

33.emptyDir卷

emptyDir卷

使用emptyDir

使用medium属性

首先让我来看看如何在一个pod的多个容器中共享数据。

emptyDir卷

emptyDir是最简单的卷类型，它从一个空目录开始。运行在pod中的应用程序可以将其需要的任何文件写入到卷中。由于卷的生命周期与pod是绑定的，所以当删除pod时，卷的内容也会丢失。

emptyDir卷对于运行在同一个pod中的容器之间共享文件特别有用。但是当单个容器需要临时写数据到磁盘时（比如在一个大数据集上进行排序操作，该数据集无法装入可用内存），它也可发挥作用。数据也可以被写入到容器的文件系统本身，但是这两者之间有细微的差别。容器的文件系统甚至可能都不是可写的，所以写入到挂载的卷中可能是唯一的选择。

使用emptyDir

我们简化一下上面提到的例子，创建一个pod，只包含WebServer和ContentAgent两个容器以及一个用于存储HTML的卷。

我们使用Nginx作为web服务器，该部分通过nginx:alpine镜像实现；使用Unix fortune命令来生成HTML内容。每次运行fortune命令都会输出一个随机的引用。创建一个脚本，每隔10秒调用fortune命令并将其输出写入到index.html文件。这一部分通过pmx/fortune镜像实现。

构建pmx/fortune镜像

创建一个名为fortune的目录

```
mkdir fortune
```

在这个目录下创建一个名为fortuneloop.sh的shell脚本

```
vim fortuneloop.sh
```

内容如下：

```
#!/bin/bash
```

```
trap "exit" SIGINT
```

```
mkdir /var/htdocs
```

```
while :
```

```
do
```

```
echo $(date) Writing fortune to /var/htdocs/index.html
/usr/games/fortune > /var/htdocs/index.html
sleep 10
done
```

```
#!/bin/bash
trap "exit" SIGINT
mkdir /var/htdocs

while :
do
    echo $(date) Writing fortune to /var/htdocs/index.html
    /usr/games/fortune > /var/htdocs/index.html
    sleep 10
done
```

上面脚本第二行表示该脚本监听并拦截的信号，当通过Ctrl+C组合键产生SIGINT（终止进程）信号时，就会终止脚本执行。

在fortune目录下创建Dockerfile：

```
FROM ubuntu:latest
RUN apt-get update ; apt-get -y install fortune
ADD fortuneloop.sh /bin/fortuneloop.sh
WORKDIR /bin
RUN chmod +x /bin/fortuneloop.sh
ENTRYPOINT /bin/fortuneloop.sh
```

```
FROM ubuntu:latest
RUN apt-get update ; apt-get -y install fortune
ADD fortuneloop.sh /bin/fortuneloop.sh
ENTRYPOINT /bin/fortuneloop.sh
~
~
```

准备好这两个文件后就可以开始构建镜像了：

`docker build -t pmx/fortune .`

```
[root@k8s-master fortune]# docker build -t pmx/fortune .
Sending build context to Docker daemon 3.072kB
Step 1/4 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
Digest: sha256:c95a8e48bf88e9849f3e0f723d9f49fa12c5a00cfc6e60d2bc99d87555295e4c
Status: Downloaded newer image for ubuntu:latest
--> f643c72bc252
Step 2/4 : RUN apt-get update ; apt-get -y install fortune
--> Running in 0665323c7a68
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [21.6 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [682 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [197 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [718 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
```

将镜像推送到阿里云：

`docker tag db09f996200e registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest`

```
docker push registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest
```

```
[root@k8s-master fortune]# docker tag db09f996200e registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest
[root@k8s-master fortune]# docker push registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest
The push refers to repository [registry.cn-shanghai.aliyuncs.com/pmx/fortune]
32e12ac05901: Pushed
b459e4142280: Pushed
f6253634dc78: Pushed
9069f84dbbe9: Pushed
bacd3af13903: Pushed
latest: digest: sha256:7b15d7d3c1b66802079a295ff81b81e74c9be396d004b80b2e399f070412a2fd size: 1362
[root@k8s-master fortune]#
```

创建pod

```
vim fortune-pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: fortune
spec:
  containers:
    - image: registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest
      name: html-generator
      volumeMounts:
        - name: html
          mountPath: /var/htdocs
    - image: nginx:alpine
      name: web-server
      volumeMounts:
        - name: html
          mountPath: /usr/share/nginx/html
          readOnly: true
      ports:
        - containerPort: 80
          protocol: TCP
  volumes:
    - name: html
      emptyDir: {}
```

apiVersion: v1

kind: Pod

metadata:

name: fortune

spec:

containers:

– image: registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest

name: html-generator

volumeMounts:

– name: html

mountPath: /var/htdocs

– image: nginx:alpine

name: web-server

volumeMounts:

– name: html

mountPath: /usr/share/nginx/html

```
readOnly: true
ports:
- containerPort: 80
  protocol: TCP
volumes:
- name: html
  emptyDir: {}
```

kubectl create -f fortune-pod.yaml

```
[root@k8s-master test]# kubectl create -f fortune-pod.yaml
pod/fortune created
[root@k8s-master test]# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
dnsutils	1/1	Running	2	84d
fortune	0/2	ContainerCreating	0	7s
test-tomcat-deploy-6b8b5545b6-6jpxg	1/1	Running	1	56d
test-tomcat-deploy-6b8b5545b6-mlnhm	1/1	Running	1	56d
test-tomcat-deploy-6b8b5545b6-vhxn2	1/1	Running	1	56d

```
[root@k8s-master test]# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
dnsutils	1/1	Running	2	84d
fortune	2/2	Running	0	61s
test-tomcat-deploy-6b8b5545b6-6jpxg	1/1	Running	1	56d
test-tomcat-deploy-6b8b5545b6-mlnhm	1/1	Running	1	56d
test-tomcat-deploy-6b8b5545b6-vhxn2	1/1	Running	1	56d

这个pod包含两个容器以及一个挂载到两个容器中不同路径的卷。

第一容器命名为html-generator，运行registry.cn-shanghai.aliyuncs.com/pmx/fortune:latest镜像，名为html的卷挂载到该容器的/var/htdocs目录上。

第二个容器命名为web-server，运行nginx:alpine镜像，html卷挂载到该容器的/usr/share/nginx/html目录上并且设置为read-only。

html-generator容器启动后会每隔10秒将fortune命令的输出内容写入到/var/htdocs/index.html文件。由于卷是挂载到/var/htdocs目录上的，index.html文件会被写入到卷中而不是容器的顶层。一旦web-server容器启动，它就开始为/usr/share/nginx/html目录（这是Nginx默认的服务目录）中的任意HTML文件提供服务。由于我们将卷挂载在这个位置，Nginx将为运行fortune循环的容器输出的index.html文件服务。最终效果是，发送HTTP请求到pod的80端口的客户端将接收当前fortune消息作为响应。

查看消息

如果想查看上面web-server返回的消息，我们首先需要能够访问pod。可以使用端口转发将本地的某个端口映射到pod的80端口上。

执行如下命令：

kubectl port-forward fortune 8080:80

```
[root@k8s-master ~]# kubectl port-forward fortune 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

现在执行如下命令查看返回的随机消息：

```
curl http://localhost:8080
```

```
[root@k8s-master ~]# curl http://localhost:8080
Q:   Where's the Lone Ranger take his garbage?
A:   To the dump, to the dump, to the dump dump dump!

Q:   What's the Pink Panther say when he steps on an ant hill?
A:   Dead ant, dead ant, dead ant dead ant dead ant...
[root@k8s-master ~]#
```

等待几秒钟再次执行即可看到不同的消息：

```
[root@k8s-master ~]# curl http://localhost:8080
Q:   What did Tarzan say when he saw the elephants coming over the hill?
A:   "The elephants are coming over the hill."

Q:   What did he say when saw them coming over the hill wearing
      sunglasses?
A:   Nothing, for he didn't recognize them.
[root@k8s-master ~]#
```

使用medium属性

在上面的例子中，我们使用的卷类型为emptyDir，它实际上位于pod所在工作节点的磁盘上，因此其性能取决于节点磁盘的类型。我们还可以让Kubernetes在一个tmpfs文件系统（在内存中，而不是磁盘上）上创建emptyDir。具体方法如下：

volumes:

- name: html

- emptyDir:

- medium: Memory

这样，emptyDir中的文件就会于内存中。