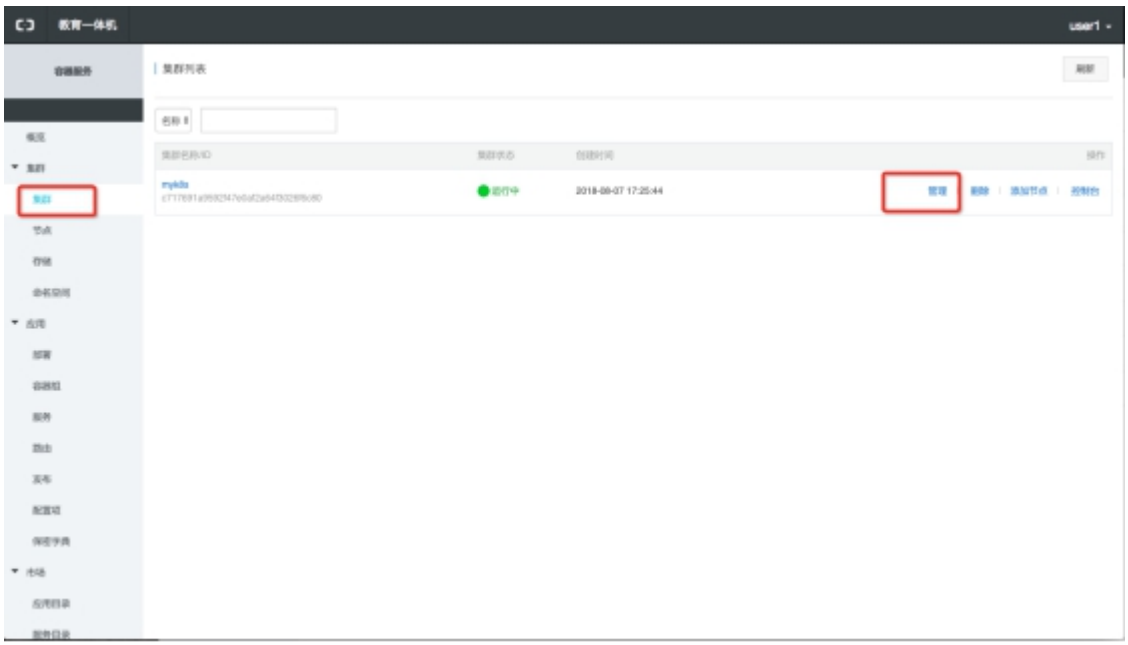
实验课程6-5 使用kubectl创建和部署应用

我们在前面的实践课程中分别学习了如何通过氢气服务的界面部署pod、deployment和service资源，实际上还有一个更强大的客户端工具kubectl可以帮助我们来做这些事情。

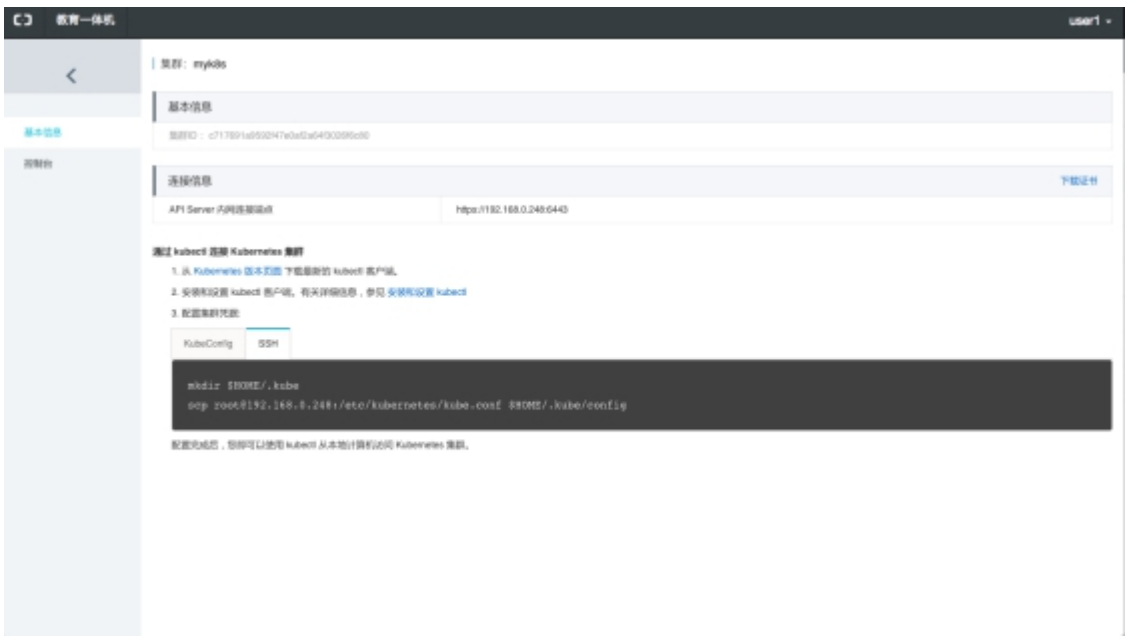
1、通过kubectl连接容器集群

依次点击 集群-> 管理 查看集群信息：

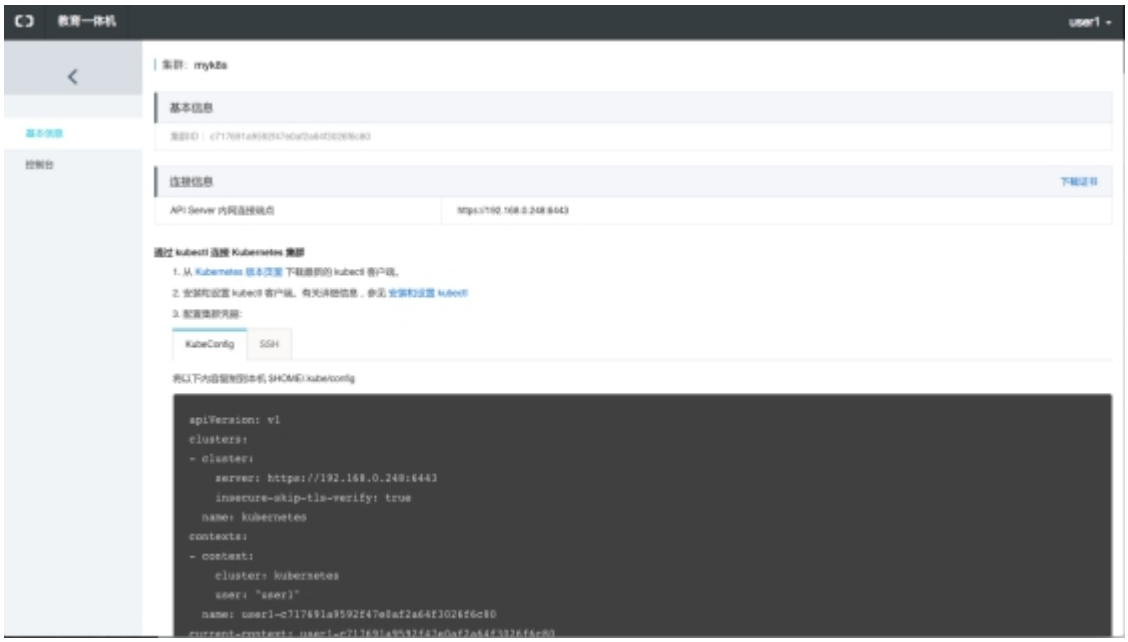


通过ssh拷贝连接集群所需的配置文件到本地， 本示例的命令为：

```
mkdir $HOME/.kube
scp root@192.168.0.248:/etc/kubernetes/kube.conf $HOME/.kube/config
```



或者在本地创建~/.kube/config文件并复制粘贴KubeConfig的内容：



2、安装kubectl客户端到本地

```
scp root@192.168.0.248:/usr/bin/kubectl /usr/bin/kubectl
```

验证是否可以连接集群：

```
  $ kubectl get all -n student-001
NAME                           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deploy/java-demo-deployment    4         4         4            4           17h

NAME                                    DESIRED   CURRENT   READY     AGE
rs/java-demo-deployment-864d7458c7      4         4         4         17h

NAME                           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deploy/java-demo-deployment    4         4         4            4           17h

NAME                                    DESIRED   CURRENT   READY     AGE
rs/java-demo-deployment-864d7458c7      4         4         4         17h

NAME                                          READY     STATUS    RESTARTS   AGE
po/java-demo-deployment-864d7458c7-5x75x      1/1       Running   0          3h
po/java-demo-deployment-864d7458c7-6tk2n      1/1       Running   0          17h
po/java-demo-deployment-864d7458c7-gqhbs      1/1       Running   0          3h
po/java-demo-deployment-864d7458c7-rlktm      1/1       Running   0          3h
po/java-demo-pod                              1/1       Running   0          17h

NAME                     TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
svc/java-demo-service    NodePort   172.19.10.98   <none>        80:31536/TCP   3h
```

这条命令列出了当前命名空间student-001下的所有资源。

3、修改并使用kubectl部署deployment-v1.yaml

查看当前deployment资源：

```
  $ kubectl get deployments -n student-001
  NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
  java-demo-deployment   4         4         4            4           17h

  $ cat deployment-v1.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-v1  # 元数据， deployment的标签名称
    name: java-demo-v1    # 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-v1
      spec:
        containers:
        \- name: java-demo-v1 # 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：

```
  $ kubectl create -f deployment-v1.yaml
  deployment "java-demo-v1" created

  $ kubectl get deployments -n student-001
  NAME                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
  java-demo-deployment   4         4         4            4           17h
  java-demo-v1           2         2         2            0           1m
```

4、修改并使用kubectl部署service-v1.yaml

查看当前service资源：

```
  $ kubectl get services -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

$ cat service-v1.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-v1  # 元数据， service的标签名称
  name: java-demo-v1    # 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-v1
  selector:
    app: java-demo-v1 # 应用选择
  type: NodePort        # 向外部用户暴露端口的方式
```

部署和查看：

```
$ kubectl create -f service-v1.yaml
service "java-demo-v1" created
```

```
$ kubectl get services -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   58s
```

5、访问java-demo-v1服务

Hello CodePipeline!