



云原生边缘计算公开课

Cloud Native Edge Computing

程庞钢

顺丰科技



边缘应用管理及常见问题定位

Cloud Native Edge Computing

- 云原生边缘计算公开课 -



讲师介绍

姓名 程庞钢

单位 顺丰科技

长期从事IT软件研发和项目管理工作，擅长高并发架构设计。

2014年入职顺丰，任边缘容器负责人（高工），从事容器技术设计研发。

在边缘容器技术引入、落地、实施有完整的实践经验。

目录

Contents

01 应用基本概念

02 边缘应用部署原理与演示

01

应用基本概念

KubeEdge.io

应用基本概念

- Pod

- 一组功能相关的Container的封装
- K8s创建、调度和管理的最小部署单元
- 同一Pod里的容器共享同一个网络命名空间、IP地址以及端口空间

- Workloads

一组功能相关的Pod的封装

- **ReplicaSet**: 副本控制器，用于解决Pod的扩容和缩容问题
- **Deployment**: 为Pod和ReplicaSet提供声明式更新。可以在Deployment对象中指描述您所期望的理想状态（预期的运行状态），Deployment控制器为您将现在的实际状态转换为您期望的状态。
- **StatefulSet**: 有状态应用。用于解决各个Pod实例独立声明周期管理，提供各个实例的启动顺序和唯一性
- **DaemonSet**: 能够让所有或者一些特点的Node节点运行同一个Pod

应用基本概念

• 健康检查

有命令行、httpGet请求以及TCP Socket方式

- 业务探针 (Readiness Probe) : 探针异常时不会重启容器, 只会拿掉服务后端的endpoints
- 存活探针 (Liveness Probe) : 探针异常时会重启容器

• 外部输入

- 环境变量: 可以通过key-value或者envFrom配置文件/密钥两种方式
- ConfigMap: 以卷形式挂载到容器内使用, 用于存储应用使用的配置数据
- Secret: 用于存储应用需要的敏感信息

```
spec:
  containers:
    - livenessProbe:
        failureThreshold: 5
        exec:
          command:
            - "/bin/sh",
            - "-c"
            - "echo 'iam ok'"
          initialDelaySeconds: 60
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /readiness
            port: 8081
            scheme: HTTP
          initialDelaySeconds: 3
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
```

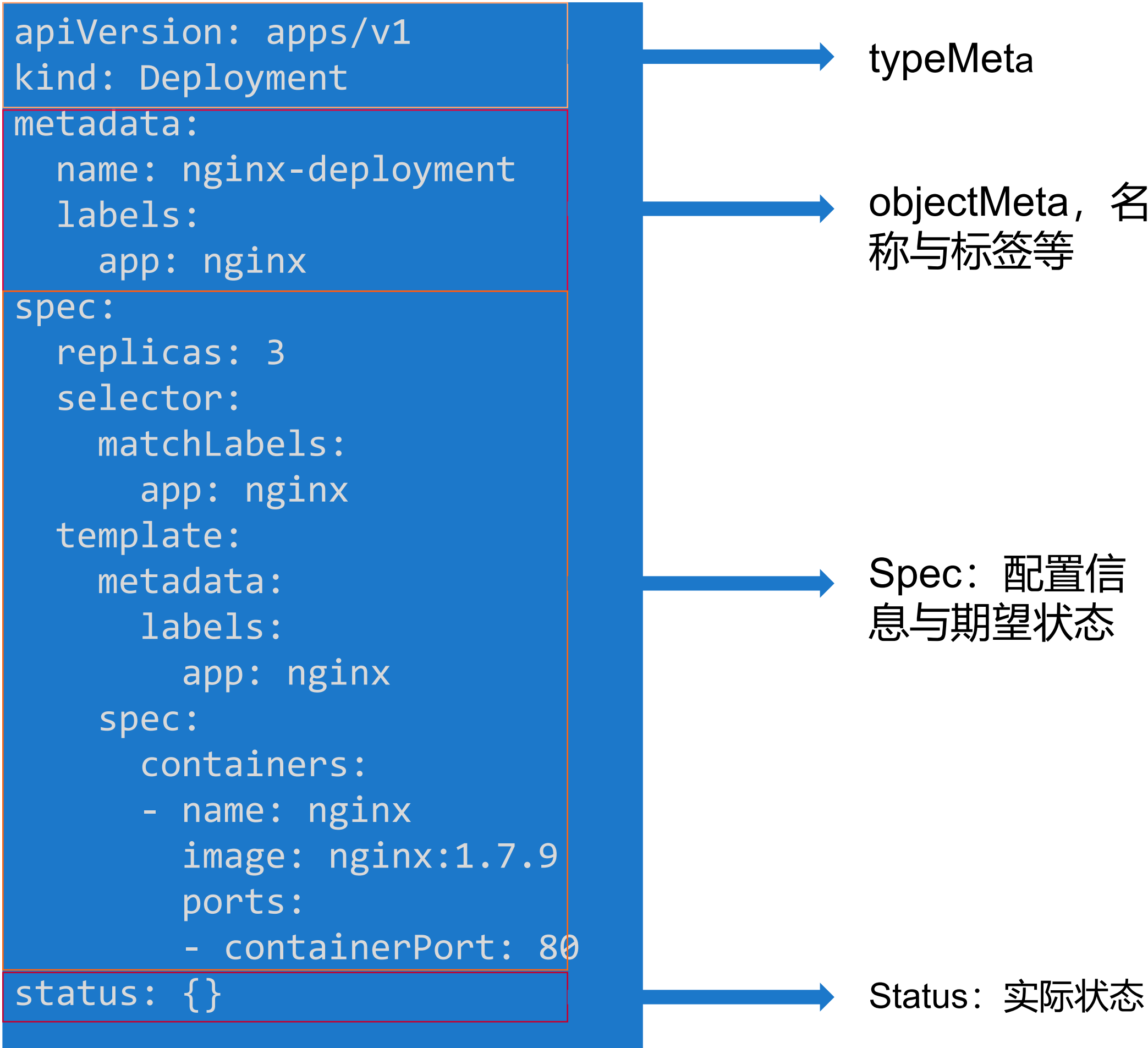
```
spec:
  containers:
    - env:
        - name: APP_NAME
          value: test
        - name: USER_NAME
          valueFrom:
            secretKeyRef:
              key: username
              name: secret
        envFrom:
        - configMapRef:
            name: config
        volumeMounts:
        - mountPath: /usr/local/config
          name: cfg
        - mountPath: /usr/local/secret
          name: sct
        volumes:
        - configMap:
            defaultMode: 420
            items:
            - key: age
              path: age
              name: config
            name: cfg
        - name: sct
          secret:
            defaultMode: 420
```

应用基本概念

- **Label**: 用于资源分类，以key-value形式标记，可以在创建资源时设置，也可以在后期添加和修改。
- **Node Affinity**: 节点亲和，让Pod在一组指定的Node运行
- **Pod Affinity**: Pod亲和。让Pod与指定Service的一组Pod在相同Node上运行
- **Pod Anti-Affinity**: Pod反亲和。让同一Service的Pod分在不同Node上运行
- **Taints**: 污点。使节点排斥一类特定的Pod
- **Toleration**: 容忍度。允许调度器调度带有对应污点的Pod

```
spec:
  nodeSelector:
    gpu: true
```

```
spec:
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        ...
```

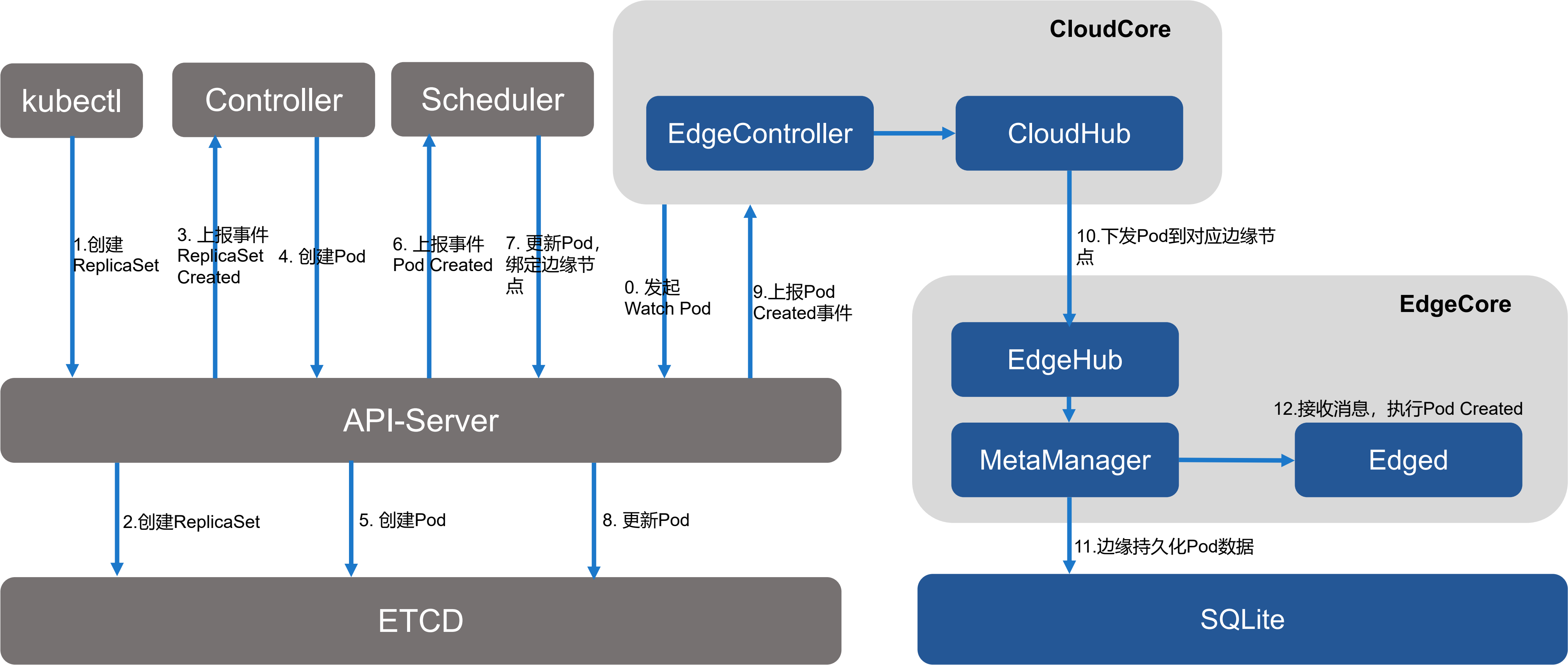


02

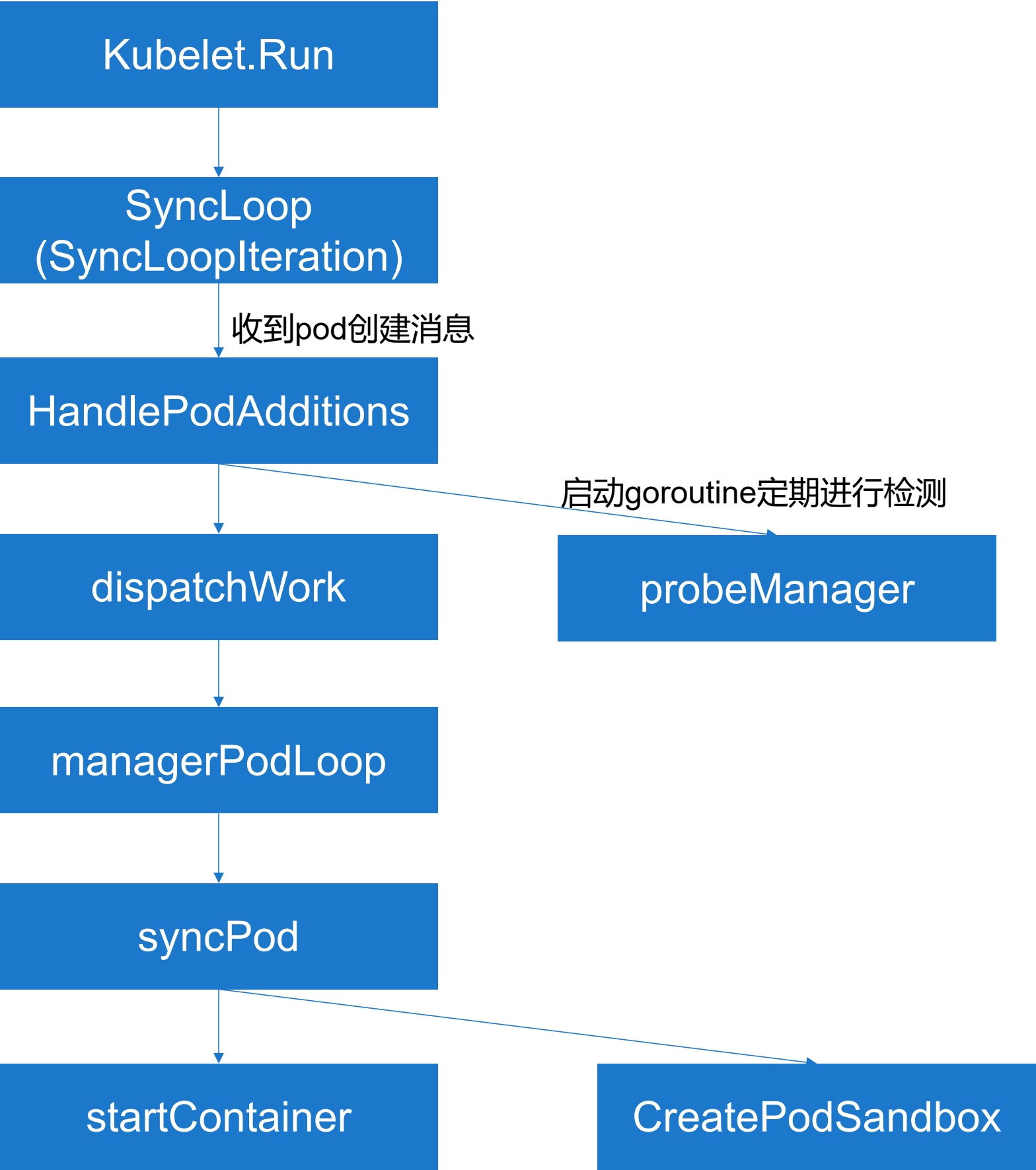
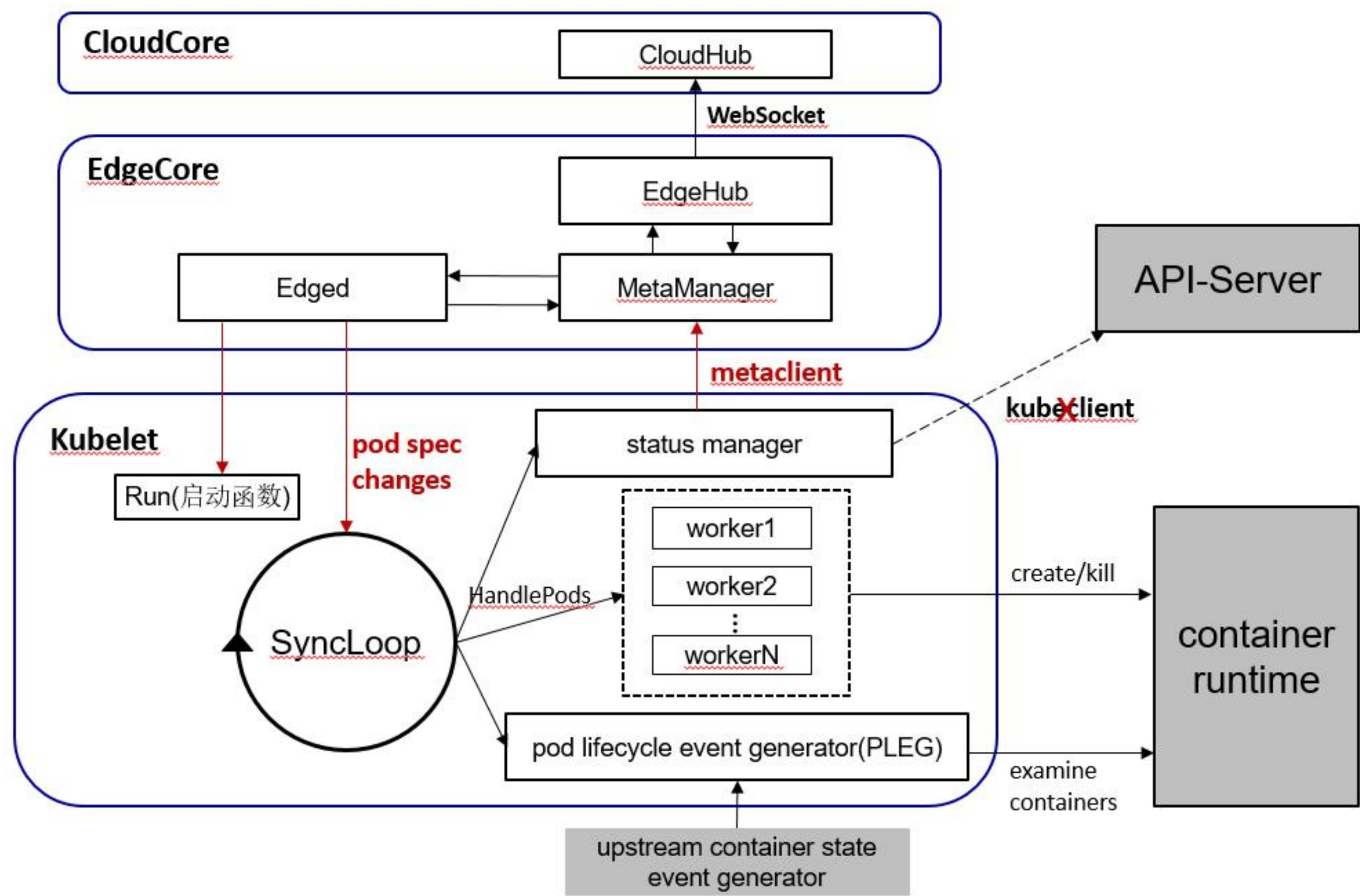
边缘应用部署与演示

KubeEdge.io

边缘应用部署流程-----云端



边缘应用部署流程----边端



边缘应用部署----边端同节点服务访问

- ① 【必选】部署Nginx应用(deploy/ds/sts)
- ② 【必选】部署Nginx应用对应的Service
- ③ 【必选】部署client应用
- ④ 【可选】配置应用的configmap/secret/env
- ⑤ 【可选】配置健康检查
- ⑥ 【可选】配置亲和节点调度策略
- ⑦ 【可选】配置反亲和节点调度策略
- ⑧ 【可选】配置应用迁移策略tolerations

边缘应用部署演示----边端跨节点服务访问

- ① 【必选】部署Nginx应用(deploy/ds/sts)
- ② 【必选】部署Nginx应用对应的Service
- ③ 【必选】部署client应用
- ④ 【可选】配置应用的configmap/secret/env
- ⑤ 【可选】配置健康检查
- ⑥ 【可选】配置亲和节点调度策略
- ⑦ 【可选】配置反亲和节点调度策略
- ⑧ 【可选】配置应用迁移策略tolerations

边缘应用部署演示----集群外访问

- ① 【必选】部署Nginx应用(deploy/ds/sts)
- ② 【可选】部署Nginx应用对应的Service
- ③ 【可选】配置应用的configmap/secret/env
- ④ 【可选】配置健康检查
- ⑤ 【可选】配置亲和节点调度策略
- ⑥ 【可选】配置反亲和节点调度策略
- ⑦ 【可选】配置应用迁移策略tolerations

边缘应用部署演示----其他使用

- ① 事件查看与使用（实验性）
- ② 应用日志持久化与查看
- ③ 应用shell/webshell/重启/删除
- ④ 边缘自治之节点离线访问（实验性）
- ⑤ cloudcore日志规范化输出（实验性）与查看
- ⑥ edgecore日志规范化输出（实验性）与查看
- ⑦ edgemesh日志规范化输出（实验性）与查看

边缘应用部署演示----附件

- ① <https://github.com/cheng-hunter/image/blob/main/test-client.yaml>
- ② <https://github.com/cheng-hunter/image/blob/main/test-configmap.yaml>
- ③ <https://github.com/cheng-hunter/image/blob/main/test-deploy.yaml>
- ④ <https://github.com/cheng-hunter/image/blob/main/test-ds.yaml>
- ⑤ <https://github.com/cheng-hunter/image/blob/main/test-secret.yaml>
- ⑥ <https://github.com/cheng-hunter/image/blob/main/test-sts.yaml>

添加社区小助手

课后答疑 | 大咖交流 | 社区礼品



Github: <https://github.com/kubeedge>



Website: <https://kubeedge.io>



Slack Channel: <https://kubeedge.slack.com>



KubeEdge公众号



课程小助手

谢谢

个人邮箱: chengpanggang@126.com

