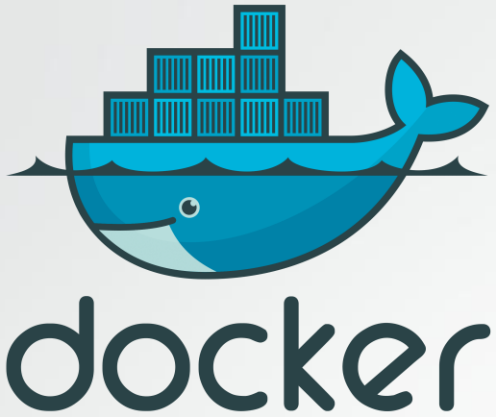


Kubernetes Training

Wolfgang Ofner

kubernetes

About Me

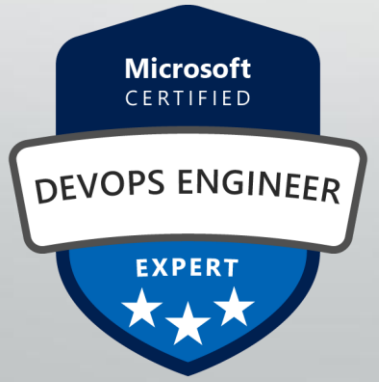
Freelance Cloud Architect, Toronto, Canada

Focus on Azure, Kubernetes, DevOps, and .NET

https://programmingwithwolfgang.com

https://www.linkedin.com/in/wolfgangofner

https://www.youtube.com/@programmingwithwolfgang



Examples and Code Files

Examples and Code Files - GitHub

Agenda

Day 1

- Docker
- Kubernetes Theory
- Kubernetes Exercises

Breaks can be taken at any time outside theory blocks

Questions and conversations are highly encouraged

Please leave your cameras on

Agenda

Day 2

- Kubernetes Theory and Exercises
- Helm
- AKS Security

Breaks can be taken at any time outside theory blocks

Questions and conversations are highly encouraged

Please leave your cameras on

Agenda

Day 3

- Ingress and Gateway API
- Cert-Manager

Agenda

Day 4

- Configure production ready AKS

Challenges of modern Software

Deploy 100 times a day

Versioning

Dependencies

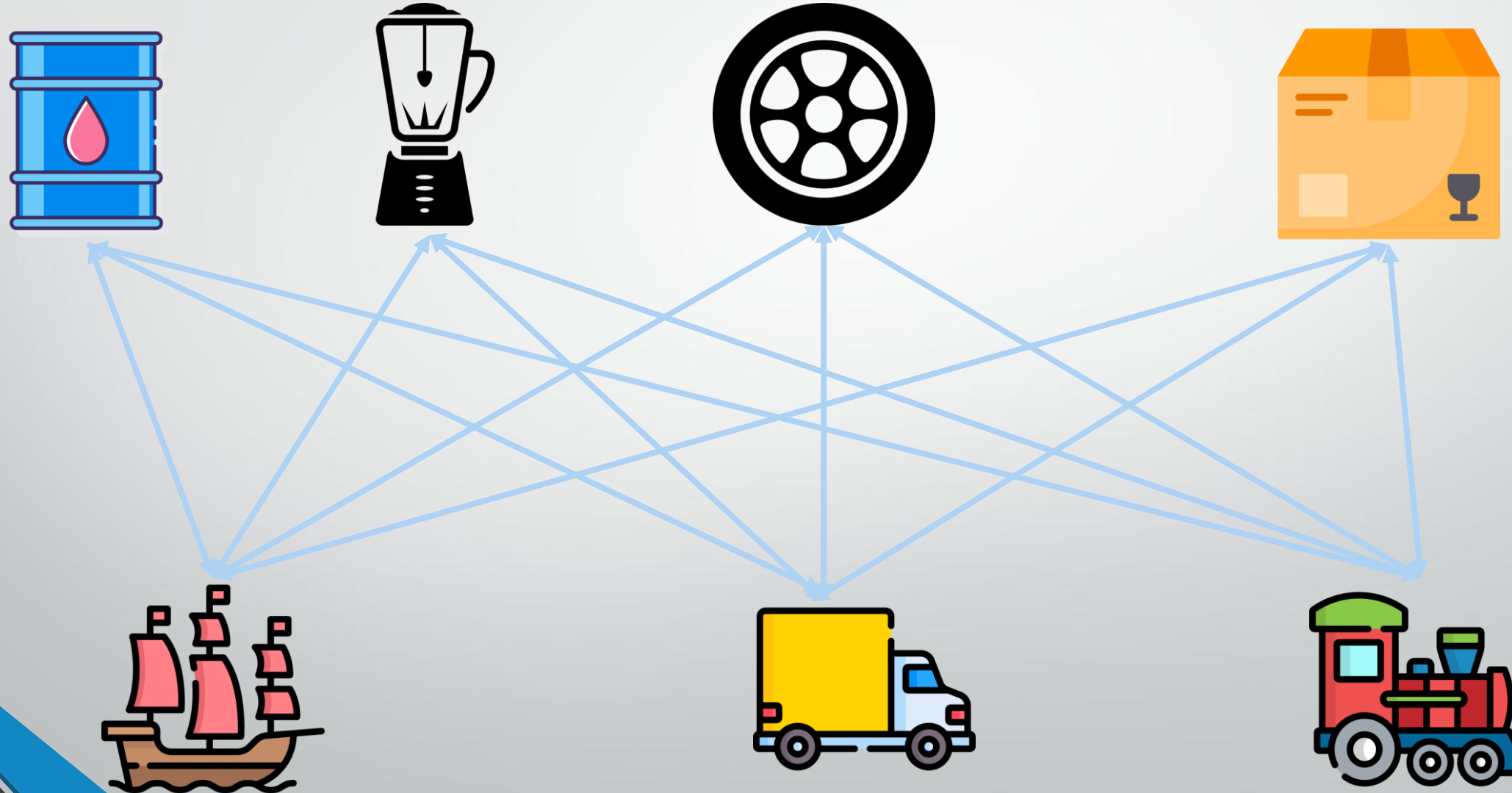
Easy to test

Fast to set up on target machine

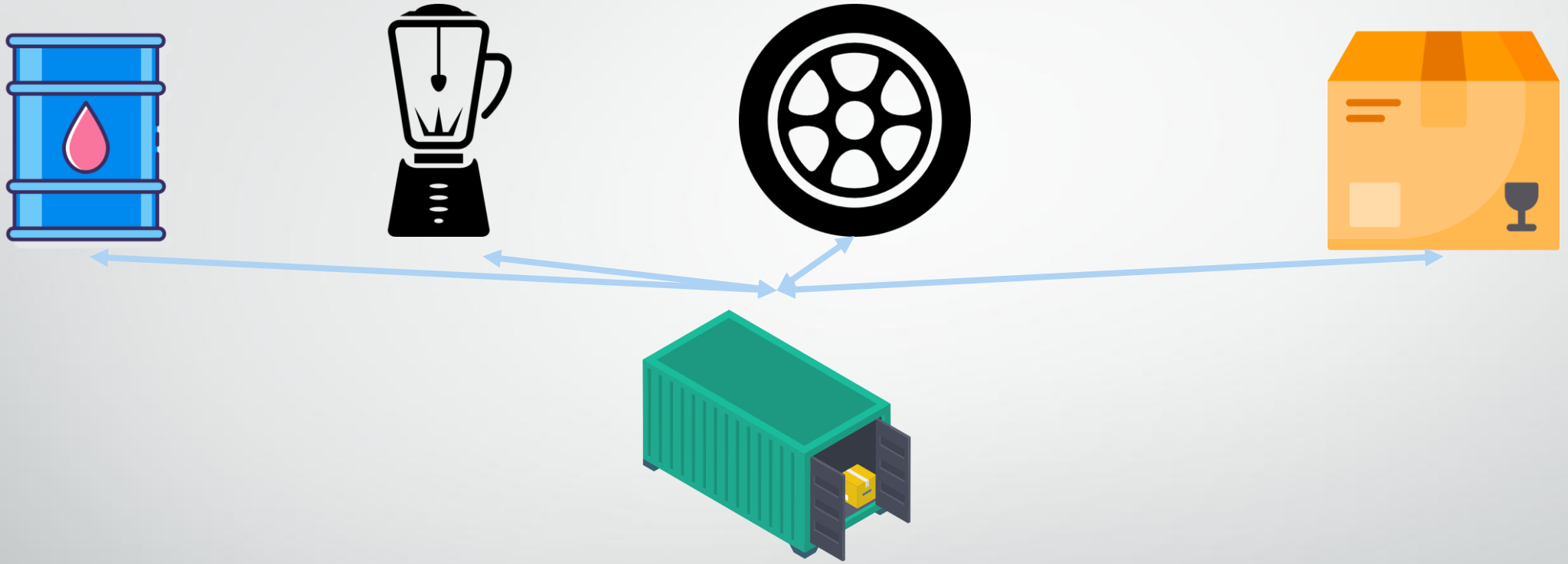
Monolithic software

Database deployments

Freighter transport before 1956



Invention of the the Container (1956)




IT WORKS ON MY MACHINE
THEN WE'LL SHIP YOUR MACHINE
AND THAT IS HOW DOCKER WAS BORN

Container Solutions

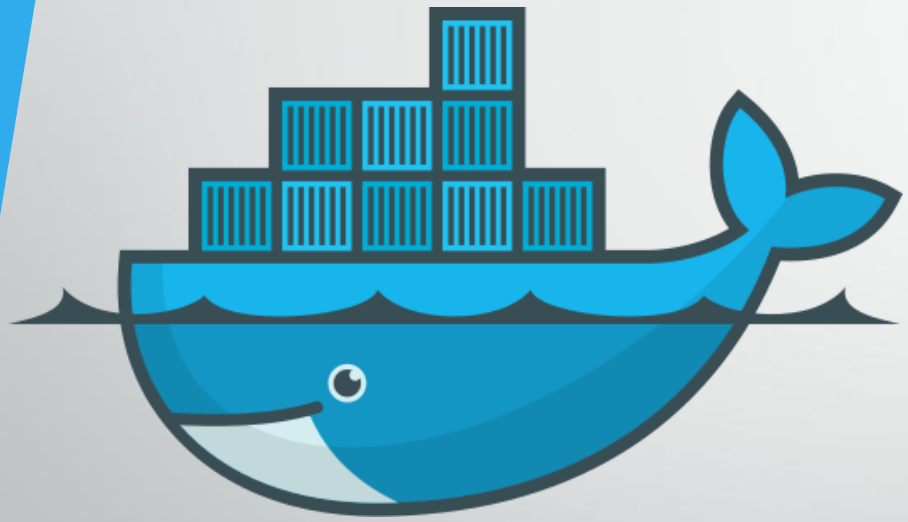
Docker

Podman

Lxc

Crio

Rancher Desktop



docker

Docker Containers

Dockerfile: blueprint

Container: Instance of this blueprint

Versioned artifact

Container image is always bit by bit identical when deployed

Docker Containers

Images for different build platforms, e.g. x86, x64, ARM

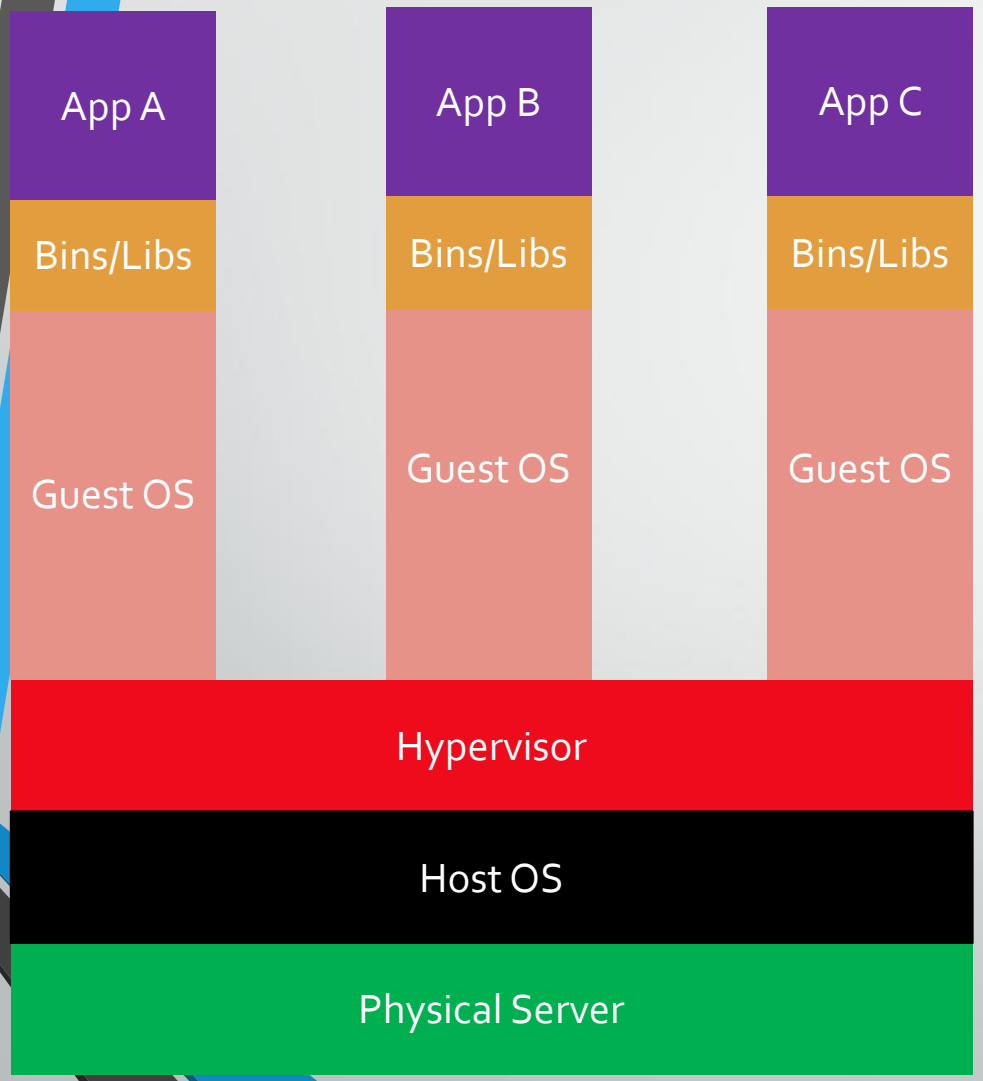
OCI (Open Container Initiative) compliant

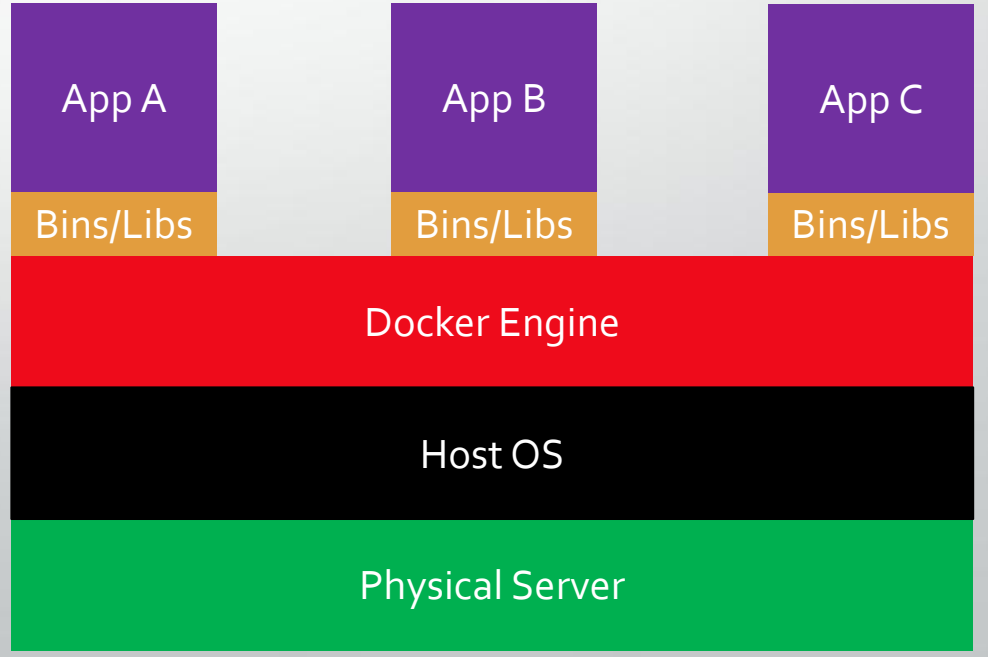
Abstracts underlying infrastructure

Fast start up times

Pet vs. Kettle

Virtual Machine vs. Container





Dockerfile

Blueprint to build Docker Image

Can be based on existing images

Commands to update the base OS and install additional software

Build artifacts to include, such as a developed application

Command to run when the container is launched

Dockerfile

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["CustomerApi/CustomerApi.csproj", "CustomerApi/"]
RUN dotnet restore "CustomerApi/CustomerApi.csproj"
COPY . .
WORKDIR "/src/CustomerApi"
RUN dotnet build "CustomerApi.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "CustomerApi.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "CustomerApi.dll"]
```

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["CustomerApi/CustomerApi.csproj", "CustomerApi/"]
COPY ["CustomerApi.Data/CustomerApi.Data.csproj", "CustomerApi.Data/"]
COPY ["CustomerApi.Domain/CustomerApi.Domain.csproj", "CustomerApi.Domain/"]
COPY ["CustomerApi.Service/CustomerApi.Service.csproj", "CustomerApi.Service/"]
COPY ["CustomerApi.Messaging.Send/CustomerApi.Messaging.Send.csproj", "CustomerApi.Messaging.Send/"]
COPY ["CustomerApi.Database.Build/CustomerApi.Database.Build.csproj", "CustomerApi.Database.Build/"]
COPY ["Tests/CustomerApi.Test/CustomerApi.Test.csproj", "Tests/CustomerApi.Test/"]
COPY ["Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj", "Tests/CustomerApi.Service.Test/"]
COPY ["Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj", "Tests/CustomerApi.Data.Test/"]
COPY ["CustomerApi/nuget.config", ""]
COPY ["*.props", "./"]

ARG PAT=localhost
RUN sed -i "s|</configuration>|<packageSourceCredentials><MicroserviceDemoNugets><add key=\"Username\" value=\"PAT\" /><add key=\"ClearTextPassword\" value=\"${PAT}\" /></MicroserviceDemoNugets></pa

RUN dotnet restore "CustomerApi/CustomerApi.csproj" --configfile "./nuget.config"
RUN dotnet restore "CustomerApi.Database.Build/CustomerApi.Database.Build.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Test/CustomerApi.Test.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj" --configfile "./nuget.config"
COPY . .

RUN dotnet build "CustomerApi/CustomerApi.csproj" -c Release -o /app/build --no-restore
RUN dotnet build "Tests/CustomerApi.Test/CustomerApi.Test.csproj" -c Release --no-restore
RUN dotnet build "Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj" -c Release --no-restore
RUN dotnet build "Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj" -c Release --no-restore

FROM build AS dacpac
ARG BuildId=localhost
LABEL dacpac=${BuildId}
WORKDIR /src
RUN dotnet build "CustomerApi.Database.Build/CustomerApi.Database.Build.csproj" -c Release -o /dacpacs --no-restore

FROM build AS test
ARG BuildId=localhost
LABEL test=${BuildId}
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput=
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results2.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results3.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput

FROM build AS publish
RUN dotnet publish "CustomerApi/CustomerApi.csproj" --no-restore -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "CustomerApi.dll"]
```

```
FROM ubuntu:20.04
RUN DEBIAN_FRONTEND=noninteractive apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get upgrade -y

RUN DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends \
    apt-transport-https \
    apt-utils \
    ca-certificates \
    curl \
    git \
    iputils-ping \
    jq \
    lsb-release \
    software-properties-common \
    wget

RUN curl -sL https://aka.ms/InstallAzureCLIDeb | bash

RUN wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
RUN dpkg -i packages-microsoft-prod.deb
RUN rm packages-microsoft-prod.deb
RUN echo 'deb http://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_20.04/ /' | tee /et
RUN curl -fsSL https://download.opensuse.org/repositories/devel:kubic:libcontainers:stable/xUbuntu_20.04/Release.key |
RUN apt-get update
RUN apt-get install -y dotnet-sdk-6.0
RUN apt-get install -y dotnet-sdk-7.0
RUN apt -y install podman fuse-overlayfs
```

Docker Demo

Onion File System

Every command is a new layer

Layers can be cached

Faster builds

11 Layer

```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["CustomerApi/CustomerApi.csproj", "CustomerApi/"]
RUN dotnet restore "CustomerApi/CustomerApi.csproj"
COPY . .
WORKDIR "/src/CustomerApi"
RUN dotnet build "CustomerApi.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "CustomerApi.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "CustomerApi.dll"]
```

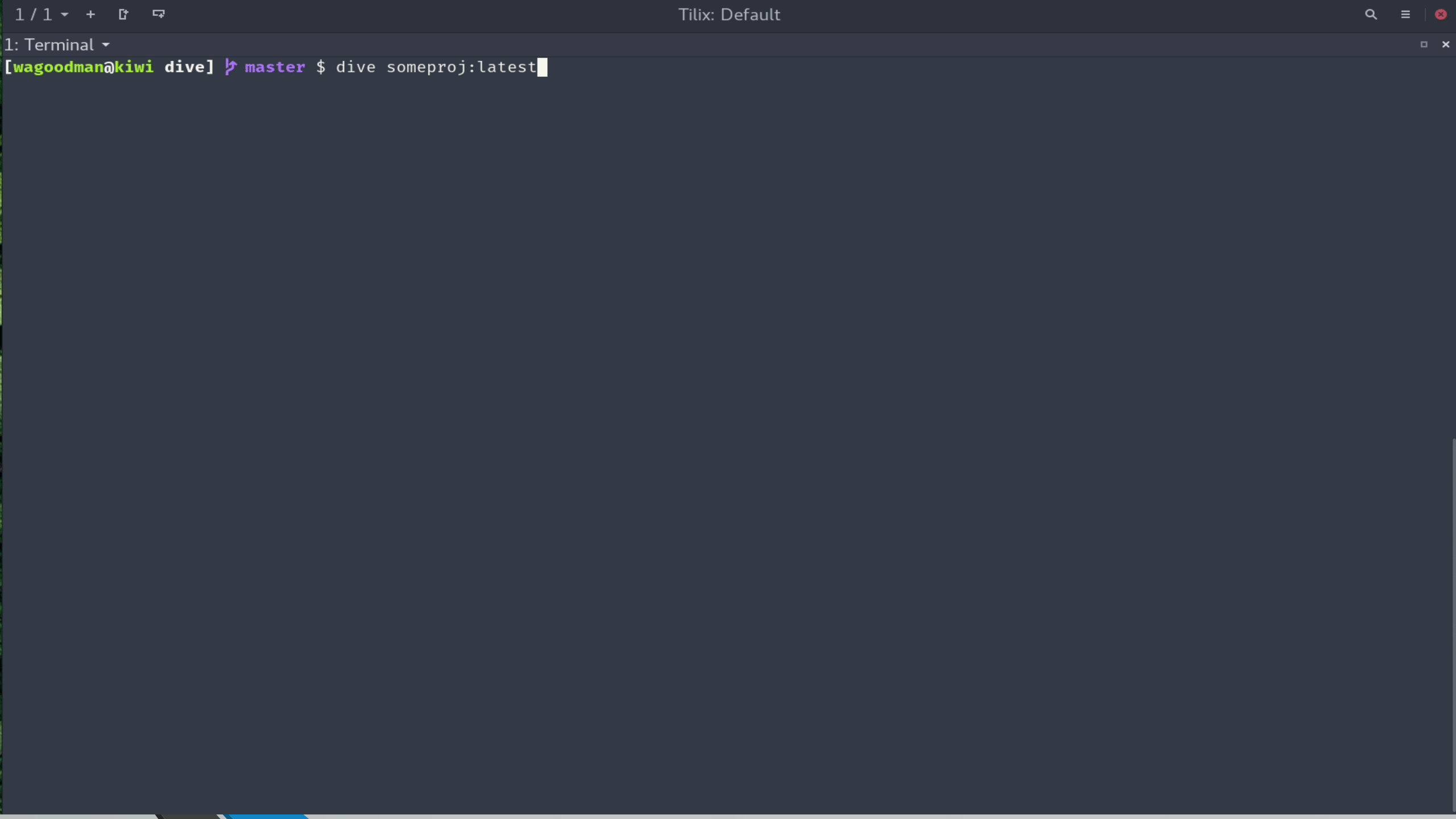
```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["CustomerApi/CustomerApi.csproj", "CustomerApi/"]
COPY ["CustomerApi.Data/CustomerApi.Data.csproj", "CustomerApi.Data/"]
COPY ["CustomerApi.Domain/CustomerApi.Domain.csproj", "CustomerApi.Domain/"]
COPY ["CustomerApi.Service/CustomerApi.Service.csproj", "CustomerApi.Service/"]
COPY ["CustomerApi.Messaging.Send/CustomerApi.Messaging.Send.csproj", "CustomerApi.Messaging.Send/"]
COPY ["CustomerApi.Database.Build/CustomerApi.Database.Build.csproj", "CustomerApi.Database.Build/"]
COPY ["Tests/CustomerApi.Test/CustomerApi.Test.csproj", "Tests/CustomerApi.Test/"]
COPY ["Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj", "Tests/CustomerApi.Service.Test/"]
COPY ["Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj", "Tests/CustomerApi.Data.Test/"]
COPY ["CustomerApi/nuget.config", ""]
COPY ["*.props", "./"]

ARG PAT=localhost
RUN sed -i "s|</configuration>|<packageSourceCredentials><MicroserviceDemoNugets><add key=\"Username\" value=\"PAT\" /><add key=\"ClearTextPassword\" value=\"${PAT}\" /></MicroserviceDemoNugets></pa

RUN dotnet restore "CustomerApi/CustomerApi.csproj" --configfile "./nuget.config"
RUN dotnet restore "CustomerApi.Database.Build/CustomerApi.Database.Build.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Test/CustomerApi.Test.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj" --configfile "./nuget.config"
RUN dotnet restore "Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj" --configfile "./nuget.config"
COPY . .

RUN dotnet build "CustomerApi/CustomerApi.csproj" -c Release -o /app/build --no-restore
RUN dotnet build "Tests/CustomerApi.Test/CustomerApi.Test.csproj" -c Release --no-restore
RUN dotnet build "Tests/CustomerApi.Service.Test/CustomerApi.Service.Test.csproj" -c Release --no-restore
RUN dotnet build "Tests/CustomerApi.Data.Test/CustomerApi.Data.Test.csproj" -c Release --no-restore

FROM build AS dacpac
ARG BuildId=localhost
LABEL dacpac=${BuildId}
WORKDIR /src
RUN dotnet build "CustomerApi.Database.Build/CustomerApi.Database.Build.csproj" -c Release -o /dacpacs --no-restore

FROM build AS test
ARG BuildId=localhost
LABEL test=${BuildId}
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results2.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results3.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutput

FROM build AS publish
RUN dotnet publish "CustomerApi/CustomerApi.csproj" --no-restore -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "CustomerApi.dll"]
```

57 Layer

Inspect Layers

Docker history <IMAGE_ID>

Dive: https://github.com/wagoodman/dive

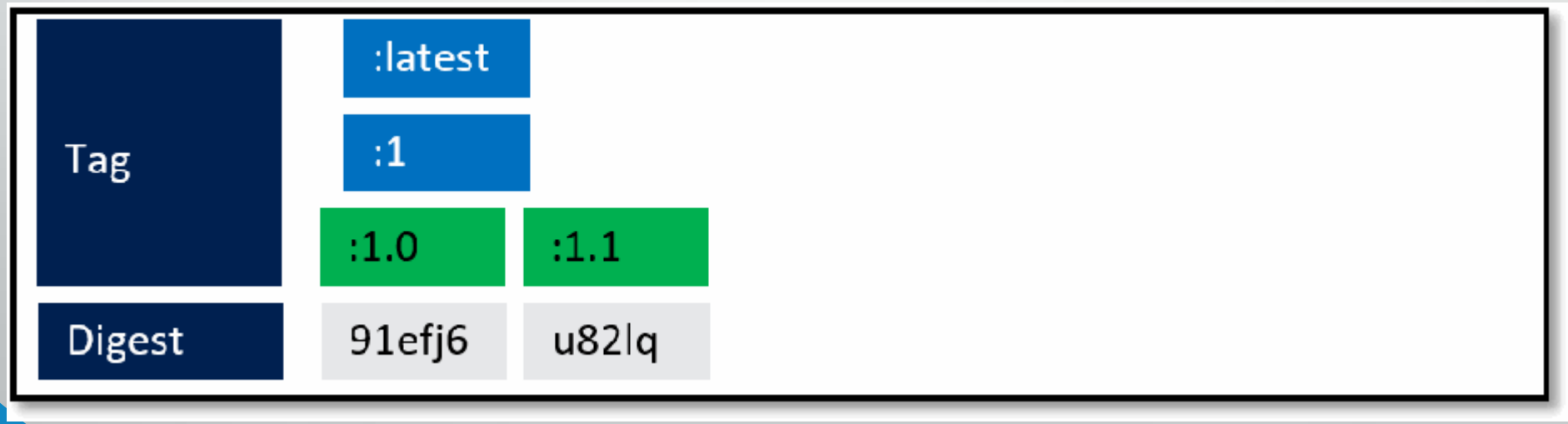

1 / 1
Tilix: Default
1: Terminal
[wagoodman@kiwi dive] master $ dive someproj:latest

Tags

Decide what version you run at any given time

“Latest” by default

Used for versioning



Container Registry

Repository to store container images

Docker Hub

Filters

1 - 15 of 15 results for **wolfgangofner**.

Best Match

Products

- [ ] Images
- [ ] Extensions
- [ ] Plugins

Trusted Content

- [ ] Docker Official Image
- [ ] Verified Publisher
- [ ] Sponsored OSS

Operating Systems

- [ ] Linux
- [ ] Windows

Architectures

- [ ] ARM
- [ ] ARM 64

wolfgangofner/microservicedemo · 100K+ · ☆0

By wolfgangofner · Updated 3 years ago

Linux x86-64

wolfgangofner/customerapi · 3.3K · ☆0

By wolfgangofner · Updated 17 days ago

Image for my NET 6 Microservice demo.

Linux x86-64

wolfgangofner/kubernetesdeploymentdemo · 2.1K · ☆0

By wolfgangofner · Updated 4 years ago

Linux x86-64

wolfgangofner/orderapi · 1.5K · ☆0

Container Registry

Repository to store container images

Docker Hub

Public vs. private registry

Azure Container Registry (ACR)

Additional functionalities like:

- Geo-replication
- Retention Policies
- Security scanning

Docker Compose

YAML file

Define container dependencies

Run all dependent containers

Docker Compose

Advantages

- Configure dependencies between containers
- Restart policy
- Easy to start

Docker Compose

Disadvantages

- Monitoring
- Load Balancing
- Deployment
- SSL Certificate

Docker Compose

YAML file

Define container dependencies

Run all dependent containers

Advantages

- Configure dependencies between containers
- Restart policy
- Easy to start

Disadvantages

- Monitoring
- Load Balancing
- Deployment
- SSL Certificate

```yaml
version: "3.9"

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql
```

Docker Recap

Small images

Fast start up and deployment

Reusable and portable

Immutable → “Works on my machine”

Containers allow you to run your software even if your infrastructure provider does not support it

Docker Command

List running containers

docker ps

List images

docker image ls

Download an image from a registry

docker pull wolfgangofner/customerapi

Build an image from a Dockerfile

docker build . [-f CustomerApi/Dockerfile]

Tag an image

docker tag customerapi wolfgangofner/customerapi

Push an image to a registry

docker push wolfgangofner/customerapi

Start a container

docker run -p 32789:80 -p 32788:443 wolfgangofner/customerapi

Docker Compose Demo

Exercise

Docker Exercise

- Run an image from Dockerhub
- Create a new application and build it in a Dockerfile
- Upload your image to Dockerhub
- Build and run a docker-compose file
- Use different docker commands to interact with images

containers don't solve all problems

Container Orchestrator

Multi-Node Management

Resource Management

Load balancing

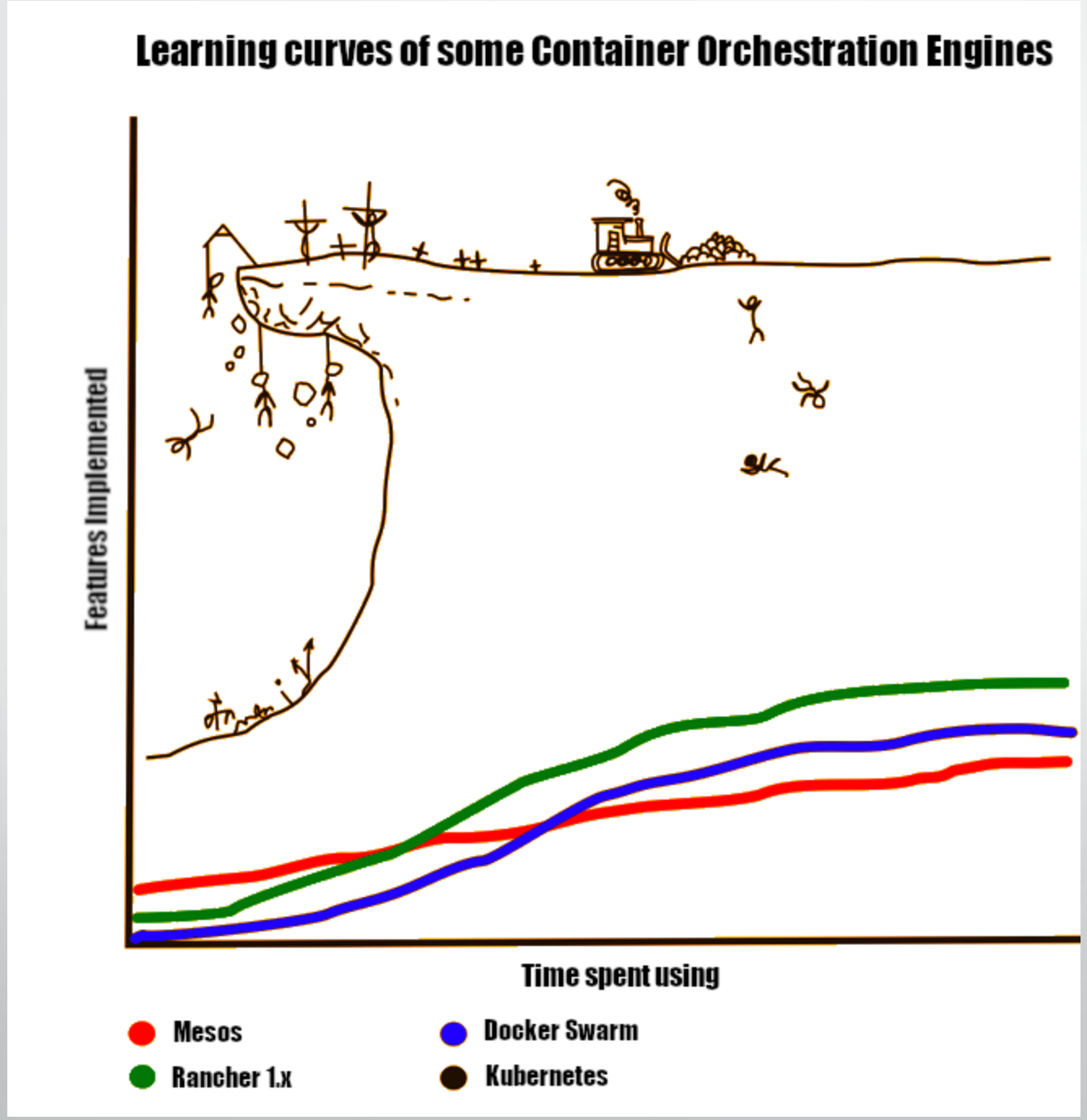
Monitoring and Self Healing

Zero Downtime Deployments

Manage TLS certificates

kubernetes

Container Orchestrator


Learning curves of some Container Orchestration Engines
Features Implemented
Time spent using
Mesos
Docker Swarm
Rancher 1.x
Kubernetes

Kubernetes

Kubernetes is an open-source system for automating computer application deployment, scaling, and **management of container applications**

First Release on 10 July 2015

Based on Google's Borg

Designed by Google and is now maintained by the Cloud Native Computing Foundation

Written in Go and open-source

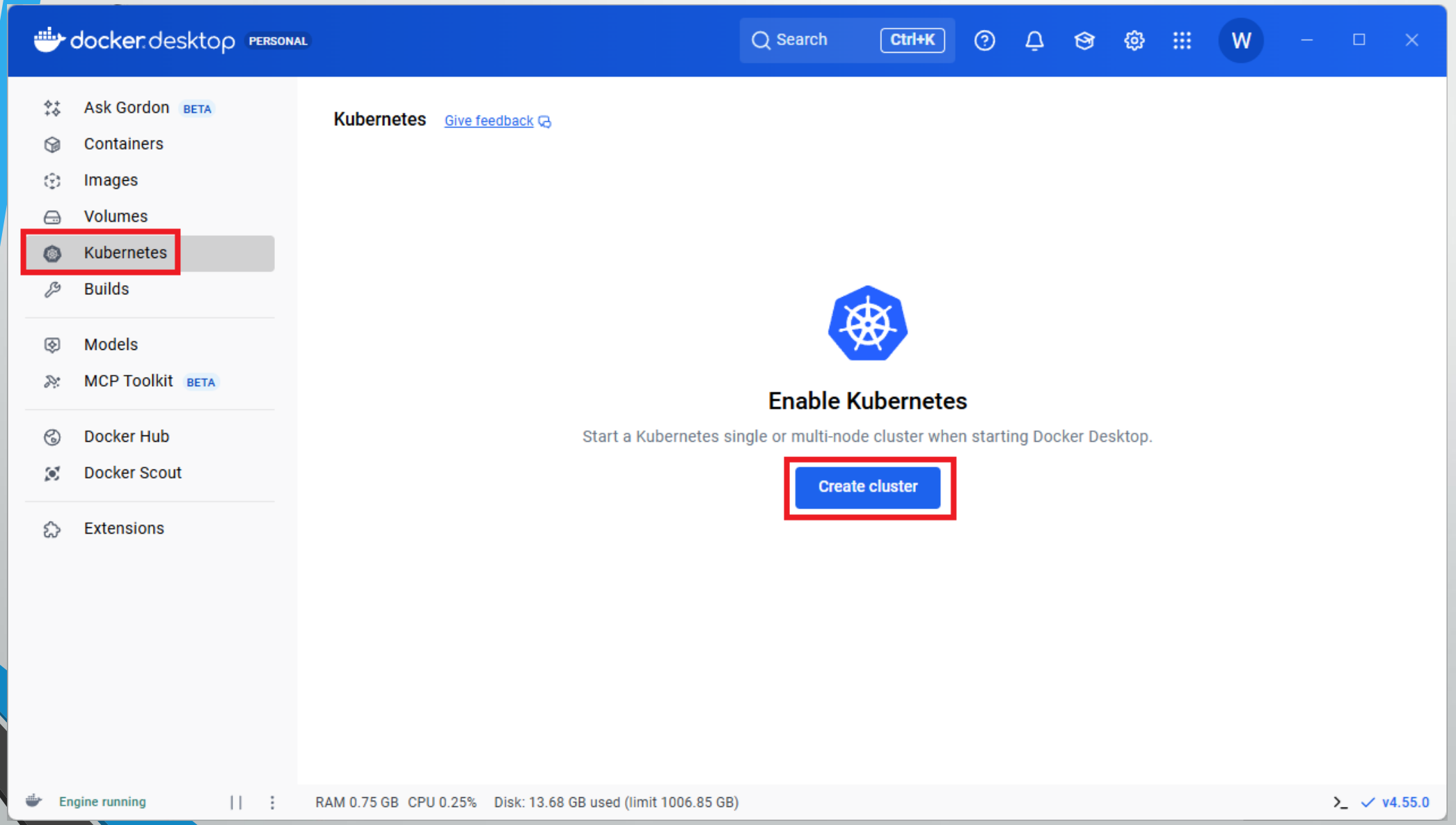
"K8s" → K-8 character-s

Kubernetes

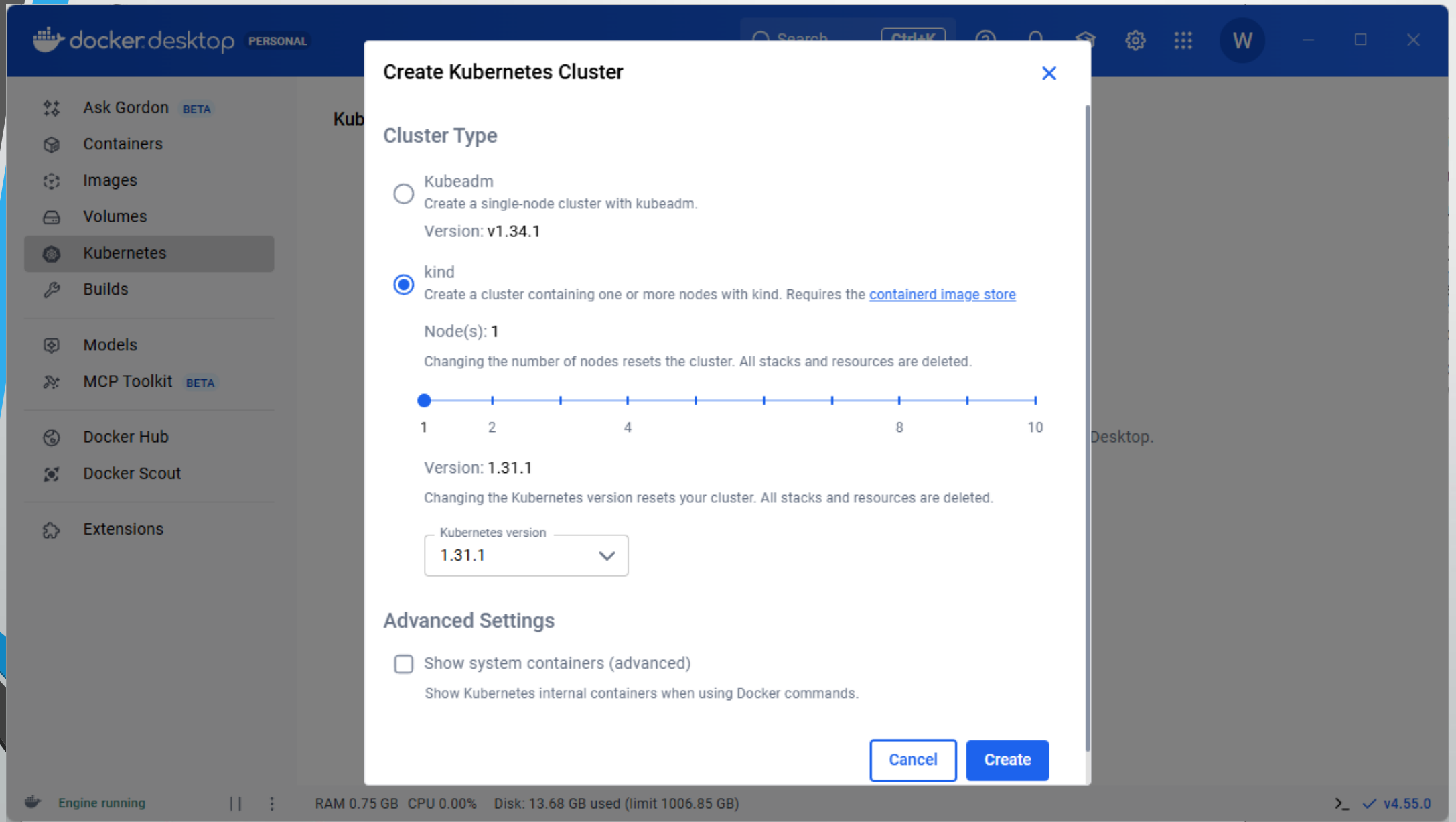
Multiple distributions

- K3s
- MicroK8s
- Kind
- Red Hat OpenShift
- Azure Kubernetes Service
- Amazon Elastic Kubernetes
- Google Kubernetes Engine

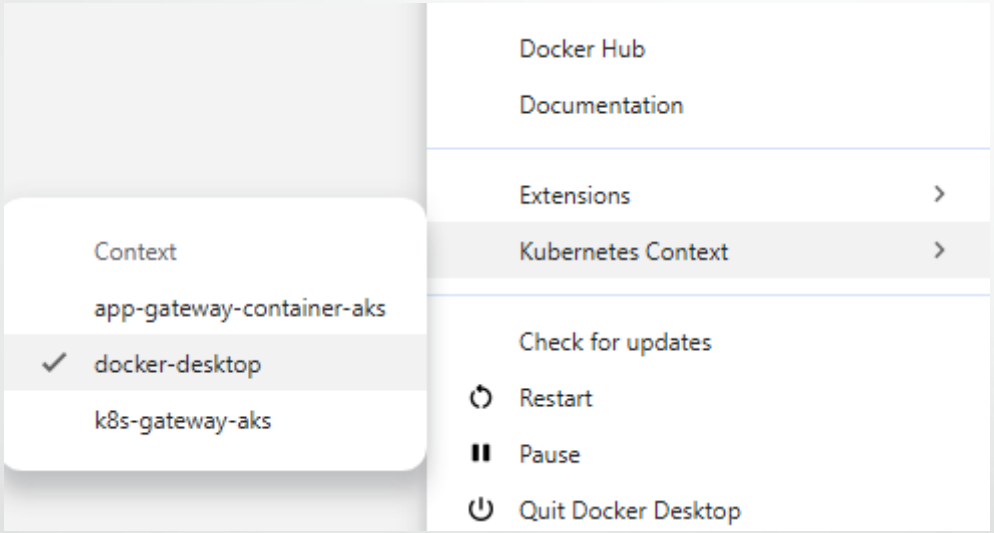
K8s locally


docker desktop
PERSONAL
Search
Ctrl+K
W
Ask Gordon
BETA
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit
BETA
Docker Hub
Docker Scout
Extensions
Kubernetes
Give feedback
Enable Kubernetes
Start a Kubernetes single or multi-node cluster when starting Docker Desktop.
Create cluster
Engine running
RAM 0.75 GB CPU 0.25%
Disk: 13.68 GB used (limit 1006.85 GB)
v4.55.0

K8s locally


docker desktop
PERSONAL
Search
Ctrl+K
W
Ask Gordon BETA
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions
Kub
Desktop.
Create Kubernetes Cluster
Cluster Type
Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.34.1
kind
Create a cluster containing one or more nodes with kind. Requires the containerd image store
Node(s): 1
Changing the number of nodes resets the cluster. All stacks and resources are deleted.
1
2
4
8
10
Version: 1.31.1
Changing the Kubernetes version resets your cluster. All stacks and resources are deleted.
Kubernetes version
1.31.1
Advanced Settings
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Cancel
Create
Engine running
RAM 0.75 GB CPU 0.00% Disk: 13.68 GB used (limit 1006.85 GB)
v4.55.0

K8s locally


Docker Hub
Documentation
Extensions
Kubernetes Context
Context
app-gateway-container-aks
docker-desktop
k8s-gateway-aks
Check for updates
Restart
Pause
Quit Docker Desktop

K8s locally

K3s

- Lightweight
- Great for edge and IoT
- Easy to install

curl -sfL https://get.k3s.io | sh -

Kubernetes Features

Self-healing

Service discovery and load balancing

Secret and configuration management

Horizontal scaling

Zero downtime deployments

Batch execution

Namespaces

Configuration in JSON or YAML

Self-healing Demo

Kubernetes Components

Master Node (Control Plane)

- kube-apiserver
- etcd
- kube-scheduler
- kube-control-manager
- Master Node is managed by cloud vendor

Worker Node

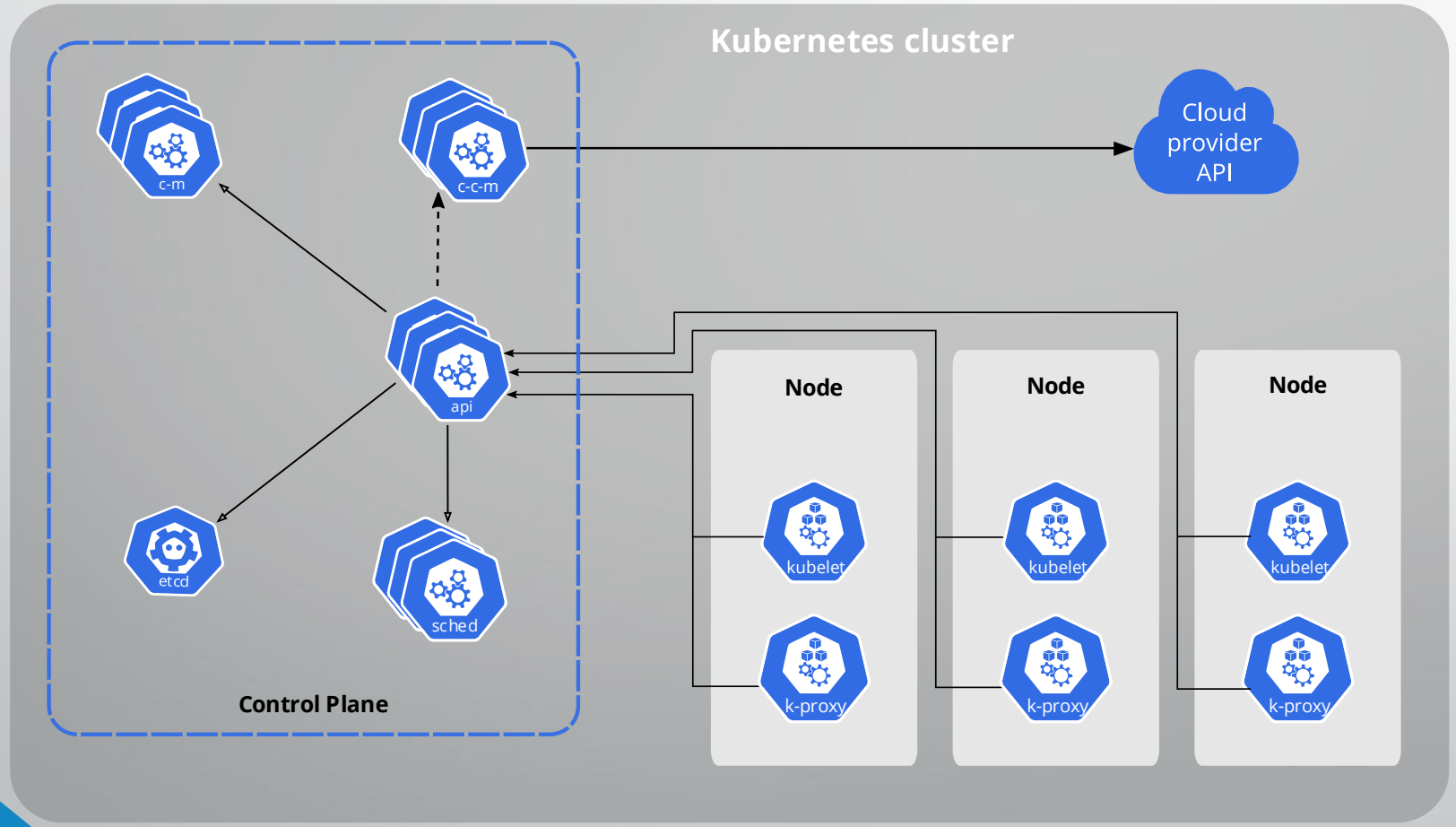
- kubelet
- kube-proxy
- Container runtime

Kubernetes Components

Addons

- DNS
- Networking
- Storage
- Dashboard
- …

Kubernetes Components



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

kube-proxy

Scheduler

Control plane

Node

Pod

A pod is the smallest unit in K8s

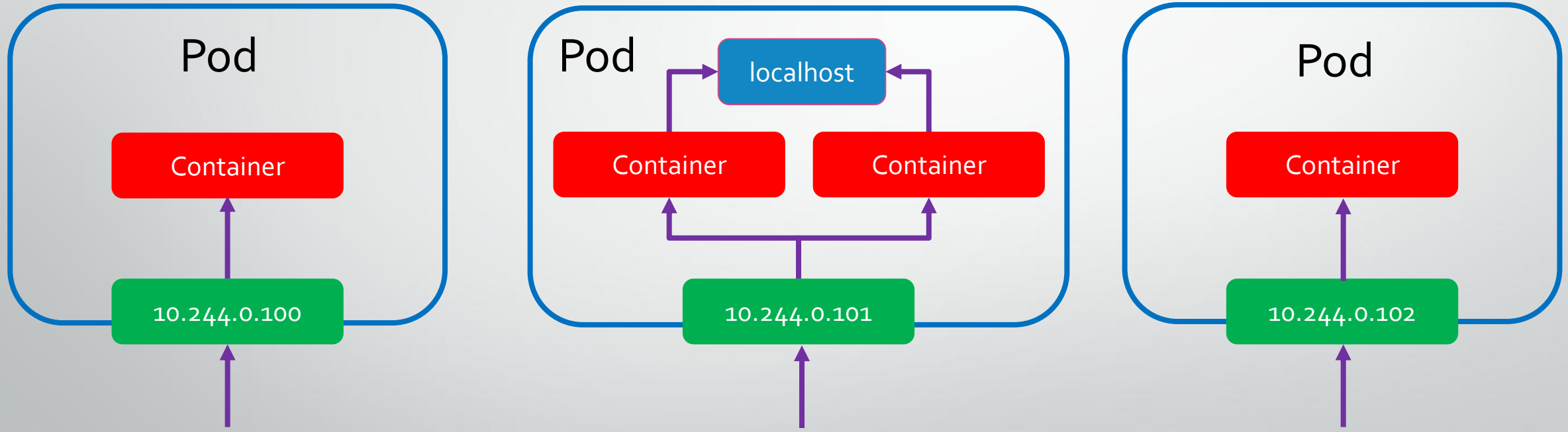
Pods wrap one ore more containers

Provides a way to set environment variables and mount storage

Containers inside a pod can communicate via localhost

Multiple containers should only be combined in a pod if they are interdependent

Pods and Containers



Kubernetes Networking

The default network IP range for pods is 10.244.0.0/16

Network CIDR range can be configured with addons:

- Cilium
- Flannel
- Calico

Namespaces

Used to create virtual cluster inside a physical cluster

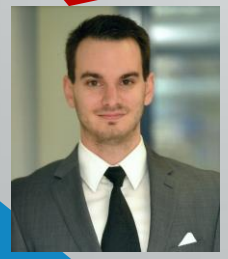
- Isolation
- Resource segregation
- Multiple environments
- Resource quotas
- Access control

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernetes what you want
- Kubernetes will figure out a way to the desired state
- etcd holds the current state component

Hey Kubernetes, run 3 pods of wolfgangofner/customerapi



Let me check if I am already running your pods

Currently there is 1 pod of wolfgangofner/customerapi running

Starting 2 more pods of wolfgangofner/customerapi

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernetes what you want
- Kubernetes will figure out a way to the desired state
- etcd holds the current state component

Configuration Handling

- YAML or JSON files
- Kubernetes CLI called kubectl
- kubectl communicates with Kubernetes API

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernetes what you want
- Kubernetes will figure out a way to the desired state
- etcd holds the current state compone

Configuration Handling

- YAML or JSON files
- Kubernetes CLI called kubectl
- kubectl communicates with Kubernet

YAML File

```yaml
apiVersion: v1
kind: Service
metadata:
  name: kubernetesdemo-service
spec:
  type: LoadBalancer
  selector:
    app: kubernetesdemo
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kubernetesdemo-deployment
  labels:
    app: kubernetesdemo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kubernetesdemo
  template:
    metadata:
      labels:
        app: kubernetesdemo
    spec:
      containers:
      - name: kubernetesdemo
        image: wolfgangofner/kubernetesdeploymentdemo:start
        ports:
        - containerPort: 80
```

Labels and Annotations

Labels

- Key valuepairs that are bound to objects like deployments or pods with a maximum of 63 character
- app:MyAppName
- Used to filter or select objects
- Can be changed or deleted at any times

Labels and Annotations

Annotations

- Also key value pairs but without the character limitation
- Can not be used for filtering or selecting objects

```
metadata:
  creationTimestamp: "2021-10-17T11:58:22Z"
  labels:
    component: apiserver
    provider: kubernetes
```

Metadata

Age	51m
Labels	component:apiserver provider:kubernetes

Services

Pods come and go

IP addresses will change

Services stay for the entire lifetime of the application

Persistent entry point

Fixed IP address

Load Balancing

Load Balancer Demo

Services

Pods come and go

IP addresses will change

Services stay for the entire lifetime of the application

Persistent entry point

Fixed IP address

Load Balancing

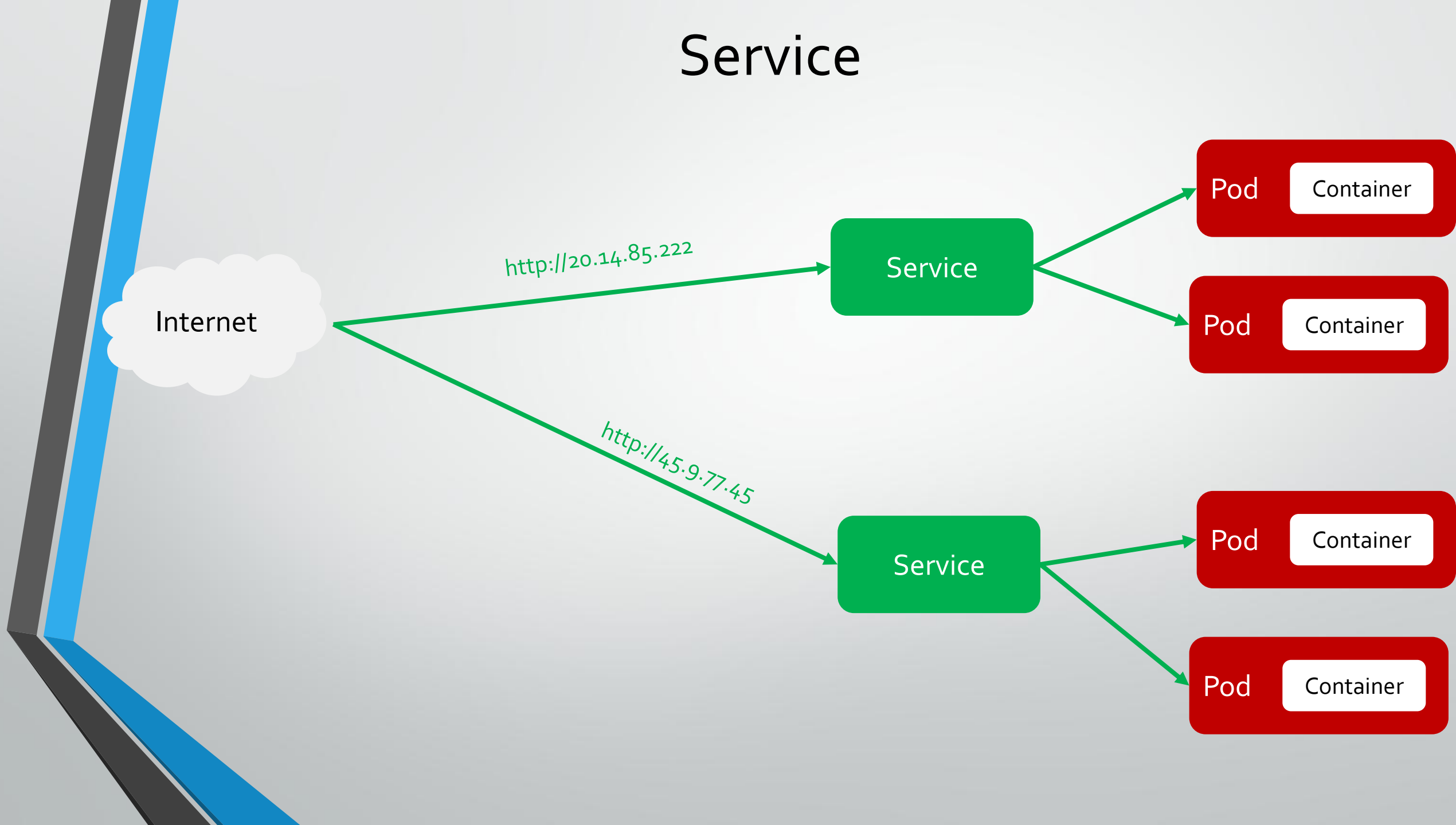
Pods and Services are matched using Labels

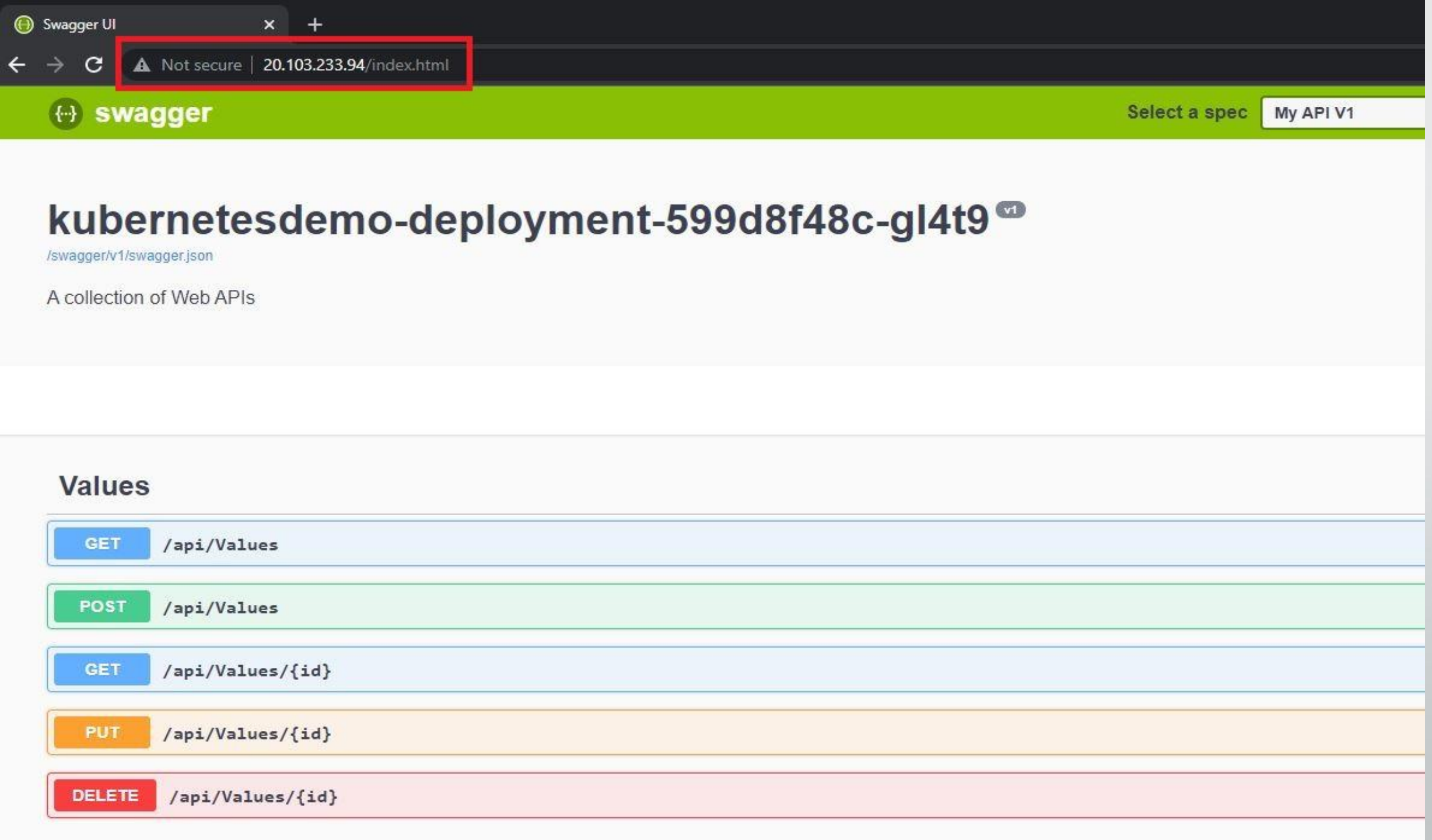
Services

Four types of Services:

- **LoadBalancer**: exposes the Service using an external load balancer
- **ClusterIP**: makes the Service accessible only from within the cluster
- **NodePort**: exposes the Service at each node's IP at a static port
- **ExternalName**: maps the service to an existing DNS FQDN

Service


Internet
http://20.14.85.222
http://45.9.77.45
Service
Service
Pod
Container
Pod
Container
Pod
Container
Pod
Container


Swagger UI
Not secure | 20.103.233.94/index.html
swagger
Select a spec
My API V1
kubernetesdemo-deployment-599d8f48c-gl4t9
v1
/swagger/v1/swagger.json
A collection of Web APIs
Values
GET /api/Values
POST /api/Values
GET /api/Values/{id}
PUT /api/Values/{id}
DELETE /api/Values/{id}

Kubectl Commands

Get resource

kubectl get pods/service/deployment

Delete resource

kubectl delete pod/service/deployment

Display information about resource

kubectl desrcibe pod/node/service resource-name

Add/update new resource

kubectl apply --f myfile.yaml [namespace=my-namespace]

Set current namespace

kubectl config set-context --current --namespace=my-namespace

Kubernetes Cheat Sheet:
https://kubernetes.io/docs/reference/kubectl/cheatsheet

Exercise

Exercise

Connect to the Kubernetes cluster

Create a new namespace

- kubectl create ns my-namespace

Apply provided YAML file to your namespace

- kubectl apply -f filepath --namespace=my-namespace
- kubectl get all -n my-namespace

Set Replicas to 3 and check what happens

Replace image in Deployment with your Docker image

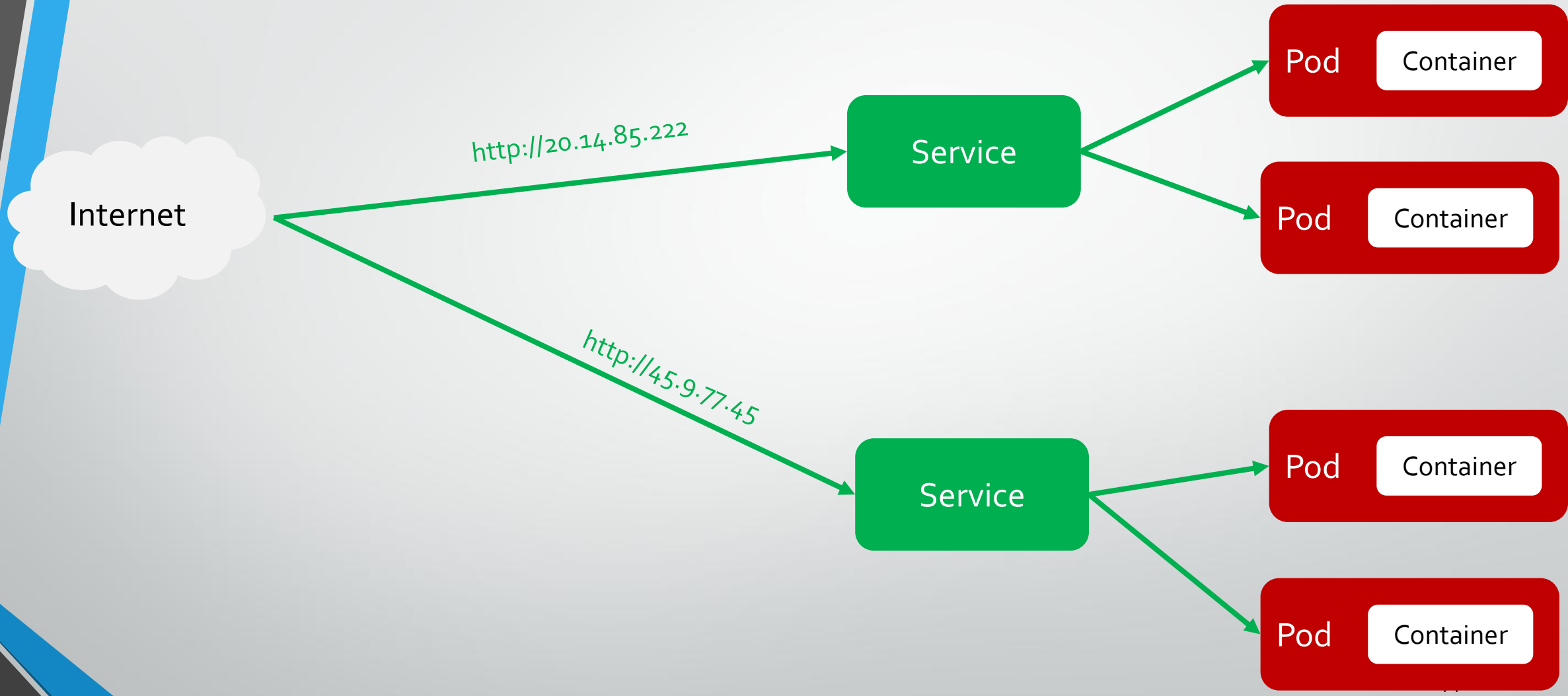
Recap

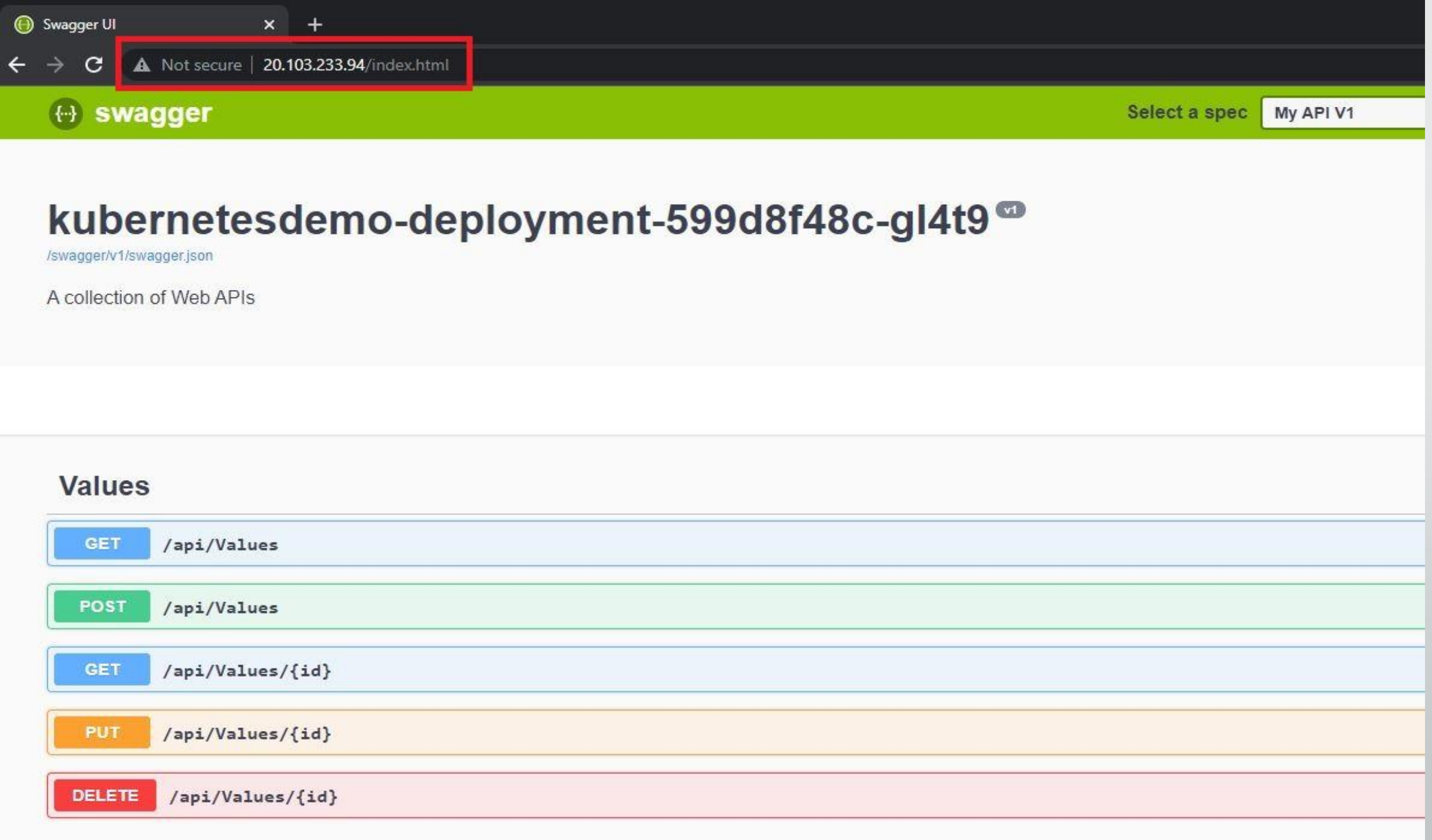
Container are immutable objects

Pods run containers in Kubernetes

Services are an entry point and route requests to pods

Service


Internet
http://20.14.85.222
http://45.9.77.45
Service
Service
Pod
Container
Pod
Container
Pod
Container
Pod
Container


Swagger UI
Not secure | 20.103.233.94/index.html
swagger
Select a spec
My API V1
kubernetesdemo-deployment-599d8f48c-gl4t9
v1
/swagger/v1/swagger.json
A collection of Web APIs
Values
GET /api/Values
POST /api/Values
GET /api/Values/{id}
PUT /api/Values/{id}
DELETE /api/Values/{id}

Recap

Needed for production environment:

- Secret Management
- Health Checks
- TLS Certificates
- Deployments
- Resource Management

Secrets

Base64 encoded

Automatically decrypted when attached to a pod

Can be used in config file or environment variable

Cert-Issuer

Kubernetes resource

Handles certificate requests

Supported issuers

- **Let's Encrypt**
- HashiCorp Vault
- Venafi
- Private PKI (Public Key Infrastructure)

```yaml
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    email: <Your Email>
    privateKeySecretRef:
      name: letsencrypt
    solvers:
    - http01:
        ingress:
          class: nginx
          podTemplate:
            spec:
              nodeSelector:
                "kubernetes.io/os": linux
```

Cert-Manager

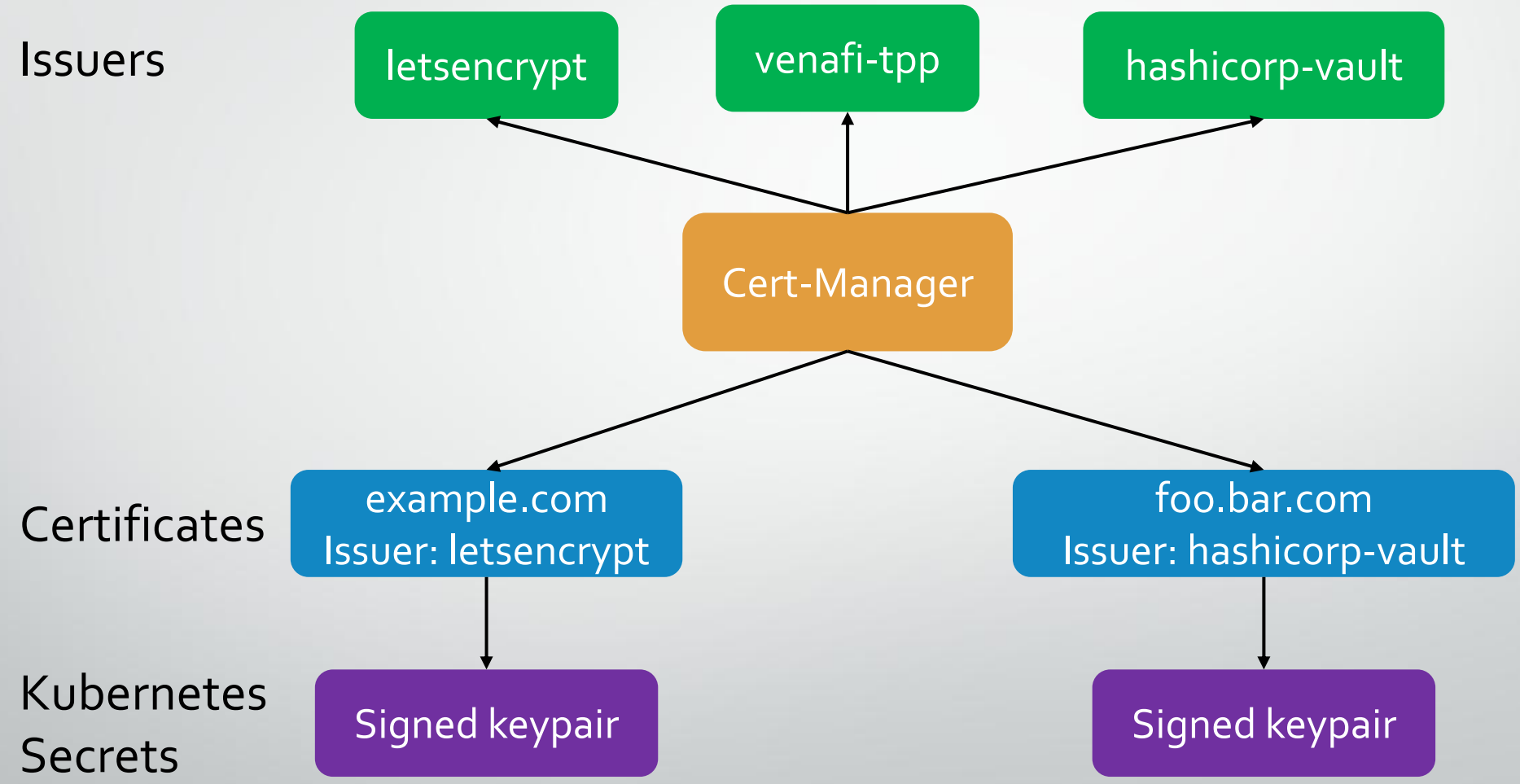
Manages obtaining and renewing of certificates

Can use variety of CAs like Let's Encrypt, HashiCorp Vault, and Venafi

Updates certificates at a configured time before expiry

Uses Cert Issuer to issue certificates

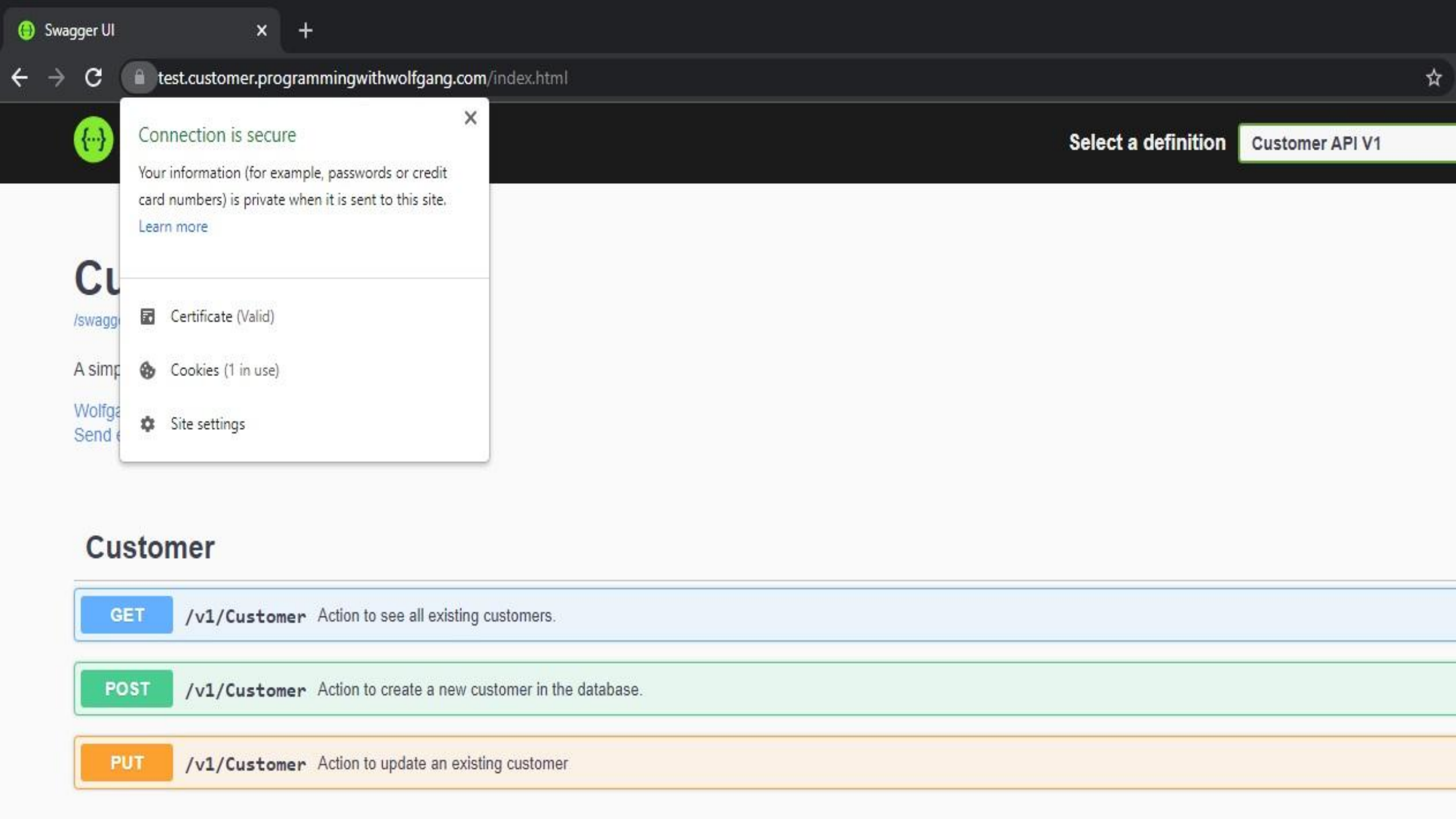
Cert-Manager



kedademoapi-tls

Summary Metadata Resource Viewer YAML

```yaml
---
apiVersion: v1
data:
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZXVENDQkVHZ0F3SUJBZ0lTQS
  tls.key: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb2dJQkFBS0NBUUVBNG
kind: Secret
metadata:
  annotations:
    cert-manager.io/alt-names: test.kedademo.programmingwithwolfgang.com
    cert-manager.io/certificate-name: kedademoapi-tls
    cert-manager.io/common-name: test.kedademo.programmingwithwolfgang.com
    cert-manager.io/ip-sans: ""
    cert-manager.io/issuer-group: cert-manager.io
    cert-manager.io/issuer-kind: ClusterIssuer
    cert-manager.io/issuer-name: letsencrypt
    cert-manager.io/uri-sans: ""
  creationTimestamp: "2021-10-17T12:07:46Z"
```


Swagger UI
test.customer.programmingwithwolfgang.com/index.html
Connection is secure
Your information (for example, passwords or credit card numbers) is private when it is sent to this site.
Learn more
Certificate (Valid)
Cookies (1 in use)
Site settings
Select a definition
Customer API V1
Customer
GET /v1/Customer Action to see all existing customers.
POST /v1/Customer Action to create a new customer in the database.
PUT /v1/Customer Action to update an existing customer

Ingress Controller

Entry point into the cluster

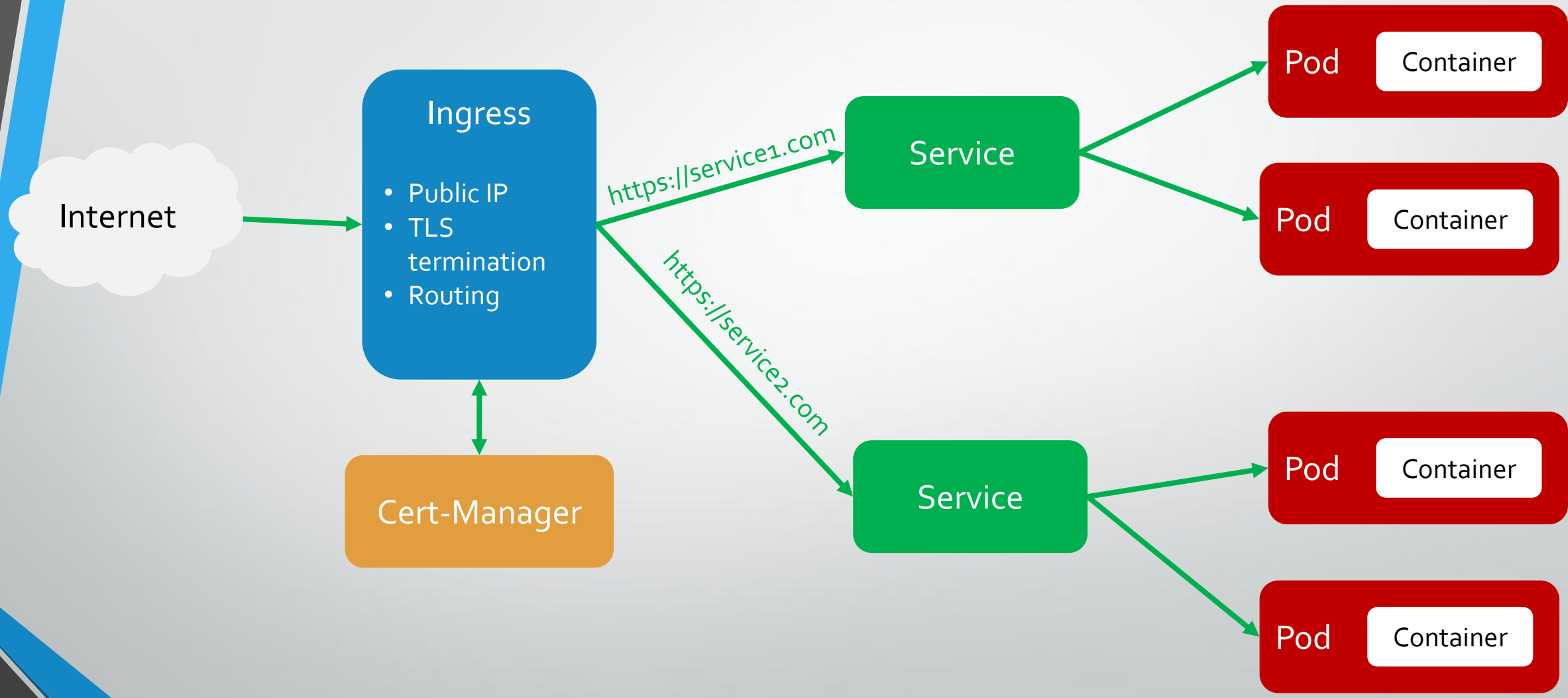
Processes HTTPS traffic

Reverse proxy redirects requests to Application

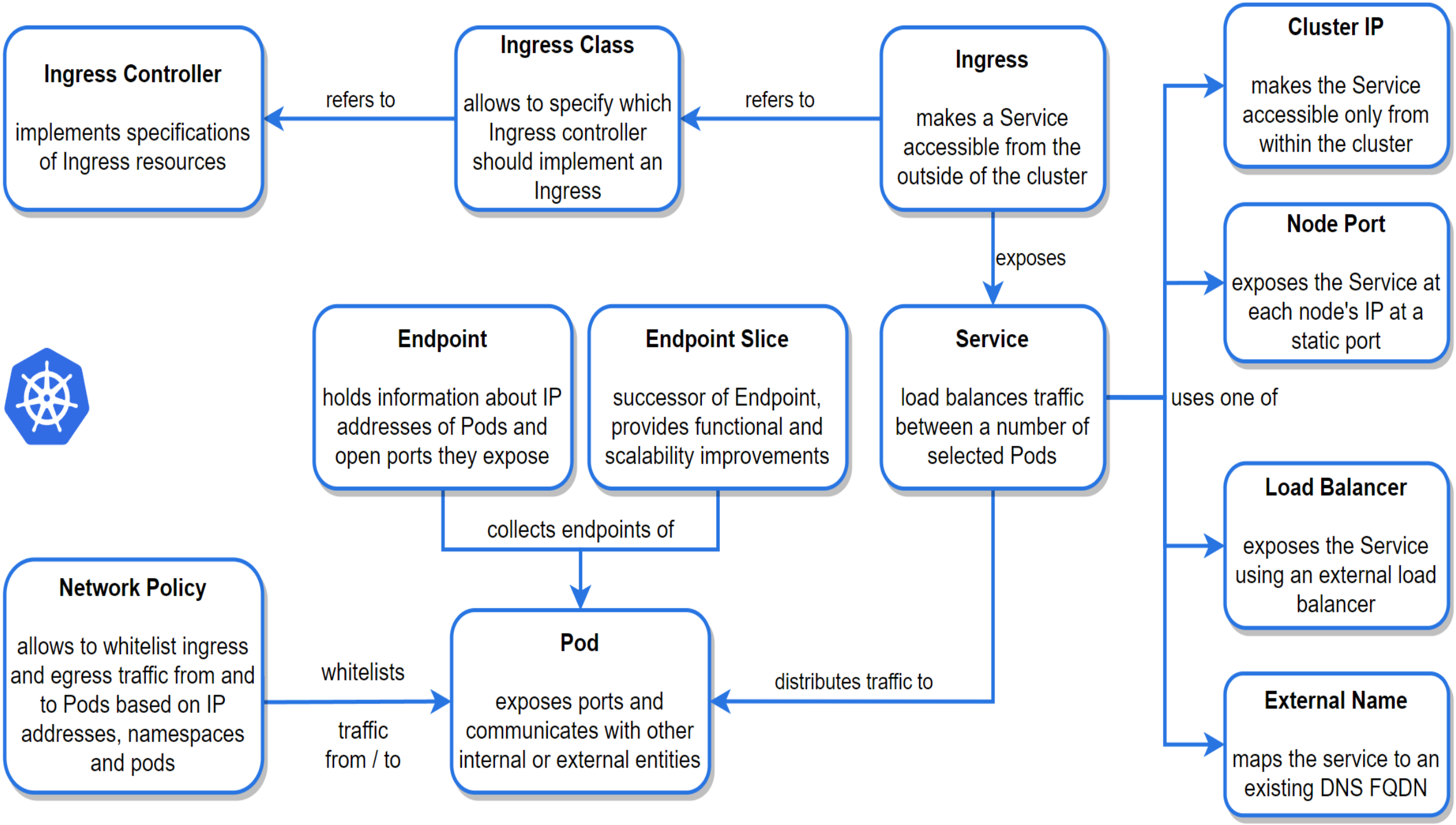
Popular Ingress Controller:

- NGINX
- Traefik
- HAProxy

Ingress Controller



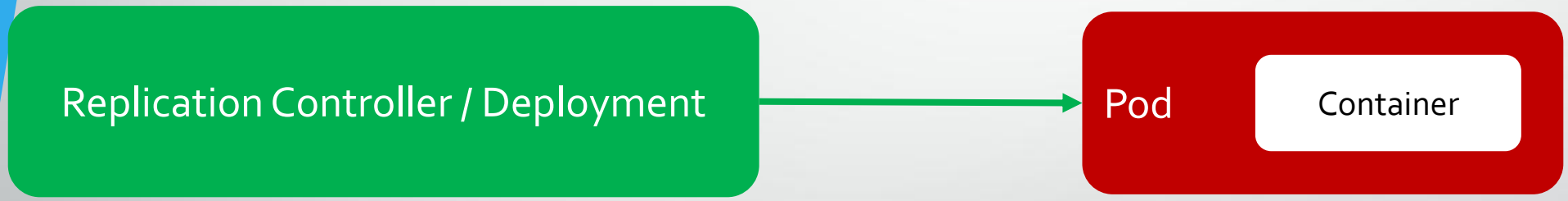
Pull Request Demo


Ingress Controller
implements specifications of Ingress resources
refers to
Ingress Class
allows to specify which Ingress controller should implement an Ingress
refers to
Ingress
makes a Service accessible from the outside of the cluster
exposes
Cluster IP
makes the Service accessible only from within the cluster
Node Port
exposes the Service at each node's IP at a static port
Load Balancer
exposes the Service using an external load balancer
External Name
maps the service to an existing DNS FQDN
uses one of
Service
load balances traffic between a number of selected Pods
Endpoint
holds information about IP addresses of Pods and open ports they expose
Endpoint Slice
successor of Endpoint, provides functional and scalability improvements
collects endpoints of
Pod
exposes ports and communicates with other internal or external entities
distributes traffic to
Network Policy
allows to whitelist ingress and egress traffic from and to Pods based on IP addresses, namespaces and pods
whitelists
traffic from / to

Pod Deployment

Pods are not directly deployed

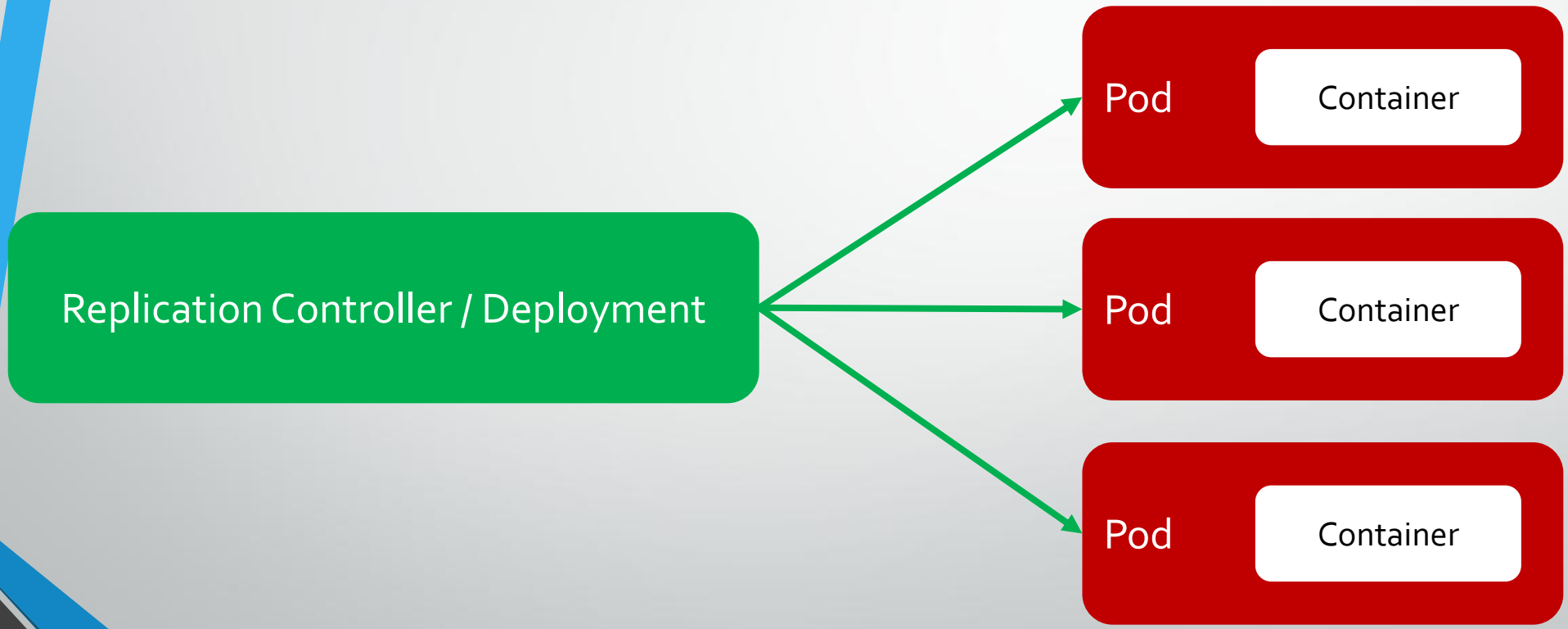
Deployment / Replicate Set create pods



Pod Deployment

Pods are not directly deployed

Deployment / Replicate Set create pods



Pod Deployment

Pods are not directly deployed

Deployment / Replicate Set create pods

Deployments manage ReplicaSets

Manages stateless applications

DaemonSet,CronJob, StatefulSet

Alternatively to Deployments, pods can be run using DaemonSets, CronJobs, and StatefulSets

CronJobs can be scheduled to start pods

StatefulSets manage stateful applications

DaemonSet,CronJob, StatefulSet

DeamonSets run pods on every node in the cluster

- Logging
- Monitoring
- Backup
- Reports
- Automated testing
- Databases

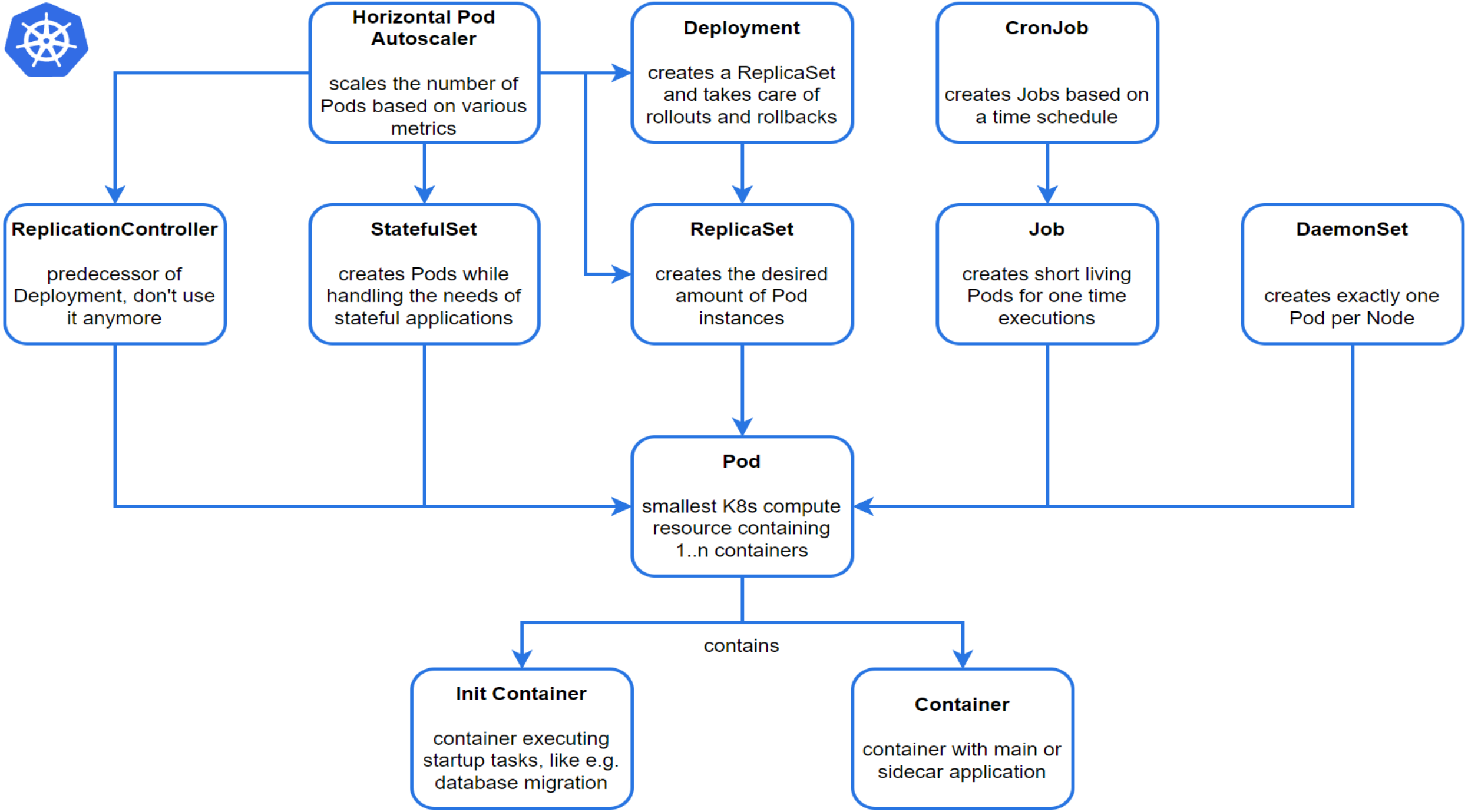
Horizontal Pod Autoscaler (HPA)

Queries resource utilization, e.g. CPU and RAM usage

Instructs ReplicationSet to scale out or scale in

Configures minimum and maximum number of pods

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: customerapi
spec:
  maxReplicas: 10
  minReplicas: 1
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: customerapi
  targetCPUUtilizationPercentage: 50
```


Horizontal Pod Autoscaler
scales the number of Pods based on various metrics
Deployment
creates a ReplicaSet and takes care of rollouts and rollbacks
CronJob
creates Jobs based on a time schedule
ReplicationController
predecessor of Deployment, don't use it anymore
StatefulSet
creates Pods while handling the needs of stateful applications
ReplicaSet
creates the desired amount of Pod instances
Job
creates short living Pods for one time executions
DaemonSet
creates exactly one Pod per Node
Pod
smallest K8s compute resource containing 1..n containers
contains
Init Container
container executing startup tasks, like e.g. database migration
Container
container with main or sidecar application

Auto Scaling Demo

Docker Health Checks

Docker provides a simple health check

--interval=DURATION

--timeout=DURATION

--start-period=DURATION

--retries should be set to N

Docker Health Checks

Problems with simple health checks

- Application startup may be longer than expected
- Different startup and health checks
- Specific port for checks

Liveness Probe

Checks if pod is alive

Sends HTTP request to check pod

Alive if answer >= HTTP 200 & < HTTP 400

- Pod will be restarted if dead

Configuration is part of the Deployment

```
livenessProbe:
  httpGet:
    path: /health
    port: http
  initialDelaySeconds: 15
```

Readiness Probe

Checks if pod is ready to receive traffic

Sends HTTP request to check pod

Alive if answer >= HTTP 200 & < HTTP 400

- Traffic will be routed to the pod when ready

Configuration is part of the Deployment

```
readinessProbe:
  httpGet:
    path: /health
    port: http
  initialDelaySeconds: 15
```

Resource Requests & Resource Limits

1000 Millicores = 1 Core

Memory is defined in bytes

Mebibyte = ~1MB

Configured in Deployment

Resource Requests & Resource Limits

Resource Requests

- Describe how many free resources a node has to have
- CPU and/or RAM

```
resources:
  limits:
    cpu: 0.3
    memory: 128Mi
  requests:
    cpu: 100m
    memory: 64Mi
```

Resource Limits

- Maximum resources a pod is allowed to use
- Pod gets throttled when is uses too many resources
- If throttling is not successful → pod will be evicted
- CPU and/or RAM

Resource Quotas

Limit resource usage within a namespace

Configured in the ResourceQuota object

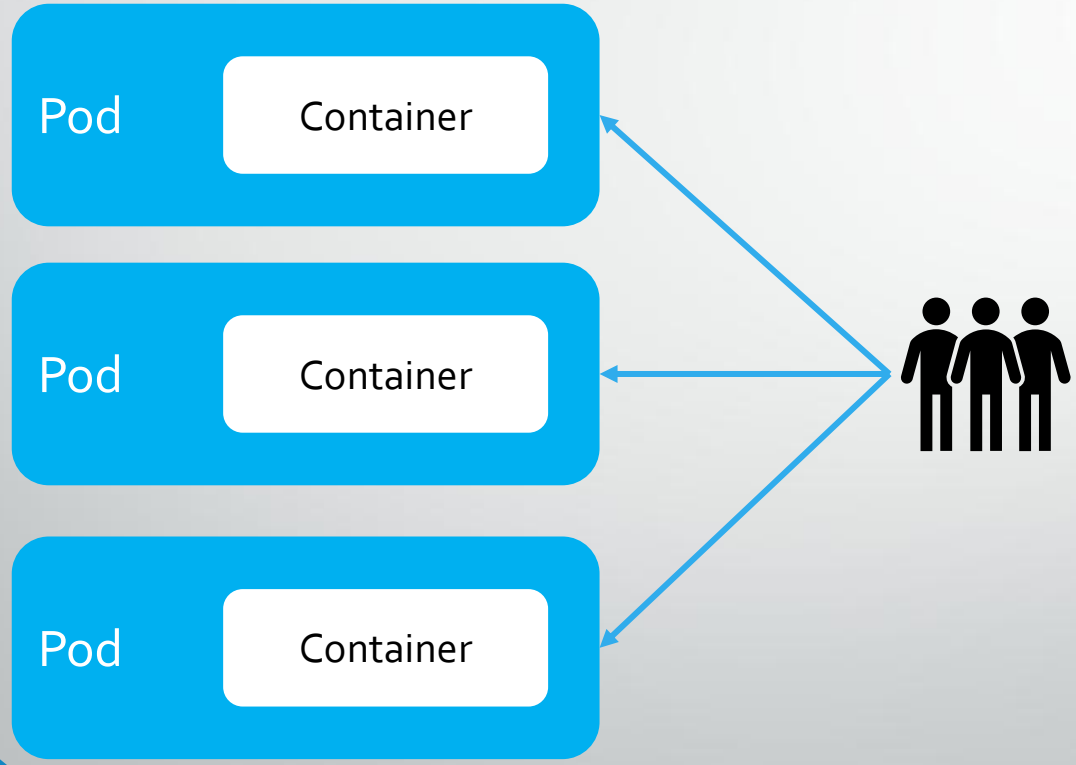
Restrict the following resources

- Number of pods
- CPU and RAM per request
- Total CPU and RAM usage

Deployment Strategies

Blue Green deployment

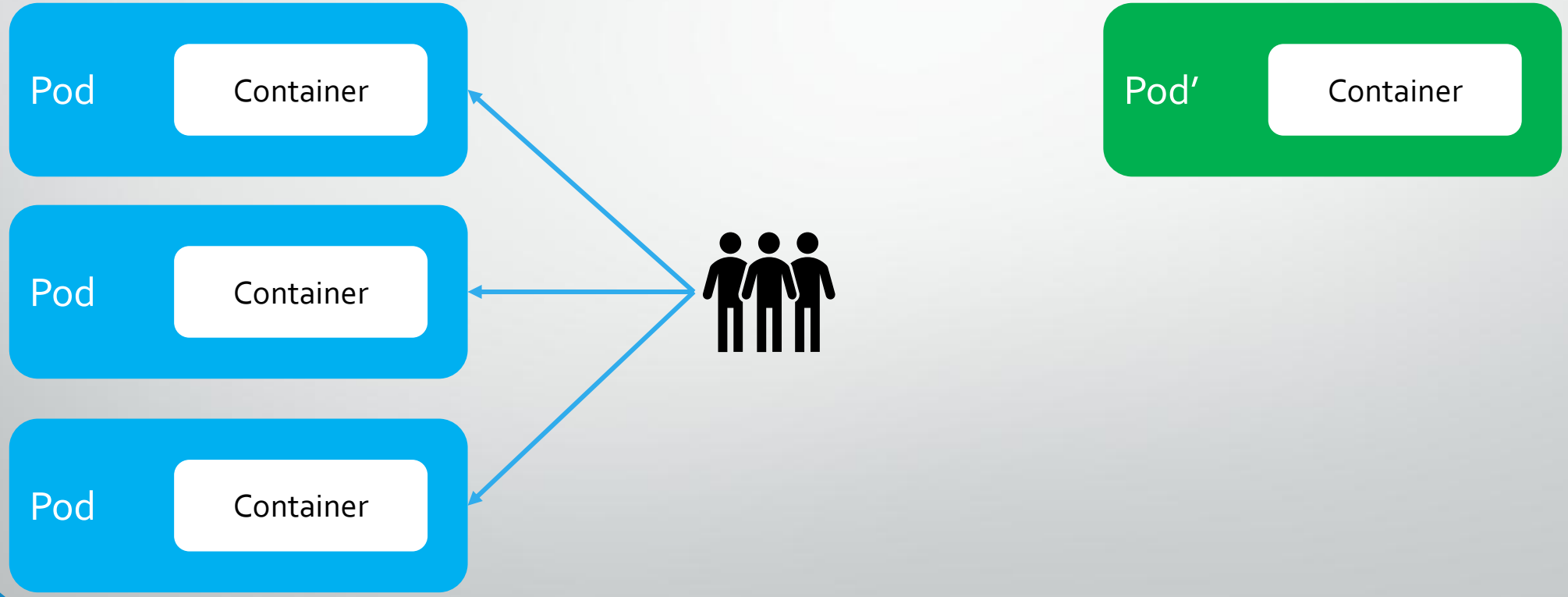
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

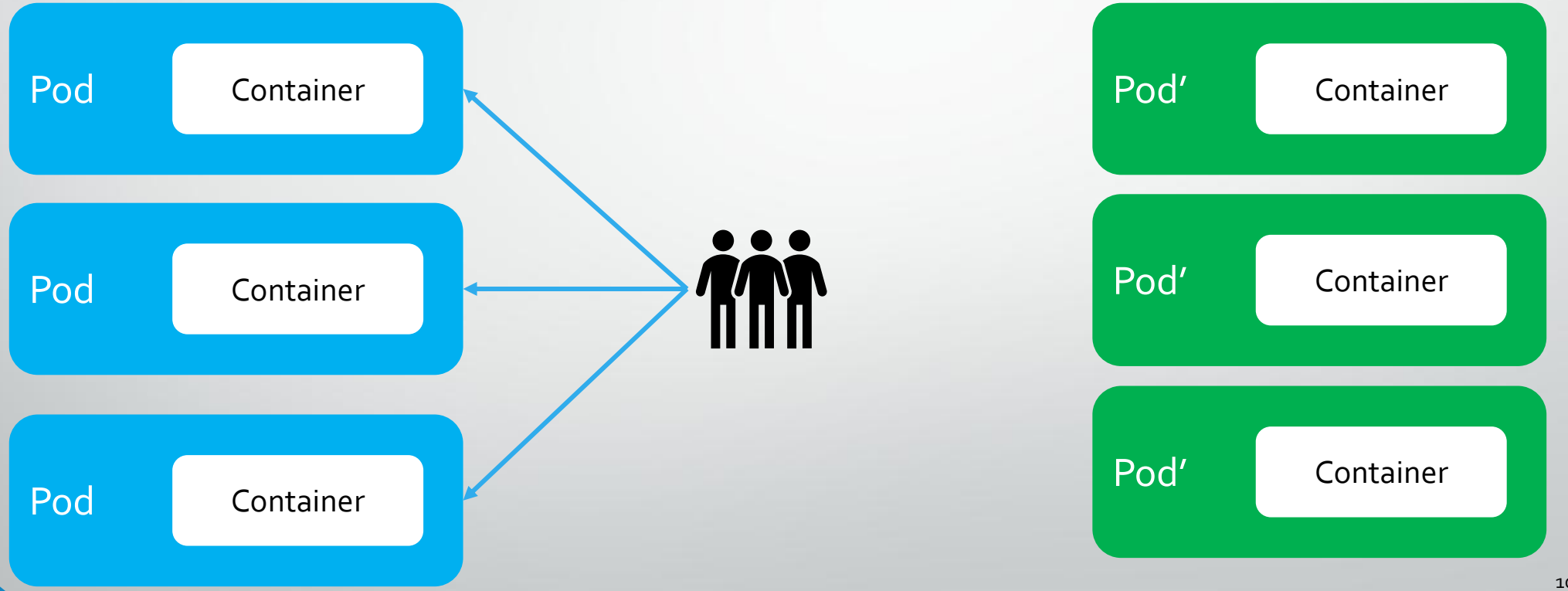
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

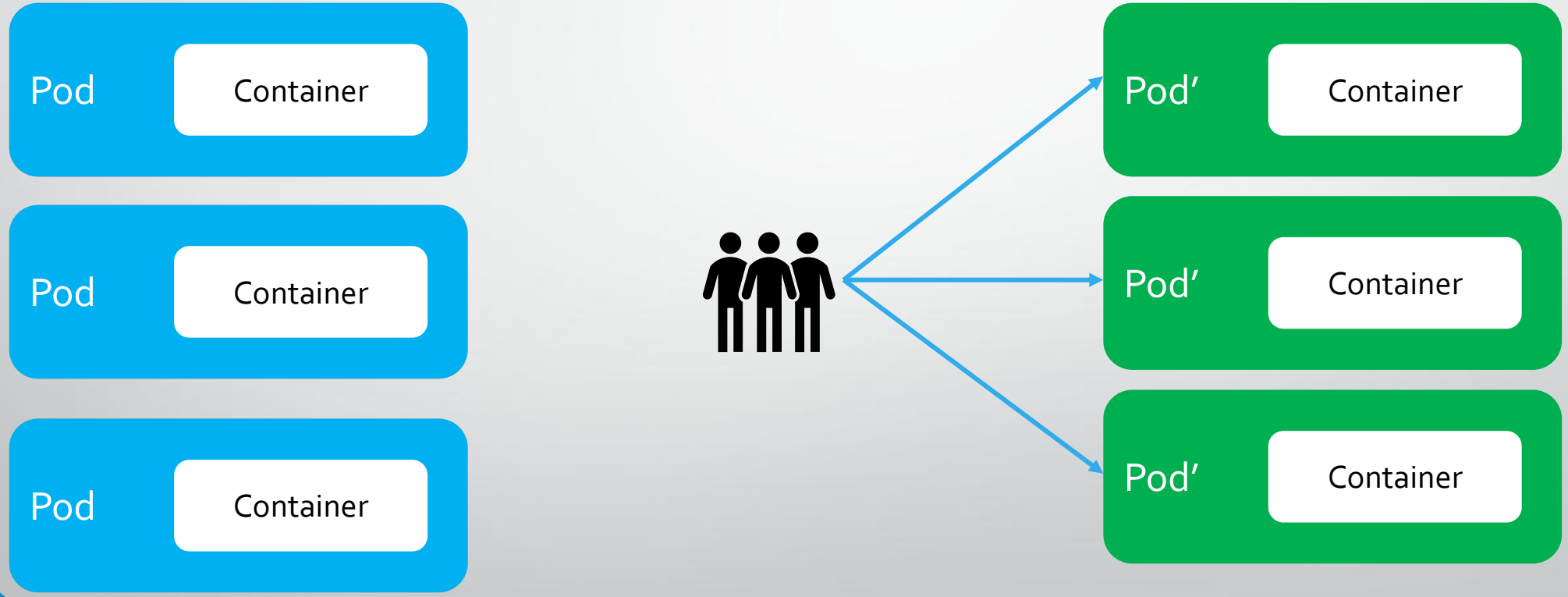
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

Start all new pods and then switch



Deployment Strategies

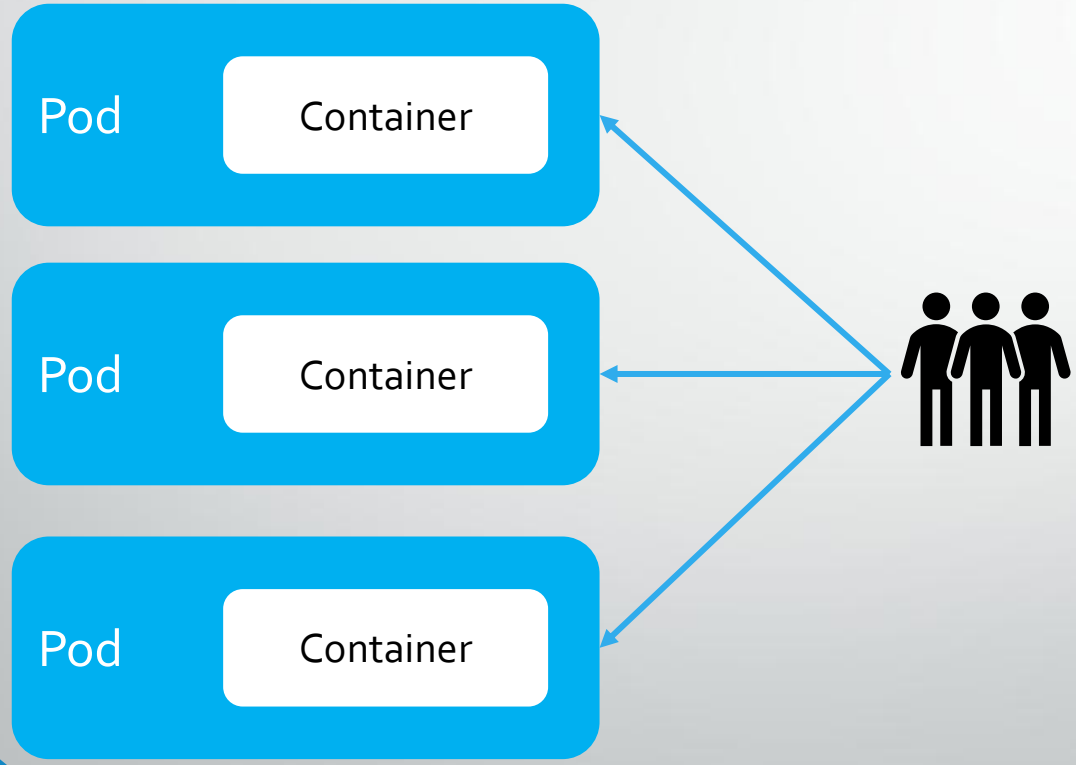
Rolling deployment

Replace old pods with new ones until all are replaced

Deployment Strategies

Rolling deployment

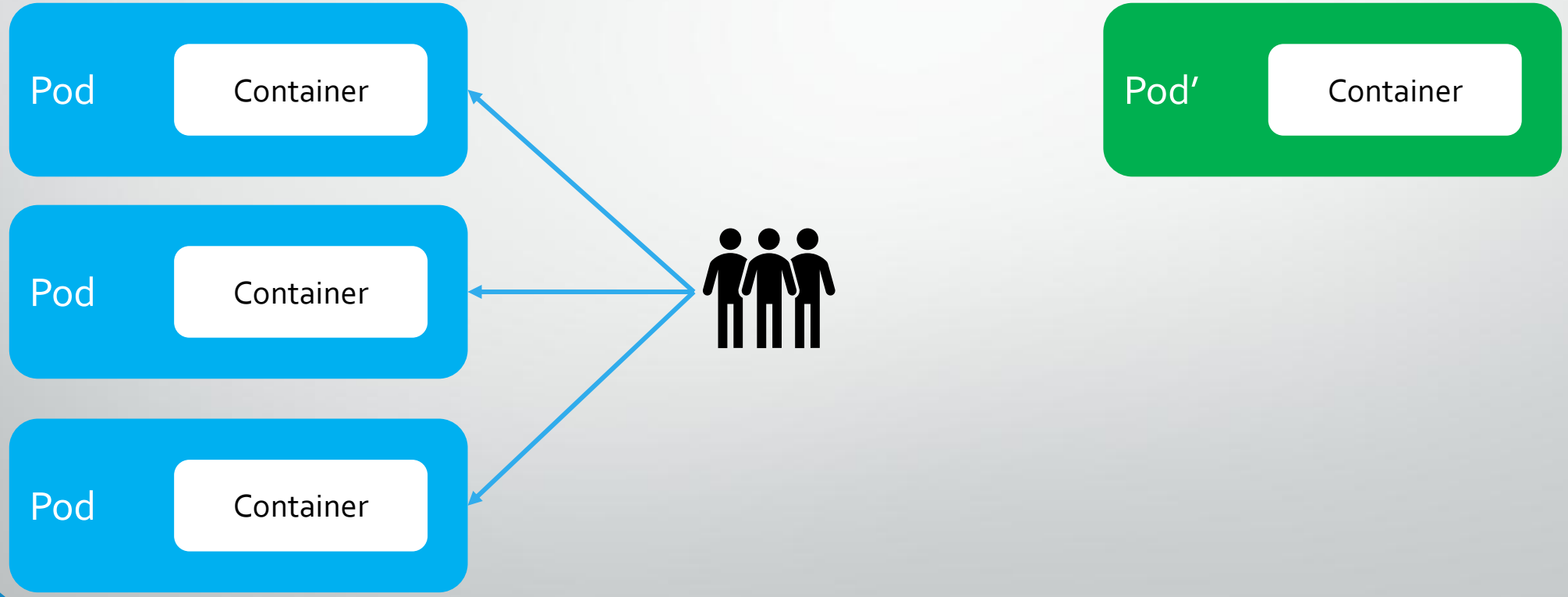
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

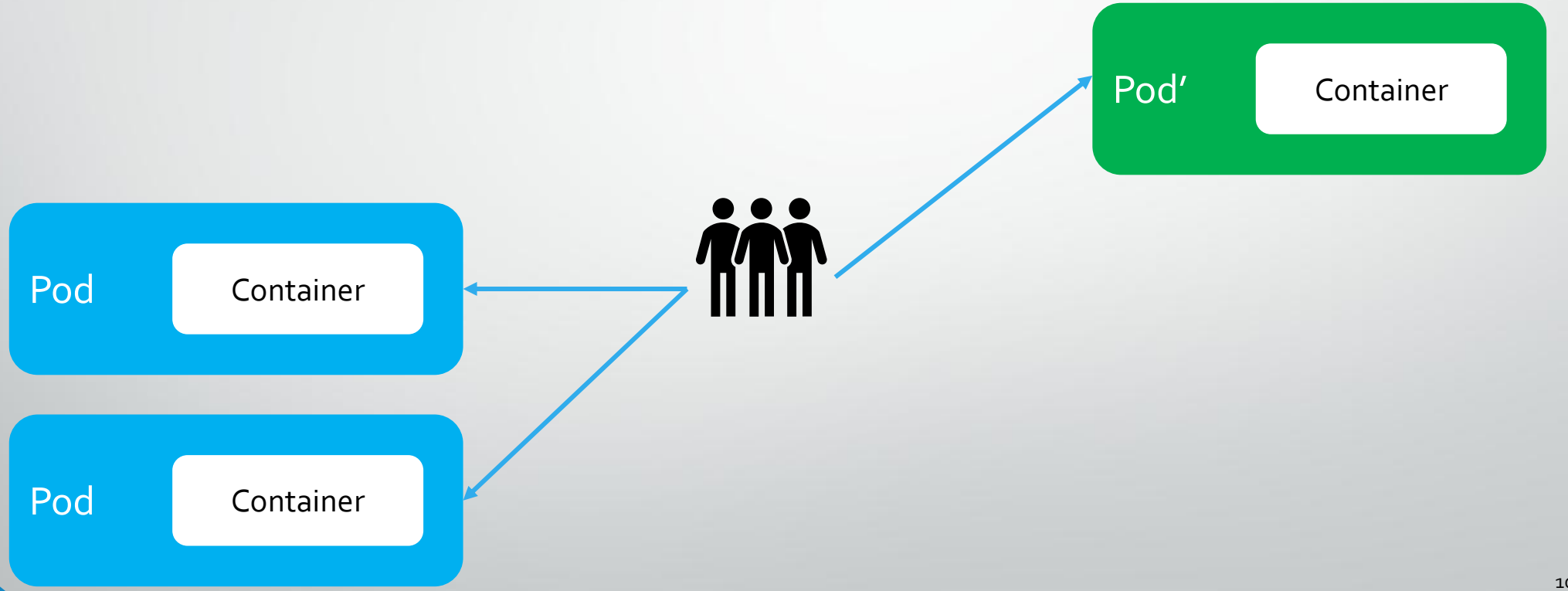
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

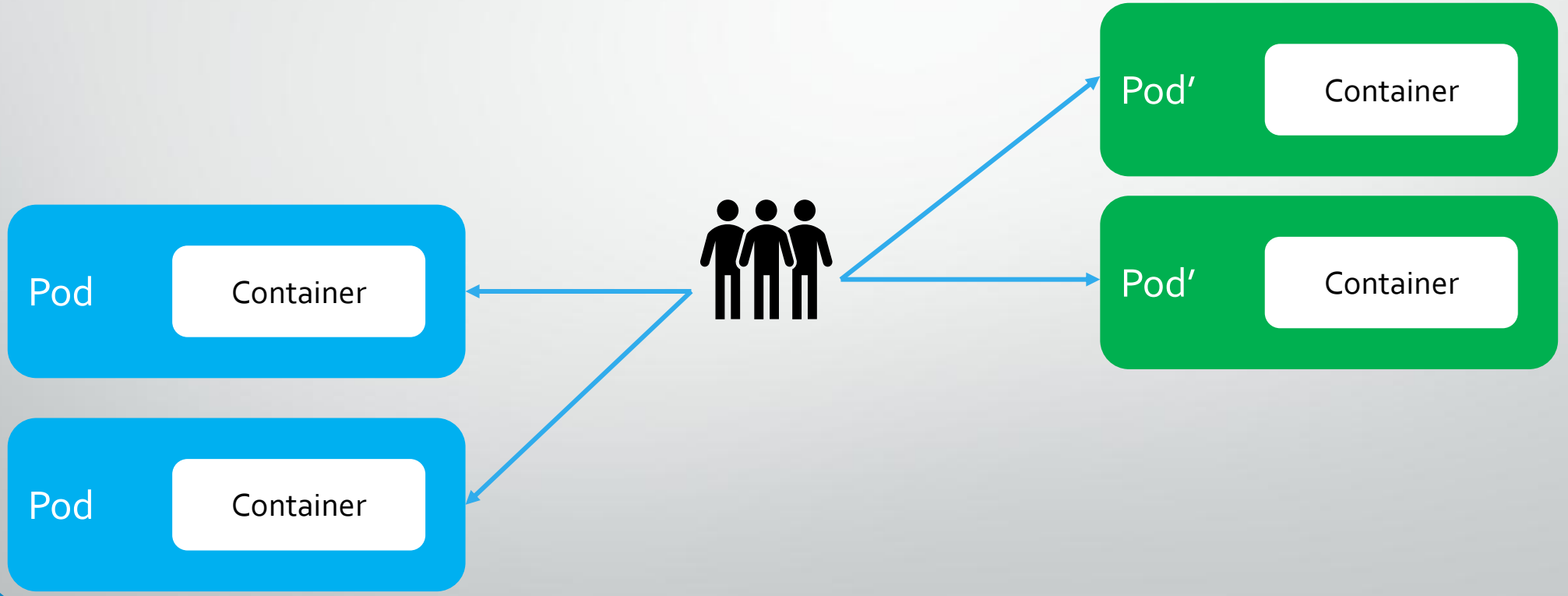
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

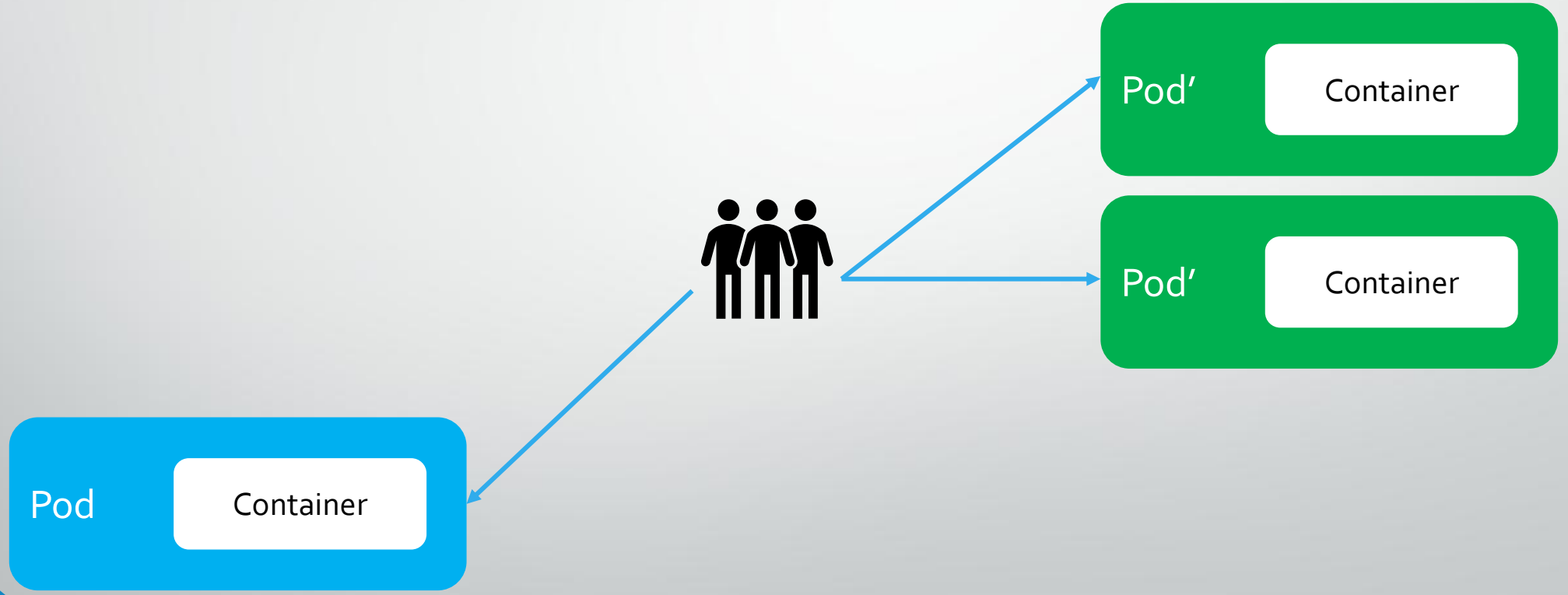
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

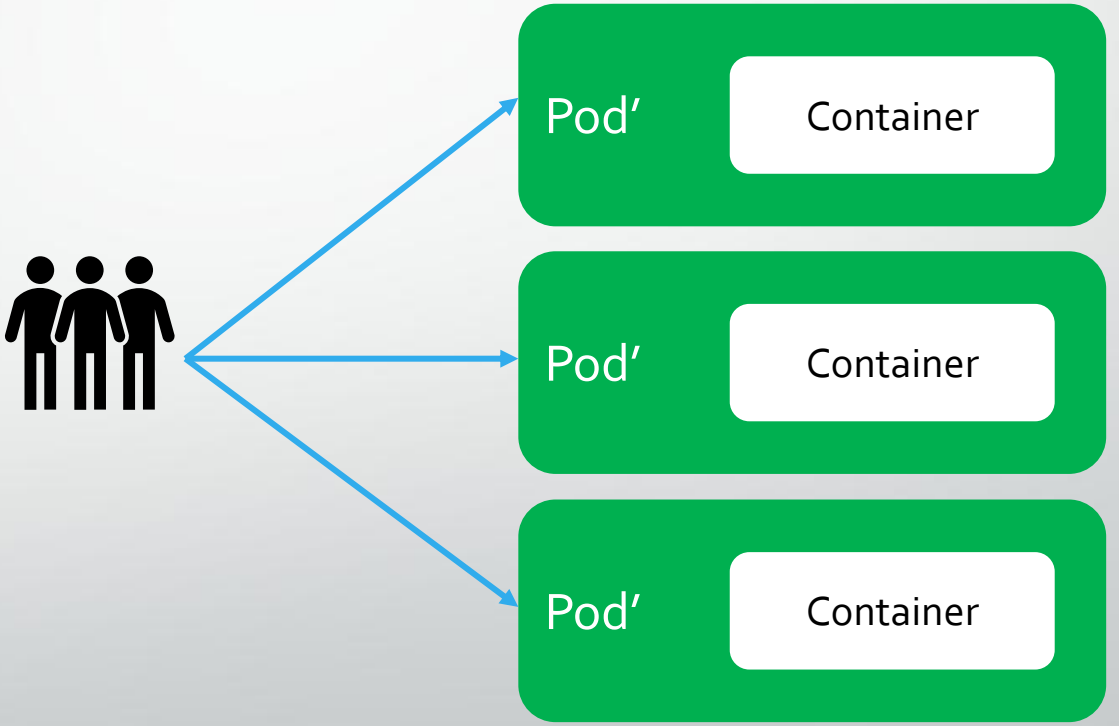
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

Replace old pods with new ones until all are replaced

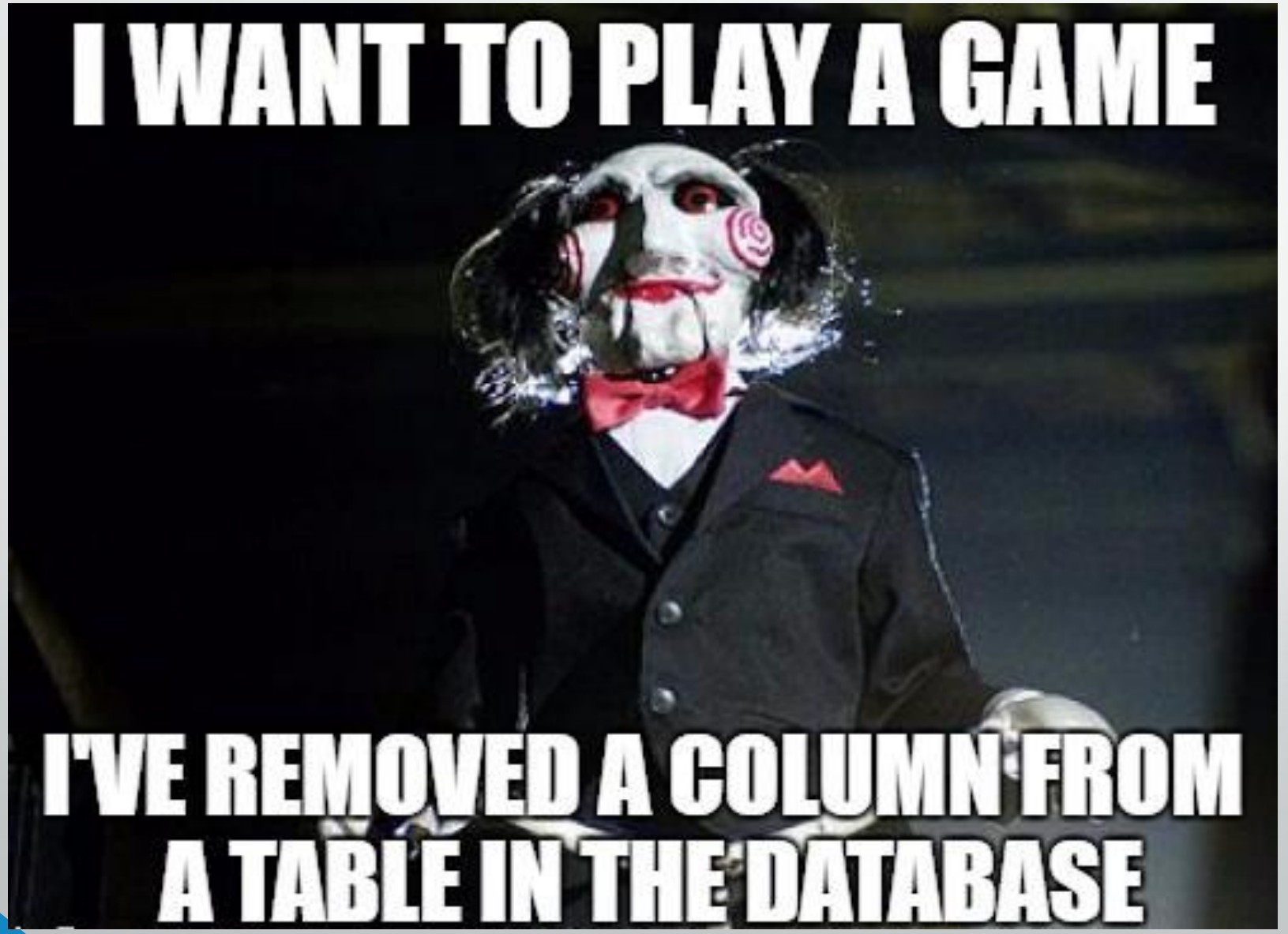


Deployment Strategies

Rolling deployment

- Default deployment mode
- Application must support two versions

Deployment Strategies



Deployment Strategies

Canary deployment

- Route only a small percentage of traffic to the new version
- Reduce the blast radius of a bad deployment
- A/B testing

Zero Downtime Deployment Demo

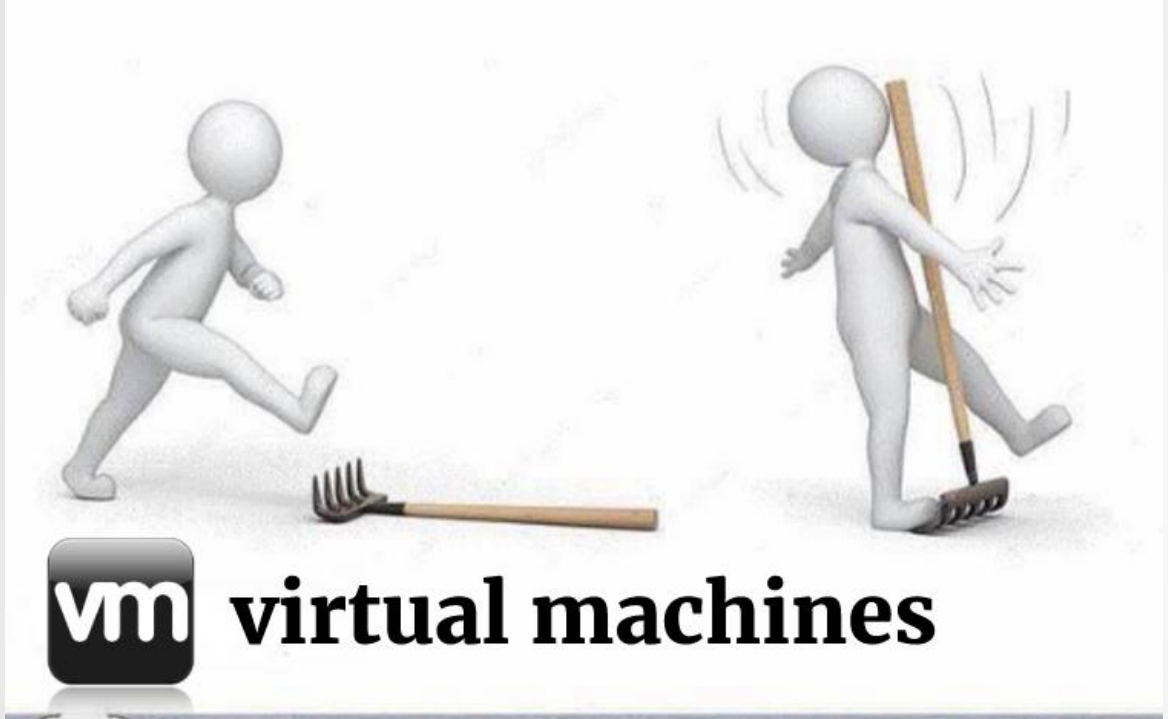
Deployment Strategies

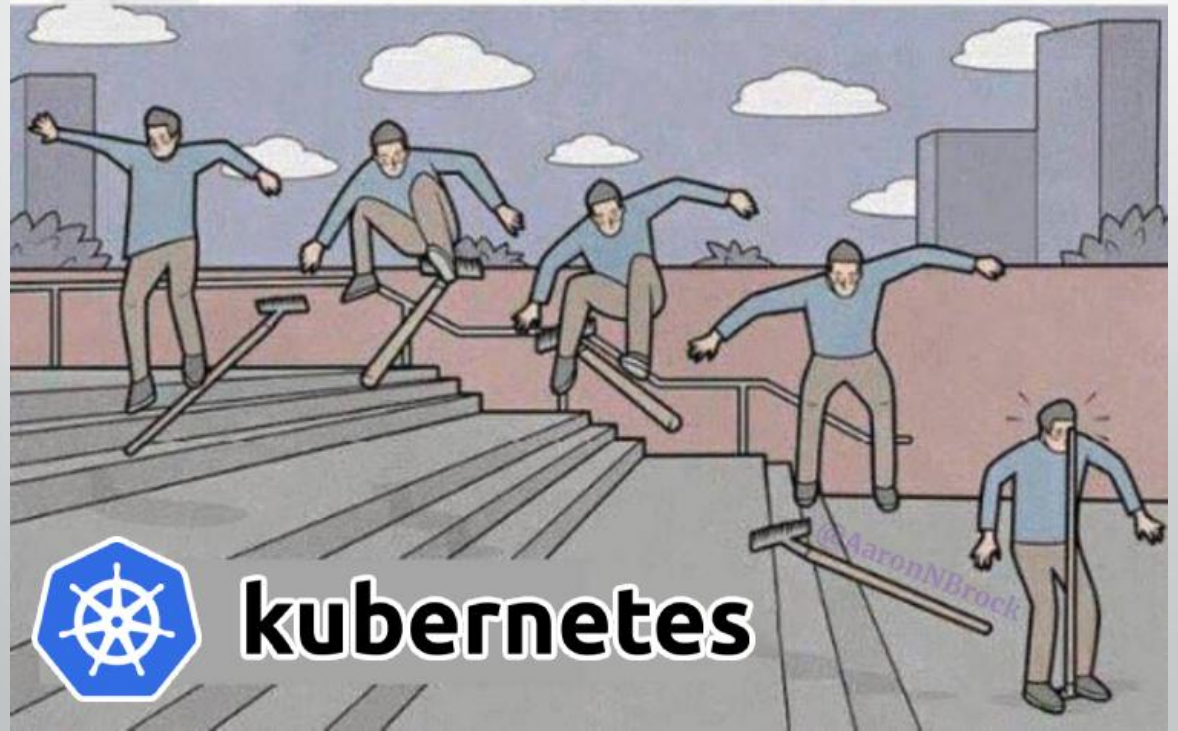
maxSurge

- Max number of pods that can be created at a time
- Absolut number or percentage
- Default: 25%

maxUnavailable

- Max number of pods that can be available during the deployment
- Absolut number or percentage
- Default: 25%


virtual machines


kubernetes

Considerations when using K8s

Cloud-native architecture

Microservices

.NET Full Frameworks vs .NET (Core)

DevOps process and culture

Deploy fast and often

Fast paced development and deployment

Cloud hosted vs. on-premises

When not to use Kubernetes

Skills and experience of the team

Application that will be barely change

Big monolithic applications

Quick results

Very simple applications

Cloud hosted vs. on-premises


@AaronNBrock
kubernetes
My Static Website

Kubectl Commands

Get resource

kubectl get pods/service/deployment

Delete resource

kubectl delete pod/service/deployment

Display information about resource

kubectl desrcibe pod/node/service resource-name

Add/update new resource

kubectl apply –f myfile.yaml [namespace=my-namespace]

Set current namespace

kubectl configt set-context --current --namespace=my-namespace

Kubernetes Cheat Sheet:
https://kubernetes.io/docs/reference/kubectl/cheatsheet

Exercise

Exercise

Play around with Kubernetes

- Connect to the cluster
- See what components are available using the CLI and the dashboard
- Update an existing application
- Change the ports in the Service
- Implement an HPA and test autoscaling
- Add a Readiness and Liveness Probe

Useful links

- Kubernetes Documentation
- Demo Application CustomerApi
- Microservice Series - From Zero to Hero
- Phippy Goes To The Zoo – PDF
- Phippy Goes To The Zoo - Video

Helm

Complex Configuration

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    meta.helm.sh/release-name: customerapi-customerapi-test
    meta.helm.sh/release-namespace: customerapi-test
  creationTimestamp: "2021-11-01T15:10:45Z"
  generation: 4
  labels:
    app: customerapi
    app.kubernetes.io/managed-by: Helm
    chart: customerapi-0.1.126
    draft: draft-app
    heritage: Helm
    release: customerapi-customerapi-test
    manager: kube-controller-manager
    operation: Update
    time: "2021-11-01T15:13:51Z"
  name: customerapi
  namespace: customerapi-test
  resourceVersion: "19234"
  uid: f45511b7-7599-41fb-a129-4973a5926ca2
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 0
  selector:
    matchLabels:
      app: customerapi
      release: customerapi-customerapi-test
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      annotations:
        buildID: ""
      creationTimestamp: null
      labels:
        app: customerapi
        draft: draft-app
        release: customerapi-customerapi-test
    spec:
      containers:
      - env:
        - name: AzureServiceBus__ConnectionString
          valueFrom:
            secretKeyRef:
              key: AzureServiceBus__ConnectionString
              name: customerapi-connectionstrings
```

Helm

Packet Manager for Kubernetes

Helps to manage Kubernetes applications

Template Engine

Bundle of YAML files is called Helm charts

Helm charts describe applications

Simple sharing of Helm charts via ArtifactHub.io

Helm

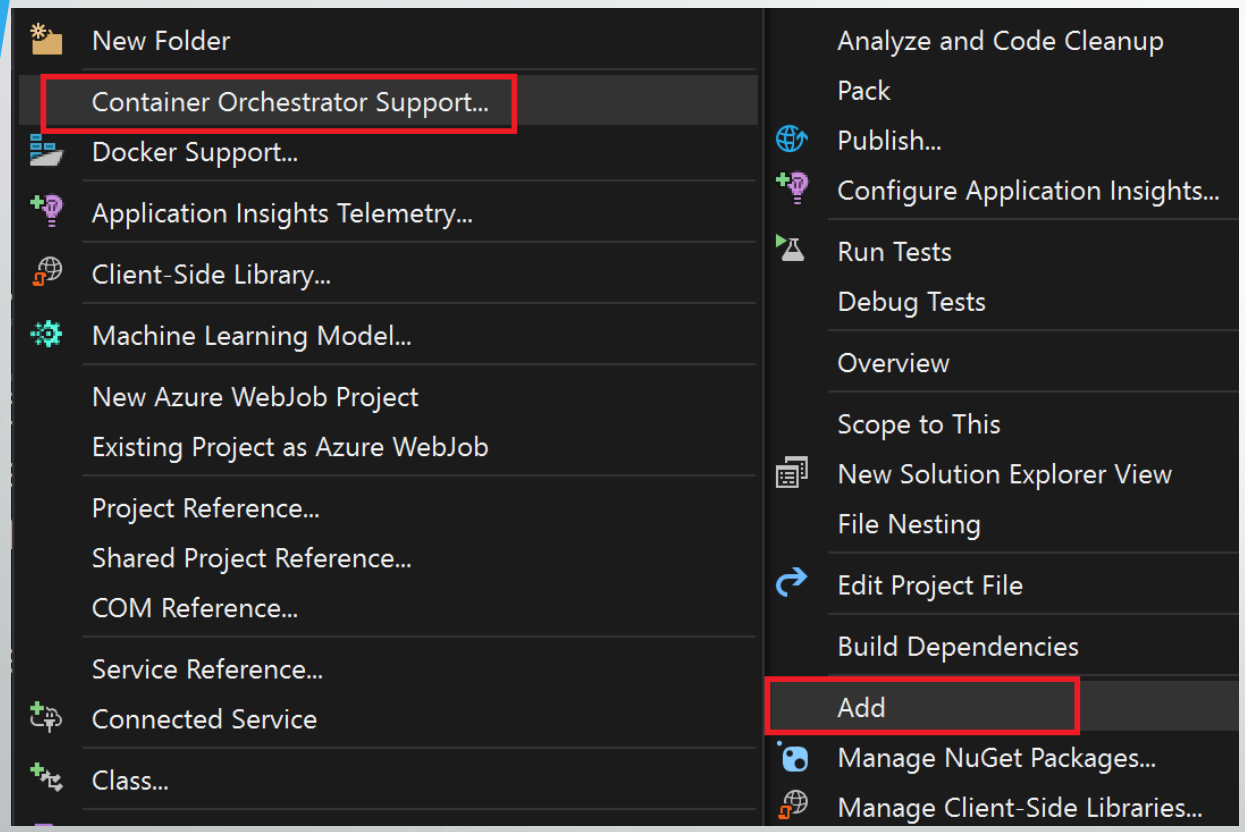
Install Helm

- Linux: curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
- Windows: winget install Helm.Helm
- Mac: brew install helm

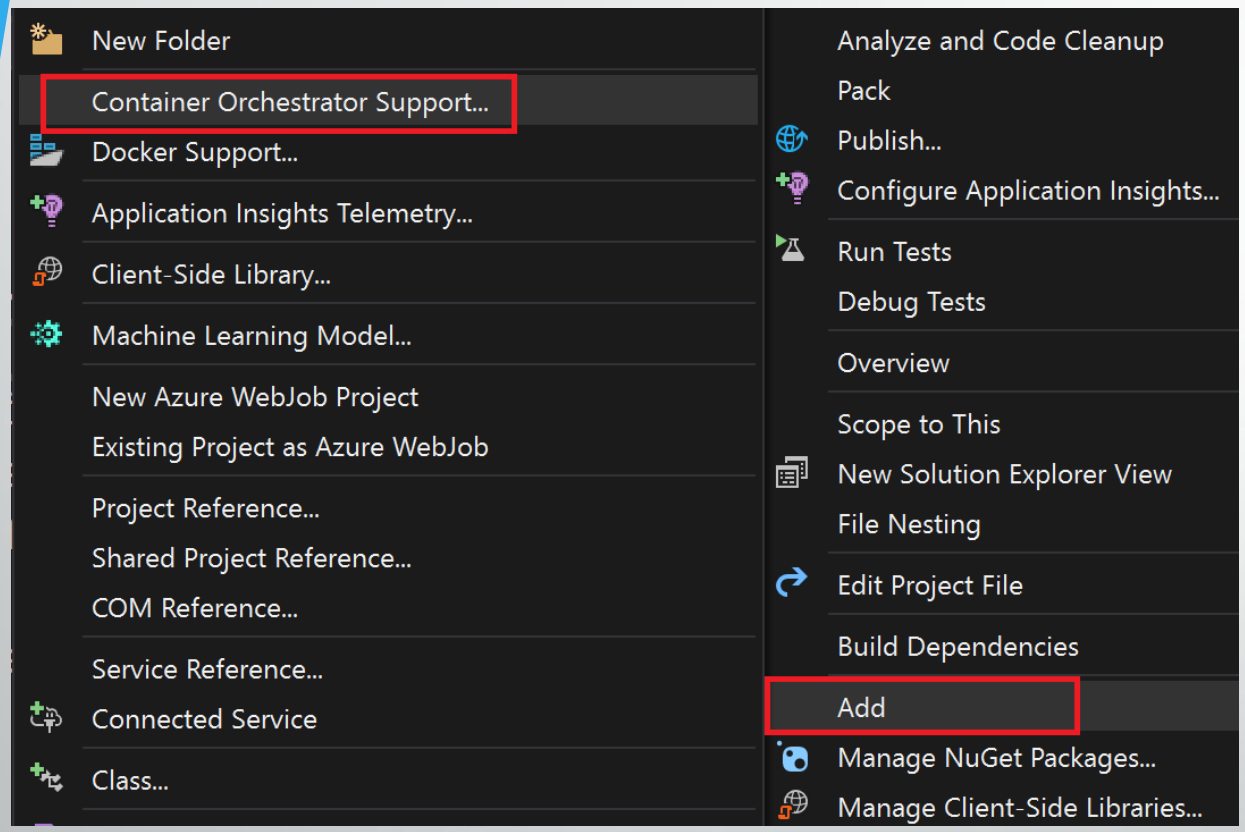
Helm Charts

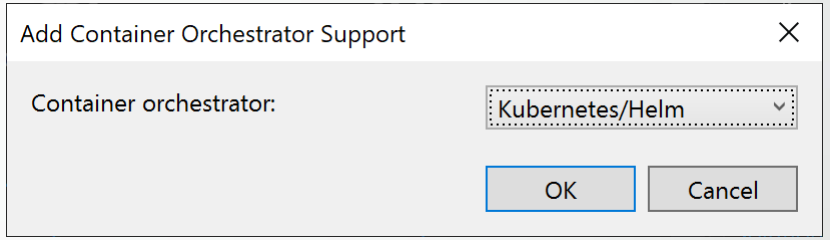
chartname/	
Chart.yaml	A YAML file containing information about the chart
LICENSE	OPTIONAL: A plain text file containing the license for the chart
README.md	OPTIONAL: A human-readable README file
values.yaml	The default configuration values for this chart
charts/	A directory containing any charts upon which this chart depends.
templates/	A directory of templates that, when combined with values, will generate valid Kubernetes manifest files.
templates/NOTES.txt	OPTIONAL: A plain text file containing short usage notes

Add Helm Charts in Visual Studio

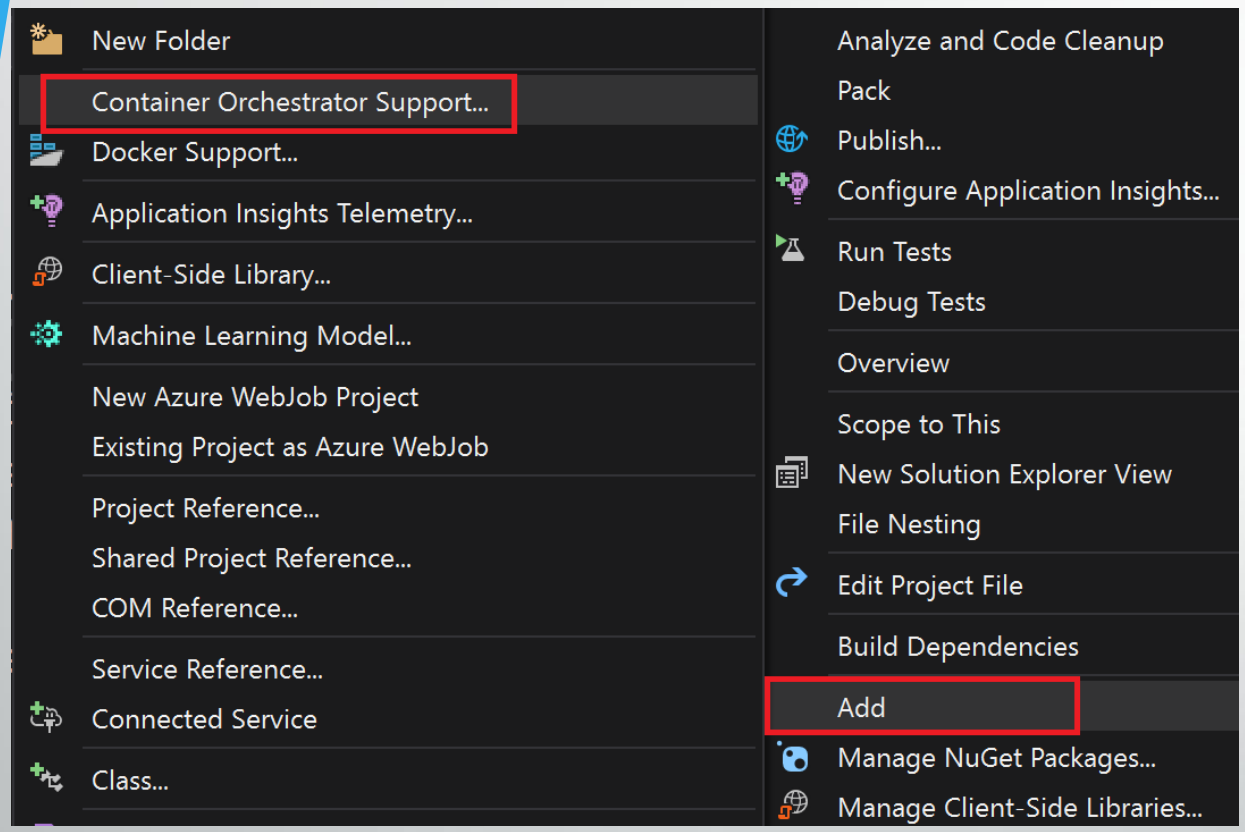


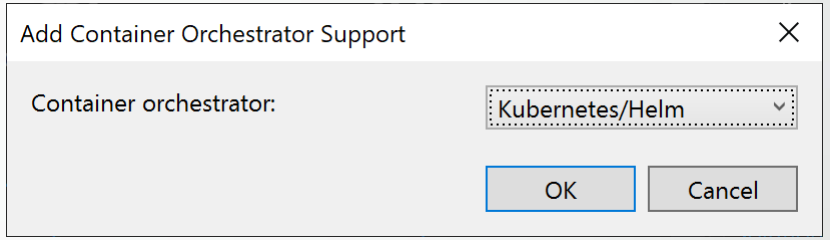
Add Helm Charts in Visual Studio

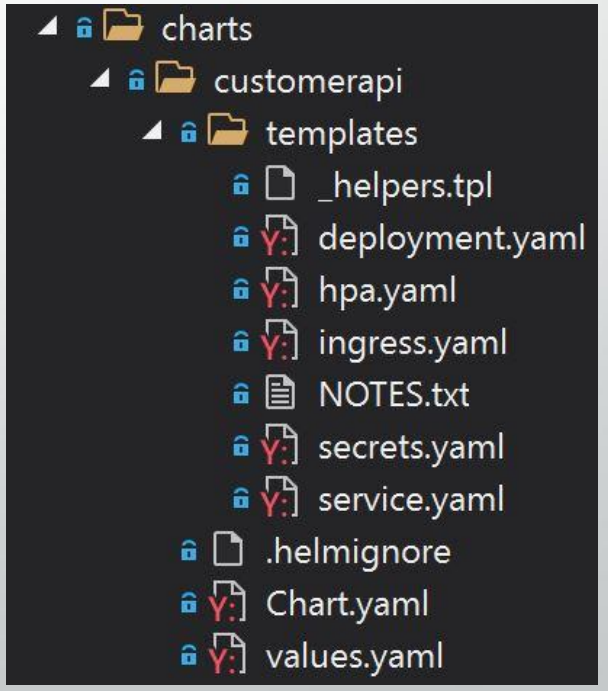




Add Helm Charts in Visual Studio







Add Helm Charts with CLI

mkdir charts

cd charts

helm create <ChartName>

```yaml
template:
  metadata:
    annotations:
      buildID: ""
    creationTimestamp: null
    labels:
      app: customerapi
      draft: draft-app
      release: customerapi-customerapi-test
  spec:
    containers:
    - env:
      - name: AzureServiceBus__ConnectionString
        valueFrom:
          secretKeyRef:
            key: AzureServiceBus__ConnectionString
            name: customerapi-connectionstrings
      - name: ConnectionStrings__CustomerDatabase
        valueFrom:
          secretKeyRef:
            key: ConnectionStrings__CustomerDatabase
            name: customerapi-connectionstrings
      image: wolfgangofner/customerapi:0.1.402
      imagePullPolicy: IfNotPresent
      livenessProbe:
        failureThreshold: 3
        httpGet:
          path: /health
          port: http
          scheme: HTTP
        initialDelaySeconds: 15
        periodSeconds: 10
        successThreshold: 1
        timeoutSeconds: 1
      name: customerapi
      ports:
      - containerPort: 80
        name: http
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ template "customerapi.fullname" . }}
  labels:
    app: {{ template "customerapi.name" . }}
    chart: {{ template "customerapi.chart" . }}
    draft: {{ .Values.draft | default "draft-app" }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
spec:
  revisionHistoryLimit: 0
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: {{ template "customerapi.name" . }}
      release: {{ .Release.Name }}
  template:
    metadata:
      labels:
        app: {{ template "customerapi.name" . }}
        draft: {{ .Values.draft | default "draft-app" }}
        release: {{ .Release.Name }}
      annotations:
        buildID: {{ .Values.buildID | default "" | quote }}
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
          imagePullPolicy: {{ .Values.image.pullPolicy }}
          ports:
            - name: http
              containerPort: {{ .Values.deployment.containerPort }}
              protocol: TCP
          {{- if .Values.probes.enabled }}
          livenessProbe:
            httpGet:
              path: /health
              port: http
            initialDelaySeconds: 15
```

Values.yaml

```yaml
fullnameOverride: customerapi
replicaCount: 1
image:
  repository: __Repository__
  tag: __BuildNumber__
  pullPolicy: IfNotPresent
imagePullSecrets: []
service:
  type: LoadBalancer
  port: 80

deployment:
  containerPort: 80

probes:
  enabled: false
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ template "customerapi.fullname" . }}
  labels:
    app: {{ template "customerapi.name" . }}
    chart: {{ template "customerapi.chart" . }}
    draft: {{ .Values.draft | default "draft-app" }}
    release: {{ .Release.Name }}
    heritage: {{ .Release.Service }}
spec:
  revisionHistoryLimit: 0
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: {{ template "customerapi.name" . }}
      release: {{ .Release.Name }}
  template:
    metadata:
      labels:
        app: {{ template "customerapi.name" . }}
        draft: {{ .Values.draft | default "draft-app" }}
        release: {{ .Release.Name }}
      annotations:
        buildID: {{ .Values.buildID | default "" | quote }}
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
          imagePullPolicy: {{ .Values.image.pullPolicy }}
          ports:
            - name: http
              containerPort: {{ .Values.deployment.containerPort }}
              protocol: TCP
          {{- if .Values.probes.enabled }}
          livenessProbe:
            httpGet:
              path: /health
              port: http
            initialDelaySeconds: 15
```

Override Values in CI/CD Pipeline

```yaml
fullnameOverride: customerapi
replicaCount: 1
image:
  repository: __Repository__
  tag: __BuildNumber__
  pullPolicy: IfNotPresent
imagePullSecrets: []
service:
  type: LoadBalancer
  port: 80

deployment:
  containerPort: 80

probes:
  enabled: false
```

```yaml
variables:
  ApiName: 'customerapi'
  BuildNumber: $(GitVersion.FullSemVer)
  Repository: 'wolfgangofnerbbv/$(ApiName)'
```

```yaml
steps:
  - task: Tokenizer@0
    displayName: 'Run Tokenizer'
```

Helm Commands

List all deployments

helm ls

Install Helm Chart

helm install my-release-name my-helm-chart-name --namespace my-namespace

helm install customer customerapi

Update Release

helm upgrade customer customerapi

Uninstall Release

helm uninstall customer

Rollback Release

helm rollback customer

Release Management

Install or upgrade Charts

Helm will only update components that have changed since the last release

Release Management

Install or upgrade Charts

Helm will only update components that have changed since the last release

```
PS C:\Users\Wolfgang> helm ls --all-namespaces
NAME                            NAMESPACE               REVISION        UPDATED                                 STATUS CHART                            APP VERSION
cert-manager                    cert-manager            1               2021-10-17 12:00:51.12066323 +0000 UTC  deployecert-manager-v1.5.4              v1.5.4
customerapi-customerapi-test    customerapi-test        1               2021-10-17 12:08:31.037732341 +0000 UTC deployecustomerapi-0.1.402              1.0
ingress-nginx                   ingress-basic           1               2021-10-17 12:01:25.840929561 +0000 UTC deployeingress-nginx-4.0.6              1.0.4
keda                            keda                    1               2021-10-17 12:04:21.520467691 +0000 UTC deployekeda-2.4.0                       2.4.0
kedademoapi-kedademoapi-prod    kedademoapi-prod        1               2021-10-17 12:09:50.399832095 +0000 UTC deployekedademoapi-0.1.417              1.0
kedademoapi-kedademoapi-test    kedademoapi-test        1               2021-10-17 12:06:59.357610554 +0000 UTC deployekedademoapi-0.1.417              1.0
loki                            loki-grafana            1               2021-10-17 12:02:10.25707972 +0000 UTC  deployeloki-stack-2.0.3                 v2.0.0
orderapi-orderapi-test          orderapi-test           1               2021-10-17 12:08:05.018913511 +0000 UTC failed orderapi-0.1.421                 1.0
```

Helm Demo

Exercise

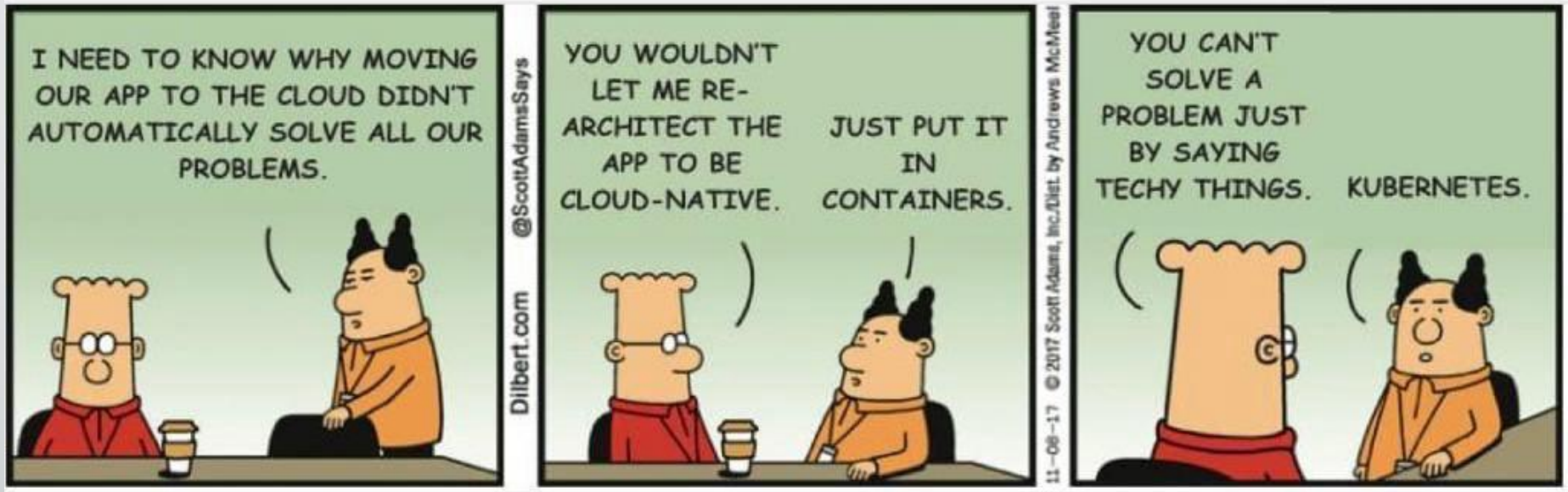
Play around with Helm

- Create a Helm chart
- Install a public Helm chart in your K8s cluster
- Deploy an application to K8s using your Helm chart
- Update a Helm deployment

Useful links:

- Helm Documentation
- Demo Application CustomerApi
- Microservice Series - From Zero to Hero

Solved all your problems. You're welcome.



Authentication and Authorization in Kubernetes

User, Service Account, Group

User

- Account for humans
- Cluster wide

Service Account

- Account for technical users
- Scoped at namespace level

Group

- Contains users OR service accounts

Roles and RoleBinding

Roles

- Represent user permissions
- Permissions within a namespace

RoleBinding

- Assign roles to users
- Assignments are for given namespace

ClusterRoles and ClusterRoleBinding

ClusterRoles

- Represent user permissions
- Permissions span entire cluster

ClusterRoleBinding

- Assign roles to users
- Assignments span the entire cluster

AKS Authentication Options

Local accounts with Kubernetes RBAC

Microsoft Entra ID Authentication with Kubernetes RBAC

Microsoft Entra ID Authentication with Azure RBAC

Create Kubernetes cluster

Automatic upgrade scheduler

Every week on Sunday (recommended)

Start on: Sat May 10 2025 00:00 +00:00 (Coordinated Universal Time)

Edit schedule

Node security channel type

Security channel scheduler

Local accounts with Kubernetes RBAC
Use built-in Kubernetes role-based access control for authorization checks on the cluster.

Microsoft Entra ID authentication with Kubernetes RBAC
Use Microsoft Entra ID for authentication and Kubernetes native RBAC for authorization.

Microsoft Entra ID authentication with Azure RBAC
Use Azure role assignments for authorization checks on the cluster.

Choose between local accounts or Microsof
authorization needs.

Authentication and Authorization

Local accounts with Kubernetes RBAC

Once the cluster is deployed, use the Kubernetes CLI to manage RBAC configurations. Learn more

Previous | Next | Review + create

Give feedback

Local Accounts with Kubernetes RBAC

Default authentication mode for AKS

No link between Microsoft Entra and AKS

Use K8s build-in authentication

Token is stored unencrypted in .kube config file

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUU2RENDQXRDZ0
    server: https://azurecloudnative-aks-dns-537mjg6w.hcp.canadacentral.azmk8s.io:443
  name: AzureCloudNative-aks
contexts:
- context:
    cluster: AzureCloudNative-aks
    user: clusterUser_AzureCloudNative-rg_AzureCloudNative-aks
  name: AzureCloudNative-aks
current-context: AzureCloudNative-aks
kind: Config
preferences: {}
users:
- name: clusterUser_AzureCloudNative-rg_AzureCloudNative-aks
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZIakNDQXdoZ0F3SU
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlKSndJQkFBS0NBZ0VBNT
```

Local Accounts with Kubernetes RBAC

Only recommended when non of the users are in Entra

User management can become very challenging

Local accounts should be disabled for better security

Local Accounts with Kubernetes RBAC Demo

Entra ID Authentication with Kubernetes RBAC

Authentication via Microsoft Entra

Authorization via Kubernetes RBAC

Entra ID Authentication with Kubernetes RBAC

Authentication via Microsoft Entra

Authorization via Kubernetes RBAC

Admin creates role bindings between K8s role and Entra user or group

Entra user or group needs “Azure Kubernetes Cluster User Role” role to download credentials

```yaml
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader
  namespace: read
rules:
- apiGroups: [""]
  resources: ["pods", "services", "endpoints", "persistentvolumeclaims",
  verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
  resources: ["deployments", "daemonsets", "replicasets", "statefulsets"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["batch"]
  resources: ["jobs", "cronjobs"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
  resources: ["ingresses"]
```

```yaml
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: reader-user-binding
  namespace: read
subjects:
  - kind: Group
    name: 24975d09-19e9-47a5-aa3b-e952c693c016 # Entra ID
    namespace: read
roleRef:
  kind: Role # or ClusterRole
  name: reader
  apiGroup: rbac.authorization.k8s.io
```

Entra ID Authentication with Kubernetes RBAC

```
PS C:\Demo> kubectl get all -n read
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx     1/1     Running   0          4m32s
Error from server (Forbidden): replicationcontrollers is forbidden
: User "demo.user@programmingwithwolfgang.com" cannot list resourc
e "replicationcontrollers" in API group "" in the namespace "read"
Error from server (Forbidden): horizontalpodautoscalers.autoscalin
g is forbidden: User "demo.user@programmingwithwolfgang.com" canno
t list resource "horizontalpodautoscalers" in API group "autoscali
ng" in the namespace "read"
```

Entra ID Authentication with Kubernetes RBAC

Auditing access to the cluster can be cumbersome

Management with Entra IDs

Access can be given to Entra users and groups

Entra ID Authentication with Kubernetes RBAC Demo

Entra ID Authentication with Azure RBAC

Manage access to the cluster with Azure only

Use Azure RBAC roles to manage permissions inside the cluster

Recommended way to manage AKS clusters

"Azure Kubernetes Service Cluster User Role" to download credentials

Assign built-in or custom roles

Namespace specific permissions via Azure CLI only

Entra ID Authentication with Azure RBAC Demo

Exercise

Create a new cluster with Azure RBAC

Authorize a colleague with any role

Test the access to the cluster/namespace

Use Kubernetes role-based access control with Microsoft Entra ID in Azure Kubernetes Service

Use Azure role-based access control for Kubernetes Authorization

Pod Scheduling

Resource Quotas

Restrict resource usage in a namespace or cluster

- CPU
- RAM
- Object count (pod, secrets, services, etc.)
- Storage

Resources can not be scheduled if they exceed the quota

Resource Quotas Demo

Node Selector

Schedule pod on specific node

Node selection with matching label

```
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  3s    default-scheduler  0/2 nodes are available: 2 node(s) didn't match Pod's node affinity/sele
ctor. preemption: 0/2 nodes are available: 2 Preemption is not helpful for scheduling..
```

Affinity and Anti-Affinity

Affinity and anti-affinity expand the types of constraints

Configure where pods can be scheduled / not scheduled

Pod and node affinity

Rules can be hard or soft

Constrain pod scheduling based on already running pods

Affinity and Anti-Affinity Demo

Taints and Tolerations

Taint has the format key=value:effect → kubectl taint nodes node1 key1=value1:NoSchedule

Remove taint with - at the end → kubectl taint nodes node1 key1=value1:NoSchedule-

Available Taints:

- NoSchedule
- PreferNoSchedule
- NoExecute

Taints and Tolerations

When to use taints and tolerations

- Dedicated nodes
- Special hardware
- Eviction
- Pod separation

Taints and Tolerations

Error message when pod does not satisfy taint

```
Events:
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  16s   default-scheduler  0/2 nodes are available: 2 node(s) had untolerated taint {my
key: myvalue}. preemption: 0/2 nodes are available: 2 Preemption is not helpful for scheduling..
```

Taints and Tolerations Demo

Exercise

Add ResourceQuotas to a namespace

Use affinity and anti-affinity to control pod scheduling

Apply taints and tolerations

Test different variations and get comfortable with each feature

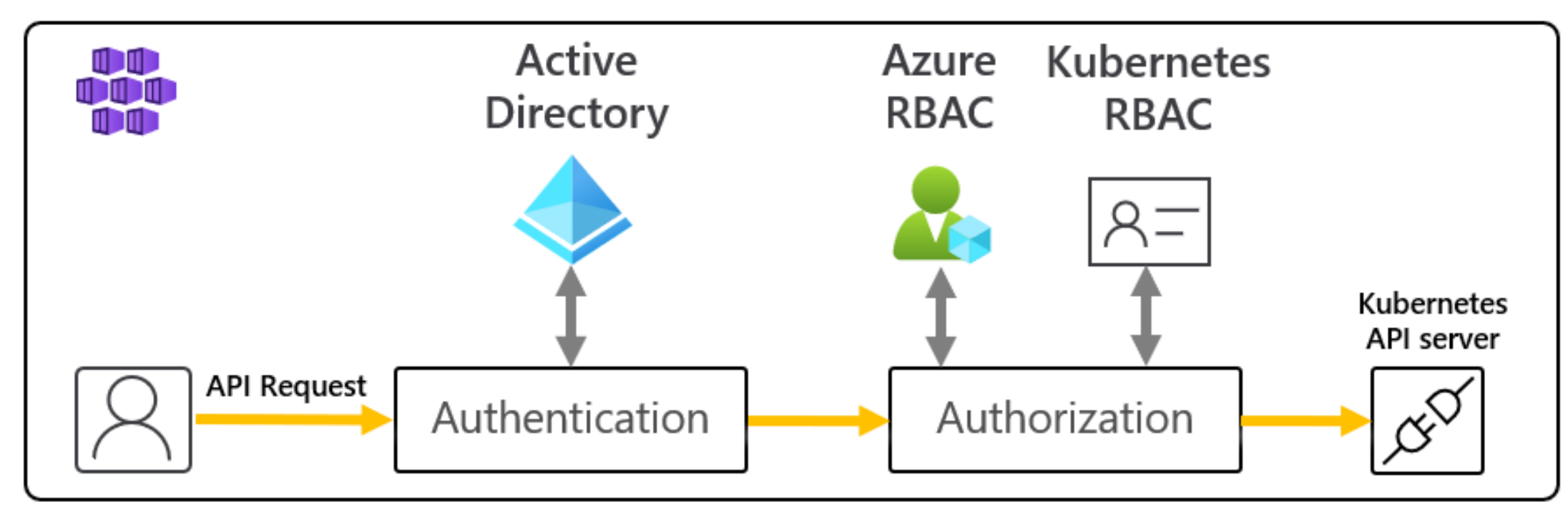
Resource Quotas

Assigning Pods to Nodes

Entra Workload ID with Azure Kubernetes Service

Microsoft Entra Integration with AKS

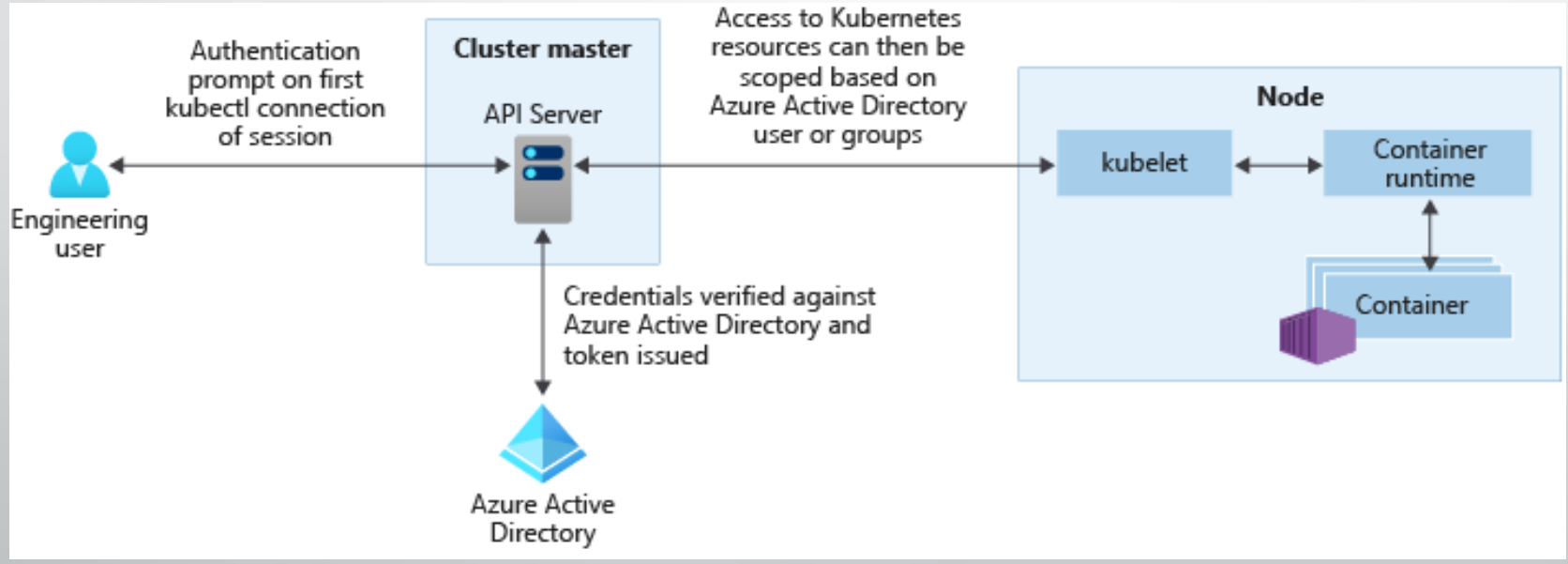
Azure AAD Authentication with Kubernetes or Azure RBAC

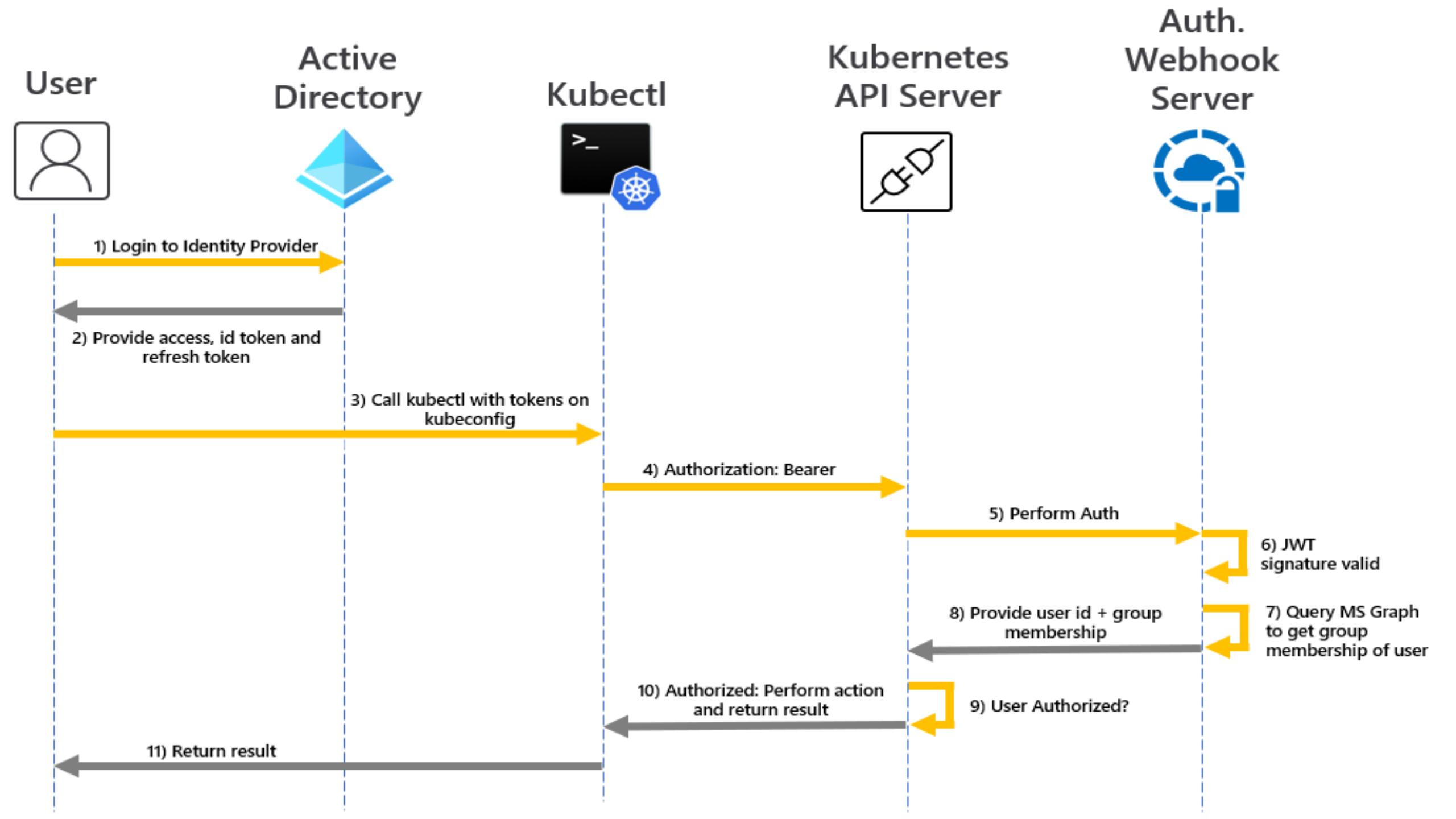


Microsoft Entra Integration with AKS

User runs az aks get-credentials command

User is prompted to sign with their Microsoft Entra credentials after executing the first kubectl command




User
Active Directory
Kubectl
Kubernetes API Server
Auth. Webhook Server
1) Login to Identity Provider
2) Provide access, id token and refresh token
3) Call kubectl with tokens on kubeconfig
4) Authorization: Bearer
5) Perform Auth
6) JWT signature valid
7) Query MS Graph to get group membership of user
8) Provide user id + group membership
9) User Authorized?
10) Authorized: Perform action and return result
11) Return result

Access and Password Management

Let's ditch passwords and go passwordless

- Managed identities are assigned to applications and services
- Identities are given permissions to access other services
- Authentication is based on Microsoft Entra and uses Azure RBAC

Giving an identity to a VM is easy but how do we manage different applications in a Kubernetes cluster?

Entra Workload ID

Giving an identity to the cluster is insufficient since every application might need access to different services

Create a managed identity per application

The managed identity is linked via identity federation to a service account

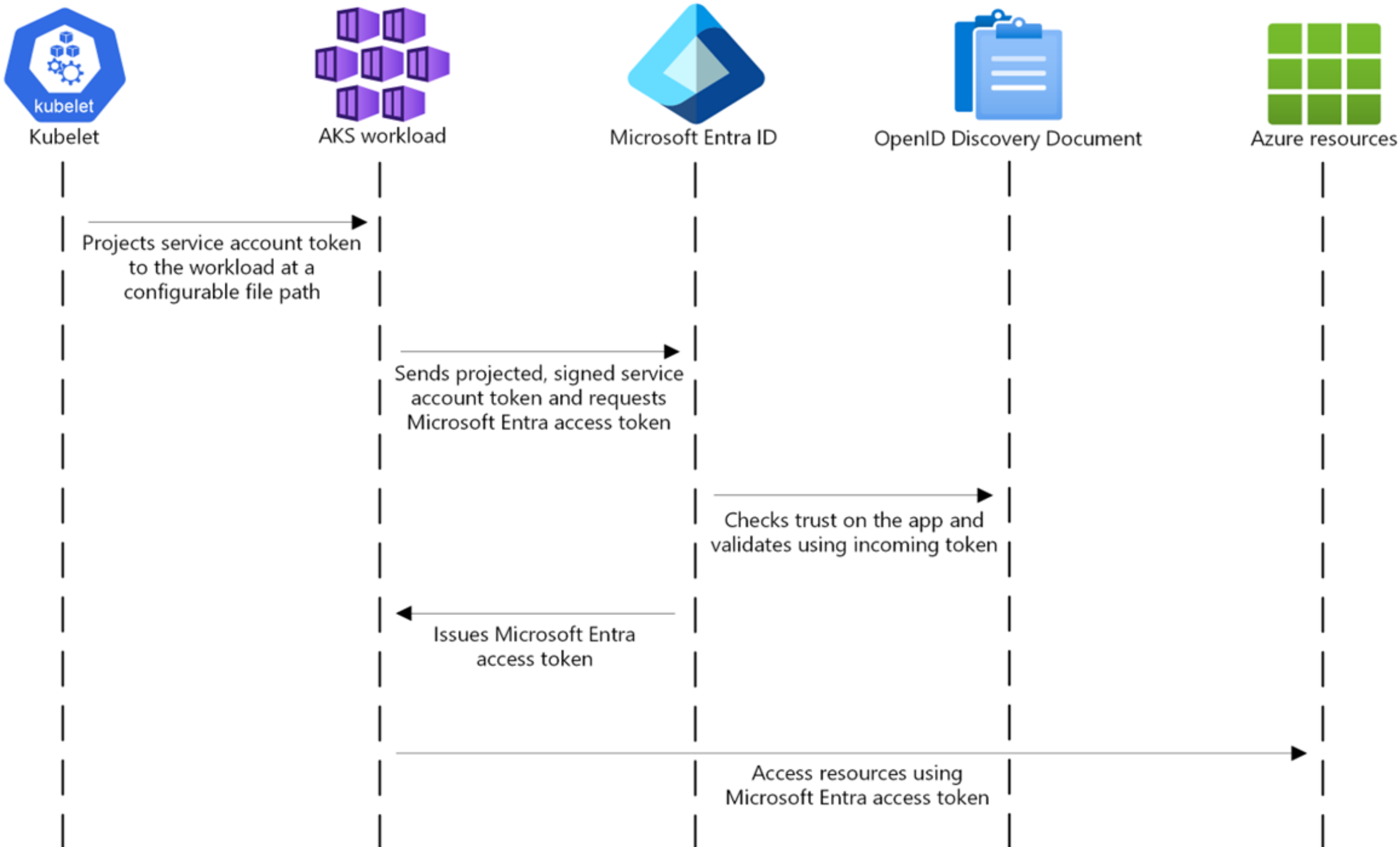
Entra Workload ID

AKS must have Oidc issuer enabled

Label **azure.workload.identity/use: "true"** must be set on the pod

The Service Account references the managed identity ID

The managed identity requests an access token from Entra to access other Azure services


kubelet
Kubelet
AKS workload
Microsoft Entra ID
OpenID Discovery Document
Azure resources
Projects service account token to the workload at a configurable file path
Sends projected, signed service account token and requests Microsoft Entra access token
Checks trust on the app and validates using incoming token
Issues Microsoft Entra access token
Access resources using Microsoft Entra access token

Entra Workload ID Demo

Access Azure Container Registry

Image Pull Secret

Use service principal credentials to access ACR

Manage service principals

Use credentials to access ACR

No granular access control to ACR repositories

Pods need to reference secret to access ACR

Outdated approach and not recommended anymore

Authenticate ACR from AKS

Use AKS identity to access ACR

Attach ACR to AKS during creation or afterwards

Pods don't need to reference a secret

AKS identity gets the **AcrPull** permission assigned

No granular access control to ACR repositories

Image Pull Secret Demo

Access ACR with Repository Token

Token can be scoped at repository level

Granular permissions per repository and token (read, write, etc.)

Password can expire

Token do not support docker push and pull of signed images

Access ACR with Repository Token

Pod needs to reference secret with token

2 passwords for rotation

Recommended way to access ACR if AKS identity is not an option

Repository Token Demo

Exercise

Create an Azure Container Registry

Add an image to the registry

- az acr import -n <ACRNAME> --source docker.io/library/nginx:latest --image nginx

Create a repository token

Deploy an image from your ACR

Create a token with repository-scoped permissions

Authenticate with Azure Container Registry (ACR) from Azure Kubernetes Service (AKS)

Pull images from an Azure container registry to a Kubernetes cluster using a pull secret

Azure Key Vault Provider for Secrets Store CSI Driver

Azure Key Vault

Secrets are not encrypted in Kubernetes

Azure Key Vault features

- Secure management of keys, secrets and certificates
- Access control and audit
- Automated rotation of secrets and certificates
- Ease of use
- Data protection

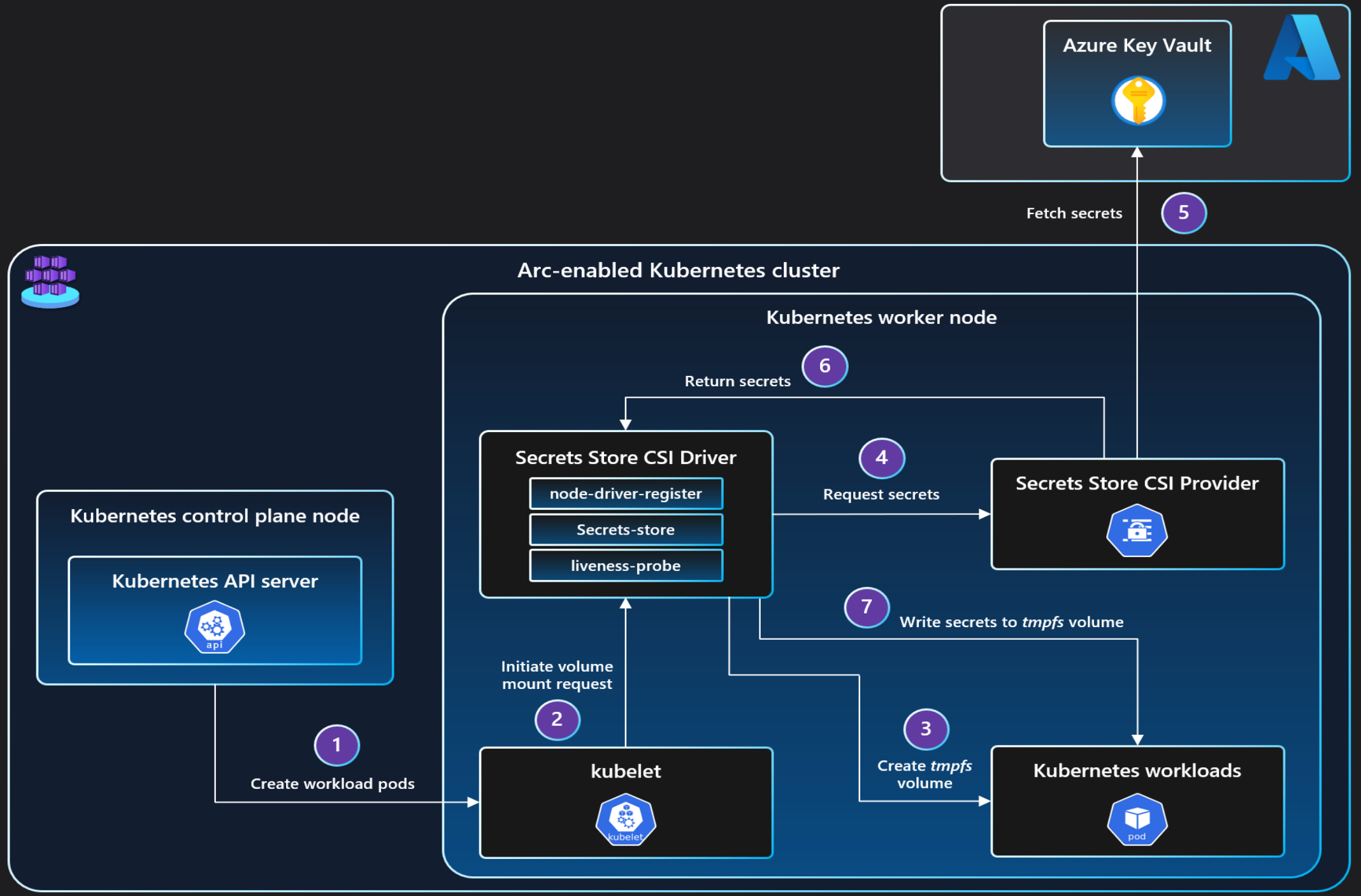
Azure Key Vault Provider for Secrets

Mount secrets, keys, and certificates to pods

Auto-rotate secrets

Sync Azure Key Vault with Kubernetes secrets

Separation of concerns


Azure Key Vault
Fetch secrets
5
Arc-enabled Kubernetes cluster
Kubernetes worker node
6
Return secrets
Secrets Store CSI Driver
node-driver-register
Secrets-store
liveness-probe
4
Request secrets
Secrets Store CSI Provider
Kubernetes control plane node
Kubernetes API server
api
7
Write secrets to tmpfs volume
Initiate volume mount request
2
1
Create workload pods
kubelet
kubelet
3
Create tmpfs volume
Kubernetes workloads
pod

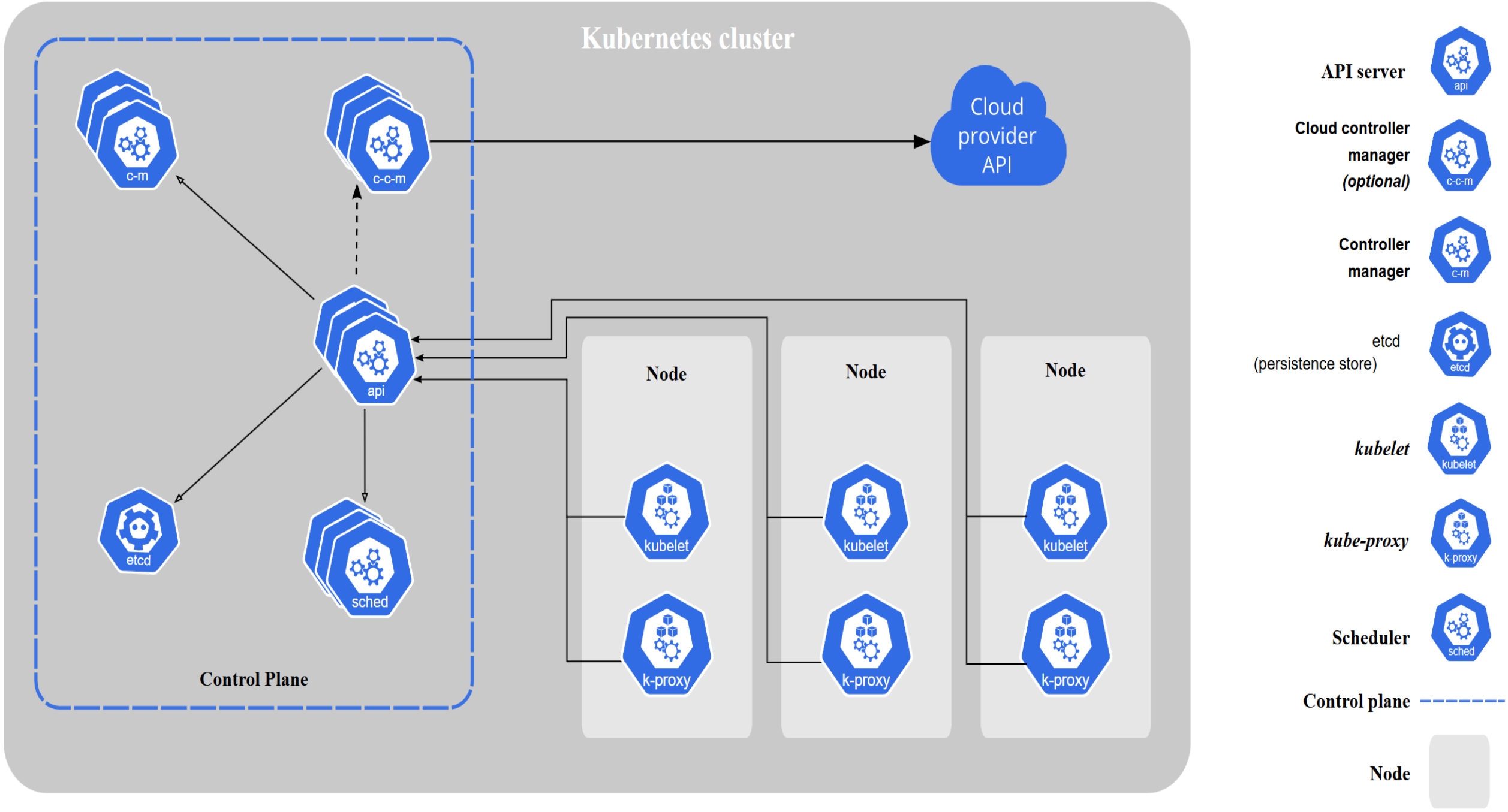
Azure Key Vault Provider for Secrets Store CSI Driver Demo

Private AKS Cluster

Private AKS Cluster

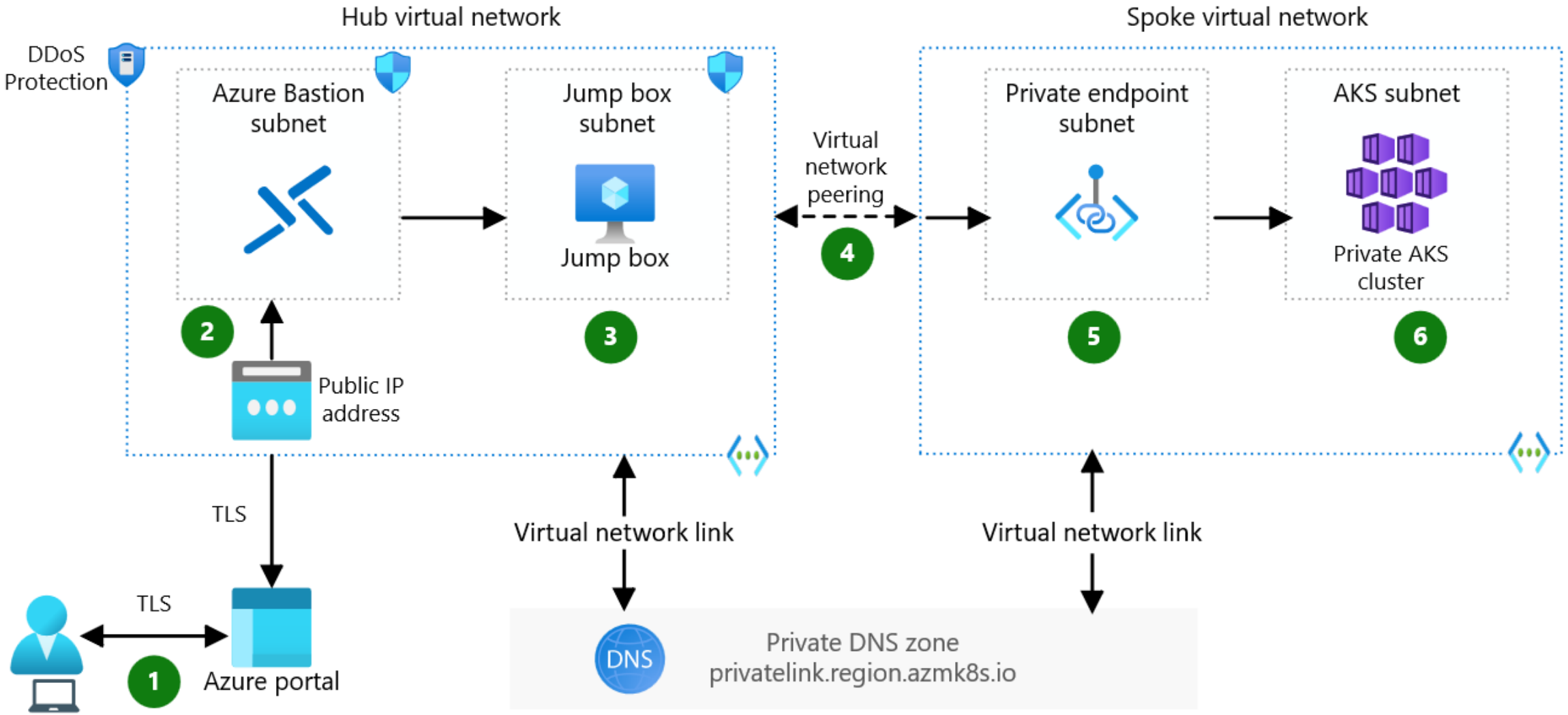
Disable public access to API

Communication between Worker and Master nodes over a private connection


Kubernetes cluster
c-m
c-c-m
Cloud provider API
api
etcd
sched
Control Plane
Node
kubelet
k-proxy
Node
kubelet
k-proxy
Node
kubelet
k-proxy
API server
Cloud controller manager (optional)
Controller manager
etcd (persistence store)
kubelet
kube-proxy
Scheduler
Control plane
Node

Private AKS Cluster

Use Azure Bastion and a Jump box


Hub virtual network
Spoke virtual network
DDoS Protection
Azure Bastion subnet
Jump box subnet
Jump box
Virtual network peering
Private endpoint subnet
AKS subnet
Private AKS cluster
2
3
4
5
6
Public IP address
TLS
TLS
Virtual network link
Virtual network link
DNS
Private DNS zone
privatelink.region.azmk8s.io
1
Azure portal

Private AKS Cluster

Use Azure Bastion and a Jump box

Wrap command with “az aks command invoke”

```
PS C:\Users\Wolfgang> az aks command invoke `
>>          -g AzureCloudeNative-rg `
>>          -n AzureCloudeNative-aks `
>>          --command "kubectl get ns"
```

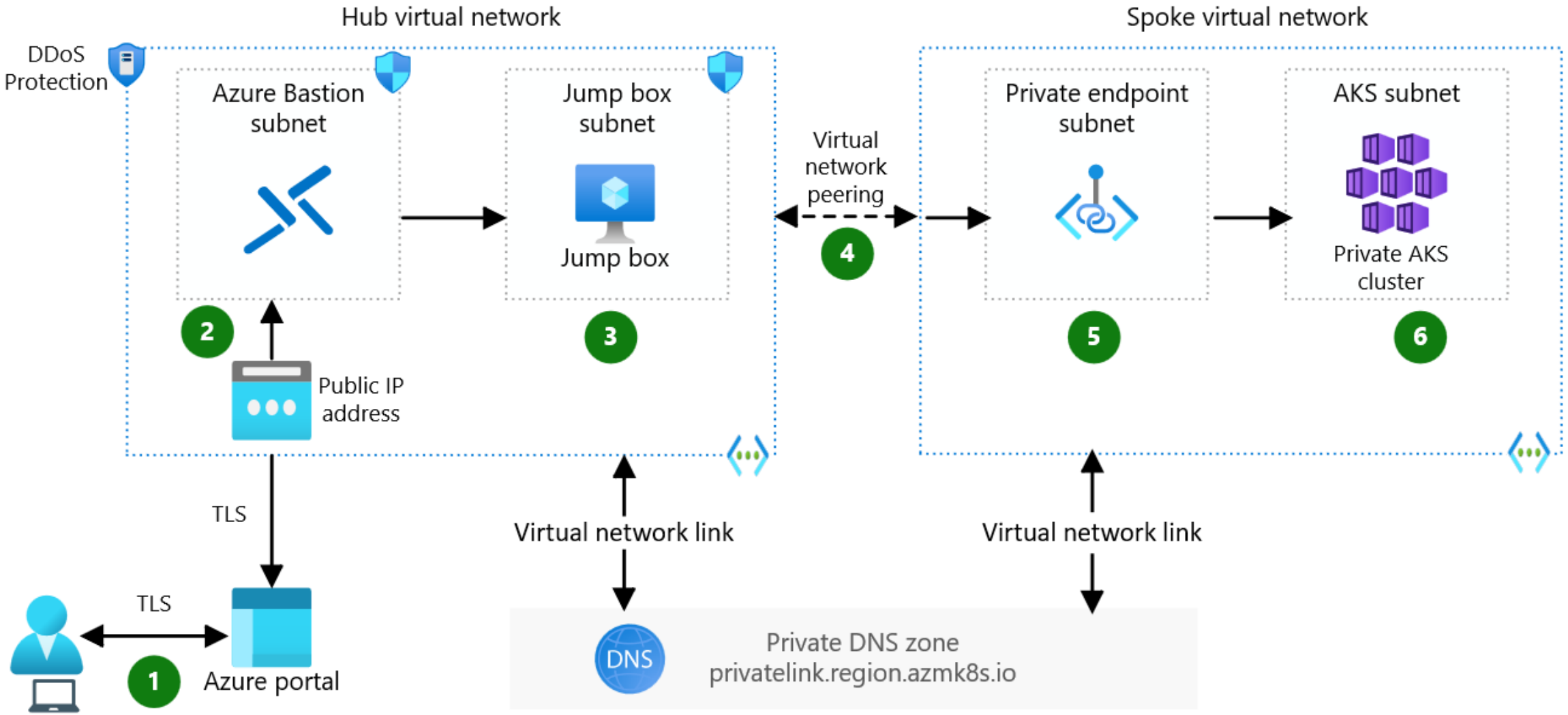
Private AKS Cluster

Use Azure Bastion and a Jump box

Wrap command with “az aks command invoke”

VPN/ExpressRoute connection

New feature to access AKS through Azure Bastion

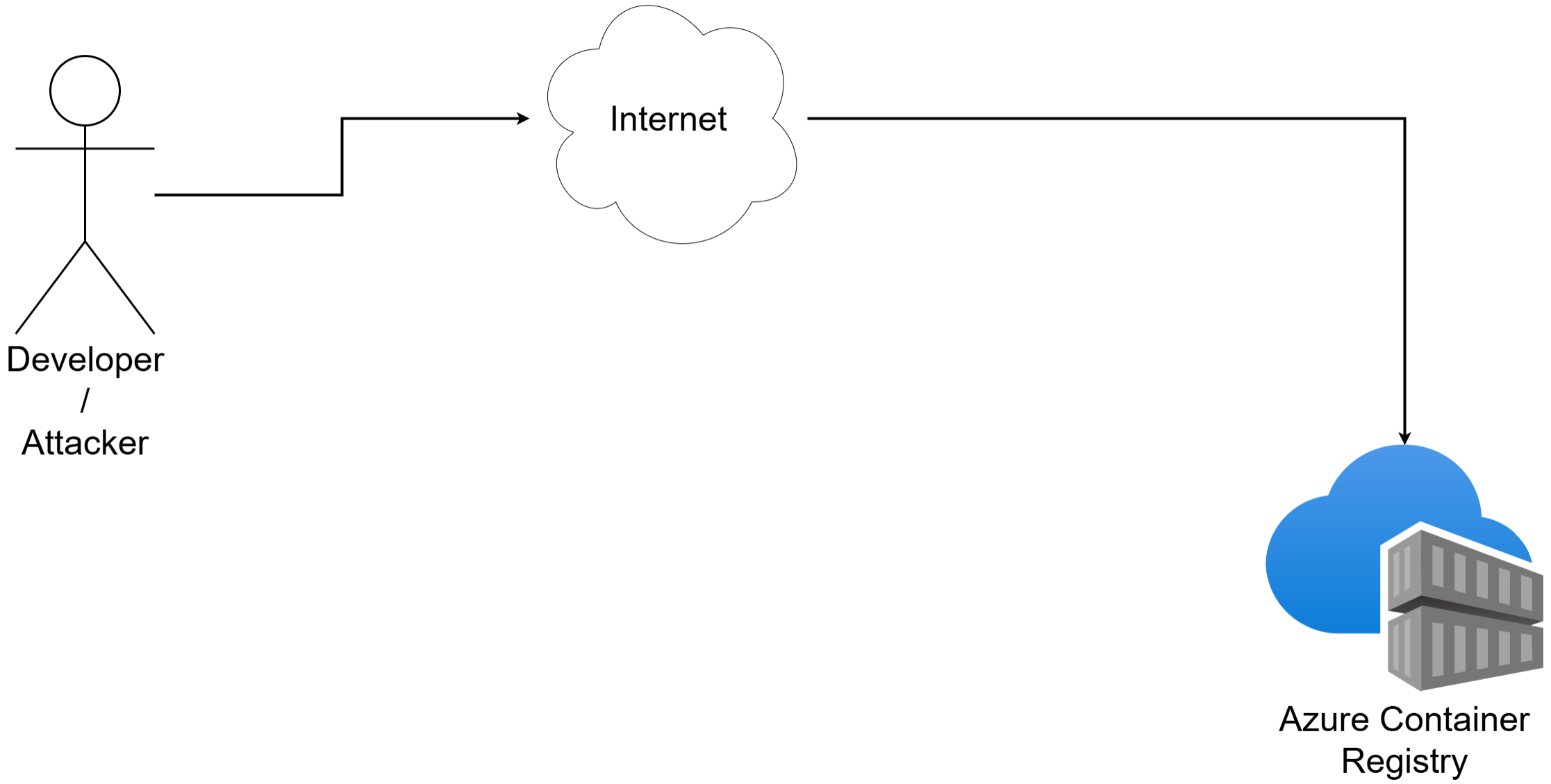

Hub virtual network
Spoke virtual network
DDoS Protection
Azure Bastion subnet
Jump box subnet
Jump box
Virtual network peering
Private endpoint subnet
AKS subnet
Private AKS cluster
2
3
4
5
6
Public IP address
TLS
TLS
Virtual network link
Virtual network link
DNS
Private DNS zone
privatelink.region.azmk8s.io
1
Azure portal

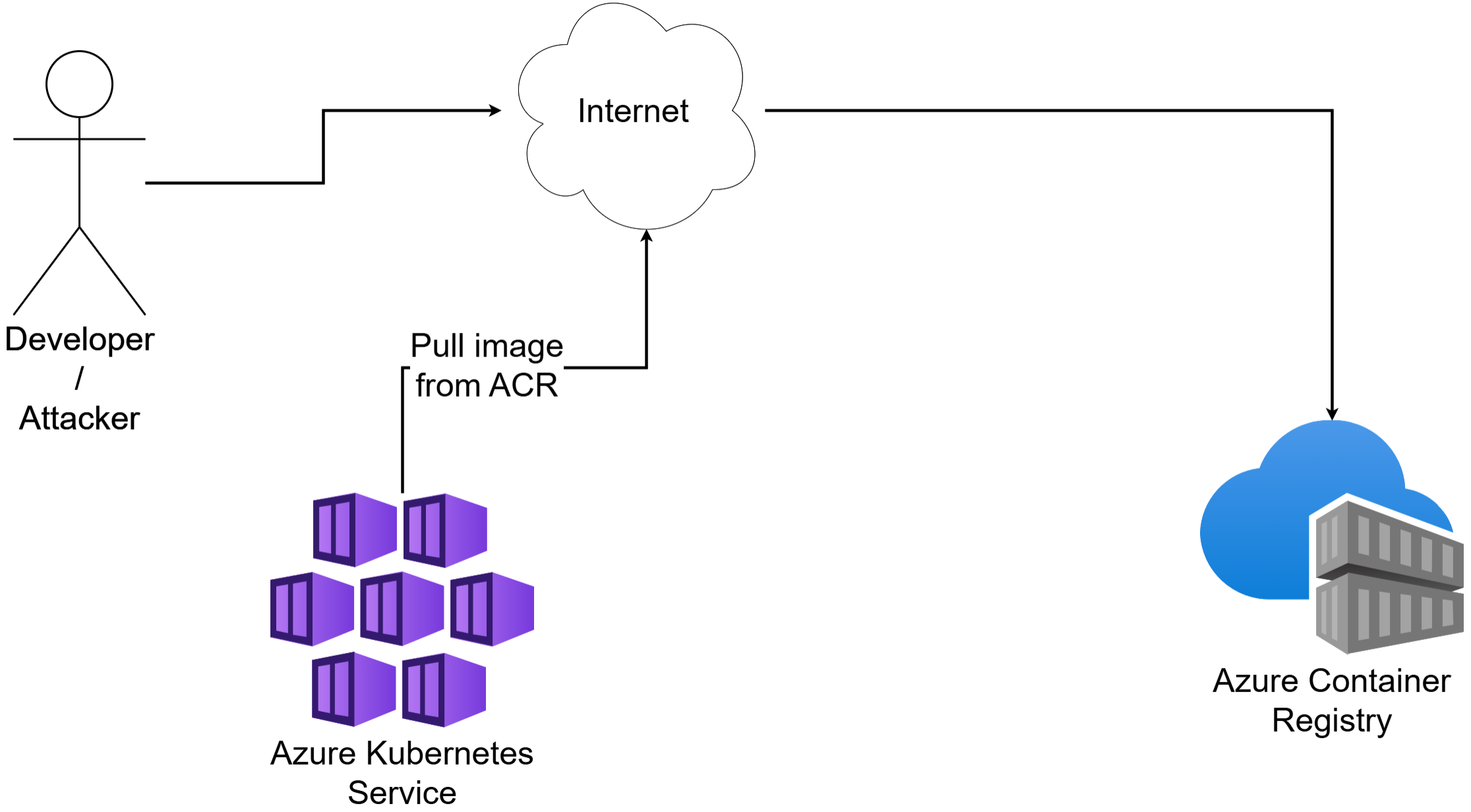
Azure Container Registry

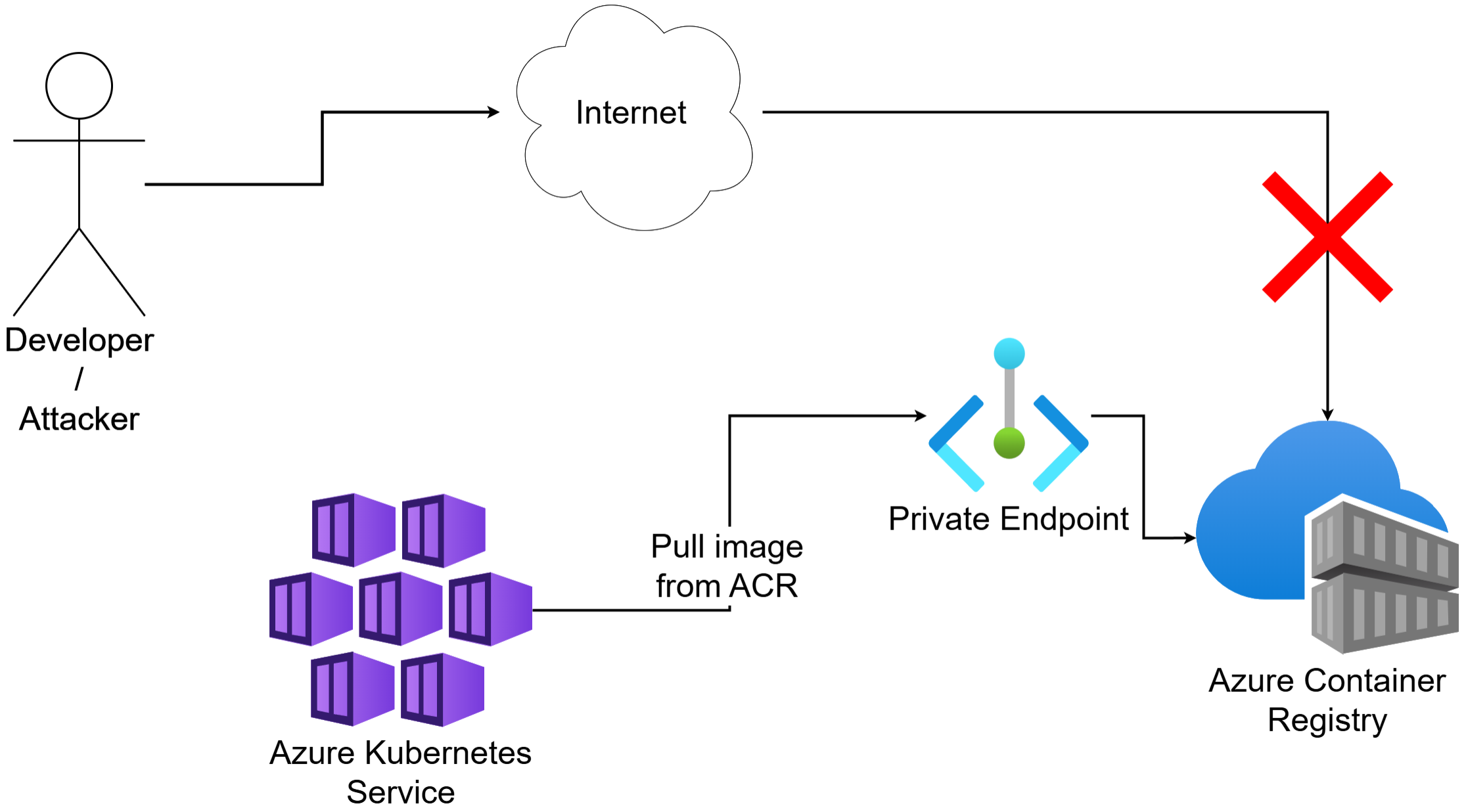
Store images in private registry

Keep everything private

AKS should download images over private connection


Internet
Developer
/
Attacker
Azure Container
Registry


Internet
Developer
/
Attacker
Pull image
from ACR
Azure Kubernetes
Service
Azure Container
Registry


Internet
Developer
/
Attacker
Private Endpoint
Pull image
from ACR
Azure Kubernetes
Service
Azure Container
Registry

Azure Container Registry

No changes for pull operation necessary

Private DNS-Zone resolves public FQDN

Build agent needs to push images over private endpoint

kubernetestrainingwolfgang | Networking

Container registry

Search

Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Settings
Access keys
Encryption
Identity
Networking
Microsoft Defender for Cloud
Properties
Locks

Public access | Private access

Save | Discard | Refresh

Public network access:
- (•) All networks
- ( ) Selected networks
- ( ) Disabled

Data endpoints

Login server: kubernetestrainingwolfgang.azurecr.io

Geo-replications: Configure

Use dedicated data endpoint: [ ]

Location	Data endpoint
Canada Central	*.blob.core.windows.net

kubernetestrainingwolfgang | Networking

Container registry

Search

Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Settings
Access keys
Encryption
Identity
Networking
Microsoft Defender for Cloud
Properties
Locks

Public access | Private access

Save | Discard | Refresh

Public network access:

- ○ All networks
- ○ Selected networks
- ◉ Disabled

Firewall exception

☐ Allow trusted Microsoft services to access this container registry

kubernetestrainingwolfgang | Repositories

Container registry

Search

Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Settings
Access keys
Encryption
Identity
Networking
Microsoft Defender for Cloud
Properties
Locks
Services
Repositories
Webhooks
Geo-replications
Tasks
Connected registries (Preview)

Looks like you don't have access to this content. Are firewalls and virtual networks enabled?

Summary

Session ID
315d015d5828408b99d4e047965f7d27

Resource ID
/subscriptions/e347e896-c1d2-4aea-b63d-2c7f5f6acc7e...

Extension
Microsoft_Azure_ContainerRegistries

Content
RepositoryBrowseBlade

Error code
403

kubernetestrainingwolfgang | Networking

Container registry

Search

Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Settings
Access keys
Encryption
Identity
Networking
Microsoft Defender for Cloud

Public access | Private access

+ Create a private endpoint connection | Approve | Reject | Remove | Refresh

Filter by name | All connection states

Connection name	Connection state	Private endpoint
No results		

Create a private endpoint ...

1 Basics 2 Resource 3 Virtual Network 4 DNS 5 Tags 6 Review + create

Use private endpoints to privately connect to a service or resource. Your private endpoint must be in the same region as your virtual network, but can be in a different region from the private link resource that you are connecting to. Learn more

Project details

Subscription * : Visual Studio Enterprise Subscription

Resource group * : KubernetesDemo

Create new

Instance details

Name * : Acr-privateEndpoint

Network Interface Name * : Acr-privateEndpoint-nic

Region * : Canada Central

Create a private endpoint ...

Basics | Resource | 3 Virtual Network | 4 DNS | 5 Tags | 6 Review + create

Networking

To deploy the private endpoint, select a virtual network subnet. Learn more

Virtual network: aks-vnet-26650984 (MC_KubernetesDemo_privateCluster_canadacentral)

Subnet *: aks-subnet

Network policy for private endpoints: Disabled (edit)

Private IP configuration

- (•) Dynamically allocate IP address
- ( ) Statically allocate IP address

Application security group

Configure network security as a natural extension of an application's structure. ASG allows you to group virtual machines and define network security policies based on those groups. You can specify an application security group as the source or destination in an NSG security rule Learn more

+ Create

Application security group

Create a private endpoint ...

✓ Basics ✓ Resource ✓ Virtual Network 4 DNS 5 Tags 6 Review + create

Private DNS integration

To connect privately with your private endpoint, you need a DNS record. We recommend that you integrate your private endpoint with a private DNS zone. You can also utilize your own DNS servers or create DNS records using the host files on your virtual machines. Learn more

Integrate with private DNS zone (•) Yes ( ) No

Configuration name	Subscription	Resource group	Private DNS zone
privatelink-azurecr-io	Visual Studio Enterpri...	MC_KubernetesDem...	(new) privatelink.azurecr.io

```
PS D:\Daten\Presentations\2024 - Workframe Training - CAA> kubectl apply -f .\podPrivate.yaml
pod/private-registry created
PS D:\Daten\Presentations\2024 - Workframe Training - CAA> kubectl get pods
NAME               READY   STATUS    RESTARTS   AGE
private-registry   1/1     Running   0          41s
```

Security Best Practices

Trusted Access

Azure resources may not be able to access K8s API if AKS is private

- Azure services may not be in the authorized IP range
- Azure services would have to use private endpoint access model
- You do not want to give an Azure services admin access

Trusted Access

Use managed identity to access Kubernetes API server

Assign identity trusted access role

For example, Azure Backup to bypass private endpoint limitation

Azure Security Baseline for AKS

Security baseline provides security recommendations for cloud solutions

Monitor recommendations with Microsoft Defender for Cloud

Azure security baseline for Azure Kubernetes Service (AKS)

Azure Policy

Enforce organizational standards at scale

- Limit CPU and memory resources of pods
- AKS should use managed identities
- Ensure pods have readiness or liveness probes configured

Azure Policy

Azure Policy effect

- Audit
- Deny
- Allow

Use built-in or custom policies

Azure Policy

Azure Policy-addon needs to be installed on AKS cluster

Kubernetes cluster pod security baseline standards for Linux-based workloads

Azure Policy

Policies (5) Groups (0) Parameters (4) JSON Assignments (0)

Filter by reference ID, policy name or ID... Type : **All selected** Evaluation type : **All selected**

Policy	Reference ID	Type	Evaluati...	Default effect
Kubernetes cluster should not allow privileged containers	NoPrivilegedContain...	BuiltIn	Automated	Audit
Kubernetes cluster pods should only use approved host network and port range	BlockUsingHostNetw...	BuiltIn	Automated	Audit
Kubernetes cluster containers should not share host process ID or host IPC namespace	BlockUsingHostProce...	BuiltIn	Automated	Audit
Kubernetes cluster containers should only use allowed capabilities	ContainerCapabilities	BuiltIn	Automated	Audit
Kubernetes cluster pod hostPath volumes should only use allowed host paths	NoHostPathVolume	BuiltIn	Automated	Audit

Azure Policy

Policies (5) Groups (0) Parameters (4) JSON Assignments (0)

Filter by reference ID, policy name or ID... Type : **All selected** Evaluation type : **All selected**

Policy	Reference ID	Type	Evaluati...	Default effect
Kubernetes cluster should not allow privileged containers	NoPrivilegedContain...	BuiltIn	Automated	Audit
Kubernetes cluster pods should only use approved host network and port range	BlockUsingHostNetw...	BuiltIn	Automated	Audit
Kubernetes cluster containers should not share host process ID or host IPC namespace	BlockUsingHostProce...	BuiltIn	Automated	Audit
Kubernetes cluster containers should only use allowed capabilities	ContainerCapabilities	BuiltIn	Automated	Audit
Kubernetes cluster pod hostPath volumes should only use allowed host paths	NoHostPathVolume	BuiltIn	Automated	Audit

```
PS D:\Daten\Presentations\2024 - Workframe Training - CAA> kubectl apply -f .\podPrivileged.yaml
Error from server (Forbidden): error when creating ".\\podPrivileged.yaml": admission webhook "validation.gatekeeper.sh" denied the request: [azurepolicy-k8sazurev2noprivilege-bf86cd1783e70278c386] Privileged container is not allowed: nginx-privileged, securityContext: {"privileged": true}
```

Image Signing

Verify source and integrity of image

"Use Image Integrity to ensure only trusted images are deployed" Azure Policy in preview for AKS

Image Signing

[Preview]: Use Image Integrity to ensure only trusted images are deployed

Initiative Definition

Assign initiative | Edit definition | Duplicate definition | Delete initiative

Essentials

Name	: [Preview]: Use Image Integrity to ensure only trusted images ar...	Definition location	: --
Description	: Use Image Integrity to ensure AKS clusters deploy only trusted ...	Definition ID	: /providers/Microsoft.Authorization/policySetDefinitions/af28bf...
Category	: Kubernetes	Type	: Built-in
Version	: 1.1.0-preview		

Policies (3) | Groups (0) | Parameters (4) | JSON | Assignments (0)

Filter by reference ID, policy name or ID... | Type : **All selected** | Evaluation type : **All selected**

Policy	Reference ID	Type	Evaluati...	Default effect
[Preview]: Deploy Image Integrity on Azure Kubernetes Service	deployAKSImageInte...	BuiltIn	Automated	[if(equals(parameters...
Deploy Azure Policy Add-on to Azure Kubernetes Service clusters	deployAKSPolicyAdd...	BuiltIn	Automated	[if(equals(parameters...
[Image Integrity] Kubernetes clusters should only use images signed by notation	imageIntegrityNotati...	BuiltIn	Automated	Audit

Image Signing

Enable trust on registry

Enable trust on client through environment variable

Sign images using Notation CLI and AKV plugin

[Sign container images with Notation and Azure Key Vault using a self-signed certificate]

Container Security

Have non-root user in container

Run containers as non privileged

Scan images in CI/CD pipeline for vulnerabilities

Scan images using Microsoft Defender in Azure Container Registry

Container Security

Code analysis in CI/CD pipeline

- OWASP Zap
- Mend Bolt
- Dependabot
- Azure DevOps Advanced Security

Only allow reviewed NuGet packages

Deployments and Configuration

Infrastructure as code

- Audit deployments
- Review configuration changes
- Access management
- Documentation

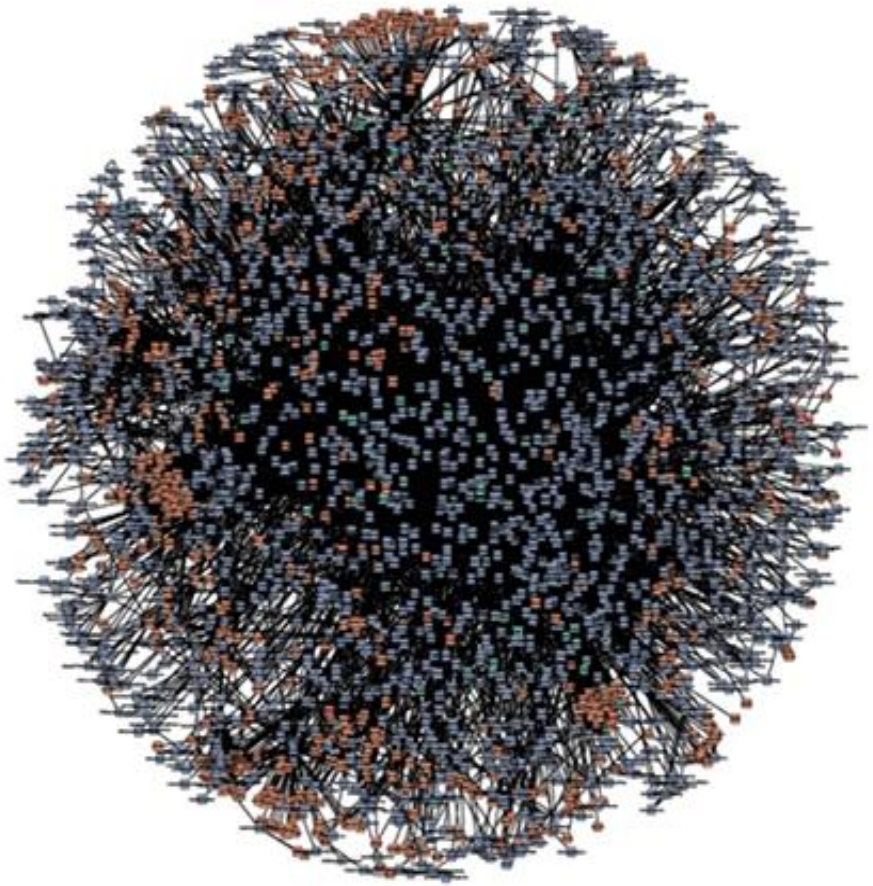
No manual changes or deployments

Service Mesh

Service Mesh

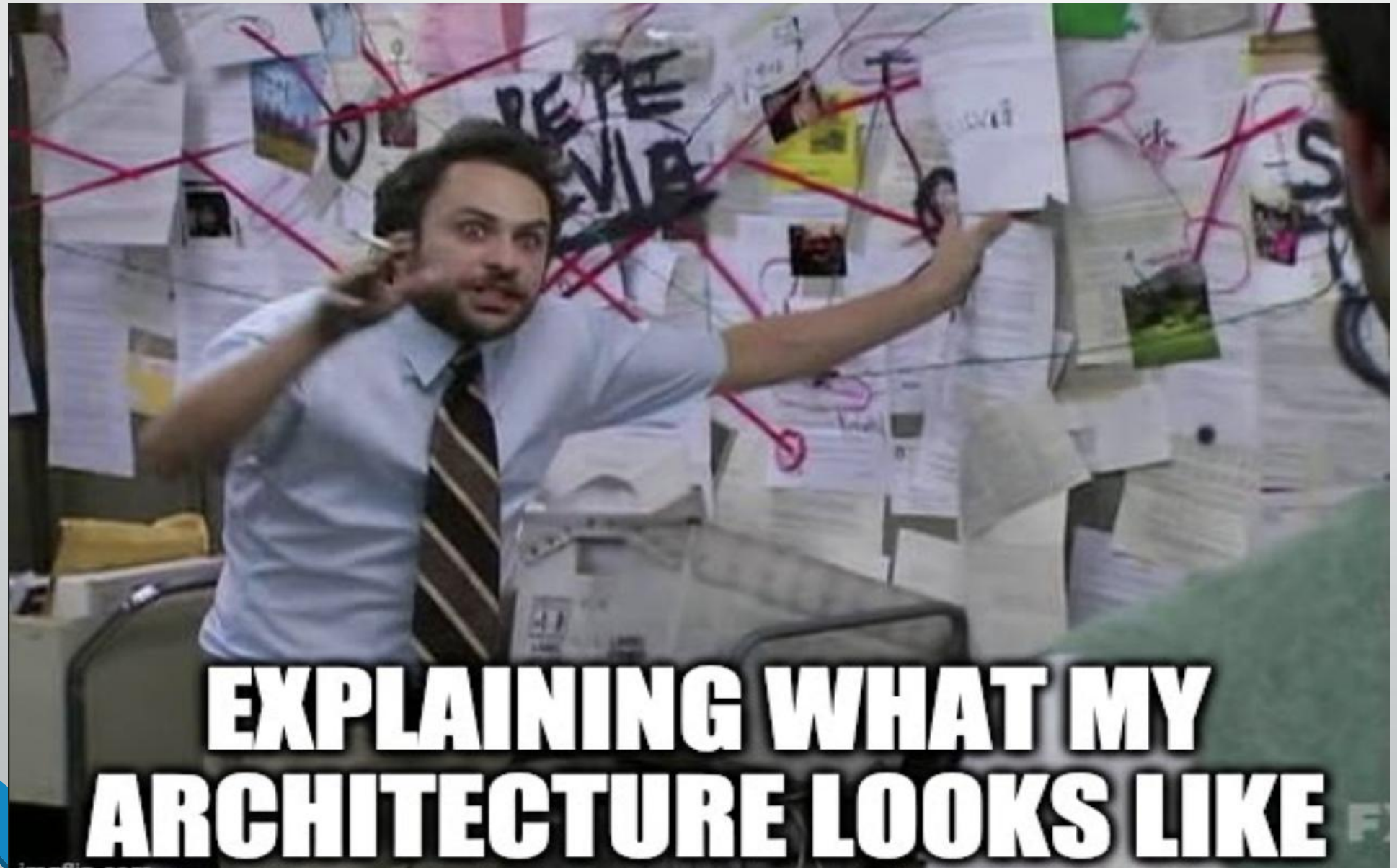
Applications are becoming more and more complex

Hundreds or thousands of services or containers need to be managed



amazon.com

NETFLIX


EXPLAINING WHAT MY
ARCHITECTURE LOOKS LIKE

Service Mesh

Applications are becoming more and more complex

Hundreds or thousands of services or containers need to be managed

Network traffic needs to be managed

- Frontend is only allowed to talk to the backend
- Backend is only allowed to talk to the database

Service Mesh

Separation of concerns

- Developers don’t want to manage a cluster
- Admins can’t know every application / programming language

Service Mesh

Testability

- Outages, e.g., Service is not available
- Degradation, e.g., Service response is slow
- Canary deployment

Security

- Encryption of traffic
- Monitor traffic flow

Service Mesh Addons

Service Meshes offer many addons:

- Grafana
- Prometheus
- Jaeger
- Zipkin
- Loki

Use sidecar pattern to inject proxy into application pod

Identification via labels

Service Mesh Advantages

Separation of Concerns

Traffic Management

- Request routing
- Fault injection
- Traffic shifting

Service Mesh Advantages

Security

- Certificate management
- TLS configuration

Monitoring and observability through addons

Service Mesh Disadvantages

Resource usage

- Envoy proxy uses 350m CPU and 40 MB RAM per 1000 requests

Yet another (complex) tool to learn

Additional complexity

Service meshes don't mesh together

Service Mesh

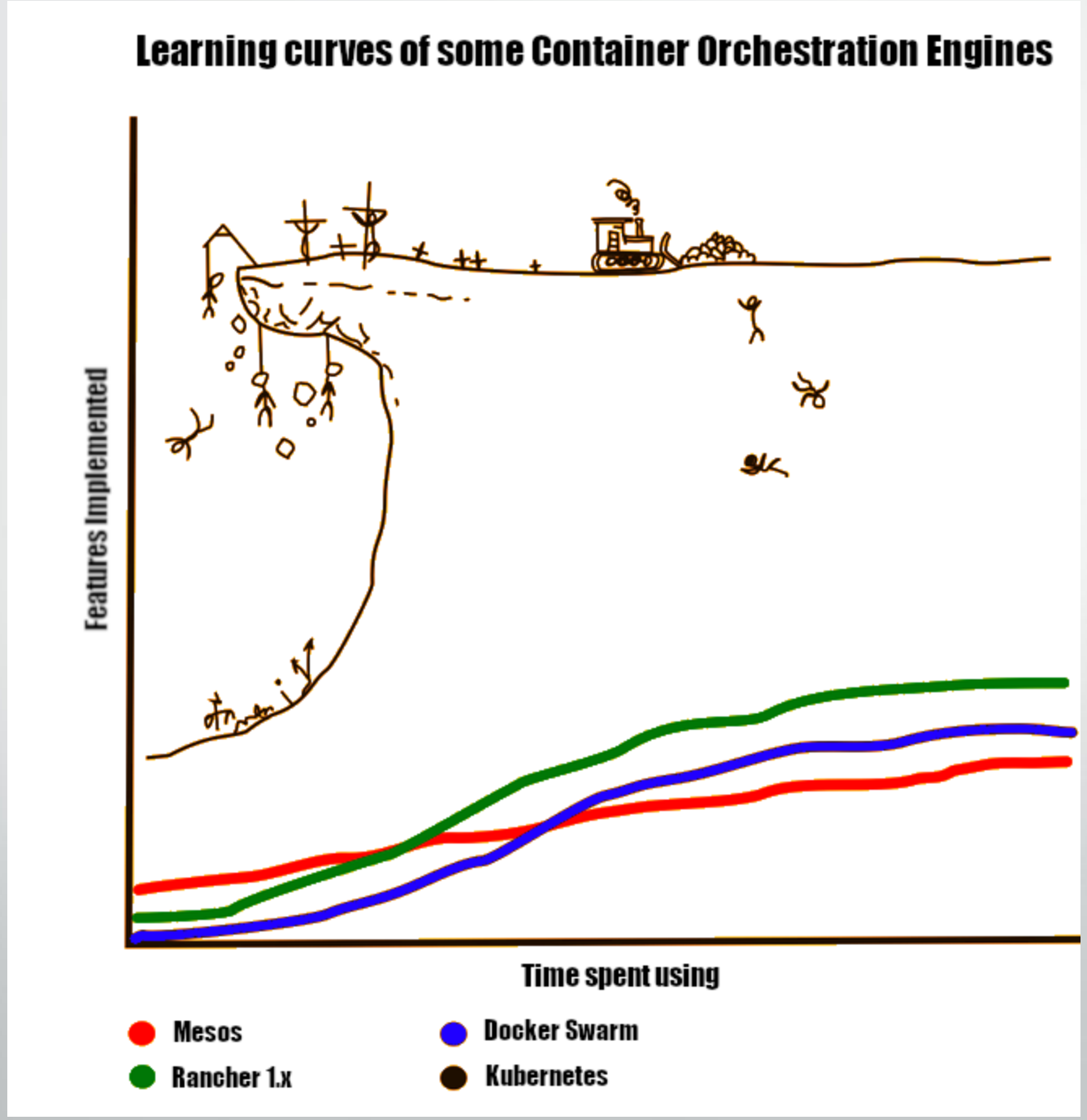
Istio

Consul

Linkerd

Azure Open Service Mesh

Advanced Tools and Learning Path


Learning curves of some Container Orchestration Engines
Features Implemented
Time spent using
Mesos
Docker Swarm
Rancher 1.x
Kubernetes

Certification

Kubernetes Fundamentals Training (35 hours)

Certified Kubernetes Administrator (CKA)

Certified Kubernetes Application Developer (CKAD)

Kustomize

Kubernetes configuration management tool

Scalable for various deployment sizes

Efficient configuration management

Enables customization without directly editing manifest files

Documentation

GitOps

Automation using Git and CI/CD pipelines

Configuration as code

History of config changes + Pull Request reviews

Automated deployments when cluster is not accessible

ArgoCD

Flux

KEDA

Kubernetes Event-driven Autoscaling

Enables autoscaling workloads based on external events

- Azure Service Bus
- Apache Kafka
- Redis Streams
- MongoDB
- Azure DevOps Pipeline

KEDA

Optimize resource usage

64 built-in scalers

Can be used without any changes in applications

Open-source

Documentation

Wolfgang Meetup Talk

How to Build Docker Images with Podman using an Azure DevOps Agent in Kubernetes

Cilium

Networking, security and observability

Layer 3 to layer 7 security policies

mTLS between pods

Visibility into network and application layer interactions

Open-source

Documentation

Secret Management

Secrets are not encrypted by default

Secrets should be encrypted to keep thems ecret

- Database connection strings
- Passwords
- Certificates

Secret Management

Tools

- Azure Key Vault Provider for Secrets Store CSI Driver
- HashiCorp Vault
- Sealed Secrets

Kubernetes Operations

Resource utilization

Tagging

Namespace limits

Pod security policies

Networking policies

RBAC policies

Logging

Kubernetes Production Best Practices Checklist