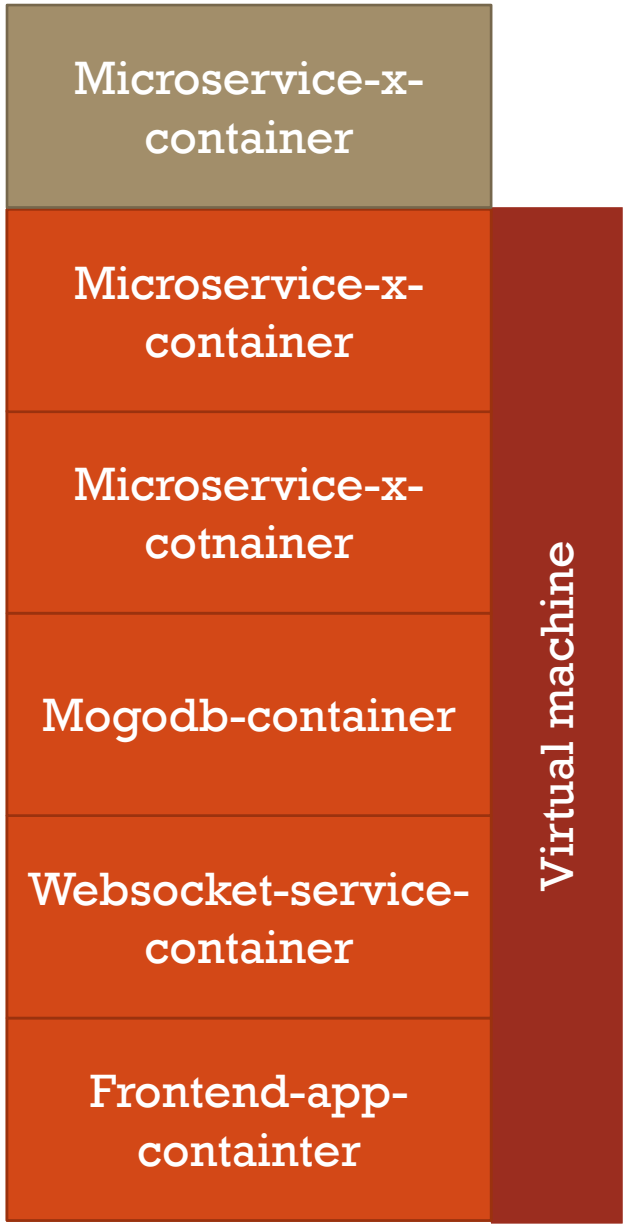
kubernetes

Jasper Vercnocke

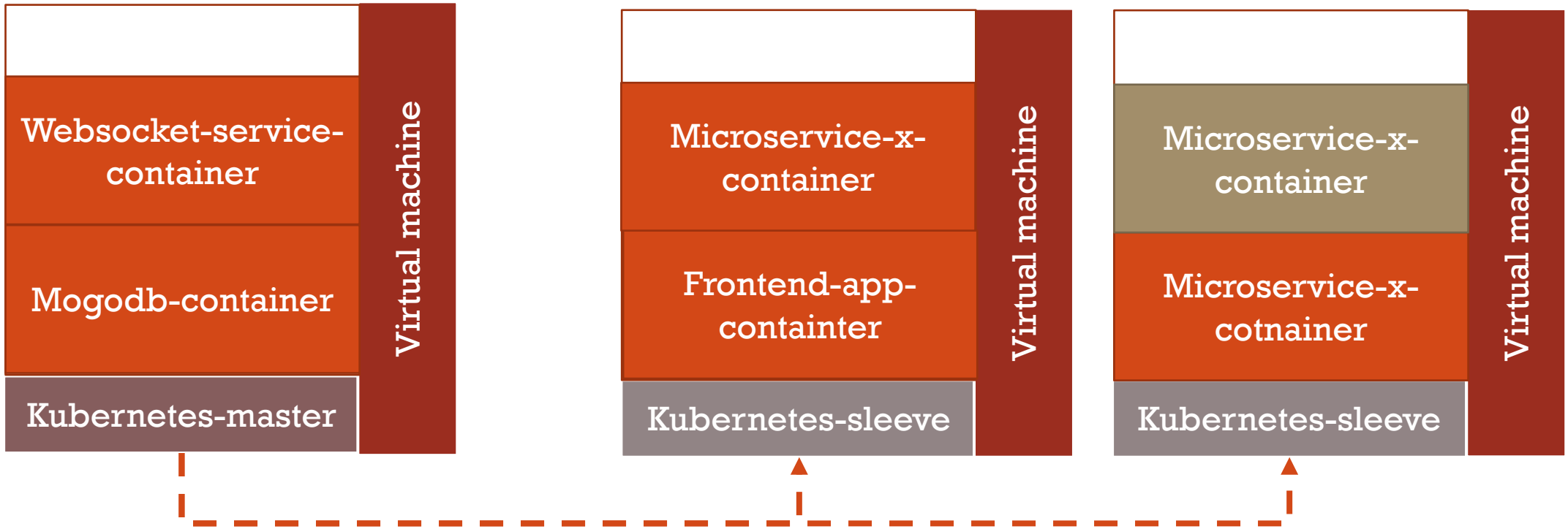
OVERVIEW

CLASSIC



OVERVIEW

KUBERNETES



DEFINITION OF KUBERNETES

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation

IN-DEPTH: POD

A Pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers

io.k8s.api.core.v1.Pod (v1@pod.json)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
    resources:
      limits:
        memory: "128Mi"
        cpu: "500m"
```

IN-DEPTH: DEPLOYMENT

A Deployment provides declarative updates for Pods and ReplicaSets.

io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
```

IN-DEPTH: PERSISTENTVOLUME

A *PersistentVolume* (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes

io.k8s.api.core.v1.PersistentVolume (v1@persistentvolume.json)

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
```

IN-DEPTH: PERSISTENTVOLUMECLAIM

A *PersistentVolumeClaim* (PVC) is a request for storage by a user.

Claims can request specific size and access modes (ReadWriteOnce, ReadOnlyMany or ReadWriteMany, see AccessModes).

io.k8s.api.core.v1.PersistentVolumeClaim (v1@persistentvolumeclaim.json)

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

IN-DEPTH: VOLUMEMOUNTS

io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      volumes:
        - name: task-pv-storage
          persistentVolumeClaim:
            claimName: task-pv-claim
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
          volumeMounts:
            - mountPath: "/usr/share/nginx/html"
              name: task-pv-storage
          resources:
            limits:
              memory: "128Mi"
              cpu: "500m"
```

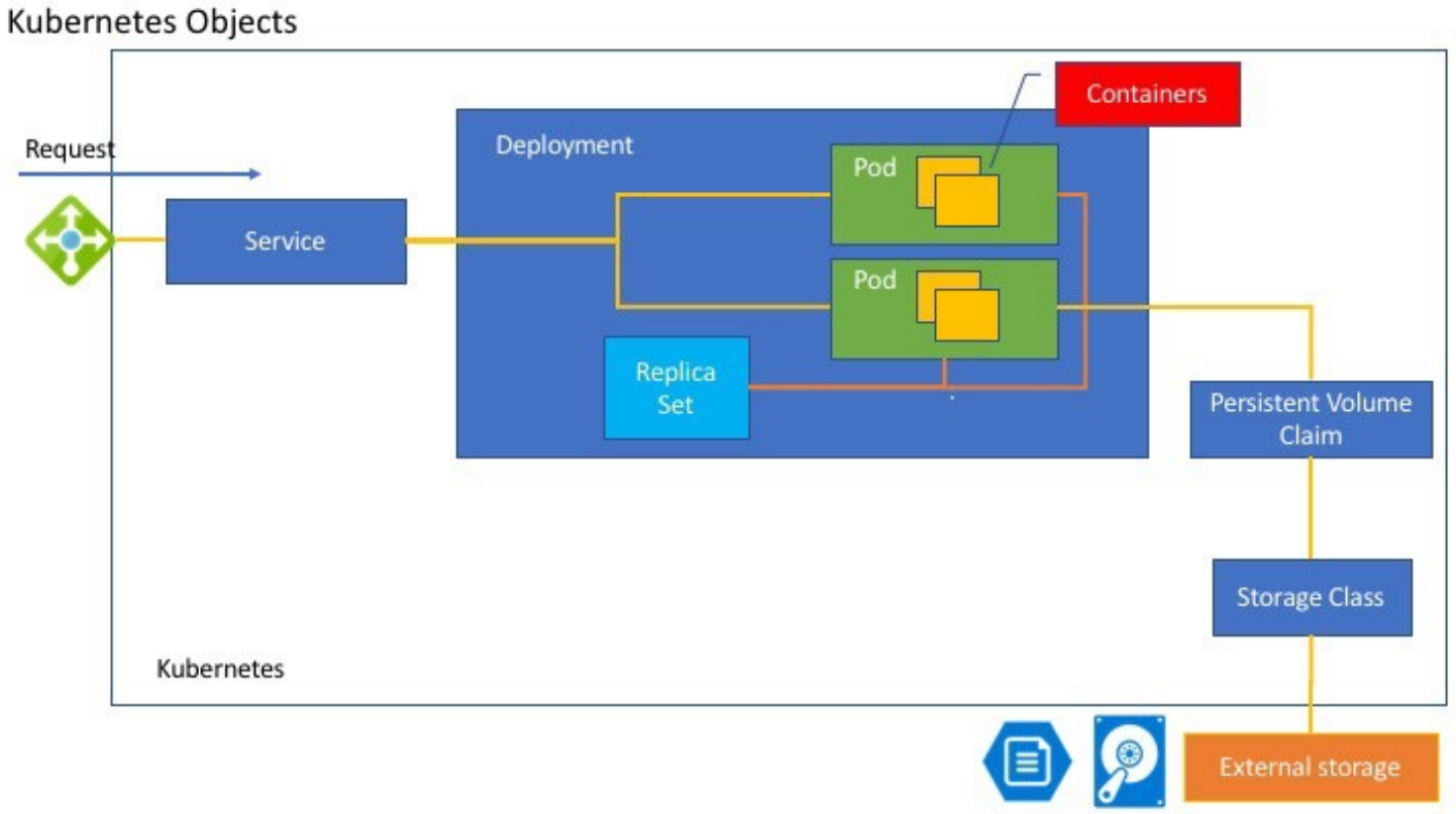
IN-DEPTH: SERVICE

A Service is an abstraction which defines a logical set of Pods and a policy by which to access them

```
io.k8s.api.core.v1.Service (v1@service.json)
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
  - name: name-of-service-port
    protocol: TCP
    port: 80
    targetPort: 80

```

IN-DEPTH: OVERVIEW



https://tsuyoshiushio.medium.com/kubernetes-in-three-diagrams-6aba8432541c

CONFIGMAP

A ConfigMap is an API object used to store non-confidential data in key-value pairs. Pods can consume ConfigMaps as environment variables, command-line arguments, or as configuration files in a volume.

io.k8s.api.core.v1.ConfigMap (v1@configmap.json)

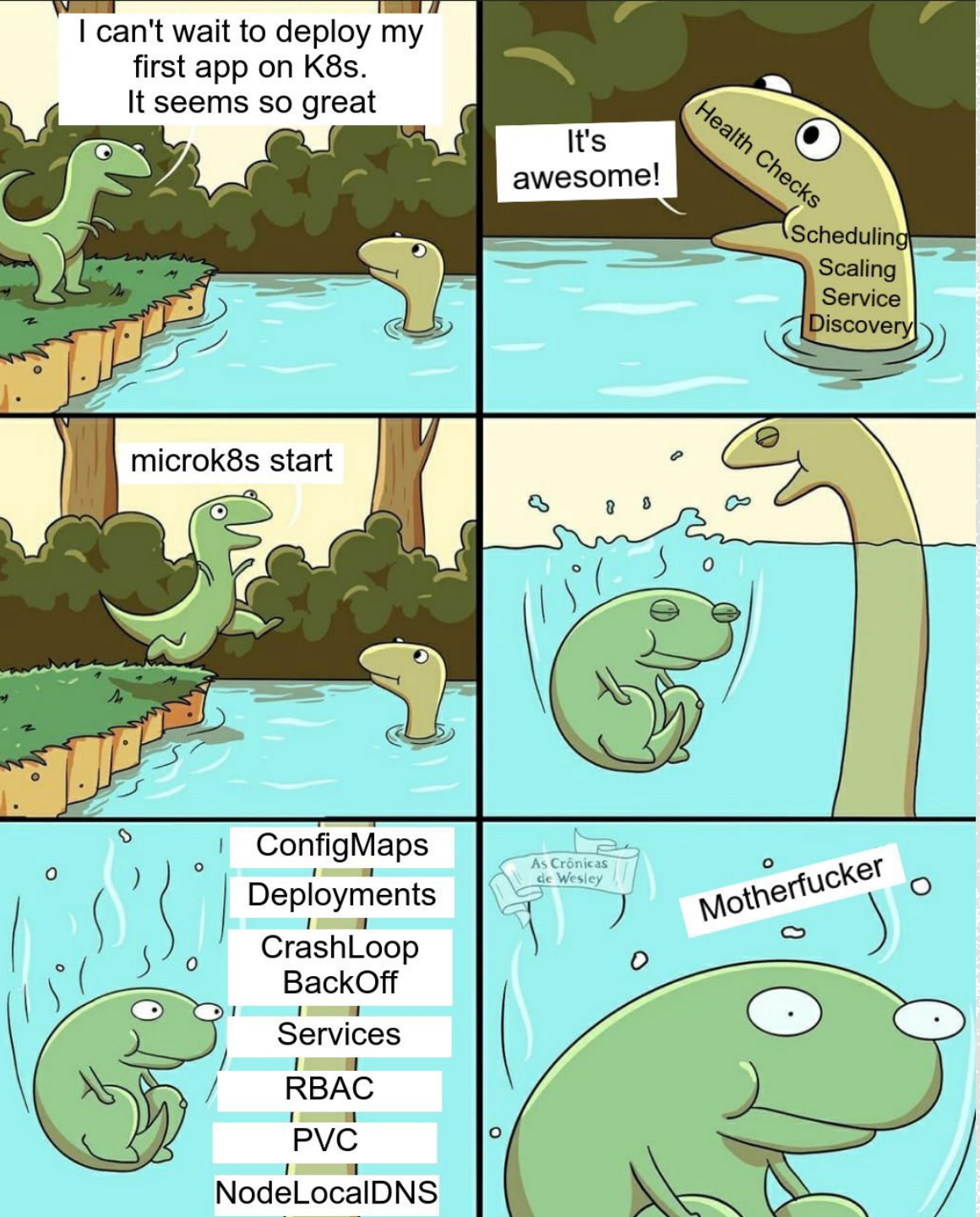
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: env
data:
  {
    "API_URL": "https://api.bootcamp.chat.nexiosdevios.be"
  }

```

SECRETS

A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key

```
io.k8s.api.core.v1.Secret (v1@secret.json)
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: docker_credentials
5  type: Opaque
6  data:
7    username: YWRtaW4=
8    password: MWYyZDFlMmU2N2Rm
```



HANDS-ON

https://github.com/NexiosIT/bootcamp-kubernetes