

## CKAD

### Certified Kubernetes Application Developer (CKAD) Program

Web: [www.dumpscollection.com](http://www.dumpscollection.com)

Email: [support@.dumpscollection.com](mailto:support@.dumpscollection.com)

Version: Demo

[ Total Questions: 10 ]



# **IMPORTANT NOTICE**

## **Feedback**

We have developed quality product and state-of-art service to ensure our customers interest. If you have any suggestions, please feel free to contact us at [feedback@dumpscopy.com](mailto:feedback@dumpscopy.com)

## **Support**

If you have any questions about our product, please provide the following items:

- ⦿ exam code
- ⦿ screenshot of the question
- ⦿ login id/email

please contact us at [support@dumpscopy.com](mailto:support@dumpscopy.com) and our technical experts will provide support within 24 hours.

## **Copyright**

The product of each order has its own encryption code, so you should use it independently. Any unauthorized changes will inflict legal punishment. We reserve the right of final explanation for this statement.

**Question #1**

You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

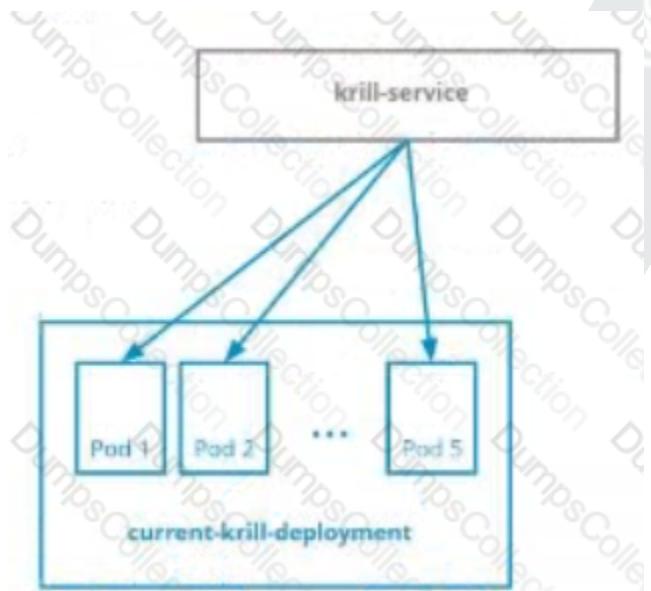
```
[candidate@node-1] $ kubectl config use-context sk8s
```

**Context**

You are asked to prepare a Canary deployment for testing a new application release.

**Task:**

A Service named krill-Service in the goshark namespace points to 5 pod created by the Deployment named current-krill-deployment

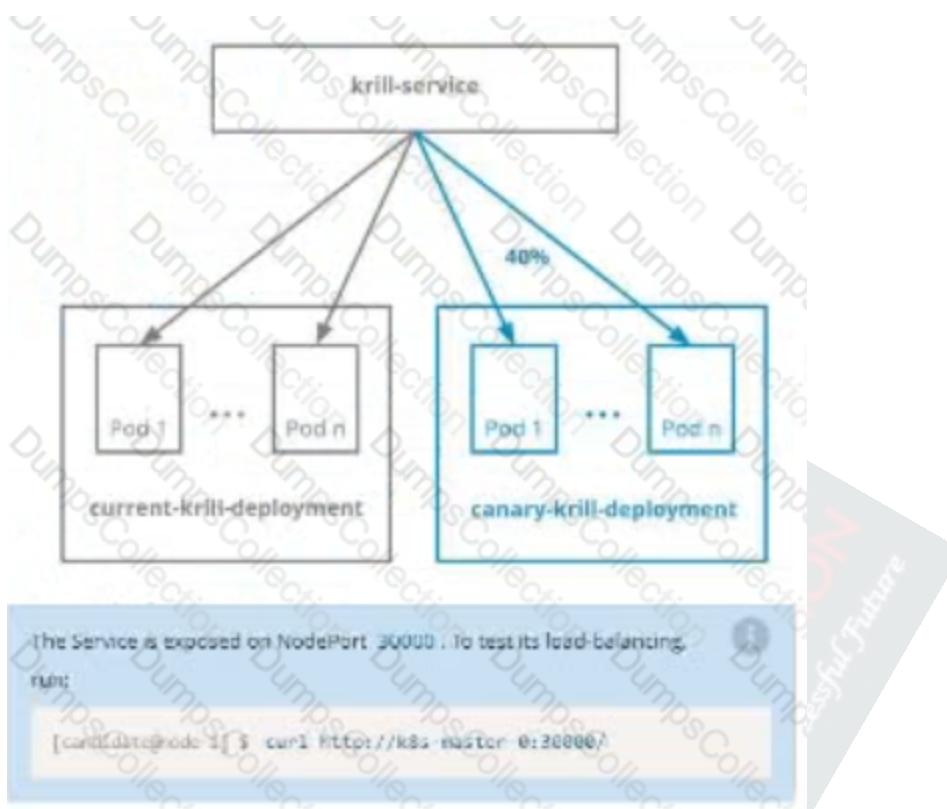


1) Create an identical Deployment named canary-kill-deployment, in the same namespace.

2) Modify the Deployment so that:

- A maximum number of 10 pods run in the goshawk namespace.

- 40% of the krill-service 's traffic goes to the canary-krill-deployment pod(s)



See the solution below.

## Explanation

Solution:

```
candidate@node-1:~/humane-storks$ kubectl scale deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-storks$ kubectl get deploy -n goshawk
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
canary-krill-deployment   4/4     4          4          46s
current-krill-deployment 5/5     5          5          7h22m
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
--2022-09-24 11:43:51-- https://k8s.io/examples/admin/resource/quota-pod.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::.
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://kubernetes.io/examples/admin/resource/quota-pod.yaml [following]
--2022-09-24 11:43:52-- https://kubernetes.io/examples/admin/resource/quota-pod.yaml
Resolving kubernetes.io (kubernetes.io)... 147.75.40.148
Connecting to kubernetes.io (kubernetes.io)|147.75.40.148|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90 [application/x-yaml]
Saving to: 'quota-pod.yaml'

quota-pod.yaml          100%[=====] 90  --.-KB/s   in 0s

2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml
```

```
[File Edit View Terminal Tabs Help]

2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml
candidate@node-1:~/humane-storks$ kubectl create -f quota-pod.yaml
resourcequota/pod-demo created
candidate@node-1:~/humane-storks$ kubectl get quota -n go
No resources found in go namespace.
candidate@node-1:~/humane-storks$ kubectl get quota -n goshawk
NAME      AGE     REQUEST   LIMIT
pod-demo  19s    pods: 9/10
candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-kvtjr
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-4whfm
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-dfk7l
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-z5zrt
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-2774b
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$
```

## Question #2



Task:

Modify the existing Deployment named broker-deployment running in namespace quetzal so that its containers.

- 1) Run with user ID 30000 and
- 2) Privilege escalation is forbidden

The broker-deployment manifest file can be found at:

`~/daring-moccasin/broker-deployment.yaml`

See the solution below.

## Explanation

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
containers:
  - name: broker
    image: redis:alpine
    ports:
      - containerPort: 6379
    securityContext:
      runAsUser: 30000
      privileged: false
:wq!
```

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/daring-moccasin/broker-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/daring-moccasin/broker-deployment.yaml
deployment.apps/broker-deployment configured
candidate@node-1:~$ kubectl get pods -n quetzal
NAME                  READY   STATUS    RESTARTS   AGE
broker-deployment-65446d6d94-868p6  1/1     Running   0          30s
broker-deployment-65446d6d94-8dn7l  1/1     Running   0          32s
broker-deployment-65446d6d94-p4h4l  1/1     Running   0          31s
candidate@node-1:~$ kubectl get deploy -n quetzal
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
broker-deployment 3/3     3           3           7h3m
candidate@node-1:~$
```

Question #3



Task:

A pod within the Deployment named buffalo-deployment and in namespace gorilla is logging errors.

1) Look at the logs identify errors messages.

Find errors, including User “system:serviceaccount:gorilla:default” cannot list resource “deployment” [...] in the namespace “gorilla”

2) Update the Deployment buffalo-deployment to resolve the errors in the logs of the Pod.

The buffalo-deployment ‘S manifest can be found at -/prompt/escargot/buffalo-deployment.yaml

See the solution below.

## Explanation

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME          READY   STATUS    RESTARTS   AGE
backend-deployment-59d449b99d-cxct6  1/1     Running   0          20s
backend-deployment-59d449b99d-h2zjq  0/1     Running   0          9s
backend-deployment-78976f74f5-b8c85  1/1     Running   0          6h40m
backend-deployment-78976f74f5-flfsj  1/1     Running   0          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3      3           3          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3      3           3          6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME          READY   STATUS    RESTARTS   AGE
buffalo-deployment-776844df7f-r5fsb  1/1     Running   0          6h38m
buffalo-deployment-859898c6f5-zx5gj  0/1     ContainerCreating   0          8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
buffalo-deployment  1/1      1           1          6h38m
candidate@node-1:~$ █
```

```

File Edit View Terminal Tabs Help

candidate@node-1:~$ vi ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
candidate@node-1:~$ vim .vimrc
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME READY STATUS RESTARTS AGE
backend-deployment-59d449b99d-cxct6 1/1 Running 0 20s
backend-deployment-59d449b99d-h2zjq 0/1 Running 0 9s
backend-deployment-78976f74f5-b8c85 1/1 Running 0 6h40m
backend-deployment-78976f74f5-flfsj 1/1 Running 0 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla

```

Text Description automatically generated

```

File Edit View Terminal Tabs Help
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME READY STATUS RESTARTS AGE
backend-deployment-59d449b99d-cxct6 1/1 Running 0 20s
backend-deployment-59d449b99d-h2zjq 0/1 Running 0 9s
backend-deployment-78976f74f5-b8c85 1/1 Running 0 6h40m
backend-deployment-78976f74f5-flfsj 1/1 Running 0 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME READY STATUS RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running 0 6h38m
buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating 0 8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME READY UP-TO-DATE AVAILABLE AGE
buffalo-deployment 1/1 1 1 6h38m
candidate@node-1:~$ 

```

```

File Edit View Terminal Tabs Help
NAME          READY   STATUS    RESTARTS   AGE
backend-deployment-59d449b99d-cxct6  1/1    Running   0          20s
backend-deployment-59d449b99d-h2zjq  0/1    Running   0          9s
backend-deployment-78976f74f5-b8c85  1/1    Running   0          6h40m
backend-deployment-78976f74f5-flfsj  1/1    Running   0          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3           3           6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3           3           6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME          READY   STATUS    RESTARTS   AGE
buffalo-deployment-776844df7f-r5fsb  1/1    Running   0          6h38m
buffalo-deployment-859898c6f5-zx5gj  0/1    ContainerCreating   0          8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
buffalo-deployment  1/1     1           1           6h38m
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017

```

Text Description automatically generated

```

File Edit View Terminal Tabs Help
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2022-09-24T04:27:03Z"
  generation: 1
  labels:
    app: nginx
    name: ckad00017-deployment
  namespace: ckad00017
  resourceVersion: "3349"
  uid: lcd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
      type: RollingUpdate
    template:
      metadata:
-- INSERT --

```

```

File Edit View Terminal Tabs Help
resourceVersion: "3349"
uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: nginx
  spec:
    containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
          - containerPort: 80
            protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
-- INSERT --
46,14 39%

```

```

File Edit View Terminal Tabs Help
backend-deployment-59d449b99d-h2zjq 0/1 Running 0 9s
backend-deployment-78976f74f5-b8c85 1/1 Running 0 6h40m
backend-deployment-78976f74f5-flfsj 1/1 Running 0 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME READY STATUS RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running 0 6h38m
buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating 0 8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME READY UP-TO-DATE AVAILABLE AGE
buffalo-deployment 1/1 1 1 6h38m
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$ 

```

```

File Edit View Terminal Tabs Help
buffalo-deployment 1/1 1 1 6h38m
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose service/ckad00017 --name=cherry --port=8888 --type=NodePort
service/cherry exposed
candidate@node-1:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes ClusterIP  10.96.0.1  <none>        443/TCP   77d
candidate@node-1:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
cherry   NodePort  10.100.100.176 <none>        8888:30683/TCP 24s
candidate@node-1:~$ kubectl get svc

```

Text Description automatically generated

```

File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
cherry   NodePort  10.100.100.176 <none>        8888:30683/TCP 46s
candidate@node-1:~$ history
1 vi ~/spicy-pikachu/backend-deployment.yaml
2 kubectl config use-context sk8s
3 vim .vimrc
4 vim ~/spicy-pikachu/backend-deployment.yaml
5 kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
6 kubectl get pods -n staging
7 kubectl get deploy -n staging
8 vim ~/spicy-pikachu/backend-deployment.yaml
9 kubectl config use-context k8s
10 kubectl set serviceaccount deploy app-1 app -n frontend
11 kubectl config use-context k8s
12 vim ~/prompt-escargot/buffalo-deployment.yaml
13 kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
14 kubectl get pods -n gorilla
15 kubectl get deploy -n gorilla
16 kubectl config use-context k8s
17 kubectl edit deploy ckad00017-deployment -n ckad00017
18 kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
19 kubectl get svc
20 kubectl get svc -n ckad00017
21 kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
22 kubectl get svc -n ckad00017
23 history
candidate@node-1:~$ 

```

Question #4

You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-context sk8s
```

Task:

Update the Deployment app-1 in the frontend namespace to use the existing ServiceAccount app.

See the solution below.

## Explanation

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
candidate@node-1:~$ vi ~/spicy-pikachu/backend-deployment.yaml  
candidate@node-1:~$ kubectl config use-context sk8s  
Switched to context "sk8s".  
candidate@node-1:~$ vim .vimrc  
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml  
candidate@node-1:~$ kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml  
deployment.apps/backend-deployment configured  
candidate@node-1:~$ kubectl get pods -n staging  
NAME READY STATUS RESTARTS AGE  
backend-deployment-59d449b99d-cxct6 1/1 Running 0 20s  
backend-deployment-59d449b99d-h2zjq 0/1 Running 0 9s  
backend-deployment-78976f74f5-b8c85 1/1 Running 0 6h40m  
backend-deployment-78976f74f5-flfsj 1/1 Running 0 6h40m  
candidate@node-1:~$ kubectl get deploy -n staging  
NAME READY UP-TO-DATE AVAILABLE AGE  
backend-deployment 3/3 3 3 6h40m  
candidate@node-1:~$ kubectl get deploy -n staging  
NAME READY UP-TO-DATE AVAILABLE AGE  
backend-deployment 3/3 3 3 6h41m  
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml  
candidate@node-1:~$ kubectl config use-context k8s  
Switched to context "k8s".  
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend  
deployment.apps/app-1 serviceaccount updated  
candidate@node-1:~$
```

### Question #5

Set configuration context:

```
[student@node-1] $ kubectl config  
use-context k8s
```

### Set Configuration Context:

```
[student@node-1] $ | kubectl
```

Config use-context k8s

### Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port 90 . As this port changes to 5050 an additional container needs to be added to the poller pod which adapts the container to connect to this new port. This should be realized as an ambassador container within the pod.

### Task

- Update the nginxsvc service to serve on port 5050.
- Add an HAProxy container named haproxy bound to port 90 to the poller pod and deploy the enhanced pod. Use the image haproxy and inject the configuration located at /opt/KDMC00101/haproxy.cfg, with a ConfigMap named haproxy-config, mounted into the container so that haproxy.cfg is available at /usr/local/etc/haproxy/haproxy.cfg. Ensure that you update the args of the poller container to connect to localhost instead of nginxsvc so that the connection is correctly proxied to the new service endpoint. You must not modify the port of the endpoint in poller's args . The spec file used to create the initial poller pod is available in /opt/KDMC00101/poller.yaml

See the solution below.

### Explanation

#### Solution:

To update the nginxsvc service to serve on port 5050, you will need to edit the service's definition yaml file. You can use the kubectl edit command to edit the service in place.

```
kubectl edit svc nginxsvc
```

This will open the service definition yaml file in your default editor. Change the targetPort of the service to 5050 and save the file.

To add an HAProxy container named haproxy bound to port 90 to the poller pod, you will need to edit the pod's definition yaml file located at /opt/KDMC00101/poller.yaml.

You can add a new container to the pod's definition yaml file, with the following configuration:

containers:

```
- name: haproxy
```

```
image: haproxy
```

ports:

- containerPort: 90

volumeMounts:

- name: haproxy-config

mountPath: /usr/local/etc/haproxy/haproxy.cfg

subPath: haproxy.cfg

args: ["haproxy", "-f", "/usr/local/etc/haproxy/haproxy.cfg"]

This will add the HAProxy container to the pod and configure it to listen on port 90. It will also mount the ConfigMap haproxy-config to the container, so that haproxy.cfg is available at /usr/local/etc/haproxy/haproxy.cfg.

To inject the configuration located at /opt/KDMC00101/haproxy.cfg to the container, you will need to create a ConfigMap using the following command:

```
kubectl create configmap haproxy-config --from-file=/opt/KDMC00101/haproxy.cfg
```

You will also need to update the args of the poller container so that it connects to localhost instead of nginxsvc. You can do this by editing the pod's definition yaml file and changing the args field to args: ["poller", "--host=localhost"].

Once you have made these changes, you can deploy the updated pod to the cluster by running the following command:

```
kubectl apply -f /opt/KDMC00101/poller.yaml
```

This will deploy the enhanced pod with the HAProxy container to the cluster. The HAProxy container will listen on port 90 and proxy connections to the nginxsvc service on port 5050. The poller container will connect to localhost instead of nginxsvc, so that the connection is correctly proxied to the new service endpoint.

Please note that, this is a basic example and you may need to tweak the haproxy.cfg file and the args based on your use case.

#### Question #6

You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate-node-1] $ kubectl config use-context sk8s
```

Task:

Create a Pod named nginx resources in the existing pod resources namespace.

Specify a single container using nginx:stable image.

Specify a resource request of 300m cpus and 1G1 of memory for the Pod's container.

See the solution below.

## Explanation

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
```

Text Description automatically generated with medium confidence

```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx:stable
    name: nginx-resources
    resources:
      requests:
        cpu: 300m
        memory: "1Gi"
```

Text Description automatically generated

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
candidate@node-1:~$ kubectl create -f hw.yaml
pod/nginx-resources created
candidate@node-1:~$ kubectl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0          13s
candidate@node-1:~$ kubectl describe pods -n pod-resources
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
memory: 1Gi
Environment: <none>
Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dmx9j (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-dmx9j:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
  QoS Class:      Burstable
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age  From           Message
  ----  -----  --  --  -----
  Normal Scheduled  20s default-scheduler  Successfully assigned pod-resources/nginx-resources to k8s-node-0
  Normal Pulling   19s kubelet         Pulling image "nginx:stable"
  Normal Pulled    13s kubelet         Successfully pulled image "nginx:stable" in 6.55664052s
  Normal Created   13s kubelet         Created container nginx-resources
  Normal Started   12s kubelet         Started container nginx-resources
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl create deploy expose -n ckad00014 --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml> 
```

## Question #7



## Context

Your application's namespace requires a specific service account to be used.

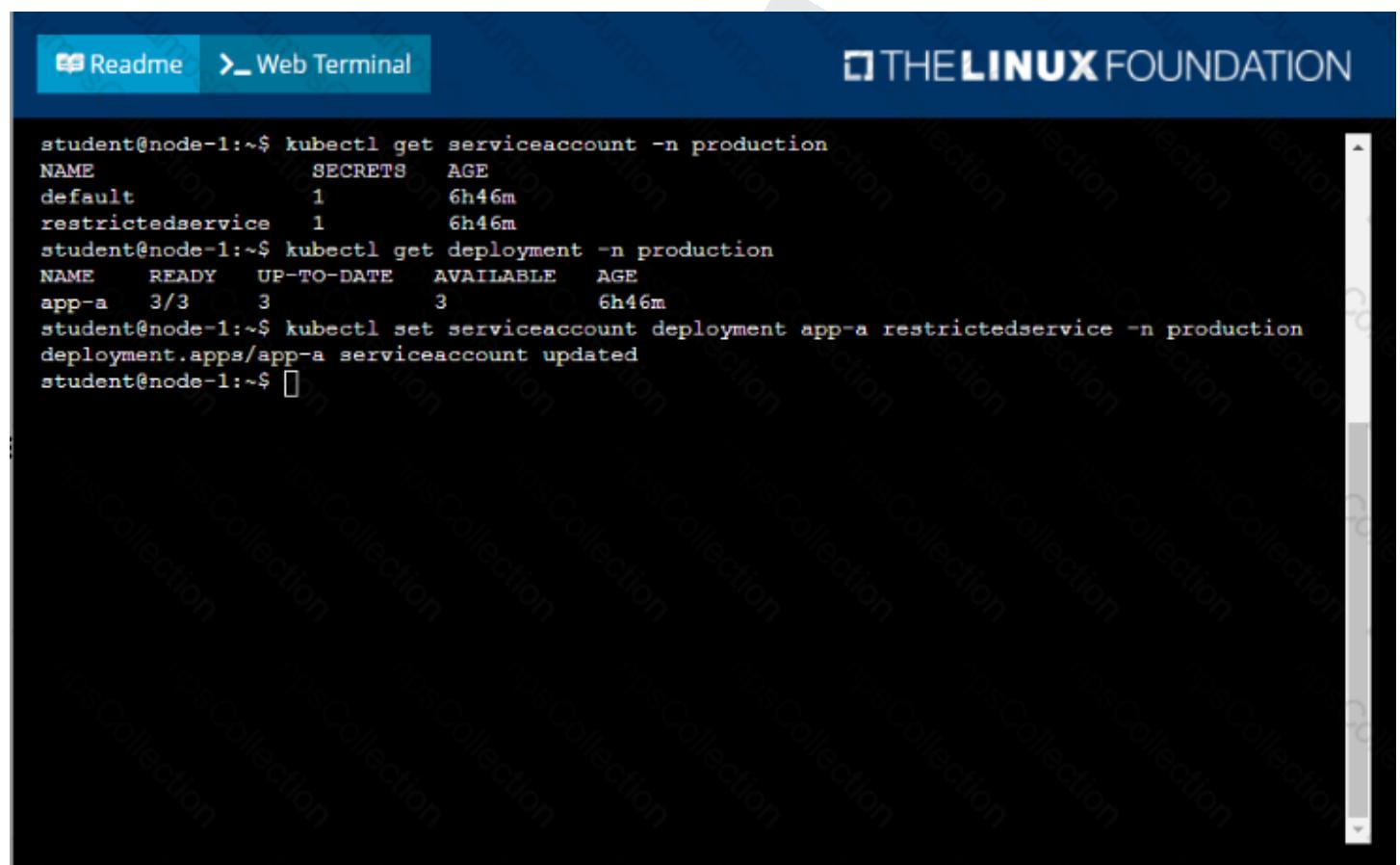
### Task

Update the app-a deployment in the production namespace to run as the restrictedservice service account. The service account has already been created.

See the solution below.

### Explanation

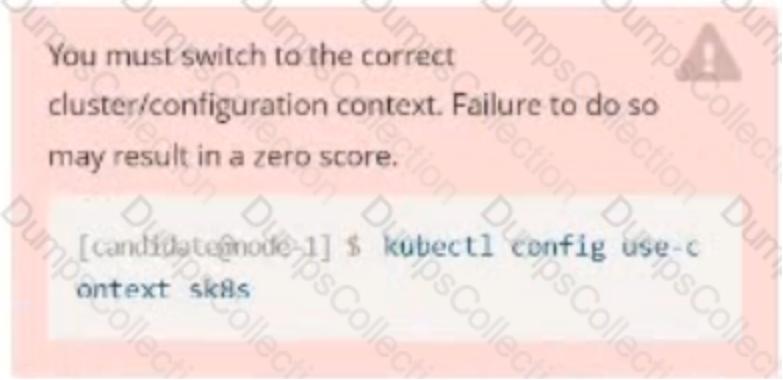
Solution:



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active and displays the following command-line session:

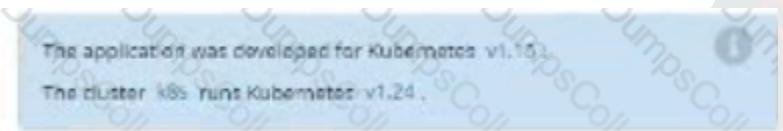
```
student@node-1:~$ kubectl get serviceaccount -n production
NAME      SECRETS   AGE
default   1          6h46m
restrictedservice   1          6h46m
student@node-1:~$ kubectl get deployment -n production
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
app-a   3/3     3           3           6h46m
student@node-1:~$ kubectl set serviceaccount deployment app-a restrictedservice -n production
deployment.apps/app-a serviceaccount updated
student@node-1:~$ 
```

Question #:8



Task:

- 1) Fix any API depreciation issues in the manifest file -/credible-mite/www.yaml so that this application can be deployed on cluster K8s.



- 2) Deploy the application specified in the updated manifest file -/credible-mite/www.yaml in namespace cobra

See the solution below.

## Explanation

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim -/credible-mite/www.yaml
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
apiVersion: apps/v1
kind: Deployment
metadata:
  name: www-deployment
  namespace: cobra
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: "nginx:stable"
          ports:
            - containerPort: 80
          volumeMounts:
            - mountPath: /var/log/nginx
              name: logs
          env:
            - name: NGINX_ENTRYPOINT_QUIET_LOGS
              value: "1"
      volumes:
        - name: logs
          emptyDir: {}
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
deployment.apps/expose created
candidate@node-1:~$ kubectl get pods -n ckad00014
NAME          READY   STATUS        RESTARTS   AGE
expose-85dd99d4d9-25675  0/1    ContainerCreating  0          6s
expose-85dd99d4d9-4fhcc  0/1    ContainerCreating  0          6s
expose-85dd99d4d9-fld7j  0/1    ContainerCreating  0          6s
expose-85dd99d4d9-tt6rm  0/1    ContainerCreating  0          6s
expose-85dd99d4d9-vjd8b  0/1    ContainerCreating  0          6s
expose-85dd99d4d9-vtzpq  0/1    ContainerCreating  0          6s
candidate@node-1:~$ kubectl get deploy -n ckad00014
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
expose   6/6     6           6           15s
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ kubectl apply -f ~/credible-mite/www.yaml
deployment.apps/www-deployment created
candidate@node-1:~$ kubectl get pods -n cobra
NAME          READY   STATUS        RESTARTS   AGE
www-deployment-d899c6b49-d6ccg  1/1    Running      0          6s
www-deployment-d899c6b49-f796l  0/1    ContainerCreating  0          6s
www-deployment-d899c6b49-ztfcw  0/1    ContainerCreating  0          6s
candidate@node-1:~$ kubectl get deploy -n cobra
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
www-deployment  3/3     3           3           11s
candidate@node-1:~$ kubectl get pods -n cobra
NAME          READY   STATUS        RESTARTS   AGE
www-deployment-d899c6b49-d6ccg  1/1    Running      0          14s
www-deployment-d899c6b49-f796l  1/1    Running      0          14s
www-deployment-d899c6b49-ztfcw  1/1    Running      0          14s
candidate@node-1:~$
```

### Question #9

Set configuration context:

```
[student@node-1] $ kubectl config  
use-context k8s
```

### Task

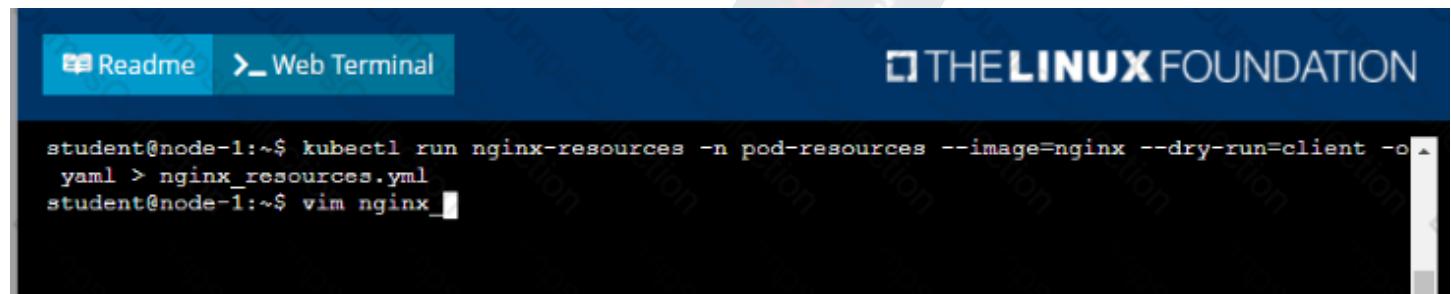
You are required to create a pod that requests a certain amount of CPU and memory, so it gets scheduled to a node that has those resources available.

- Create a pod named nginx-resources in the pod-resources namespace that requests a minimum of 200m CPU and 1Gi memory for its container
- The pod should use the nginx image
- The pod-resources namespace has already been created

See the solution below.

## Explanation

Solution:



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active, displaying the command: `student@node-1:~$ kubectl run nginx-resources --image=nginx --dry-run=client -o yaml > nginx_resources.yml`. Below the command, the terminal prompt shows: `student@node-1:~$ vim nginx_`. The background of the terminal window features a watermark with the text "Collection" repeated diagonally.



The screenshot shows a terminal window with two tabs: "Readme" and "Web Terminal". The "Web Terminal" tab is active, displaying the generated YAML configuration for the pod:

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

At the bottom of the terminal, the command `"nginx_resources.yml" 16L, 289C` is visible. The background of the terminal window features a watermark with the text "Collection" repeated diagonally.

Readme ➔ Web Terminal

THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources:
      requests:
        cpu: 200m
        memory: "1Gi"
```

-- INSERT --

15, 22 All

Readme ➔ Web Terminal

THE LINUX FOUNDATION

```
student@node-1:~$ kubectl run nginx-resources --image=nginx --dry-run=client -o yaml > nginx_resources.yml
student@node-1:~$ vim nginx_resources.yml
student@node-1:~$ kubectl create -g nginx_resources.yml
Error: unknown shorthand flag: 'g' in -g
See 'kubectl create --help' for usage.
student@node-1:~$ kubectl create -f nginx_resources.yml
pod/nginx-resources created
student@node-1:~$ kubectl get pods -n pod-re
```

Readme ➔ Web Terminal

THE LINUX FOUNDATION

```
student@node-1:~$ kubectl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0          8s
student@node-1:~$
```

Question #:10

No configuration context change is required for this task.

Task:

A Dockerfile has been prepared at -/human-stork/build/Dockerfile

- 1) Using the prepared Dockerfile, build a container image with the name macque and tag 3.0. You may install and use the tool of your choice.

*Multiple image builders and tools have been pre-installed in the base system, including: docker, skopeo, buildah, img, and podman.*

Please do not push the built image to a registry, run a container, or otherwise consume it.

- 2) Using the tool of your choice export the built container image in OC-format and store it at -/human-stork/macque 3.0 tar

See the solution below.

## Explanation

Solution:

```

candidate@node-1:~$ cd humane-stork/build/
candidate@node-1:~/humane-stork/build$ ls -l
total 16
-rw-r--r-- 1 candidate candidate 201 Sep 24 04:21 Dockerfile
-rw-r--r-- 1 candidate candidate 644 Sep 24 04:21 text1.html
-rw-r--r-- 1 candidate candidate 813 Sep 24 04:21 text2.html
-rw-r--r-- 1 candidate candidate 383 Sep 24 04:21 text3.html
candidate@node-1:~/humane-stork/build$ sudo docker build -t macaque:3.0 .
Sending build context to Docker daemon 6.144kB
Step 1/5 : FROM docker.io/lfcncf/nginx:mainline
--> ea335eea17ab
Step 2/5 : ADD text1.html /usr/share/nginx/html/
--> 8967ee9ee5d0
Step 3/5 : ADD text2.html /usr/share/nginx/html/
--> cb0554422f26
Step 4/5 : ADD text3.html /usr/share/nginx/html/
--> 62e879ab821e
Step 5/5 : COPY text2.html /usr/share/nginx/html/index.html
--> 331c8a94372c
Successfully built 331c8a94372c
Successfully tagged macaque:3.0
candidate@node-1:~/humane-stork/build$ sudo docker save macaque:3.0 > ~/humane-stork/macaque-3.0.tar
candidate@node-1:~/humane-stork/build$ cd ..
candidate@node-1:~/humane-stork$ ls -l
total 142532
drwxr-xr-x 2 candidate candidate 4096 Sep 24 04:21 build
-rw-rw-r-- 1 candidate candidate 145948672 Sep 24 11:39 macaque-3.0.tar
candidate@node-1:~/humane-stork$ 
```

```

File Edit View Terminal Tabs Help
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl label pod ckad00018-newpod -n ckad00018 db-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ kubectl apply -f ~/chief-cardinal/nosql.yaml
deployment.apps/nosql configured
candidate@node-1:~$ kubectl get pods -n crayfish
NAME           READY   STATUS    RESTARTS   AGE
nosql-74cccf7d64-lkqlg  1/1     Running   0          3m2s
candidate@node-1:~$ kubectl get deploy -n crayfish
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nosql   1/1     1           1           7h16m
candidate@node-1:~$ cd humane-stork/build/
candidate@node-1:~/humane-stork/build$ ls -l
total 16
-rw-r--r-- 1 candidate candidate 201 Sep 24 04:21 Dockerfile
-rw-r--r-- 1 candidate candidate 644 Sep 24 04:21 text1.html
-rw-r--r-- 1 candidate candidate 813 Sep 24 04:21 text2.html
-rw-r--r-- 1 candidate candidate 383 Sep 24 04:21 text3.html
candidate@node-1:~/humane-stork/build$ sudo docker build -t macaque:3.0 .
Sending build context to Docker daemon 6.144kB
Step 1/5 : FROM docker.io/lfcncf/nginx:mainline
--> ea335eea17ab
Step 2/5 : ADD text1.html /usr/share/nginx/html/
--> 8967ee9ee5d0
Step 3/5 : ADD text2.html /usr/share/nginx/html/
--> cb0554422f26
Step 4/5 : ADD text3.html /usr/share/nginx/html/

```

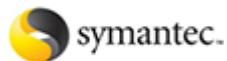
```
File Edit View Terminal Tabs Help
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ kubectl apply -f ~/chief-cardinal/nosql.yaml
deployment.apps/nosql configured
candidate@node-1:~$ kubectl get pods -n crayfish
NAME          READY   STATUS    RESTARTS   AGE
nosql-74cccf7d64-lkqlg  1/1     Running   0          3m2s
candidate@node-1:~$ kubectl get deploy -n crayfish
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nosql       1/1     1           1           7h16m
candidate@node-1:~$ cd humane-stork/build/
candidate@node-1:~/humane-stork/build$ ls -l
total 16
-rw-r--r-- 1 candidate candidate 201 Sep 24 04:21 Dockerfile
-rw-r--r-- 1 candidate candidate 644 Sep 24 04:21 text1.html
-rw-r--r-- 1 candidate candidate 813 Sep 24 04:21 text2.html
-rw-r--r-- 1 candidate candidate 383 Sep 24 04:21 text3.html
candidate@node-1:~/humane-stork/build$ sudo docker build -t macaque:3.0 .
Sending build context to Docker daemon 6.144kB
Step 1/5 : FROM docker.io/lfcncnfg/nginx:mainline
--> ea335eeal7ab
Step 2/5 : ADD text1.html /usr/share/nginx/html/
--> 8967ee9ee5d0
Step 3/5 : ADD text2.html /usr/share/nginx/html/
--> cb0554422f26
Step 4/5 : ADD text3.html /usr/share/nginx/html/
--> 62e879abb821e
Step 5/5 : COPY text2.html /usr/share/nginx/html/index.html
--> 331c8a94372c
Successfully built 331c8a94372c
Successfully tagged macaque:3.0
candidate@node-1:~/humane-stork/build$ sudo docker save macaque:3.0 > ~/humane-stork/macaque-3.0.tar
```

# About dumpscollection.com

[dumpscollection.com](#) was founded in 2007. We provide latest & high quality IT / Business Certification Training Exam Questions, Study Guides, Practice Tests.

We help you pass any IT / Business Certification Exams with 100% Pass Guaranteed or Full Refund. Especially Cisco, CompTIA, Citrix, EMC, HP, Oracle, VMware, Juniper, Check Point, LPI, Nortel, EXIN and so on.

View list of all certification exams: [All vendors](#)



We prepare state-of-the art practice tests for certification exams. You can reach us at any of the email addresses listed below.

- ⌚ Sales: [sales@dumpscollection.com](mailto:sales@dumpscollection.com)
- ⌚ Feedback: [feedback@dumpscollection.com](mailto:feedback@dumpscollection.com)
- ⌚ Support: [support@dumpscollection.com](mailto:support@dumpscollection.com)
- ⌚ Skype ID: [crack4sure@gmail.com](mailto:crack4sure@gmail.com)

Any problems about IT certification or our products, You can write us back and we will get back to you within 24 hours.

**15% Discount Coupon Code:**

**DC15disc**