实验课程6-6 使用私有镜像部署应用

我们在前面的实践课程中在创建和部署deployment资源时， 使用的容器镜像是公共的， 这样是不安全的， 本节课程我们来学习创建secret资源，并拉取私有镜像来部署应用。

1、在容器镜像仓库服务中设置java-demo仓库为私有仓库

2、部署deployment-v2.yaml和service-v2.yaml

```
$ cat deployment-v2.yaml
  apiVersion: extensions/v1beta1  # API版本
  kind: Deployment       # 资源类型， Deployment
  metadata:       # 元数据
    labels:       # 元数据， 标签列表
      name: java-demo-v2  # 元数据， deployment的标签名称
    name: java-demo-v2    # 元数据， deployment的名字
    namespace: student-001 # 元数据，deployment的命名空间
  spec:                    # deployment中容器模板的详细定义
    replicas: 2            # pod的副本数
    template:
      metadata:
        labels:
          app: java-demo-v2
      spec:
        containers:
        \- name: java-demo-v2 # 容器名称
          image: registry.cn-qingdao.aliyuncs.com/devops-demo/java-demo:latest # 容器使用的镜
像
          imagePullPolicy: Always # 获取镜像的策略
          ports:
            \- containerPort: 8080 # 容器要暴露的端口
```

部署和查看deployment资源：

```
$ kubectl create -f deployment-v2.yaml
 deployment "java-demo-v2" created
```

```
 $ kubectl get deployments -n student-001
 NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
 java-demo-deployment   4         4         4            4           18h
 java-demo-v1           2         2         2            2           16m
 java-demo-v2           2         2         2            0           29s
```

会发现deployment java-demo-v2的AVAILABLE数量为0， 查看pod资源：

```
$ kubectl get pod -n student-001
 NAME                                  READY     STATUS         RESTARTS   AGE
 java-demo-deployment-864d7458c7-5x75x   1/1       Running        0          3h
 java-demo-deployment-864d7458c7-6tk2n   1/1       Running        0          18h
 java-demo-deployment-864d7458c7-gqhbs   1/1       Running        0          3h
 java-demo-deployment-864d7458c7-rlktm   1/1       Running        0          3h
 java-demo-pod                         1/1       Running        0          18h
 java-demo-v1-5649fcc445-65mb2         1/1       Running        0          9m
 java-demo-v1-5649fcc445-tf9f8         1/1       Running        0          9m
 java-demo-v2-6c66bd6c89-4kql4         0/1       ErrImagePull   0          1m
 java-demo-v2-6c66bd6c89-dmtk5         0/1       ErrImagePull   0          1m
```

java-demo-v2应用的2个pod都有ErrImagePull的报错， 这就是因为我们把镜像仓库设置成私有的，而拉取私有镜像时需要使用凭证的。

3、创建拉取私有镜像的凭证secret

```
$ kubectl create secret docker-registry regsecret --docker-server=registry.cn-
qingdao.aliyuncs.com --docker-username=xxxxxx --docker-password=xxxxxx --docker-
email=xxxxxx -n student-001
  secret "regsecret" created
```

查看secret列表：

```
 $ kubectl get secret -n student-001
 NAME                  TYPE                                  DATA      AGE
 default-token-rs7qz   kubernetes.io/service-account-token   3         20h
 regsecret             kubernetes.io/dockerconfigjson        1         13s
```

4、修改deployment-v2.yaml使用regsecret

```
 $ cat deployment-v2.yaml
 apiVersion: extensions/v1beta1  # API版本
 kind: Deployment       # 资源类型， Deployment
 metadata:       # 元数据
```

```
    labels:       # 元数据， 标签列表
      name: java-demo-v2  # 元数据， deployment的标签名称
    name: java-demo-v2    # 元数据， deployment的名字
    namespace: student-001 # 元数据，deployment的命名空间
  spec:                    # deployment中容器模板的详细定义
    replicas: 2            # pod的副本数
    template:
      metadata:
        labels:
          app: java-demo-v2
     spec:
        containers:
        \- name: java-demo-v2 # 容器名称
          image: registry.cn-qingdao.aliyuncs.com/devops-demo/java-demo:latest # 容器使用的镜
像
          imagePullPolicy: Always # 获取镜像的策略
          ports:
            \- containerPort: 8080 # 容器要暴露的端口
        imagePullSecrets:
        \- name: regsecret
```

更新deployment：

```
  $ kubectl apply -f deployment-v2.yaml
  deployment "java-demo-v2" configured
```

查看deployment资源：

```
$ kubectl get deployments -n student-001
 NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
 java-demo-deployment   4         4         4            4           18h
 java-demo-v1           2         2         2            2           34m
 java-demo-v2           2         2         2            2           18m
```

5、部署service-v2.yaml

```
 $ cat service-v2.yaml
 apiVersion: v1  # API版本
 kind: Service       # 资源类型， Service
 metadata:       # 元数据
   labels:       # 元数据， 标签列表
     name: java-demo-v2  # 元数据， service的标签名称
   name: java-demo-v2    # 元数据， service的名字
   namespace: student-001 # 元数据，service的命名空间
 spec:
   ports:
   \- port: 80            # 提供给容器内部应用访问的端口号
     targetPort: 8080    # pod上应用监听的端口
     name: java-demo-v2
   selector:
```

```
    app: java-demo-v2 # 应用选择
  type: NodePort        # 向外部用户暴露端口的方式
```

查看service资源：

```
$ kubectl get service -n student-001
  NAME                TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
  java-demo-service   NodePort   172.19.10.98    <none>        80:31536/TCP   4h
  java-demo-v1        NodePort   172.19.10.47    <none>        80:31447/TCP   33m
  java-demo-v2        NodePort   172.19.11.214   <none>        80:30605/TCP   3s
```

6、访问服务