nsd1812 py day06

k8s

环境准备

- node1.tedu.cn 192.168.4.1/24，必须配网关，如果没有配置网关k8s无法启动。
- 将安装包拷贝到node1

安装

```
[root@node1 ~]# cd k8s_pkgs/
[root@node1 k8s_pkgs]# yum remove -y python-rhsm
[root@node1 k8s_pkgs]# yum install -y *
```

配置

```
[root@node1 k8s_pkgs]# vim /etc/sysconfig/docker
OPTIONS='--selinux-enabled=False --log-driver=journald --signature-verification=false'
[root@node1 k8s_pkgs]# vim /etc/docker/daemon.json
{
  "insecure-registries" : ["192.168.4.254:5000"]
}
[root@node1 k8s_pkgs]# vim /etc/kubernetes/apiserver
把ServiceAccount从--admission-control删掉
[root@node1 k8s_pkgs]# vim /etc/kubernetes/kubelet
KUBELET_POD_INFRA_CONTAINER="--pod-infra-container-image=192.168.4.254:5000/pod-infrastructure"
```

启服务

```
# master的组件
[root@node1 k8s_pkgs]# systemctl start etcd
[root@node1 k8s_pkgs]# systemctl enable etcd
[root@node1 k8s_pkgs]# systemctl start kube-apiserver
[root@node1 k8s_pkgs]# systemctl enable kube-apiserver
[root@node1 k8s_pkgs]# systemctl start kube-controller-manager
[root@node1 k8s_pkgs]# systemctl enable kube-controller-manager
[root@node1 k8s_pkgs]# systemctl start kube-scheduler
[root@node1 k8s_pkgs]# systemctl enable kube-scheduler
# node的组件
[root@node1 k8s_pkgs]# systemctl start docker
[root@node1 k8s_pkgs]# systemctl enable docker
[root@node1 k8s_pkgs]# systemctl start kubelet
[root@node1 k8s_pkgs]# systemctl enable kubelet
[root@node1 k8s_pkgs]# systemctl start kube-proxy
```

```
[root@node1 k8s_pkgs]# systemctl enable kube-proxy
```

tomcat+mysql

```
# 创建mysql的rc声明文件
[root@node1 tomcat_mysql]# vim mysql-rc.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: mysql
spec:
  replicas: 1   # 要求标签是app:mysql的pod数目是1
  selector:     # 选择器，查找标签是app:mysql的pod
    app: mysql
  template:     # 如果pod的数目不达标，创建满足以下条件的pod
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        image: 192.168.4.254:5000/mysql
        ports:
        - containerPort: 3306
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: "123456"
# 现在系统中没有rc / pod / 容器 / 镜像
[root@node1 tomcat_mysql]# kubectl get rc
[root@node1 tomcat_mysql]# kubectl get pod
[root@node1 tomcat_mysql]# docker ps
[root@node1 tomcat_mysql]# docker images
# 根据ymal文件创建相关资源
[root@node1 tomcat_mysql]# kubectl create -f mysql-rc.yaml
[root@node1 tomcat_mysql]# kubectl get rc
[root@node1 tomcat_mysql]# kubectl get pod
[root@node1 tomcat_mysql]# docker images
[root@node1 tomcat_mysql]# docker ps
# 查看到pod名称后，获取它的详细信息
[root@node1 tomcat_mysql]# kubectl describe pod mysql-9vj45

# 创建服务
[root@node1 tomcat_mysql]# vim mysql-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  ports:
    - port: 3306
  selector:
    app: mysql
```

```
[root@node1 tomcat_mysql]# kubectl get service
[root@node1 tomcat_mysql]# kubectl create -f mysql-svc.yaml
[root@node1 tomcat_mysql]# kubectl get service
# mysql服务没有EXTERNAL IP，因为用户不直接访问mysql，而是访问web服务

# 创建web服务
[root@node1 tomcat_mysql]# vim myweb-rc.yaml
kind: ReplicationController
metadata:
  name: myweb
spec:
  replicas: 5
  selector:
    app: myweb
  template:
    metadata:
      labels:
        app: myweb
    spec:
      containers:
        - name: myweb
          image: 192.168.4.254:5000/tomcat-app
          ports:
          - containerPort: 8080
          env:
          - name: MYSQL_SERVICE_HOST
            value: 'mysql'
          - name: MYSQL_SERVICE_PORT
            value: '3306'
[root@node1 tomcat_mysql]# kubectl create -f myweb-rc.yaml
[root@node1 tomcat_mysql]# kubectl get rc
[root@node1 tomcat_mysql]# kubectl get pod
[root@node1 tomcat_mysql]# docker ps
# myweb将起动10个容器，因为每个pod中需要有一个架构容器，还需要有一个工作容器
[root@node1 tomcat_mysql]# vim myweb-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: myweb
spec:
  type: NodePort
  ports:
    - port: 8080
      nodePort: 30001
  selector:
    app: myweb
# 访问node的300001，将会映射到pod的8080
[root@node1 tomcat_mysql]# kubectl create -f myweb-svc.yaml
[root@node1 tomcat_mysql]# kubectl get service

# 访问 http://node_ip:30001

# 如果需要动态调整pod数目，只是将rc改个数字即可
```

```
[root@node1 tomcat_mysql]# kubectl scale --replicas=3 replicationcontroller myweb
[root@node1 tomcat_mysql]# kubectl get rc
[root@node1 tomcat_mysql]# kubectl get pod
```

删除

```
[root@node1 tomcat_mysql]# kubectl delete service myweb
[root@node1 tomcat_mysql]# kubectl delete service mysql
[root@node1 tomcat_mysql]# kubectl delete rc myweb
[root@node1 tomcat_mysql]# kubectl delete rc mysql
```

php+redis主从

```
# redis-master
[root@node1 php_redis]# vim redis-master-controller.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: redis-master
  labels:
    name: redis-master
spec:
  replicas: 1
  selector:
    name: redis-master
  template:
    metadata:
      labels:
        name: redis-master
    spec:
      containers:
      - name: master
        image: 192.168.4.254:5000/redis-master
        ports:
        - containerPort: 6379
[root@node1 php_redis]# kubectl create -f redis-master-controller.yaml
[root@node1 php_redis]# vim redis-master-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: redis-master
  labels:
    name: redis-master
spec:
  ports:
  - port: 6379
    targetPort: 6379
  selector:
    name: redis-master
```

```
[root@node1 php_redis]# kubectl create -f redis-master-service.yaml

# redis-slave
[root@node1 php_redis]# vim redis-slave-controller.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: redis-slave
  labels:
    name: redis-slave
spec:
  replicas: 2
  selector:
    name: redis-slave
  template:
    metadata:
      labels:
        name: redis-slave
    spec:
      containers:
      - name: slave
        image: 192.168.4.254:5000/guestbook-redis-slave
        env:
        - name: GET_HOSTS_FORM
          value: env
        ports:
        - containerPort: 6379
[root@node1 php_redis]# kubectl create -f redis-slave-controller.yaml
[root@node1 php_redis]# vim redis-slave-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: redis-slave
  labels:
    name: redis-slave
spec:
  ports:
  - port: 6379
  selector:
    name: redis-slave
[root@node1 php_redis]# kubectl create -f redis-slave-service.yaml

# php
[root@node1 php_redis]# vim frontend-controller.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: frontend
  labels:
    name: frontend
spec:
  replicas: 3
  selector:
    name: frontend
  template:
    metadata:
      labels:
        name: frontend
    spec:
      containers:
      - name: frontend
        image: 192.168.4.254:5000/guestbook-php-frontend
        env:
        - name: GET_HOSTS_FROM
          value: env
        ports:
          - containerPort: 80
[root@node1 php_redis]# kubectl create -f frontend-controller.yaml
[root@node1 php_redis]# vim frontend-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend
  labels:
    name: frontend
spec:
  type: NodePort
  ports:
  - port: 80
    nodePort: 30001
  selector:
    name: frontend
[root@node1 php_redis]# kubectl create -f frontend-service.yaml
# 访问宿主机http://宿主机IP:30001
```