

# pod结构

kubectl explain pod [查看](#)

```
1  apiVersion: v1      #必选, 版本号, 例如v1
2  kind: Pod           #必选, 资源类型, 例如 Pod
3  metadata:           #必选, 元数据
4    name: string      #必选, Pod名称
5    namespace: string #Pod所属的命名空间,默认为"default"
6    labels:            #自定义标签列表
7      - name: string
8  spec: #必选, Pod中容器的详细定义
9    containers: #必选, Pod中容器列表
10     - name: string #必选, 容器名称
11       image: string #必选, 容器的镜像名称
12       imagePullPolicy: [ Always|Never|IfNotPresent ] #获取镜像的策略
13       command: [string] #容器的启动命令列表, 如不指定, 使用打包时使用的启动命令
14       args: [string] #容器的启动命令参数列表
15       workingDir: string #容器的工作目录
16       volumeMounts: #挂载到容器内部的存储卷配置
17         - name: string #引用pod定义的共享存储卷的名称, 需用volumes[]部分定义的卷名
18           mountPath: string #存储卷在容器内mount的绝对路径, 应少于512字符
19           readOnly: boolean #是否为只读模式
20     ports: #需要暴露的端口库号列表
21       - name: string #端口的名称
22         containerPort: int #容器需要监听的端口号
23         hostPort: int #容器所在主机需要监听的端口号, 默认与Container相同
24         protocol: string #端口协议, 支持TCP和UDP, 默认TCP
25     env: #容器运行前需设置的环境变量列表
26       - name: string #环境变量名称
27         value: string #环境变量的值
28     resources: #资源限制和请求的设置
29       limits: #资源限制的设置
30         cpu: string #Cpu的限制, 单位为core数, 将用于docker run --cpu-shares参数
31         memory: string #内存限制, 单位可以为Mib/Gib, 将用于docker run --memory参数
32       requests: #资源请求的设置
33         cpu: string #Cpu请求, 容器启动的初始可用数量
34         memory: string #内存请求, 容器启动的初始可用数量
35     lifecycle: #生命周期钩子
36       postStart: #容器启动后立即执行此钩子, 如果执行失败, 会根据重启策略进行重启
37       preStop: #容器终止前执行此钩子, 无论结果如何, 容器都会终止
38     livenessProbe: #对Pod内各容器健康检查的设置, 当探测无响应几次后将自动重启该容器
39     exec: #对Pod容器内检查方式设置为exec方式
```

```

40     command: [string] #exec方式需要制定的命令或脚本
41 httpGet:           #对Pod内个容器健康检查方法设置为HttpGet，需要制定Path、port
42     path: string
43     port: number
44     host: string
45     scheme: string
46     HttpHeaders:
47     - name: string
48       value: string
49 tcpSocket:         #对Pod内个容器健康检查方式设置为tcpSocket方式
50     port: number
51     initialDelaySeconds: 0      #容器启动完成后首次探测的时间，单位为秒
52     timeoutSeconds: 0          #对容器健康检查探测等待响应的超时时间，单位秒，默认1秒
53     periodSeconds: 0           #对容器监控检查的定期探测时间设置，单位秒，默认10秒一次
54     successThreshold: 0
55     failureThreshold: 0
56     securityContext:
57     privileged: false
58 restartPolicy: [Always | Never | OnFailure] #Pod的重启策略
59 nodeName: <string> #设置nodeName表示将该Pod调度到指定名称的node节点上
60 nodeSelector: object #设置nodeSelector表示将该Pod调度到包含这个label的node上
61 imagePullSecrets: #Pull镜像时使用的secret名称，以key: secretkey格式指定
62 - name: string
63 hostNetwork: false #是否使用主机网络模式，默认为false，如果设置为true，表示使用宿主机网络
64 volumes:          #在该pod上定义共享存储卷列表
65 - name: string     #共享存储卷名称（volumes类型有很多种）
66   emptyDir: {}      #类型为emptyDir的存储卷，与Pod同生命周期的一个临时目录。为空值
67   hostPath: string  #类型为hostPath的存储卷，表示挂载Pod所在宿主机的目录
68     path: string    #Pod所在宿主机的目录，将被用于同期中mount的目录
69   secret:           #类型为secret的存储卷，挂载集群与定义的secret对象到容器内部
70     secretName: string
71     items:
72     - key: string
73       path: string
74   configMap:        #类型为configMap的存储卷，挂载预定义的对象到容器内部
75     name: string
76     items:
77     - key: string
78       path: string

```

## 在kubernetes中基本所有资源的一级属性都是一样的，主要包含5个部分：

- **apiVersion** <string>：版本，有kubernetes内部定义，版本号必须用kubectI api-versions查询。
- **kind** <string>：类型，有kubernetes内部定义，类型必须用kubectI api-resources查询。
- **metadata** <Object>：元数据，主要是资源标识和说明，常用的有name、namespace、labels等。
- **spec** <Object>：描述，这是配置中最重要的一部分，里面是对各种资源配置的详细描述。
- **status** <Object>：状态信息，里面的内容不需要定义，由kubernetes自动生成。

## 在上面的属性中，spec是接下来研究的重点，继续看下它的常见子属性：

- **containers** <[]Object>：容器列表，用于定义容器的详细信息。
- **nodeName** <String>：根据nodeName的值将Pod调度到指定的Node节点上。
- **nodeSelector** <map[]>：根据NodeSelector中定义的信息选择该Pod调度到包含这些Label的Node上。
- **hostNetwork** <boolean>：是否使用主机网络模式，默认为false，如果设置为true，表示使用宿主网络。
- **volumes** <[]Object>：存储卷，用于定义Pod上面挂载的存储信息。
- **restartPolicy** <string>：重启策略，表示Pod在遇到故障时候的处理策略。