

## 引用对象

- Docker
  - @DockerContainer //负责应用程序的运行，包括操作系统、用户添加的文件以及元数据
  - @DockerImage //是一个只读模板，用来运行Docker容器
  - @DockerFile //文件指令集，用来说明如何自动创建Docker镜像
- Kubernetes
  - @K8sObject //引入kubernetes部署体系
  - @K8sPod : @K8sObject : : Inherited //最小调度单位
    - .spec.serviceAccountName
      - 为pod提供必要的身份认证
      - 方便Pod里面的进程调用Kubernetes API或其他外部服务而设计的
      - User account是跨namespace的，而service account则是仅局限它所在的namespace；
      - 每个namespace都会自动创建一个default service account
  - .status
    - phase
      - 挂起(Pending)
        - Pod 已被 Kubernetes 系统接受，但有一个或者多个容器镜像尚未创建。等待时间包括调度 Pod 的时间和通过网络下载镜像的时间，这可能需要花点时间。
      - 运行(Running)
        - 该 Pod 已经绑定到了一个节点上，Pod 中所有的容器都已被创建。至少有一个容器正在运行，或者正处于启动或重启状态。
      - 成功(Succeeded) //表示通过检测
        - Pod 中的所有容器都被成功终止，并且不会再重启。
      - 失败(Failed) //表示没有通过检测
        - Pod 中的所有容器都已终止了，并且至少有一个容器是因为失败终止。也就是说，容器以非0状态退出或者被系统终止。
      - 未知(Unknown) //表示检测没有正常进行
        - 因为某些原因无法取得 Pod 的状态，通常是因为与 Pod 所在主机通信失败。
    - conditions //状态详情
      - type
        - Ready
      - status
        - true
        - false
        - unknown
      - lastProbeTime //最近一次获取condition信息的时间。
      - lastTransitionTime //最后一次condition状态发生变化的时间。
      - reason //机器读的原因。特点：唯一性、单个词语描述、驼峰式显示。
      - message //最后一次condition状态变化的详情，以人可读的方式显示。
    - message
    - reason
    - hostIP
    - podIP

- startTime
- .@containerStatuses //容器状态
  - .status
    - name //唯一，无法更改
    - .state //当前状态详情
      - waiting //容器的等待状态信息
        - reason
        - message
      - running //容器运行状态信息
        - startedAt //容器上一次启动或者重启的时间
      - terminated //容器终止状态信息
        - exitCode //integer (int32),Exit status from the last termination of the container
        - signal //integer (int32),Signal from the last termination of the container
        - reason //string, (brief) reason from the last termination of the container
        - message //string,Message regarding the last termination of the container
        - startedAt //string,Time at which previous execution of the container started
        - finishedAt //string,Time at which the container last terminated
        - containerID //string,Container's ID in the format docker://<container\_id>
    - .lastState //终止状态详情
    - .ready //容器是否已经传入了活性检测探针。
    - .restartCount //容器被重启的次数，计数是基于已经死掉且没有被删除的容器个数。注意：由于垃圾回收机制，该值不会大于5。
    - .image //容器使用的镜像。
    - imageID //容器使用的镜像的ID。
    - containerID //容器的ID，以docker://<container\_id>的格式显示。
  - @K8sDeployment : @K8sObject : : Inherited
  - @K8sService : @K8sObject : : Inherited
  - @K8sIngress : @K8sObject : : Inherited
    - #Annotation
      - Ingress Controller 负责实现路由需求， Ingress负责描述路由需求
      - Ingress Controller 一个集群可以有多个
      - Ingress Controller 通过Configmap设置， Ingress 通过Annotations设置
      - Ingress Controller 设置全局规则， Ingress 设置局部规则
      - Ingress Controller 可为多个命名空间服务。
      - 需求供给分离可以做到权限隔离，又能提供配置能力。
  - @K8sCluster : @K8sObject : : Inherited
    - .status
      - .Address
        - HostName: 可以被kubelet中的--hostname-override参数替代。
        - ExternalIP: 可以被集群外部路由到的IP地址。
        - InternalIP: 集群内部使用的IP，集群外部无法访问。
      - .Condition
        - OutOfDisk: 磁盘空间不足时为True
        - Ready: Node controller 40秒内没有收到node的状态报告为Unknown，健康为True，否则为False。

- MemoryPressure: 当node没有内存压力时为True, 否则为False。
- DiskPressure: 当node没有磁盘压力时为True, 否则为False。
- .Capacity
  - CPU
  - 内存
  - 可运行的最大Pod个数
- .Info: 节点的一些版本信息, 如OS、kubernetes、docker等
- @K8sNode : @K8sObject : : Inherited
- @K8sMaster : @K8sObject : : Inherited
- @K8sEndpoint : @K8sObject : : Inherited