

Cloud native ~~technology~~ journey

Jan Egil Ring, Crayon

Agenda

- 09.05 – 09.30 **Introduction to Cloud native**
- 09.30 – 10.30 **Containers**
- 10.30 – 10.45 break
- 10.45 – 11.45 **Kubernetes**
- 11.45 – 12.00 **DevOps pipelines**
- 12.00 – 13.00 lunch
- 13.00 – 16.00 **Lab (optional)**



Cloud native
is the new
paradigm
of software
development

>35%

of all production
apps will be cloud-
native by 2022¹

What is cloud native?

Package application code & dependencies in containers, deploy as microservices and manage them using DevOps processes & tools

APIs

Expose services as light-weight APIs for easier integration



Containers

Standard deployment format to abstract code from underlying infrastructure differences



Microservices

Architectural approach to developing an application as a collection of modular, loosely coupled services

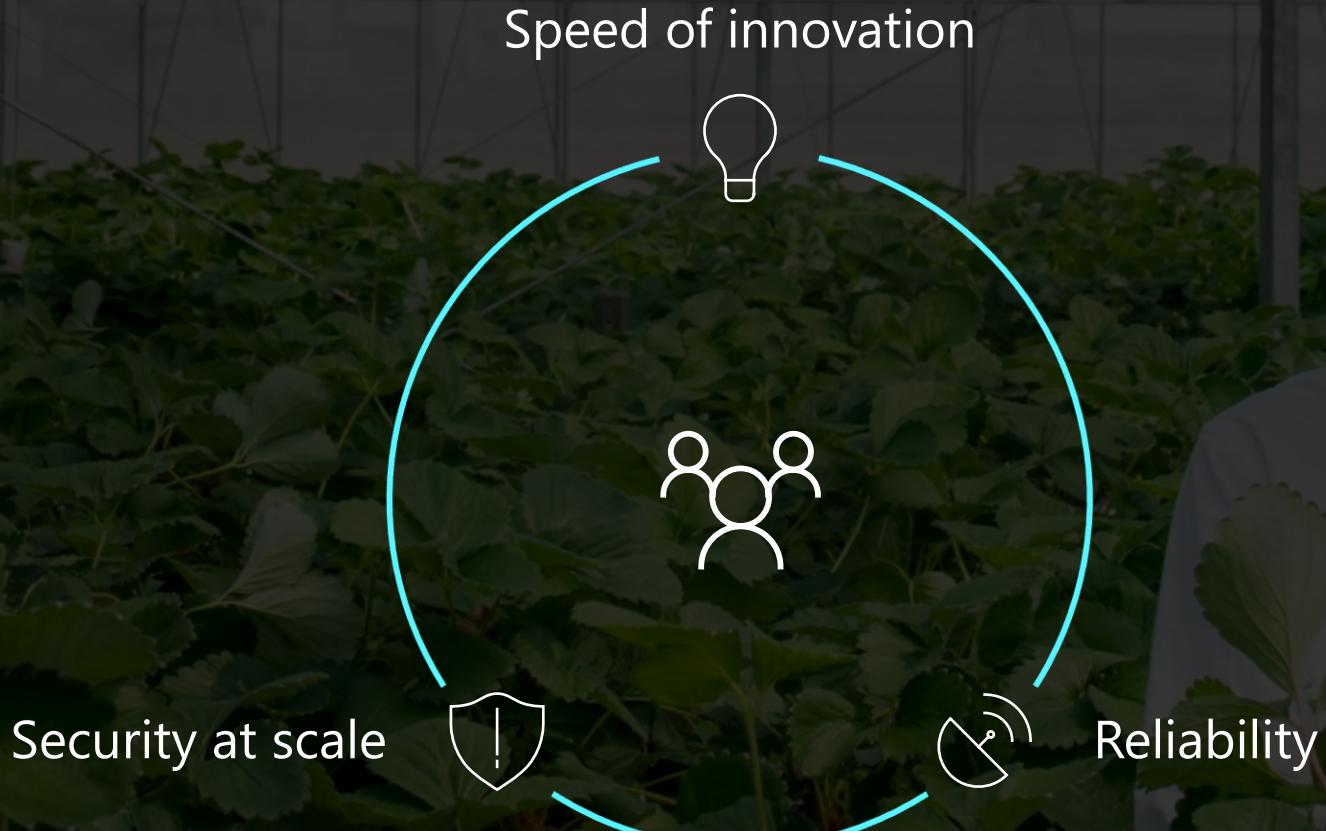


DevOps

People, processes and technology that promote collaborative building and delivery



Why cloud native?





Speed of innovation

- Strongest developer experience¹
- Most complete tool chain from Git to production
- Serverless scaling with no infrastructure to manage



Reliability

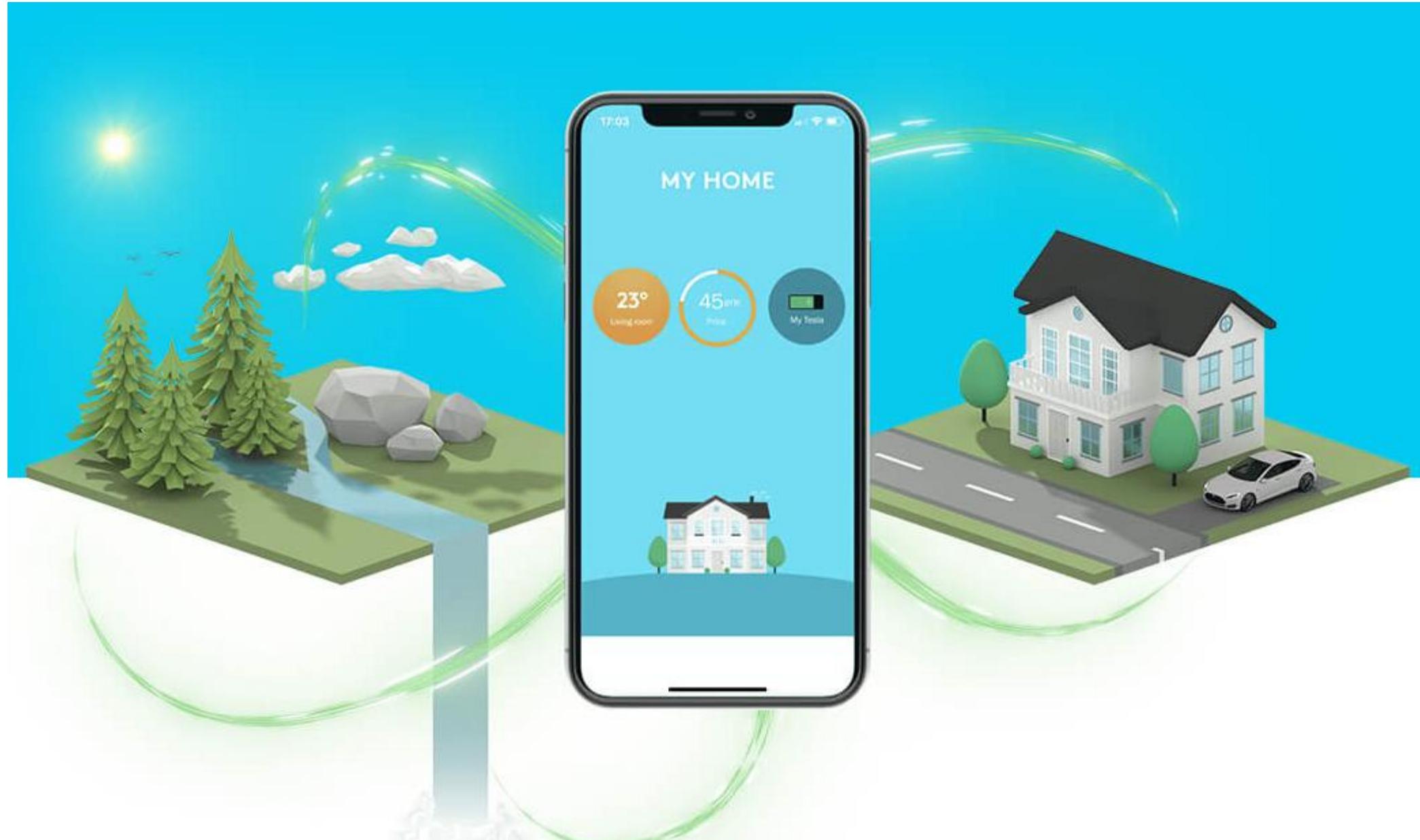
- Fully managed database services with >99.999-percent high availability
- Single-digit millisecond latencies on reads and writes
- Available in more regions than any other cloud providers



Security at scale

- \$1BN investment every year in security
- >90 compliance certifications
- Out of the box integration with Azure Policy, Active Directory and Security Center

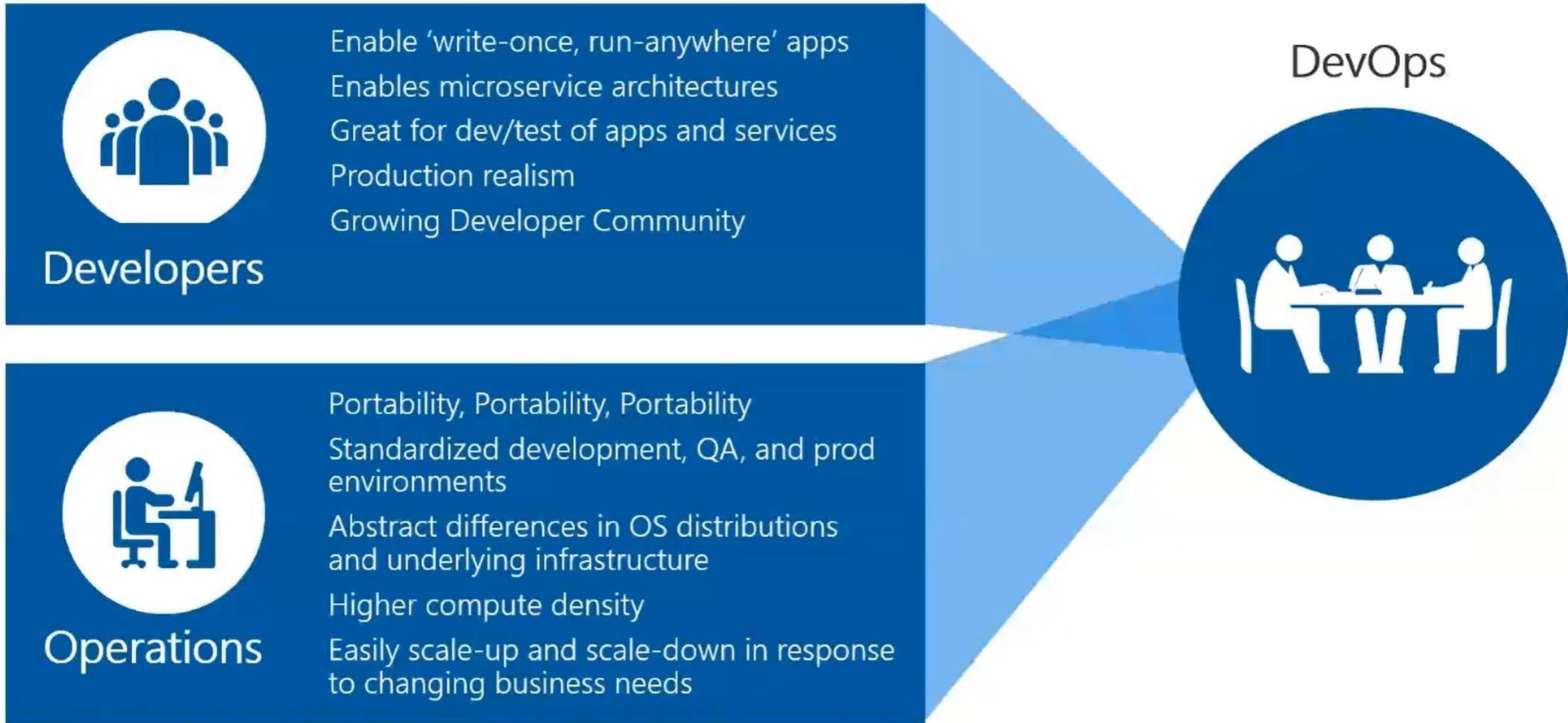
A practical example



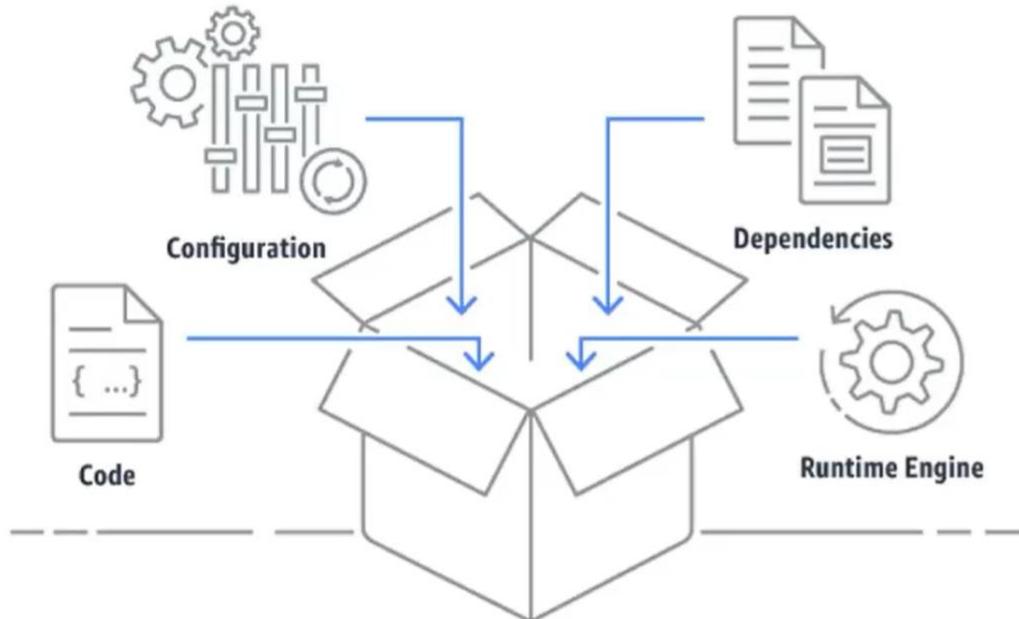
Introducing Containers & Kubernetes

What's all the hype?!

Why Containers?



What is a container?



Container solutions provide a standard way to package your application's code, configurations, and dependencies into a single object.

Images

Is what you package your application and its environment into

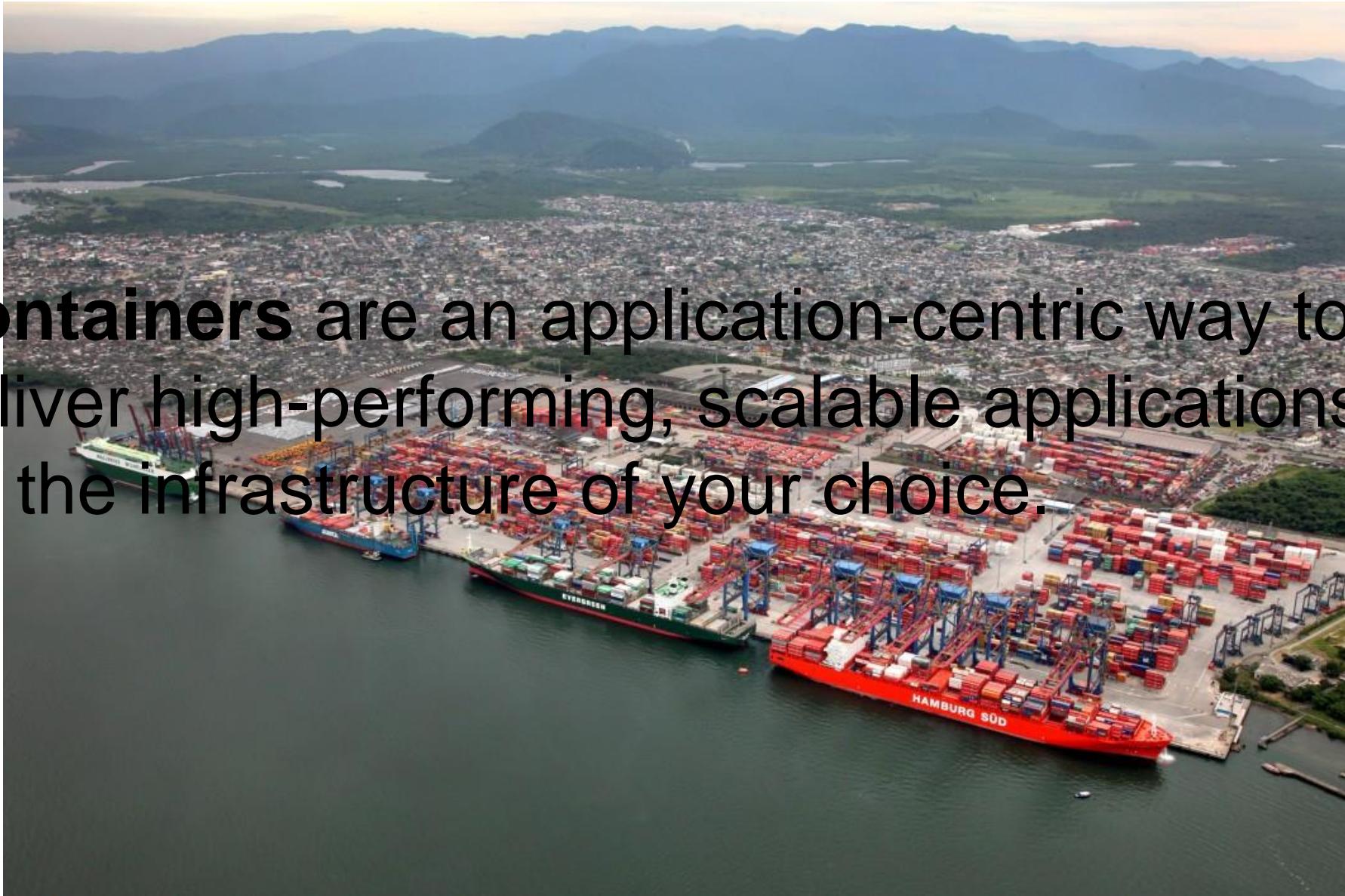
Registries

Is a repository that stores your images and allows sharing images between different people and machines

Container

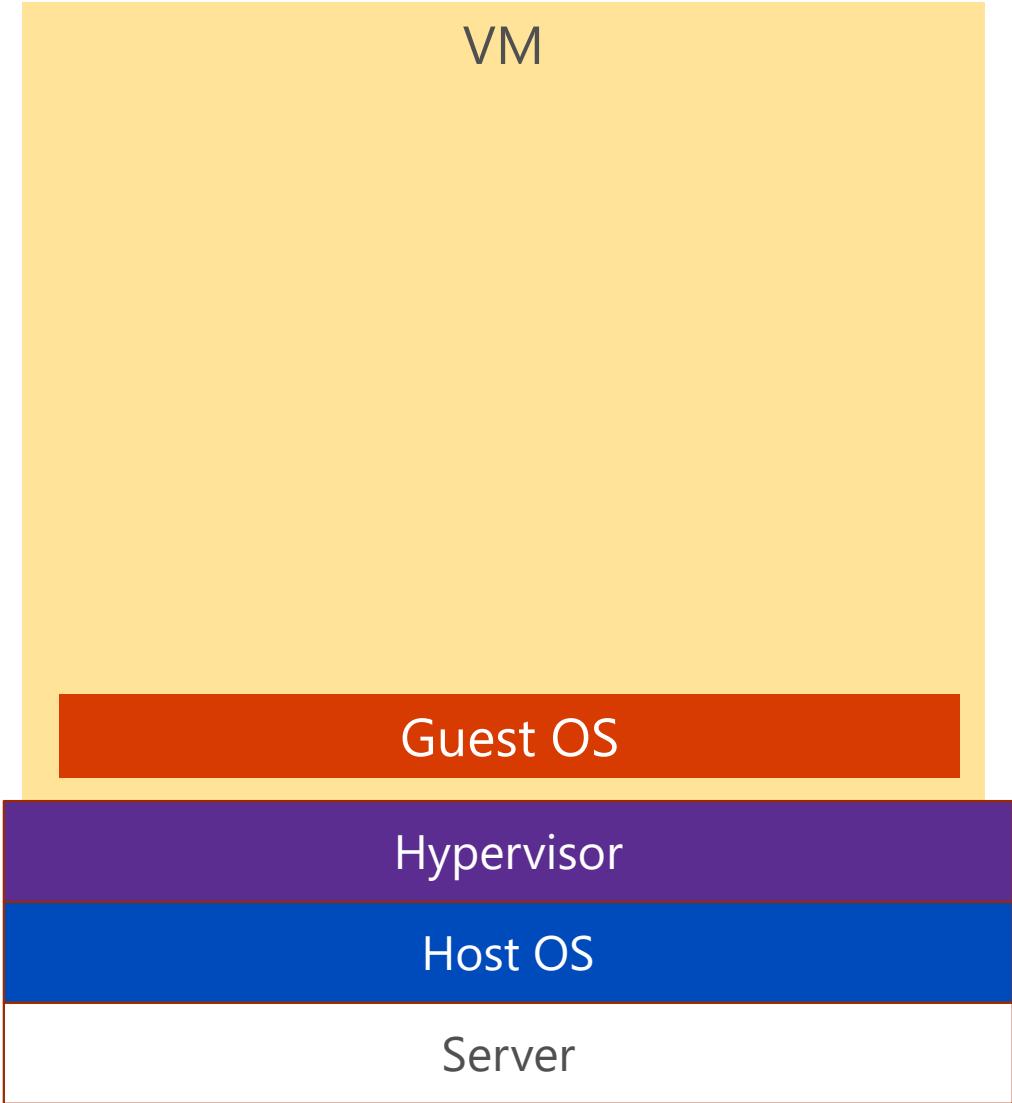
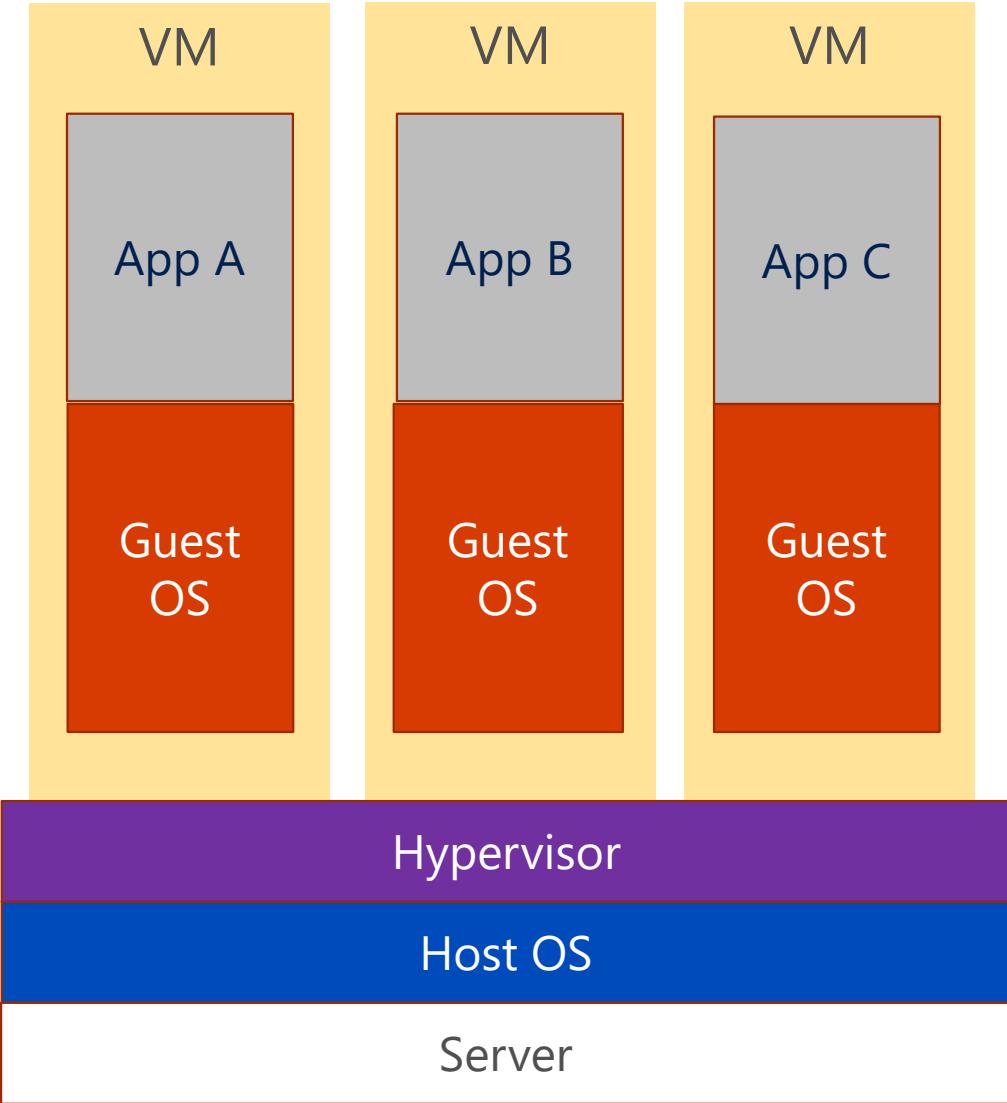
Is a process running on the host running docker, but it is “completely” isolated from the host and other processes.

Review: What is a Container?

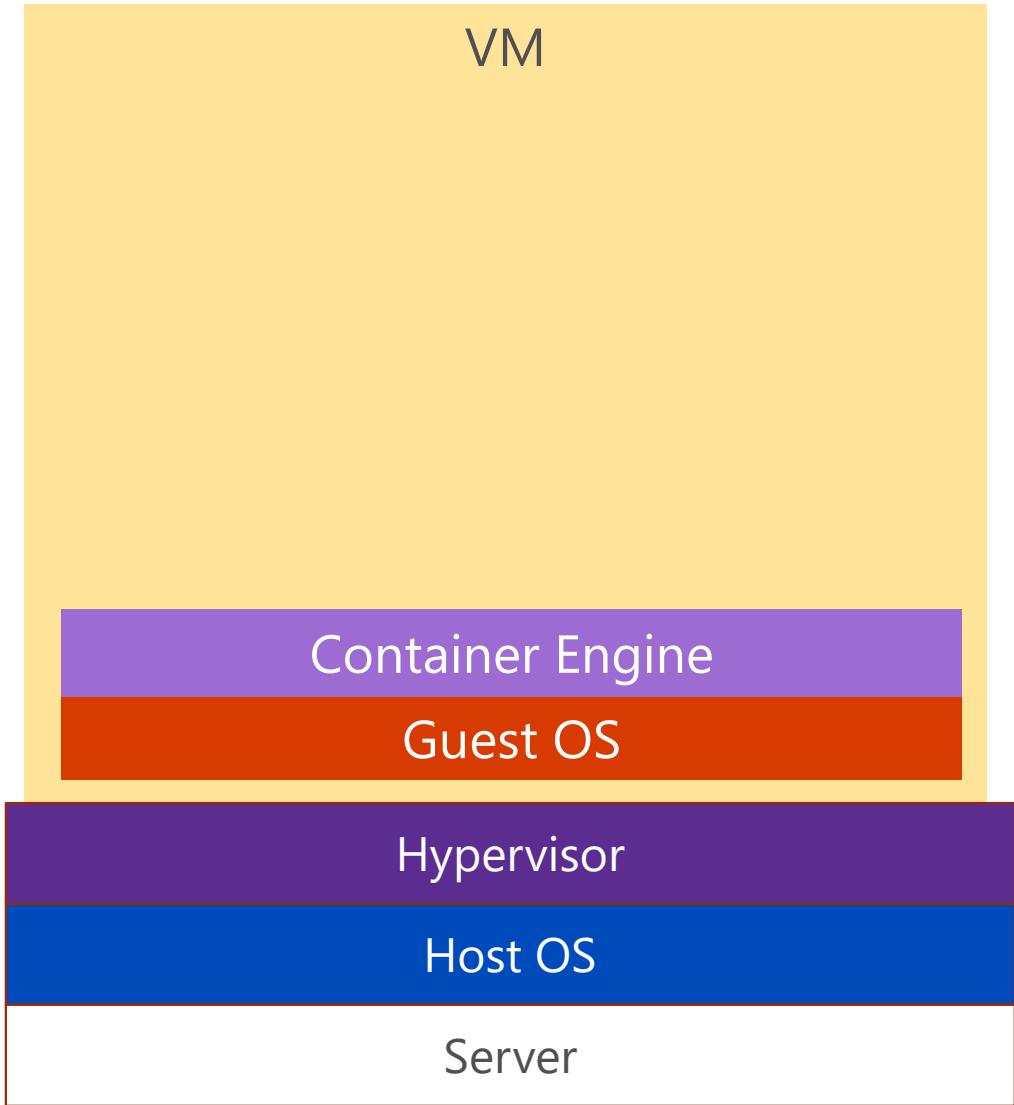
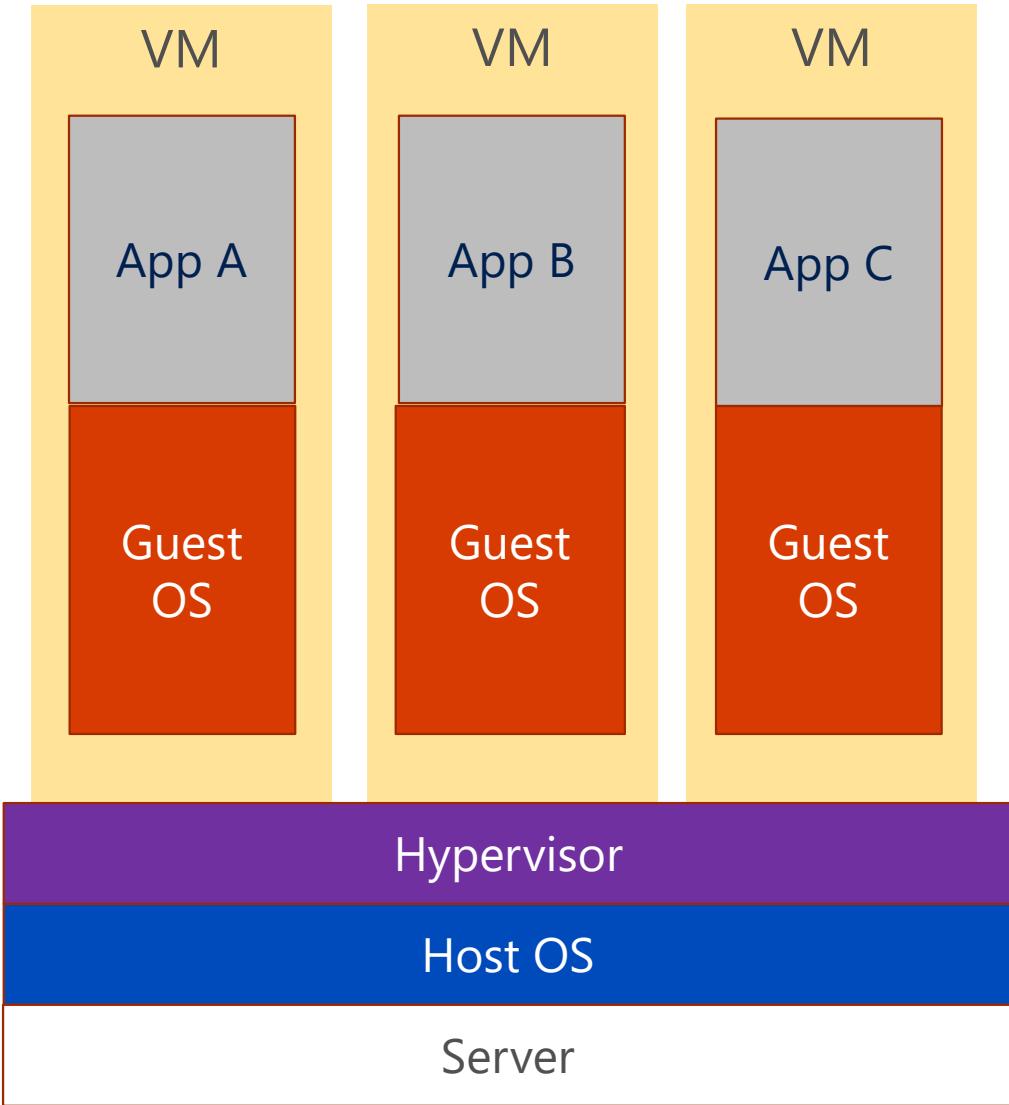


Containers are an application-centric way to deliver high-performing, scalable applications on the infrastructure of your choice.

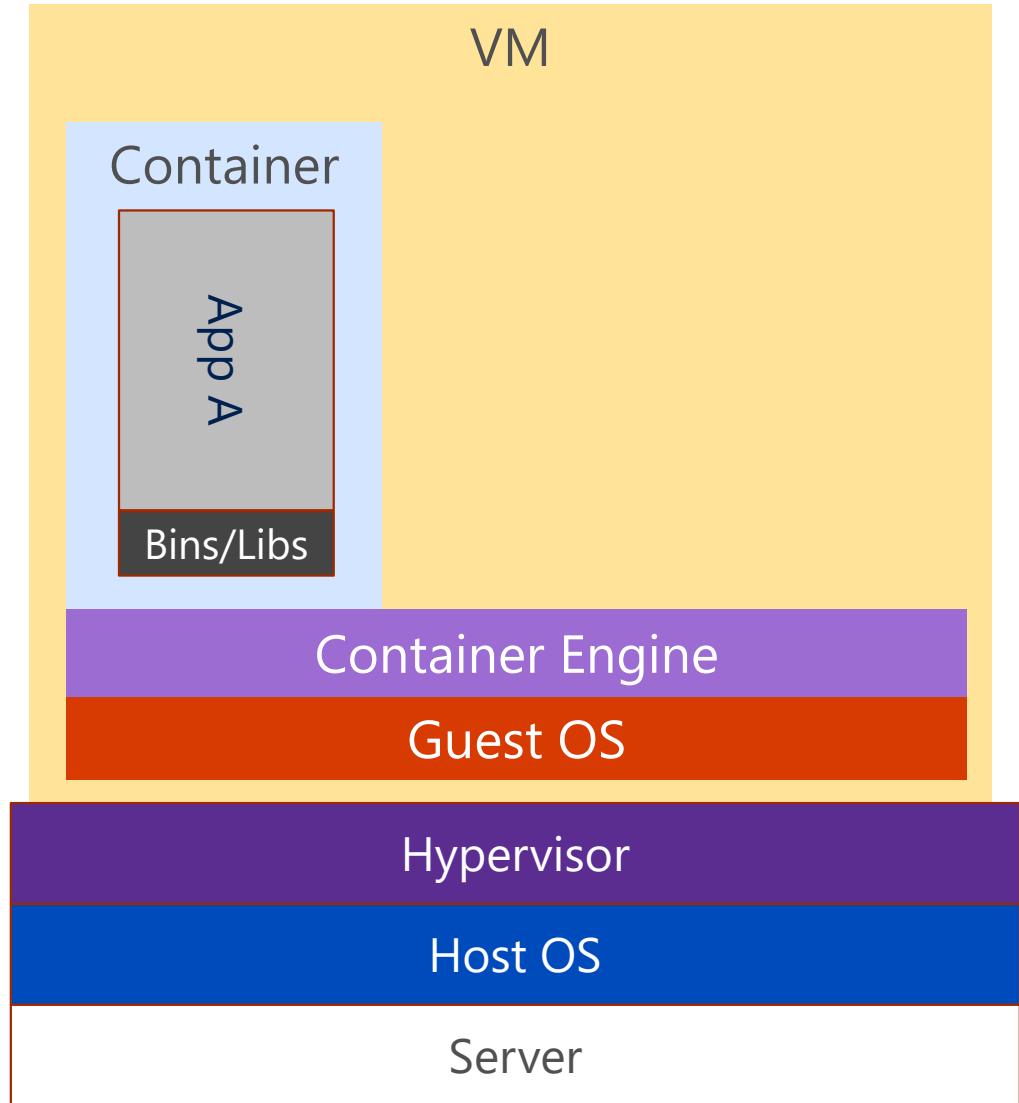
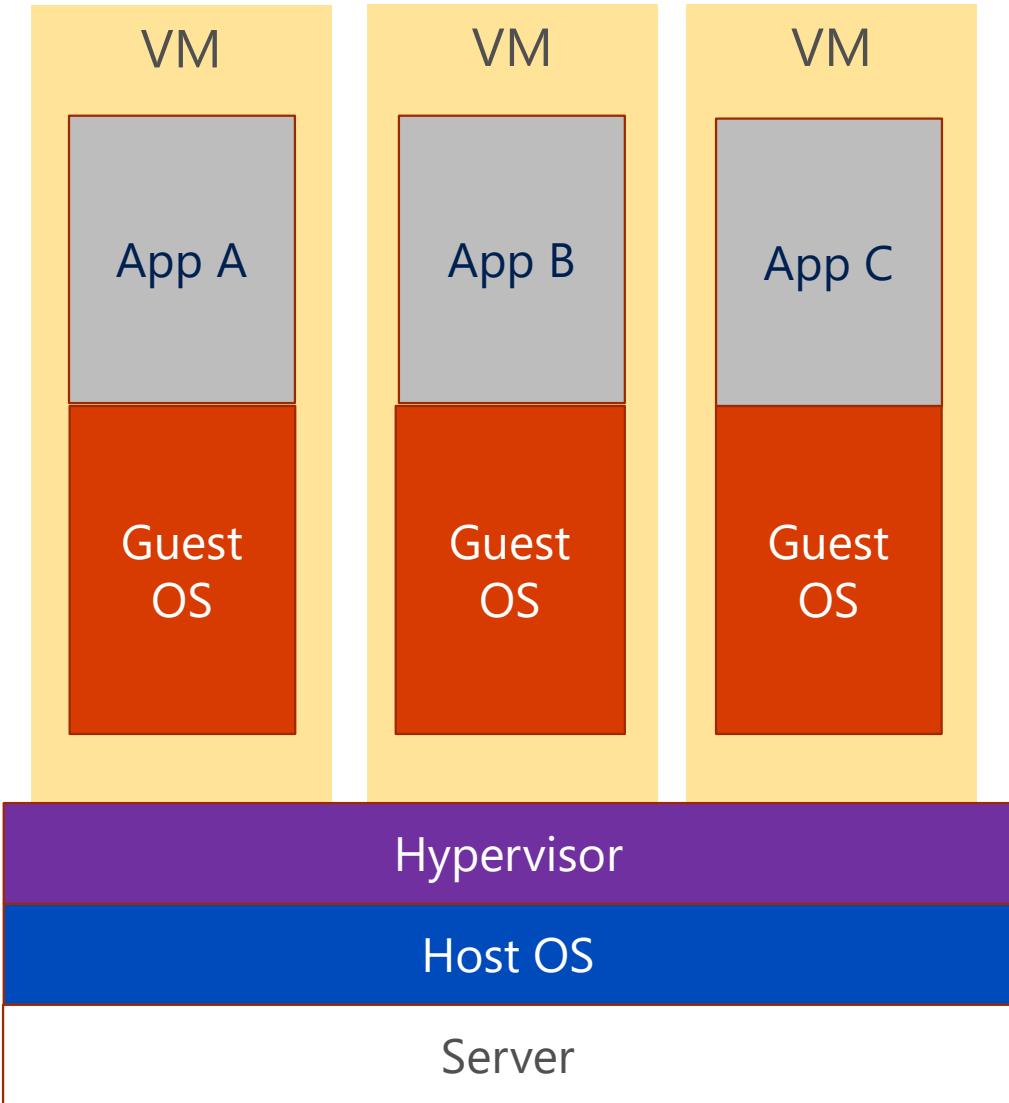
What is a Container?



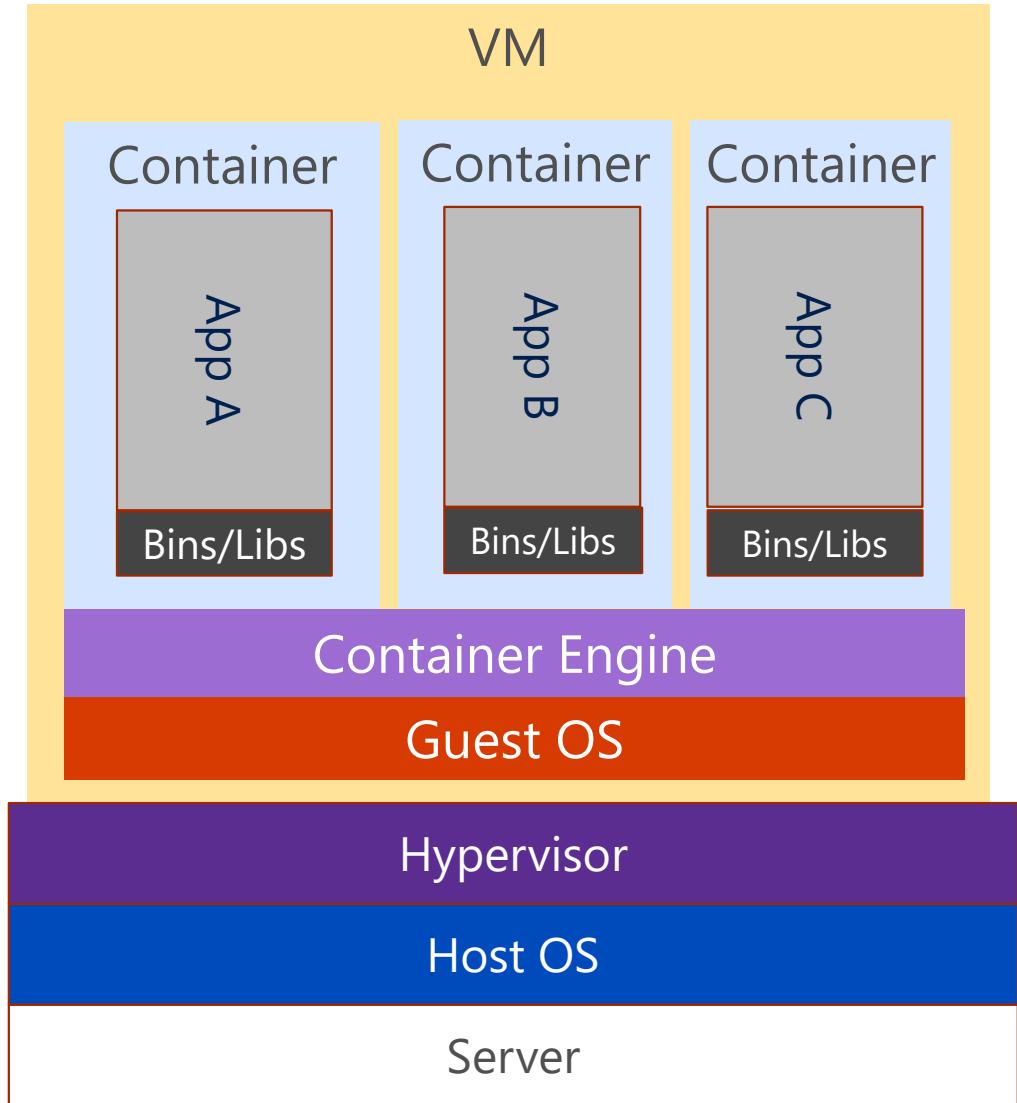
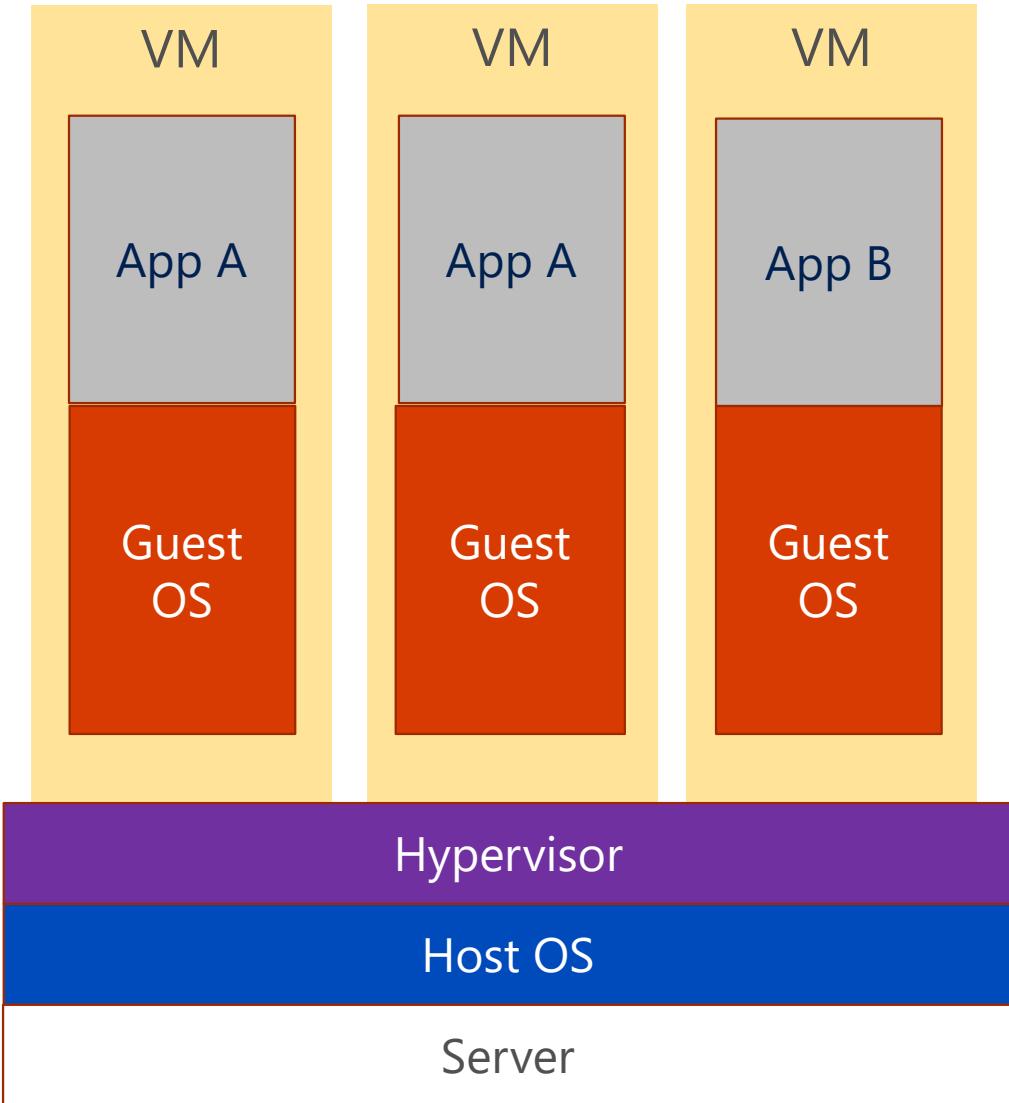
What is a Container?



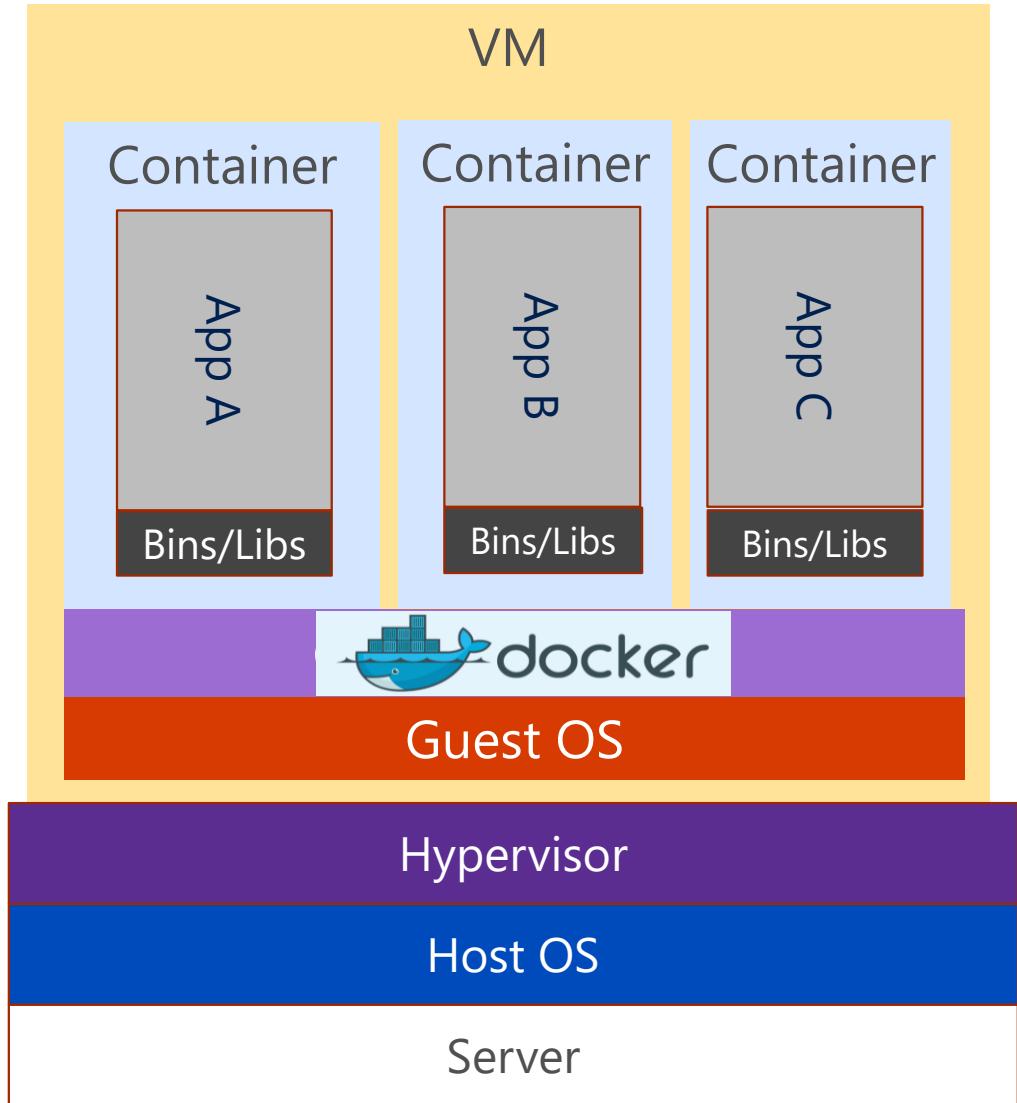
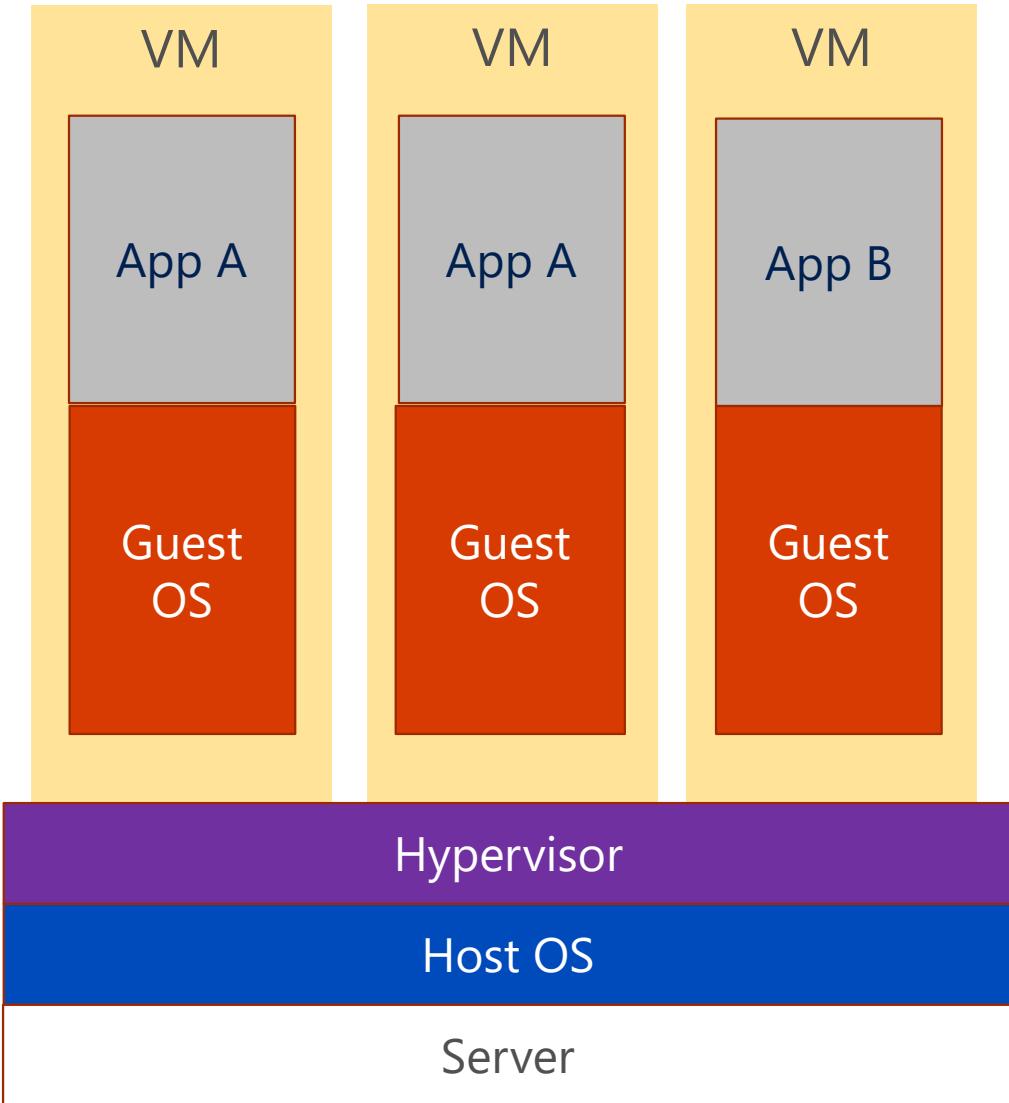
What is a Container?



What is a Container?



What is a Container?



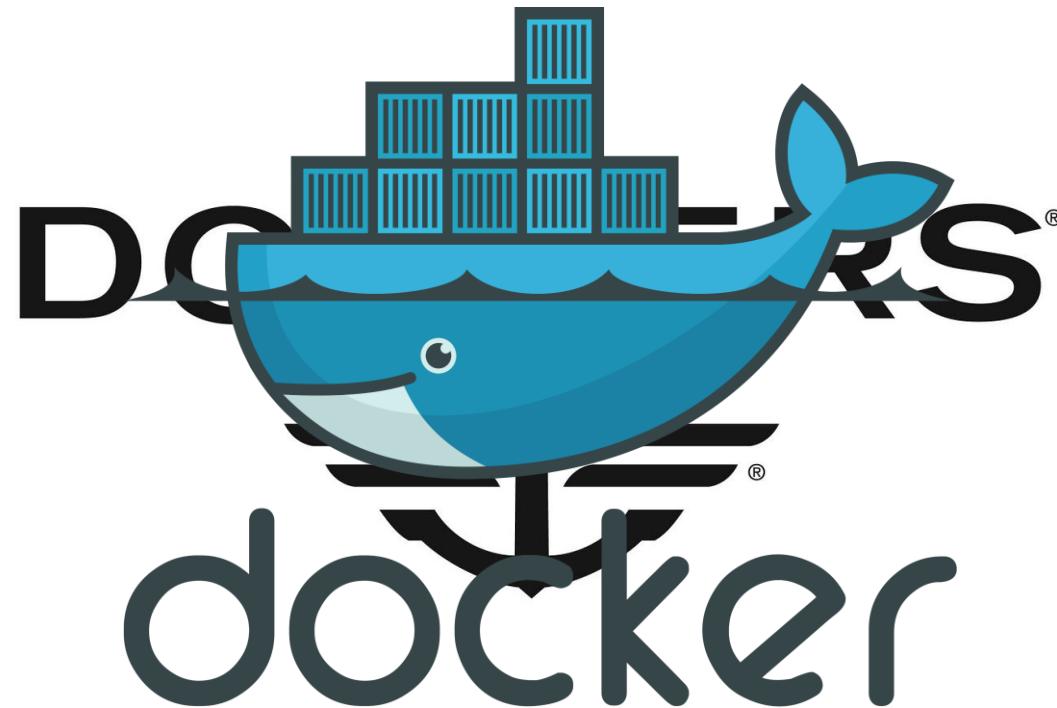
What is Docker?

- An open source container runtime
 - Works on Mac, Windows and Linux support
 - Command line tool
 - Dockerfile format for building containers
 - The Docker image format with layered file system



What is Docker?

A brief Docker demo...



Today's Scenario

- Fabrikam Medical Conferences provides conference web site services tailored to the medical community.
- After starting with a few small conferences, they now have evolved into a well-known brand and handle over 100 conferences per year, and growing.
- The technology used is the MEAN stack
 - Mongo, Express, Angular, and Node.js
 - Web sites and APIs are built as microservices hosted on Linux servers.
 - The on-prem data backend is MongoDB; also running on a separate cluster of Linux servers.

Demo

Containerize your app!

- Docker basics
- Create a Dockerfile for Fabrikam's content-api app
- Create Docker images for both the API & Web app
- Run the containers and verify that we can access the app

We'll continue at 10.25



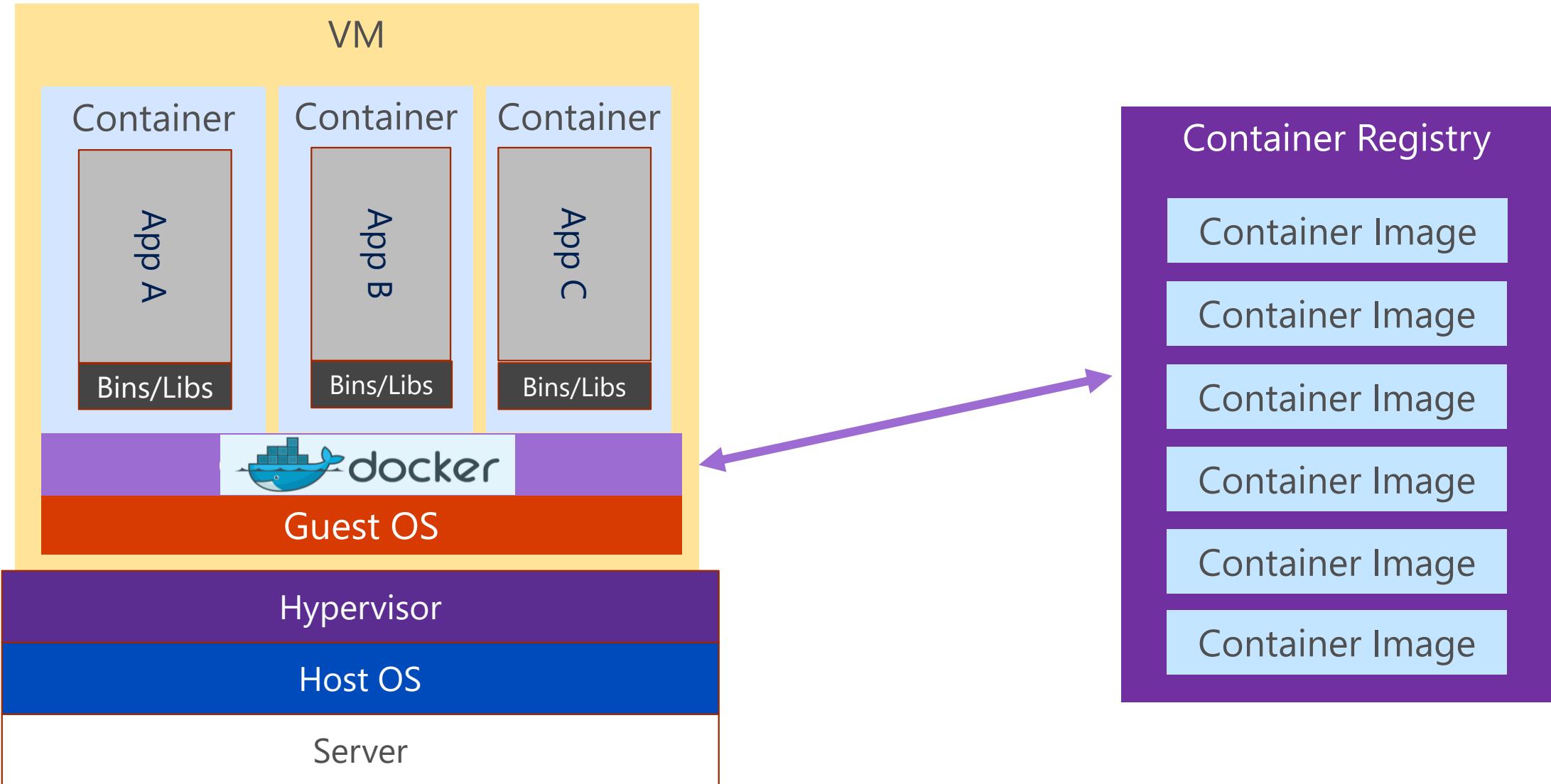
What is a Container Registry?

What is a Container Registry?

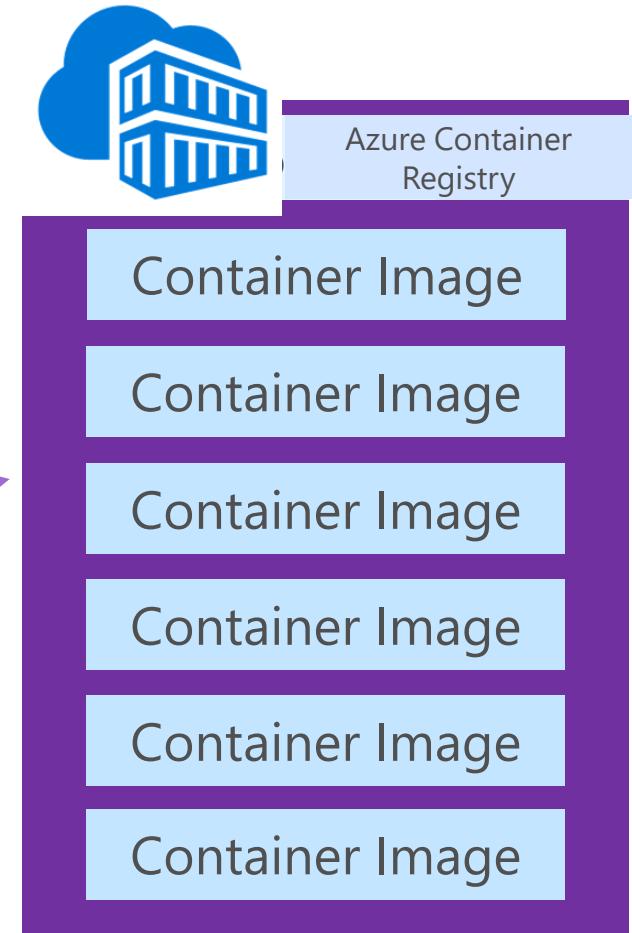
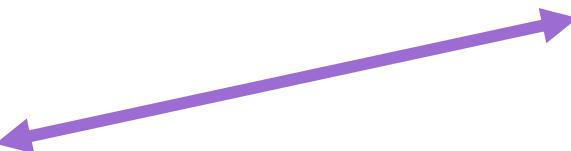
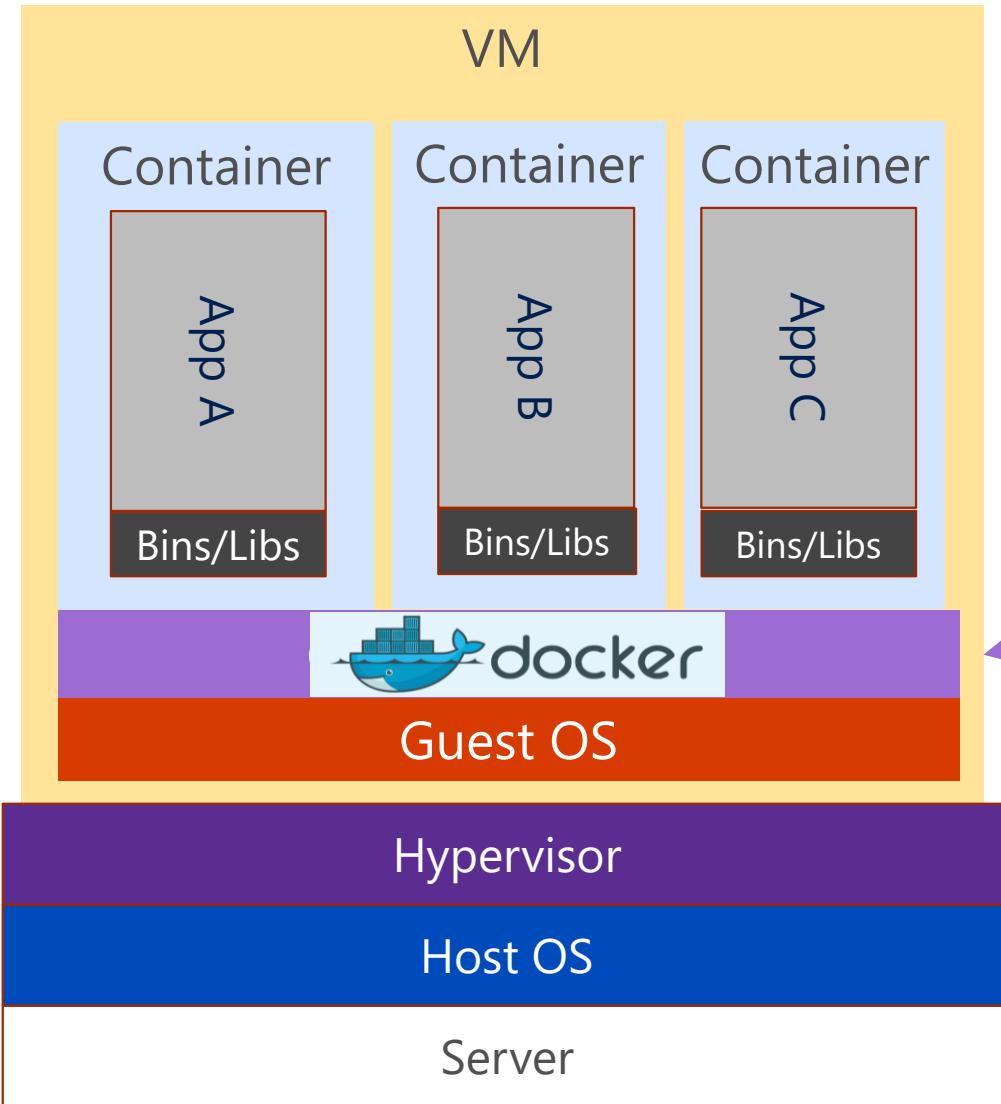
Discover your containers

- A place to store, list, and share your container images
- Like an “app store” for container images
- DockerHub is a registry hosted by Docker. Many companies publish their containers there.
- Most cloud providers offer their own Cloud Registries “as-a-service”
- Azure has the... “Azure Container Registry”!
- Docker tools can publish to a container registry
- Docker can run a container image from a registry

What is a Container Registry?



What is a Container Registry?



Azure Container Registry

- Nested repositories
 - Docker images & Helm packages
- ACR Tasks | az acr task
 - Trigger on schedule, base image change,, source code update
- Support for events and webhooks
- Support for firewalls and virtual network (preview)
- Premium
 - Content trust, Geo-Replication, Retention (preview)

Azure Container Registry

Get started

```
az acr create --name demoweuaksee1023acr --resource-group demo-weu-  
aksee1023-rg --sku Basic --admin-enabled true --location westeurope
```

```
az aks update -n demo-weu-aksee1023 -g demo-weu-aksee1023-rg --attach-acr  
demoweuaksee1023acr
```

```
az acr credential show --resource-group demo-weu-aksee1023-rg --name  
demoweuaksee1023acr
```

```
docker login demoweuaksee1023acr.azurecr.io
```

Demo

Publish your containers to a registry

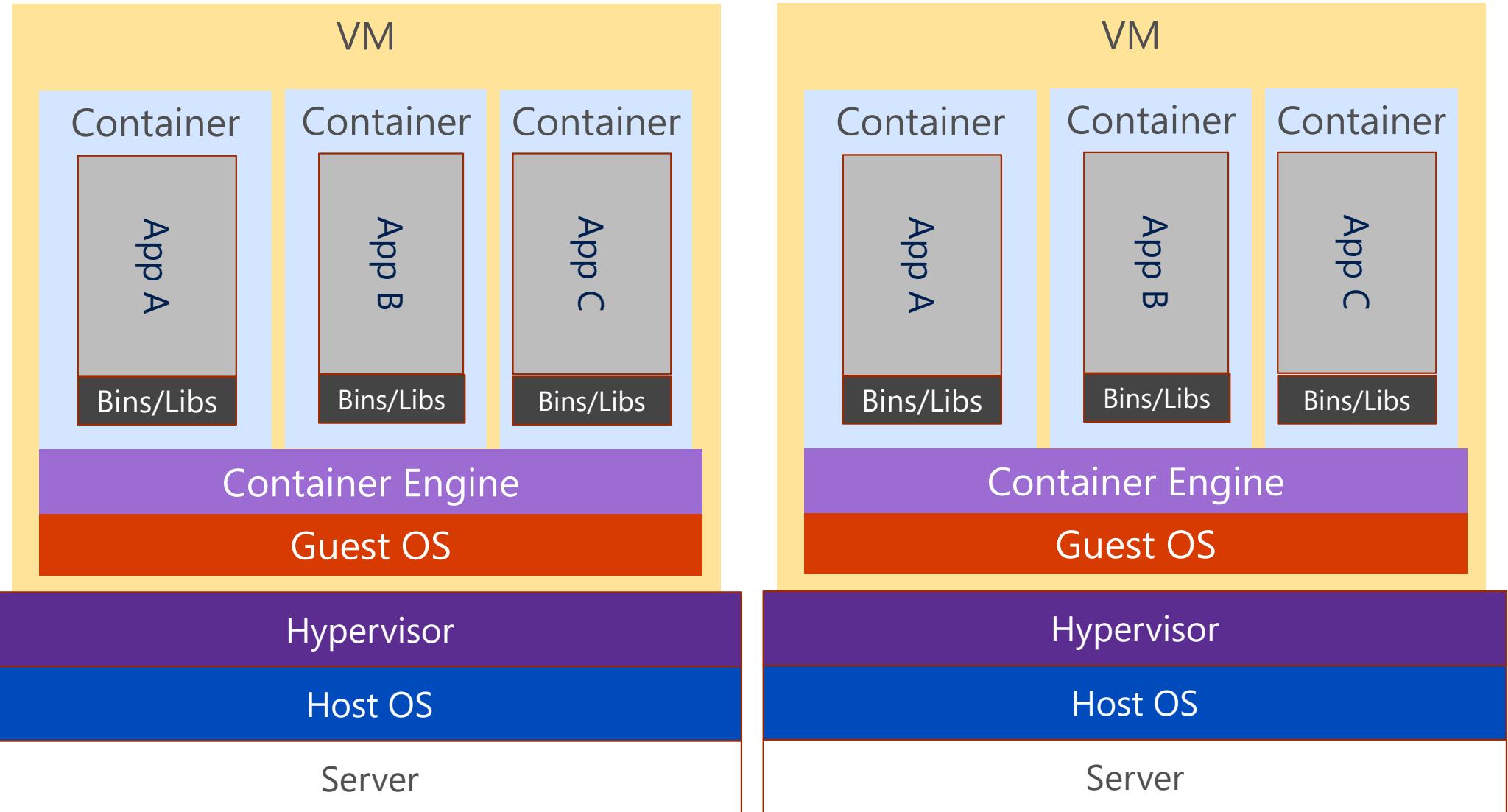
- Create an Azure Container Registry
- Set up your permissions properly
- Publish your content-api & content-web containers to ACR

Ok, I've got containers.

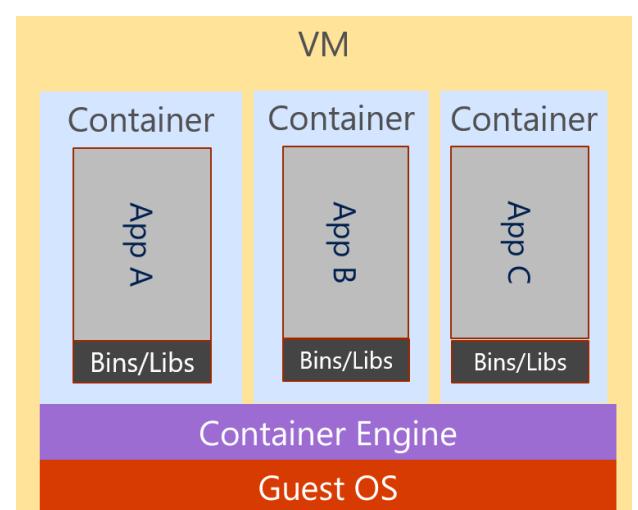
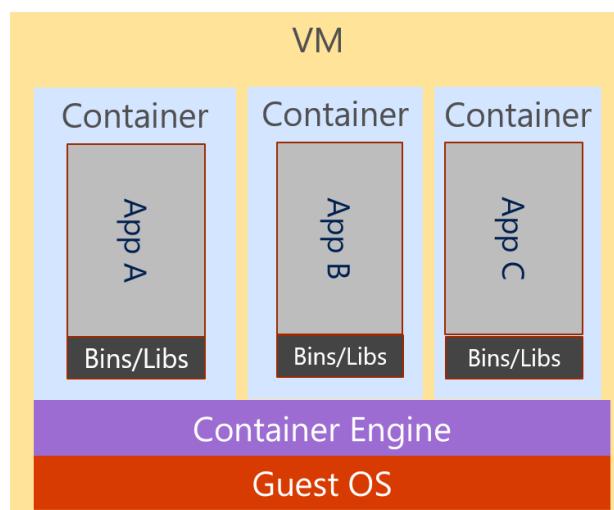
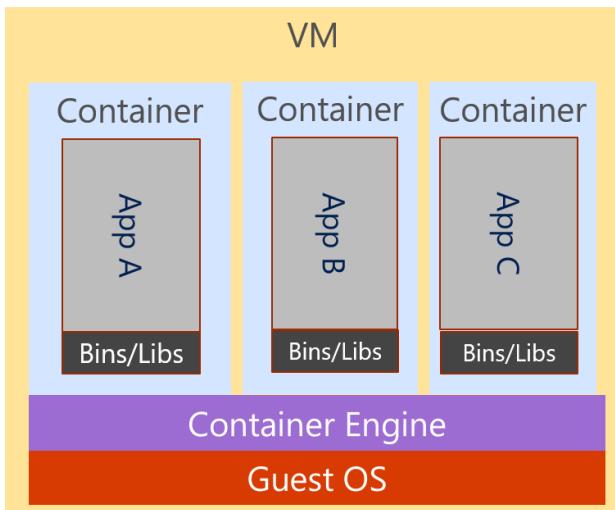
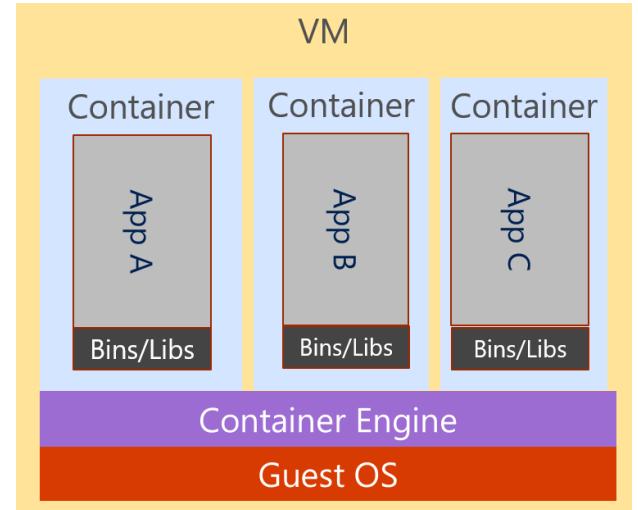
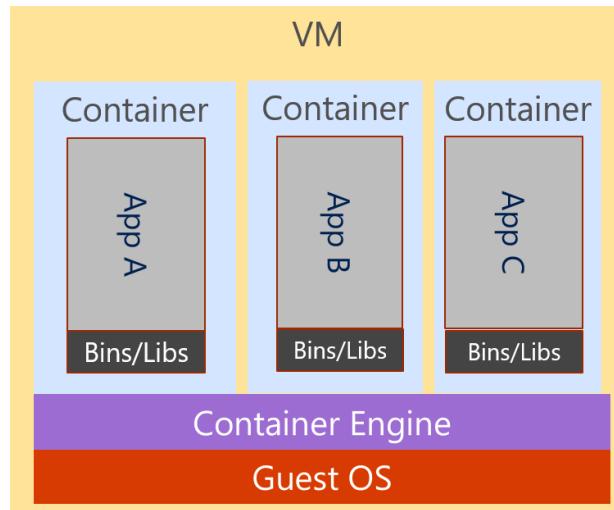
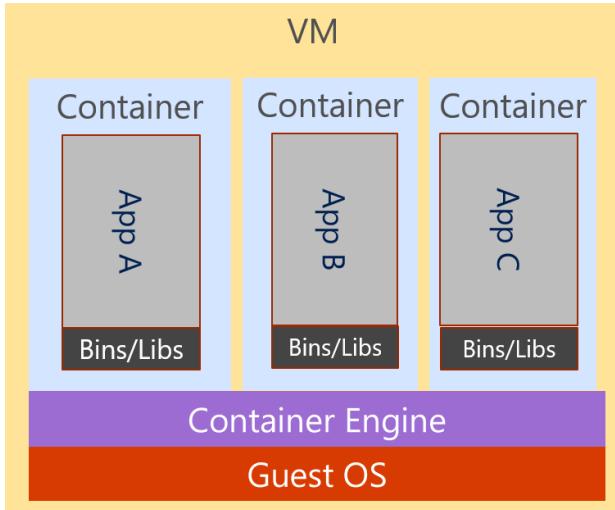
And a registry.

Now what?

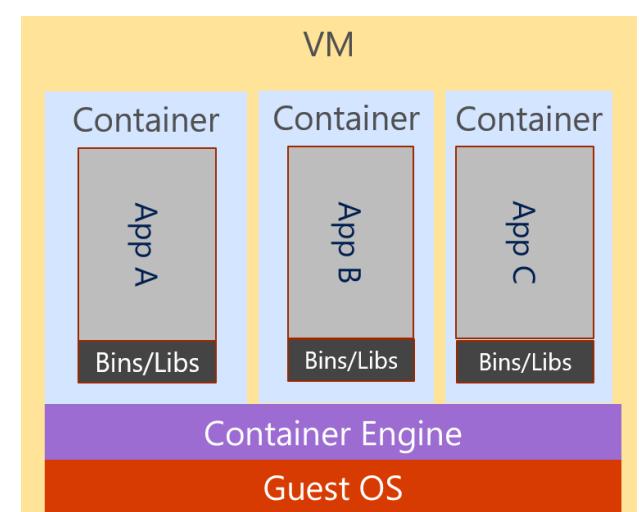
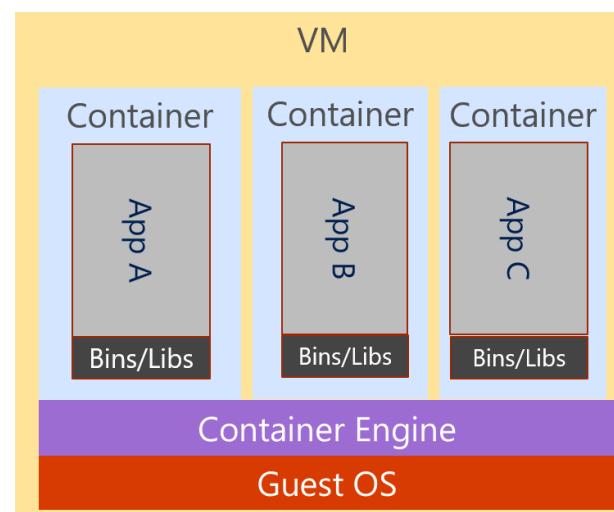
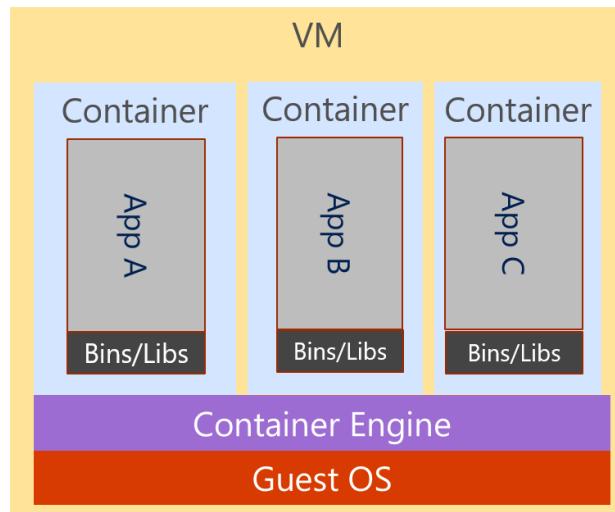
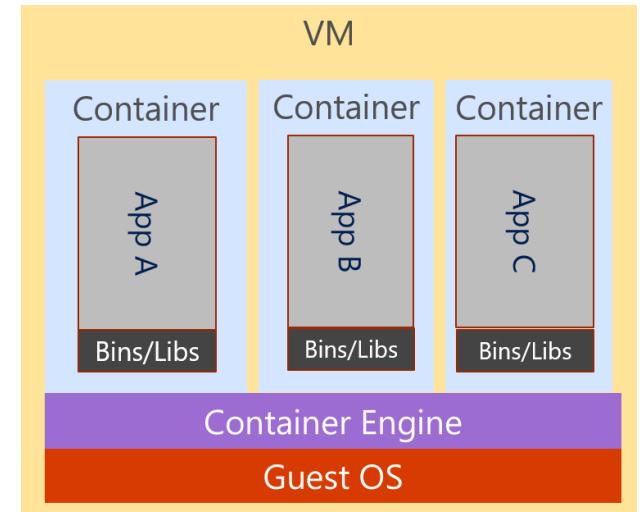
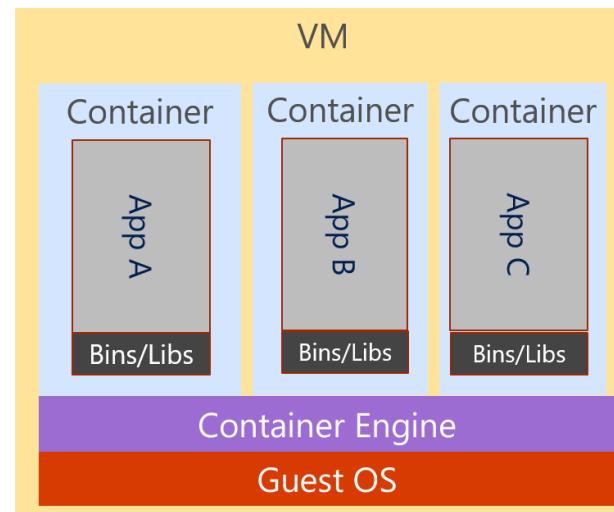
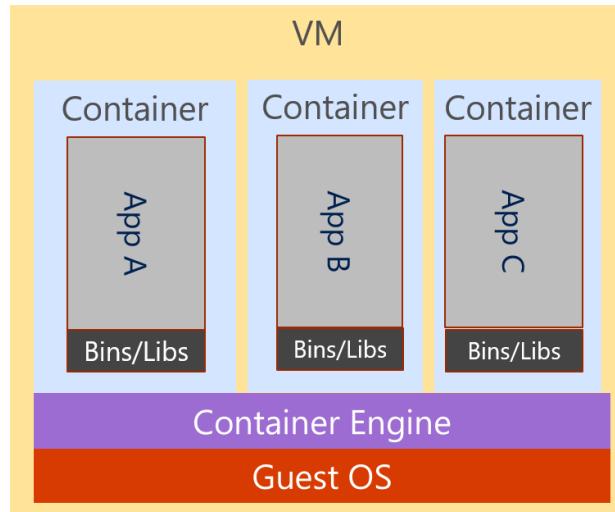
I'm going to want more VMs with containers!



And more... and more!



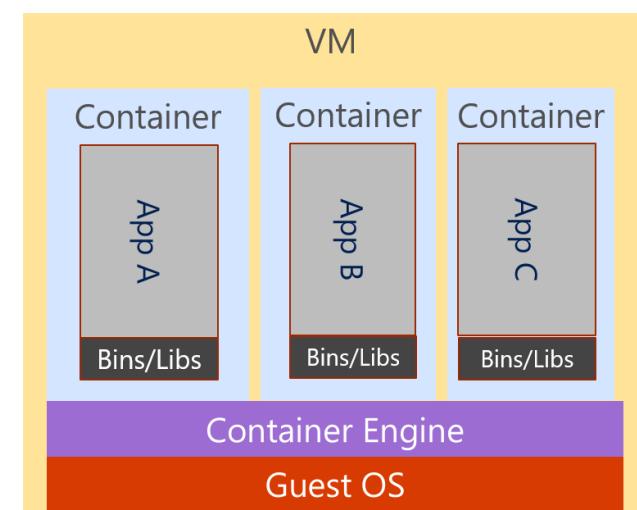
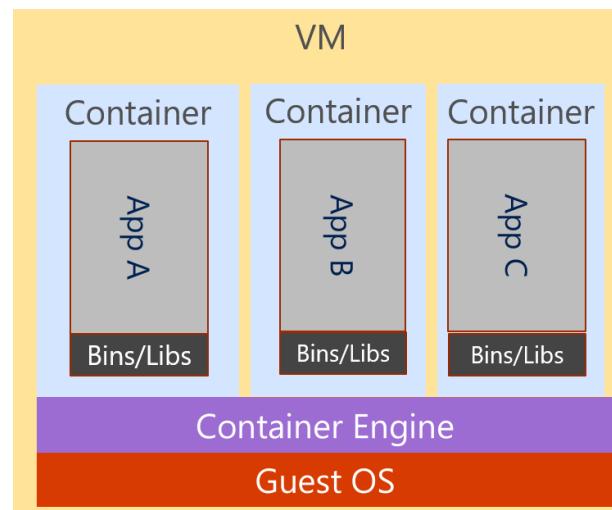
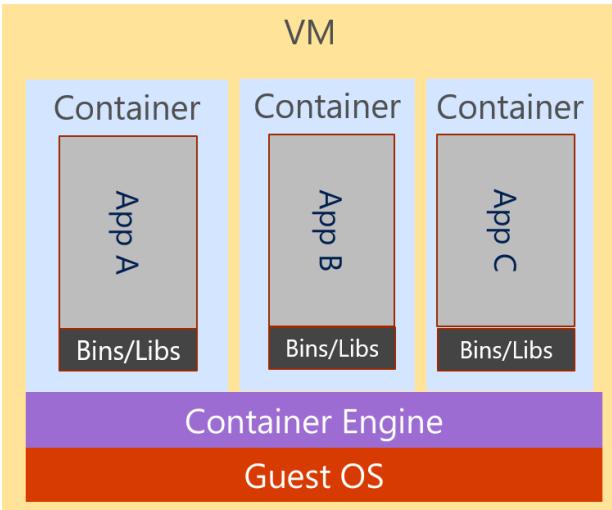
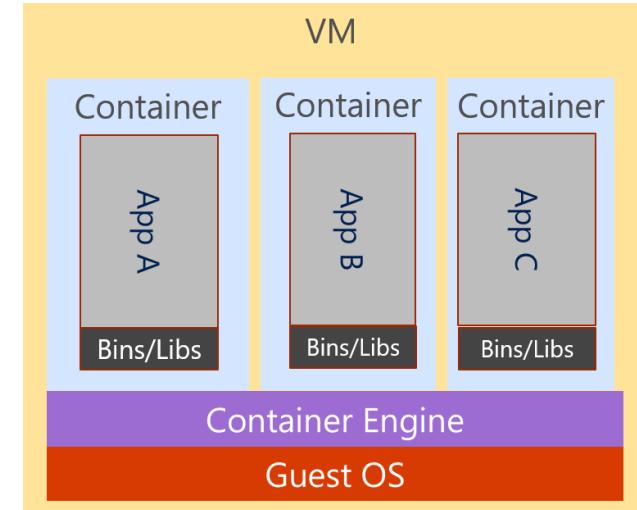
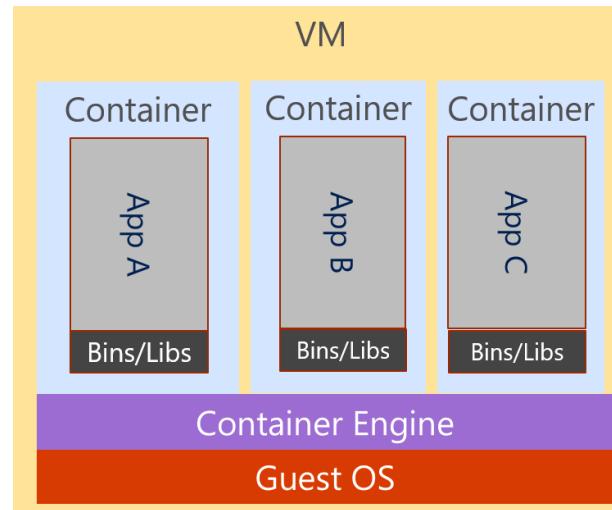
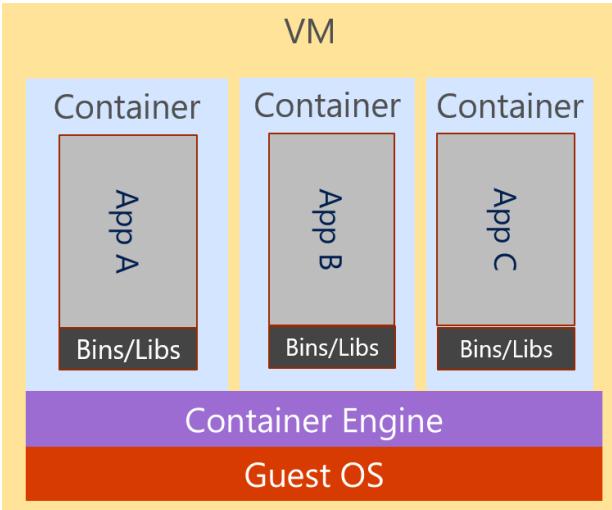
And more... and more... CHAOS!!!



What is a Container
Orchestrator?

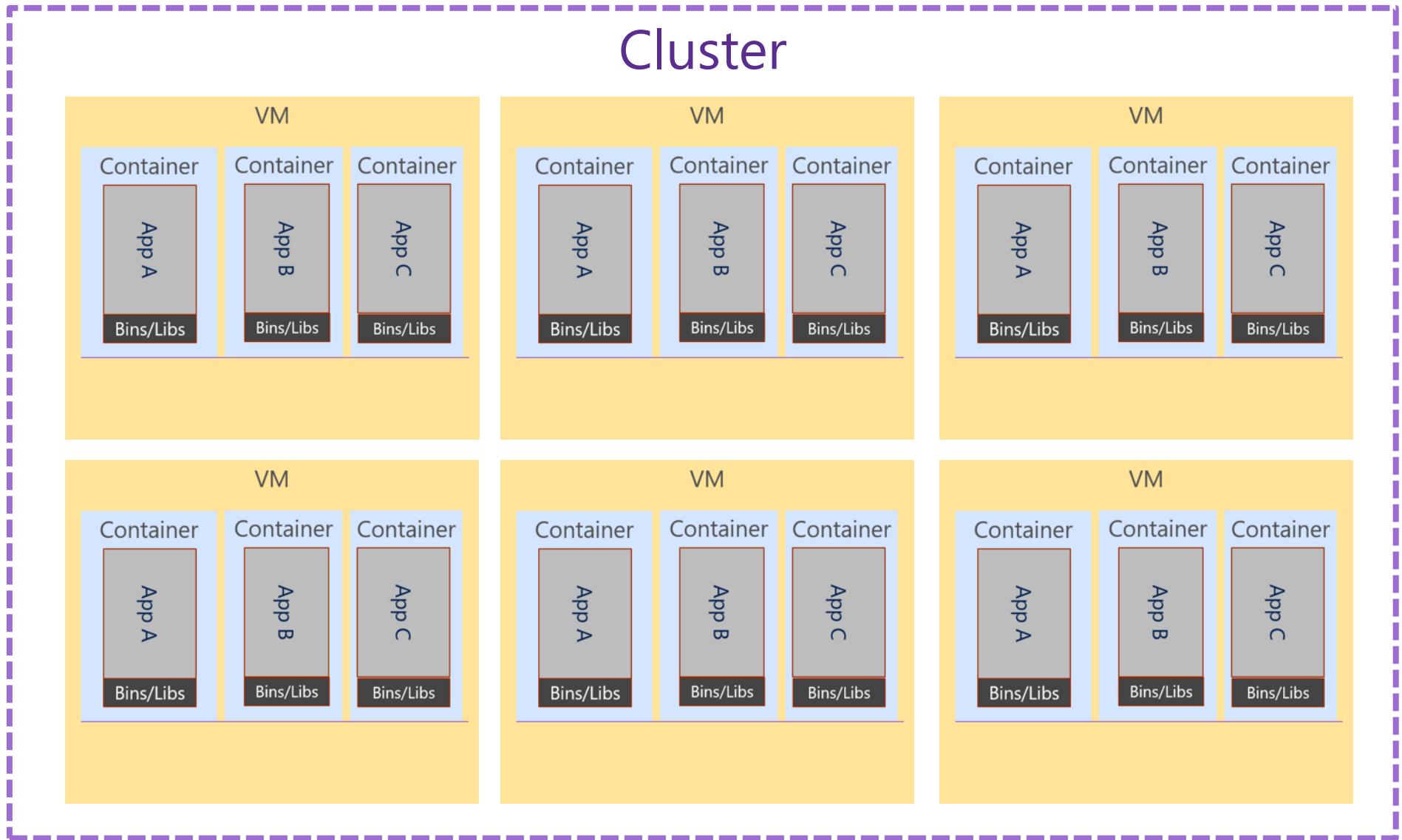
What is a Container Orchestrator?

Container Orchestrator

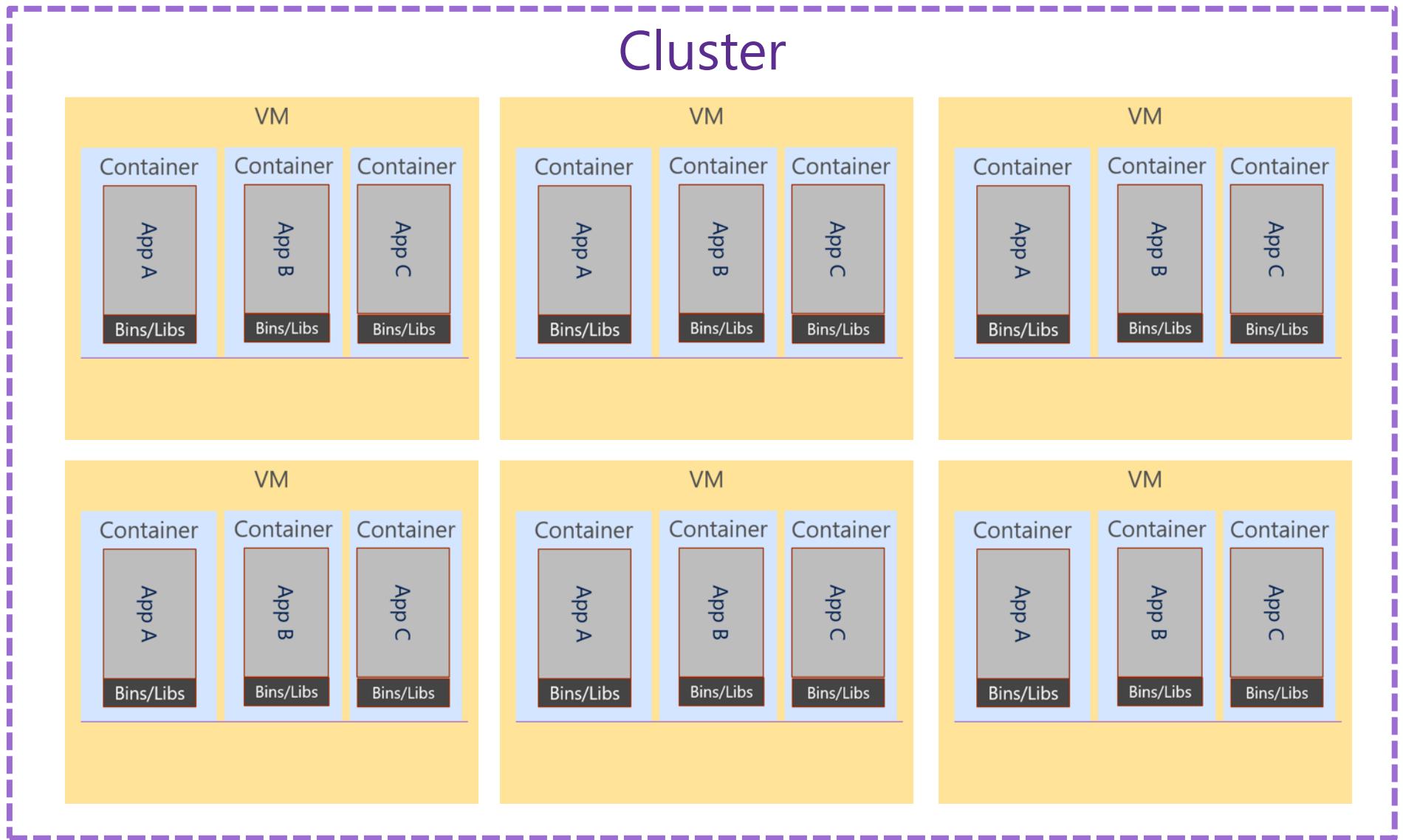


What is a Container Orchestrator?

Container Orchestrator



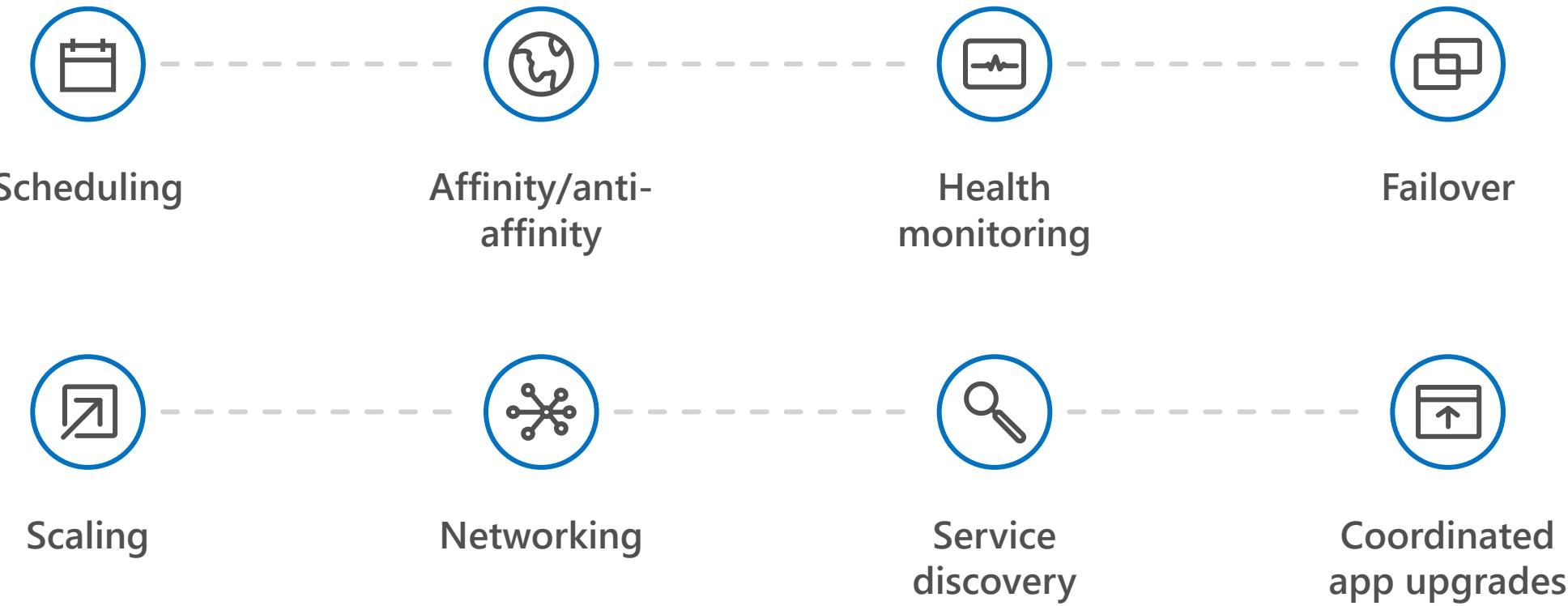
What is a Container Orchestrator?



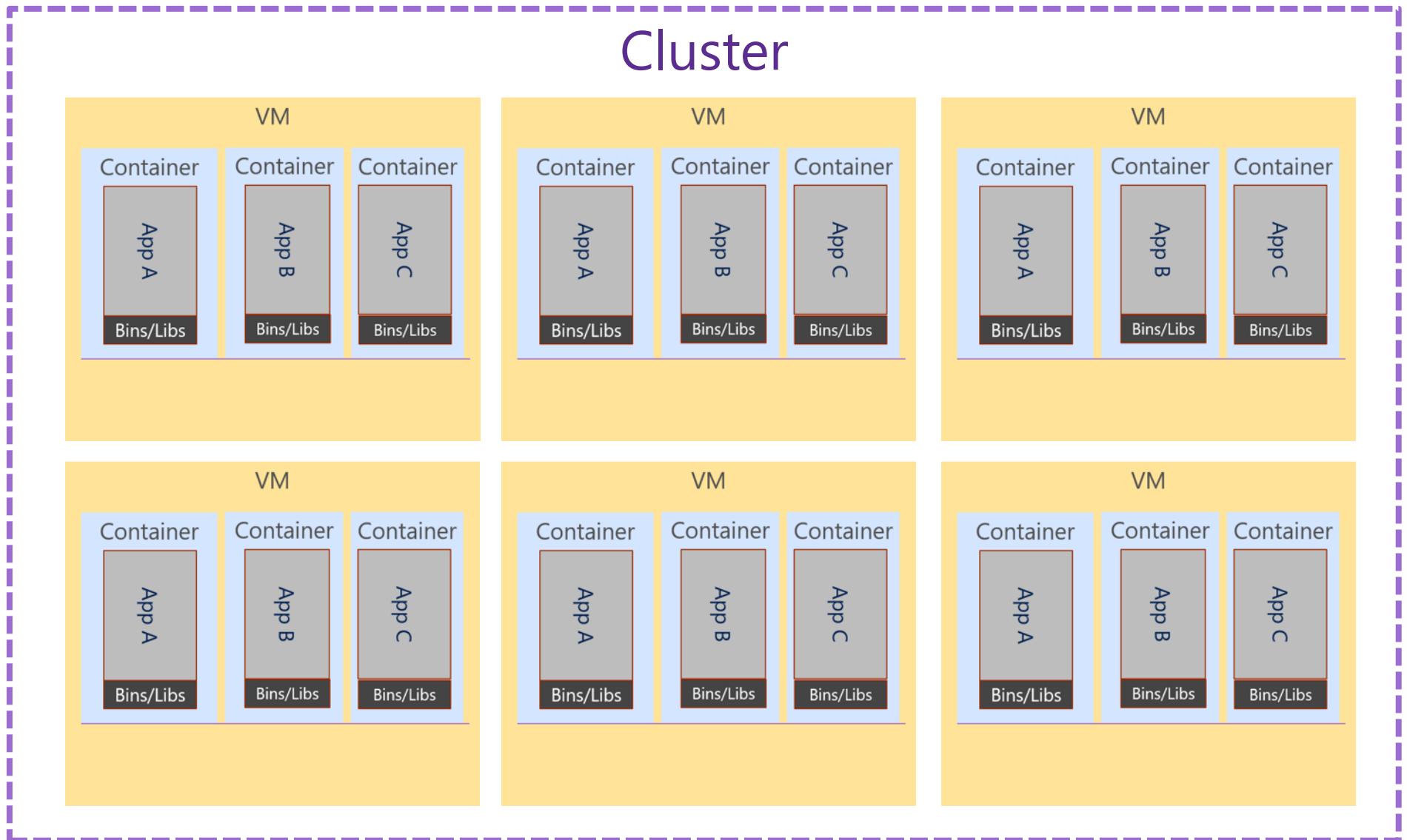
What is a Container Orchestrator?

- Container orchestrators group hosts together to form a cluster, and fulfill these requirements:
 - Are fault tolerant
 - Can scale on demand
 - Use resource optimally
 - Discover other applications automatically
 - Accessible from the external world
 - Updates and roll-backs without downtime

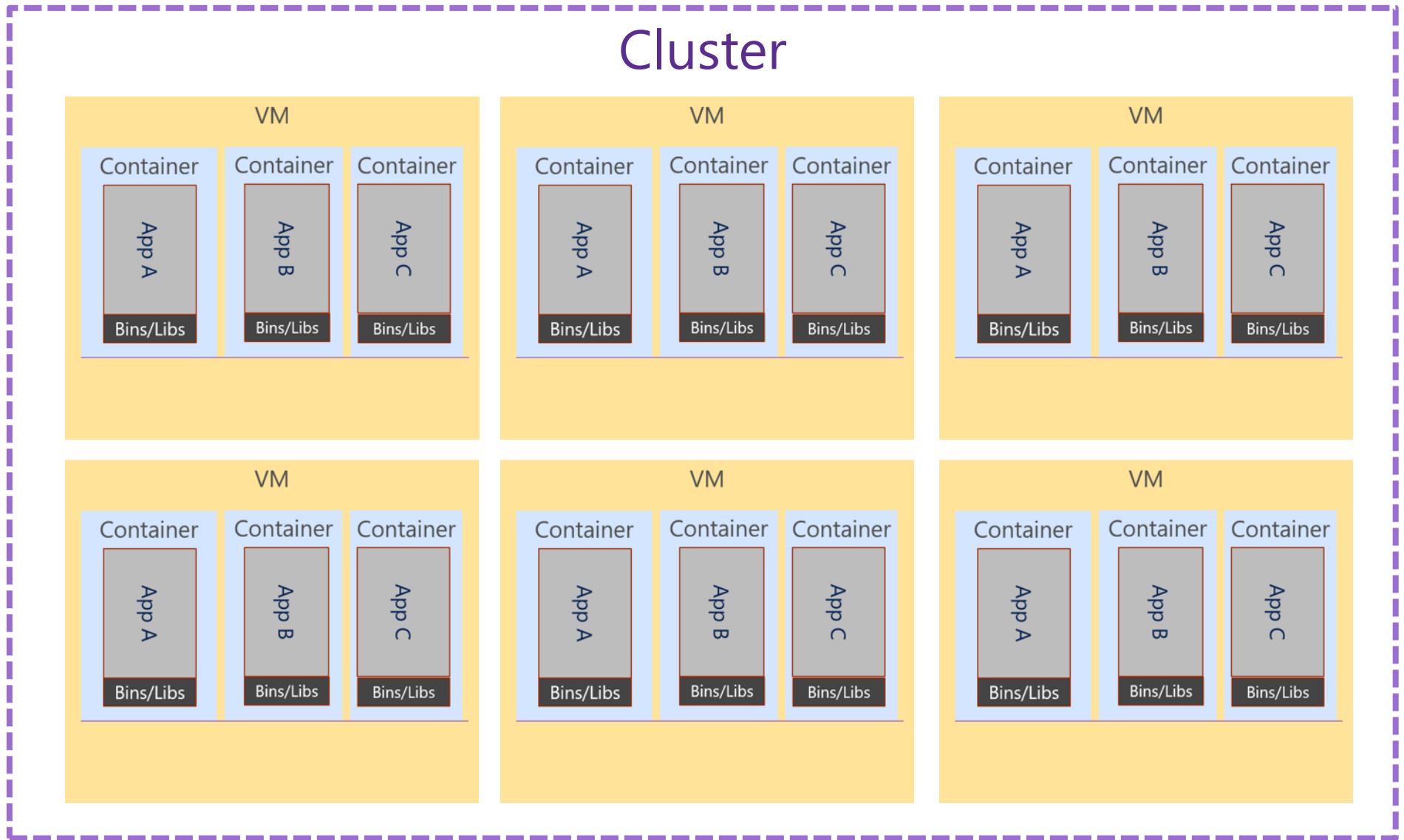
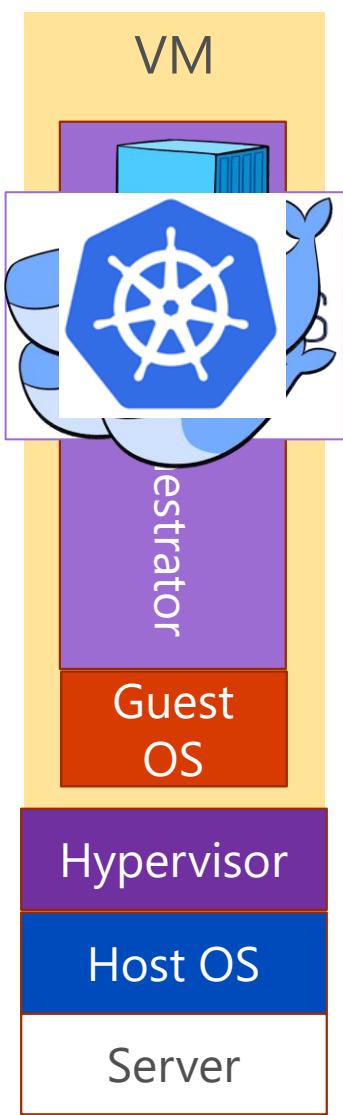
The elements of **orchestration**



What is a Container Orchestrator?



Container Orchestrator Leaders



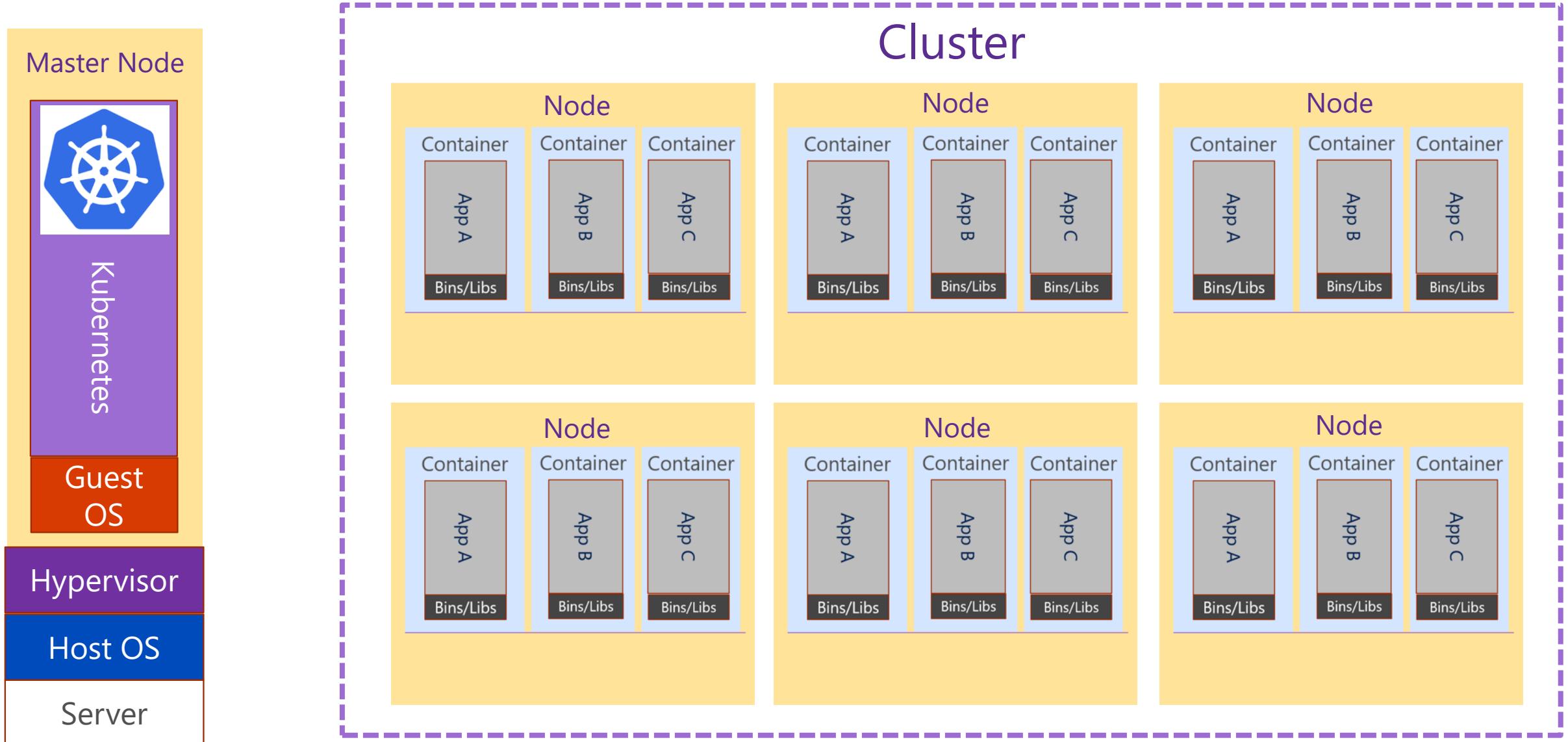
What is Kubernetes?

"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.

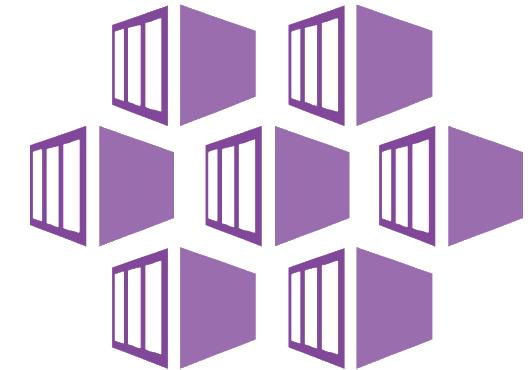


Kubernetes has a Master & Cluster of "Nodes"



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 agent nodes within your clusters, not for the master nodes

Azure structure

Account and enterprise agreement

Subscription(s)

Resource group



Role-based
access control



Tags

Resource group



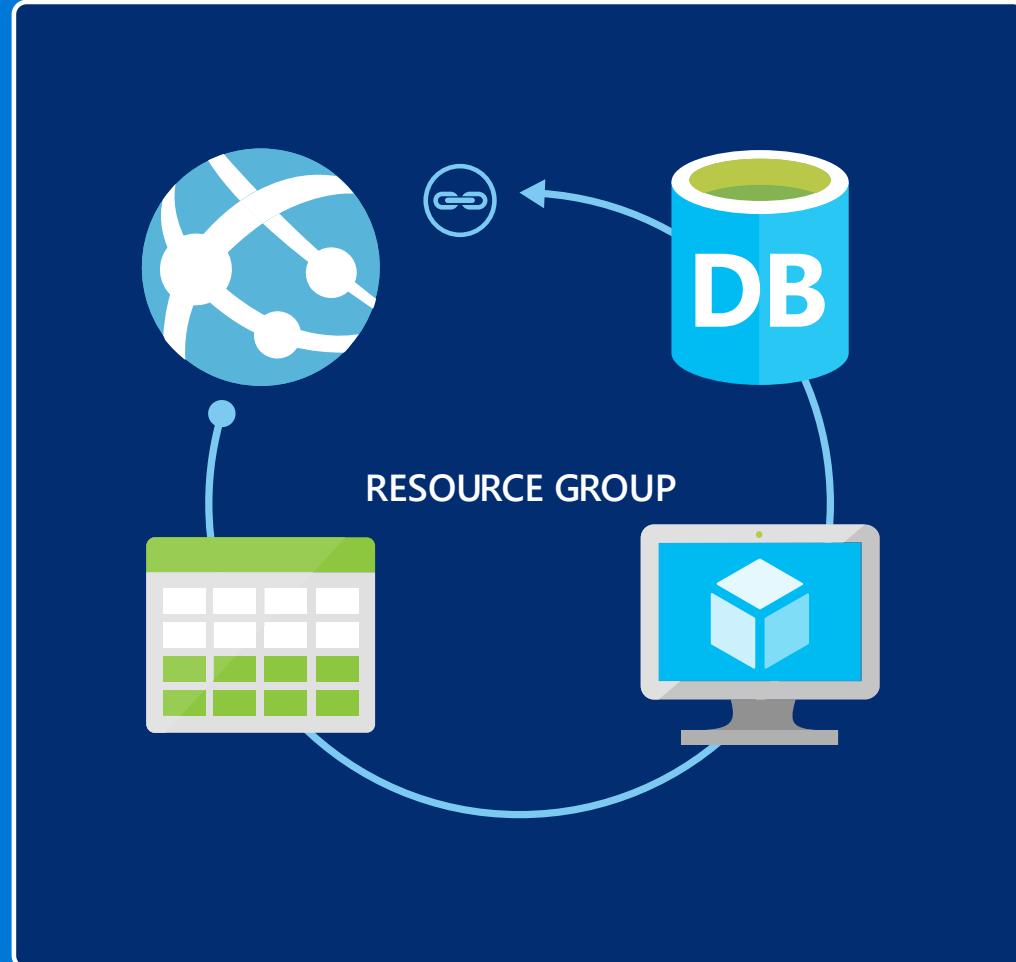
Azure Resource
Manager policy



Templates

Resource Groups

- Logical groups of multiple resources of similar or different types
- Every resource must be in a resource group, cannot be in more than one
- Resources inside resource group may span across regions
- Resource Group has a location for storing metadata

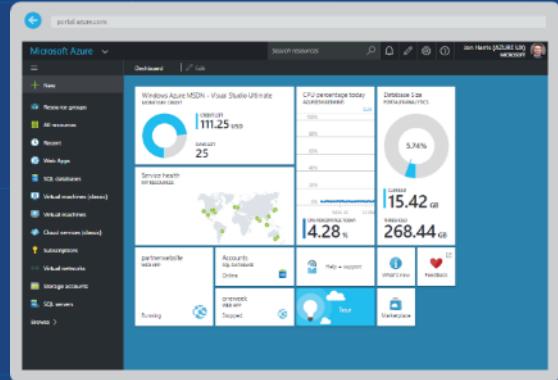


Azure Resource Manager

- Azure Resource Manager or ARM management interface for Azure



All interfaces lead to the same API...



Azure Portal

```
~/Downloads$ azure
info: Microsoft Azure: Microsoft's Cloud Platform
info: Tool version 0.9.19
help: Display help for a given command
help: help [options] [command]
help: Log in to an Azure subscription using Active Directory or a Microsoft
account identity.
help: login [options]
help: Log out from Azure subscription using Active Directory. Currently, th
e user can log out only via Microsoft organizational account
help: logout [options] [username]
help: Open the portal in a browser
```

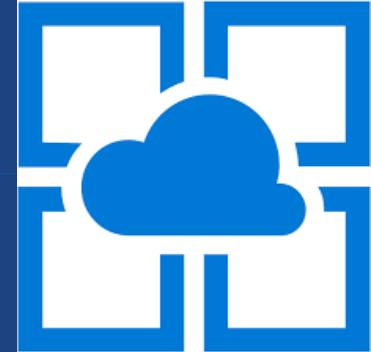
Azure CLI

```
PS C:\> Get-AzureRmServicePrincipal -SearchString "AzureKeysRollerDaemon"
DisplayNames          Type          ObjectId
-----          ----          -----
AzureKeysRollerDaemon          ServicePrincipal          672f1afa-526a-4ef6-819c-975c7cd79022

PS C:\> New-AzureRmRoleAssignment -ObjectId 672f1afa-526a-4ef6-819c-975c7cd79022 -RoleDefinitionName Contributor -Scope /subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e

RoleAssignmentId : /subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/providers/Microsoft.Authorization/roleAssignments/9182ea3d-4219-4c26-a88b-891ba649ae36
Scope             : /subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e
DisplayName       : AzureKeysRollerDaemon
SignInName        :
RoleDefinitionName : Contributor
RoleDefinitionId  : b24988ac-6180-42a0-ab88-20f7382dd24c
ObjectId          : 672f1afa-526a-4ef6-819c-975c7cd79022
ObjectType        : ServicePrincipal
```

Azure PowerShell



Azure Resource Manager

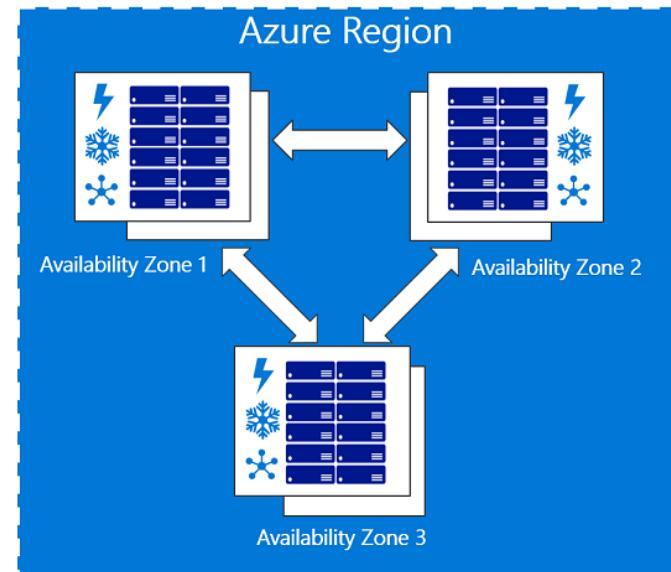
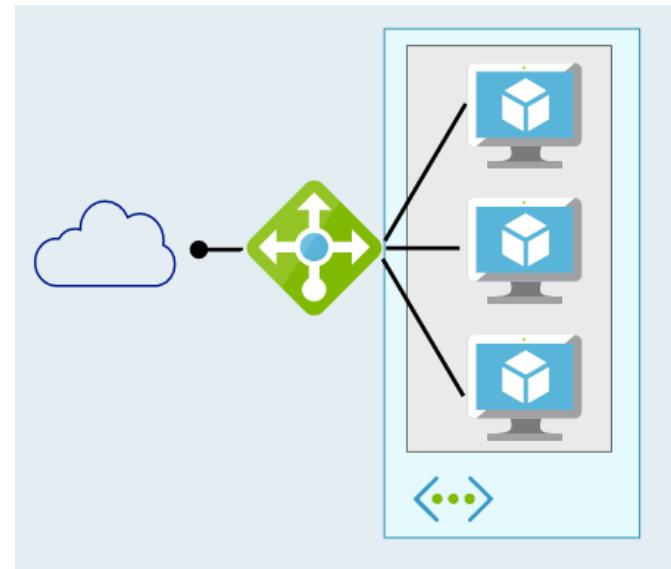


Azure API

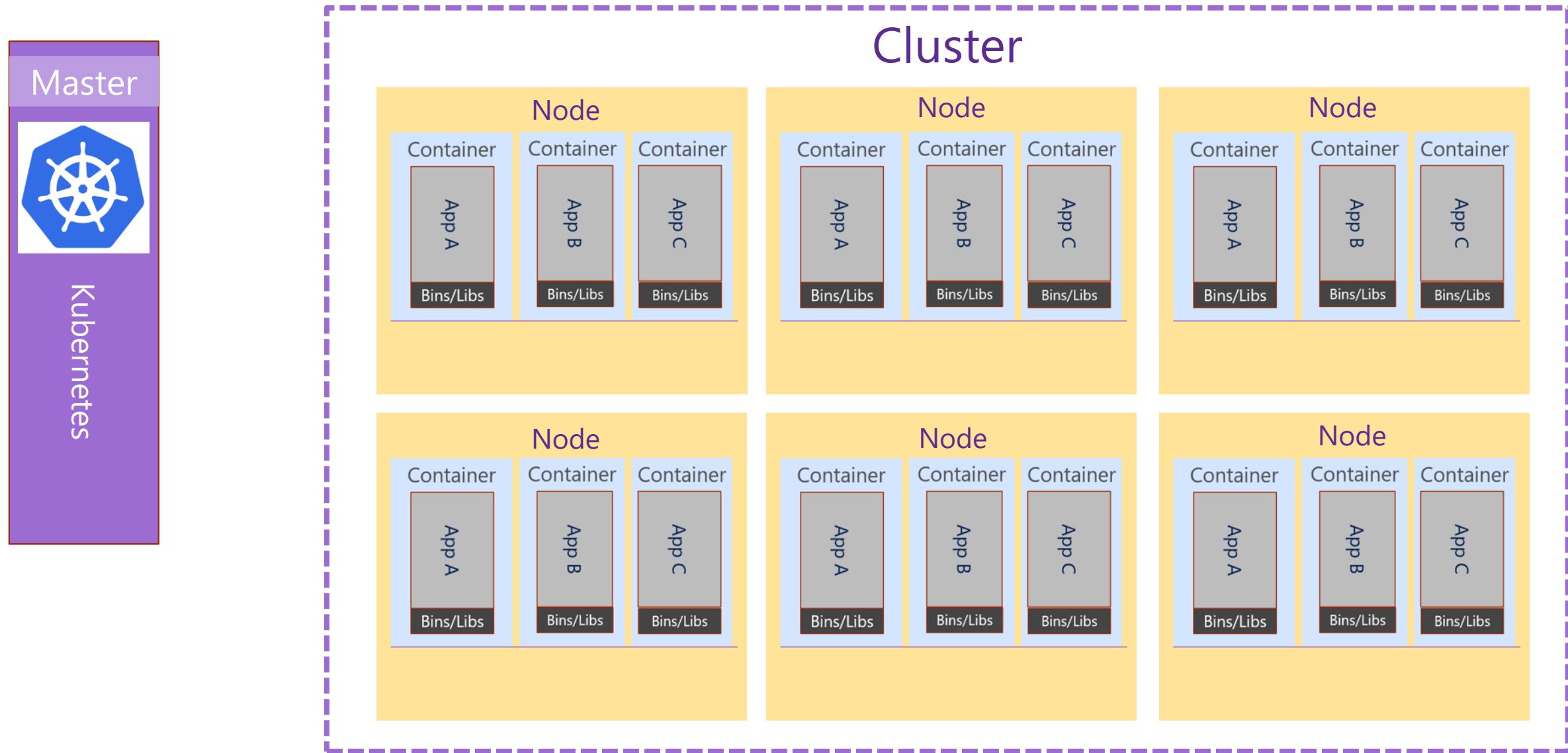
High Availability

How do you implement it?

- Availability Sets – 99.95% SLA
 - 2 or more VMs in an Availability Set configured behind a Load Balancer
- Availability Zones – 99.99% SLA
 - 2 or more VMs spread across Availability Zones configured behind a Standard Load Balancer



AKS is Kubernetes (Master) as a Service





Azure Kubernetes Service (AKS)

Get started easily

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys
\ Running ..
```

```
$ az aks install-cli
Downloading client to /usr/local/bin/kubectl ..
```

```
$ az aks get-credentials -g myResourceGroup -n myCluster
Merged "myCluster" as current context ..
```

```
$ kubectl get nodes
NAME                  STATUS  AGE   VERSION
aks-mycluster-36851231-0  Ready  4m    v1.8.1
aks-mycluster-36851231-1  Ready  4m    v1.8.1
aks-mycluster-36851231-2  Ready  4m    v1.8.1
```



Azure Kubernetes Service (AKS)

Manage an AKS cluster

```
$ az aks list -o table
```

Name	Location	ResourceGroup	KubernetesRelease	ProvisioningState
myCluster	westus2	myResourceGroup	1.7.7	Succeeded

```
$ az aks upgrade -g myResourceGroup -n myCluster --kubernetes-version 1.8.1
\ Running ..
```

```
$ kubectl get nodes
```

NAME	STATUS	AGE	VERSION
aks-mycluster-36851231-0	Ready	12m	v1.8.1
aks-mycluster-36851231-1	Ready	8m	v1.8.1
aks-mycluster-36851231-2	Ready	3m	v1.8.1

```
$ az aks scale -g myResourceGroup -n myCluster --agent-count 10
\ Running ..
```



Azure Kubernetes
Service (AKS)

Azure Kubernetes Service (AKS)

Create AKS via Portal, Azure CLI, or ARM template

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation sidebar includes links for 'Create a resource', 'All services', 'Dashboard', 'All resources', 'Resource groups', 'App Services', 'SQL databases', and 'Azure Cosmos DB'. The main content area displays a terminal window with the following command history:

```
Last login: Wed Oct 25 11:53:45 on ttys002
→ chzbrgr71 az group create --name myResourceGroup --location westus2
{
  "id": "/subscriptions/471d33fd-a776-405b-947c-467c291dc741/resourceGroups/myResourceGroup",
  "location": "westus2",
  "managedBy": null,
  "name": "myResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
→ chzbrgr71 az aks create --resource-group myResourceGroup --name myK8sCluster --agent-count 1 --generate-ssh-keys
```

Demo

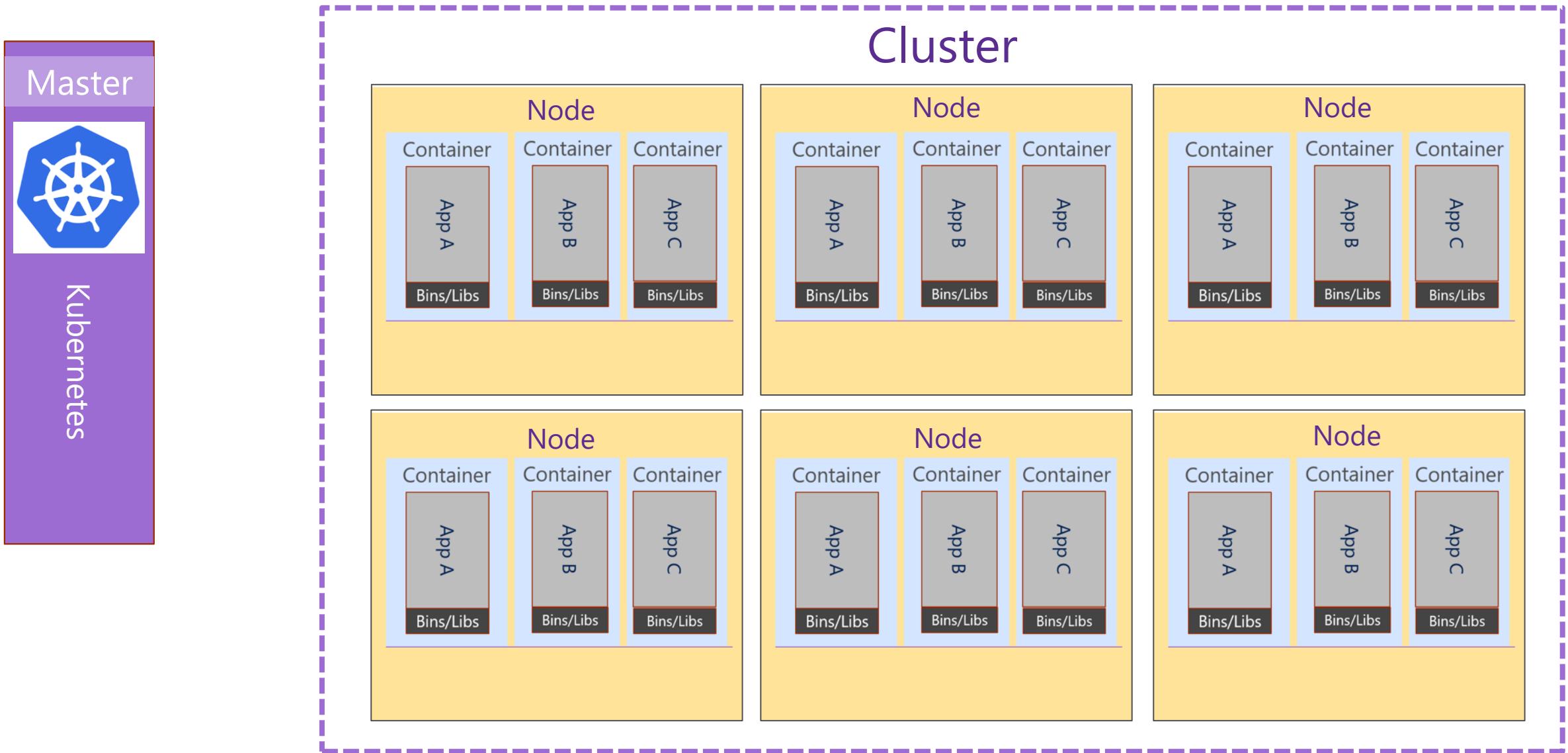
Our First Kubernetes Cluster

- Get “kubectl” installed
- Create a new, multi-node AKS cluster
- Use kubectl to prove that the cluster is a multi-node cluster and is working

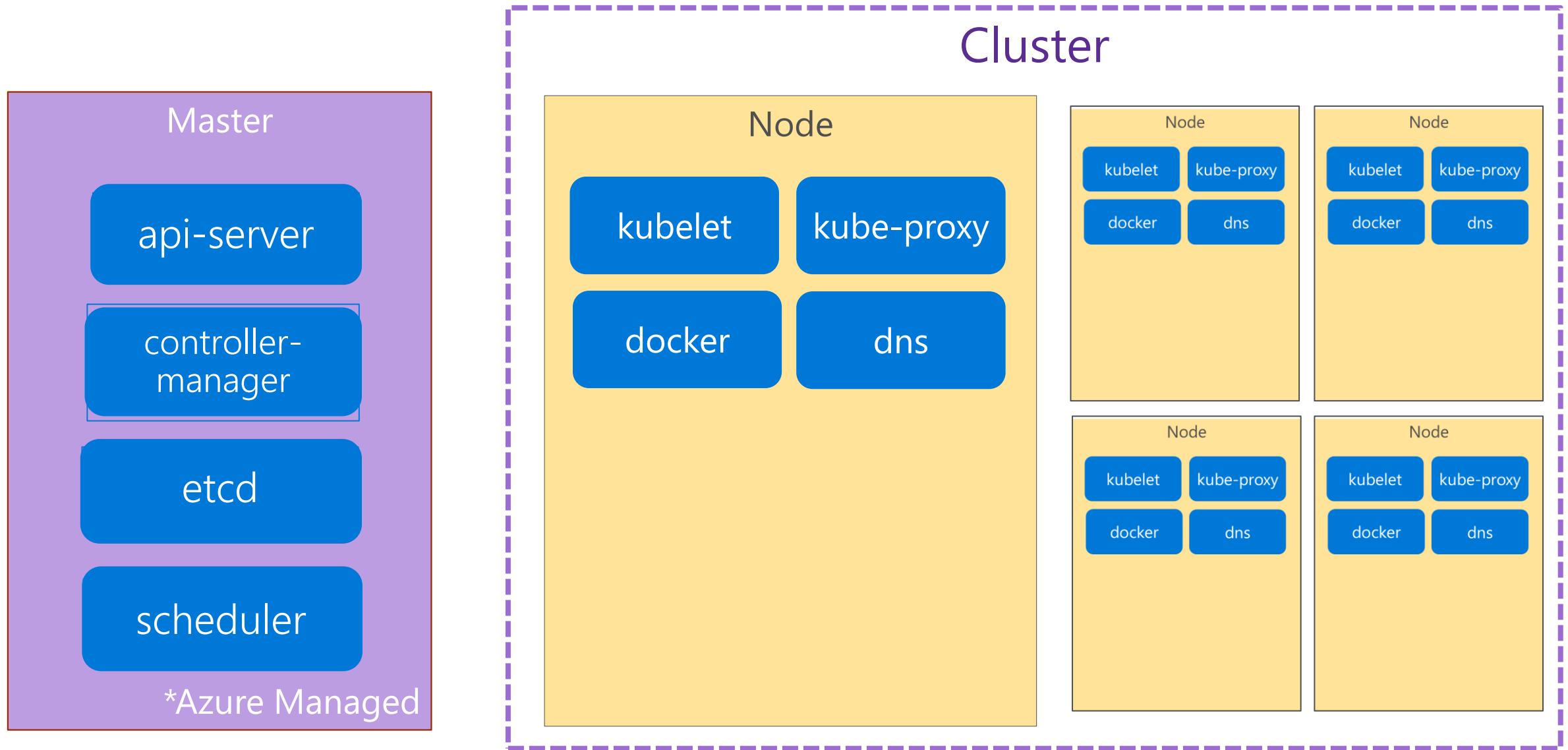
Kubernetes Fundamentals

Pods, Deployments, & Services, Oh my!

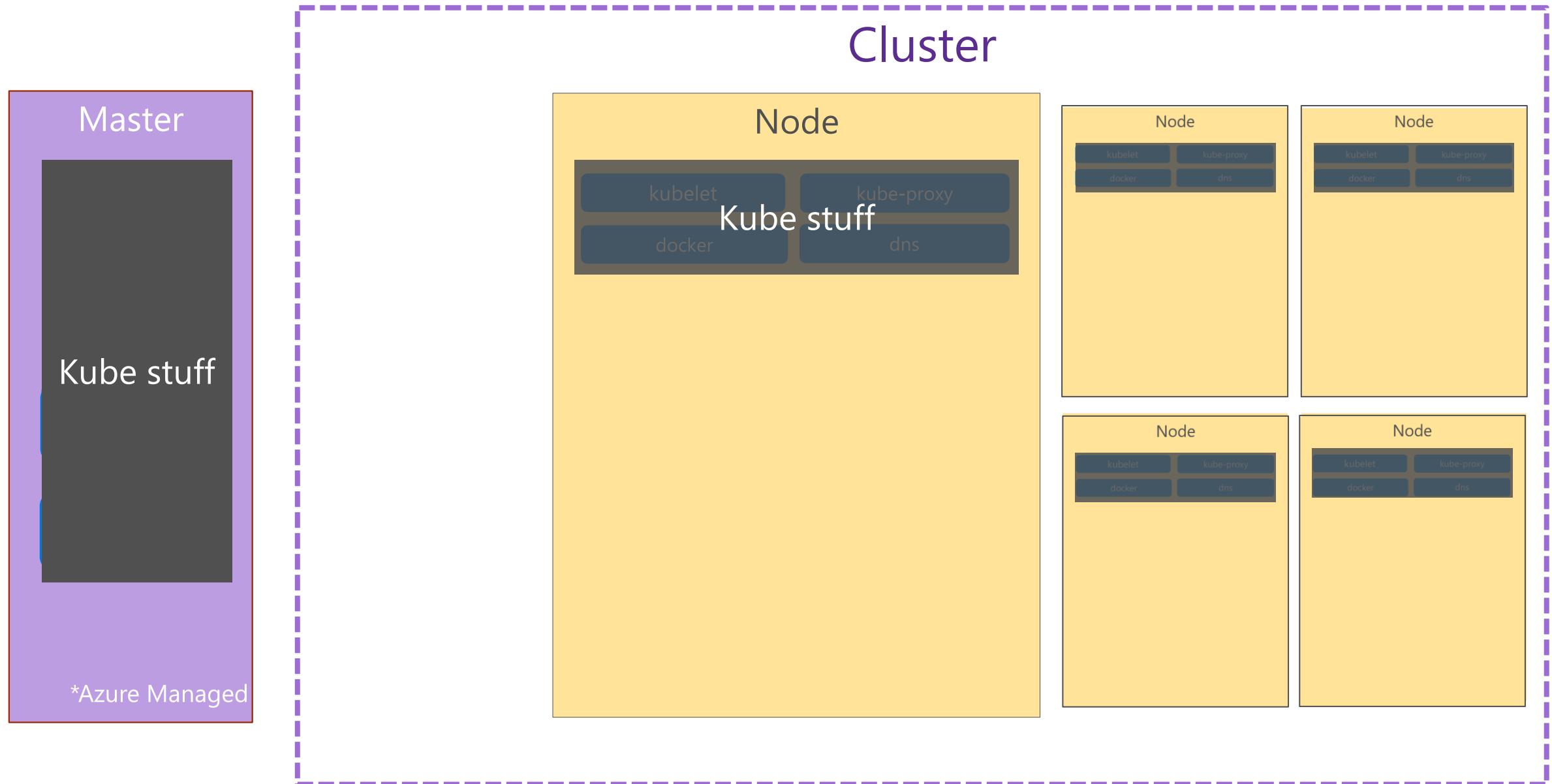
Kubernetes @ 10,000ft



Kubernetes Architecture Components

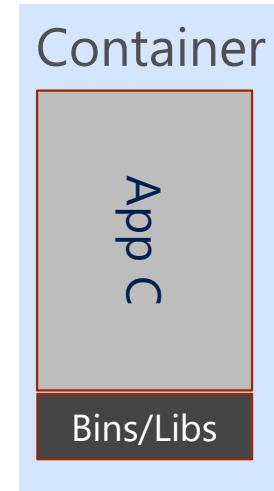
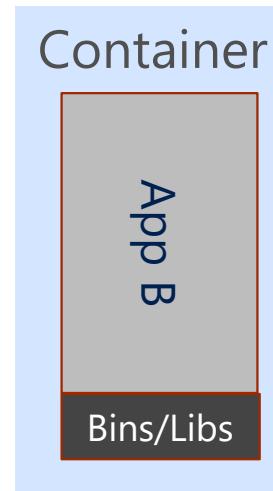
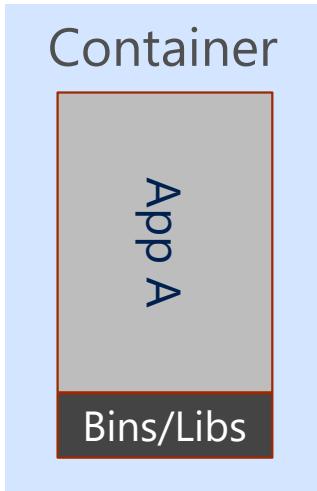


Kubernetes Architecture Components



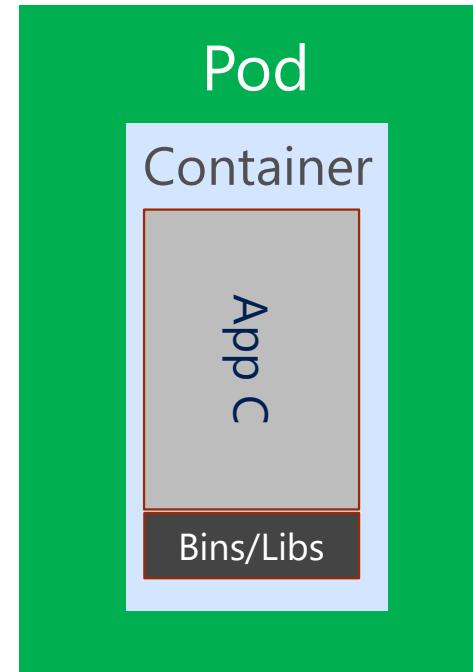
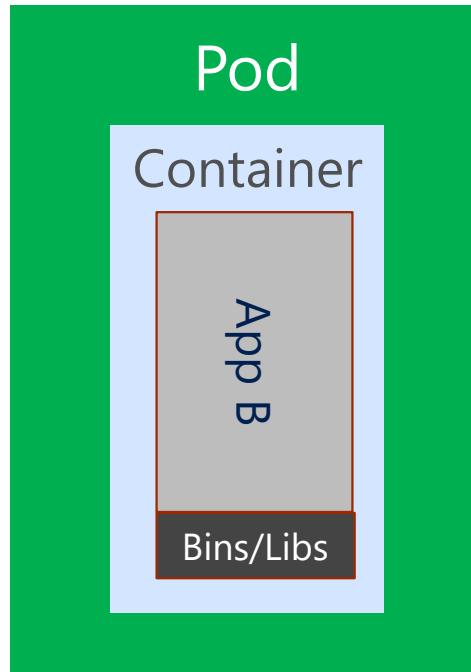
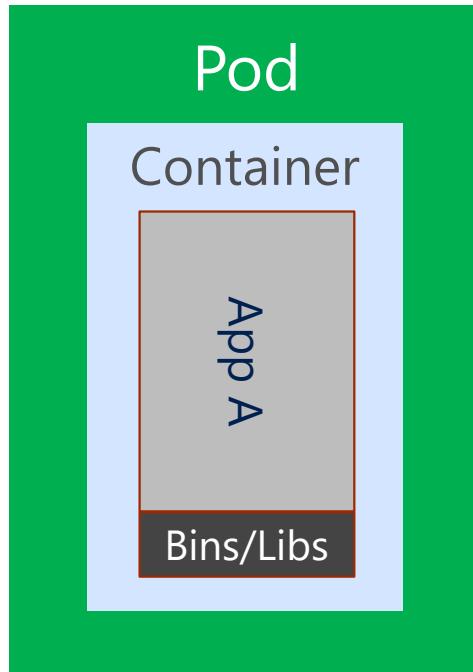
Kubernetes Architecture Components

Where do the Containers go?



Kubernetes Architecture Components

Introducing.... Pods!



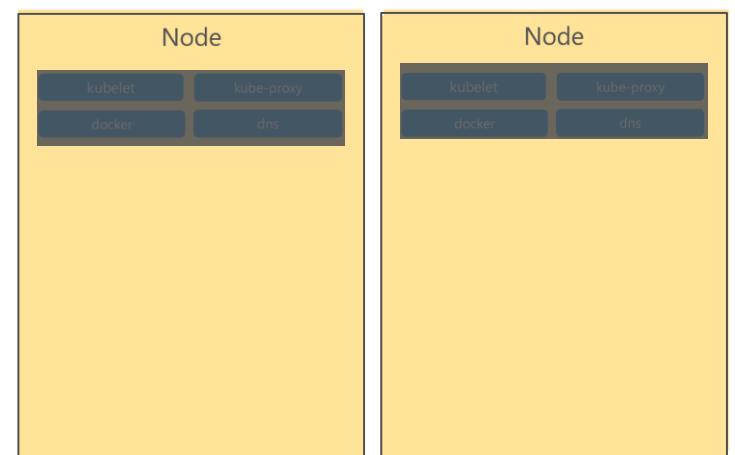
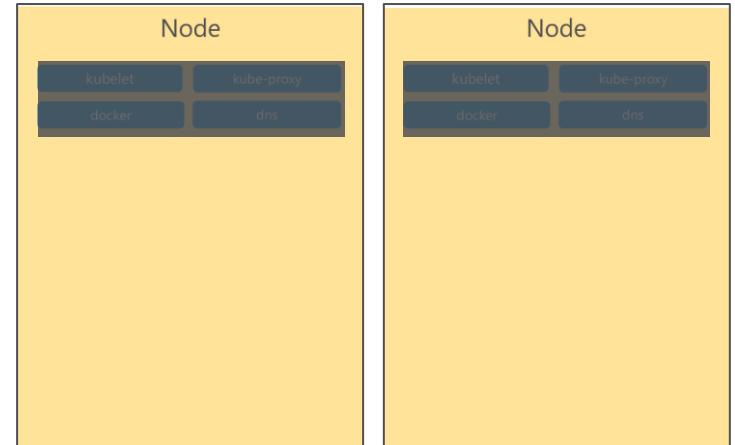
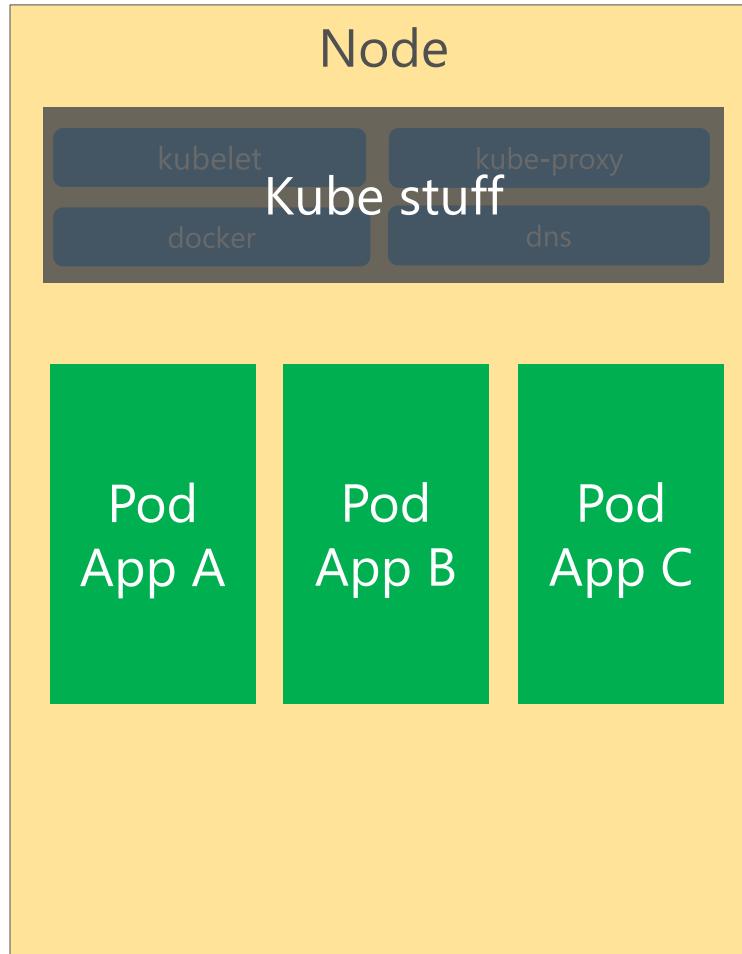
Containers go inside Pods!

Kubernetes Architecture Components

Pods



Cluster

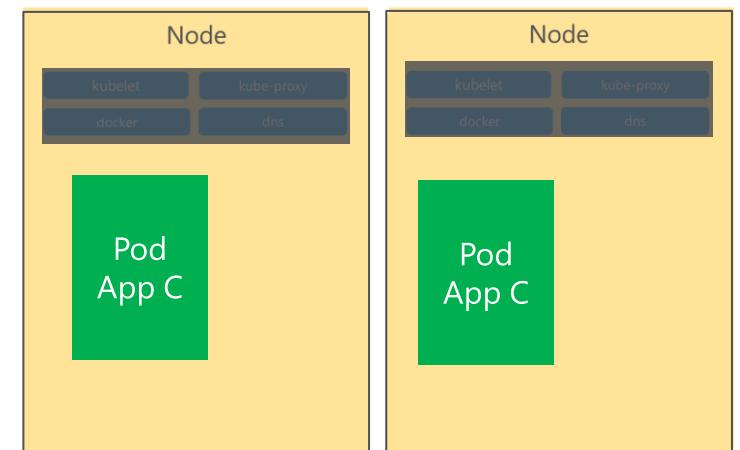
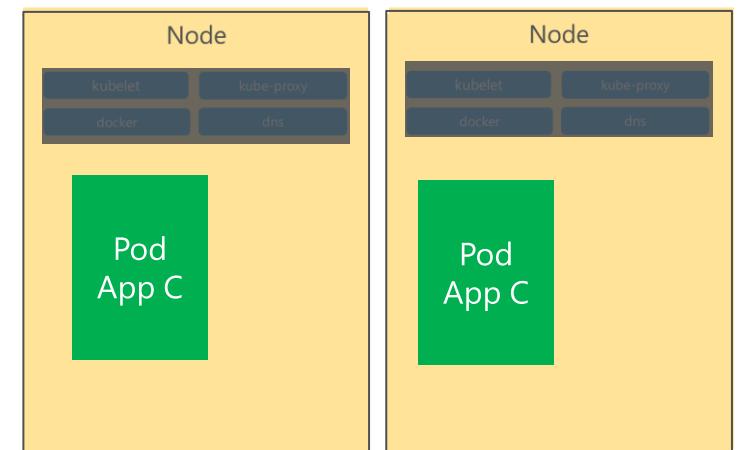
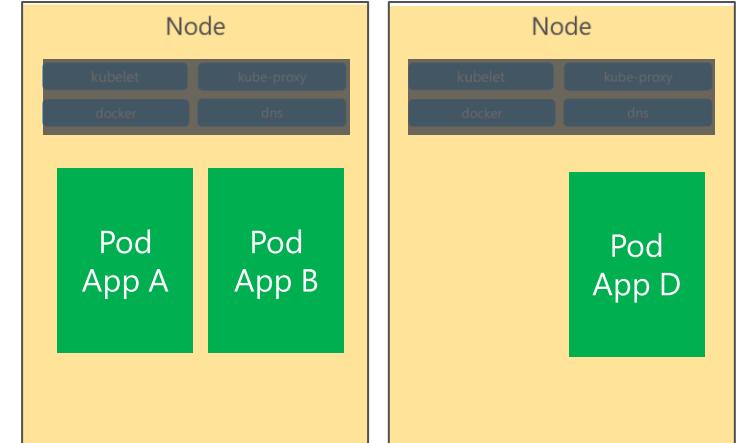
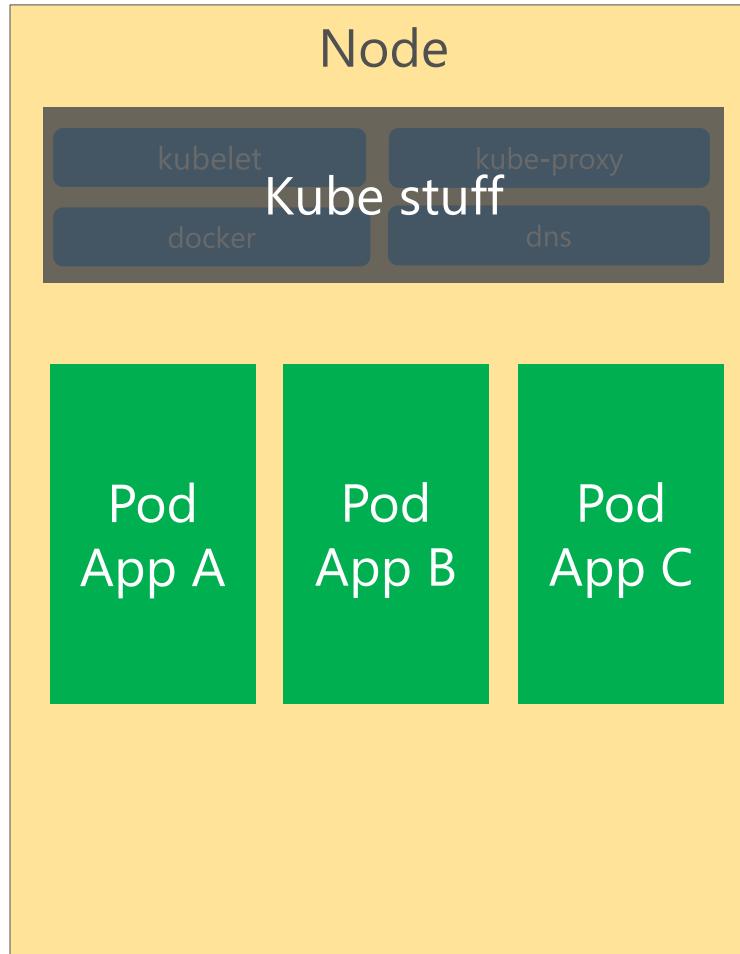


Kubernetes Architecture Components

Pods

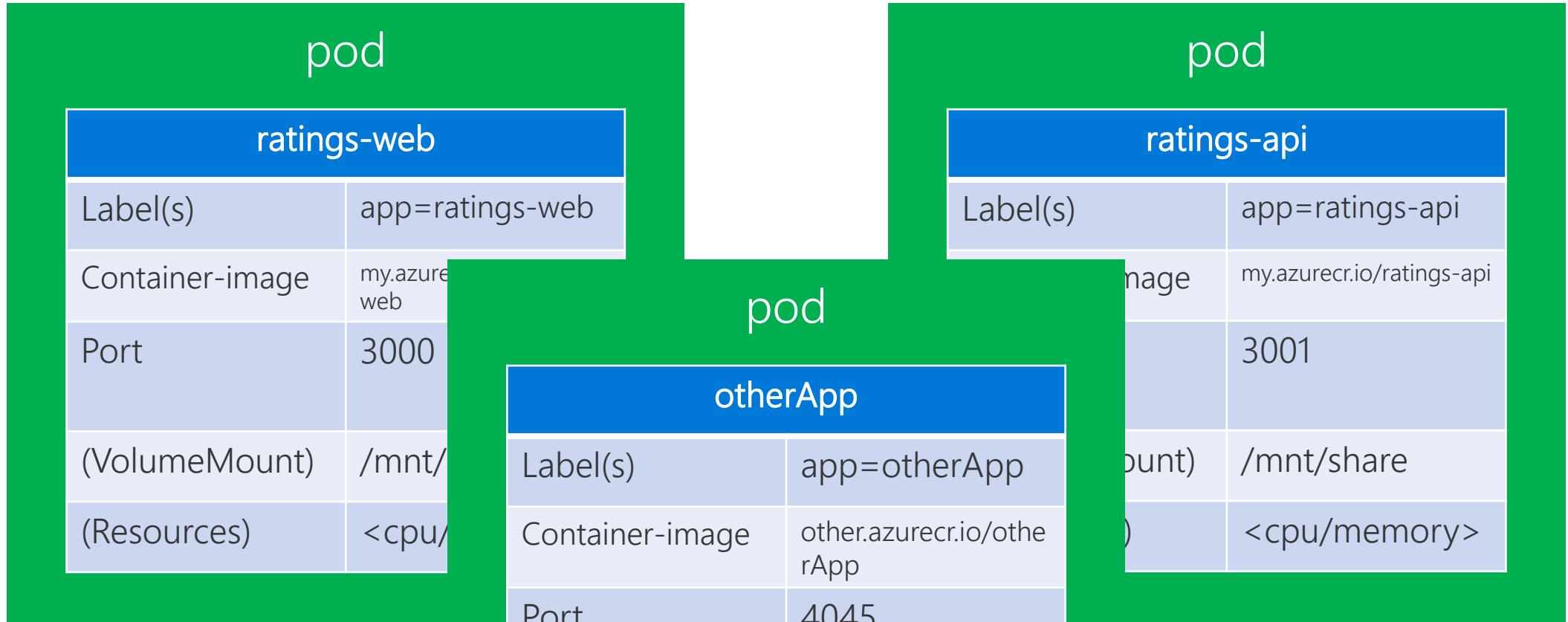


Cluster



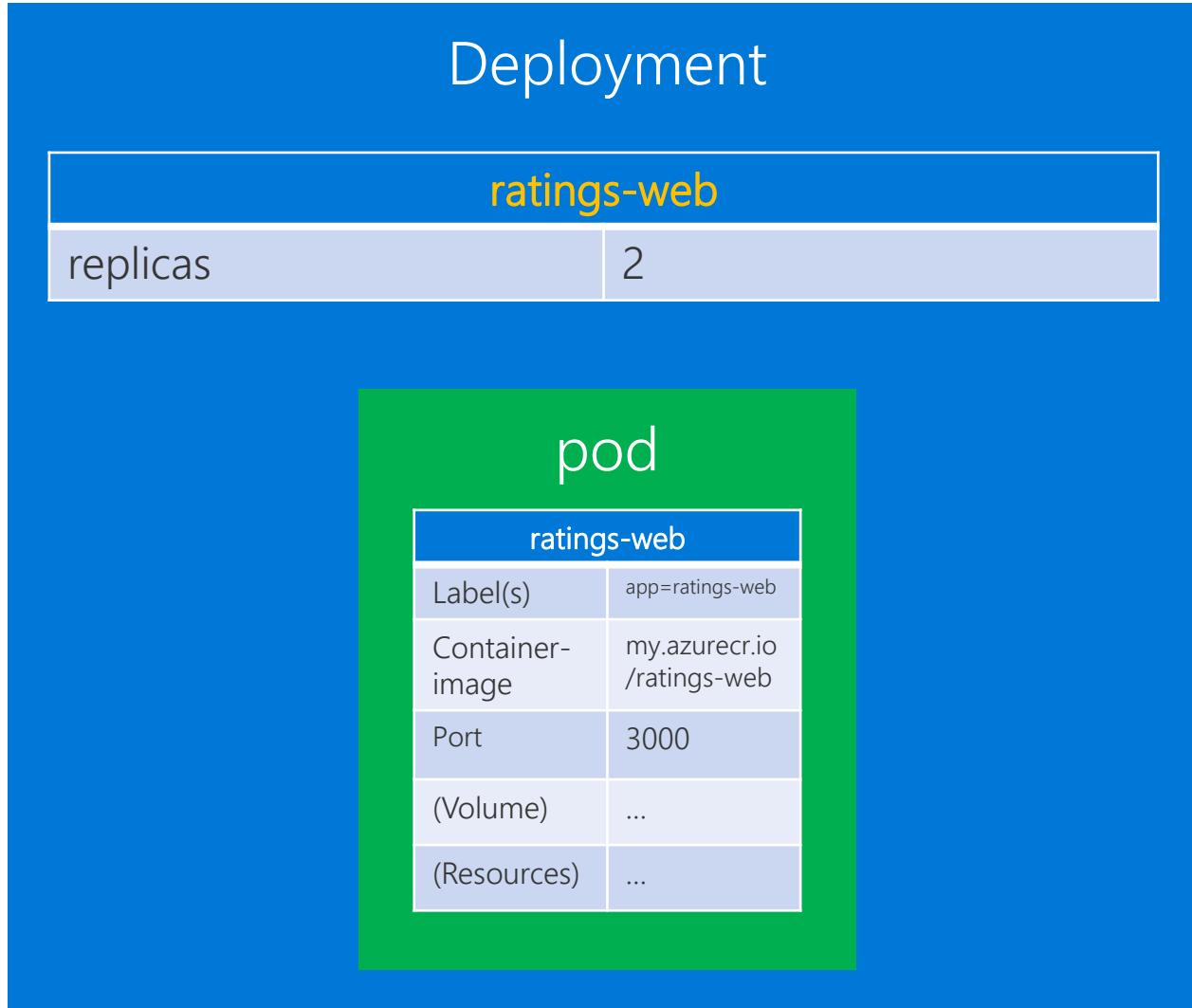
Pods – What are they?

Metadata that describes the pod! containers are configured.

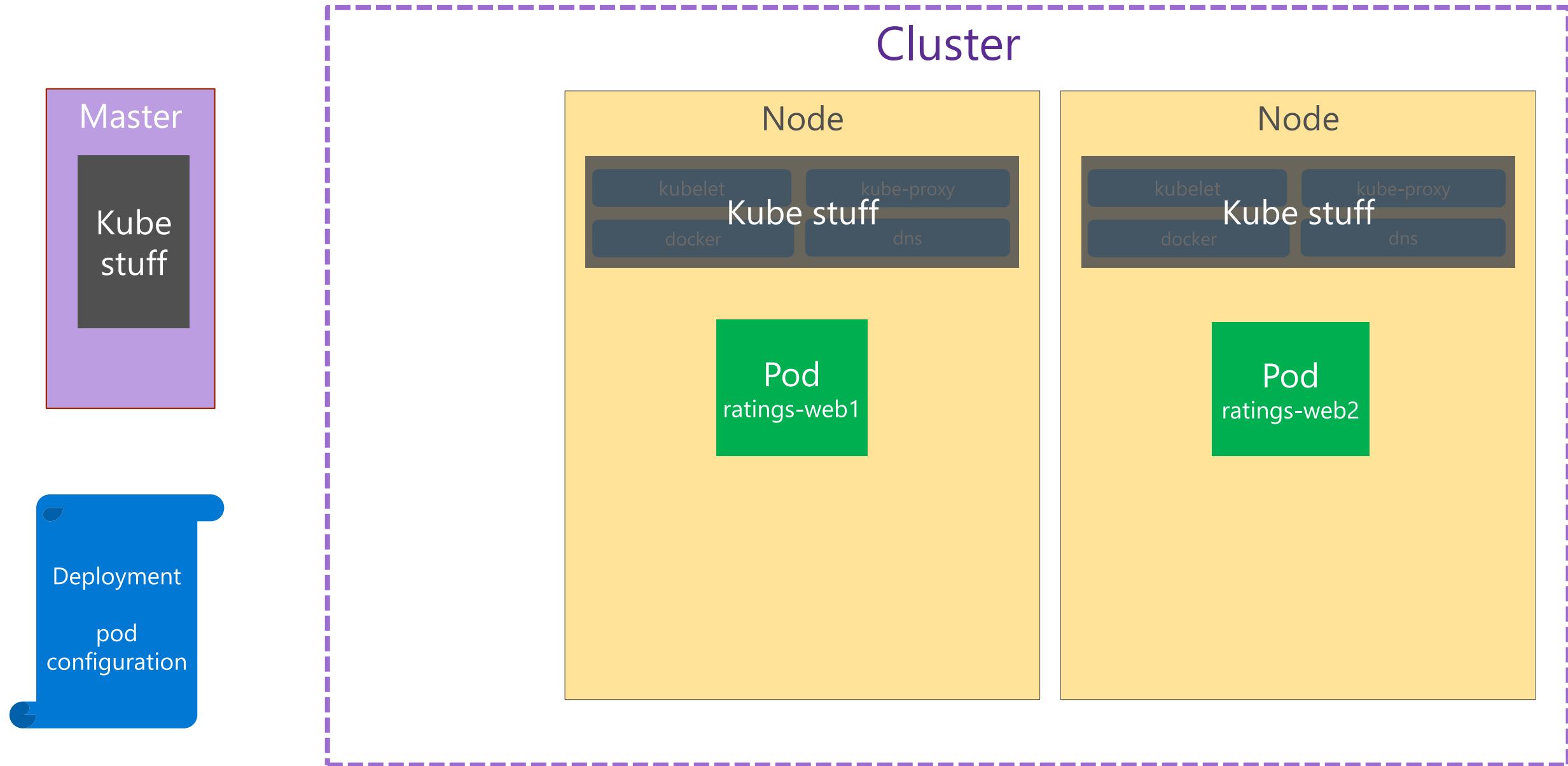


Deployments

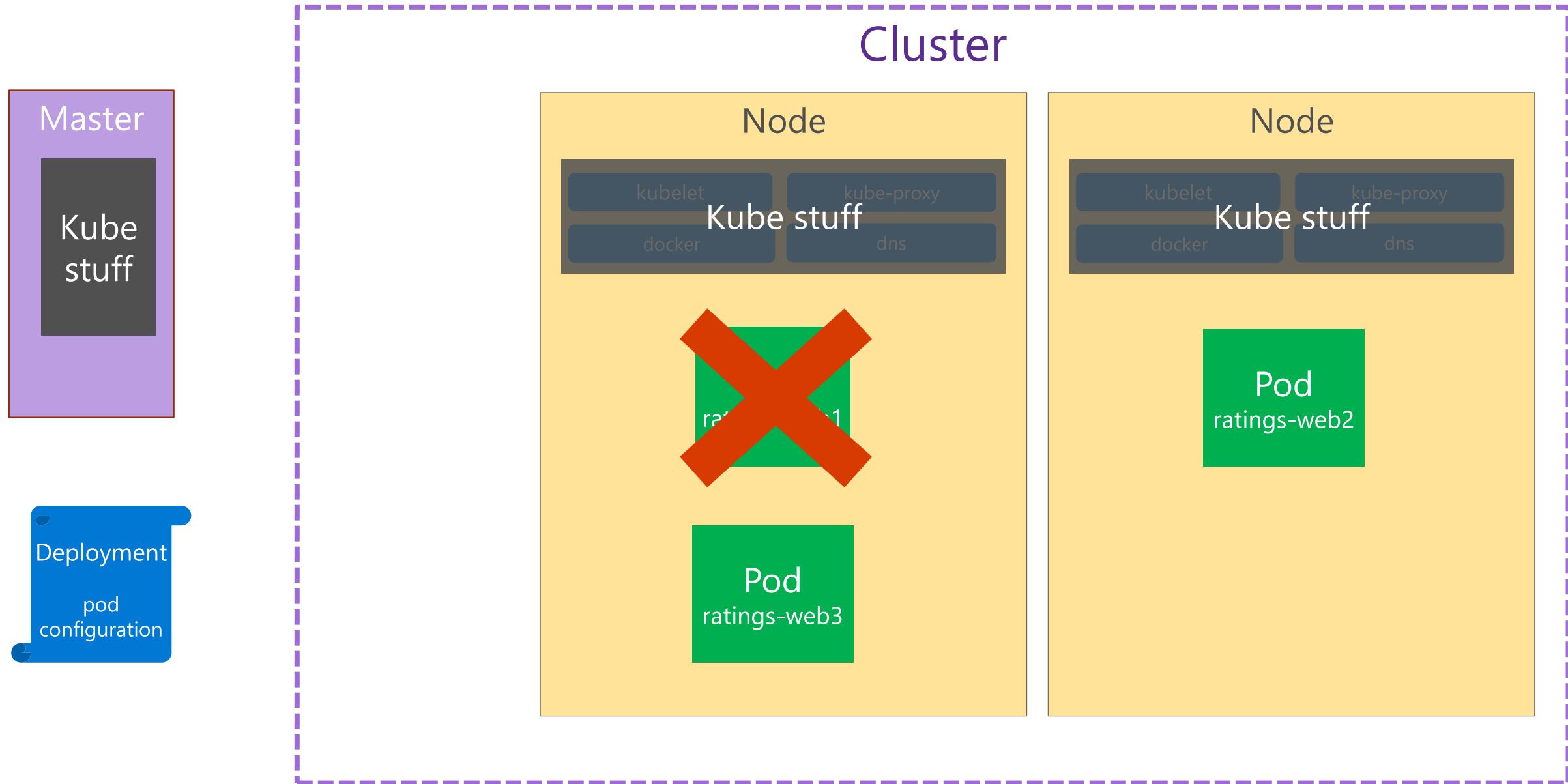
Metadata describes how to deploy a pod



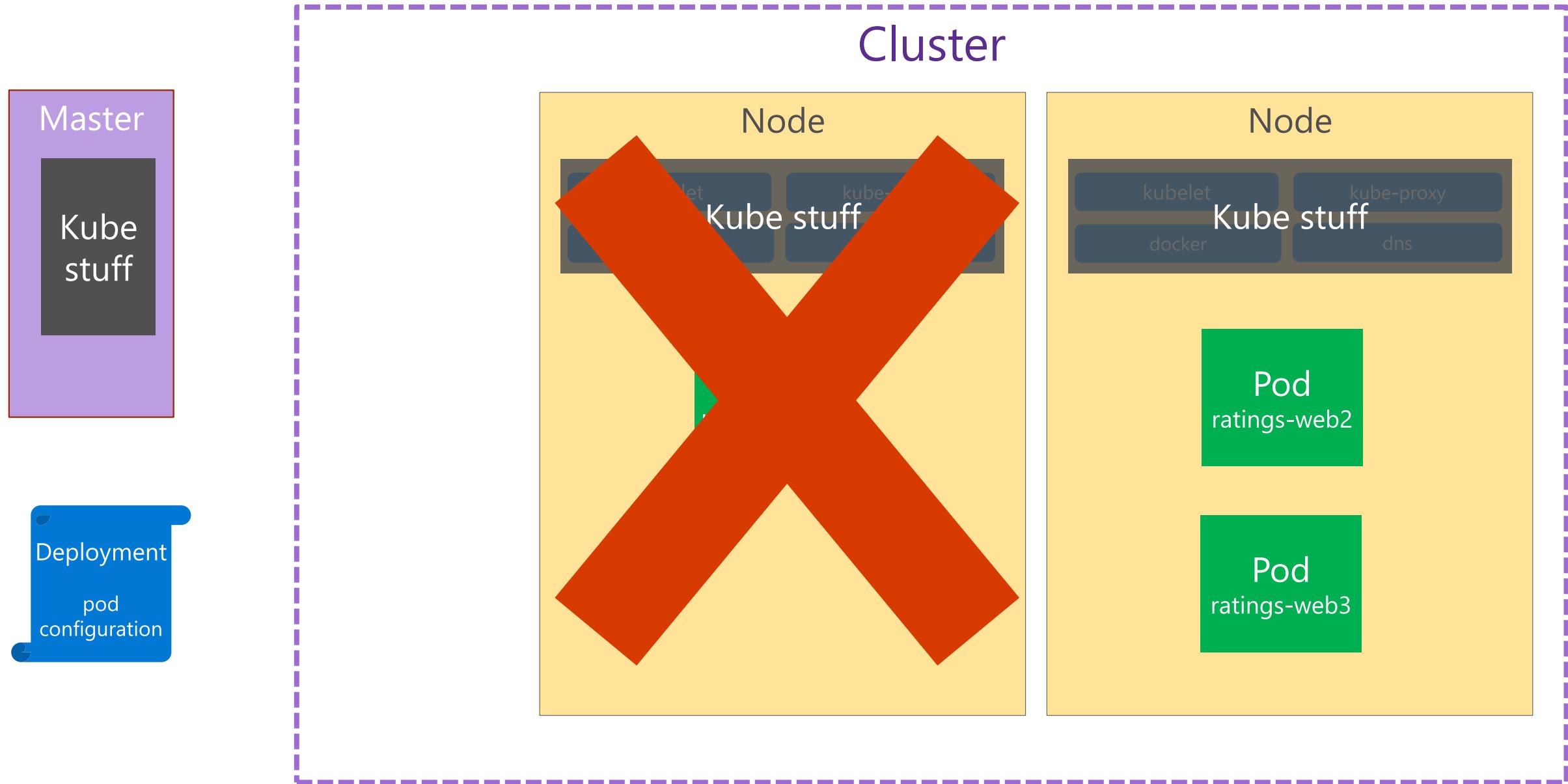
Kubernetes Deployment In Action



Kubernetes Desired State In Action – Example 1

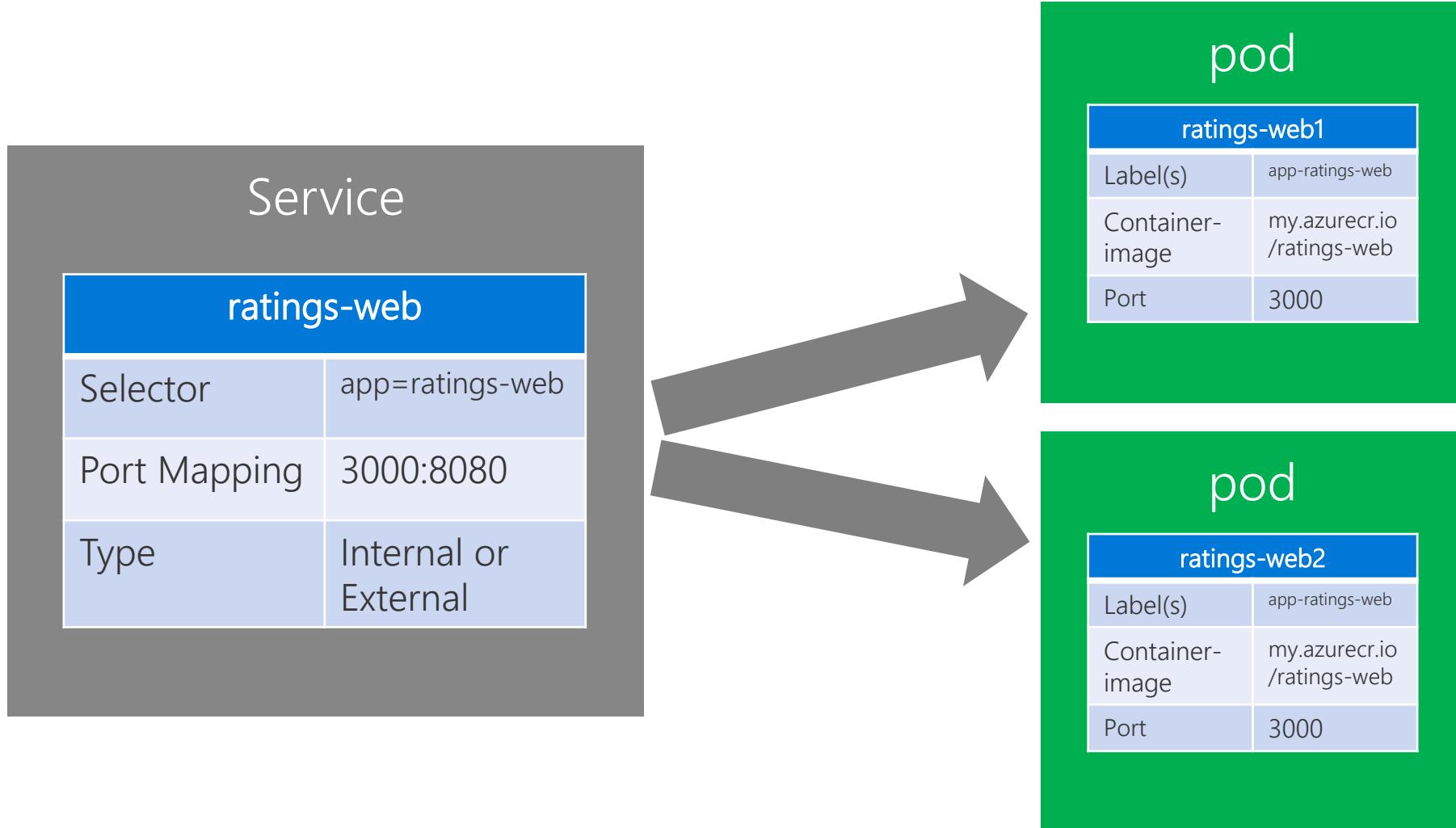


Kubernetes Desired State In Action – Example 2



Services

Metadata that describes how to reach the pods



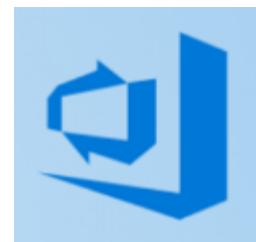
Infrastructure as Code

Three major parts

- Deploying infrastructure
 - All of the stuff outside the VM
 - Vnets, VMs, load balancers, storage, etc
- Configuration Management
 - Configuring all of the stuff inside the VM
 - Security, services, software, etc.
- Orchestration
 - Stringing it all together with automation

ARM
Templates

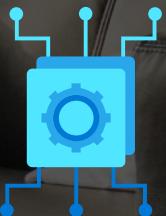
PowerShell
DSC



Thousands of customers build cloud native applications on Microsoft Azure



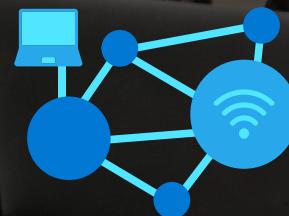
Common cloud native scenarios



Modernize business
critical applications



SaaS
delivery



Real-time
telemetry

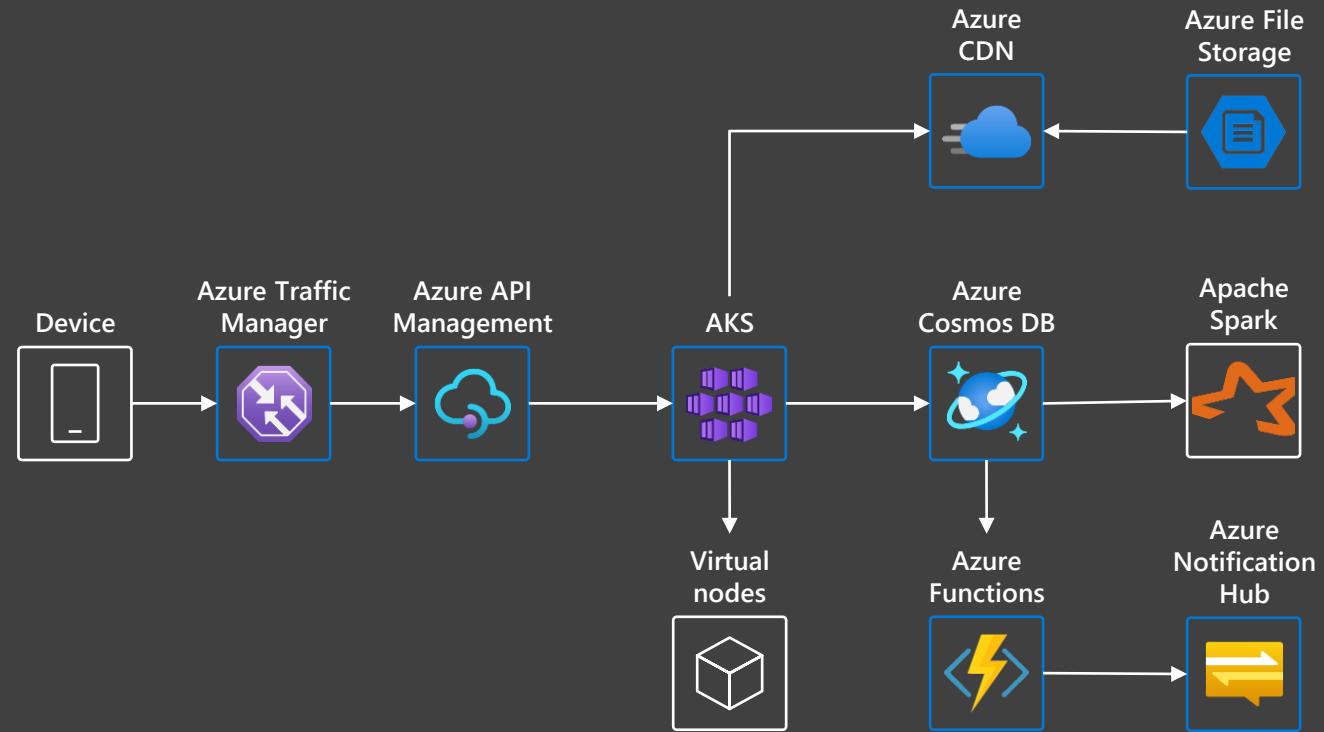


Geo-distributed
applications

Handle spikes in demand with ease

Offer customers fast and reliable service quality during seasonal and other high-traffic demand periods

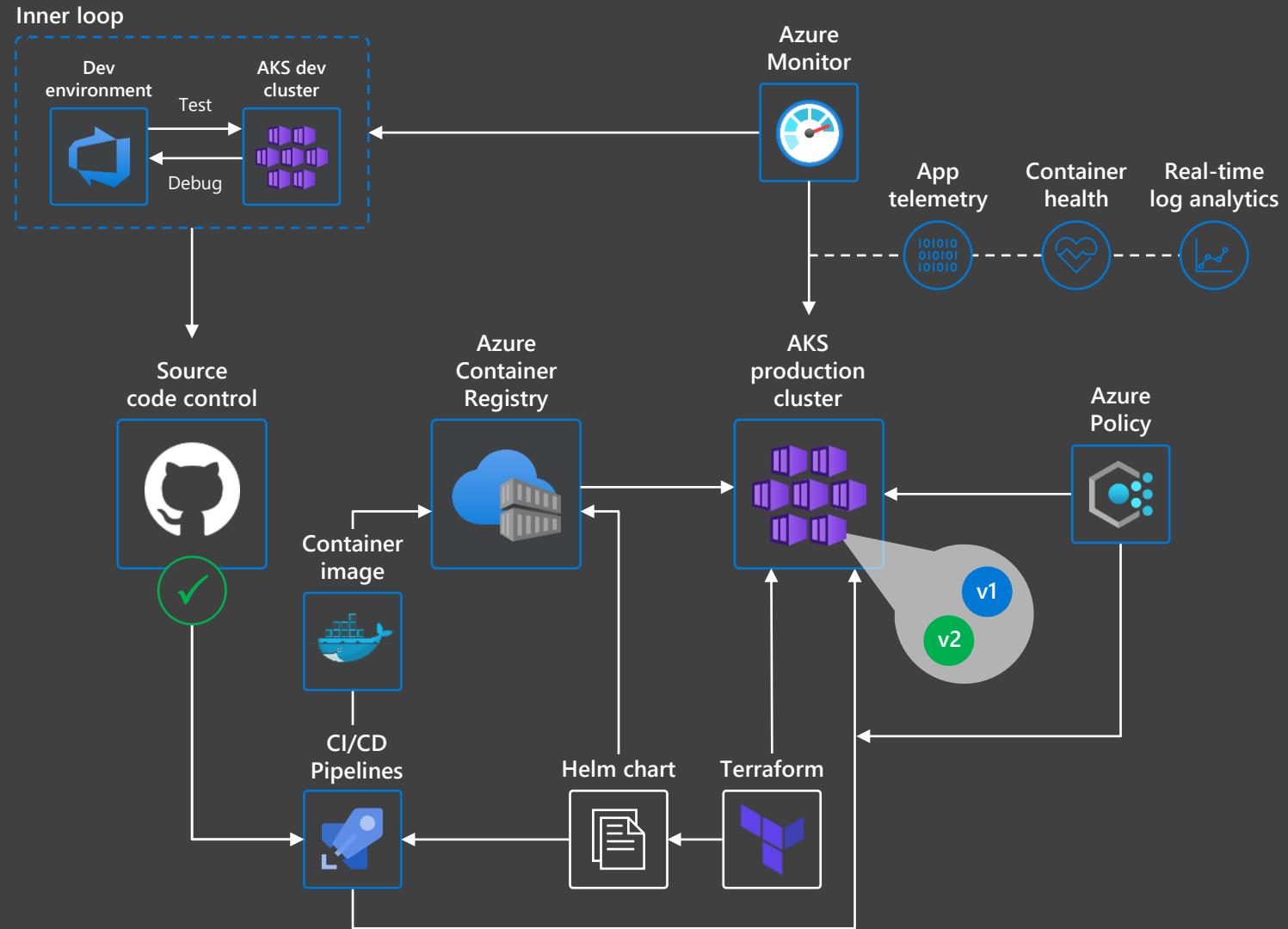
- Instant, elastic scaling handles traffic and sales bursts without managing infrastructures
- Low-latency data access from anywhere in the world for fast, robust user experiences
- High availability across multiple data centers



Continuous delivery of new features without downtime

Unlock team agility with streamlined DevOps without compromising security

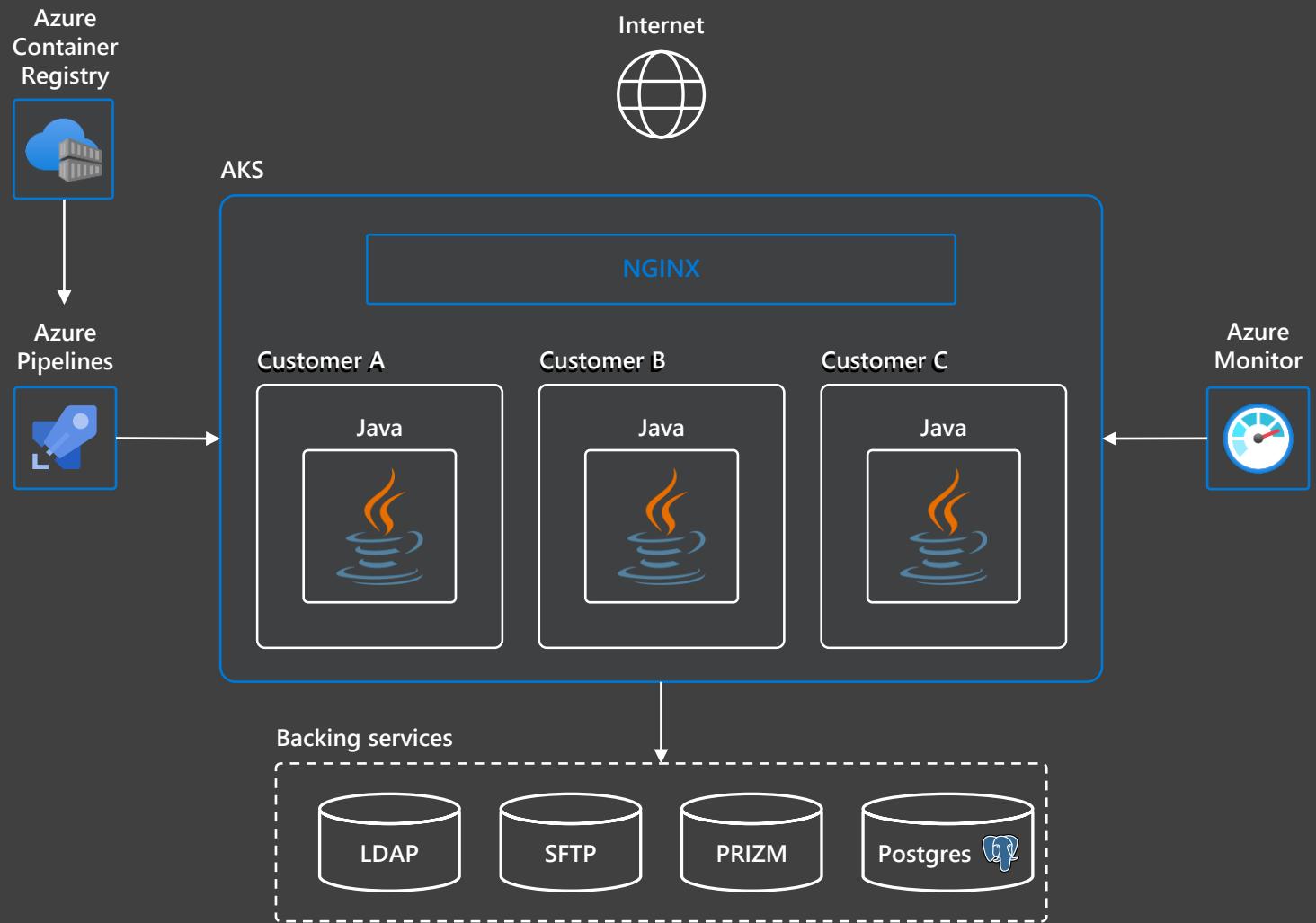
- Deliver code faster with managed Kubernetes and built-in CI/CD
- Accelerate the feedback loop with real-time monitoring
- Balance speed and security with continuous security and deep traceability



Faster customer onboarding

Faster customer environment provisioning by changing from server-based to container- and managed service-based architecture in the cloud.

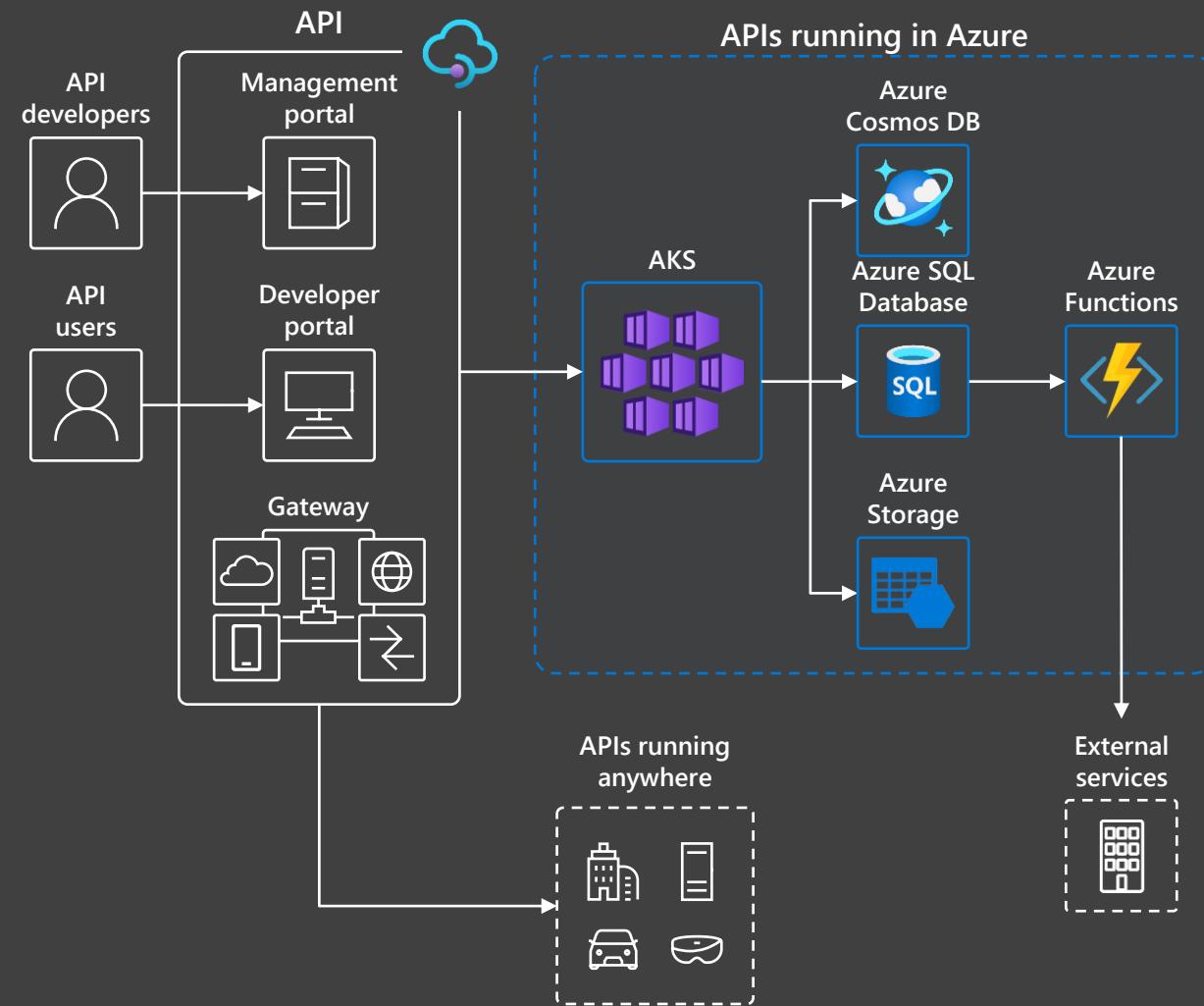
- Managed container platform with built-in CI/CD pipelines to standardize the creation, update, monitoring, and scaling of prospective customer environment.
- Reduce administrative overhead with auto patching, auto upgrade, and auto backup in case of outage.
- Highly secure data integration with intelligent threat protection .



Evolving to API-first SaaS business model

Adapt, evolve, and allow faster innovation to turn opportunities into strategic advantages

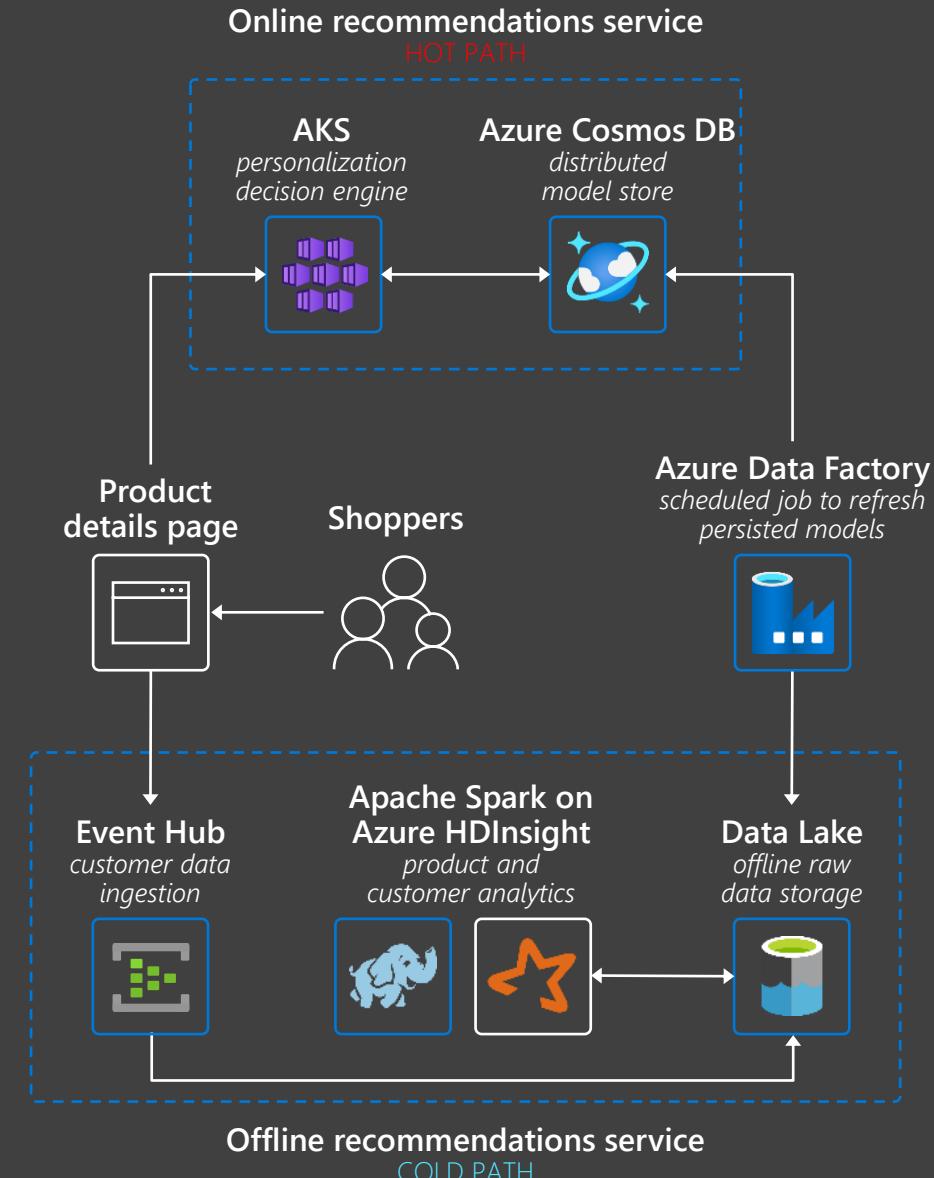
- Create API gateway and developer portal in minutes and publish APIs easily for internal or external use
- Handle any data schema and adapt quickly to rapid changes
- Connect to back-end services anywhere and manage, secure, and optimize all APIs in one place



Real-time personalized recommendations

Help customers discover items they'll love with real-time personalization and product recommendations

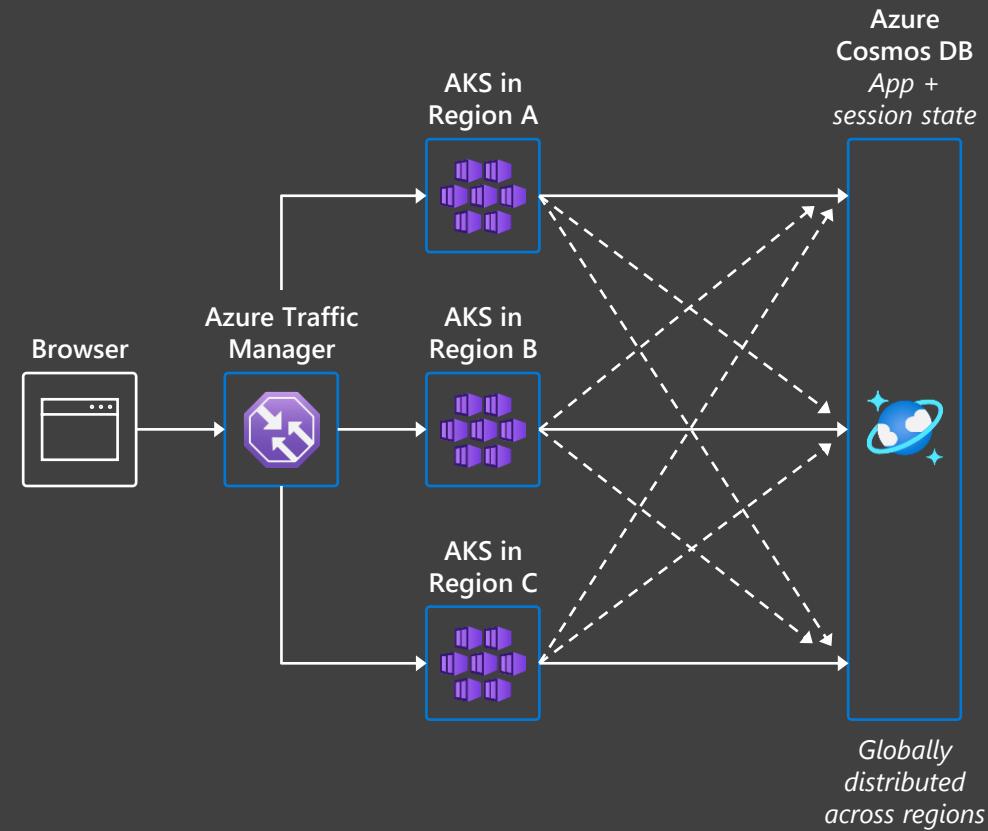
- High volumes of product data can be analyzed in milliseconds
- Low-latency ensures high app performance worldwide
- Tunable data consistency models for rapid insight



Geo-distributed apps

Put your data where your users are to give real-time access and uninterrupted service to customers anywhere in the world

- Compute and data services in more regions than any cloud provider
- High availability and turnkey global data replication across all Azure regions
- Guaranteed low-latency experience for global users



LABS



Azure Immersion Workshop: Cloud Native Apps | 2020-11-25 | Norway

By: Microsoft

Overview

Lab

⌚ 3 hour(s) and 0 minute(s)

🚀 LAUNCH LAB

🏷️ Cloud-Native,Azure

Cloud-native applications
2 hour(s), 51 minute(s) remaining

Lab Guide Lab Environment Lab Validation Help

Environment Details Lab Resources

Azure Credentials

Here are your credentials to login to <https://portal.azure.com> and access the On Demand Lab

Username: odl_user_257593@mazurelabs.onmicrosoft.com [Copy](#)

Password: gpro17HNH*Aq [Copy](#)

Service Principal Details

Application Id: 68a3ffaa-4016-44d7-ba23-301f4772a490 [Copy](#)

Display Name: https://odl_user_sp_257593 [Copy](#)

Secret Key: equp44HQSG*SG [Copy](#)

Subscription Id: c0cf37a-a3de-43d0-a2aa-f375949abb81 [Copy](#)

Tenant Id: cefcb8e7-ee30-49b8-b190-133f1daafdf85 [Copy](#)

Tenant Domain Name: mazurelabs.onmicrosoft.com [Copy](#)

Environment Details

Resource Group : fabmedical-257593

DeploymentID: 257593 [Copy](#)

LabVM DNS Name: fabmedicalqpeduzq3rd6ak.eastus.cloudapp.azure.com [Copy](#)

LabVM Username: adminfabmedical [Copy](#)

LabVM Password: Password.1!! [Copy](#)

Build Agent VM Dns Name: fabmedicalqpeduzq3rd6ak.eastus.cloudapp.azure.com [Copy](#)

Build Agent VM Username: adminfabmedical [Copy](#)

Build Agent VM Password: - [Copy](#)

Powered by [CloudLabs](#)

1. Cloud Native Applications - developer edition



Exercise 1: Create and run a Docker application

Validation Step	Status	Actions
Task 9: Setup CI Pipeline to Push Images	NOT VALIDATED	

Exercise 2: Deploying solution to Azure Kubernetes Service

Validation Step	Status	Actions
Task 3: Deploy a service using kubectl	NOT VALIDATED	
Task 5: Initialize database with a Kubernetes Job	NOT VALIDATED	
Task 6: Test the application in a browser	NOT VALIDATED	

Exercise 3: Scale the application and test HA

Validation Step	Status	Actions
Task 3: Restart containers and test HA	NOT VALIDATED	

Exercise 4: Working with services and routing application traffic

Validation Step	Status	Actions
Task 4: Perform a rolling update	NOT VALIDATED	
Task 5: Configure Kubernetes Ingress	NOT VALIDATED	

fabmedical-257593

Resource group

Search (Ctrl+ /) < + Add Edit columns Delete resource group Refresh Export to CSV Open query Assign tags Move Delete Export template

Overview Essentials

Subscription (change) : Azure Labs H-04 Deployments : 1 Succeeded
Subscription ID : c0cf37a-a3de-43d0-a2aa-f375949abb81
Tags (change) : DeploymentId : 257593 TemplateId : 1824 LaunchId : 10001 LaunchType : ON_DEMAND_LAB TenantId : 313

Filter by name... Type == all × Location == all × Add filter

Showing 1 to 18 of 18 records. Show hidden types ⓘ No grouping

<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/> acr257593	Container registry	East US
<input type="checkbox"/> ContainerInsights(fabmedical-257593)	Solution	East US
<input type="checkbox"/> content-api	Application Insights	East US
<input type="checkbox"/> content-web	Application Insights	East US
<input type="checkbox"/> fabmedical	Virtual machine	East US
<input type="checkbox"/> fabmedical-257593	Kubernetes service	East US
<input type="checkbox"/> fabmedical-257593	Azure Cosmos DB account	East US
<input type="checkbox"/> fabmedical-257593	Network interface	East US
<input type="checkbox"/> fabmedical-257593	Log Analytics workspace	East US
<input type="checkbox"/> fabmedical-ip	Public IP address	East US
<input type="checkbox"/> fabmedical-nsg	Network security group	East US
<input type="checkbox"/> fabmedical-vnet	Virtual network	East US
<input type="checkbox"/> fabmedical_OsDisk_1_406f3f912a8143359002ce3e21a2d781	Disk	East US
<input type="checkbox"/> labvm-257593	Virtual machine	East US
<input type="checkbox"/> labvm-257593_OsDisk_1_242097dcf1924e8eb019eb2ff63c4721	Disk	East US

Cloud-native applications
0 hour(s), 7 minute(s) remaining

Lab Guide Lab Environment Lab Validation Help

Exercise 4: Working with services and routing application traffic

Duration: 45 minutes

In the previous exercise, we introduced a restriction to the scale properties of the service. In this exercise, you will configure the API deployments to create pods that use dynamic port mappings to eliminate the port resource constraint during scale activities.

Kubernetes services can discover the ports assigned to each pod, allowing you to run multiple instances of the pod on the same agent node — something that is not possible when you configure a specific static port (such as 3001 for the API service).

Task 1: Scale a service without port constraints

In this task, we will reconfigure the API deployment so that it will produce pods that choose a dynamic hostPort for improved scalability.

1. From the navigation menu select **Deployments** under **Workloads**. From the view's Deployments list, select the API deployment.
2. Select **Edit**.
3. From the **Edit a Deployment** dialog, do the following:
 - o Scroll to the first spec node that describes replicas as shown in the screenshot. Set the value for replicas to **4**.
 - o Within the replicas spec, beneath the template node, find the **api** containers spec. Remove the hostPort entry for the API container's port mapping. The screenshot below shows the desired configuration after editing.

```
19: "spec": {  
20:   "replicas": 4,  
21:   "selector": {  
22:     "matchLabels": {  
23:       "k8s-app": "api"  
24:     }  
25:   },  
26:   "template": {  
27:     "metadata": {  
28:       "name": "api",  
29:       "creationTimestamp": null,  
30:       "labels": {  
31:         "k8s-app": "api"  
32:       }  
33:     },  
34:     "spec": {  
35:       "containers": [  
36:         {  
37:           "name": "api",  
38:           "image": "nginx:1.14.2",  
39:           "ports": [  
40:             {"containerPort": 80, "hostPort": 3001}  
41:           ]  
42:         }  
43:       ]  
44:     }  
45:   }  
46: }
```

ACCESSING THE LABS

Sign-Up Link (will be pasted in the chat): <https://bit.ly/3kHQvmk>

Activation code: **ACTIVATE10001** (expires Thursday, November 26, 2020)

THANK YOU FOR ATTENDING

```
PS C:\> Get-ContactInfo

Name      : Jan Egil Ring
E-mail    : jan.egil.ring@crayon.com
Twitter   : @JanEgilRing
Website   : {www.crayon.no, www.powershell.no, www.powershellmagazine.com}

PS C:\> ■
```

**Microsoft's mission is to
empower every person and
every organization on the
planet to achieve more.**

