测试环境信息：

|  |  |
| --- | --- |
| 节点类型 | IP |
| Master Node | 197.0.64.148-150 |
| Worker Node | 197.0.64.146-147 |

相关yaml文件存放在197.0.64.148 /root/sunxi目录下

Chapter 3 Pod

1. 已部署pod的完整YAML

JCFWK8S1:~/sunxi # kubectl get po

NAME READY STATUS RESTARTS AGE

centos-0 1/1 Running 0 21h

centos-1 1/1 Running 0 21h

JCFWK8S1:~/sunxi # kubectl get po centos-1 -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: 2020-04-21T08:59:10Z

generateName: centos-

labels:

app: centos

controller-revision-hash: centos-885cd8fd8

statefulset.kubernetes.io/pod-name: centos-1

name: centos-1

namespace: default

ownerReferences:

- apiVersion: apps/v1

blockOwnerDeletion: true

controller: true

kind: StatefulSet

name: centos

uid: 5c39503a-83ae-11ea-b771-005056aac524

resourceVersion: "14725460"

selfLink: /api/v1/namespaces/default/pods/centos-1

uid: 5d9a0368-83ae-11ea-8193-005056aaf4b6

spec:

containers:

- command:

- python

- -m

- SimpleHTTPServer

- "8080"

env:

- name: MONITOR\_TYPE

value: CONTAINER\_ONLY

- name: cmbc\_logs\_centos

value: stdout

- name: cmbc\_logs\_centos

value: topic=dockerpf-audit

- name: MY\_HOST\_IP

valueFrom:

fieldRef:

apiVersion: v1

fieldPath: status.hostIP

image: 197.0.208.87/public/centos7.4:20191126001

imagePullPolicy: IfNotPresent

name: centos

ports:

- containerPort: 8080

protocol: TCP

resources:

limits:

cpu: 500m

memory: 500Mi

requests:

cpu: 250m

memory: 64Mi

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

volumeMounts:

- mountPath: /var/run/secrets/kubernetes.io/serviceaccount

name: default-token-dsdbc

readOnly: true

dnsPolicy: ClusterFirst

hostname: centos-1

nodeName: jcfwk8s5

nodeSelector:

beta.kubernetes.io/arch: amd64

priority: 0

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

serviceAccount: default

serviceAccountName: default

subdomain: centos

terminationGracePeriodSeconds: 30

tolerations:

- effect: NoExecute

key: node.kubernetes.io/not-ready

operator: Exists

tolerationSeconds: 300

- effect: NoExecute

key: node.kubernetes.io/unreachable

operator: Exists

tolerationSeconds: 300

volumes:

- name: default-token-dsdbc

secret:

defaultMode: 420

secretName: default-token-dsdbc

status:

conditions:

- lastProbeTime: null

lastTransitionTime: 2020-04-21T08:59:10Z

status: "True"

type: Initialized

- lastProbeTime: null

lastTransitionTime: 2020-04-21T08:59:11Z

status: "True"

type: Ready

- lastProbeTime: null

lastTransitionTime: 2020-04-21T08:59:11Z

status: "True"

type: ContainersReady

- lastProbeTime: null

lastTransitionTime: 2020-04-21T08:59:10Z

status: "True"

type: PodScheduled

containerStatuses:

- containerID: docker://ee57d9b99671d70a31bb5167d7a9fb7e92e28dbe083834149e0138291aa07047

image: 197.0.208.87/public/centos7.4:20191126001

imageID: docker-pullable://197.0.208.87/public/centos7.4@sha256:8c245f514a97baf84ea7be813f4afcba23433995997e8aba285478134400025e

lastState: {}

name: centos

ready: true

restartCount: 0

state:

running:

startedAt: 2020-04-21T08:59:11Z

hostIP: 197.0.64.146

phase: Running

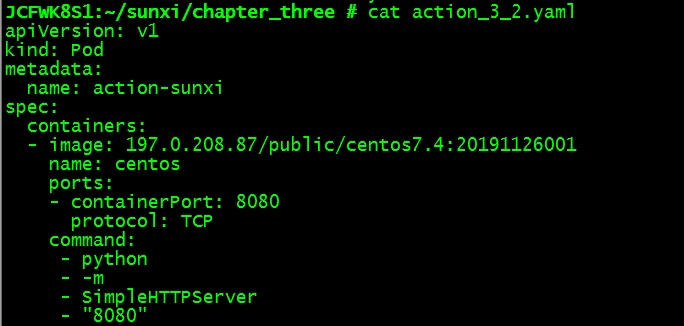
podIP: 197.28.67.18

qosClass: Burstable

startTime: 2020-04-21T08:59:10Z

1. 为pod创建一个简单的YAML描述文件

文件存储地址：197.0.64.148 /root/sunxi/chapter\_three/action\_3\_2.yaml



1. 使用kubectl create创建pod

JCFWK8S1:~/sunxi/chapter\_three # kubectl create -f action\_3\_2.yaml

pod/action-sunxi created

4.查看pod完整的描述文件：

kubectl get po action-sunxi -o yaml

kubectl get po action-sunxi -o json

5.查看pod运行状态：

JCFWK8S1:~/sunxi # kubectl get pods

NAME READY STATUS RESTARTS AGE

action-sunxi 1/1 Running 0 49m

1. 查看日志

JCFWK8S1:~/sunxi # kubectl logs action-sunxi

197.0.64.148 - - [22/Apr/2020 14:55:08] "GET / HTTP/1.1" 200 -

197.0.64.148 - - [22/Apr/2020 15:01:13] "GET / HTTP/1.1" 200 -

1. 将本地网络端口转发到pod中的8080端口

JCFWK8S1:~/sunxi # kubectl port-forward action-sunxi 8888:8080

Forwarding from 127.0.0.1:8888 -> 8080

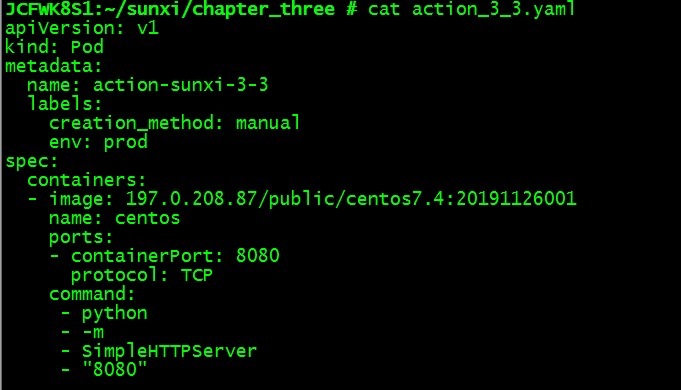
Forwarding from [::1]:8888 -> 8080

Handling connection for 8888

在另一个窗口执行以下命令：

JCFWK8S1:~ # curl localhost:8888

1. 为pod添加标签



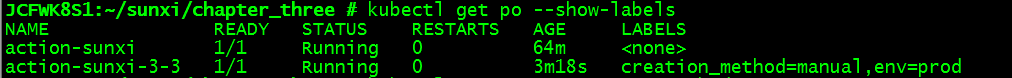
创建带标签的pod:

JCFWK8S1:~/sunxi/chapter\_three # kubectl create -f action\_3\_3.yaml

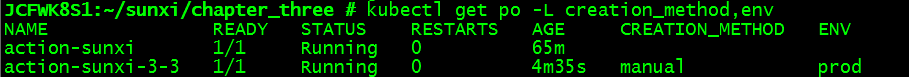
pod/action-sunxi-3-3 created

9.查看pod所带的标签：

kubectl get po --show-labels

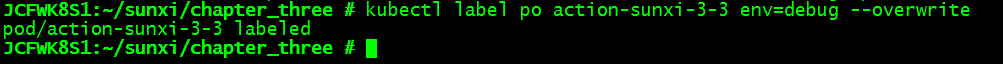


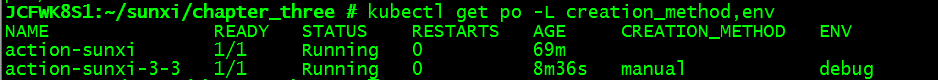
kubectl get po -L creation\_method,env



1. 修改现有pod的标签

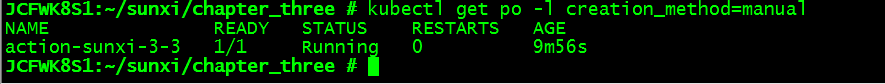
kubectl label po action-sunxi-3-3 env=debug --overwrite





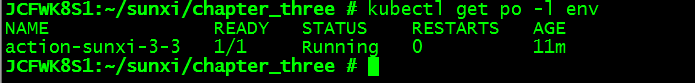
1. 使用标签选择器列出pod

kubectl get po -l creation\_method=manual



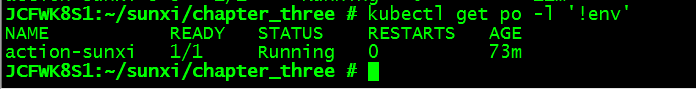
列出包含env标签的所有pod:

kubectl get po -l env



列出没有包含env标签的所有pod:

kubectl get po -l '!env'



1. 将节点通过标签选择器调度到特定节点

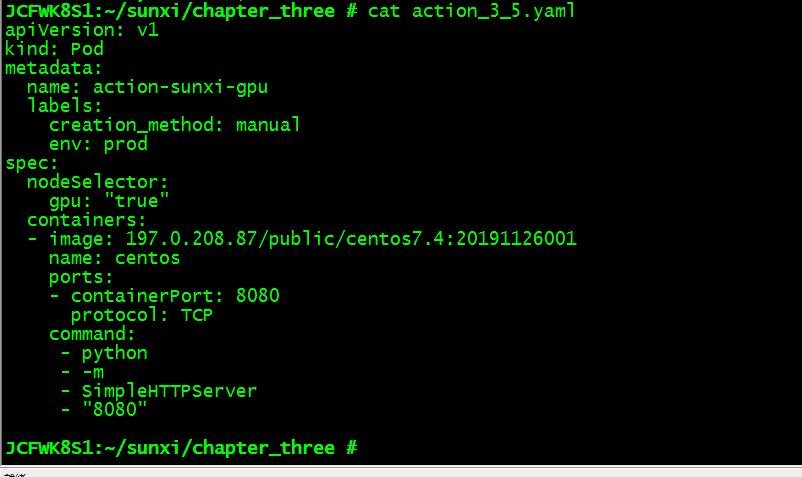
kubectl get nodes --show-labels



kubectl label node jcfwk8s5 gpu=true

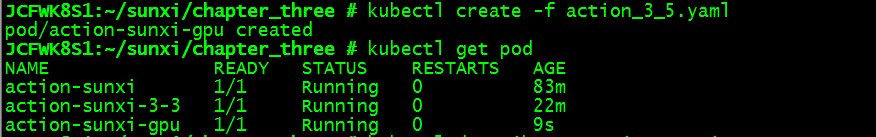


修改yaml文件，添加节点选择器

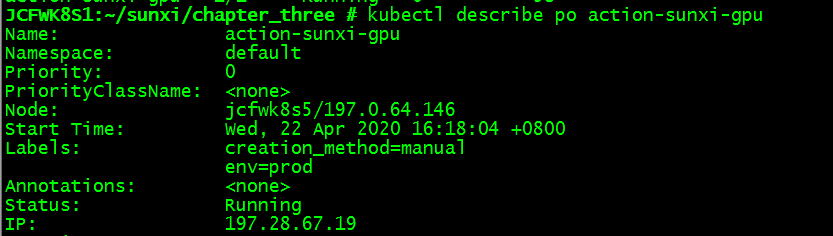


kubectl create -f action\_3\_5.yaml

kubectl get pod



kubectl describe po action-sunxi-gpu

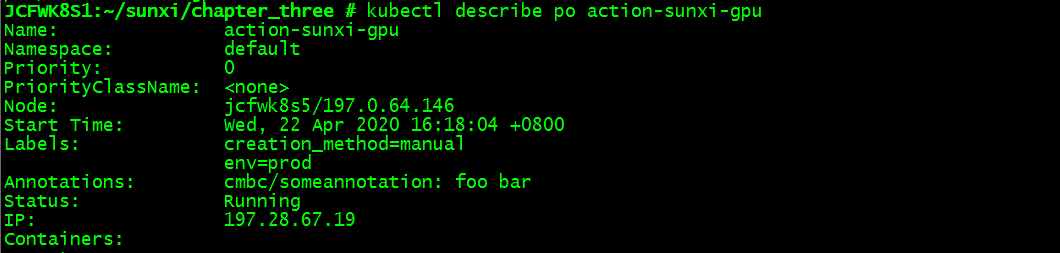


1. 添加注解

kubectl annotate pod action-sunxi-gpu cmbc/someannotation="foo bar"



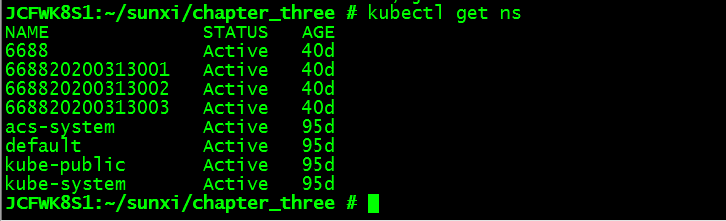
kubectl describe po action-sunxi-gpu



1. 命名空间

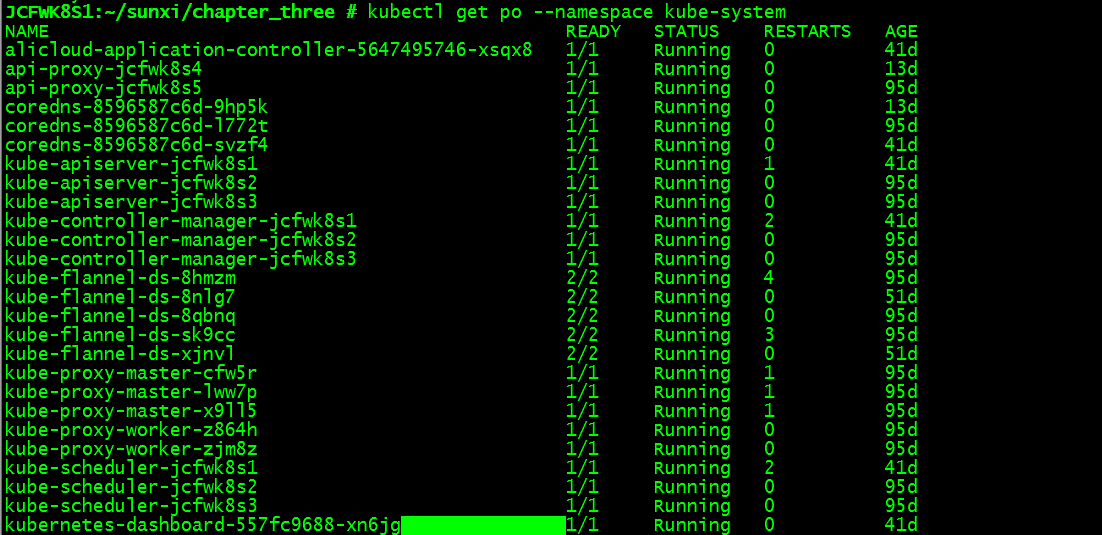
列出集群所有命名空间：

kubectl get ns



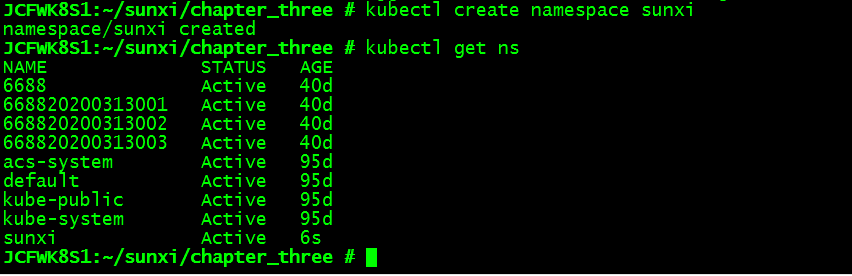
列出指定空间中包含的pod:

kubectl get po --namespace kube-system



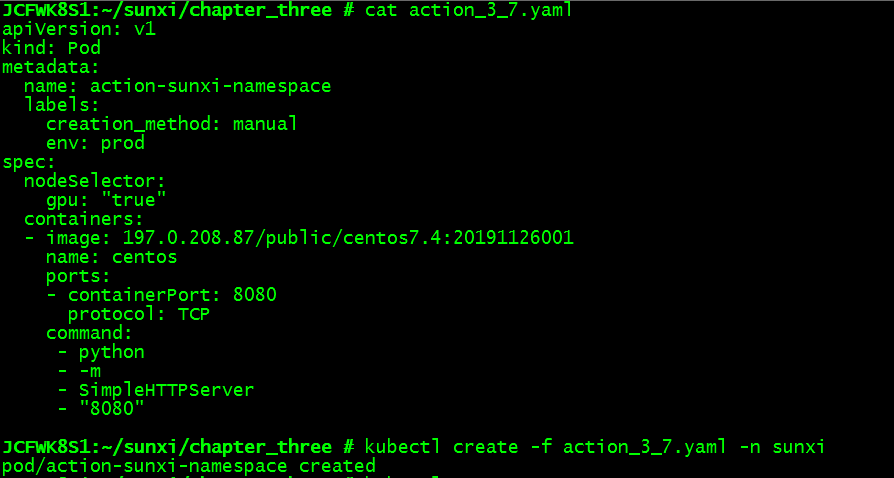
创建命名空间：

kubectl create namespace sunxi

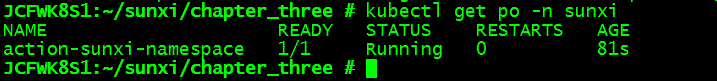


在指定命名空间中创建pod：

kubectl create -f action\_3\_7.yaml -n sunxi



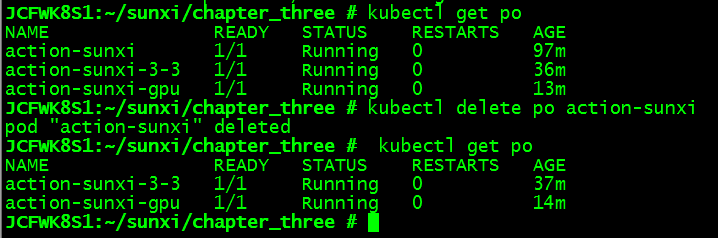
kubectl get po -n sunxi



1. 删除pod

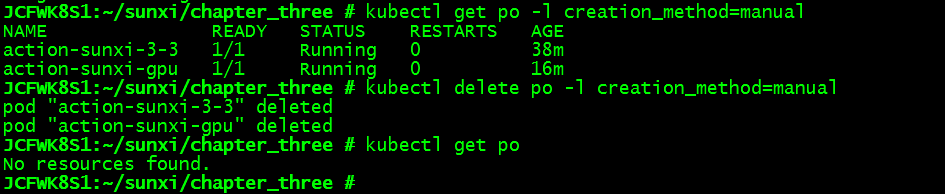
按照pod名称删除：

kubectl delete po action-sunxi



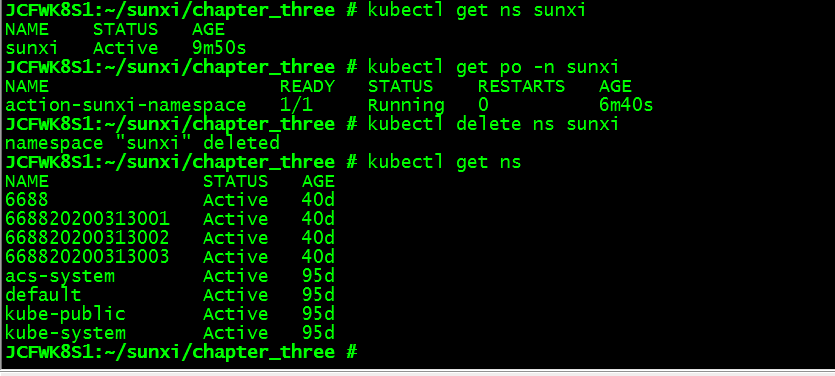
使用标签选择器删除：

kubectl delete po -l creation\_method=manual



通过删除命名空间删除pod:

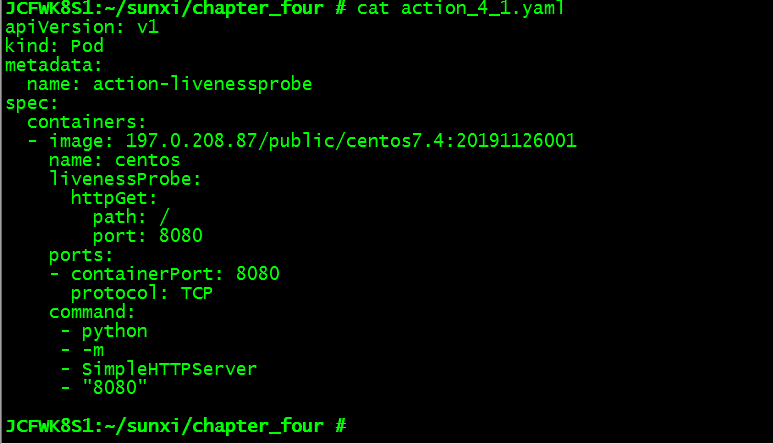
kubectl delete ns sunxi

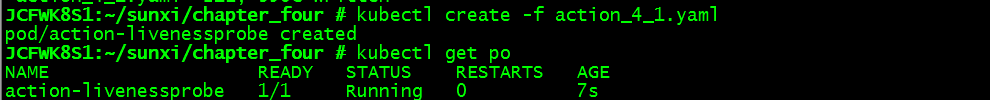


Chapter 4 ReplicationController、ReplicaSet、DaemonSet、Job、CronJob

1. 探针

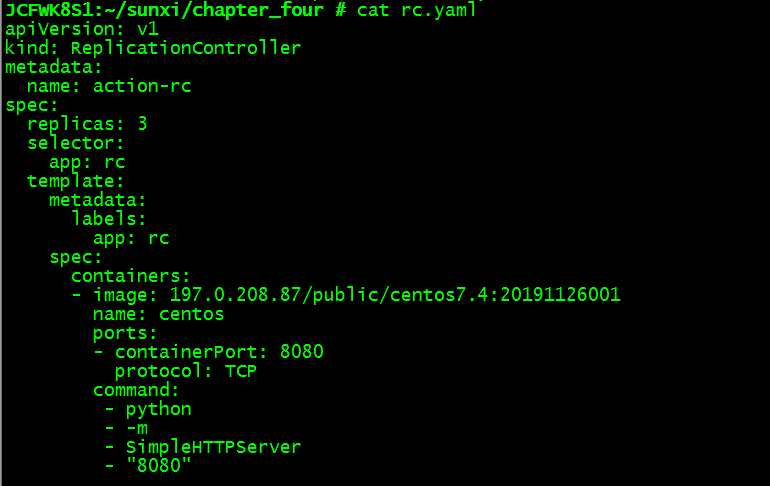
* 为pod添加存活探针



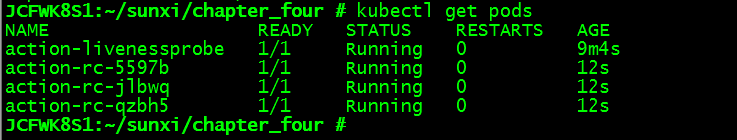


1. ReplicationController

* 创建

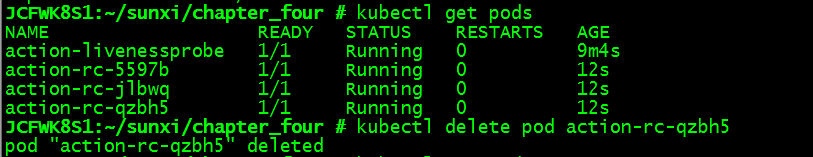


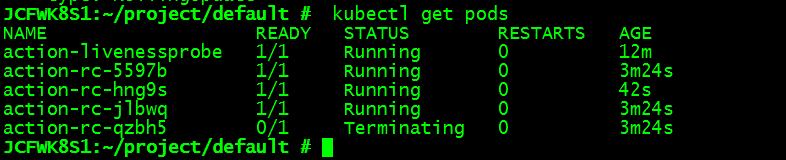
kubectl create -f rc.yaml

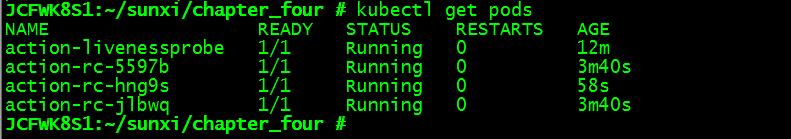


* 删除rc中的一个pod

kubectl delete pod action-rc-qzbh5

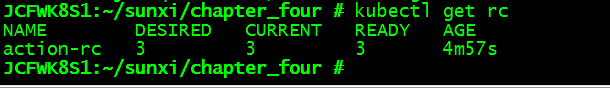






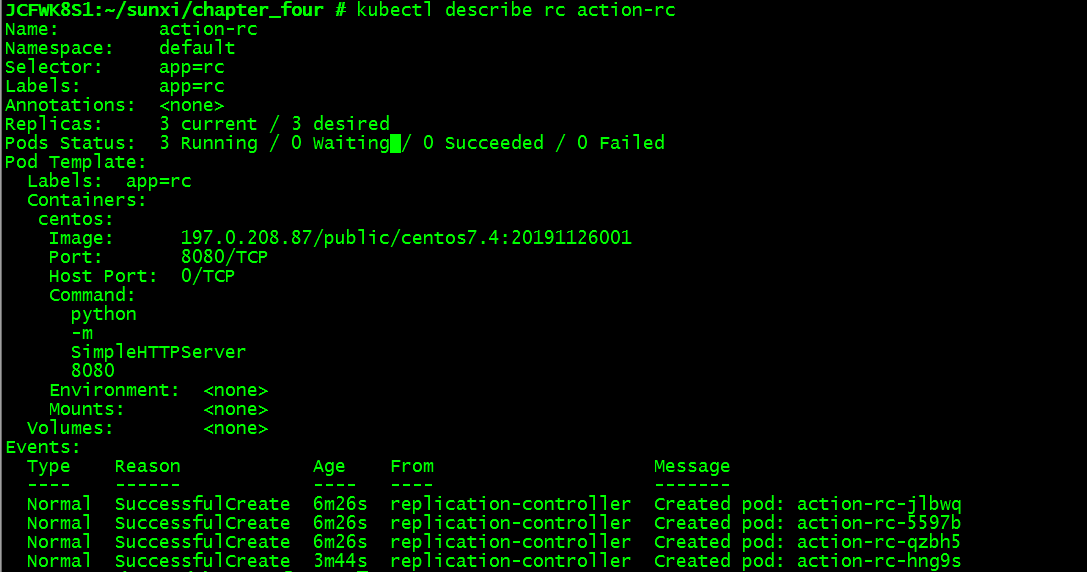
* 获取rc相关信息

kubectl get rc



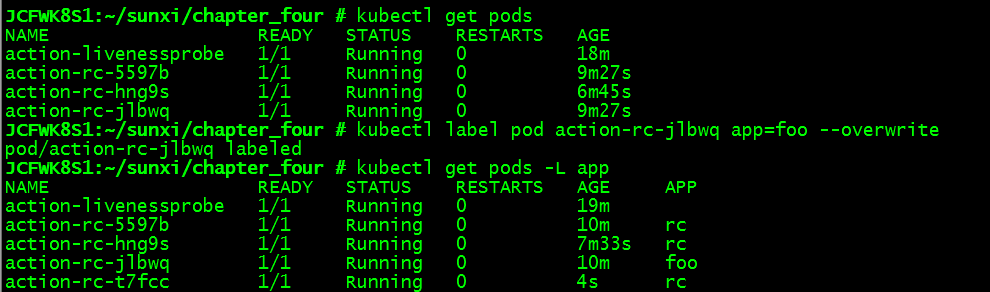
* 查看rc的附加信息

kubectl describe rc action-rc



* 将pod移入或移除rc的作用域

kubectl label pod action-rc-jlbwq app=foo --overwrite



* 修改pod模版

kubectl edit rc action-rc

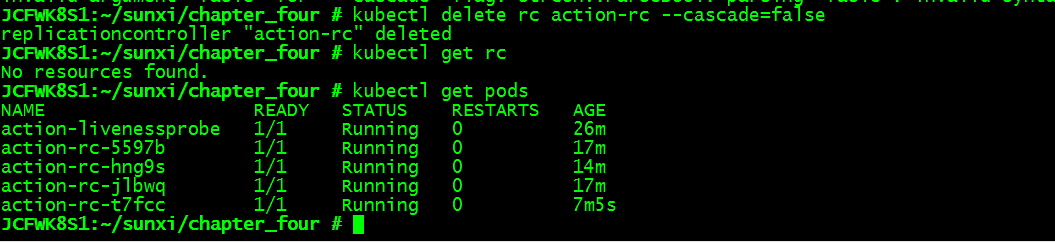
* 给rc扩容/缩容

kubectl scale rc action-rc --replicas=10

使用kubectl edit rc action-rc修改spec中的replicas

* 删除rc

kubectl delete rc action-rc --cascade=false（删除rc后pod并没有删除）

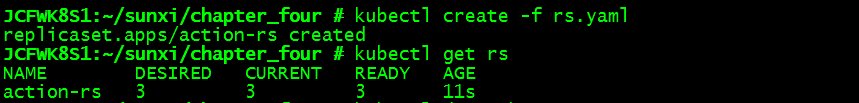


1. ReplicaSet

* 创建

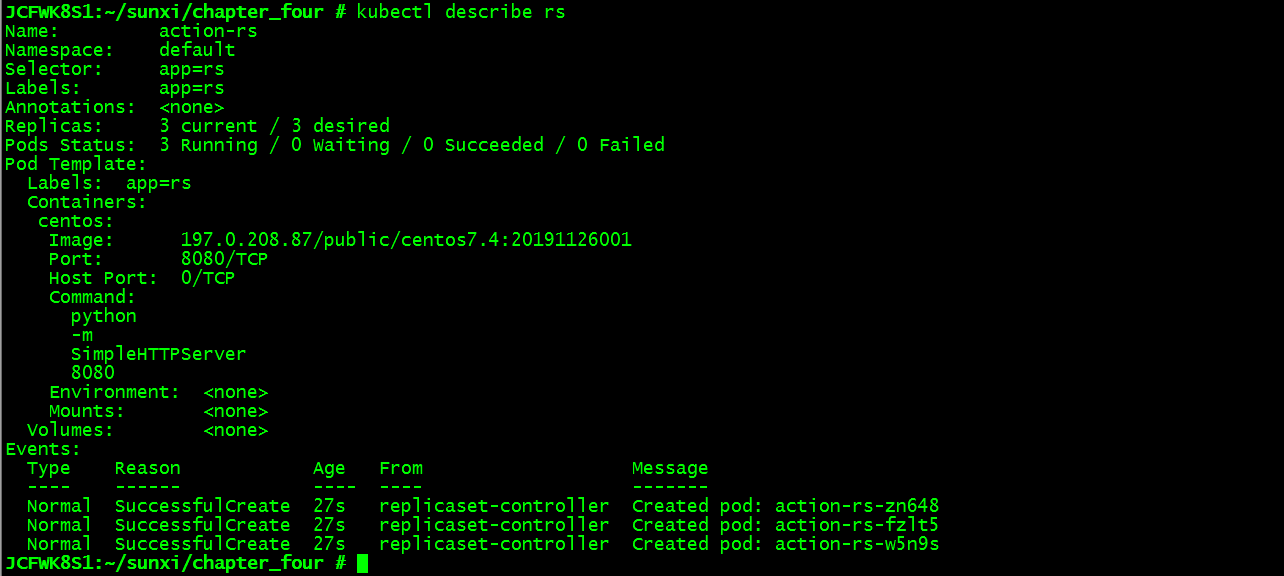


kubectl create -f rs.yaml



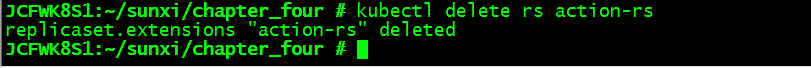
kubectl get rs

kubectl describe rs



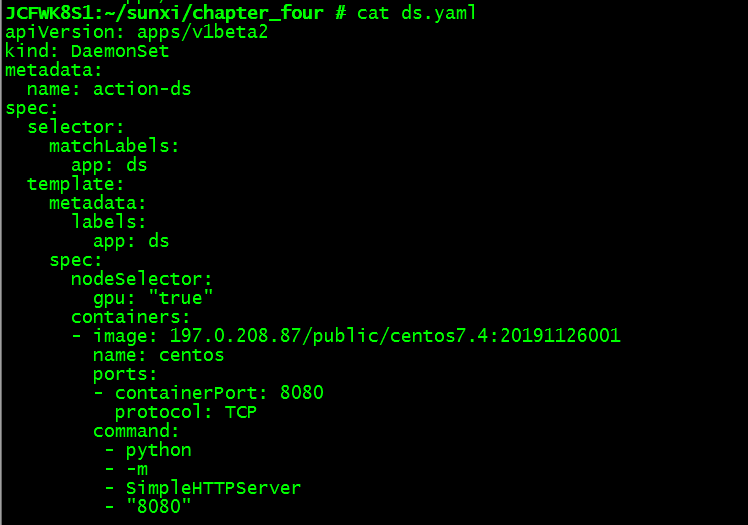
* 删除rs

kubectl delete rs action-rs



1. DaemonSet

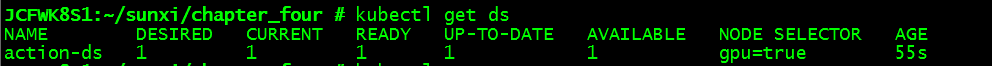
* 创建ds只在特定节点上运行

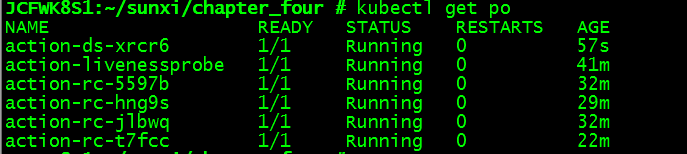


kubectl create -f ds.yaml



kubectl get ds





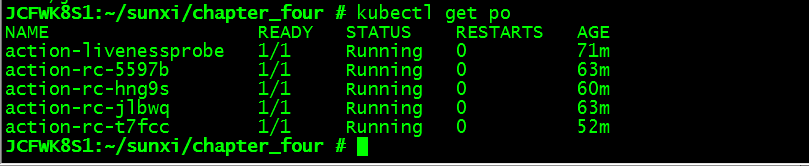
* 从节点上删除所需的标签，pod被终止了

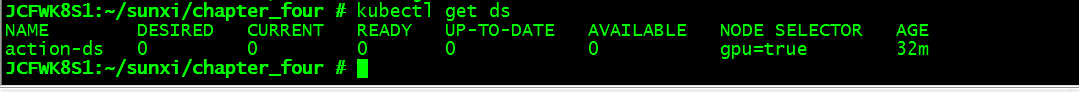
kubectl label node jcfwk8s5 gpu=false --overwrite

kubectl get nodes --show-labels



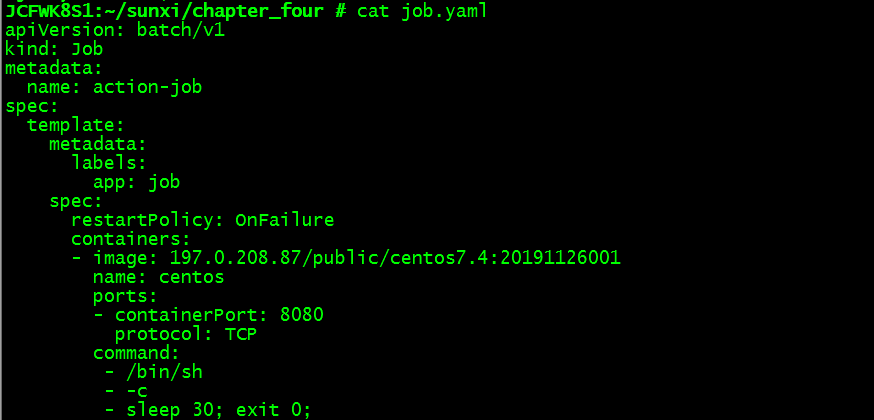
kubectl get po

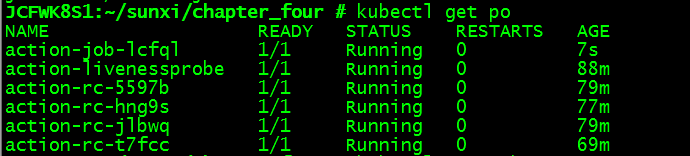


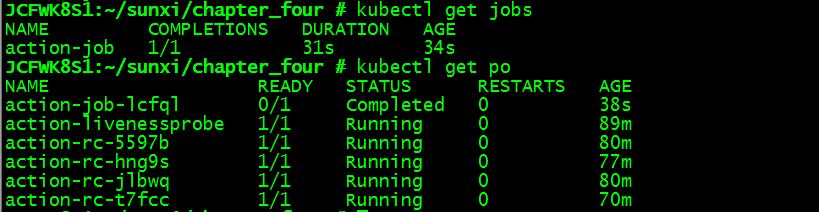


1. Job

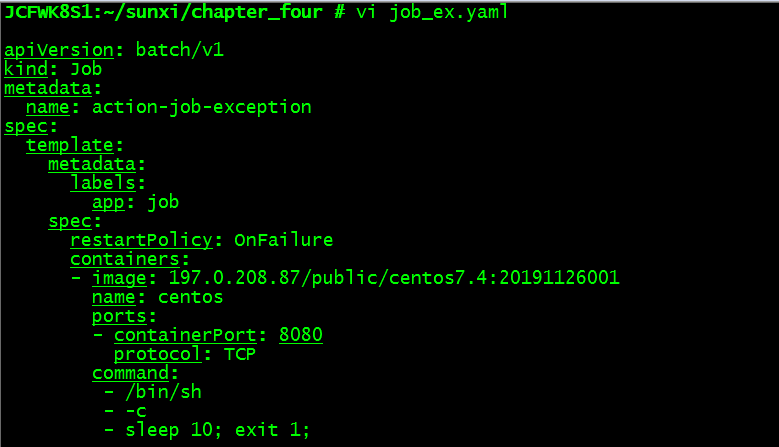
* 创建（job正常运行，如果需要模拟job异常，可以将exit 0改为1）

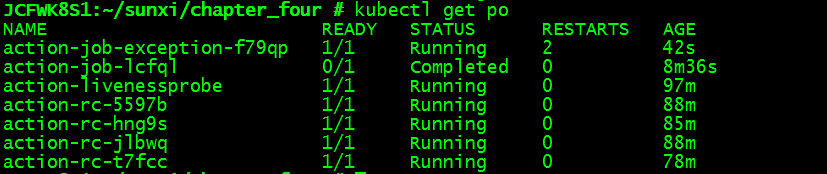
 kubectl create -f job.yaml





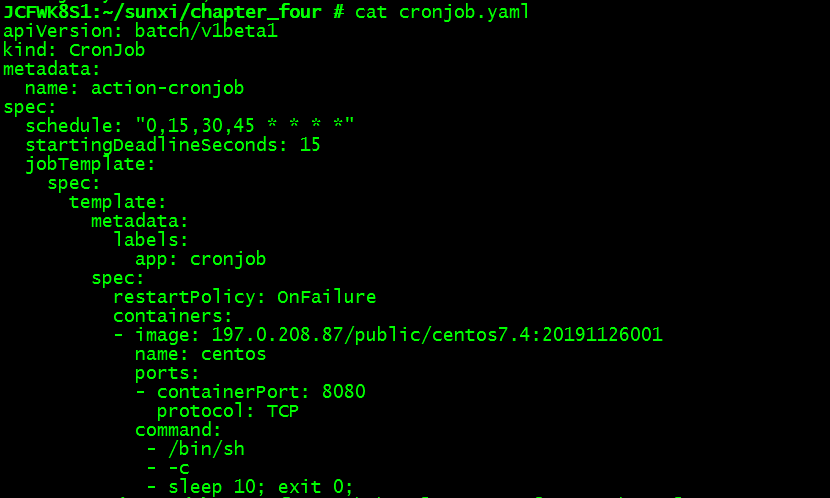
* 创建job，其运行的任务异常





6.CronJob

* 创建



kubectl create -f cronjob.yaml



kubectl get cronjob

