

Kubernetes YML 语法参考

YAML 基础

YAML 是一个可读性高，用来表达数据序列的格式。它使用空白符号缩进和大量依赖外观的特色，特别适合用来表达或编辑数据结构、各种配置文件，其让人最容易上手的特色是巧妙避开各种封闭符号，如：引号、各种括号等，这些符号在嵌套结构时会变得复杂而难以辨认。

JSON 是 YAML 的子集，因此大部分的 JSON 文件都是有效的 YAML 文件。在书写 YAML 文件时可以加入选择性的空行，以增加可读性。在一个文件中，可同时包含多个文件，并用 --- 分隔。选择性的符号 ... 可以用来表示文件结尾（在利用流的通信中，这非常有用，可以在不关闭流的情况下，发送结束信号）。

它的基本语法规则如下。

- 大小写敏感
- 使用缩进表示层级关系
- 缩进时不允许使用Tab键，只允许使用空格。
- 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
- # 表示注释，从这个字符一直到行尾，都会被解析器忽略。

YAML 中常用的基本类型只有两种：Lists 和 Maps。

YAML Lists

Lists 习惯上比较常用区块格式 (block format) 表示，也就是用短杠+空白字符作为起始。

```
args
- sleep
- "1000"
- message
- "hello, world"
```

与之等价的 JSON 格式为：

```
{
  "args": ["sleep", "1000", "message", "hello, world"]
}
```

```
}
```

YAML Maps

键值和数据由冒号及空白字符分开。区块形式（常使用与YAML数据文档中）使用缩进和换行符分隔key: value对。比如：

```
---
```

```
apiVersion: v1
```

```
kind: Pod
```

与之等价的 JSON 格式为

```
{
```

```
    "apiVersion": "v1",
```

```
    "kind": "Pod"
```

```
}
```

嵌套使用基本类型可以表达更加复杂的数据结构：

```
---
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
    name: nginx-deployment
```

```
    label:
```

```
        foo: bar
```

与之等价的 JSON 格式为

```
{
```

```
    "apiVersion": "v1",
```

```
    "kind": "Pod",
```

```
    "metadata": {
```

```
        "name": "nginx-deployment",
```

```
        "label": {
```

```
            "foo": "bar"
```

```
        }
```

```
    }
```

```
}
```

层次化结构

在 Lists 中嵌套 Maps

containers

- name: front-end
image: nginx
- name: back-end
image: python

与之等价的 JSON 格式为

```
{  
    "containers": [  
        {  
            "name": "front-end",  
            "image": "nginx"  
        }, {  
            "name": "back-end",  
            "image": "python"  
        }  
    ]  
}
```

在 Maps 中嵌套 Lists:

restartPolicy: Always
containers

- name: front-end
image: nginx
- name: back-end
image: python

与之等价的 JSON 格式为

```
{  
    "restartPolicy": "Always",  
    "containers": [  
        {  
            "name": "front-end",  
            "image": "nginx"  
        }, {  
            "name": "back-end",  
            "image": "python"  
        }  
    ]  
}
```

```
]
}
```

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.11/>

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.14/>