Volumes：

一些需要做持久化或者有状态的应用程序才会用到Volumes。

Redis-Cluster：nodes.conf

日志收集的需求：收集容器内指定文件的日志。Sidecar的方式在Pod里面加入一个收集的日志的容器，Volumes将应用程序的日志共享到sidecar容器里面。

Nginx+PHP

1. 需要持久化数据
2. 需要共享目录

挂载HostPath：

volumes:

- name: test-hostpath

hostPath:

path: /etc/hosts

type：可选字段，默认是空字符串，挂载之前不会执行任何检查

* DirectoryOrCreate： 如果不存在这个目录，就创建一个权限为0755的空目录，kubelet
* Directory：目录必须存在
* FileOrCreate：如果不存在这个文件，就创建一个权限为0644的空文件，
* File：挂载的这个文件必须存在
* Socket：挂载一个UNIX套接字。Docker.sock，必须存在
* CharDevice：字符设备，必须存在
* BlockDevice：跨设备，必须存在

EmptyDir：

主要是用于一个Pod中有多个容器，然后容器之前需要共享数据或目录。删除Pod时EmptyDir类型的卷数据也将会被删除。

apiVersion: apps/v1

kind: Deployment

metadata:

annotations:

deployment.kubernetes.io/revision: "13"

creationTimestamp: "2020-03-16T14:36:14Z"

generation: 14

labels:

app: demo-nginx

name: demo-nginx

namespace: default

resourceVersion: "2415976"

selfLink: /apis/apps/v1/namespaces/default/deployments/demo-nginx

uid: ca0c135f-31a3-464c-b1f2-10f88e9f1c90

spec:

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

app: demo-nginx

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 1

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

app: demo-nginx

spec:

containers:

- command:

- sh

- -c

- sleep 36000000000

image: nginx

imagePullPolicy: IfNotPresent

name: nginx2

resources: {}

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

volumeMounts:

- mountPath: /mnt

name: cache-volume

- command:

- sh

- -c

- sleep 36000000000

image: nginx

imagePullPolicy: IfNotPresent

name: nginx

ports:

- containerPort: 80

name: web

protocol: TCP

resources:

limits:

cpu: 100m

memory: 270Mi

requests:

cpu: 100m

memory: 70Mi

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

volumeMounts:

- mountPath: /etc/nginx/nginx.conf

name: config-volume

subPath: etc/nginx/nginx.conf

- mountPath: /mnt/

name: config-volume-non-subpath

- mountPath: /tmp/1

name: test-hostpath

- mountPath: /tmp/2

name: cache-volume

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

shareProcessNamespace: true

terminationGracePeriodSeconds: 30

volumes:

- hostPath:

path: /etc/hosts

type: File

name: test-hostpath

- configMap:

defaultMode: 420

items:

- key: nginx.conf

path: etc/nginx/nginx.conf

name: nginx-conf

name: config-volume

- configMap:

defaultMode: 420

name: nginx-conf

name: config-volume-non-subpath

- emptyDir:

medium: Memory

name: cache-volume

NFS：[https://kubernetes.io/docs/concepts/storage/volumes#nfs](https://kubernetes.io/docs/concepts/storage/volumes" \l "nfs)

<https://github.com/dotbalo/k8s/tree/master/redis/k8s-redis-cluster>

vim /etc/exports

/data/k8s-data 192.168.1.0/24(rw,sync,no\_subtree\_check,no\_root\_squash)

[root@k8s-node02 ~]# vim /etc/exports

[root@k8s-node02 ~]# exportfs -r

[root@k8s-node02 ~]# systemctl enable --now rpcbind nfs-server

Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.

[root@k8s-node02 ~]# systemctl reload nfs-server

强制删除资源：

kubectl delete po demo-nginx-656f55f894-cqt9c demo-nginx-6c5c759fdd-ktfb8 demo-nginx-6c5c759fdd-nsgzm --force --grace-period=0