

1.1 Pod 实验

1.1.1 使用 Pod

步骤 1 创建 podfile 目录

```
[root@k8s-master]# mkdir /labfile/podfile
[root@k8s-master]# cd /labfile/podfile
```

步骤 2 编写 mypod.yaml

```
[root@k8s-master podfile]# vim mypod.yaml
```

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: mypod
      image: busybox
      args:
        - /bin/sh
        - -c
        - sleep 30000
```

步骤 3 创建 pod

```
[root@k8s-master podfile]# kubectl apply -f mypod.yaml
```

```
pod/mypod created
```

步骤 4 查看 pod

```
[root@k8s-master podfile]# kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
mypod	1/1	Running	0	72s

步骤 5 查看 yaml 输出格式的 pod 信息

```
[root@k8s-master podfile]# kubectl get pod mypod -o yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
```

```
kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"Pod","metadata":{"annotations":{},"name":"mypod","namespace":"default"},"spec":{"containers":[{"args":["/bin/sh","-c","sleep 30000"],"image":"busybox","name":"mypod"}]}}
  creationTimestamp: "2019-06-28T11:01:30Z"
    name: mypod
    namespace: default
    resourceVersion: "1258931"
    selfLink: /api/v1/namespaces/default/pods/mypod
    uid: 15241cde-9994-11e9-ac91-000c290a92ce
  spec:
    containers:
      - args:
        - /bin/sh
        - -c
        - sleep 30000
        image: busybox
        imagePullPolicy: Always
        name: mypod
.....
```

可以看到，回显的信息比我们创建的 yaml 文件丰富许多，因为许多我们未在 yaml 中定义参数系统使用默认值配置。

步骤 6 使用 explain 命令

如果我们对 yaml 中的某些参数的值意味什么意思，该如何填写没有把握，可以使用 explain 命令。

如果希望了解关于 pod 中 apiVersion 的相关信息：

```
[root@k8s-master podfile]# kubectl explain pod.apiVersion
```

```
KIND:      Pod
VERSION:   v1

FIELD:     apiVersion <string>

DESCRIPTION:
  APIVersion defines the versioned schema of this representation of an
  object. Servers should convert recognized schemas to the latest internal
  value, and may reject unrecognized values. More info:
  https://git.k8s.io/community/contributors/devel/api-conventions.md#resources
```

如果希望了解关于 Pod 中，spec 里，container 下 imagePullPolicy 的定义：

```
[root@k8s-master podfile]# kubectl explain pod.spec.containers.imagePullPolicy
```

```
KIND:      Pod
VERSION:   v1

FIELD:     imagePullPolicy <string>

DESCRIPTION:
  Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always
  if :latest tag is specified, or IfNotPresent otherwise. Cannot be
  updated.
  More info:
  https://kubernetes.io/docs/concepts/containers/images#updating-images
```

步骤 7 进入 mypod 容器的命令行交互界面

```
[root@k8s-master podfile]# kubectl exec -it mypod /bin/sh
```

```
/ #
```

使用 ls 命令查看容器内目录

```
/ # ls
```

```
bin  dev  etc  home  proc  root  sys  tmp  usr  var
```

步骤 8 退出容器

在容器内输入 exit 可以退出容器

```
/ # exit
```

步骤 9 查看容器运行位置

```
[root@k8s-master podfile]# kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED
mypod	1/1	Running	0	15m	10.244.2.74	k8s-master	<none>

步骤 10 可以看到容器运行位置位于 k8s-master，切换到主机 k8s-master 中，使用 docker 命令查看运行在该主机上的容器。通过容器的名称找到 mypod 中的容器。

```
[root@k8s-master hostpathdir]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
b9de9f770a08	busybox	"/bin/sh -c 'sleep 3_'"	16 minutes ago
k8s_mypod_mypod_default_15241cde-9994-11e9-ac91-000c290a92ce_0			Up 16 minutes
11edble081a8	k8s.gcr.io/pause:3.1	"/pause"	16 minutes ago
k8s_POD_mypod_default_15241cde-9994-11e9-ac91-000c290a92ce_0			Up 16 minutes

1.1.2 运行一次性 Pod

步骤 1 创建 helloworld.yaml 文件

```
[root@k8s-master podfile]# vim helloworld.yaml
```

```
kind: Pod
apiVersion: v1
metadata:
  name: helloworld
spec:
  restartPolicy: Never
  containers:
  - name: helloworld
    image: hello-world
```

步骤 2 从 helloworld.yaml 创建 Pod

```
[root@k8s-master podfile]# kubectl apply -f helloworld.yaml
```

步骤 3 查看 pod 状态

```
[root@k8s-master podfile]# kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
helloworld	0/1	Completed	0	2s

步骤 4 查看 pod 输出信息

```
[root@k8s-master podfile]# kubectl logs helloworld
```

出现如下信息表示成功

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

步骤 5 删除本次实验创建的 pod