Kubernetes 容器编排与应用编排

郭维 才云科技

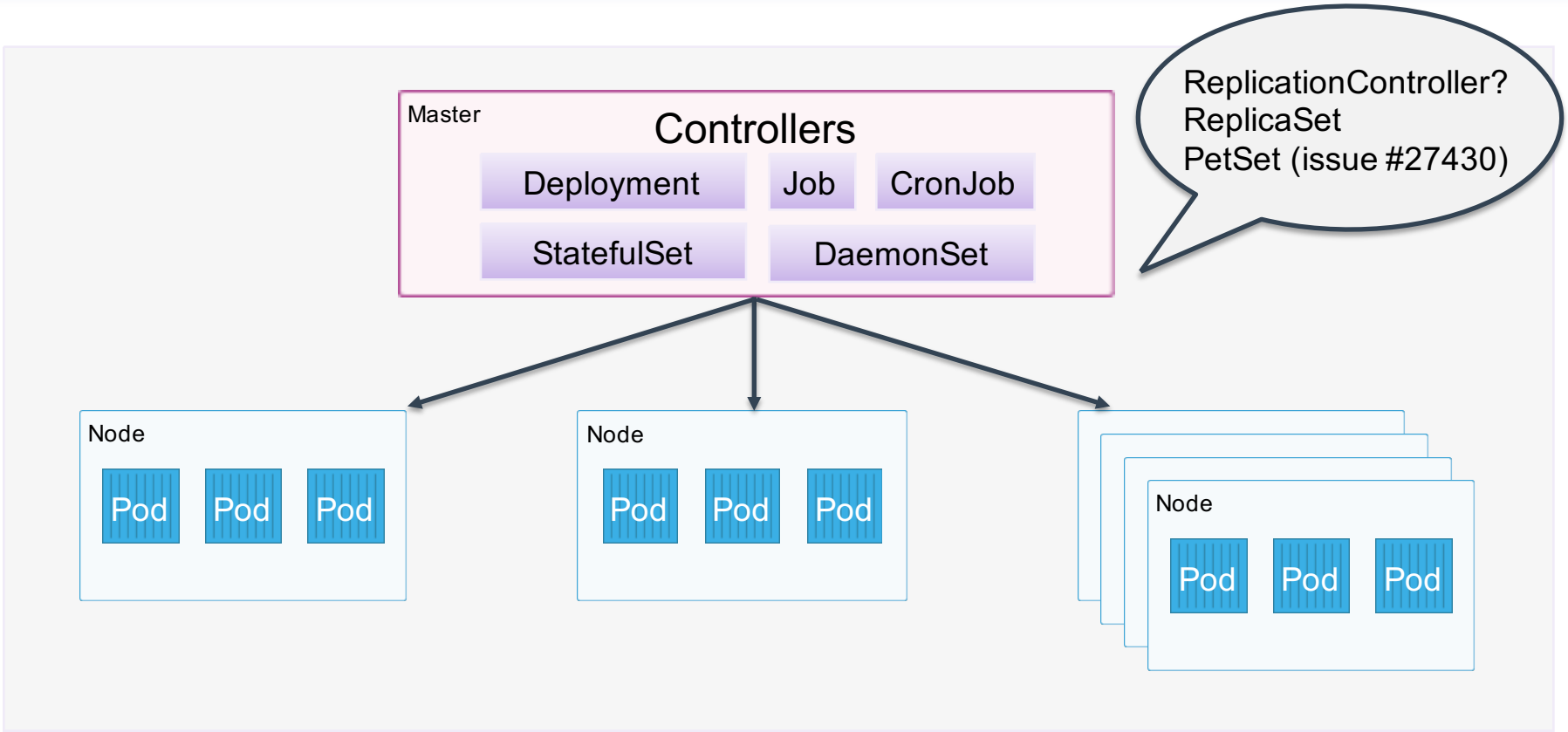
目录

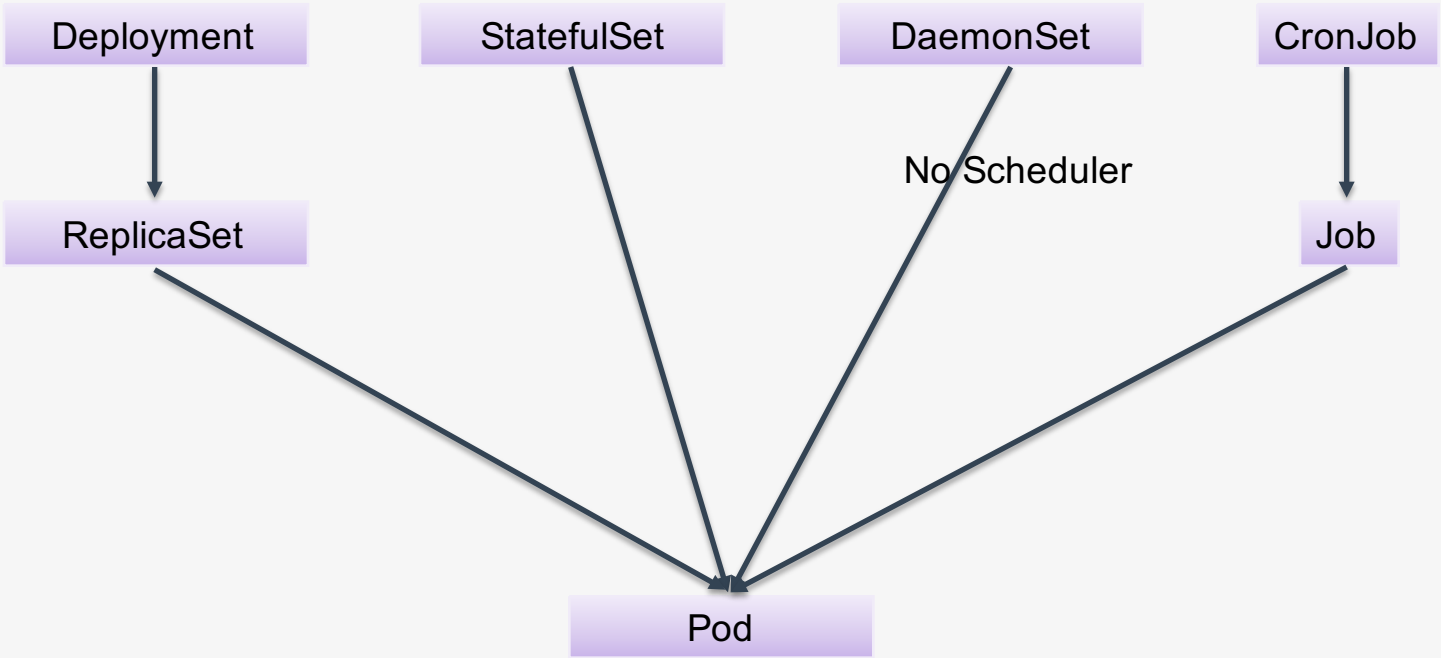
Speech content

- Kubernetes 容器编排技术
- 容器编排与应用架构
- 容器编排的困境
- 应用编排架构

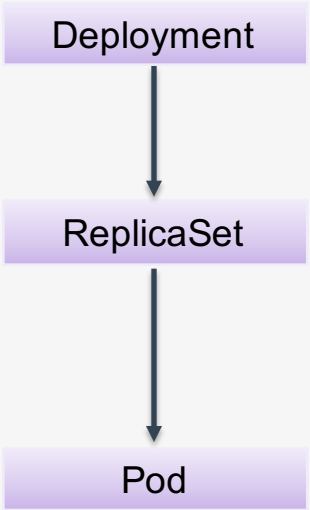
Caicloud

Kubernetes


Master
Controllers
Deployment
Job
CronJob
StatefulSet
DaemonSet
ReplicationController?
ReplicaSet
PetSet (issue #27430)
Node
Pod
Pod
Pod
Node
Pod
Pod
Pod
Node
Pod
Pod
Pod

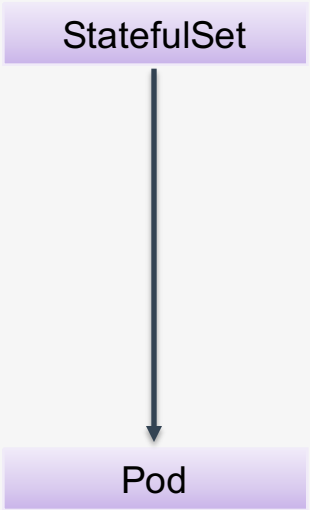

Deployment
StatefulSet
DaemonSet
CronJob
No Scheduler
ReplicaSet
Job
Pod

1. 支持指定副本数
2. 支持 Recreate 和 Rolling Update
3. 可控的滚动更新模式
4. 支持回滚
5. 可通过 HPA 实现自动弹性伸缩



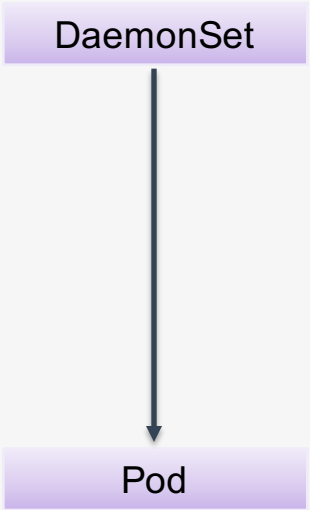

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: deployment-example
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 1
  template:
    ...
```

1. 支持指定副本数
2. Pod 具有固定且唯一的标识符
    * statefulset-example-0.example
3. Pod 可具有独立的存储
4. 一切都是有序的
    * 从 0 开始逐个创建（需等待 Ready）
    * 反向逐个删除（需等待 Terminated）
5. Update Strategy (*v1.7)
    * 反向逐个删除并重建 Pod



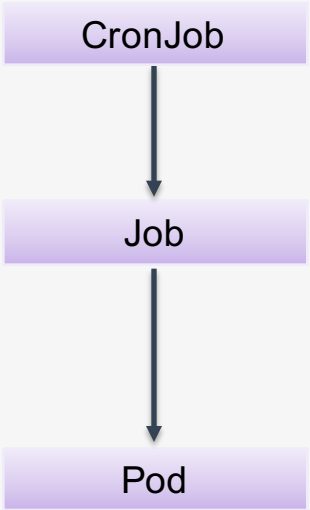

```yaml
apiVersion: v1
kind: Service
metadata:
  name: example
spec:
  clusterIP: None
  ...
---
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: statefulset-example
spec:
  replicas: 3
  serviceName: example
  volumeClaimTemplates:
    ...
  template:
    ...
```

1. 可以在指定节点一个副本
    * NodeSelector
    * Affinity
2. 可控的滚动更新模式
3. 独立的调度模式
    * 不依赖 Scheduler
    * 无视 unschedulable

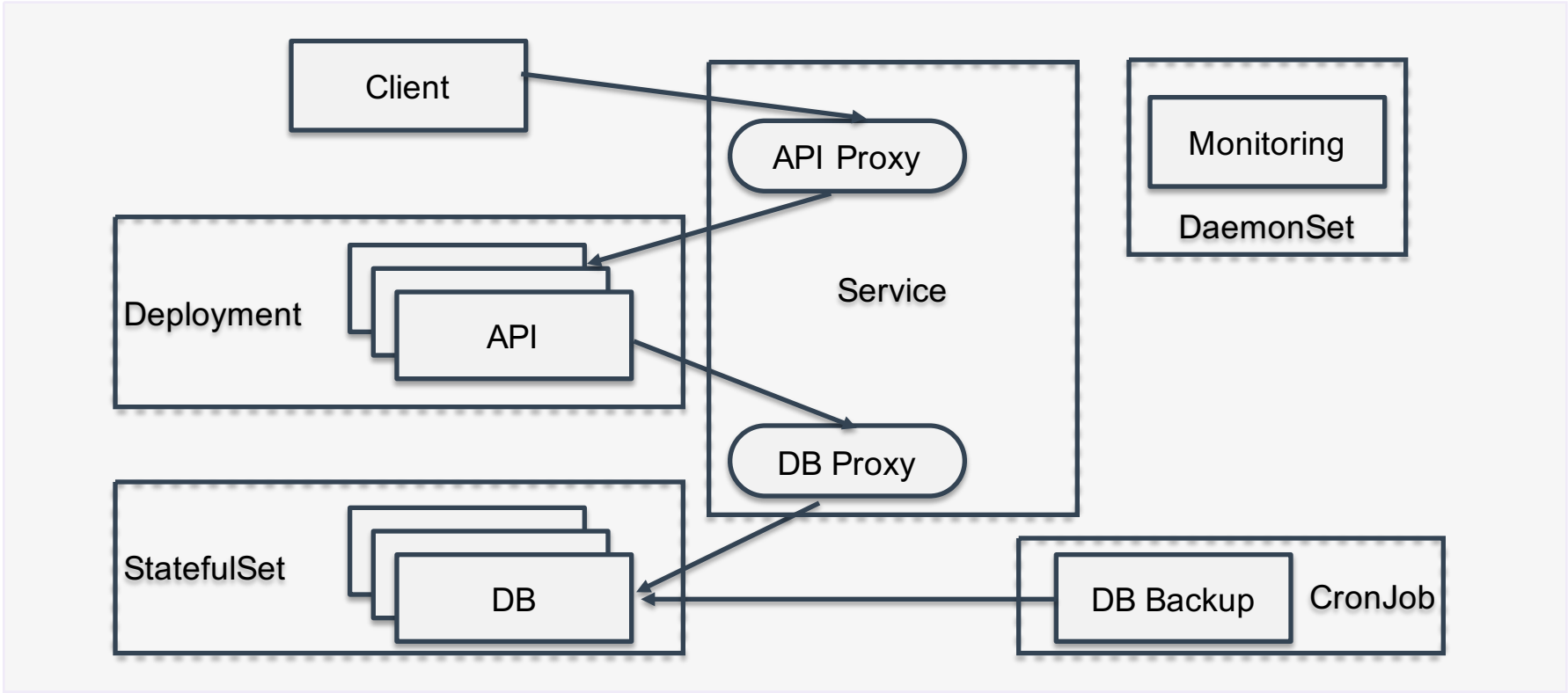


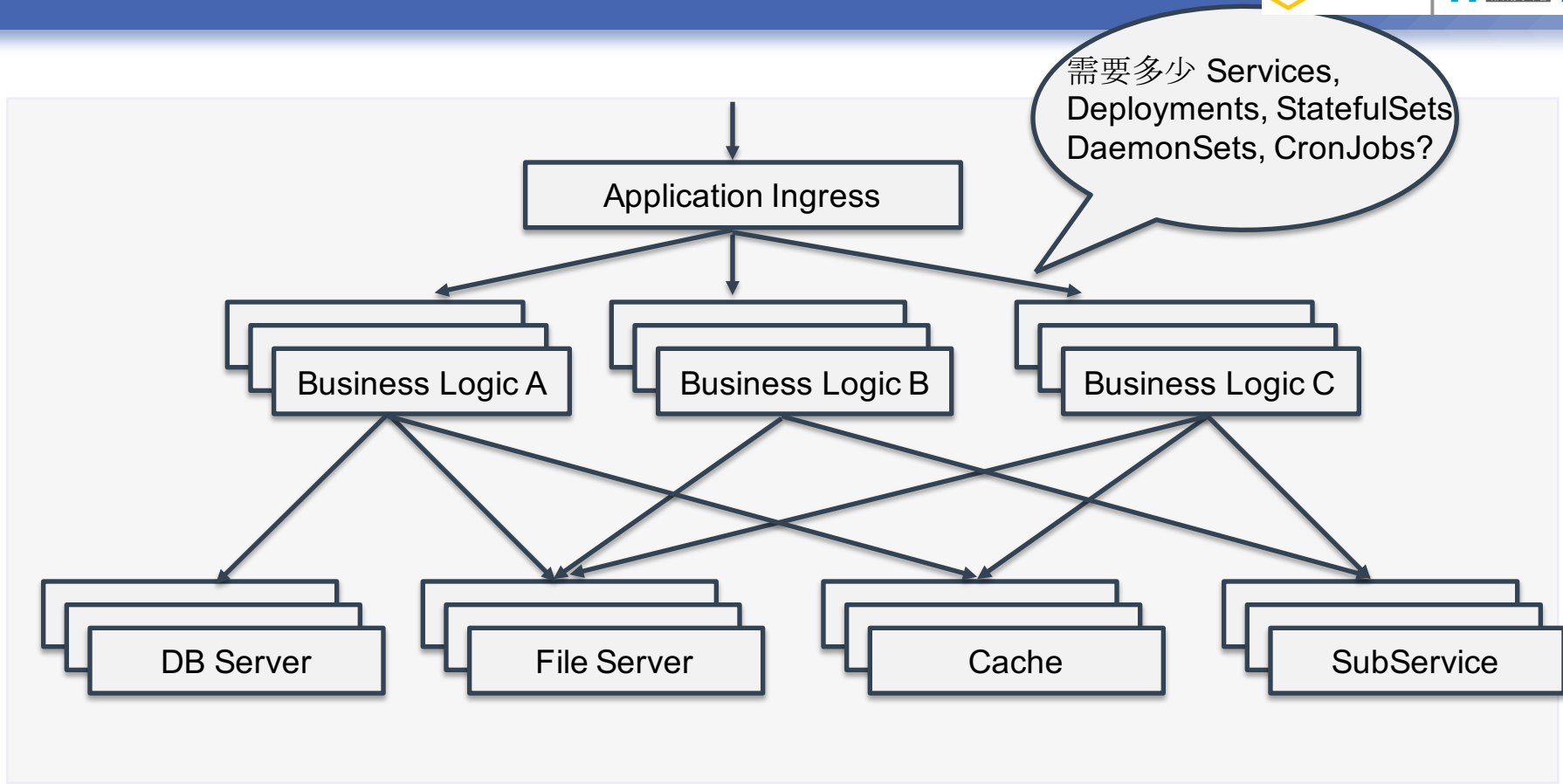

```yaml
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: daemonset-example
spec:
  updateStrategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
  template:
    ...
```

1. 定时执行的批处理任务
2. 定时任务并发策略
    * Allow
    * Forbid
    * Replace
3. 支持单任务并发控制




```yaml
apiVersion: batch/v2alpha1
kind: CronJob
metadata:
  name: cronjob-example
spec:
  schedule: "*/1 * * * *"
  concurrencyPolicy: Allow
  jobTemplate:
    spec:
      parallelism: 1
      completions: 1
      template:
      ...
```


Client
API Proxy
Monitoring
DaemonSet
Deployment
API
Service
DB Proxy
StatefulSet
DB
DB Backup
CronJob


需要多少 Services,
Deployments, StatefulSets
DaemonSets, CronJobs?
Application Ingress
Business Logic A
Business Logic B
Business Logic C
DB Server
File Server
Cache
SubService

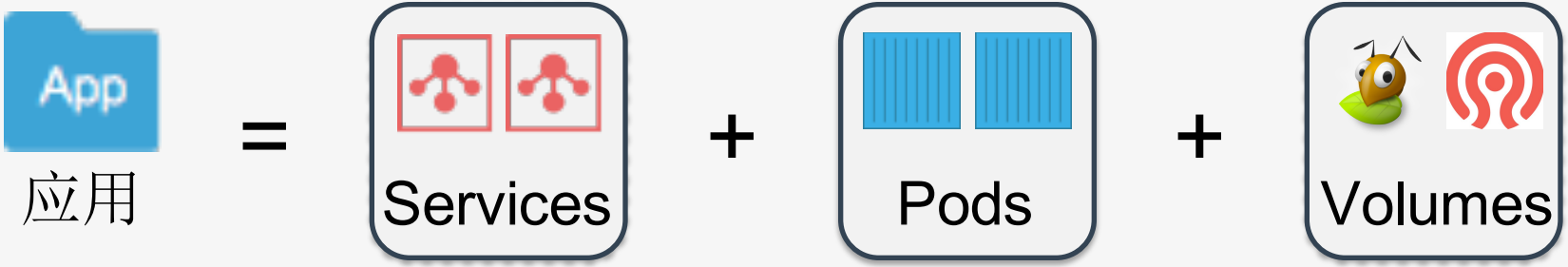
Kubernetes 为我们带来了什么？

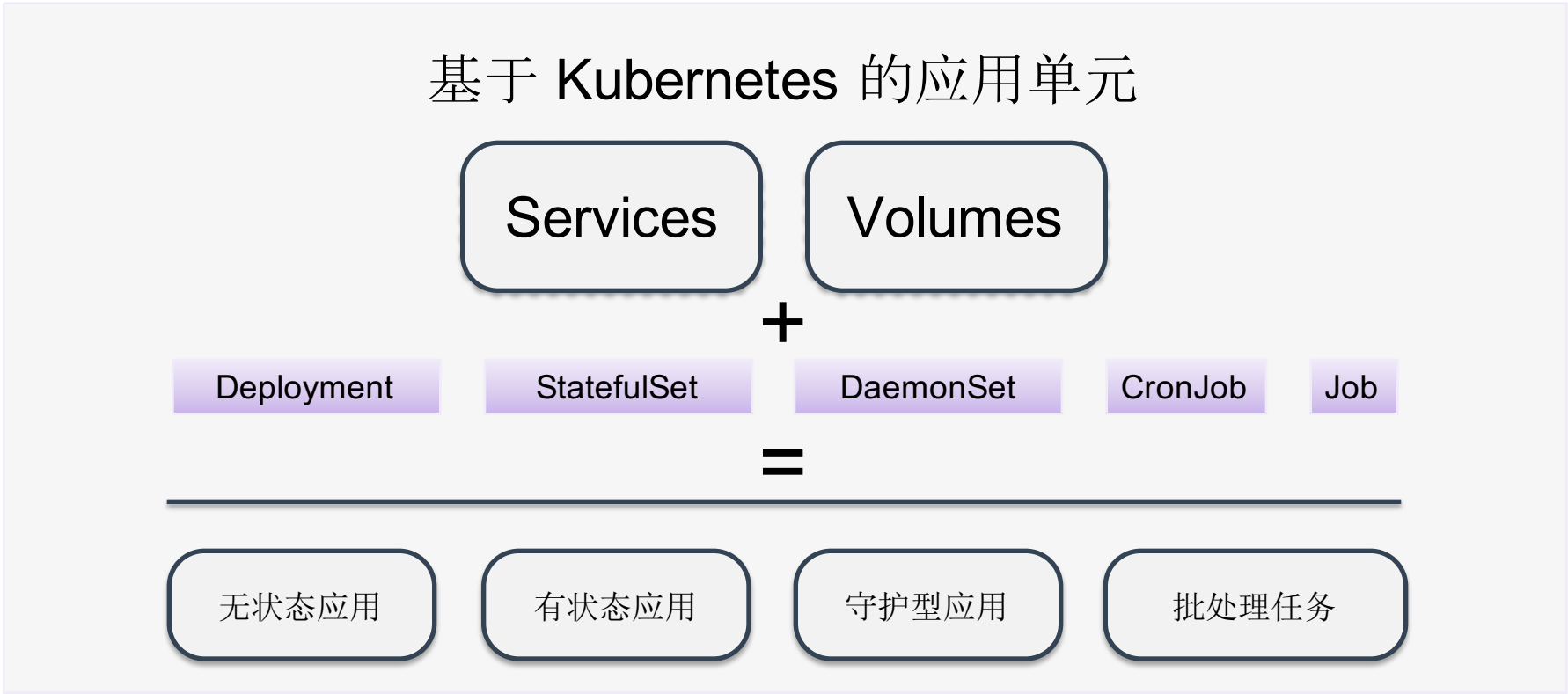
1. 提供了多种多样的基础资源封装：Service，Controllers，ConfigMap，Batch …
2. 自动化的运维机制：HPA，“VPA”，Rollback，Rolling Update
3. 极简的部署方式：YAML + Image

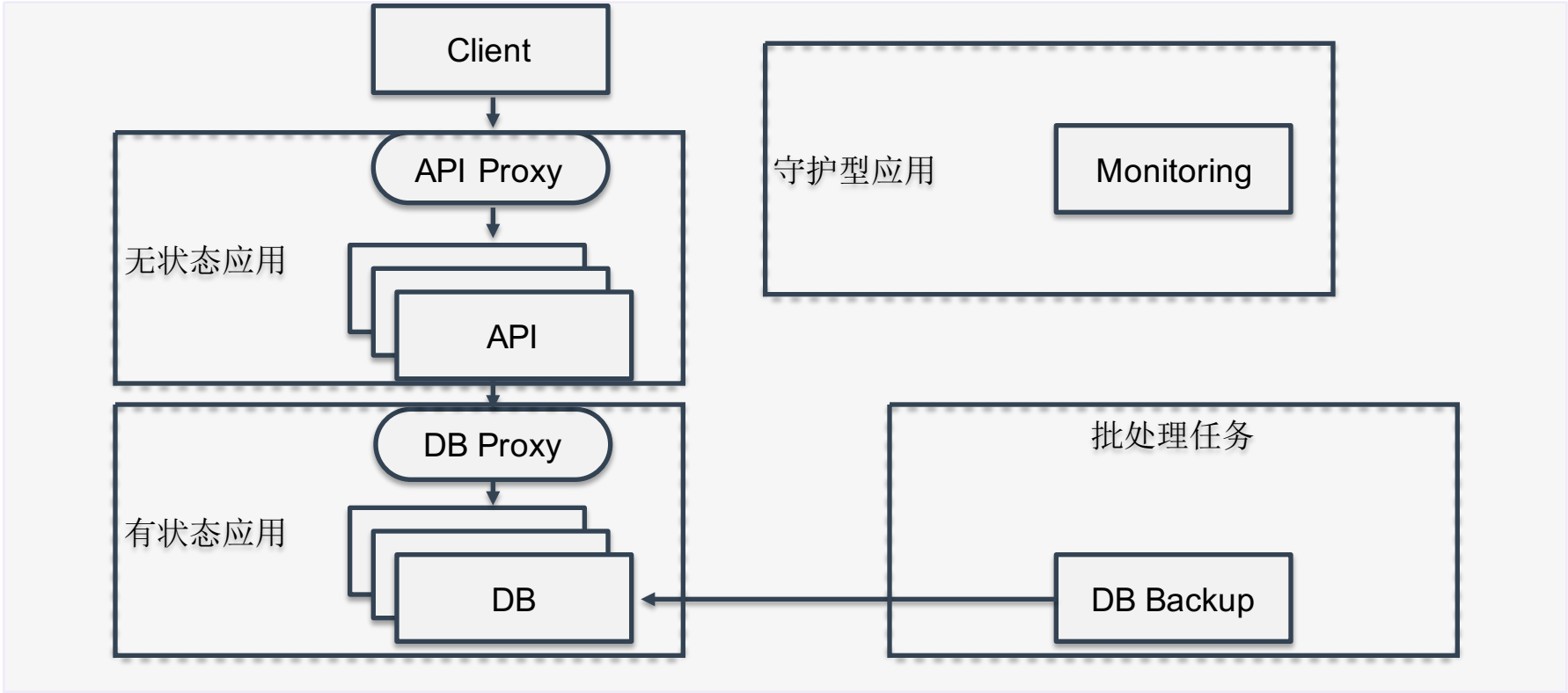
Kubernetes 的现状

1. 对于大型应用而言，需要组合使用大量的 Kubernetes 资源
2. 缺乏统一的依赖管理机制，需要用户自行解决依赖关系
3. 极简的部署方式也给用户带来了新的负担

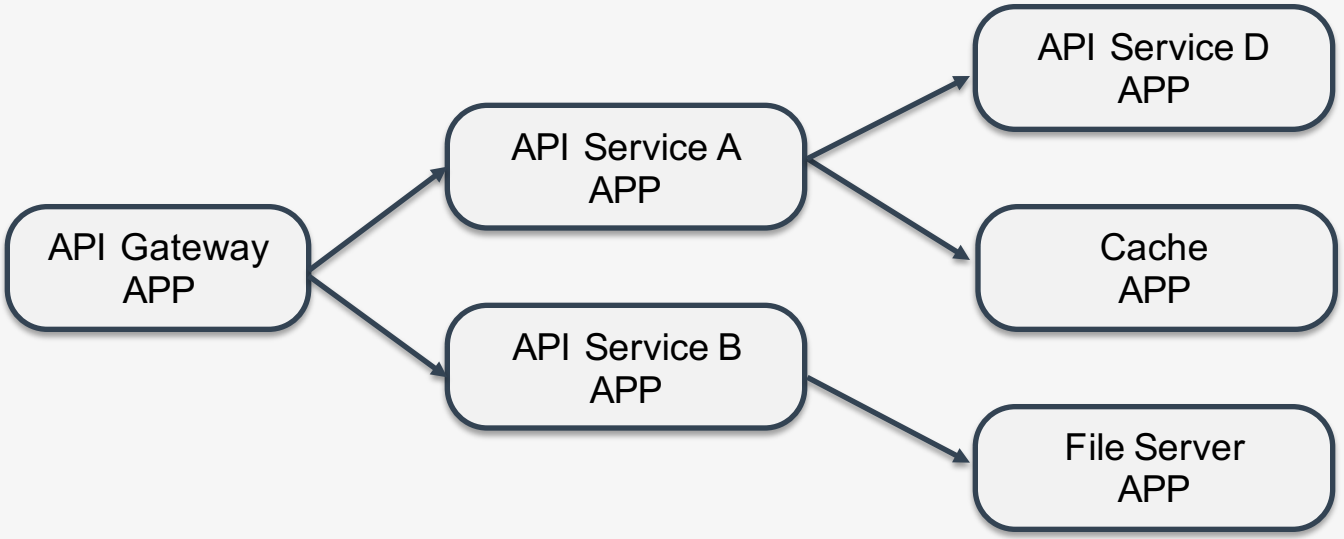
应用是什么？

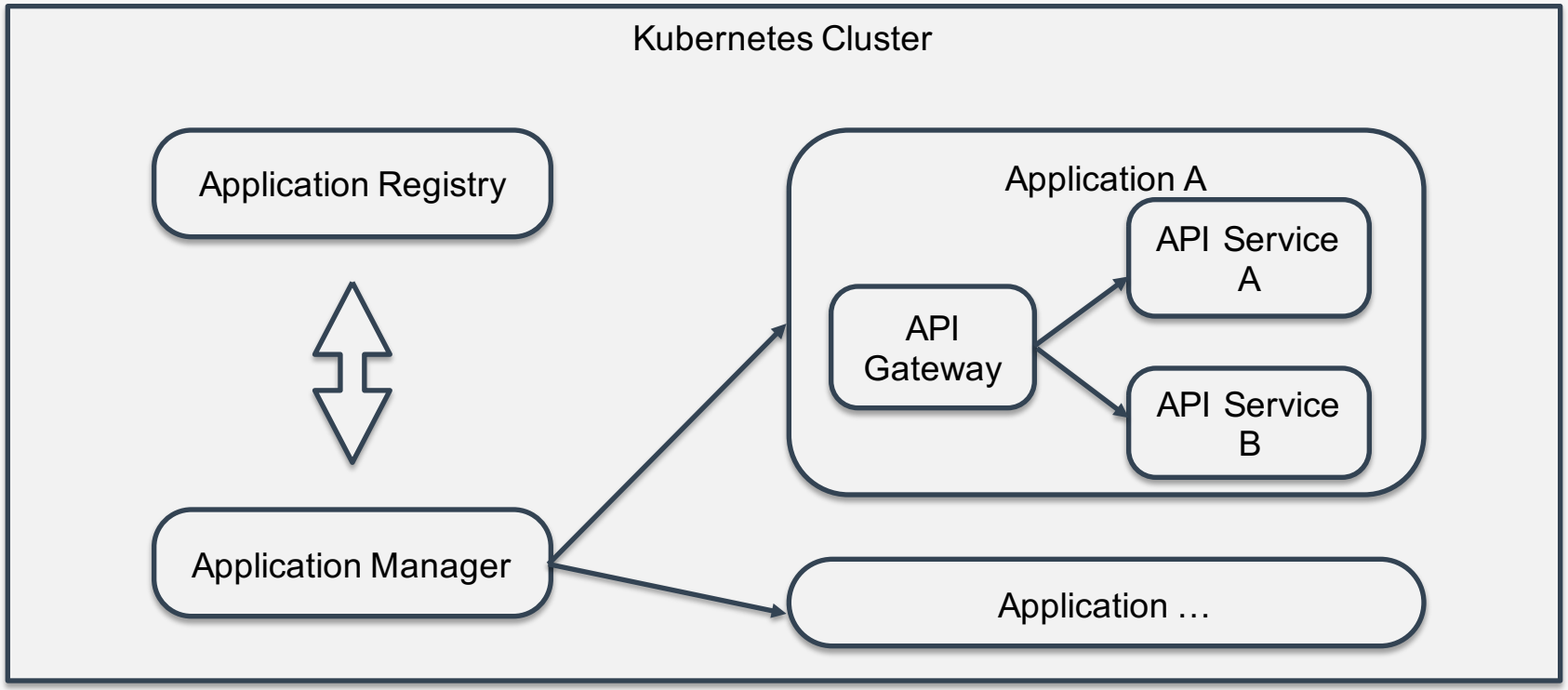



基于 Kubernetes 的应用单元
Services
Volumes
+
Deployment
StatefulSet
DaemonSet
CronJob
Job
=
无状态应用
有状态应用
守护型应用
批处理任务


Client
API Proxy
无状态应用
API
DB Proxy
有状态应用
DB
守护型应用
Monitoring
批处理任务
DB Backup

应用编排架构


API Gateway
APP
API Service A
APP
API Service B
APP
API Service D
APP
Cache
APP
File Server
APP


Kubernetes Cluster
Application Registry
Application Manager
Application A
API Gateway
API Service A
API Service B
Application …

Helm Chart

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: {{ $service.name }}
5  spec:
6    type: {{ $service.type }}
7    ports:
8    {{ range $i, $v := $service.ports -}}
9    - name: redis{{ $i }}
10     protocol: {{ $v.protocol }}
11     port: {{ $v.port }}
12     targetPort: {{ $v.targetPort }}
13   {{- end }}
```

Helm Registry

1. 负责存储和管理用户的 Charts

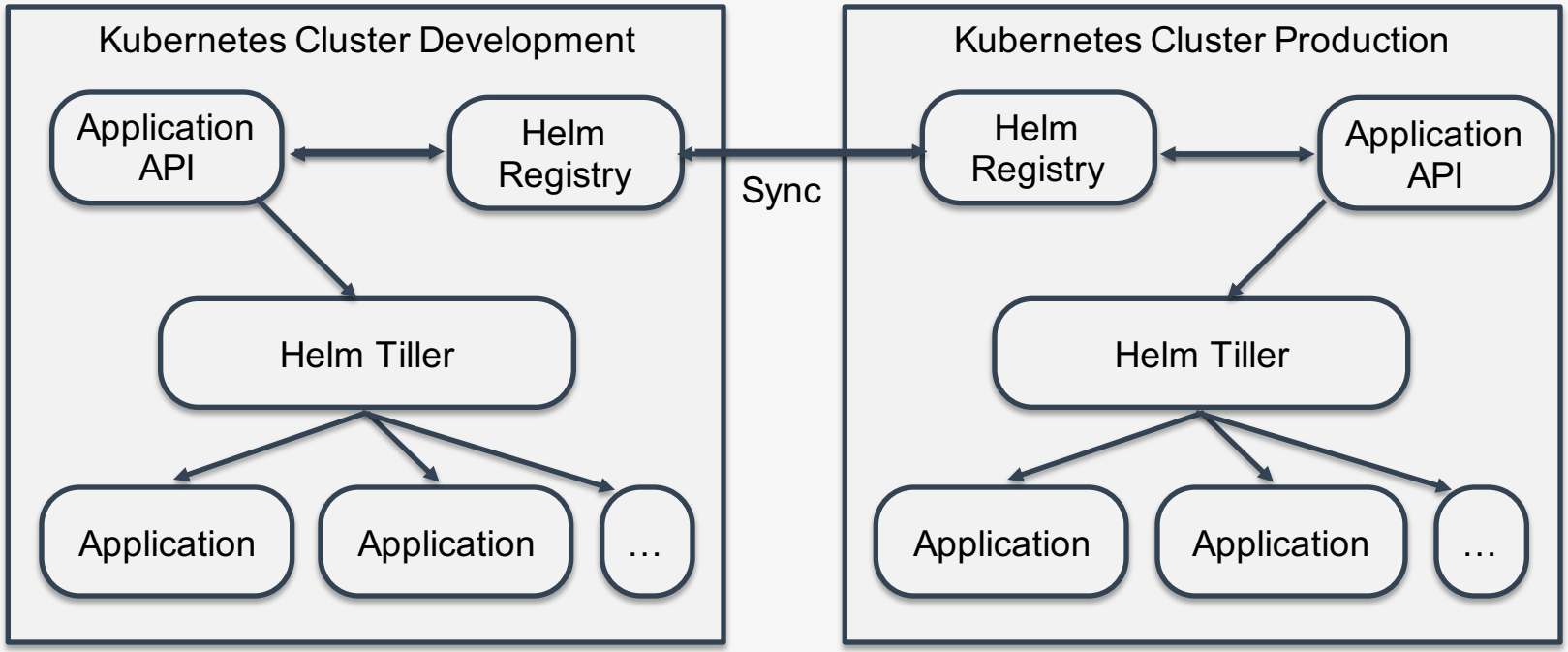
```
|- space
 |- chart
  |- version
```

2. 可选的多种存储后端（FileSystem，OSS，…）
3. 可通过 API 直接对应用进行编排
4. 开源（https://github.com/caicloud/helm-registry）

Helm Tiller

1. 负责将 Chart 部署到指定的集群当中，并管理生成的 Release（应用）
2. 支持对 Release 的 更新，删除，回滚 操作
3. 支持对 Release 的资源进行增量更新
4. Release 的状态管理
5. Kubernetes 下属子项目（https://github.com/kubernetes/helm）

多集群架构


Kubernetes Cluster Development
Application API
Helm Registry
Helm Tiller
Application
Application
…
Sync
Kubernetes Cluster Production
Helm Registry
Application API
Helm Tiller
Application
Application
…

caicloud
才云

谢谢大家！