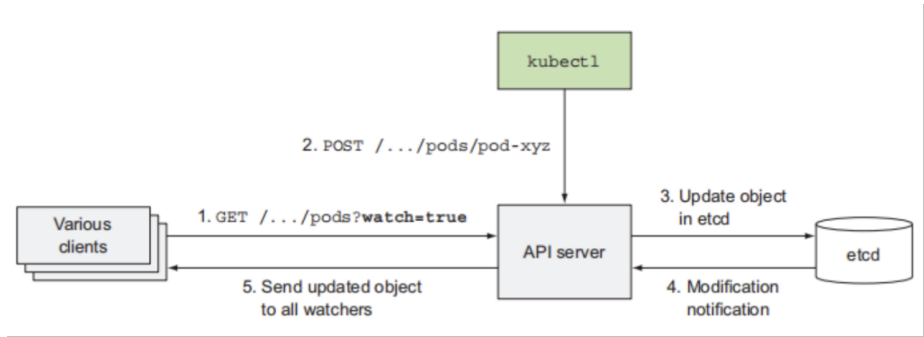


容器化基础与CICD概述

由 李睿创建, 最后修改于六月 09, 2020

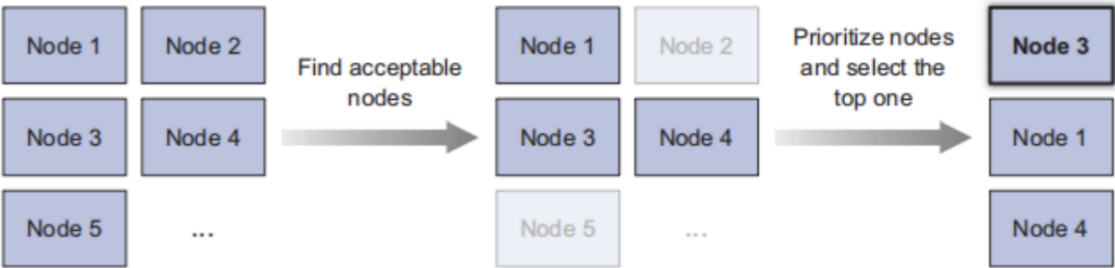
k8s是一个使用了数据库的web服务

- kubectl 是一个k8s“定制化的curl”命令行



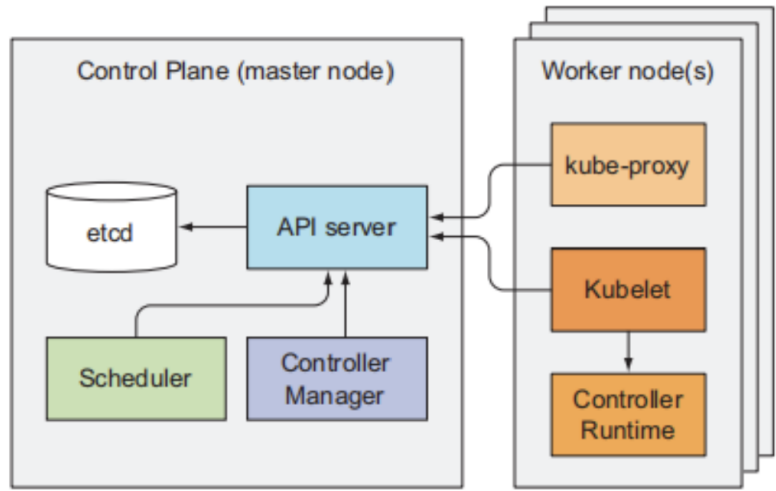
k8s是一个调度系统

- 把工作负载跑到合适的机器上

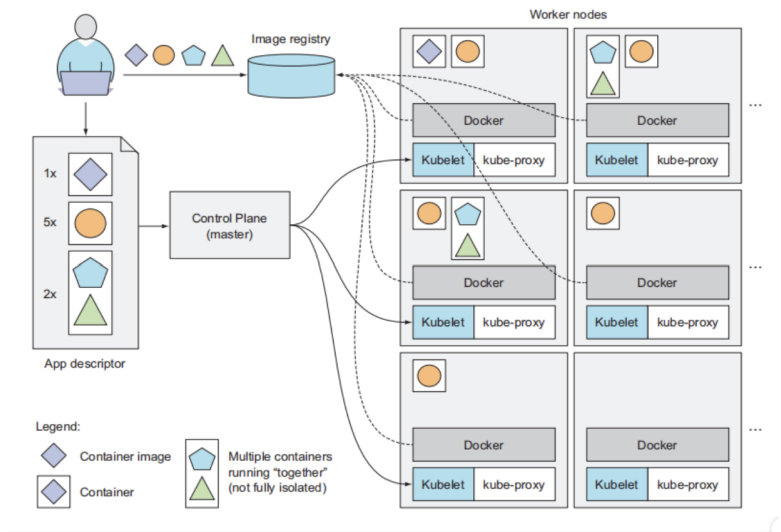


k8s的核心组件

- 完成一次调度需要哪些东西?
- 提供服务



使用k8s的工作流



k8s采用了声明式API

- 描述一个状态



命令式
(和孩子交流)



声明式
(和员工交流)

- 吃饭
- 刷牙
- 睡觉
- 唱一首歌
- 新扩一个 pod
- 删除一个 pod

- 市场占有率达到80%
- 稳定性达到99.99%
- 做一个身高体重正常的孩子
- 副本数保持在3个

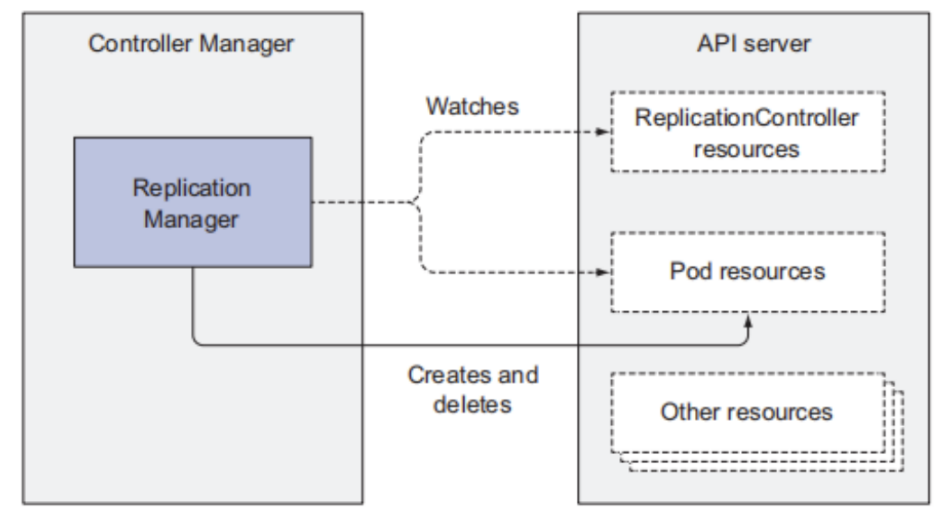
k8s能懂的“语言”是资源对象

- 声明式的
- yaml或是json格式
- 资源对象描述
- 资源对象有标签



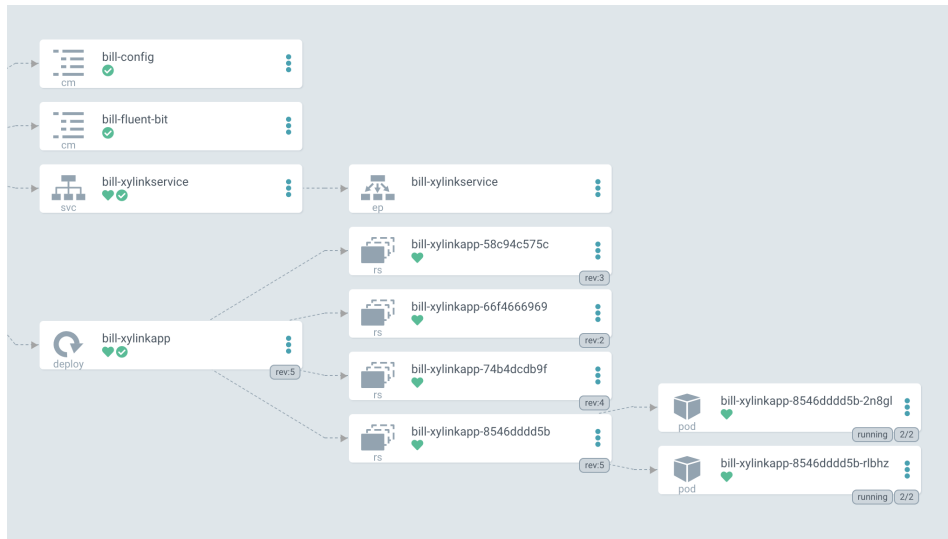
调协过程

- 实际状态
- 期望状态



常用的资源对象

为了和k8s打交道，我们需要学习一些“单词”



- pod
 - 容器组
 - 跑了几个在逻辑上有关系的一台“服务器”
 - pod IP 动态变化
 - pod所在的node动态变化
 - 基本调度单元
 - 自动创建
- initContainers
 - 前置检查容器，成功后才会运行业务容器
 - 手动创建
- ReplicaSet
 - 协调pod的副本达到期望状态
 - 自动创建
- deployment
 - 通过创建ReplicaSet来进行滚动升级
 - 手动创建
- configmap
 - 里面放了服务使用的配置文件
 - 紧紧跟随pod动态走
 - 手动创建
- service name
 - 为服务配置一个集群内的域名
 - 这样就可以无视服务的pod IP变化
 - 手动创建
- endpoints
 - 一个存放有效pod IP列表的对象
 - 自动创建

API对象资源示例

```
# deployment
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:
        - configMap:
            name: nginx-config
            name: nginx-config
      containers:
        - name: nginx
          image: nginx:1.14.2
          volumeMounts:
            - mountPath: /opt
              name: nginx-config
          ports:
            - containerPort: 80
              hostPort: 80
              name: api
              protocol: TCP

# service
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx-dev-openresty
    name: nginx
spec:
  ports:
    - name: http
      nodePort: 26003
      port: 80
      targetPort: http
  selector:
    app: nginx
  type: NodePort

# configmap
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
data:
  player_initial_lives: 3
  ui_properties_file_name: "user-interface.properties"
  game.properties: |
    enemy.types=aliens,monsters
    player.maximum-lives=5
  user-interface.properties: |
    color.good=purple
    color.bad=yellow
    allow.textmode=true

```

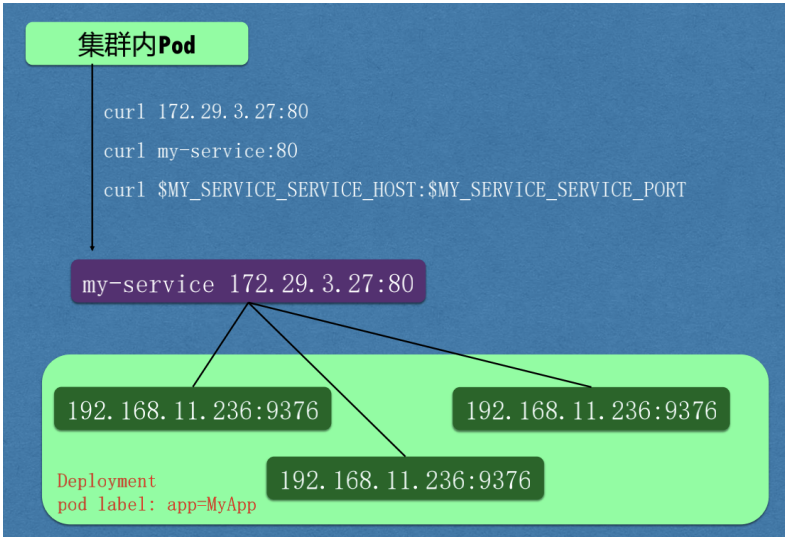
服务发现

介绍一些关于IP和port的“单词”

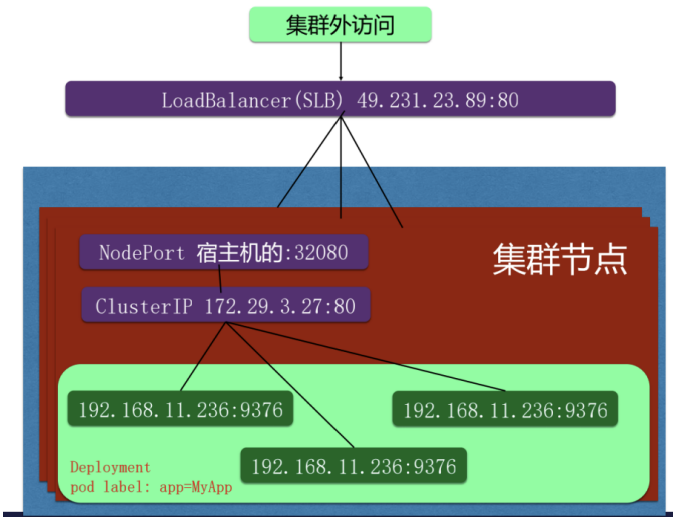
- namespace
 - 用来进行资源隔离
- pod IP / **cluster-cidr**
 - 只能在集群内使用
 - 虚拟IP
 - pod之间通讯
- service IP / **cluster IP**
 - 通过service name解析出来的IP
 - service IP : pod IP 是“一对多”的关系
- node IP
 - 服务器的IP

- 集群组件之间通讯使用
- target port
 - service 对象上提供服务的端口
- containerPort
 - pod中的容器端口
- port
 - 通过service IP 加 port 来访问服务
- nodeport
 - 通过任意的node IP 加 nodeport 来从集群外部访问集群内的服务
- hostport
 - 只能使用pod所在的node上的IP来访问服务
 - akka需要使用

集群内访问



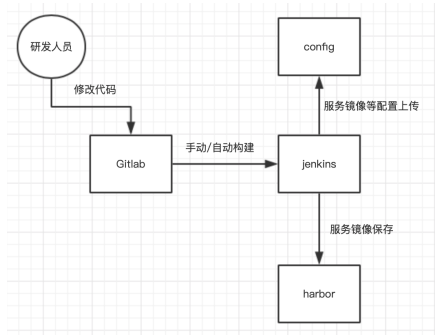
集群外访问



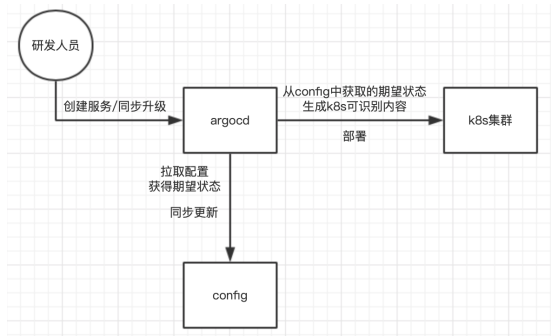
CICD概述

事务处理
CI持续交付(构建), CD持续部署(部署)

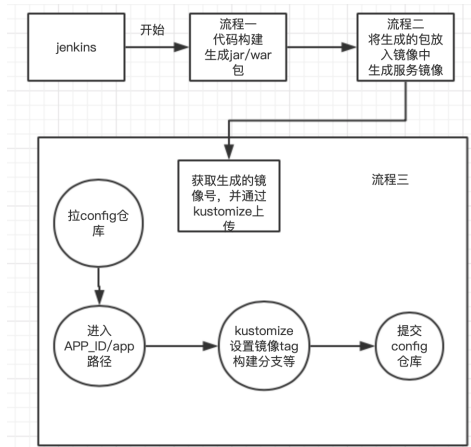
CI流程



CD流程



jenkins是如何上传镜像到config的



新服务接入

先准备好cicd文件

确定好服务使用的中间件配置是否齐全，或已存在

确定流量走向，是否需要走nginx转发，是否需要供集群外访问等

进行cicd流程部署服务

查看服务运行状态

APP DETAILS

APP DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH

Healthy ?

当前资源是否正常

Synced ?

当前服务是否是最新状态

每三分钟去获取

To tx-dev (f0c501e)

Authored by weiyuandong <weiyu...>

背景图 fix-570f895-zhangzhengya...

Sync OK ?

当前同步是否成功

To b84825a

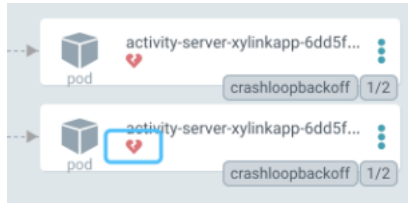
Succeeded 12 days ago (Thu May 28 2020 12:03:34 GMT+0800)

Authored by lirui <lirui@xylink.com>

Update kustomization.yaml

Progressing ?

代表pod正在启动或者启动失败



具体状态需要查看pod，红色表示不健康

点击上边的pod资源，可以查看生命周期事件，以及服务标准输出

生命周期事件可以查看pod运行不起来的原因，日志可以看服务进行到哪一步不对

SUMMARYEVENTSLOGS

INIT CONTAINERS:
CHECKSERVICE-MYS...
CONTAINERS:
FLUENT-BIT
ACCESS

at redis.clien'
at redis.clien'
at redis.clien'
Caused by: java.net.Soc
at java.net.Soc
at java.net.Soc
at java.net.Soc
at redis.clien'
g r...

SUMMARYEVENTS34184LOGS

REASONMESSAGE
BackOffBack-off restarting failed container

无标签