

k8s集群资源清单(YAML)文件书写方法

一、场景

通过前面课程的学习，播仔已经可以使用了kubectl命令在k8s集群中进行简单的操作，操作过程中播仔发现，对于一些基础查询类命令操作起来非常方便，但是对于对k8s集群中资源管理及大量资源对象编排部署（创建或删除等）操作，使用简单的命令行工具是无法满足要求的，那么如何满足播仔使用k8s集群的需求呢？

k8s集群中对资源管理和资源对象编排部署都可以通过声明样式（YAML）文件来解决，也就是可以把需要对资源对象操作编辑到YAML格式文件中，我们把这种文件叫做资源清单文件，通过kubectl命令直接使用资源清单文件就可以实现对大量的资源对象进行编排部署了。

但是，声明样式的资源清单文件应该如何编写呢？这就是接下来我们要学习的内容。

二、学习目标

- 了解YAML文件书写格式

- 了解通过YAML文件实现资源清单描述方法

三、学习步骤

序号	步骤	备注
1	YAML文件书写格式	
2	通过YAML文件实现资源清单描述方法	

四、课程内容

4.1 YAML文件书写格式

4.1.1 YAML介绍

YAML：仍是一种标记语言。

为了强调这种语言以数据做为中心，而不是以标记语言为重点。

YAML是一个可读性高，用来表达数据序列的格式。

4.1.2 基本语法

- 使用空格做为缩进
- 缩进的空格数目不重要，只要相同层级的元素左侧对齐即可
- 低版本缩进时不允许使用Tab键，只允许使用空格
- 使用#标识注释，从这个字符一直到行尾，都会被解释器忽略

4.1.3 YAML支持的数据结构

- 对象
 - 键值对的集合
 - 又称为映射(mapping) / 哈希 (hashes) / 字典 (dictionary)

```
1 对象类型：对象的一组键值对，使用冒号结构表示
2  name: Tom
3  age: 18
4
5  Yaml 也允许另一种写法，将所有键值对写成一个行内对象
6  hash: { name: Tom, age: 18 }
```

- 数组：
 - 一组按次序排列的值
 - 又称为序列 (sequence) / 列表 (list)

```
1 数组类型：一组连词线开头的行，构成一个数组
2  People
3  - Tom
4  - Jack
5
6 数组也可以采用行内表示法
7  People: [Tom, Jack]
```

- 纯量 (scalars) :
 - 单个的、不可再分的值

```
1 纯量：纯量是最基本的、不可再分的值。以下数据类型都属于
   纯量
2
3
4 数值直接以字面量的形式表示
5  number: 12.30
6
7 布尔值用true和false表示
8  isSet: true
9
10 null用 ~ 表示
11 parent: ~
12
13 时间采用 ISO8601 格式
14 iso8601: 2001-12-14t21:59:43.10-05:00
15
16 日期采用复合 iso8601 格式的年、月、日表示
17 date: 1976-07-31
18
19 YAML 允许使用两个感叹号，强制转换数据类型
```

```
20 e: !!str 123
21 f: !!str true
```

```
1 字符串默认不使用引号表示
2
3 str: 这是一行字符串
4
5
6 如果字符串之中包含空格或特殊字符，需要放在引号之中
7
8 str: '内容:   字符串'
9
10
11 单引号和双引号都可以使用，双引号不会对特殊字符转义
12 s1: '内容\n字符串'
13 s2: "内容\n字符串"
14
15
16 单引号之中如果还有单引号，必须连续使用两个单引号转义
17 str: 'labor''s day'
18
19
20 字符串可以写成多行，从第二行开始，必须有一个单空格缩
    进。换行符会被转为 空格
21 str: 这是一段
22     多行
23     字符串
24
25
26 多行字符串可以使用|保留换行符，也可以使用>折叠换行
27 this: |
28 Foo
29 Bar
```

```
30 | that
31 | Foo
32 | Bar
```

4.2 资源清单描述方法

在k8s中，一般使用YAML格式的文件来创建符合我们预期期望的pod,这样的YAML文件称为资源清单。

4.2.1 常用字段

参数名	字段类型	说明
version	String	这里是指的是K8S API的版本，目前基本上是v1，可以用kubectl api-versions命令查询
kind	String	这里指的是yam文件定义的资源类型和角色，比如:Pod
metadata	Object	元数据对象，固定值就写metadata
metadata.name	String	元数据对象的名字，这里由我们编写，比如命名Pod的名字
metadata.namespace	String	元数据对象的命名空间，由我们自身定义
Spec	Object	详细定义对象，固定值就写Spec
spec. containers[]	list	这里是Spec对象的容器列表定义，是个列表
spec containers [].name	String	这里定义容器的名字
spec.containers [].image	String	这里定义要用到的镜像名称

参数名	字段类型	说明
spec.containers [].imagePullPolicy	String	定义镜像拉取策略，有 Always、Never、IfNotPresent三个值可选(1) Always:意思是每次都尝试重新拉取镜像(2) Never:表示仅使用本地镜像(3) IfNotPresent:如果本地有镜像就使用本地镜像，没有就拉取在线镜像。上面三个值都没设置的话，默认是 Always。
spec.containers [].command[]	List	指定容器启动命令，因为是数组可以指定多个。不指定则使用镜像打包时使用的启动命令。
spec.containers [].args	List	指定容器启动命令参数，因为是数组可以指定多个。
spec.containers [].workDir	String	指定容器的工作目录

参数名	字段类型	说明
spec.containers[]. volumeMounts[]	List	指定容器内部的存储卷配置
spec.containers[]. volumeMounts[].name	String	指定可以被容器挂载的存储卷的名称
spec.containers[]. volumeMounts[].mountPath	String	指定可以被容器挂载的存储卷的路径
spec.containers[]. volumeMounts[].readOnly	String	设置存储卷路径的读写模式，ture或者false，默认为读写模式
spec.containers [].ports[]	String	指定容器需要用到的端口列表
spec.containers [].ports[].name	String	指定端口名称
spec.containers [].ports[].containerPort	String	指定容器需要监听的端口号
spec.containers [].ports[].hostPort	String	指定容所在主机需要监听的端口号，默认跟上面containerPort相同，注意设置了 hostPort同一台主机无法启动该容器的相同副本(因为主机的端口号不能相同，这样会冲突)

参数名	字段类型	说明
spec.containers [].ports[].protocol	String	指定端口协议，支持TCP和UDP，默认值为TCP
spec.containers [].env[]	List	指定容器运行前需设的环境变量列表
spec.containers [].env[].name	String	指定环境变量名称
spec.containers [].env[].value	String	指定环境变量值
spec.containers[].resources	Object	指定资源限制和资源请求的值（这里开始就是设置容器的资源上限）
spec.containers[].resources.limits	Object	指定设置容器运行时资源的运行上限
spec.containers[].resources.limits.cpu	String	指定CPU限制，单位为core数，将用于docker run -- cpu-shares参数
spec.containers[].resources.limits.memory	String	指定MEM内存的限制，单位为MiB、GiB
spec.containers[].resources.requests	Object	指定容器启动和调度时的限制设置

参数名	字段类型	说明
spec.containers[].resources.requests.cpu	String	CPU请求，单位为core数，容器启动时初始化可用数量
spec.containers[].resources.requests.memory	String	内存请求，单位为MiB、GiB，容器启动时初始化可用数量

参数名	字段类型	说明
sepc.restartPolicy	String	<p>定义Pod的重启策略，可选值为Always、OnFailure,默认值为Always。</p> <p>1.Always:Pod一旦终止运行，则无论容器时如何终止的，kubelet服务都将重启它。</p> <p>2.OnFailure:只有Pod以非零退出码终止时，kubelet才会重启该容器。如果容器正常结束（退出码为0），则kubelet将不会重启它。</p> <p>3.Never:Pod终止后，kubelet将退出码报告给Master,不会重启该Pod。</p>
spec.nodeSelector	Object	<p>定义Node的Label过滤标签，以key:value格式指定。</p>

参数名	字段类型	说明
spec.imagePullSecrets	Object	定义pull镜像时使用secret名称，以name:secretkey格式指定。
spec.hostNetwork	Boolean	定义是否使用主机网络模式，默认值为false。设置true表示使用宿主机网络，不使用docker网桥，同时设置了true将无法在同一台宿主机上启动第二个副本。

4.2.2 举例说明

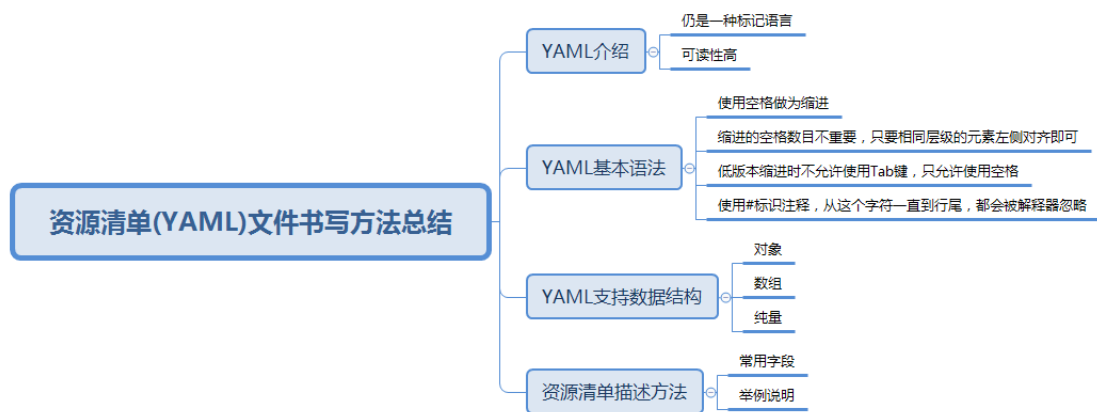
- 创建一个namespace

```
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: test
```

- 创建一个pod

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: pod1
5 spec:
6   containers:
7   - name: nginx-containers
8     image: nginx:latest
```

五、学习总结



六、课程预约

关于YAML声明式文件更多学习内容，可预约《kubernetes从入门到企业应用实战》相关课程。