

INTRODUCTION TO KUBERNETES

Carlos Barragan, Matthias Häussler, Jonas Grundler
- NovaTec Consulting GmbH -

AGENDA

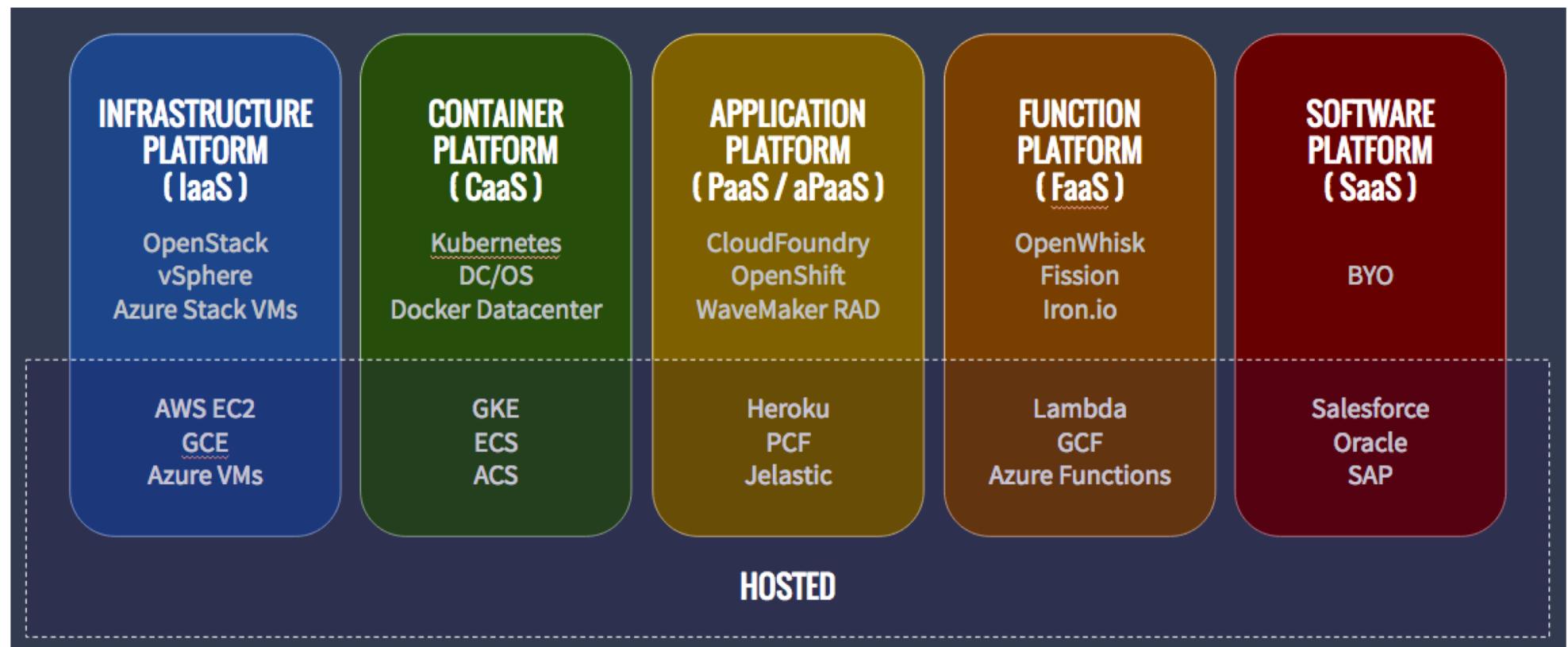
- 
- K8s Basics
 - K8s Components
 - Ingress & Load Balancing
 - Apps - Deployment

K8s Basics: Overview

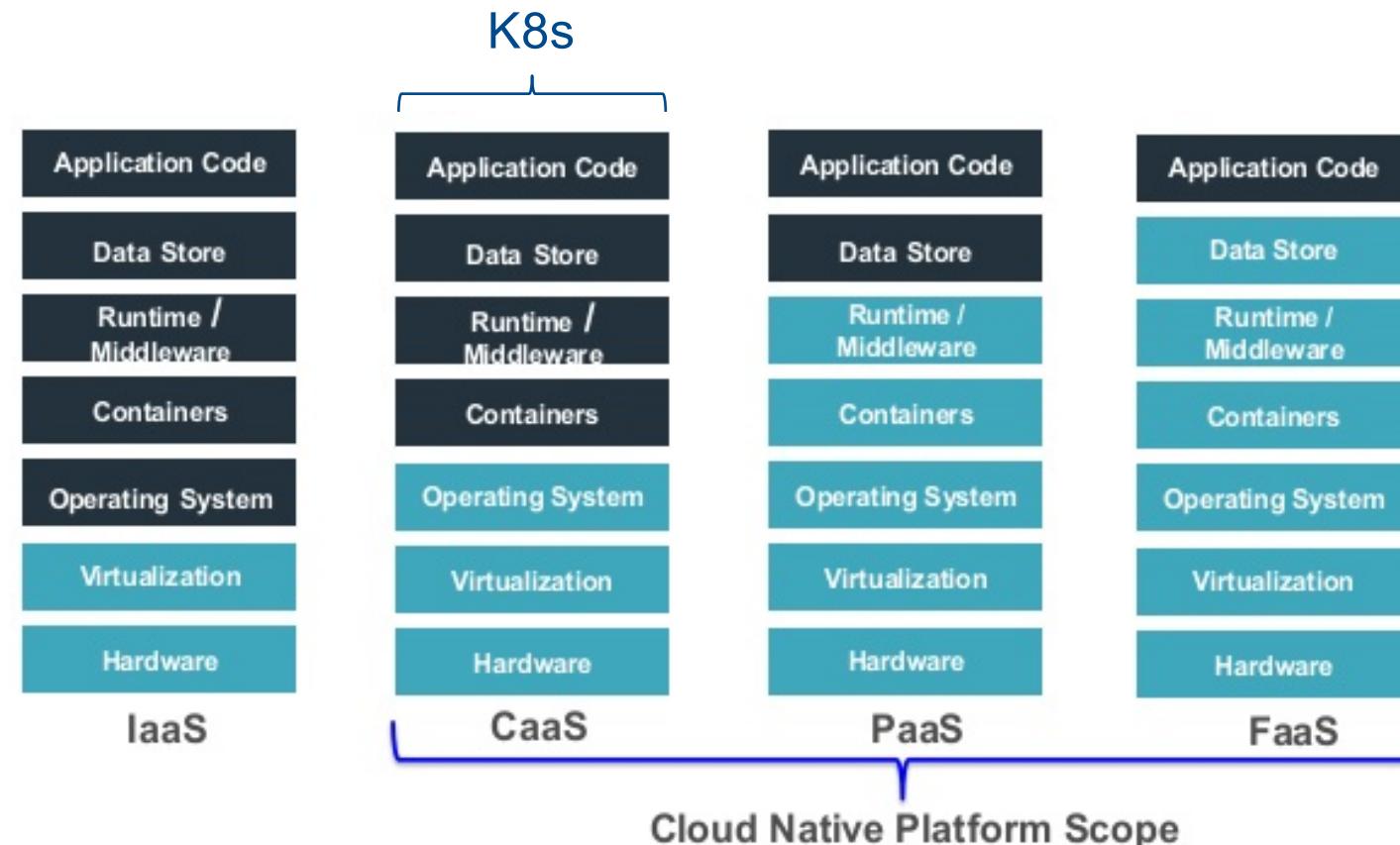
- **Kubernetes** is an open-source platform designed to automate deploying, scaling, and operating application containers
- Kubernetes v1.0 was released in 2015
- It utilizes the advantages of containerization, Continuous Integration/Delivery, DevOps, Cloud Computing and Microservices
- It provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), and enables portability across infrastructure providers
- **Docker** is the most common container runtime used in Kubernetes, but it also support other runtimes

K8s Basics: CaaS Concepts

- Provides a complete container environment for deploying and managing containers, applications and clusters
- Container Orchestration lies at the heart of container platforms
- Focus is on creating the app containers and maintaining them in the future



K8s Basics: Developer Responsibility

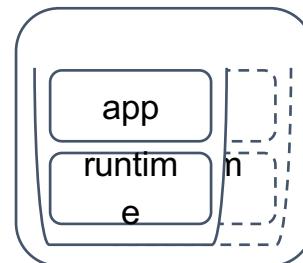


**HERE'S MY CODE
I'LL TELL YOU EXACTLY HOW YOU
SHOULD RUN IT ON THE CLOUD FOR
ME
AND DON'T YOU DARE MAKE ANY
ASSUMPTIONS ON THE DEPLOYMENT
WITHOUT MY WRITTEN CONSENT!**

(<https://medium.com/@odedia/comparing-kubernetes-to-pivotal-cloud-foundry-a-developers-perspective-6d40a911f257>)

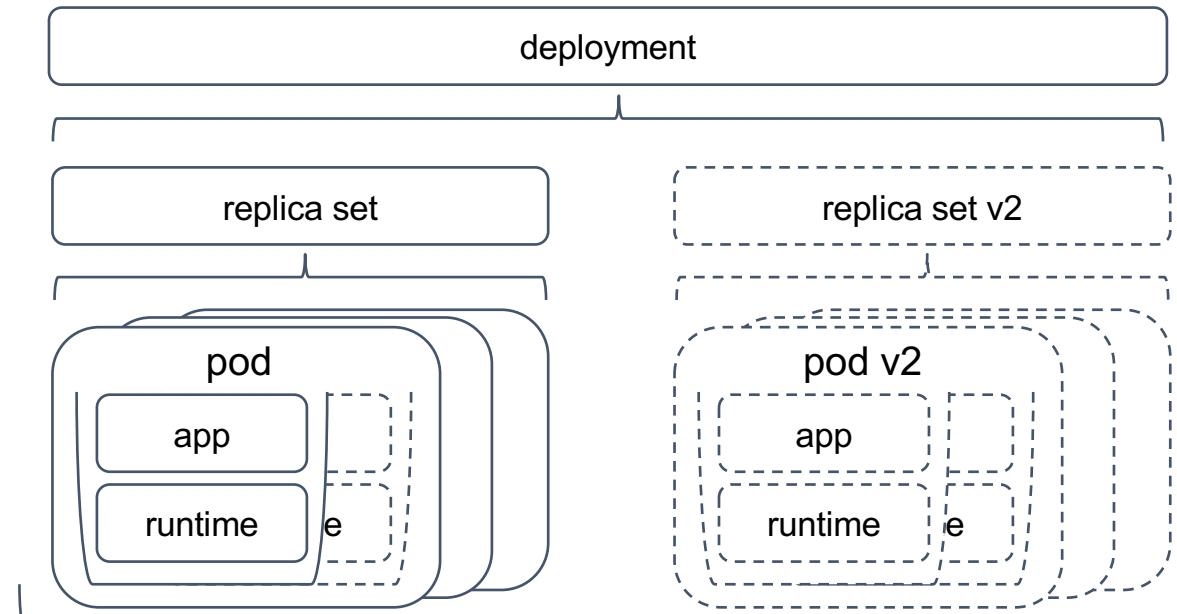
K8s Basics: Pods

- A **pod** is an abstract layer which synthesizes a certain amount of Docker containers that are meant to run together
- It represents the smallest unit of deployment, i.e., a single instance of an application in Kubernetes, which might consist of either a single container or a small number of containers that are tightly coupled
- Pods provide two kinds of shared resources for their constituent containers: *networking* and *storage*
- Containers within a pod share the same IP address, port space, hostname and storage, and can find each other via localhost



K8s Basics: Deployments

- A **deployment** is a description of the desired condition of a particular microservice (number of instances, port forwarding, docker image ...)
 - **ReplicaSets** are (usually) combined with pods when defining the deployment
 - A ReplicaSet ensures that a specified number of pod replicas are running at any given time, thus ReplicaSets enable pods to scale horizontally



K8s Basics: Services

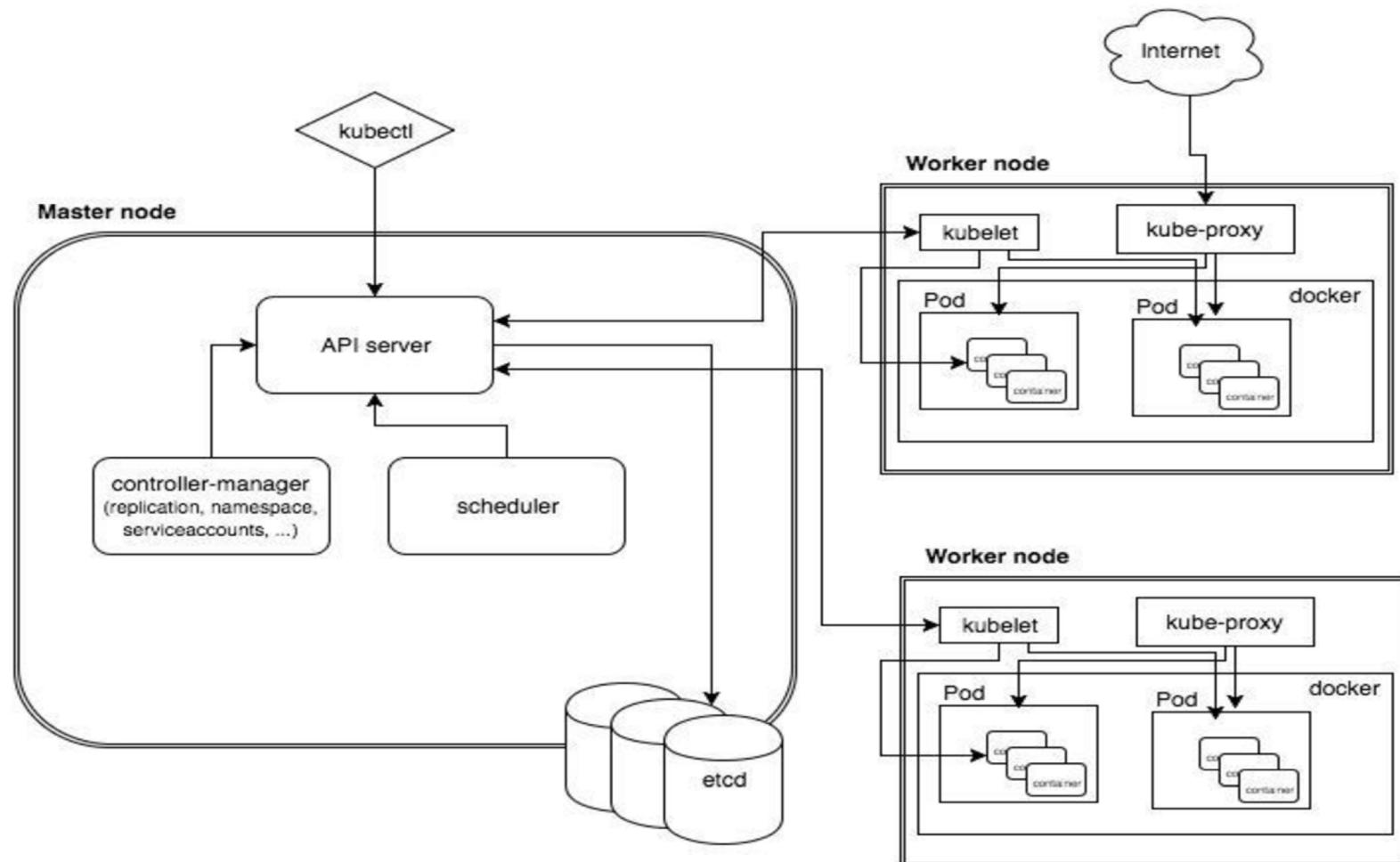
- A **service** is an abstract layer for exposing pods functionality
- It defines a logical set of pods and a policy by which to access them, sometimes referred to as a microservice
- It is defined in K8s as a REST object



AGENDA

- 
- K8s Basics
 - K8s Components
 - Ingress & Load Balancing
 - Apps - Deployment

K8s Components (1/2)



K8s Components (2/2)

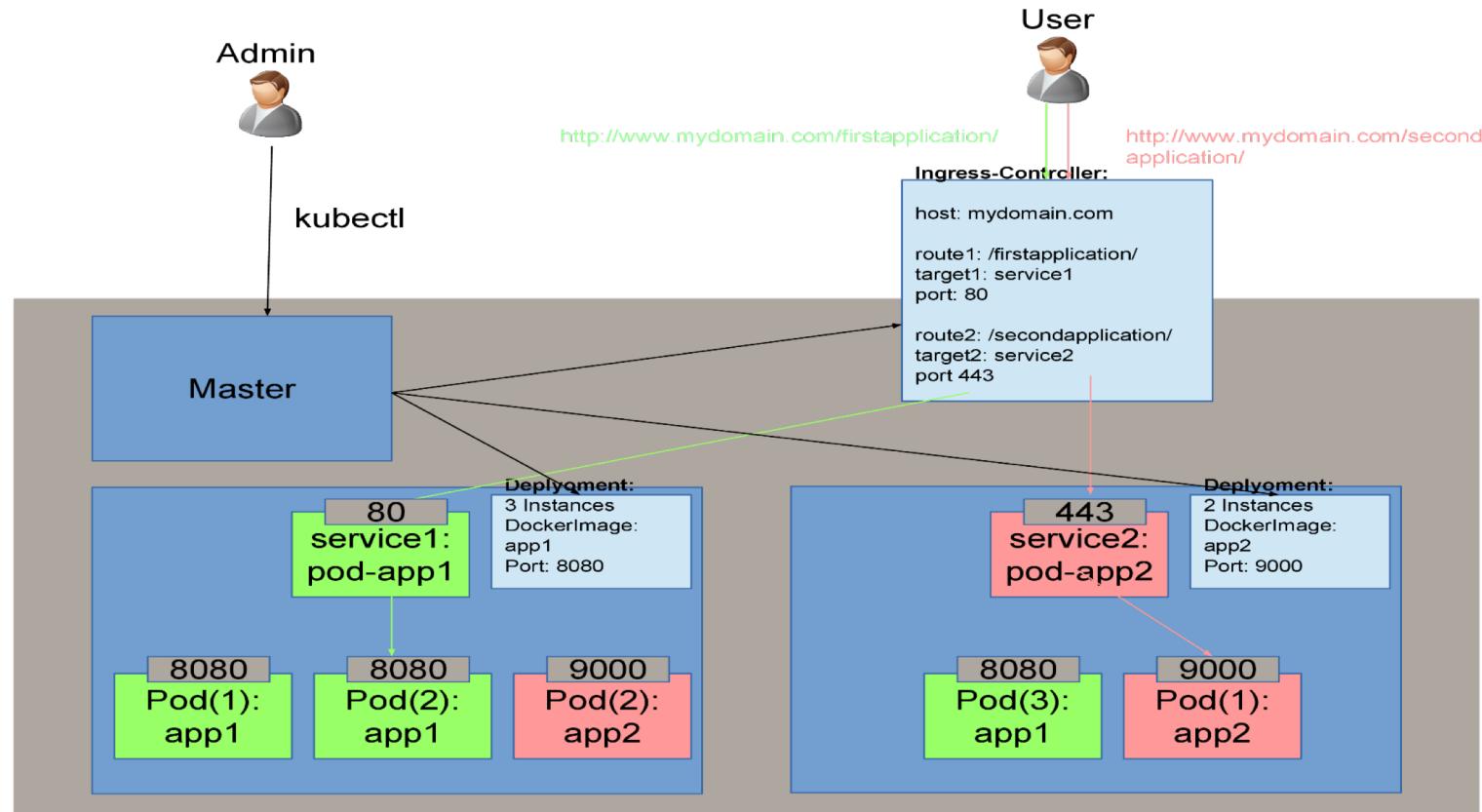
- **Cluster:** It is a logical unit, consisting of one Master node and 1..n Worker nodes
- **Master Node:** It is the server that is responsible for the Kubernetes management tasks
 - It is composed of three processes: the API service, the controller-manager and the scheduler
 - The API server exposes a highly-configurable REST interface to all of the Kubernetes resources
 - The scheduler's main responsibility is to place the containers on the node in the cluster according to various policies, metrics, and resource requirements
 - The controller manager is responsible for reconciling the state of the cluster with the desired state, as specified via the API
- **Worker Nodes:** They are the worker machines and are managed by the master components
 - Each node runs three services: the kubelet, the kube-proxy and the Docker engine
 - The kubelet interacts with the underlying Docker engine to bring up containers as needed
 - The kube-proxy is in charge of managing network connectivity to the containers

AGENDA

- 
- K8s Basics
 - K8s Components
 - Ingress & Load Balancing
 - Apps - Deployment

Ingress & Load Balancing

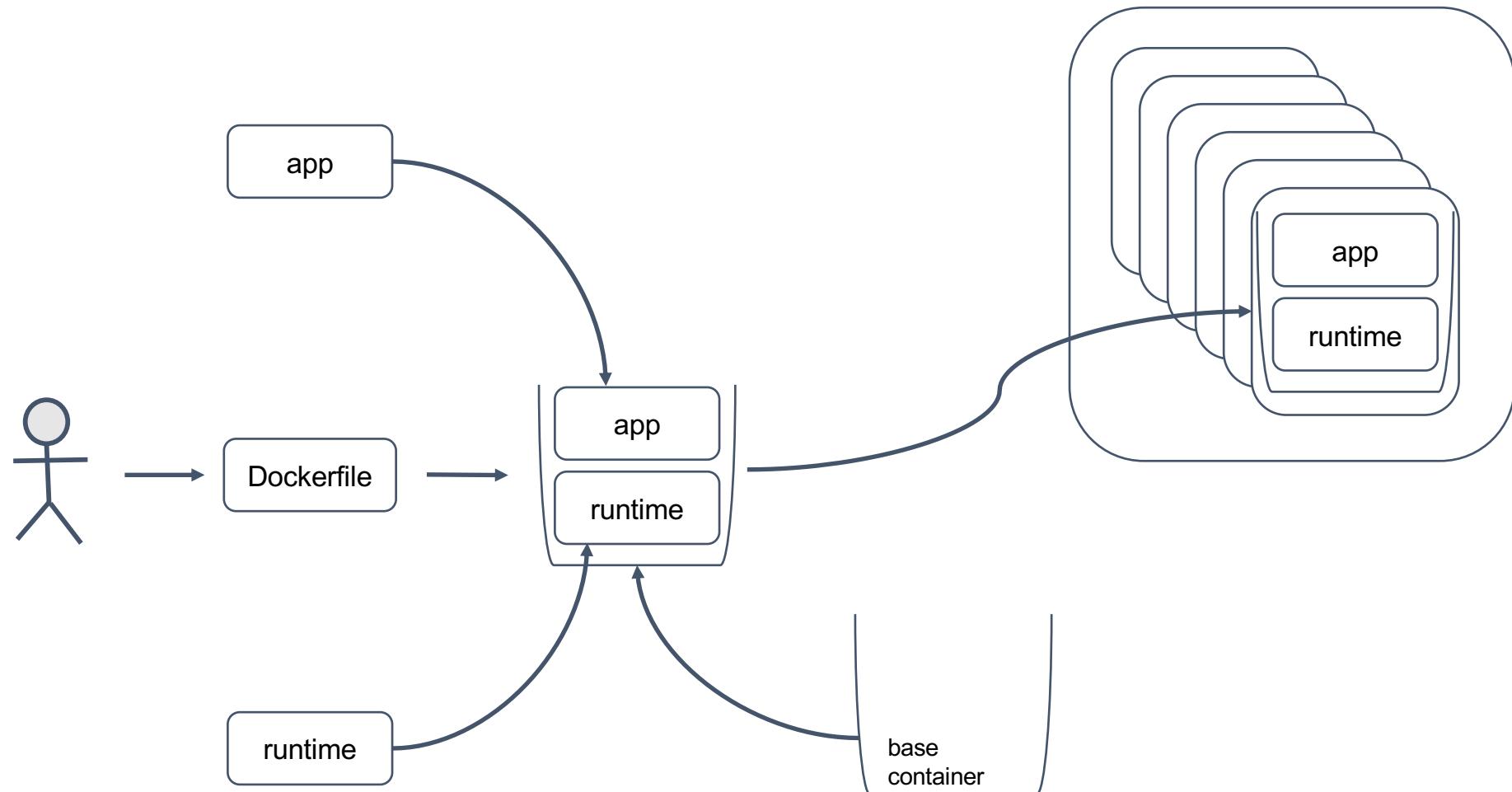
- **Ingress:** It is an API object that manages external access to the services in a cluster, typically HTTP
 - It can provide load balancing along with an Ingress-controller, which will do the actual routing of the incoming traffic



AGENDA

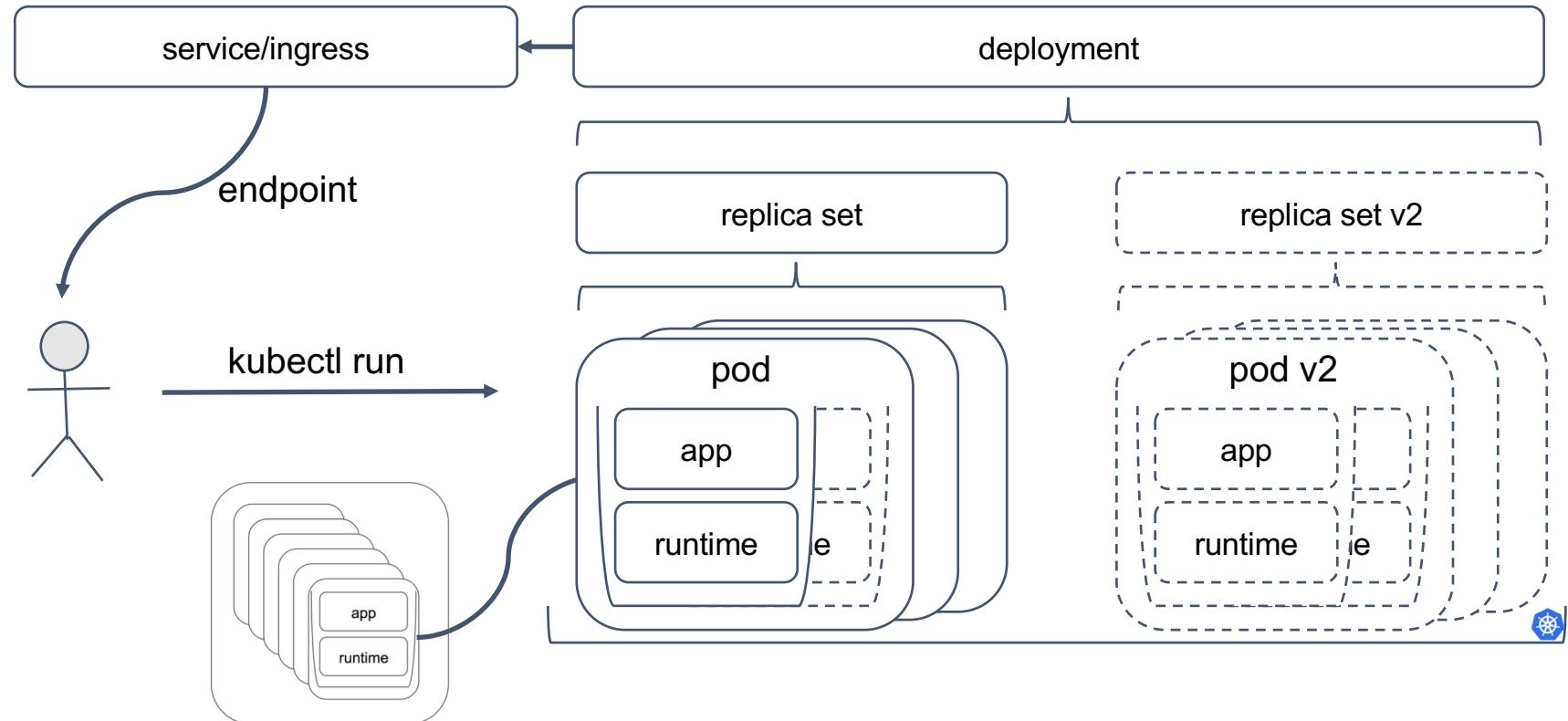
- 
- K8s Basics
 - K8s Components
 - Ingress & Load Balancing
 - Apps - Deployment

Apps – Deployment (1/6)



(Matthias Haeussler-CF Summit 2018)

Apps – Deployment (2/6)



(Matthias Haeussler-CF Summit 2018)

Apps – Deployment (3/6)

The steps needed in order to deploy a simple spring java app to K8s:

- Write a Dockerfile for the app

```
FROM openjdk:8-alpine
RUN mkdir -p /opt/simple-app
WORKDIR /opt/simple-app
COPY target/simple-app-0.0.1-SNAPSHOT.jar /opt/simple-app
CMD ["java", "-Dspring.profiles.active=k8s", "-jar", "simple-
app-0.0.1-SNAPSHOT.jar"]
```

Apps – Deployment (4/6)

- Create the Service & Deployment Yaml files for the app

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  namespace: simple-cn-app
  name: simple-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: simple-app
  template:
    metadata:
      labels:
        app: simple-app
```

```
spec:
  containers:
    - name: simple-app
      image: mirna/simple-app
      ports:
        - containerPort: 9082
      env:
        - name: SERVER_PORT
          value: "9082"
```

Apps – Deployment (5/6)

- Configure the application properties in the application.yml file based on the deployment Yaml

```
server:  
  port: ${SERVER_PORT}
```

- Frame the dependency tree & download the needed components

```
mvn clean install
```

- Build a Docker image for the app using its Dockerfile

```
docker build -t simple-app .
```

- Push the created images into a Docker registry

```
docker login
```

```
docker push simple-app
```

Apps – Deployment (6/6)

- Push the configuration changes to the K8s cluster and create the resources

```
kubectl apply -f simple-app\deployment.yml
```

- Check the created pods and services

```
kubectl get pods –namespace simple-cn-app
```

```
kubectl get services –namespace simple-cn-app
```

INTRODUCTION TO KUBERNETES

Questions?