ID23-5311 Kubernetes Workshop

Wolfgang Ofner

Simon Birrer

Wolfgang Ofner

Freelance Cloud Architect, Toronto, Canada

Focus on Azure, Kubernetes, DevOps and .NET

https://programmingwithwolfgang.com

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

https://twitter.com/wolfgang_ofner

Agenda

- Docker
  - Theory
  - Practice Time
- Kubernetes
  - Theory
- Lunch Break
- Kubernetes
  - Theory
  - Practice Time
- Advanced Tools and Learning Path

Challenges of modern Software

Deploy 100 times a day

Versioning

Dependencies

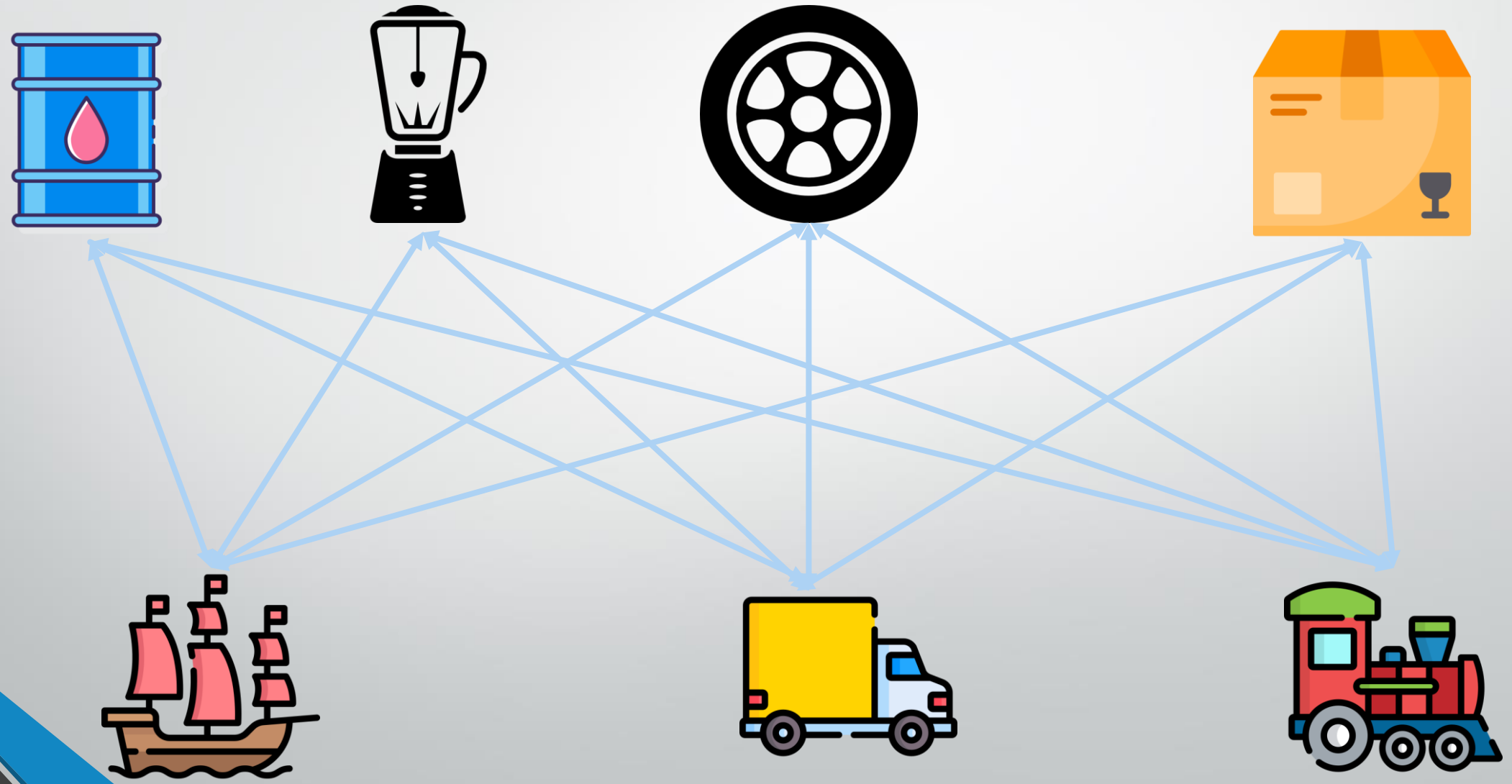
Easy to test

Fast to set up on target machine

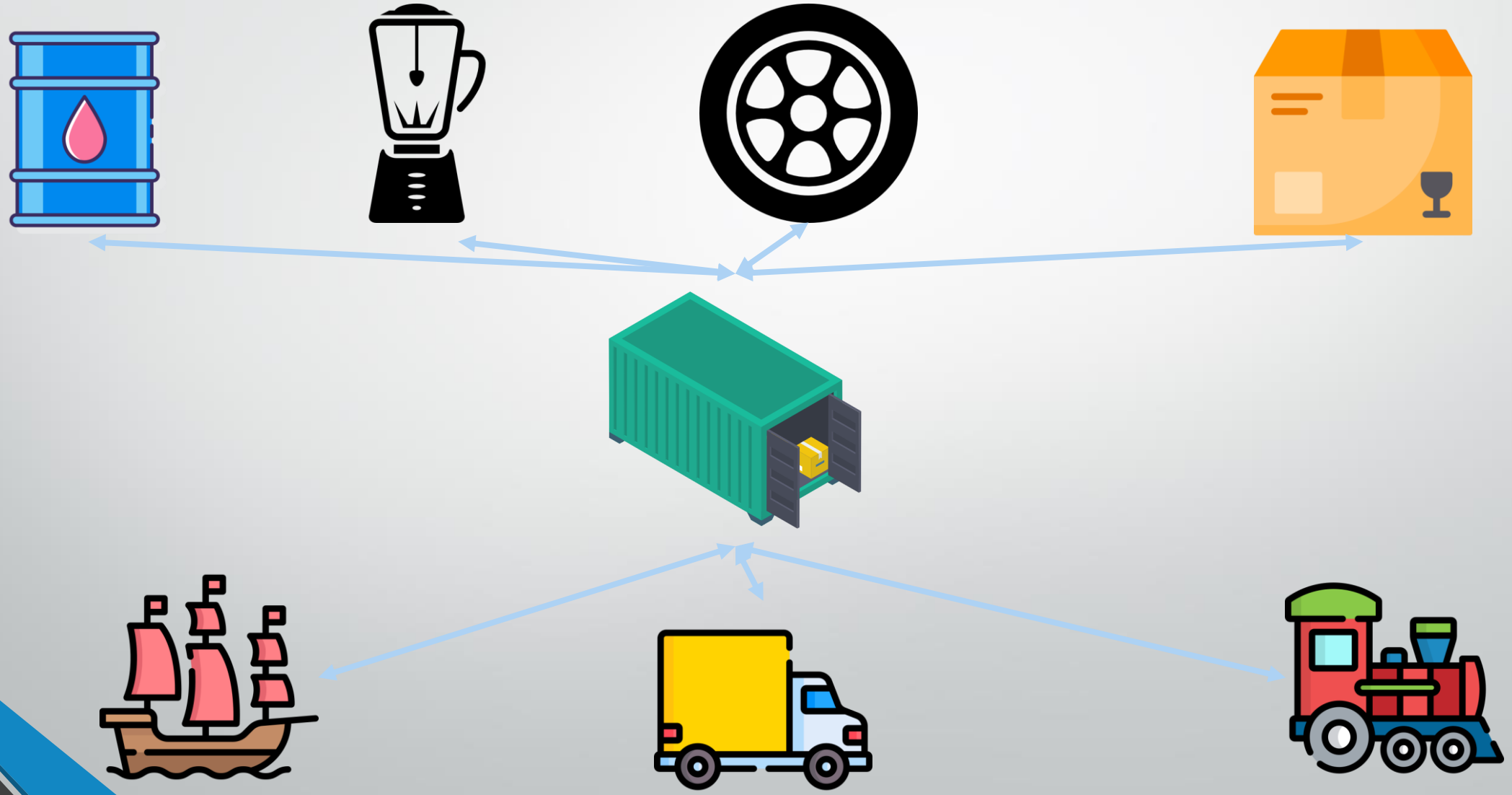
Monolithic software

Database deployments

Freighter transport before 1956



Invention of the Container (1956)




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

Container Solutions

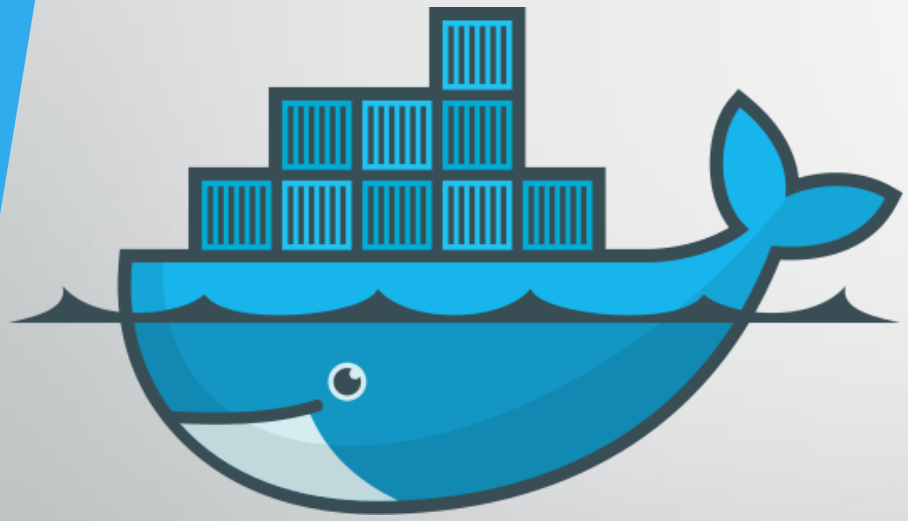
Docker

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

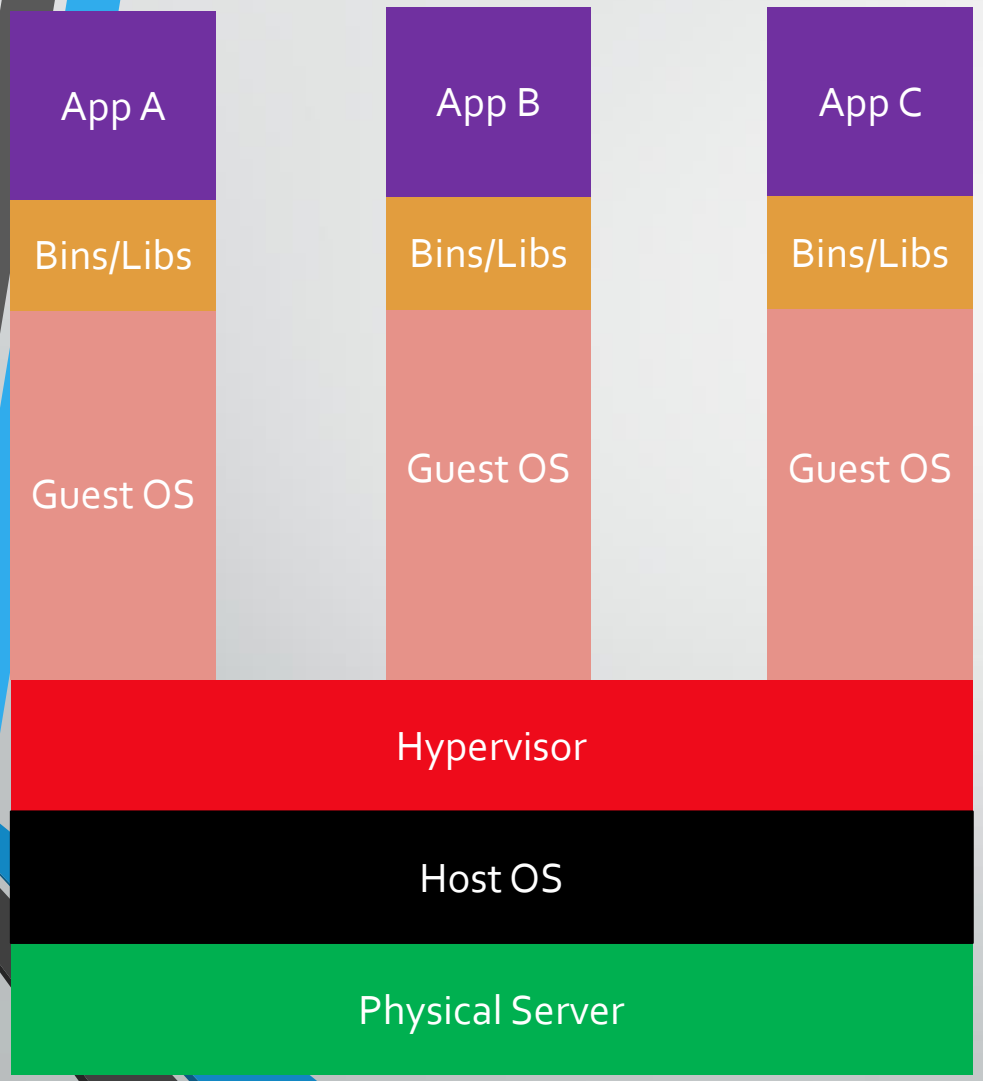
OCI (Open Container Initiative) compliant

Abstracts underlying infrastructure

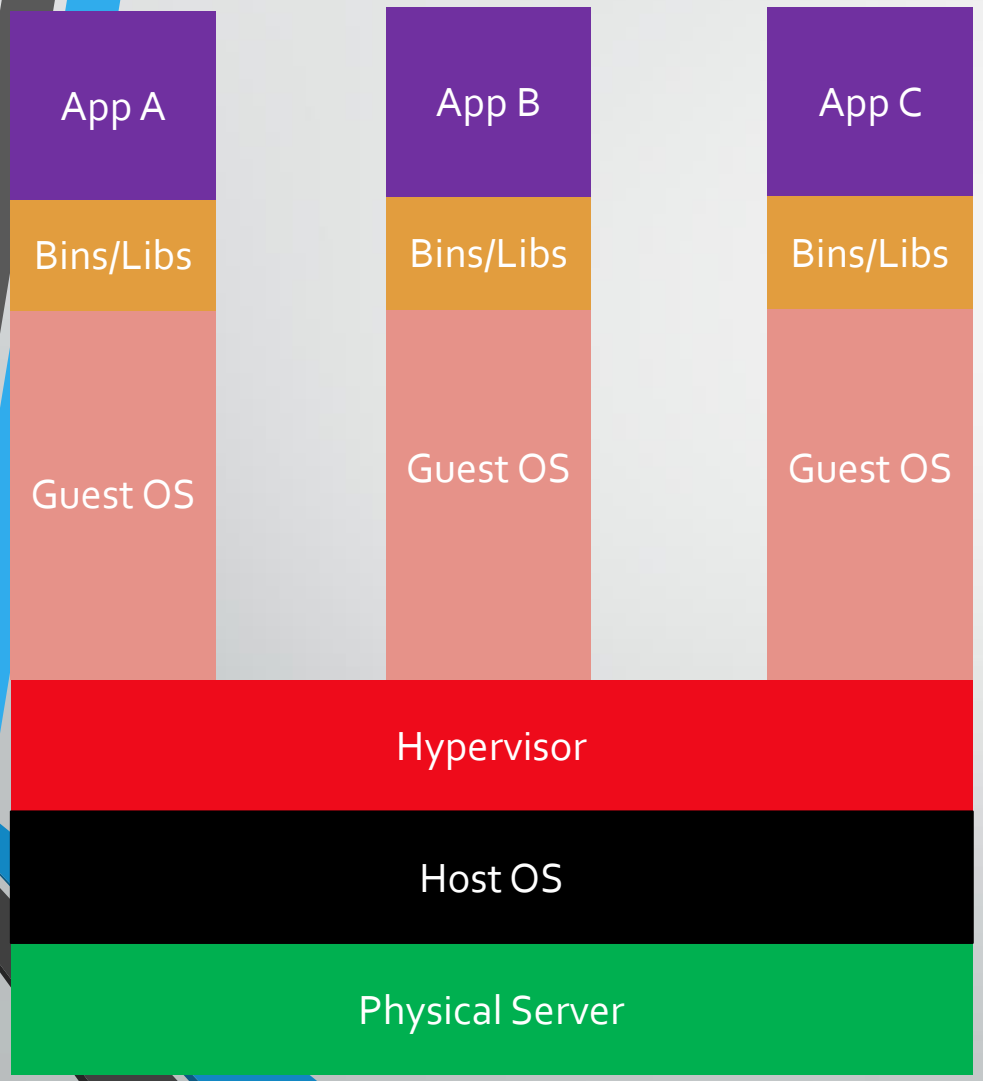
Fast start up times

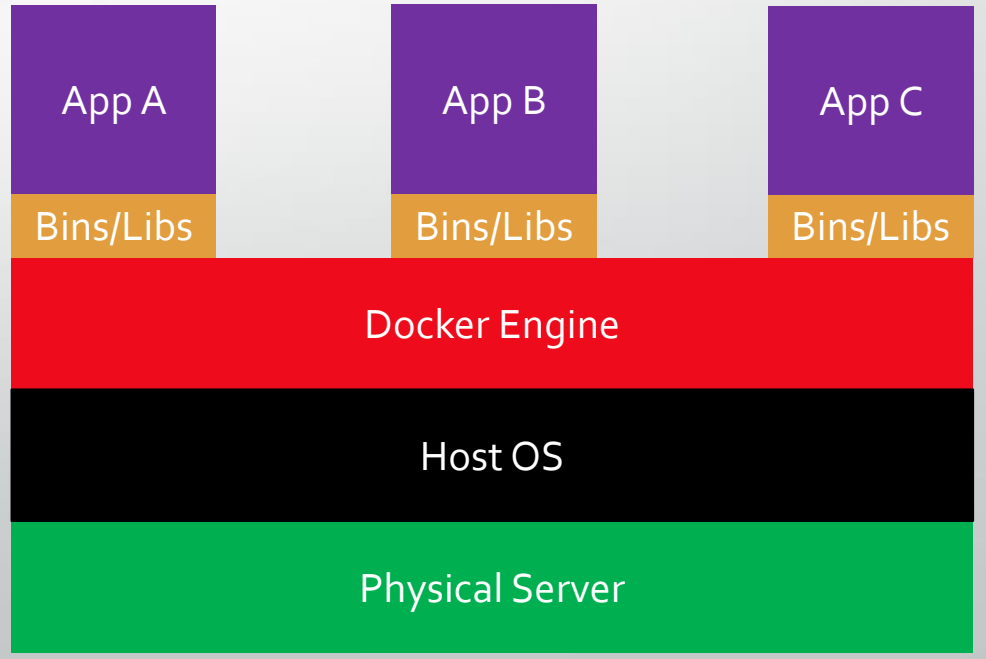
Pet vs. Kettle

Virtual Machine vs. Container



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:CoverletOutpu
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results3.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutpu

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

Onion File System

Every command is a new layer

Layers can be cached

Faster builds

```
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"]
```

Onion File System

Every command is a new layer

Layers can be cached

Faster builds

```
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"]
```

Onion File System

Every command is a new layer

Layers can be cached

Faster builds

```
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"]
```

11 Layer

57 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/"]
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:CoverletOutpu
RUN dotnet test --no-build -c Release --results-directory /testresults --logger "trx;LogFileName=test_results3.trx" /p:CollectCoverage=true /p:CoverletOutputFormat=json%2cCobertura /p:CoverletOutpu

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"]
```

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

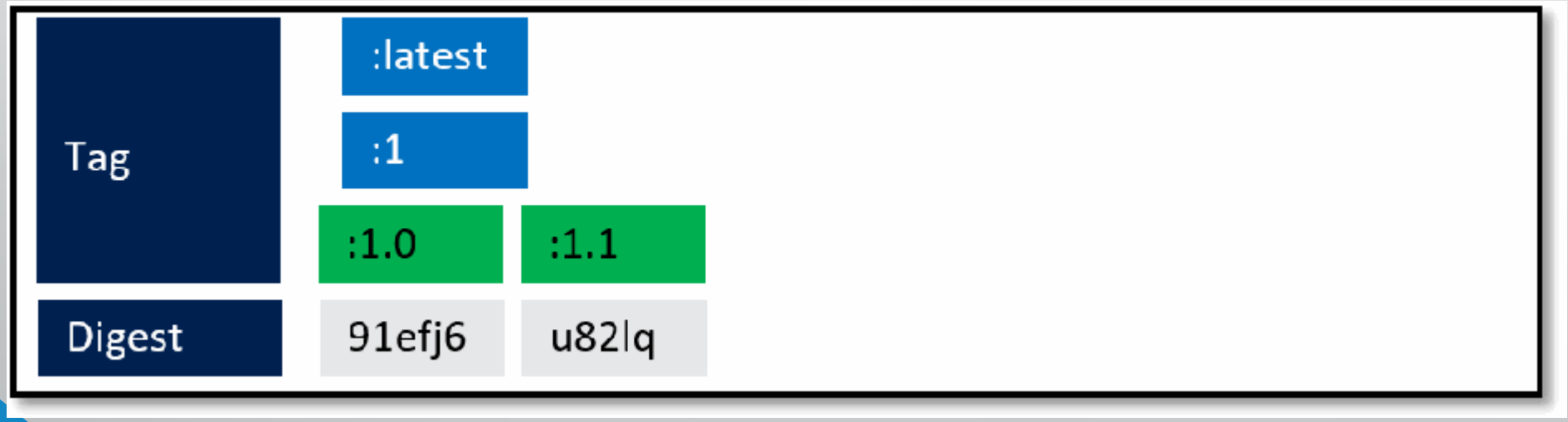
Used for versioning

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

Products

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

Trusted Content

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

Operating Systems

- [ ] Linux
- [ ] Windows

Architectures

- [ ] ARM
- [ ] ARM 64

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

Best Match

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

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

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 Commands

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

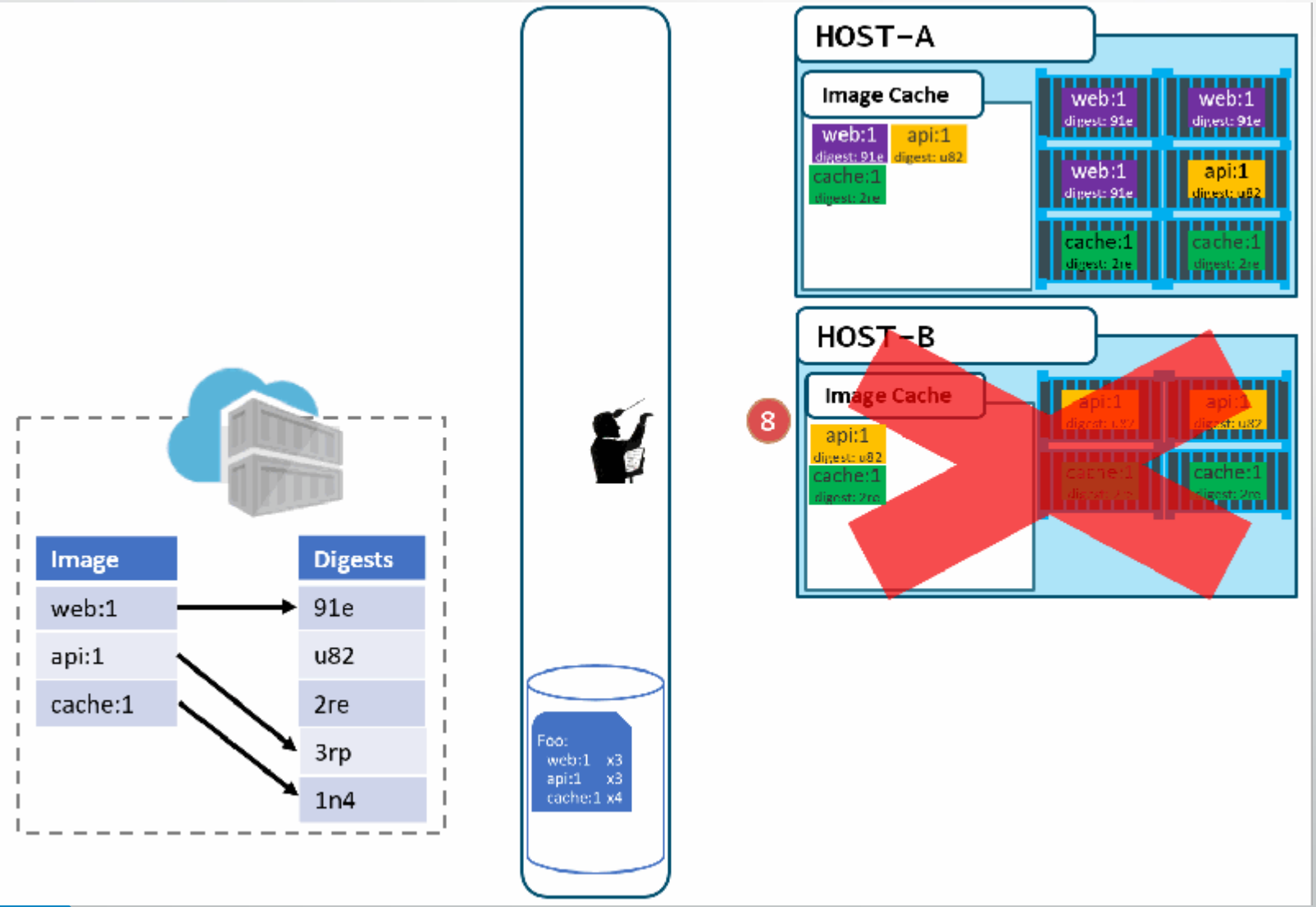
Exercise

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



Container Orchestrator

Multi-Node Management

Resource Management

Load balancing

Monitoring and Self Healing

Zero Downtime Deployments

Manage SSL certificates

Container Orchestrator

Multi-Node Management

Resource Management

Load balancing

Monitoring and Self Healing

Zero Downtime Deployments

Manage SSL certificates

Kubernetes

Docker Swarm

Marathon

Container Orchestrator

Multi-Node Management

Resource Management

Load balancing

Monitoring and Self Healing

Zero Downtime Deployments

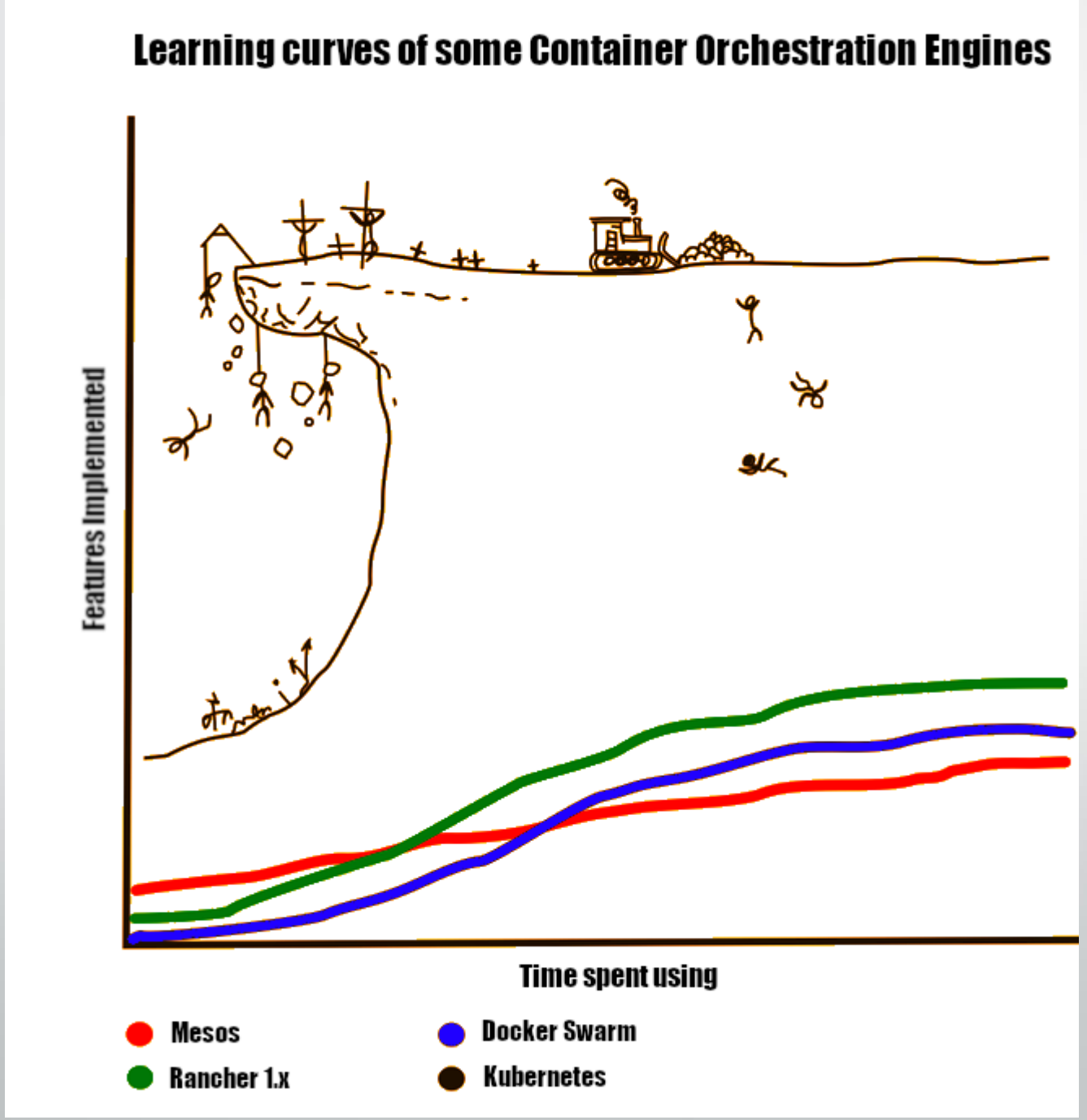
Manage SSL certificates

Kubernetes

Docker Swarm

Marathon

kubernetes


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

Kubernetes

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

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

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

Open-source

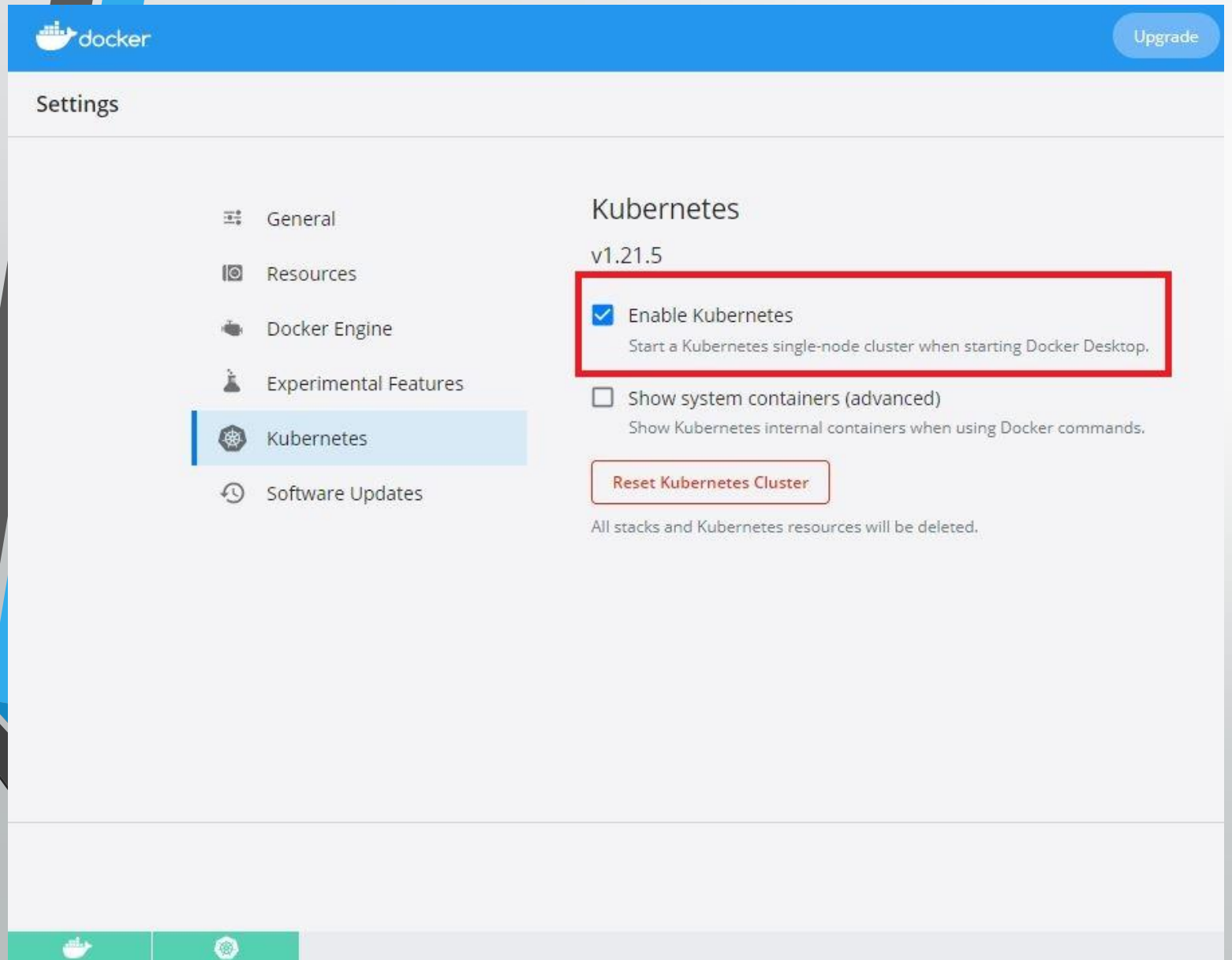
"K8s" → K-8 character-s

Kubernetes

Multiple distribution

- K3s
- MicroK8s
- RedHat OpenShift
- Mirantis
- Azure Kubernetes Service
- Amazon Elastic Kubernetes
- Google Kubernetes Engine

K8s locally


docker
Upgrade
Settings
General
Resources
Docker Engine
Experimental Features
Kubernetes
Software Updates
Kubernetes
v1.21.5
Enable Kubernetes
Start a Kubernetes single-node cluster when starting Docker Desktop.
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Reset Kubernetes Cluster
All stacks and Kubernetes resources will be deleted.

K8s locally

docker

Upgrade

Settings

- General
- Resources
- Docker Engine
- Experimental Features
- Kubernetes
- Software Updates

Kubernetes

v1.21.5

☑ Enable Kubernetes

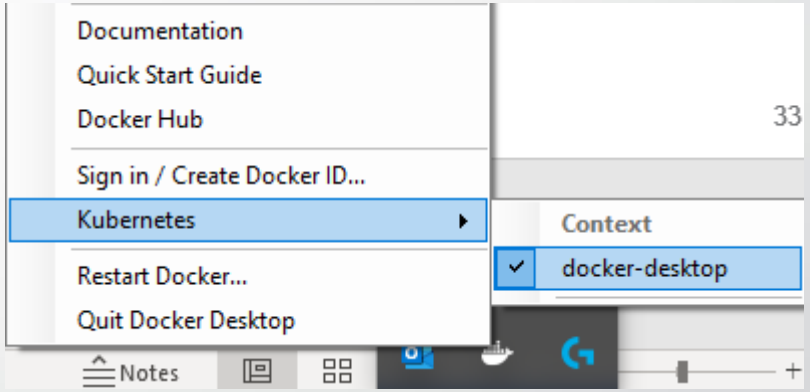
Start a Kubernetes single-node cluster when starting Docker Desktop.

☐ Show system containers (advanced)

Show Kubernetes internal containers when using Docker commands.

Reset Kubernetes Cluster

All stacks and Kubernetes resources will be deleted.



K8s locally

K3s:

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

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

Easily extensible

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

Kubernetes Components

Master Node (Control Plane)

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

Worker Node

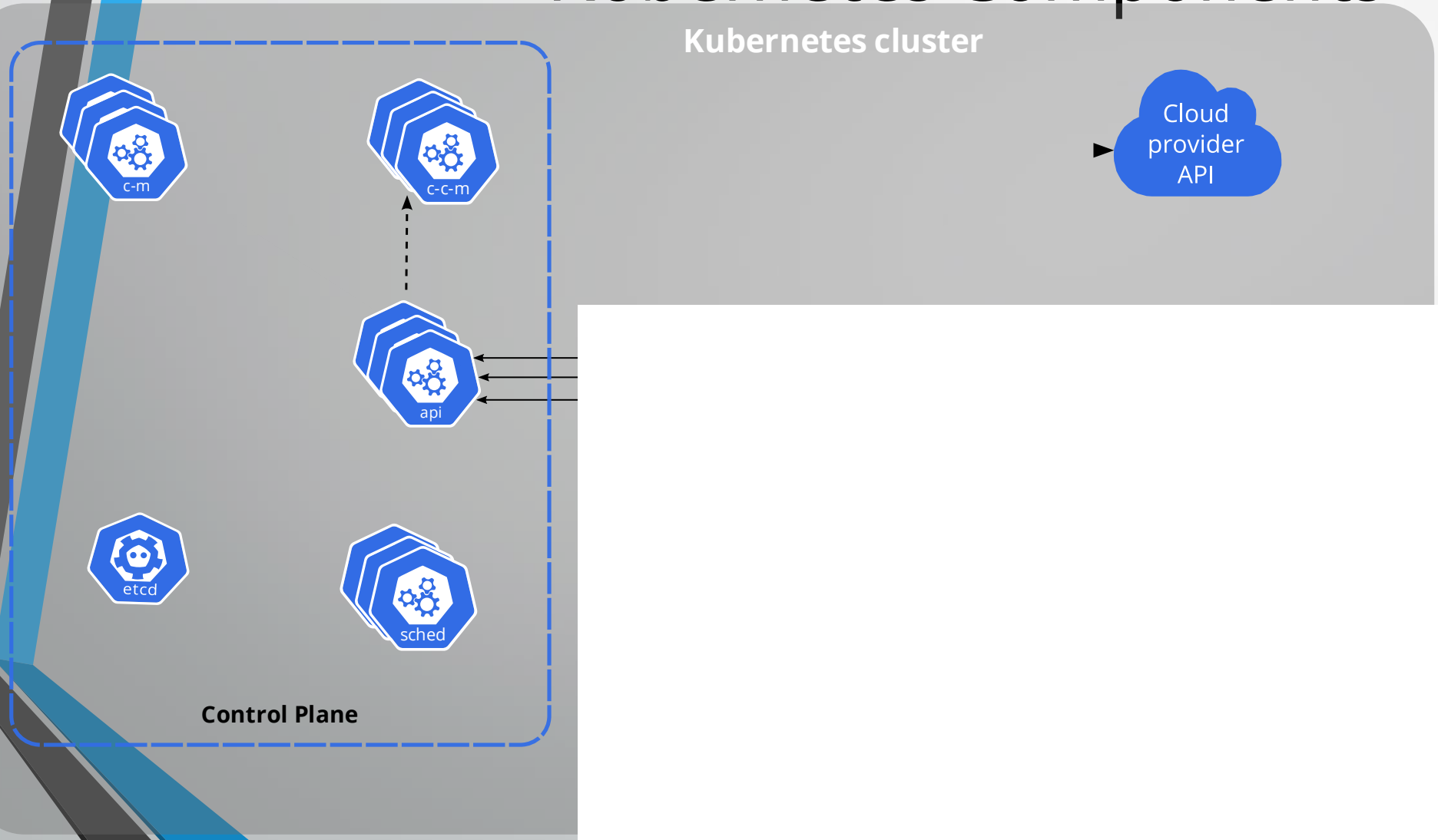
- kubelet
- kube-proxy
- Container runtime

Kubernetes Components

Addons

- DNS
- Networking
- Storage
- Dashboard
- …

Kubernetes Components

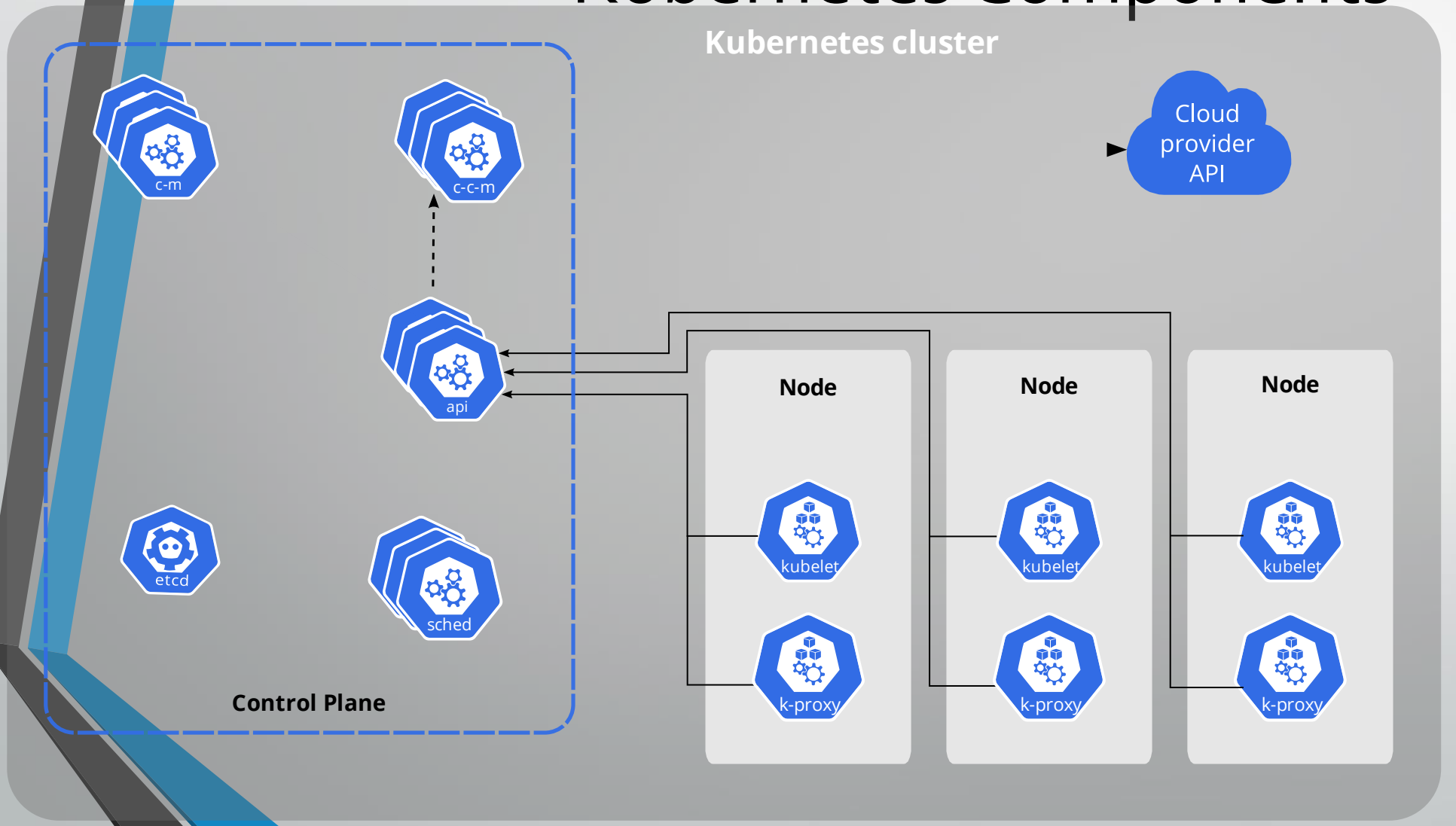


API server

Cloud controller manager *(optional)*

Controller manager

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

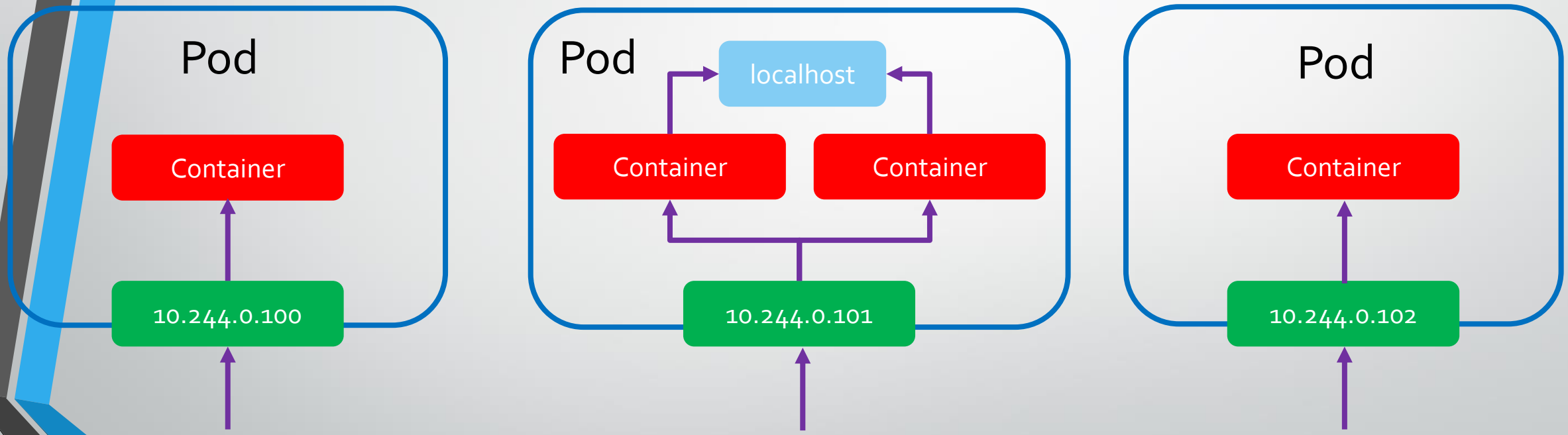
Pods wrap one ore more containers

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

Kubernetes Networking

The default network IP range for pods is 10.244.0.0/16

Network CIDR range can be configured with addon:

- Cilium
- Flannel
- Calico

Namespaces

Used to create virtual cluster inside a physical cluster

Namespaces

Used to create virtual cluster inside a physical cluster

- Isolation
- Resource Segregation
- Multiple Environments
- Resource Quotas
- Default Namespace

Kubernetes Configuration

Declarative Model and Desired State

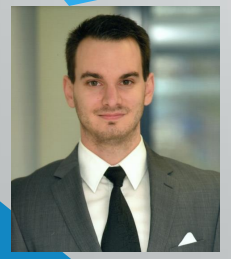
- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Hey Kubernetes, run 3 pods of wolfgangofner/customerapi

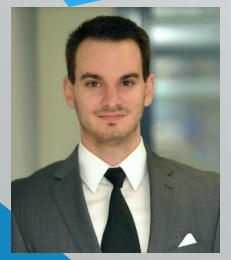


Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Hey Kubernetes, run 3 pods of wolfgangofner/customerapi



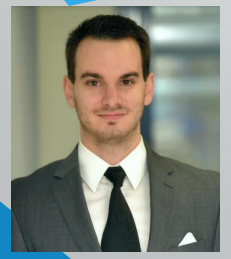
Let me check if I am already running your pods

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Hey Kubernetes, run 3 pods of wolfgangofner/customerapi



Let me check if I am already running your pods

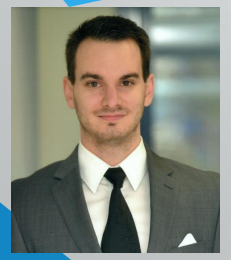
Currently there is one pod of wolfgangofner/customerapi running

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Hey Kubernetes, run 3 pods of wolfgangofner/customerapi



Let me check if I am already running your pods

Currently there is one pod of wolfgangofner/customerapi running

Starting two more pods of wolfgangofner/customerapi

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Configuration Handling

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

Kubernetes Configuration

Declarative Model and Desired State

- Tell Kubernets what you want
- Kubernetes will figure out a way to get to the desired state
- Etcd holds the current status of any K8s component

Configuration Handling

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

```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 value pairs 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

Labels

- Key value pairs 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

Annotations

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

Labels and Annotations

Labels

- Key value pairs 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

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

Pods come and go

IP addresses will change

Service 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

Service stay for the entire lifetime of the application

Persistent entry point

Fixed IP address

Load Balancing

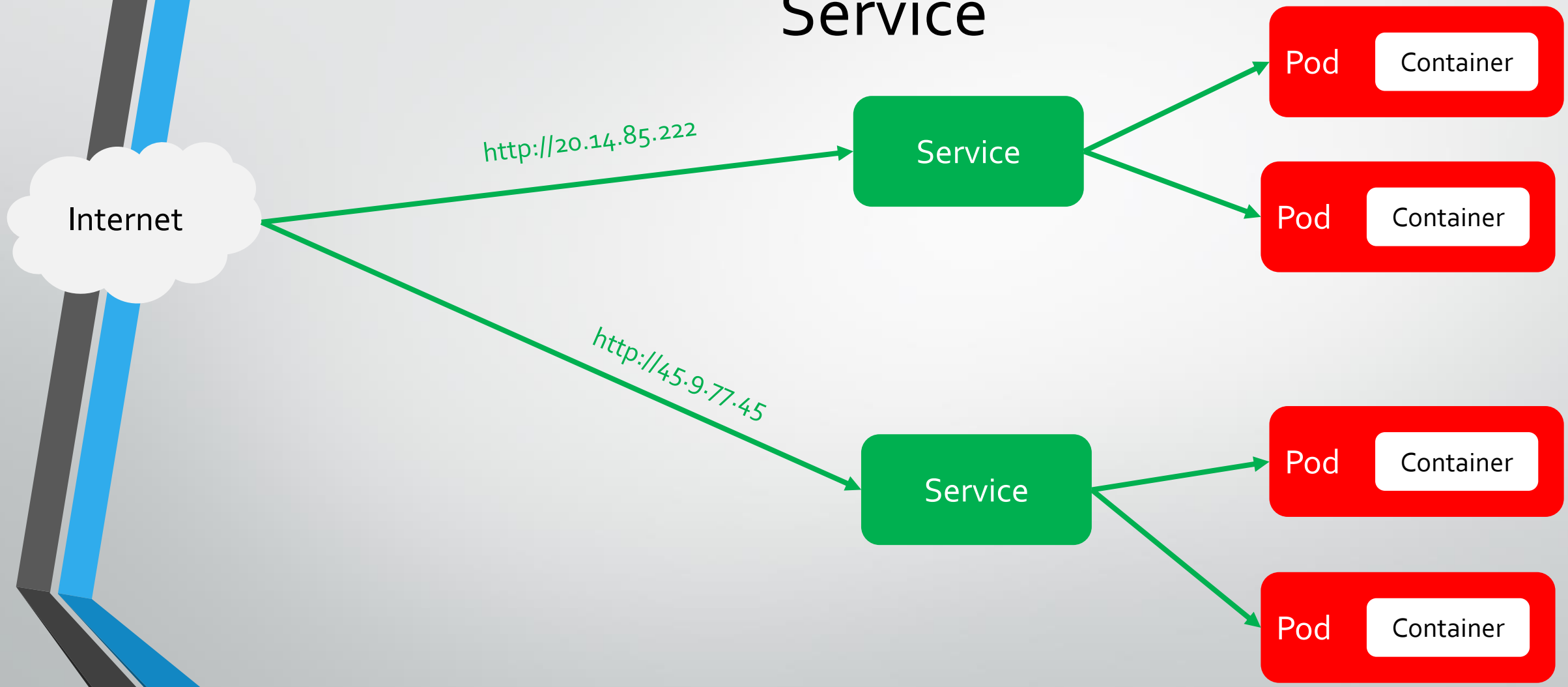
Pods and Services are matched using Labels

Services

Four types of Services

- LoadBalancer: exposes the Service using an external load balancer
- ClusterIP: makes the Services 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 configt set-context --current --namespace=my-namespace

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

Exercise

Exercise

Get Kubernetes up and running

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

Lunch Break

Recap

Questions about what we have learned so far?

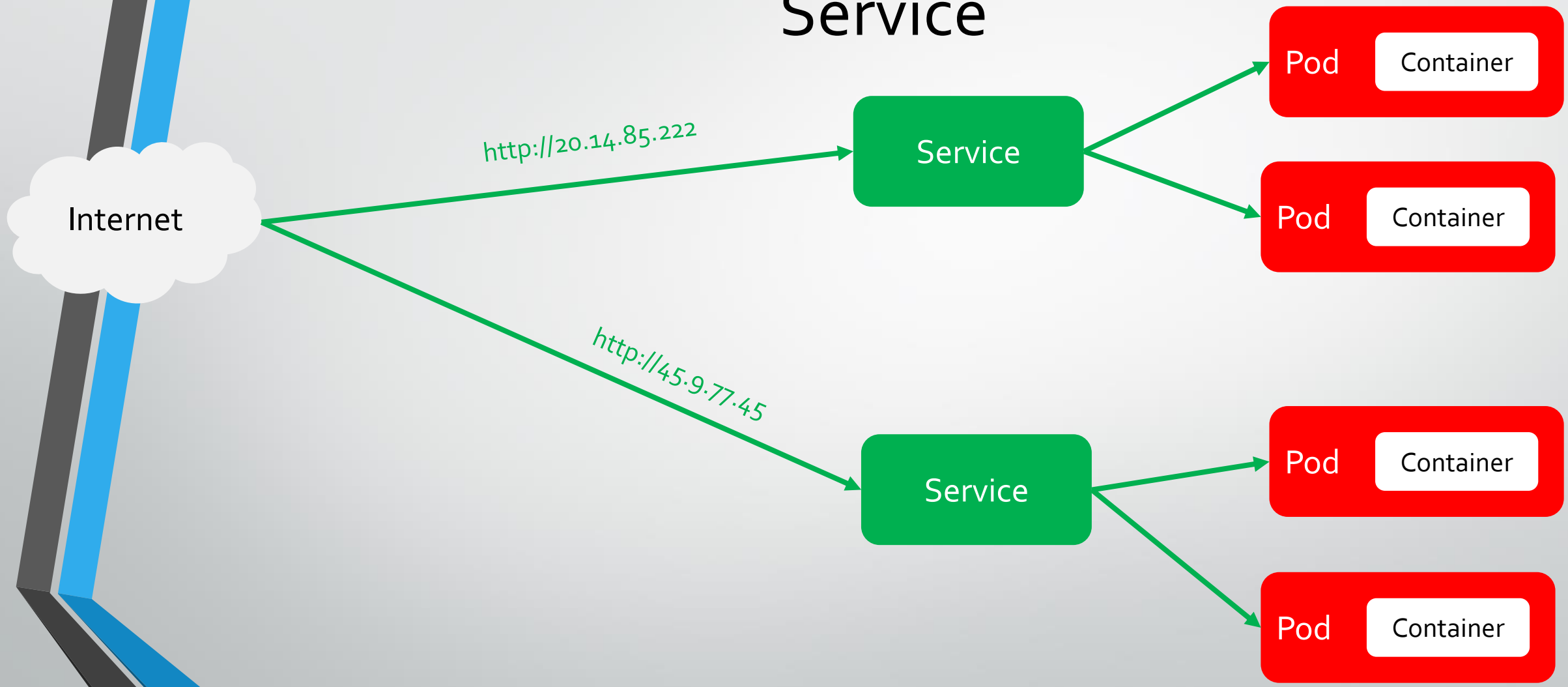
Recap

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

Container are immutable objects

Pods run containers in Kubernetes

Services are an entry point and route requests to pods

Needed for production environment:

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

Secrets

Base64 encoded

Automatically decrypted when attached to pod

Can be used in config file or environment variable

Secrets

Base64 encoded

Automatically decrypted when attached to pod

Can be used in config file or environment variable

Config and Storage > Secrets > kedademoapi-tls

kedademoapi-tls

Summary | Metadata | Resource Viewer | YAML

```yaml
---
apiVersion: v1
data:
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZXVENDQkVHZ0F3SUJBZ0lT
  tls.key: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb2dJQkFBS0NBUUVB
kind: Secret
```

Cert-Issuer

Kubernetes resource

Handles certificate requests

Supported sources:

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

Cert-Issuer

Kubernetes resource

Handles certificate requests

Supported sources:

- 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

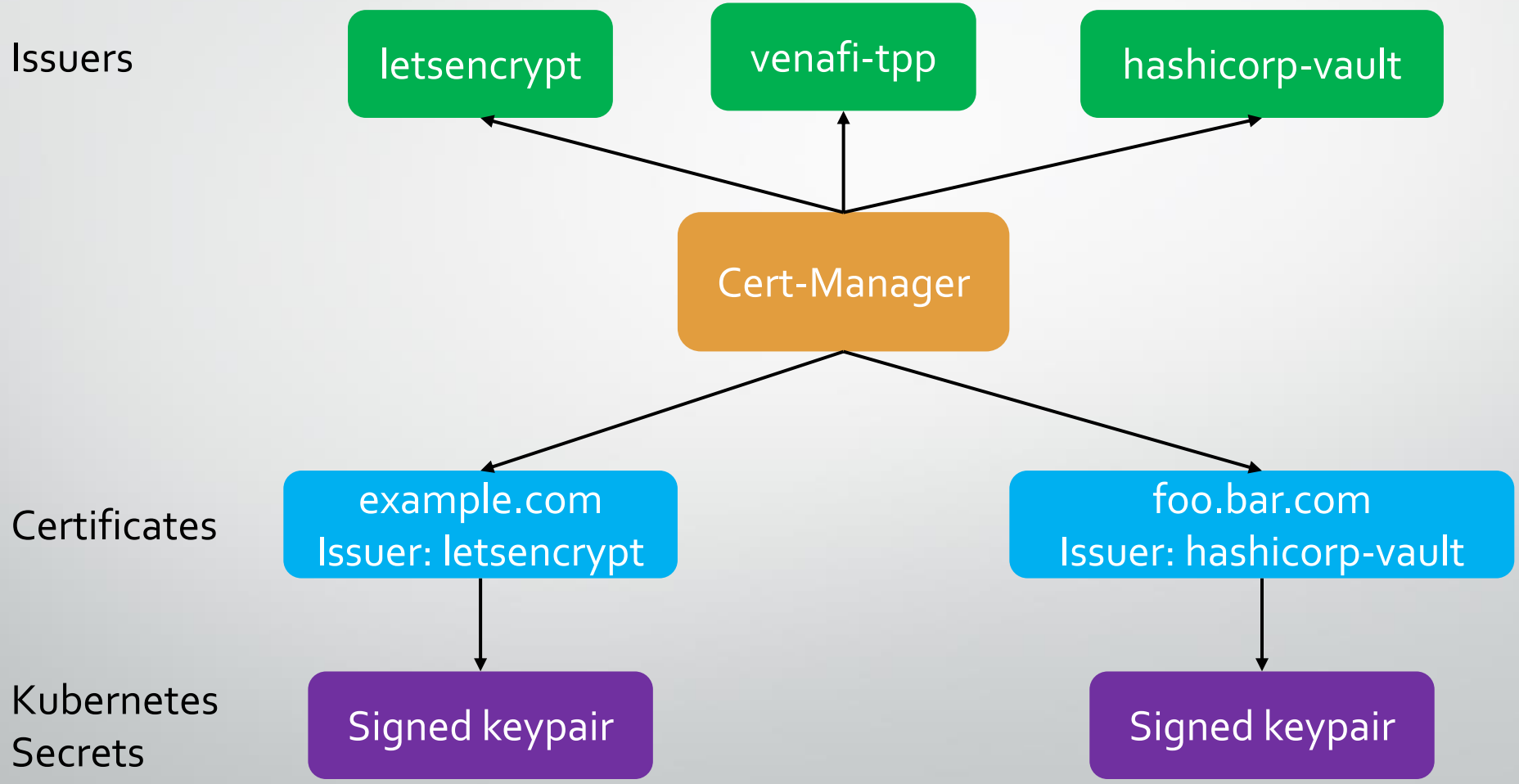
Manages obtaining and renewing of certificates

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



Config and Storage > Secrets > kedademoapi-tls

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

Cu

/swagg

A simp

Wolfga

Send

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

Ingress Controller

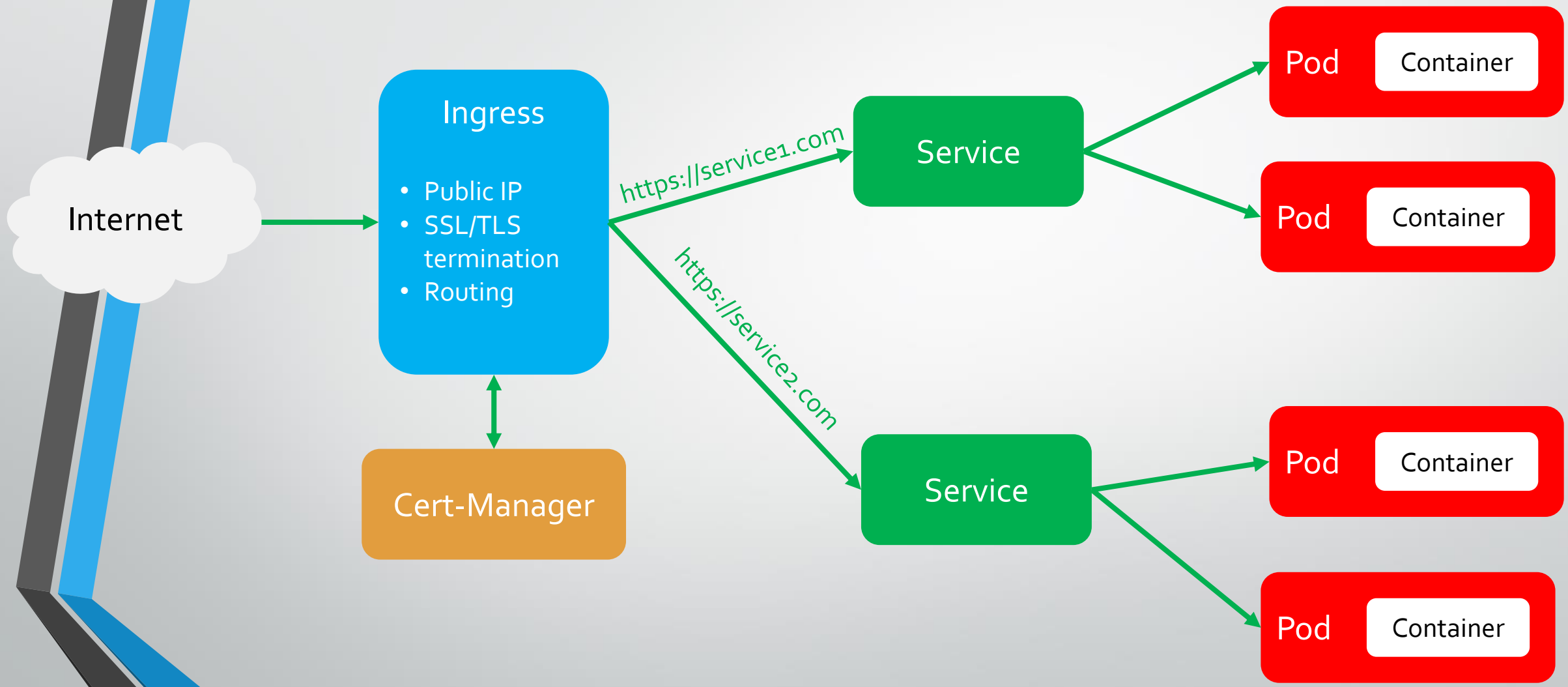
Entry point into the cluster

Processes HTTPS traffic

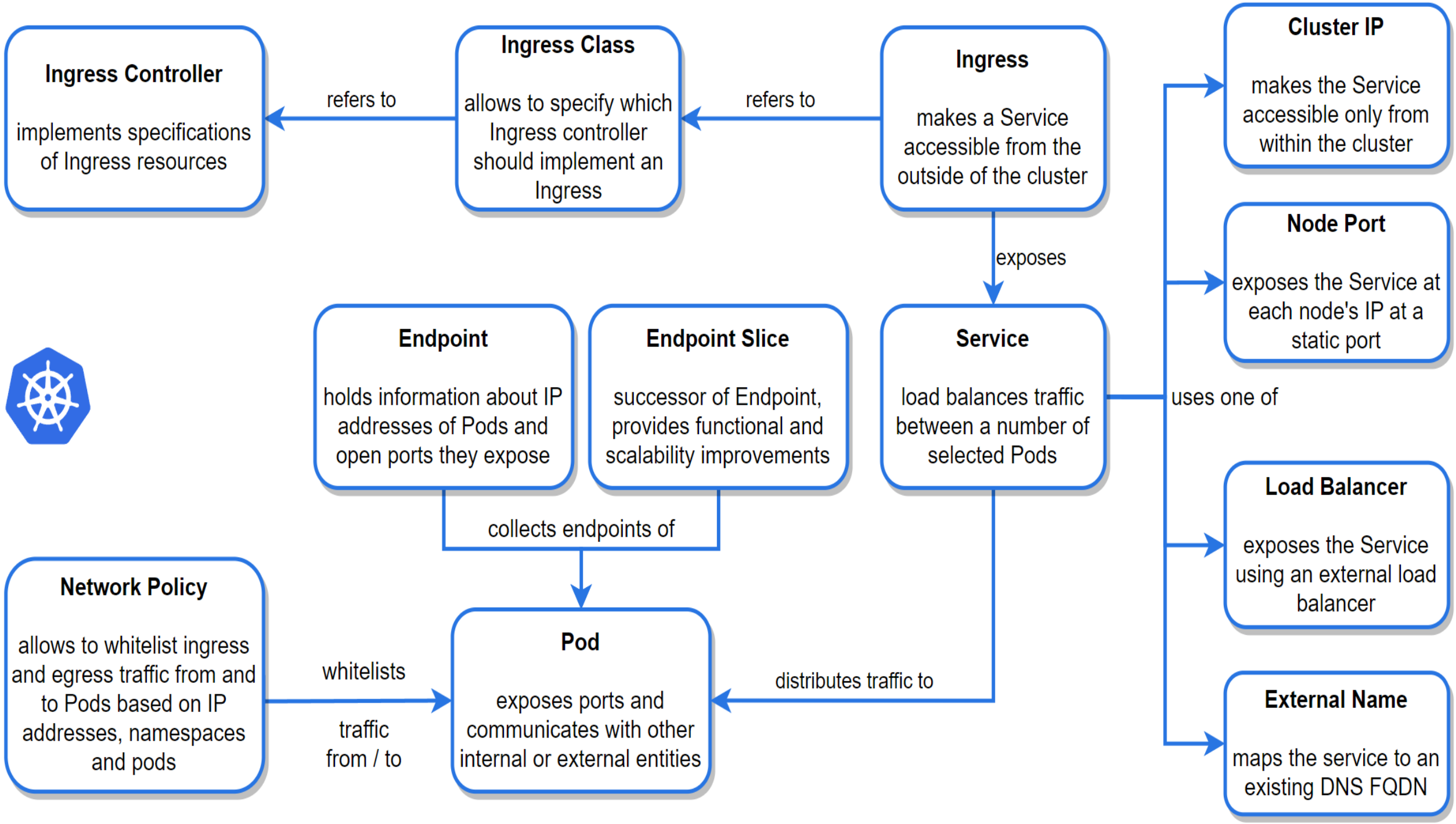
Reverse proxy redirects requests to Application

Popular Ingress Controller:

- NGINX
- Traefik
- HAProxy


Internet
Ingress
• Public IP
• SSL/TLS termination
• Routing
Cert-Manager
https://service1.com
https://service2.com
Service
Service
Pod
Container
Pod
Container
Pod
Container
Pod
Container

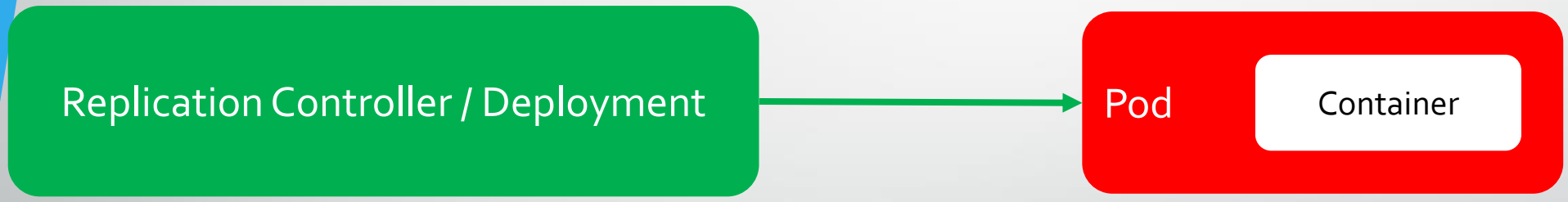
Pull Request Demo


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
Endpoint
holds information about IP addresses of Pods and open ports they expose
Endpoint Slice
successor of Endpoint, provides functional and scalability improvements
Service
load balances traffic between a number of selected Pods
uses one of
Load Balancer
exposes the Service using an external load balancer
collects endpoints of
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
exposes ports and communicates with other internal or external entities
distributes traffic to
External Name
maps the service to an existing DNS FQDN

Pod Deployment

Pods are not directly deployed

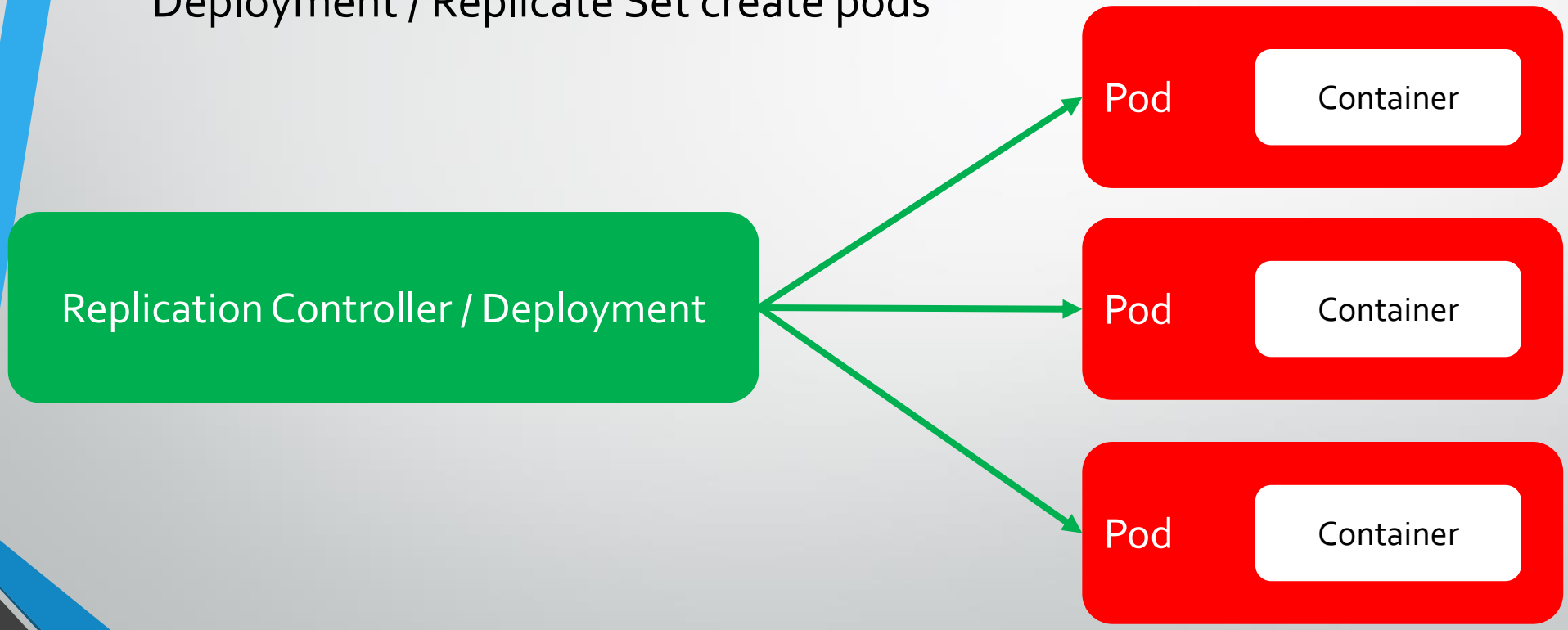
Deployment / Replicate Set create pods



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, StatefullSet

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

DaemonSet,CronJob, StatefullSet

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

CronJobs can be scheduled to start pods

StatefullSets manage stateful applications

DaemonSet,CronJob, StatefullSet

DaemonSets run pods on every node in the cluster

- Logging
- Monitoring
- Backup
- Reports
- Automated testing

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

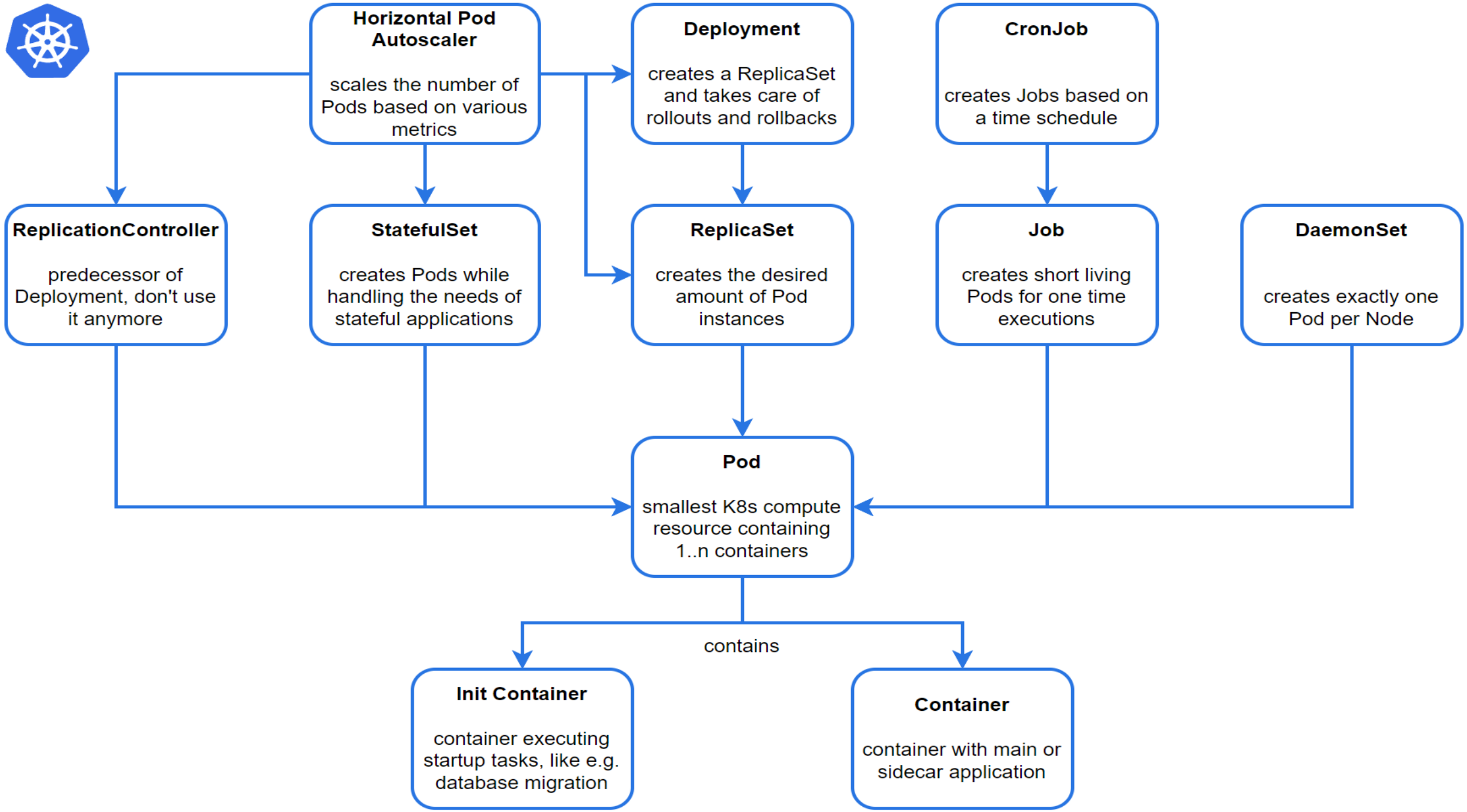
Horizontal Pod Autoscaler (HPA)

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 check
- 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

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 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

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 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

Resource Requests & Resource Limits

Resource Requests

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

Resource Limits

- Maximum resources a pod is allowed to use
- Pods gets throttled when it uses too many resources
- CPU and/or RAM

Resource Requests & Resource Limits

Resource Requests

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

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

Resource Limits

- Maximum resources a pod is allowed to use
- Pods gets throttled when it uses too many resources
- CPU and/or RAM

Resource Quotas

Limit resource usage within a namespace

Configured in the ResourceQuota object

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

10 Min Break

Deployment Strategies

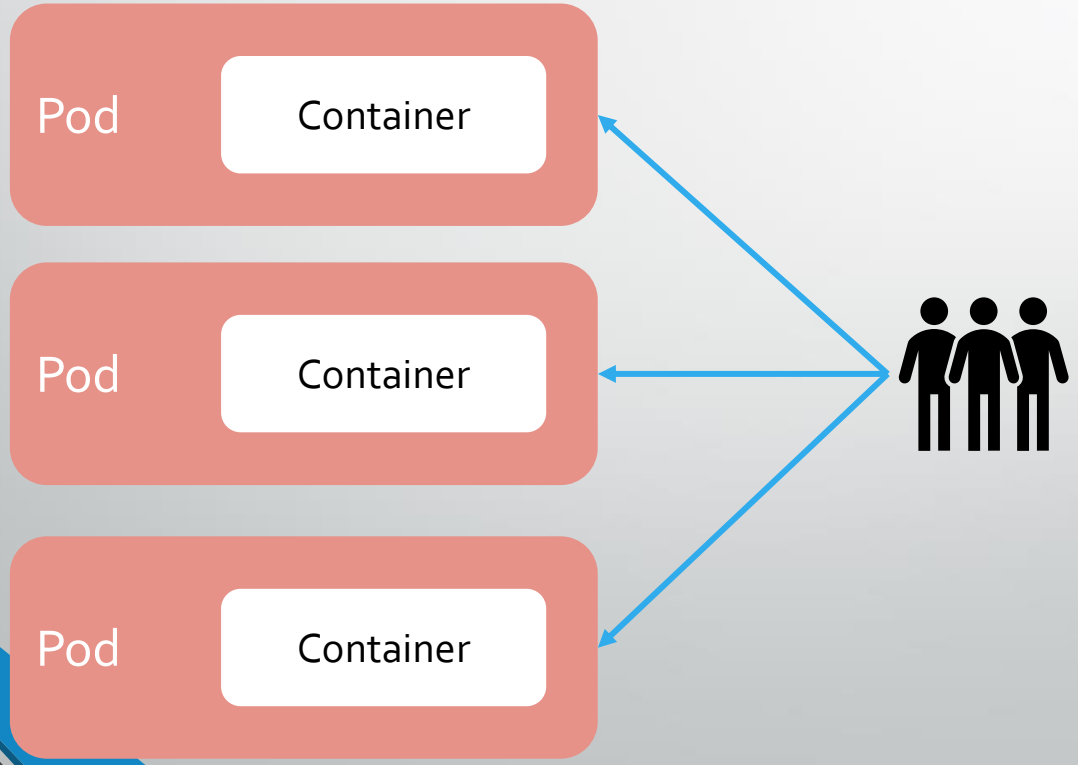
Blue Green deployment

Start all new pods and then switch

Deployment Strategies

Blue Green deployment

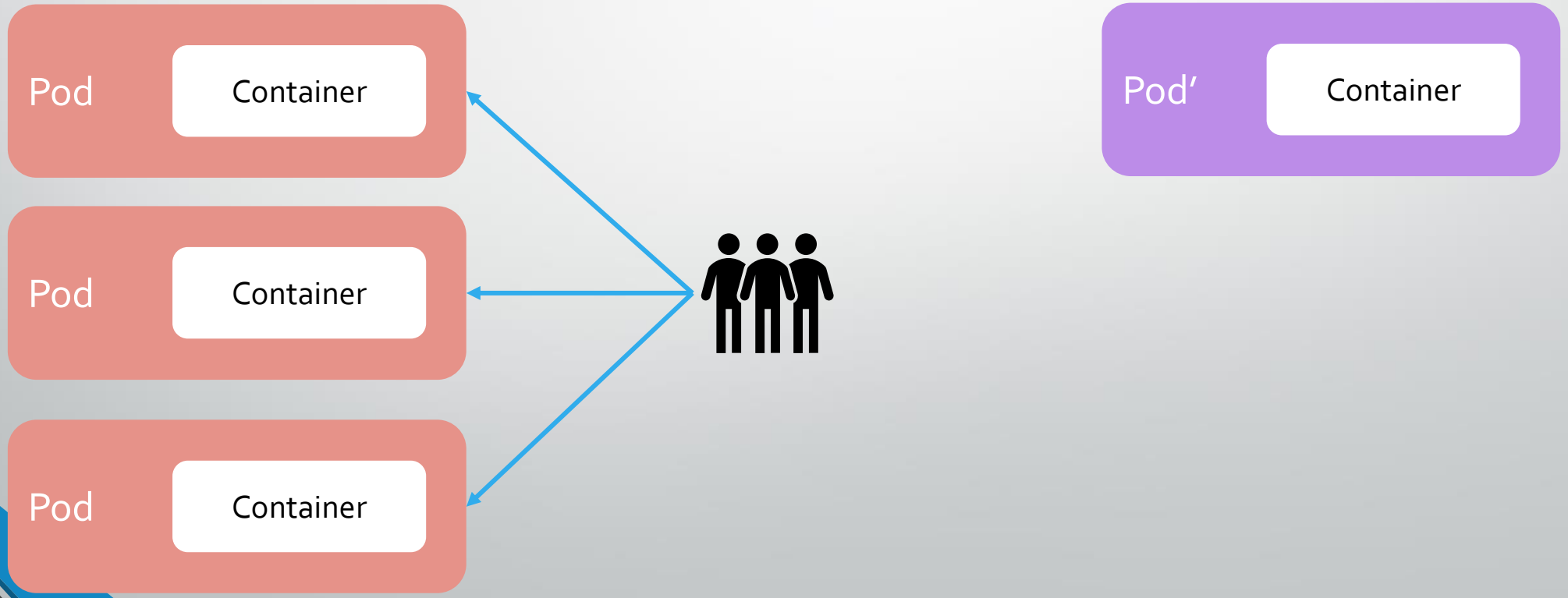
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

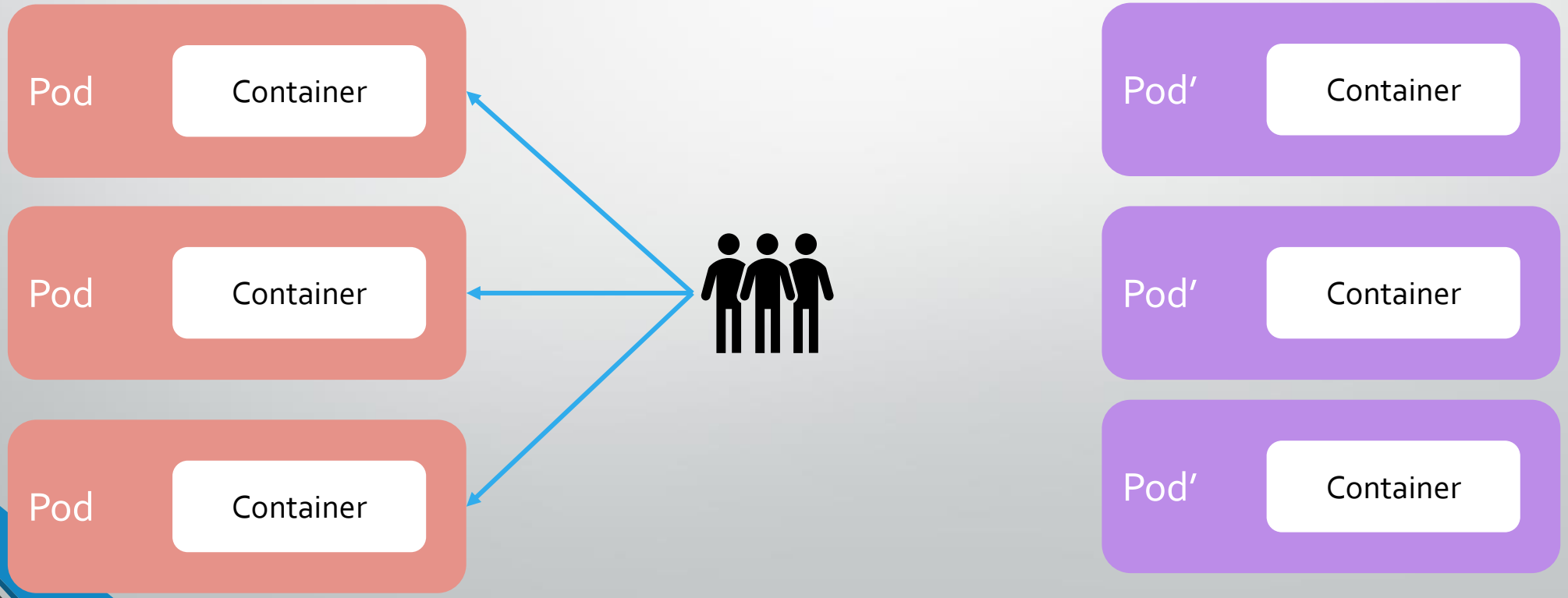
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

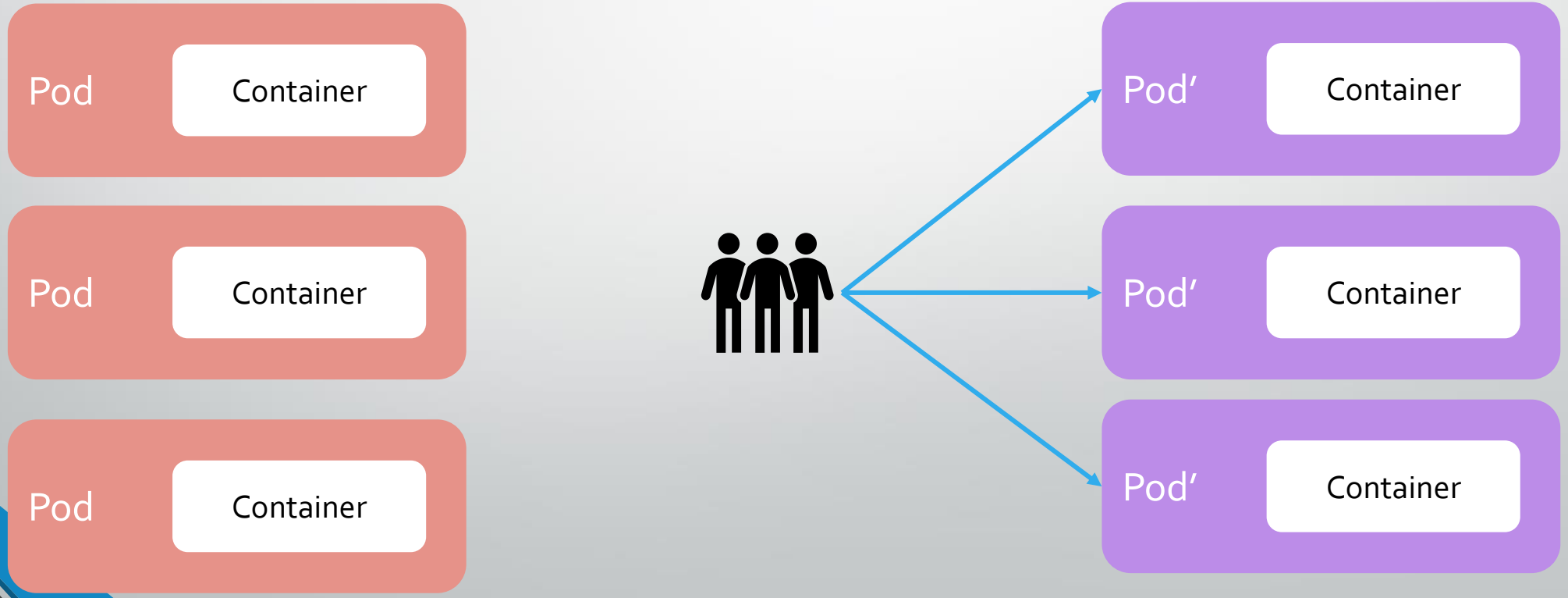
Start all new pods and then switch



Deployment Strategies

Blue Green deployment

Start all new pods and then switch



Deployment Strategies

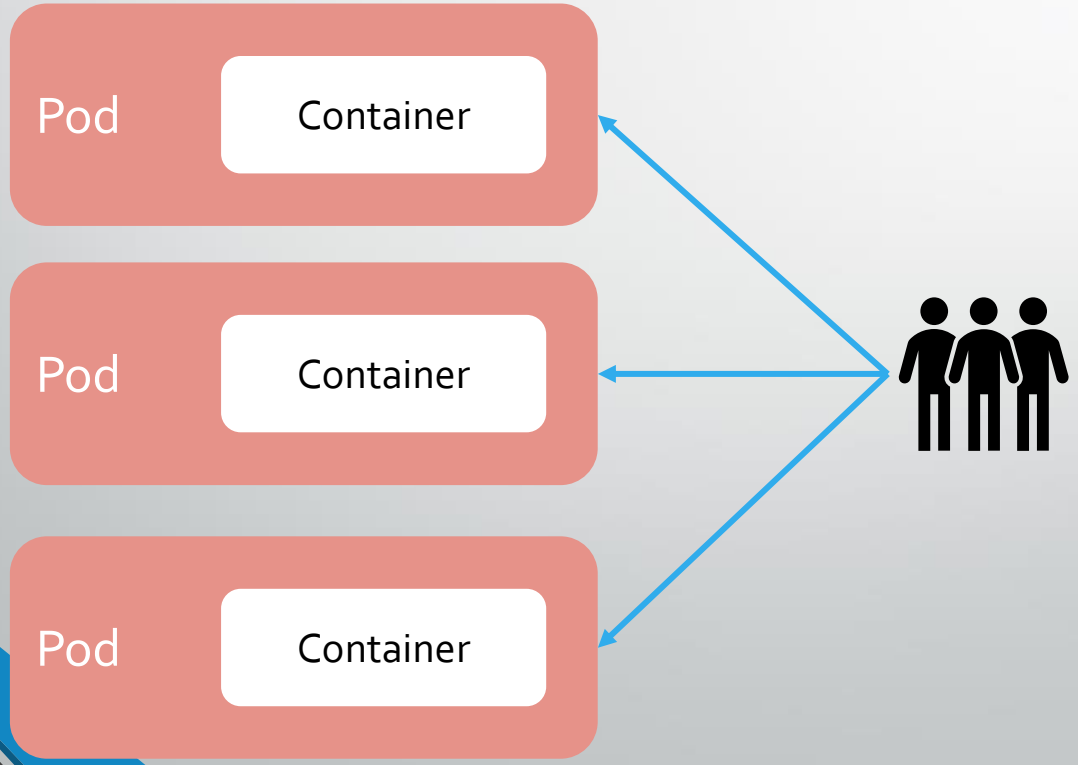
Rolling deployment

Replace old pods with new ones until all are replaced

Deployment Strategies

Rolling deployment

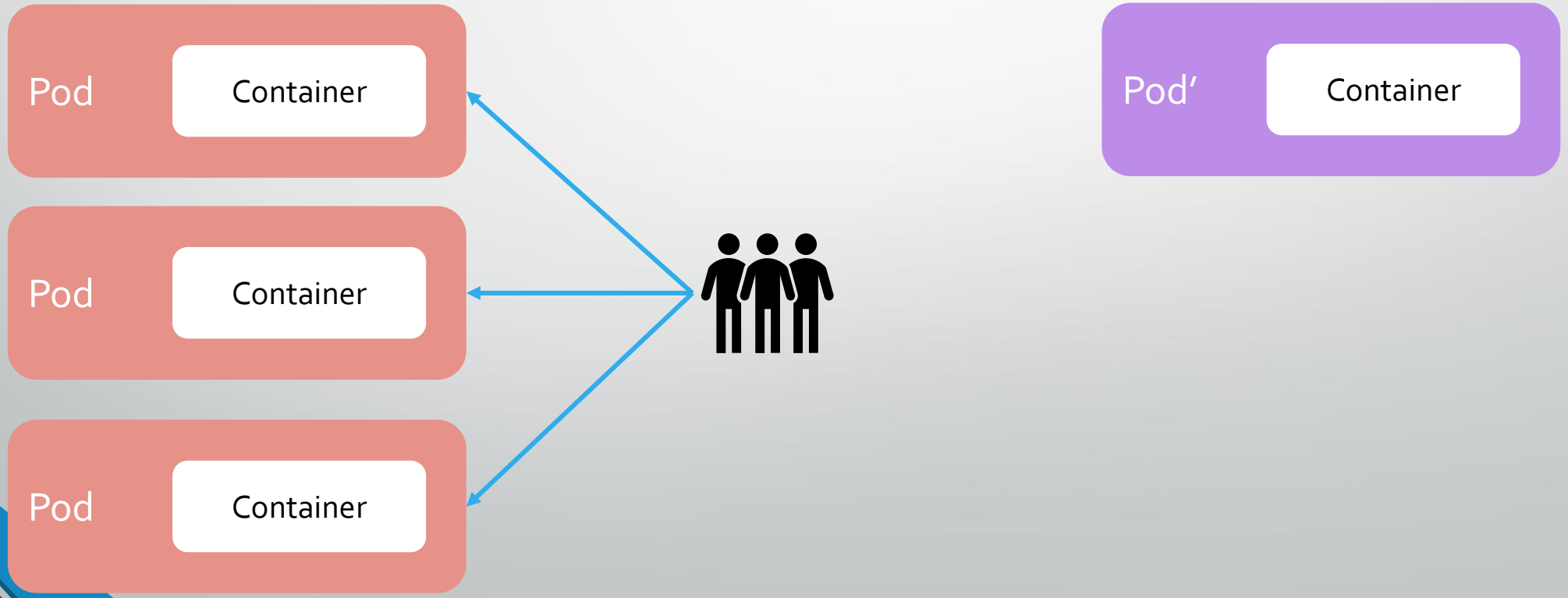
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

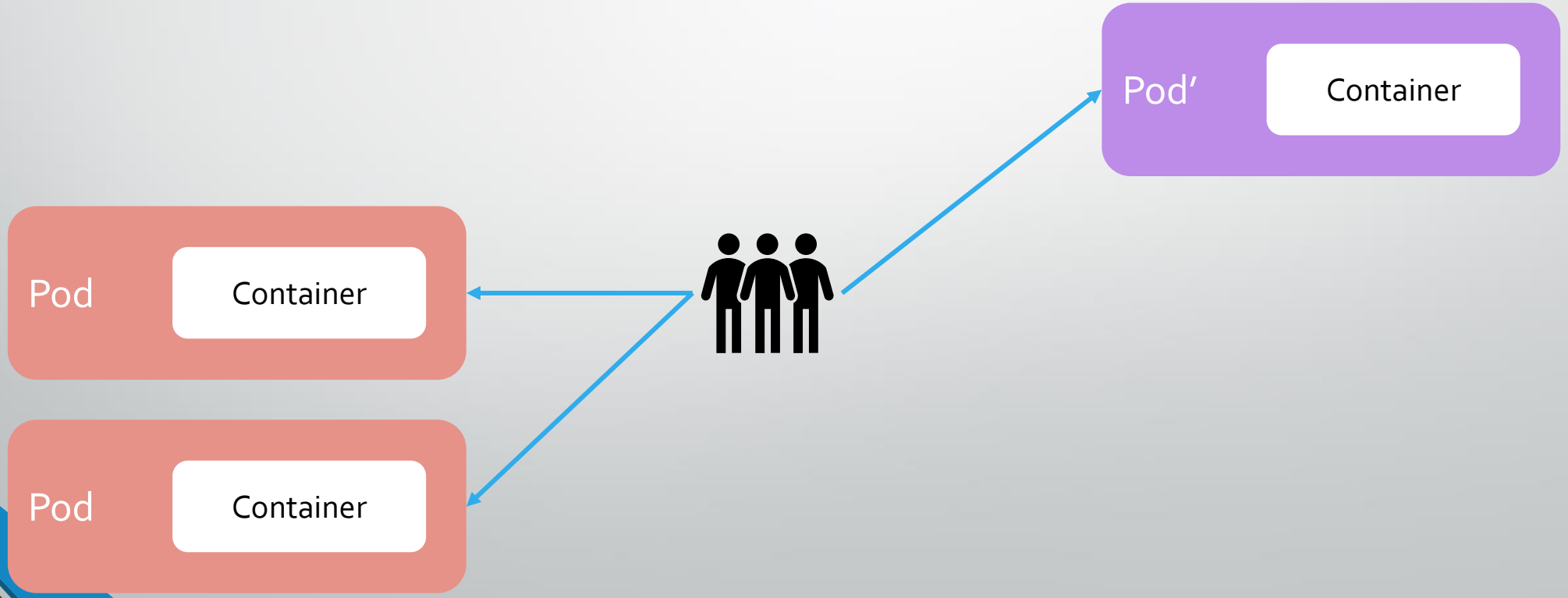
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

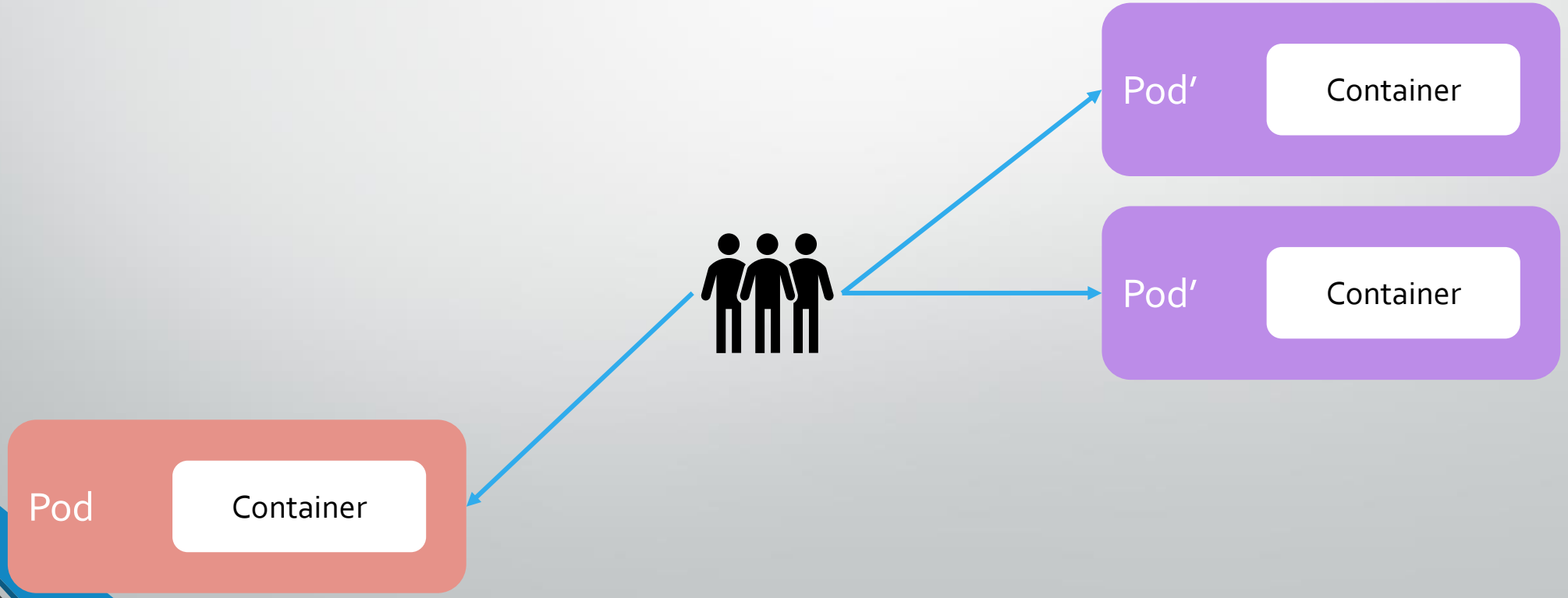
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

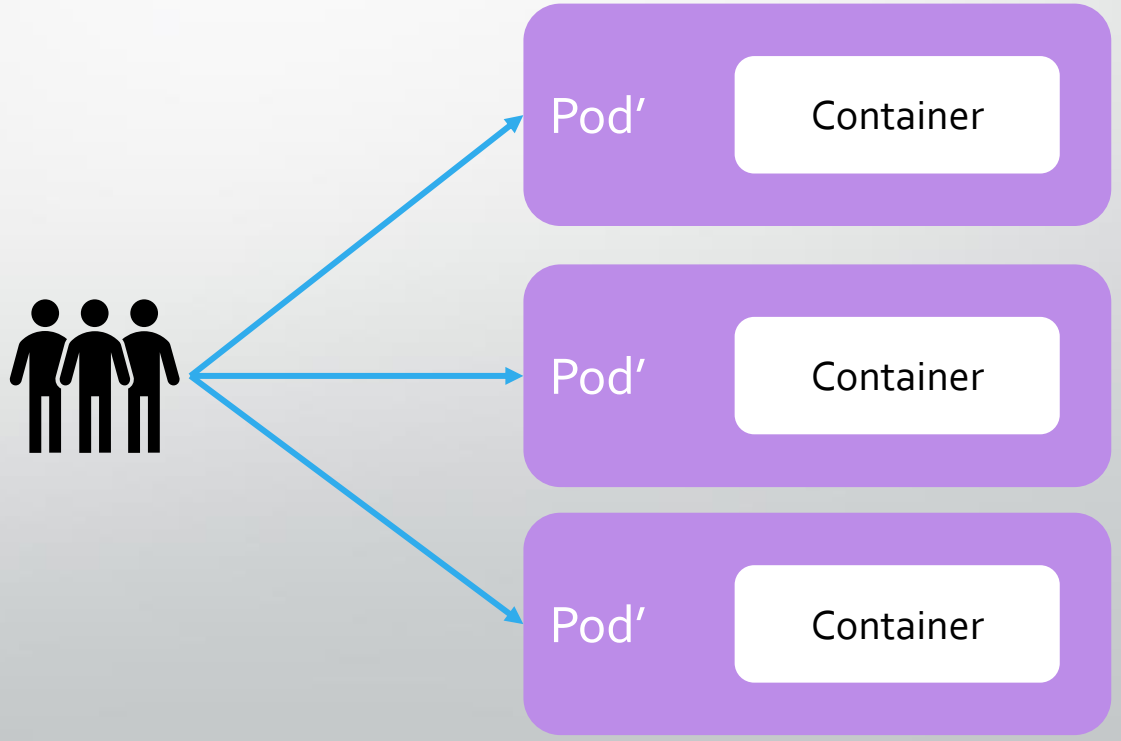
Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

Replace old pods with new ones until all are replaced



Deployment Strategies

Rolling deployment

- Default deployment mode
- Application must support two versions

Deployment Strategies

Canary deployment

- Route only a small percentage of traffic to the new version
- Reduce the blast radius of a bad deployment
- Increase percentage of traffic gradually
- 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%

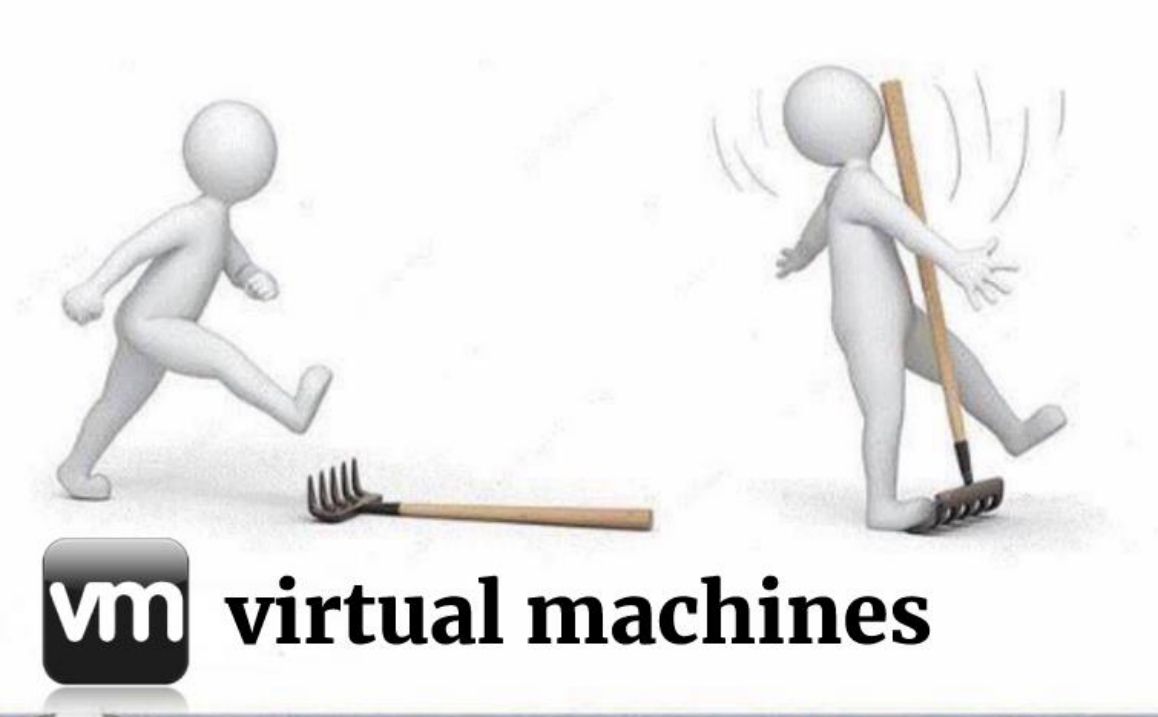
Deployment Strategies

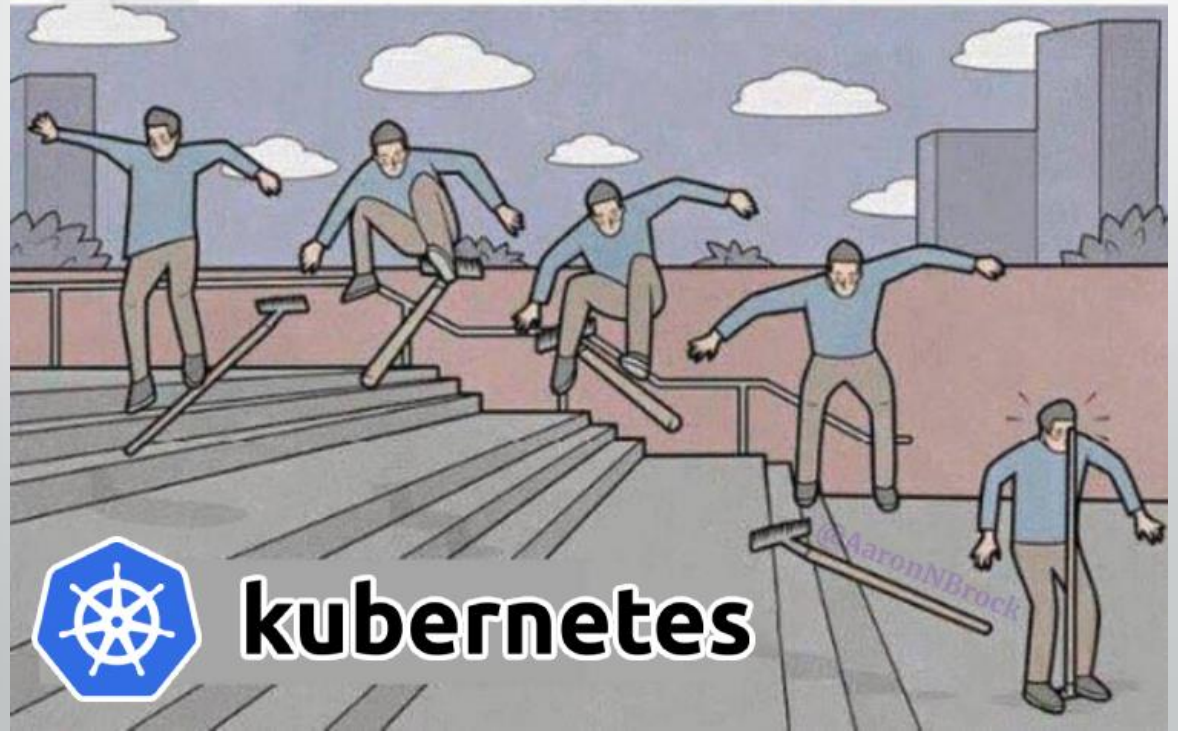
maxSurge:

- 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%


vm
virtual machines


kubernetes
@AaronNBrock

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

Exercice

Deploy your own Image from Dockerhub

Change ports in the Service

Deploy a new version of your container

Change Service Type to ClusterIP

Implement an HPA and test autoscaling

Readiness / Liveness Probe

Create an AKS cluster

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

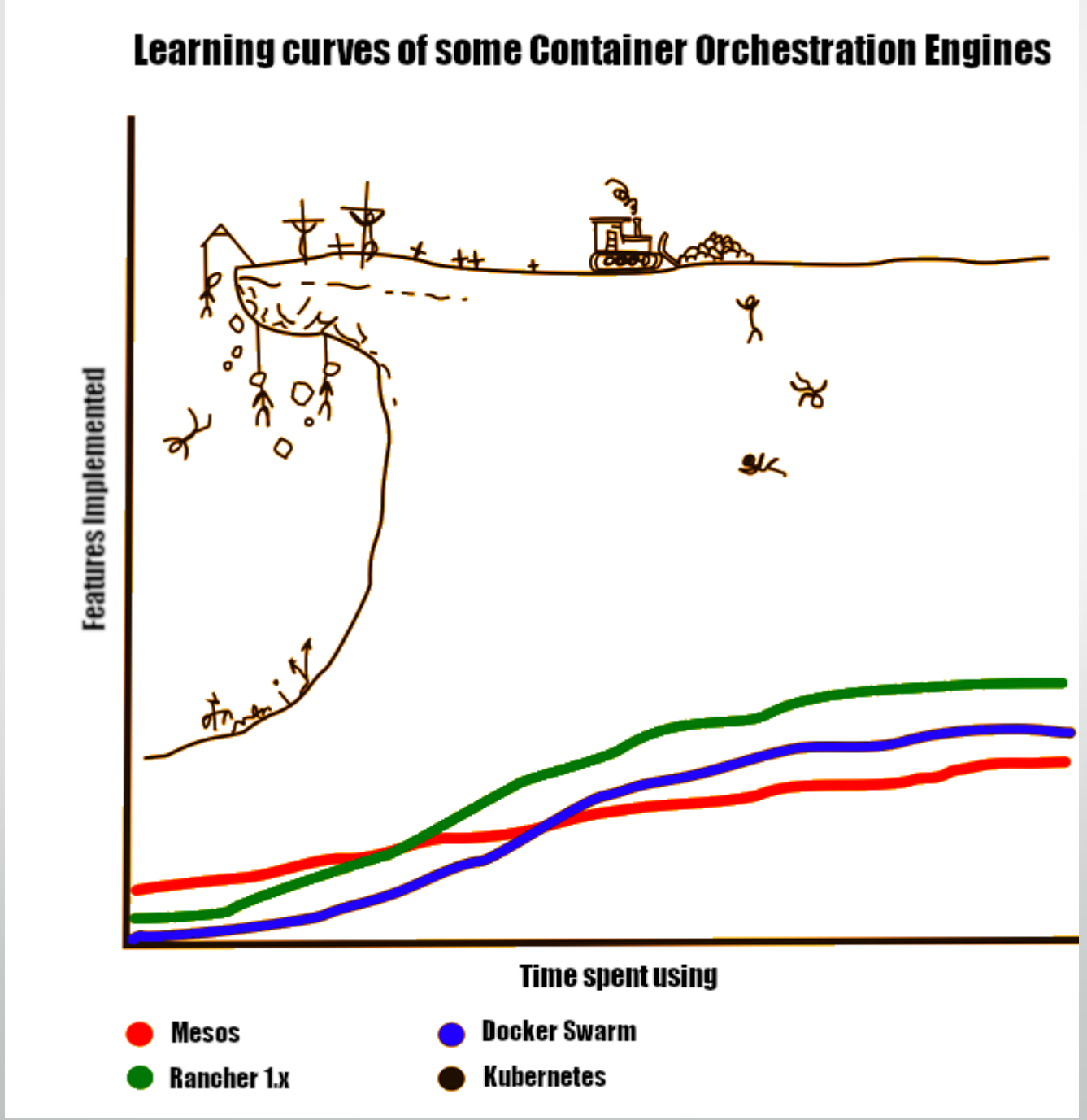
kubectl apply –f myfile.yaml [namespace=my-namespace]

Set current namespace

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

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

Advanced Tools and Learning Path


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

Certification

Kubernetes Fundamentals Training (35 hours): https://training.linuxfoundation.org/training/kubernetes-fundamentals

Certified Kubernetes Administrator (CKA): https://training.linuxfoundation.org/certification/certified-kubernetes-administrator-cka

Certified Kubernetes Application Developer (CKAD): https://training.linuxfoundation.org/certification/certified-kubernetes-application-developer-ckad

Helm

Kubernetes Package Manager

Use Helm charts to define application

Share Helm charts

Open-source

Public Helm charts can be found on https://artifacthub.io

Documentation: https://helm.sh

Kustomize

Kubernetes configuration management tool

Scalable for various deployment sizes

Efficient configuration management

Enables customization without directly editing manifest files

Documentation: https://kustomize.io

GitOps

Automation using Git and CI/CD pipelines

Configuration as code

History of config changes + Pull Request reviews

ArgoCD: https://argo-cd.readthedocs.io

Flux: https://fluxcd.io

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: https://keda.sh

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: https://cilium.io

Service Mesh

Manage communication between applications and pods

Service discovery

Failure handling

Observability: metrics, logging, tracing

Security: encryption, authentication, authorization

Service Mesh

Open-source Services:

Istio: https://istio.io

Linkerd: https://linkerd.io

Consul: https://developer.hashicorp.com/consul

Secret Management

Secrets are not encrypted by default

Secrets should be encrypted to keep them secret

- Database connection strings
- Passwords
- Certificates

Secret Management

Tools:

- Azure Key Vault Provider for Secrets Store CSI Driver: https://learn.microsoft.com/en-us/azure/aks/csi-secrets-store-driver
- HashiCorp Vault: https://www.vaultproject.io
- Sealed Secrets: https://github.com/bitnami-labs/sealed-secrets

Monitoring

Metrics

Logs

Traces

Alerting

Monitoring

Tools:

- Grafana: https://grafana.com
- Prometheus: https://prometheus.io
- Elastic Stack (ELK Stack): https://www.elastic.co/elastic-stack
- Jaeger: https://www.jaegertracing.io
- Zipkin: https://zipkin.io

Azure Arc

Manage Kubernetes with Azure

Run Azure services on your Kubernetes cluster on-premises

- Azure App Service
- Azure Machine Learning
- Azure Managed SQL Database

Authentication with Azure

Access cluster which is hidden behind a firewall

https://learn.microsoft.com/en-us/azure/azure-arc/kubernetes/overview

Kubernetes Operations

Resource utilization

Tagging

Namespace limits

Pod security policies

Networking policies

RBAC policies

Logging

Checklist: https://learnk8s.io/production-best-practices

More Resources

Kubernetes: https://kubernetes.io

Docker: https://www.docker.com

K3s: https://k3s.io

Networking: https://www.cni.dev/docs

Graphics: https://shipit.dev/posts/kubernetes-overview-diagrams.html

Code: https://code.komaxgroup.com/global/training/innovation-days/2023/id23-5311-kubernetes-workshop

Exercice for the Fast Ones

Helm

- Install Helm, create a Helm chart, check content, deploy helm chart, download public Helm chart

Keda Demo

- Azure Service Bus, Scale to zero, Azure DevOps Agent Scaling

Use a dashboard to see what's going on in your cluster

Install Istio and try out the demo

Q&A