# **Kubernetes持久化存储**

## **前言**

之前我们有提到数据卷：emptydir ，是本地存储，pod重启，数据就不存在了，需要对数据持久化存储

对于数据持久化存储【pod重启，数据还存在】，有两种方式

* nfs：网络存储【通过一台服务器来存储】

## **步骤**

### **持久化服务器上操作**

* 找一台新的服务器nfs服务端，安装nfs
* 设置挂载路径

使用命令安装nfs

yum install -y nfs-utils

首先创建存放数据的目录

mkdir -p /data/nfs

设置挂载路径

# 打开文件

vim /etc/exports

# 添加如下内容

/data/nfs **\***(rw,no\_root\_squash)

执行完成后，即部署完我们的持久化服务器

### **Node节点上操作**

然后需要在k8s集群node节点上安装nfs，这里需要在 node1 和 node2节点上安装

yum install -y nfs-utils

执行完成后，会自动帮我们挂载上

**启动nfs服务端**

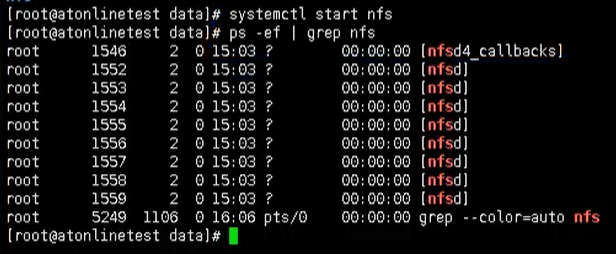
下面我们回到nfs服务端，启动我们的nfs服务

# 启动服务

systemctl start nfs

# 或者使用以下命令进行启动

service nfs-server start



### **K8s集群部署应用**

最后我们在k8s集群上部署应用，使用nfs持久化存储

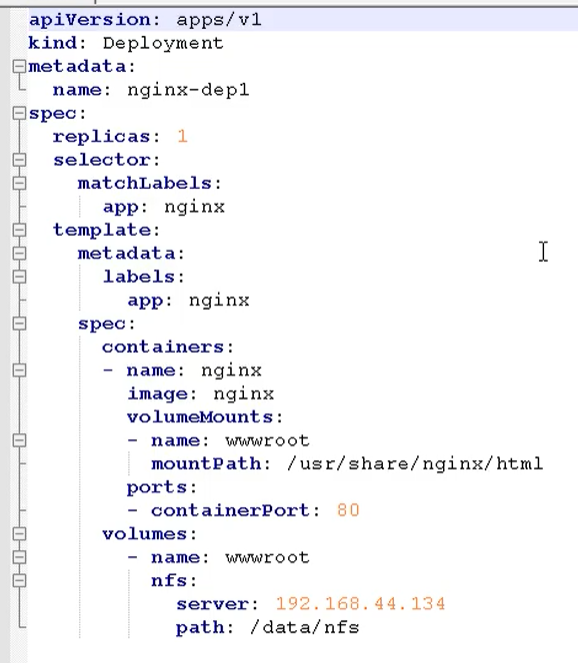
# 创建一个pv文件

mkdir pv

# 进入

cd pv

然后创建一个yaml文件 nfs-nginx.yaml



通过这个方式，就挂载到了刚刚我们的nfs数据节点下的 /data/nfs 目录

最后就变成了： /usr/share/nginx/html -> 192.168.44.134/data/nfs 内容是对应的

我们通过这个 yaml文件，创建一个pod

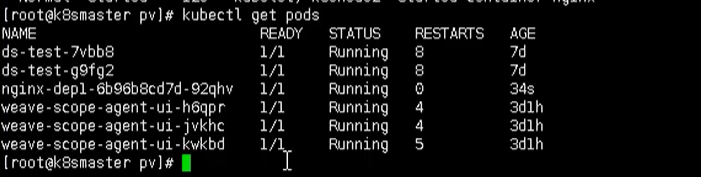
kubectl apply -f nfs-nginx.yaml

创建完成后，我们也可以查看日志

kubectl describe pod nginx-dep1

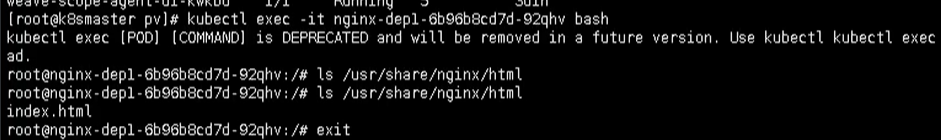


可以看到，我们的pod已经成功创建出来了，同时下图也是出于Running状态



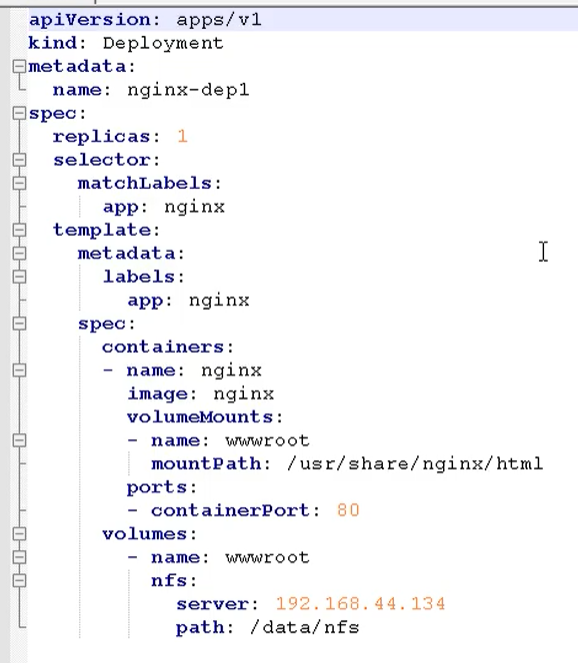
下面我们就可以进行测试了，比如现在nfs服务节点上添加数据，然后在看数据是否存在 pod中

# 进入pod中查看kubectl exec -it nginx-dep1 bash



## **PV和PVC**

对于上述的方式，我们都知道，我们的ip 和端口是直接放在我们的容器上的，这样管理起来可能不方便



所以这里就需要用到 pv 和 pvc的概念了，方便我们配置和管理我们的 ip 地址等元信息

PV：持久化存储，对存储的资源进行抽象，对外提供可以调用的地方【生产者】

PVC：用于调用，不需要关心内部实现细节【消费者】

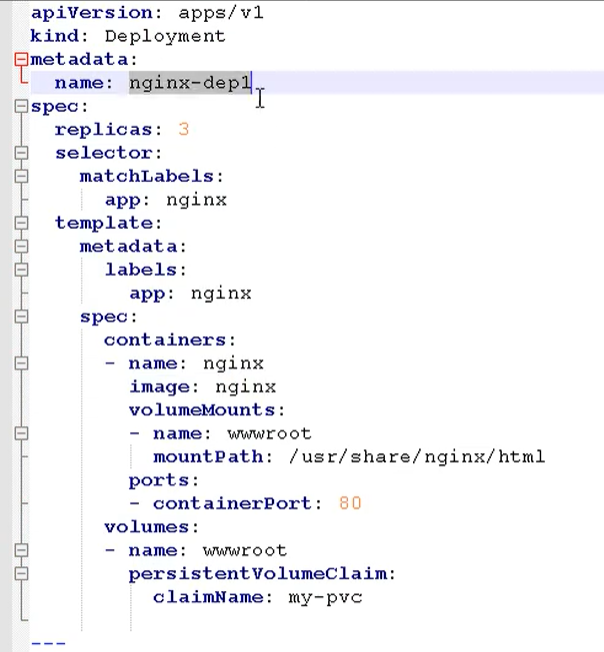
PV 和 PVC 使得 K8S 集群具备了存储的逻辑抽象能力。使得在配置Pod的逻辑里可以忽略对实际后台存储 技术的配置，而把这项配置的工作交给PV的配置者，即集群的管理者。存储的PV和PVC的这种关系，跟 计算的Node和Pod的关系是非常类似的；PV和Node是资源的提供者，根据集群的基础设施变化而变 化，由K8s集群管理员配置；而PVC和Pod是资源的使用者，根据业务服务的需求变化而变化，由K8s集 群的使用者即服务的管理员来配置。

### **实现流程**

* PVC绑定PV
* 定义PVC
* 定义PV【数据卷定义，指定数据存储服务器的ip、路径、容量和匹配模式】

### **举例**

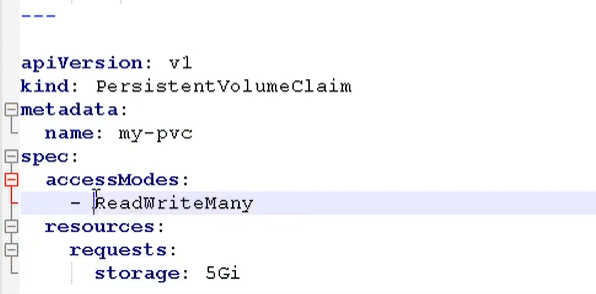
创建一个 pvc.yaml



第一部分是定义一个 deployment，做一个部署

* 副本数：3
* 挂载路径
* 调用：是通过pvc的模式

然后定义pvc



然后在创建一个 pv.yaml