ITDumpsKR
IT인증시험최신덤프문제

ITDumpsKR

PDF버전 + PC테스트엔진 + 온라인테스트엔진

PDF버전: 편하고 쉽게 공부하기. 출력가능한 PDF. 운영 시스템 플랫폼을 무시한 전자파일형태입니다.

PC테스트엔진: 고객님의 사용에 편리하도록 여러개의 PC에 설치가능합니다.

온라인테스트엔진: 온라인테스트엔진은 WEB브라우저를 기초로 한 소프트엔진이기에 Windows/Mac/Anfroid/iOS등을 지지합니다.

http://www.itdumpskr.com

IT 인증시험 한방에 패스시키는 최신버전 시험대비덤프

Exam : **CKAD**

Title : Linux Foundation Certified Kubernetes Application Developer Exam

Vendor : Linux Foundation

Version : DEMO

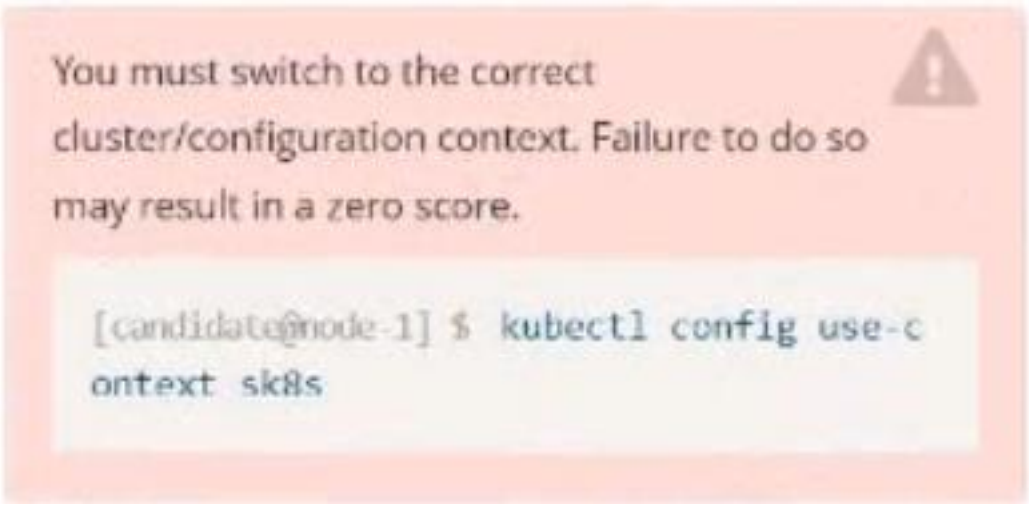
NO.1 Refer to Exhibit.

Set Configuration Context:
[student@node-1] $ | kubectl
Config use-context k8s
Context
A web application requires a specific version of redis to be used as a cache.
Task
Create a pod with the following characteristics, and leave it running when complete:
* The pod must run in the web namespace.
The namespace has already been created
* The name of the pod should be cache
* Use the lfccncf/redis image with the 3.2 tag
* Expose port 6379

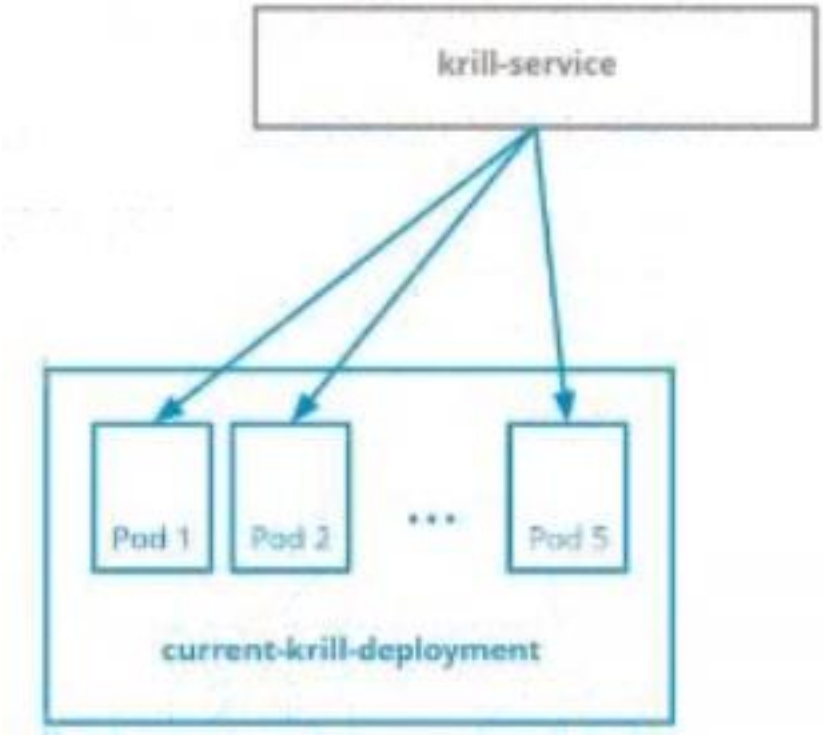
Answer:

Solution:

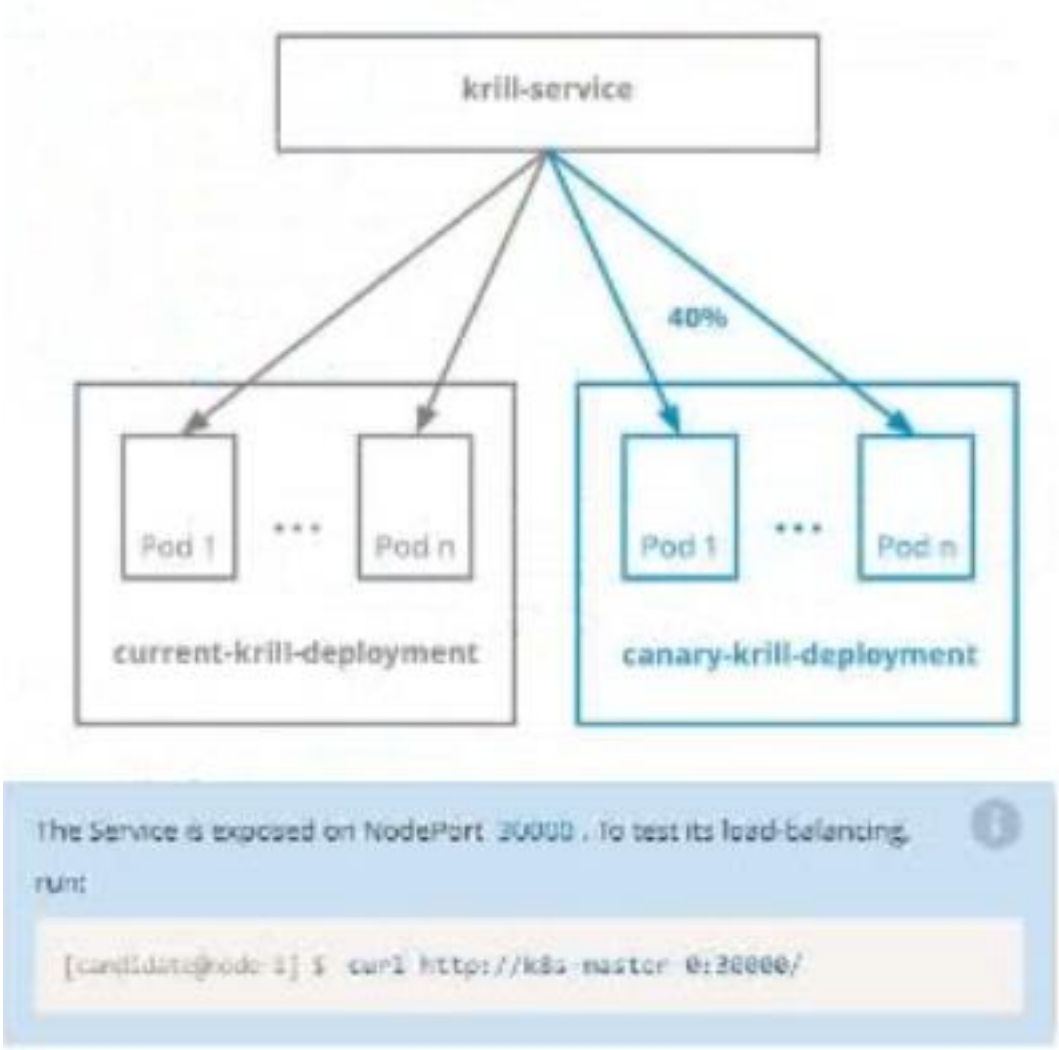
NO.2 Refer to Exhibit.



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)



Answer:

Solution:

```
candidate@node-1:~/humane-stork$ kubectl  scale  deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-stork$ 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-stork$ wget https://k8s.io/examples/
File  Edit  View  Terminal  Tabs  Help
candidate@node-1:~/humane-stork$ 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-stork$ 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-stork$ vim quota-pod.yaml
candidate@node-1:~/humane-stork$ kubectl  create  -f quota-pod.yaml
resourcequota/pod-demo created
candidate@node-1:~/humane-stork$ kubectl get quota -n go
No resources found in go namespace.
candidate@node-1:~/humane-stork$ kubectl get quota -n goshawk
NAME       AGE   REQUEST      LIMIT
pod-demo   19s   pods: 9/10
candidate@node-1:~/humane-stork$ 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-stork$ 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-stork$ 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-stork$ 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-stork$ 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-stork$
```

NO.3 Refer to Exhibit.

Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format B.
Task:
* Create a deployment named deployment-xyz in the default namespace, that:
* Includes a primary
lfccncf/busybox:1 container, named logger-dev
* includes a sidecar lfccncf/fluentd:v0.12 container, named adapter-zen
* Mounts a shared volume /tmp/log on both containers, which does not persist when the pod is deleted
* Instructs the logger-dev
container to run the command

```
while true; do
echo "i luv cncf" >> /
tmp/log/input.log;
sleep 10;
done
```

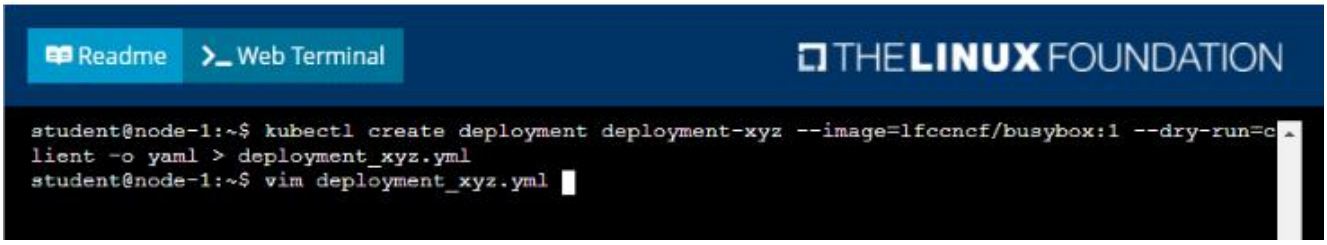
which should output logs to /tmp/log/input.log in plain text format, with example values:

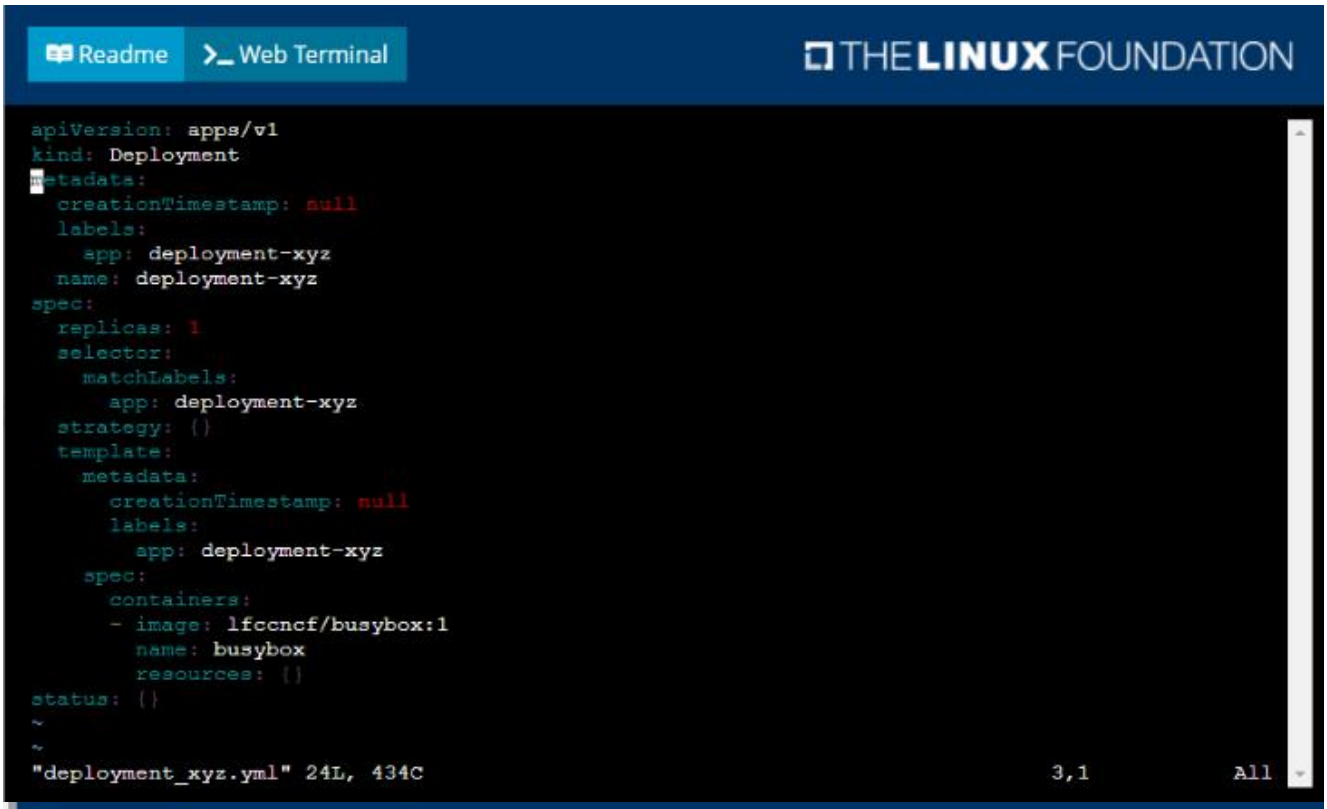
```
i luv cncf
i luv cncf
i luv cncf
```

* The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.* in Fluentd JSON format. Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configma p.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container

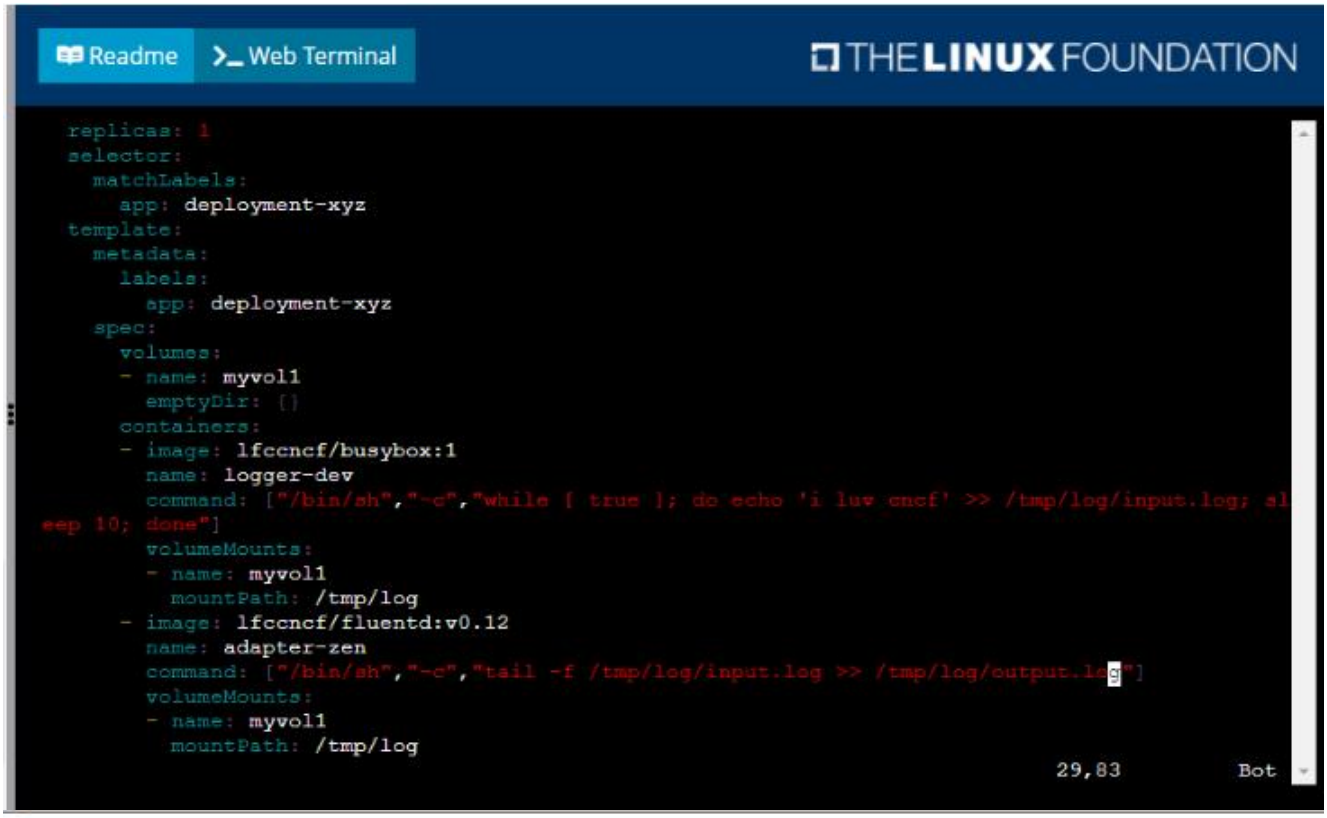
Answer:

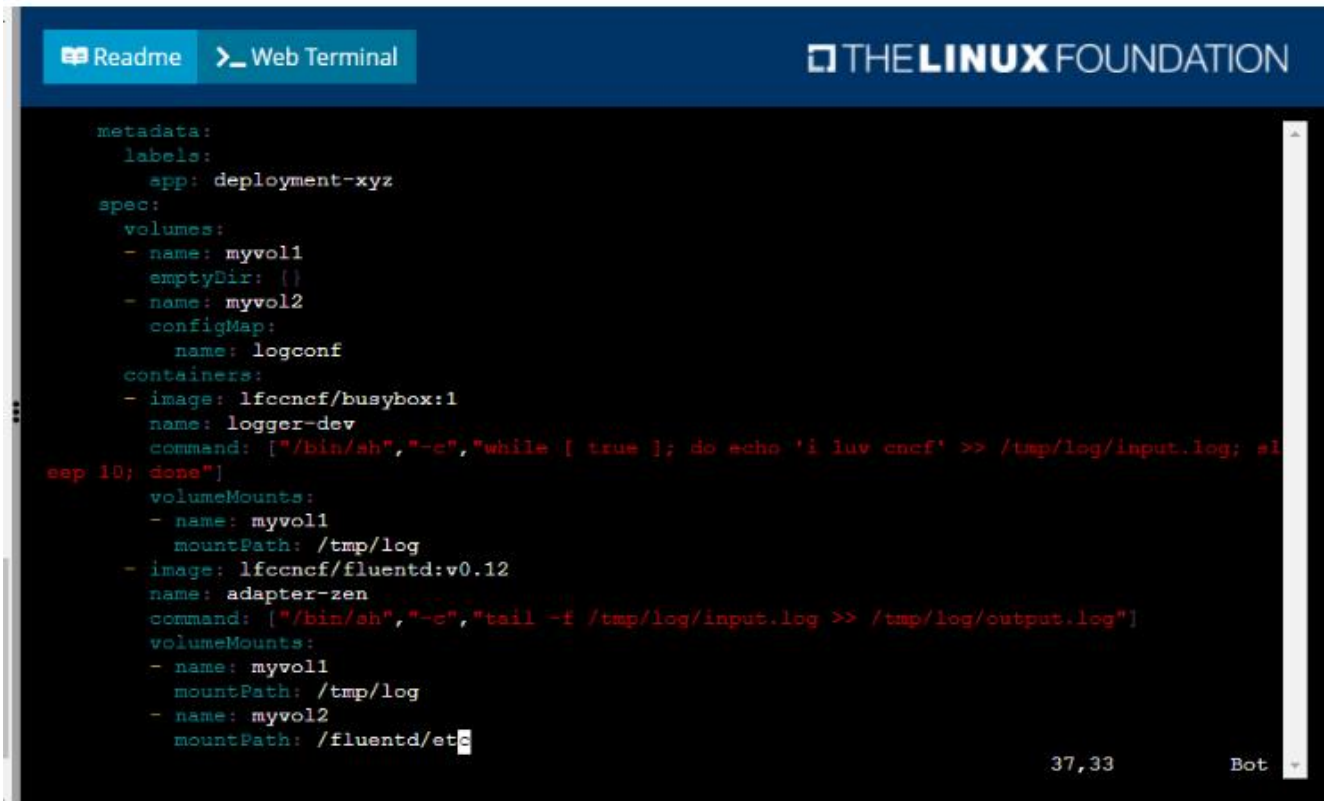
Solution:

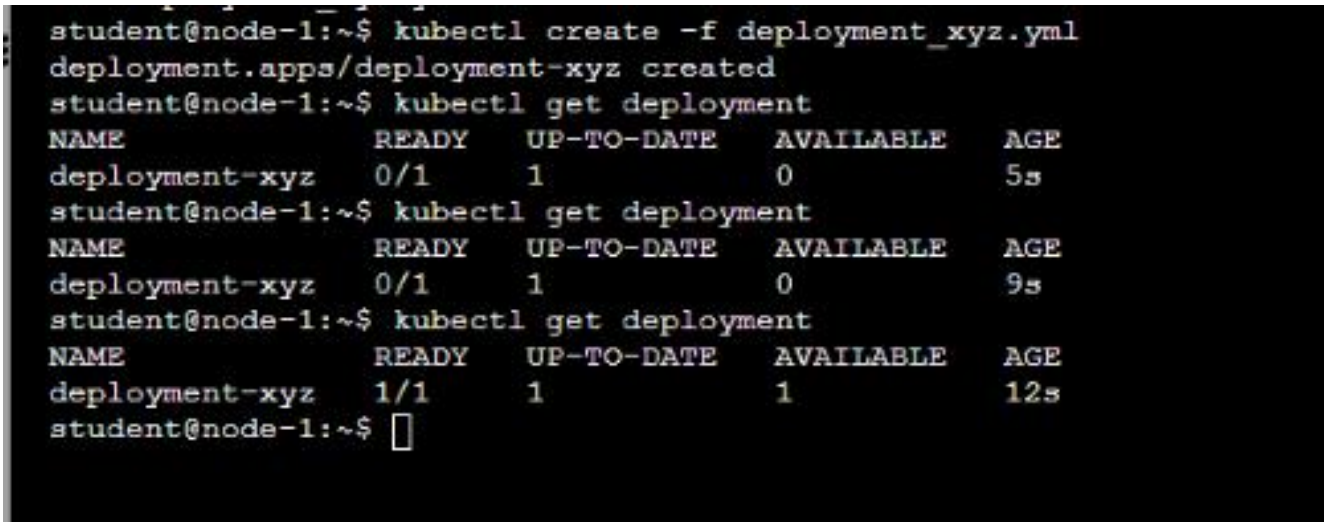



Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
creationTimestamp: null
labels:
app: deployment-xyz
name: deployment-xyz
spec:
replicas: 1
selector:
matchLabels:
app: deployment-xyz
strategy: {}
template:
metadata:
creationTimestamp: null
labels:
app: deployment-xyz
spec:
containers:
- image: lfccncf/busybox:1
name: busybox
resources: {}
status: {}
~
~
"deployment_xyz.yml" 24L, 434C
3,1
All


Readme
Web Terminal
THE LINUX FOUNDATION
kind: Deployment
metadata:
labels:
app: deployment-xyz
name: deployment-xyz
spec:
replicas: 1
selector:
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
3 lines yanked
27,22
Bot


Readme
Web Terminal
THE LINUX FOUNDATION
replicas: 1
selector:
matchLabels:
app: deployment-xyz
template:
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
containers:
- image: lfccncf/busybox:1
name: logger-dev
command: ["/bin/sh","-c","while [true]; do echo 'i luv cncf' >> /tmp/log/input.log; sl
eep 10; done"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/output.log"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
29,83
Bot


Readme
Web Terminal
THE LINUX FOUNDATION
metadata:
labels:
app: deployment-xyz
spec:
volumes:
- name: myvol1
emptyDir: {}
- name: myvol2
configMap:
name: logconf
containers:
- image: lfccncf/busybox:1
name: logger-dev
command: ["/bin/sh","-c","while [true]; do echo 'i luv cncf' >> /tmp/log/input.log; sl
eep 10; done"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- image: lfccncf/fluentd:v0.12
name: adapter-zen
command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/output.log"]
volumeMounts:
- name: myvol1
mountPath: /tmp/log
- name: myvol2
mountPath: /fluentd/etc
37,33
Bot


student@node-1:~$ kubectl create -f deployment_xyz.yml
deployment.apps/deployment-xyz created
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 0/1 1 0 5s
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 0/1 1 0 9s
student@node-1:~$ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
deployment-xyz 1/1 1 1 12s
student@node-1:~$