

# 腾讯云基于Kubernetes的应用编排实践

于广游@腾讯云

# 目录

- 为什么需要应用编排
- k8s和社区的应用编排
- 腾讯云容器服务应用编排

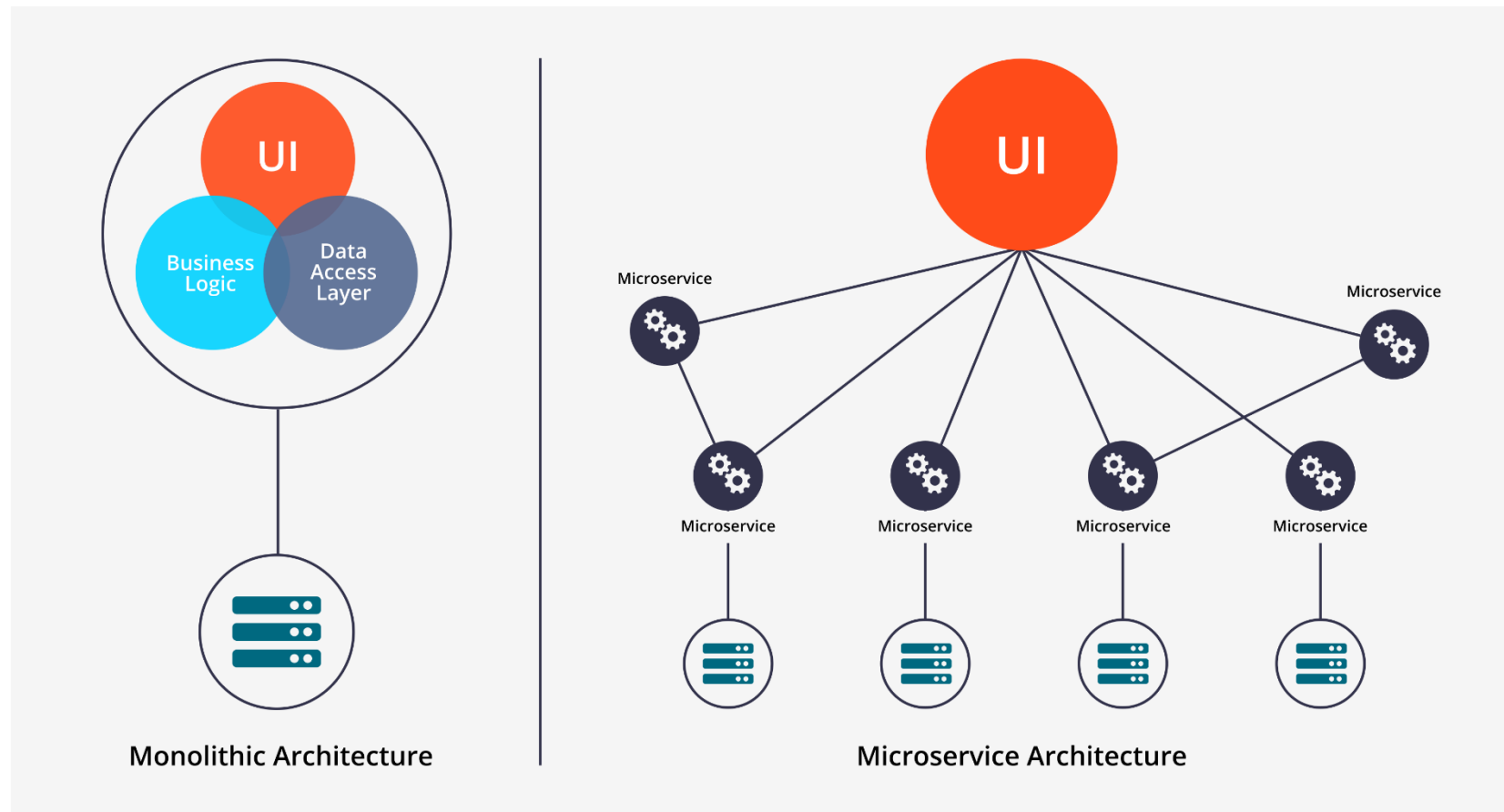
# 为什么需要应用编排

## 单体式应用的问题

- 开发效率低
- 维护困难
- 稳定性差
- 扩展性差

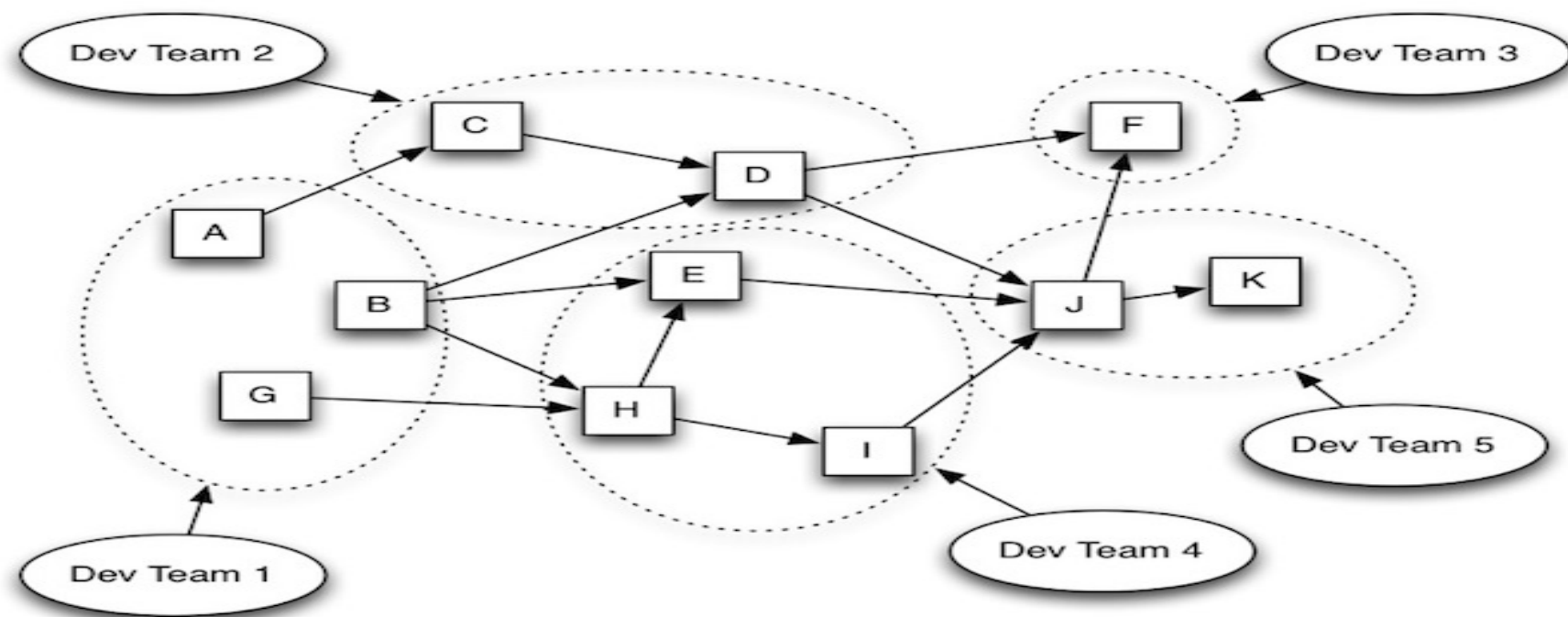
## 微服务架构带来的收益

- 开发效率高
- 维护简单
- 稳定性好
- 扩展性好



# 为什么需要应用编排

但也会带来很多问题



## 单体式

- 开发  
部署单个程序直接进行开发
- 测试  
单个组件测试
- 交付  
单个组件交付给运维
- 部署  
部署单个组件即可部署成功
- 回滚  
回滚单个组件
- 监控  
监控单个组件的状态

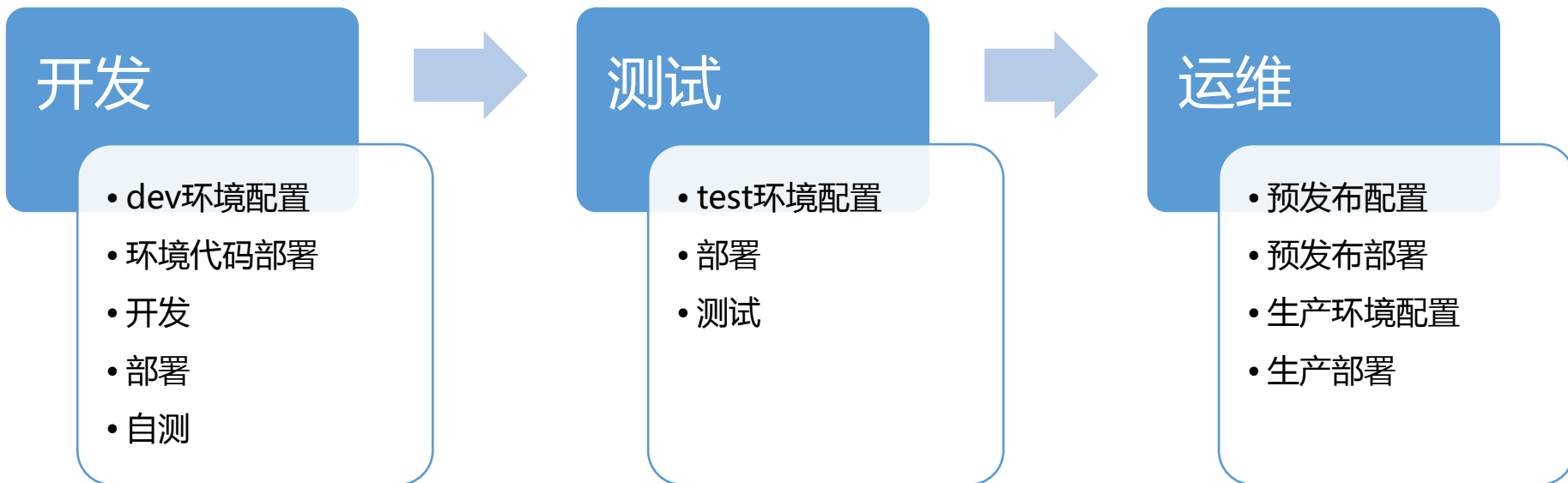
## 微服务

- 开发  
同时涉及多个有依赖关系的组件开发
- 测试  
提交多个组件，这些组件之间有严格的版本依赖
- 交付  
清单中包含多个组件，每个组件的包和配置的版本必须固定
- 部署  
需要一次发布多个组件
- 回滚  
回滚多个组件，谨防因此故障
- 监控  
一个组件的异常可能是其他组件引起的

# 为什么需要应用编排

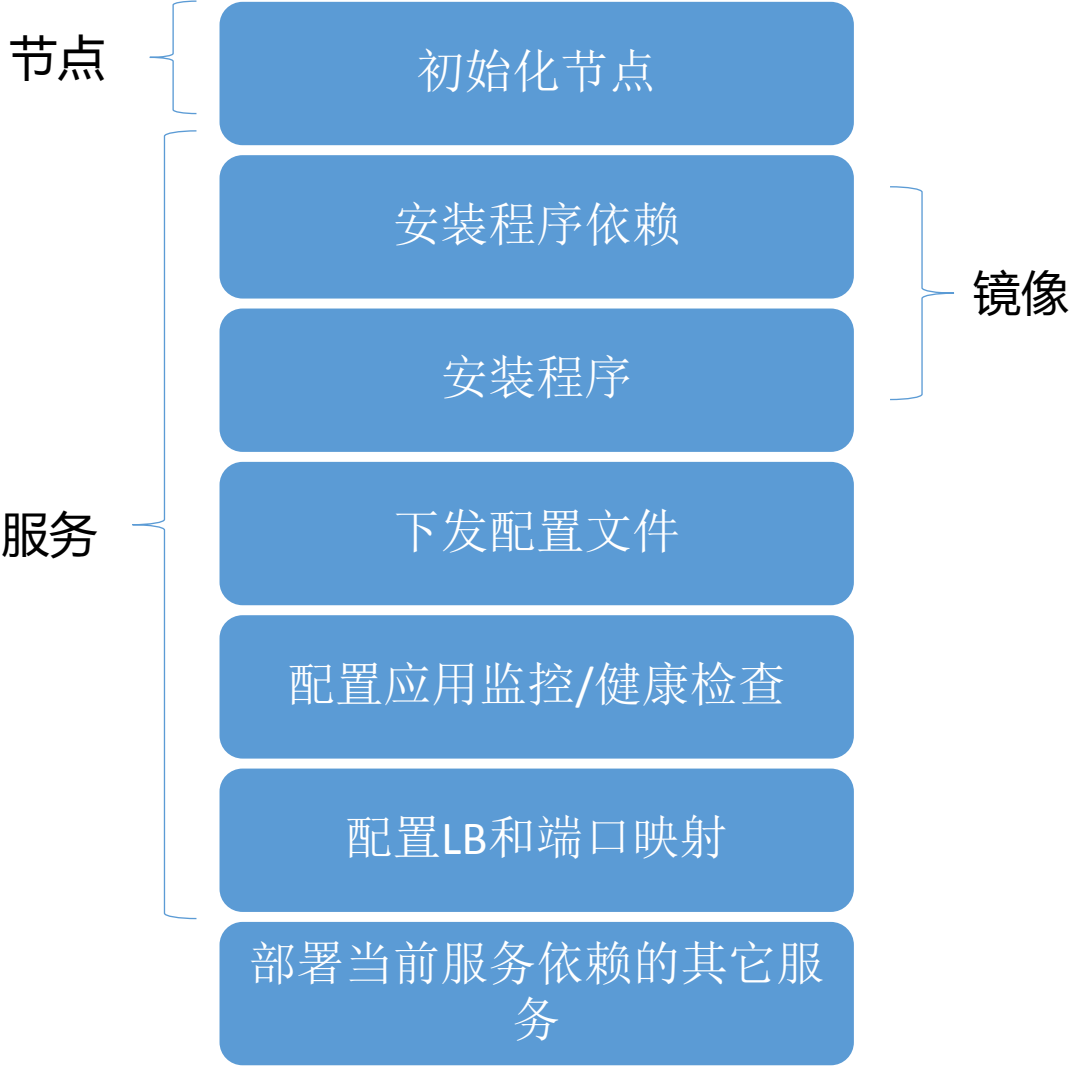
## 微服务架构在交付部署上带来的问题

- 开发、测试环境维护困难，影响效率
- 交付、发布困难，故障率高
  - 漏提，提错配置、镜像
  - 回滚出错造成二次故障
- 监控，问题发现不直观



怎么解决这些问题？

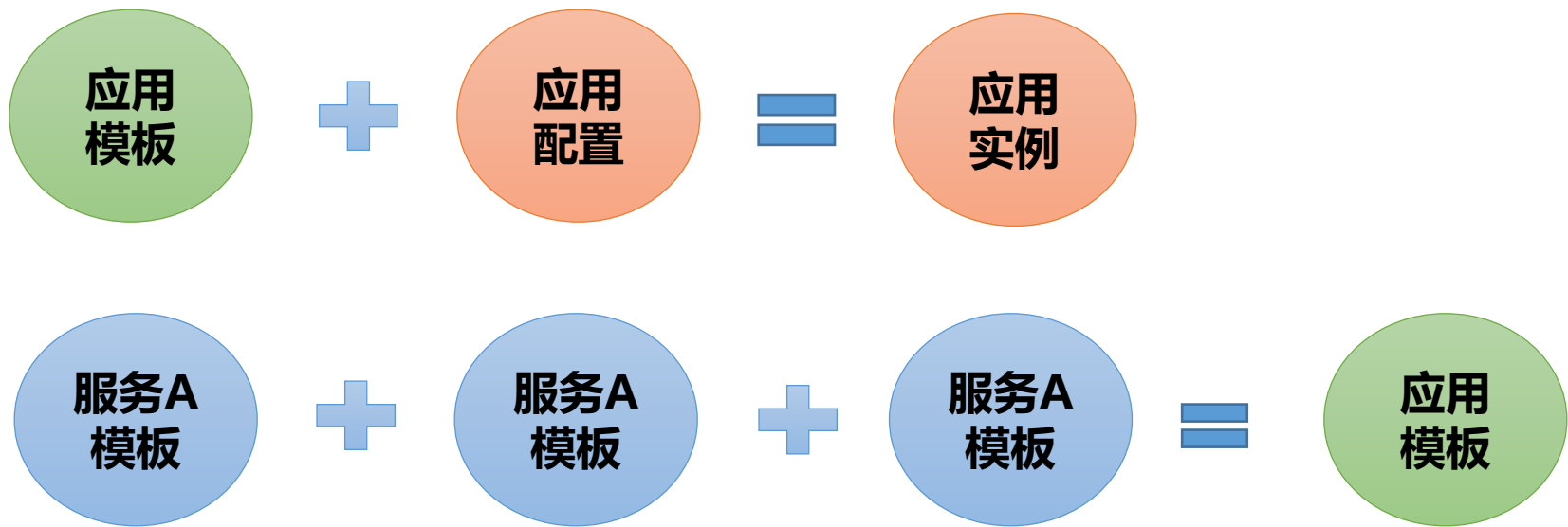
# 怎样手工部署应用





# 有问题，加抽象





快速，标准，自动化

# K8S和社区的应用编排

## Helm 和 Charts

```
wordpress/  
  Chart.yaml          # A YAML file containing information about the chart  
  LICENSE             # OPTIONAL: A plain text file containing the license for the chart  
  README.md          # OPTIONAL: A human-readable README file  
  values.yaml         # The default configuration values for this chart  
  charts/            # OPTIONAL: A directory containing any charts upon which this chart depends.  
  templates/         # OPTIONAL: A directory of templates that, when combined with values,  
                    # will generate valid Kubernetes manifest files.  
  templates/NOTES.txt # OPTIONAL: A plain text file containing short usage notes
```

# K8S和社区的应用编排

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "fullname" . }}
    chart: "{{ .Chart.Name }}-{{ .Chart.Version }}"
    release: "{{ .Release.Name }}"
    heritage: "{{ .Release.Service }}"
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: {{ template "fullname" . }}
    spec:
      containers:
        - name: {{ template "fullname" . }}
          image: "{{ .Values.image }}"
          imagePullPolicy: {{ default "" .Values.imagePullPolicy }}
          env:
            - name: ALLOW_EMPTY_PASSWORD
              {{- if .Values.allowEmptyPassword }}
              value: "yes"
              {{- else }}
              value: "no"
              {{- end }}
```

分支语句

内置变量

变量替换

```
{{- if .Values.persistence.enabled -}}
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "fullname" . }}
    chart: "{{ .Chart.Name }}-{{ .Chart.Version }}"
    release: "{{ .Release.Name }}"
    heritage: "{{ .Release.Service }}"
  annotations:
    {{- if .Values.persistence.storageClass }}
    volume.beta.kubernetes.io/storage-class: {{ .Values.persistence.storageClass }}
    {{- else }}
    volume.alpha.kubernetes.io/storage-class: default
    {{- end }}
spec:
  accessModes:
    - {{ .Values.persistence.accessMode | quote }}
  resources:
    requests:
      storage: {{ .Values.persistence.size | quote }}
{{- end -}}
```

管道

函数

# K8S和社区的应用编排

## k8s的问题

- 原生不提供对应用的支持

## Helm的问题

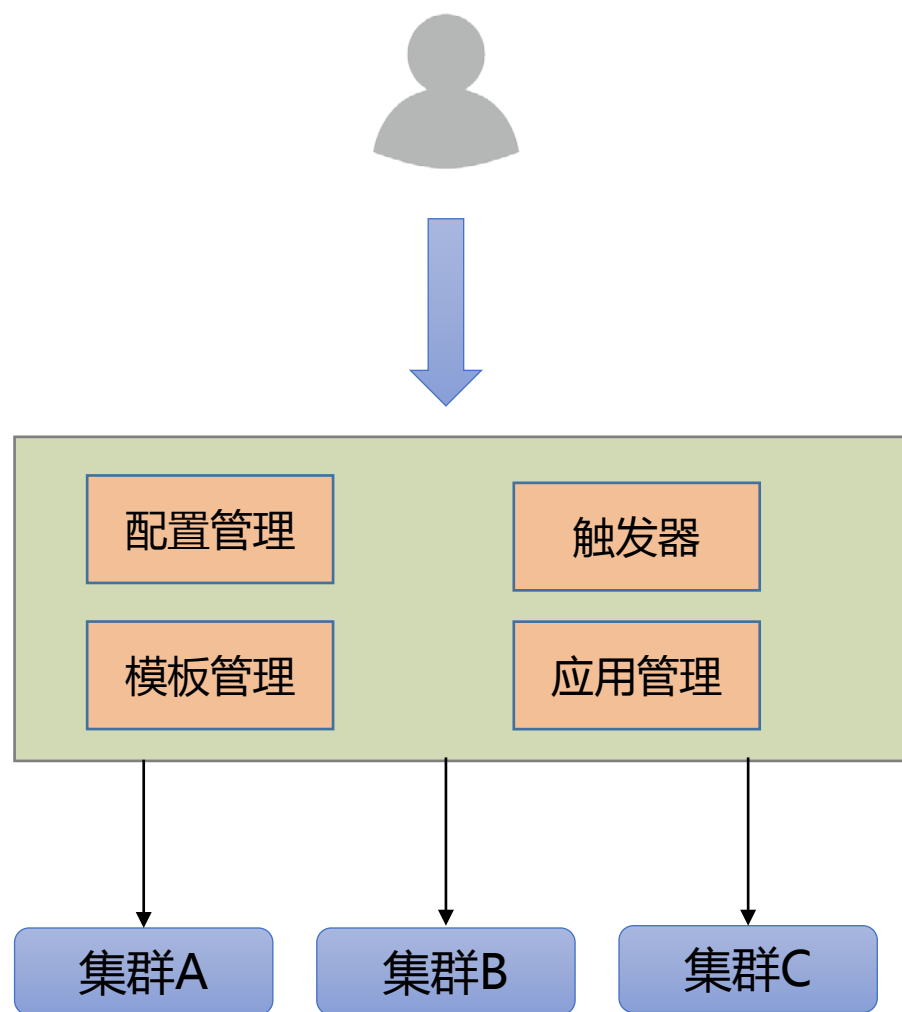
- 更侧重于包管理
- 语法复杂
- 不支持多集群
- 不支持应用下按服务更新
- 应用状态查看不友好

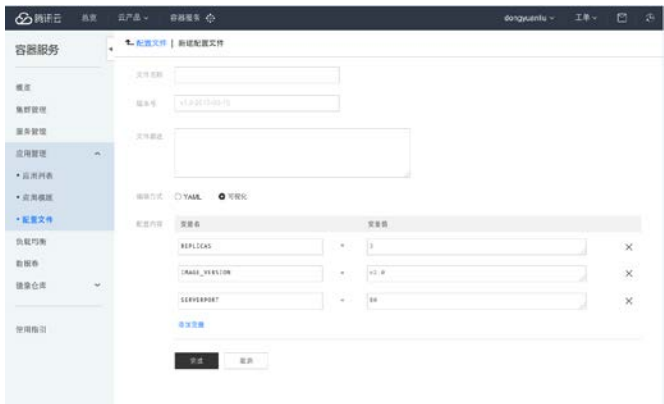


**kubernetes**

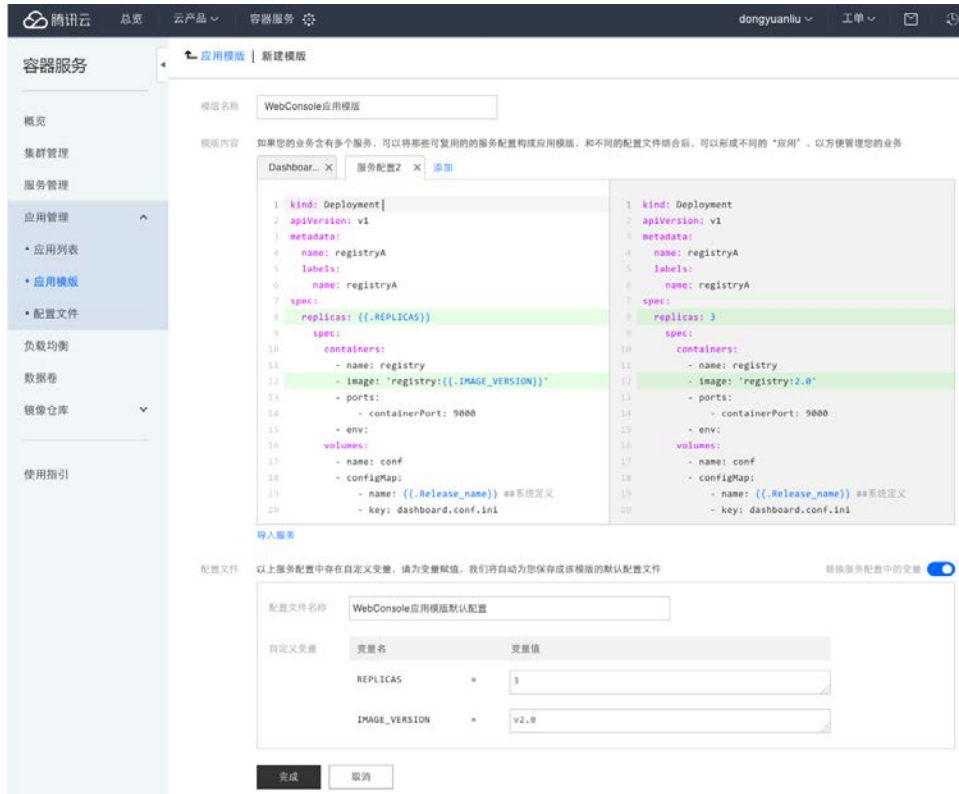


- 模板管理
  - helm语法子集，变量替换
  - 可视化页面创建应用模板
- 配置管理
  - 替换模板中变量
  - 版本管理
  - 可视化diff
- 应用管理
  - 查看应用下服务的运行、负载状态
  - 应用定义的更新与发布解耦
  - 变更记录与回滚
- 触发器
  - 提交代码自动构建镜像
  - 镜像构建成功push到registry
  - 镜像更新事件触发直接部署到服务或更新应用配置

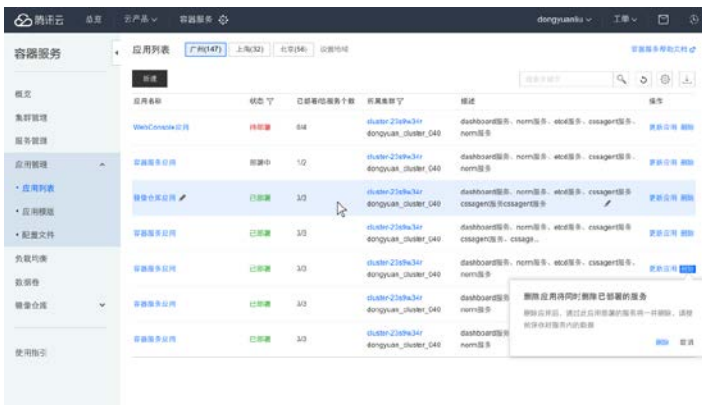




## 应用配置



## 应用模板



## 应用列表

通过触发器，可以在每次生成新的Tag（镜像版本）时，自动执行动作，如：更新使用该镜像仓库的服务

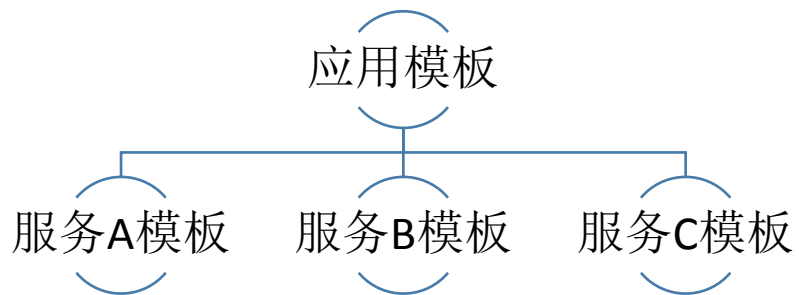


## 触发器

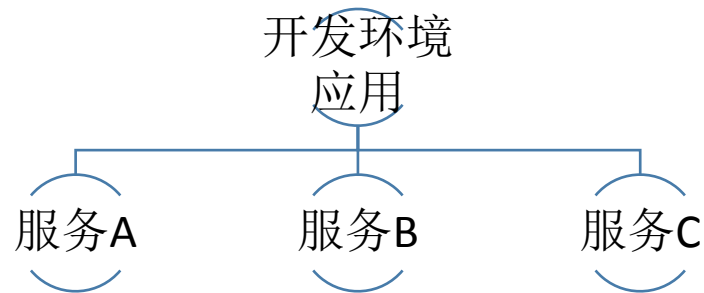
添加触发器

# 应用场景一

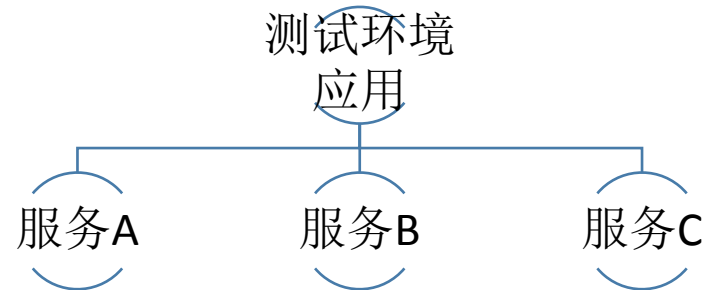
## 应用模板部署多环境



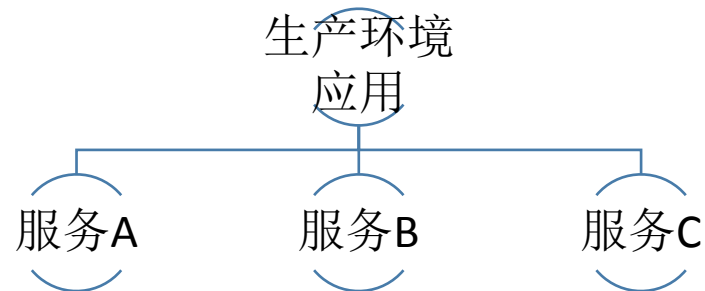
开发环境配置



测试环境配置



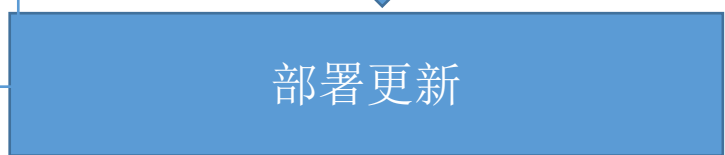
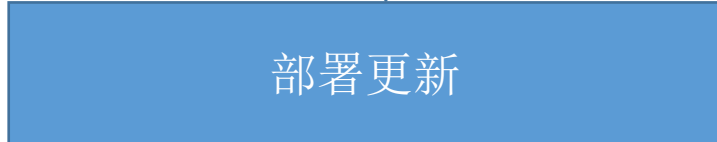
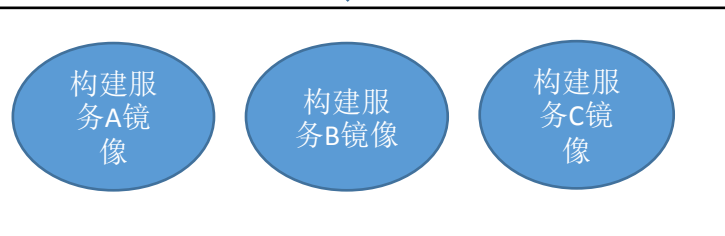
生产环境配置





# 应用场景二

## 镜像变更在多环境间传递



# 应用场景三

## 定义更新与发布分离

### 基本信息

应用名称	WebConsole应用 
运行集群	dongyuan_cluster_040 ( <a href="#">cluster-23s9w34r</a> )
应用描述	dashboard服务、norm服务、etcd服务、cssagent服务, cssagent服务 cssagent服务、cssagent服务、cssagent服务 
配置文件	<a href="#">WebConsole应用配置默认配置/v1.2-20170515</a>
创建时间	2017-05-15 10:43:22
修改时间	2017-05-15 10:43:22

### 服务列表

服务名	状态	操作
<a href="#">dashboard服务</a>	已部署	<a href="#">部署</a> <a href="#">取消部署</a> <a href="#">查看YAML</a>
<a href="#">norm服务</a>	未部署	<a href="#">部署</a> <a href="#">取消部署</a> <a href="#">查看YAML</a>
<a href="#">etcd服</a>	部署中	<a href="#">部署</a> <a href="#">取消部署</a> <a href="#">查看YAML</a>
<a href="#">cssagent服务</a>	已部署	<a href="#">部署</a> <a href="#">取消部署</a> <a href="#">查看YAML</a>

# 其他功能

- 应用发布记录与回滚
- 常见应用的快速部署
- 未来
  - 蓝绿发布
  - 应用下日志聚合
  - 调用关系展示

# 总结

- 为什么需要应用编排
  - 微服务使应用下服务数量增大，维护环境，发布更新非常困难
  - 抽象，标准化
- K8s与社区对应用编排有怎样的支持
  - k8s原生支持服务的编排
  - Helm支持快速创建应用
  - 在应用的运维管理方面较为缺失
- 腾讯云容器服务功能
  - 侧重于交付与运维管理
  - 应用模板和配置实现快速部署服务
  - 触发器将变更在不同环境间传递
  - 查看、管理应用下服务

**THANKS!**