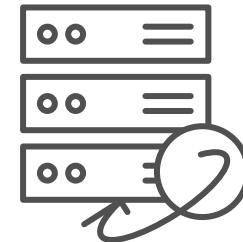
Easy updates

Take the pain out of updates with in-place upgrades and custom hooks



Simple sharing

Charts are easy to version, share, and host on public or private servers



Rollbacks

Use `helm rollout` to roll back to an older version of a release with ease



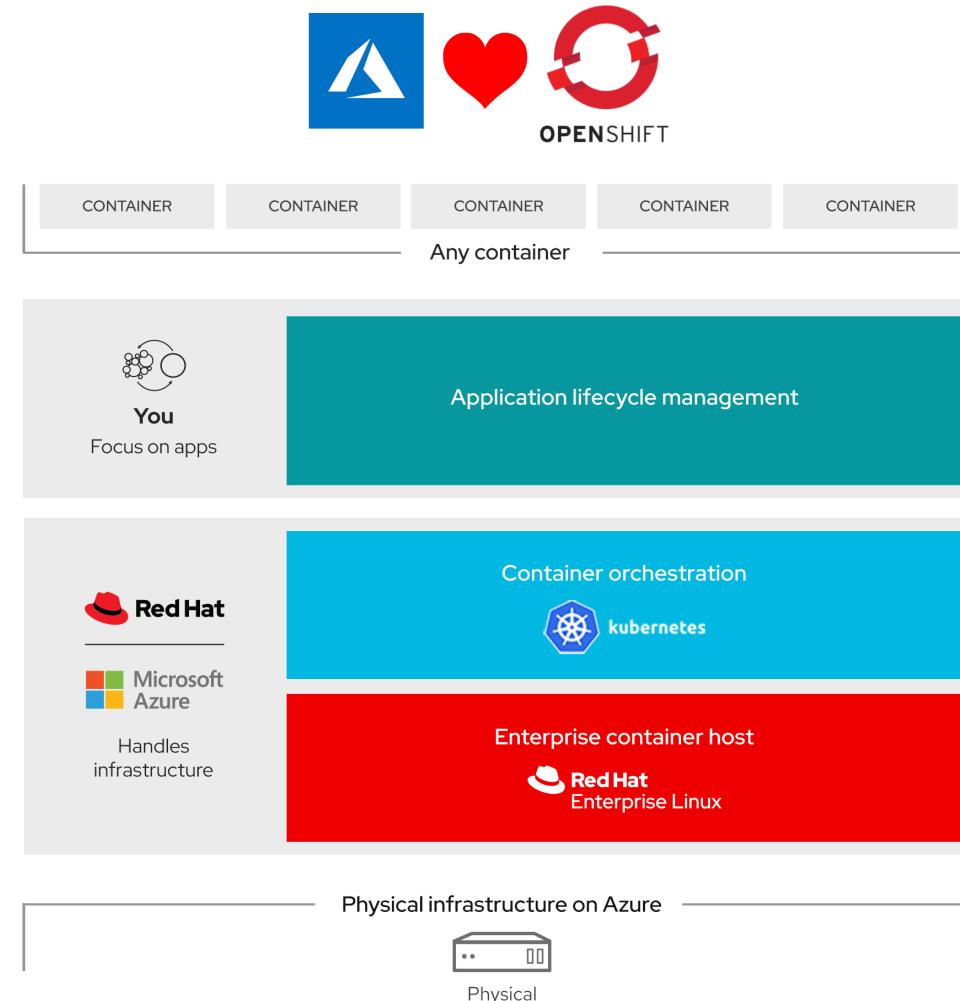
Monitoring

Many Choices

- Kubernetes Dashboard
- Azure Monitor for AKS
- 3rd Party Open Source Tools
 - Prometheus/Grafana
 - Heapster/InfluxDB/Grafana
 - Heapster/ELK Stack
 - Datadog/Dynatrace
 - New Relic

Exciting update: Azure Red Hat OpenShift

A fully managed service of Red Hat OpenShift on Azure, jointly, engineered, operated and supported by Microsoft and Red Hat



Session resources

- Official Kubernetes docs: <https://kubernetes.io/docs/home/>
- Learning path for k8s: <https://aka.ms/learnkubernetes>
- AKS hands on: <https://aksworkshop.io/>
- Helm 3.0: <https://helm.sh/blog/helm-3-preview-pt7/>
- AKS best practices: <https://docs.microsoft.com/en-us/azure/aks/best-practices>

Q&A



Thank you!

