

Exam Questions CKA

Certified Kubernetes Administrator (CKA) Program

<https://www.2passeasy.com/dumps/CKA/>



NEW QUESTION 1

Create and configure the service front-end-service so it's accessible through NodePort and routes to the existing pod named front-end.

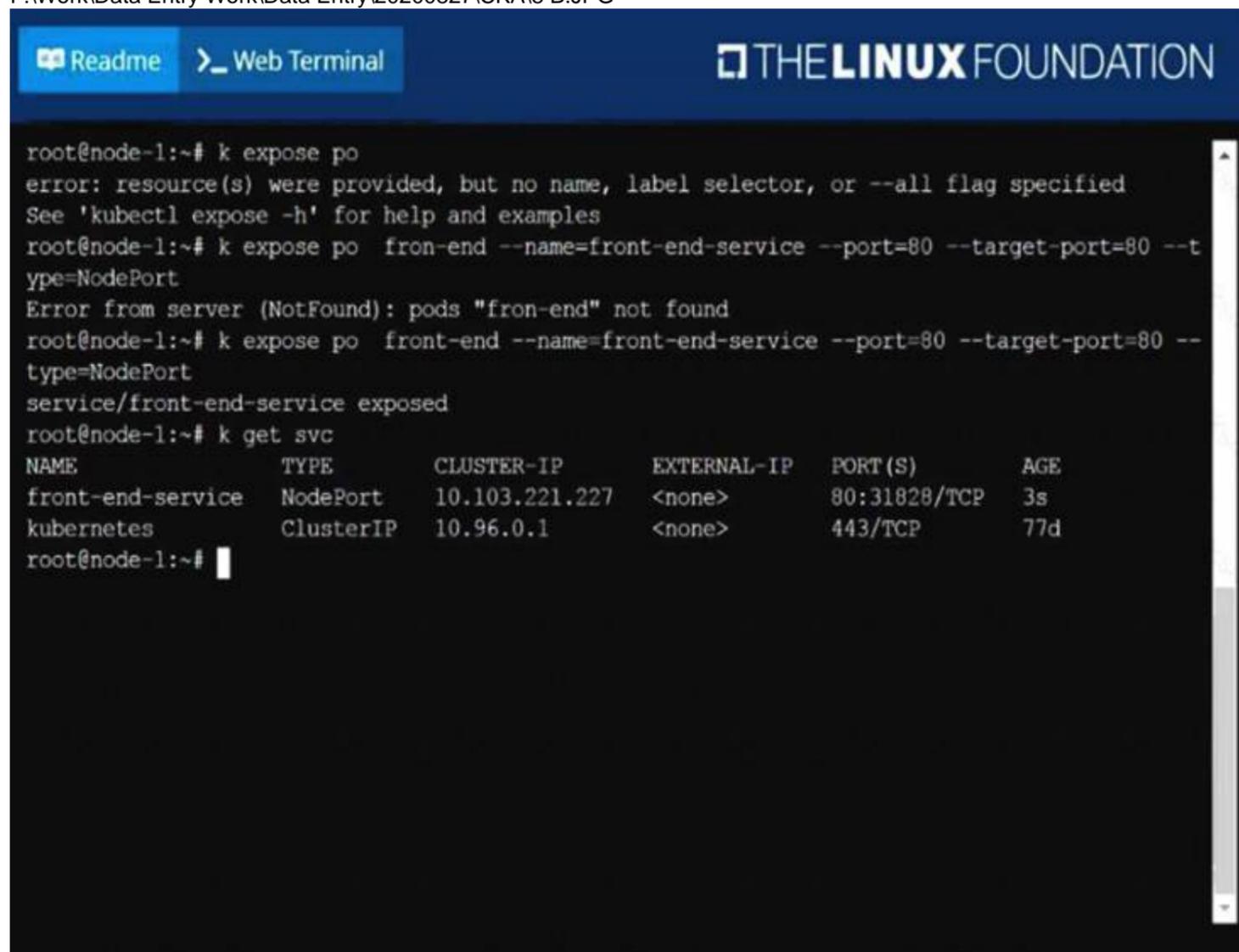
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\8 B.JPG



```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po  fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-end" not found
root@node-1:~# k expose po  front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
front-end-service   NodePort            10.103.221.227  <none>           80:31828/TCP     3s
kubernetes          ClusterIP           10.96.0.1       <none>           443/TCP          77d
root@node-1:~#
```

NEW QUESTION 2

A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state, ensuring that any changes are made permanent.

You can ssh to the failed node using:

```
[student@node-1] $ | ssh Wk8s-node-0
```

You can assume elevated privileges on the node with the following command:

```
[student@w8ks-node-0] $ | sudo -i
```

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 C.JPG

```
Readme > Web Terminal THE LINUX FOUNDATION

root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME             STATUS    ROLES    AGE   VERSION
wk8s-master-0   Ready    master   77d   v1.18.2
wk8s-node-0     NotReady <none>   77d   v1.18.2
wk8s-node-1     Ready    <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
█
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 D.JPG

```
Readme > Web Terminal THE LINUX FOUNDATION

wk8s-node-0     NotReady <none>   77d   v1.18.2
wk8s-node-1     Ready    <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic

   https://microk8s.io/ has docs and details.

4 packages can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
█
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 E.JPG

Readme Web Terminal THE LINUX FOUNDATION

```

https://microk8s.io/ has docs and details.

4 packages can be updated.
1 update is a security update.

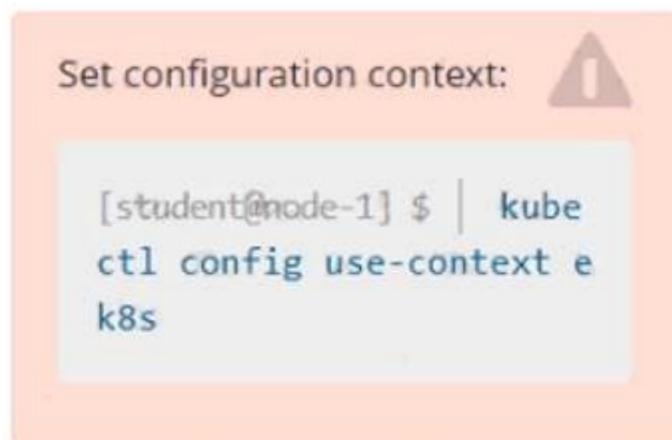
New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/systemd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME             STATUS    ROLES    AGE   VERSION
wk8s-master-0   Ready    master   77d   v1.18.2
wk8s-node-0     Ready    <none>   77d   v1.18.2
wk8s-node-1     Ready    <none>   77d   v1.18.2
root@node-1:~# █

```

NEW QUESTION 3

Score: 4%



Task

Set the node named ek8s-node-1 as unavailable and reschedule all the pods running on it.

- A. Mastered
- B. Not Mastered

Answer: A**Explanation:**

SOLUTION:

```

[student@node-1] > ssh ek8s
kubectl cordon ek8s-node-1
kubectl drain ek8s-node-1 --delete-local-data --ignore-daemonsets --force

```

NEW QUESTION 4

Create a pod that having 3 containers in it? (Multi-Container)

- A. Mastered
- B. Not Mastered

Answer: A**Explanation:**

image=nginx, image=redis, image=consul Name nginx container as "nginx-container" Name redis container as "redis-container" Name consul container as "consul-container"

Create a pod manifest file for a container and append container section for rest of the images

```
kubectl run multi-container --generator=run-pod/v1 --image=nginx -- dry-run -o yaml > multi-container.yaml
# then
vim multi-container.yaml apiVersion: v1
kind: Pod metadata: labels:
run: multi-container name: multi-container spec:
containers:
- image: nginx
name: nginx-container
- image: redis
name: redis-container
- image: consul
name: consul-container restartPolicy: Always
```

NEW QUESTION 5

Perform the following tasks:

- > Add an init container to hungry-bear (which has been defined in spec file /opt/KUCC00108/pod-spec-KUCC00108.yaml)
- > The init container should create an empty file named/workdir/calm.txt
- > If /workdir/calm.txt is not detected, the pod should exit
- > Once the spec file has been updated with the init container definition, the pod should be created

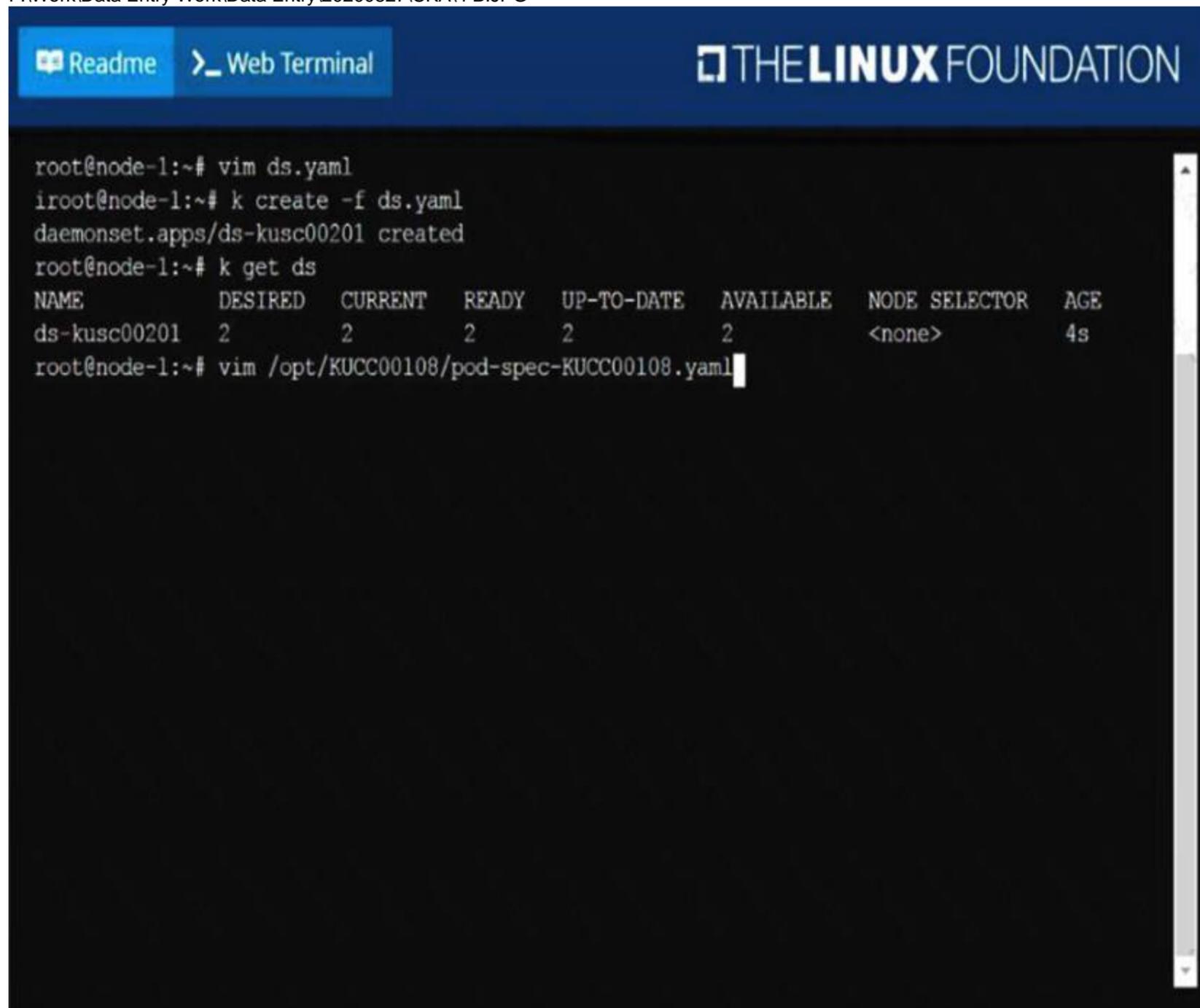
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

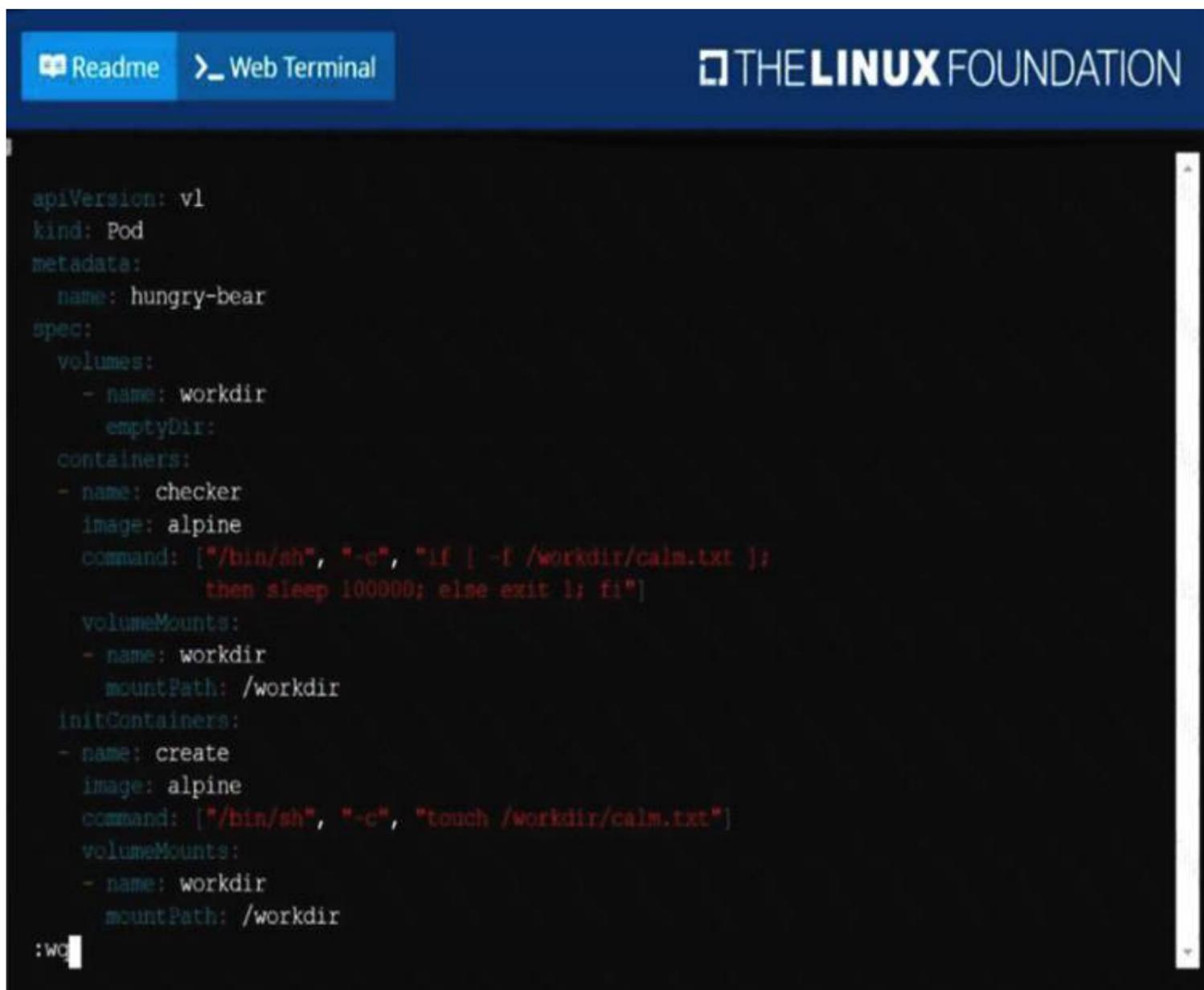
solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 B.JPG



```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201   2        2        2      2           2          <none>         4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
```

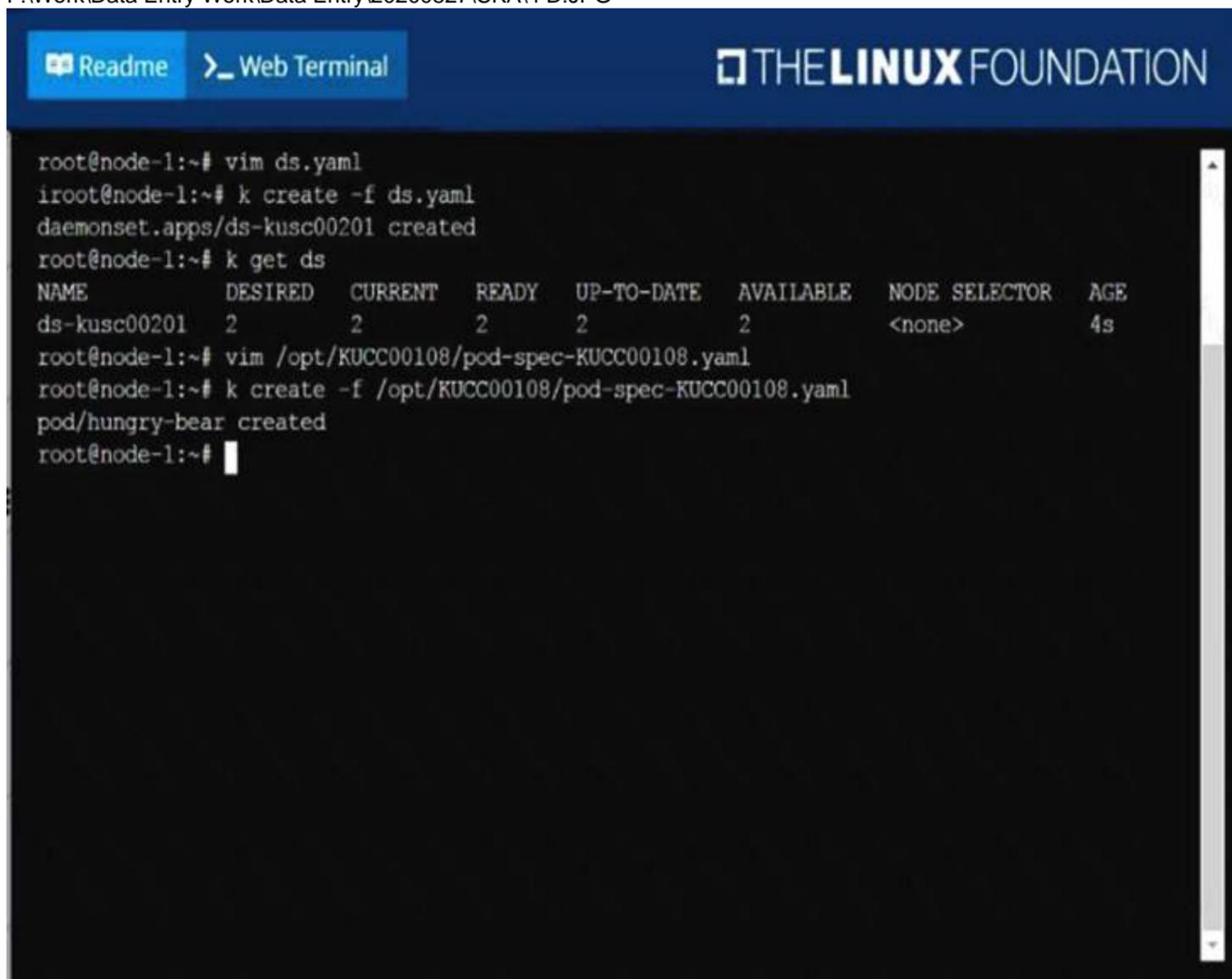
F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 C.JPG



The screenshot shows a web terminal interface with a dark background. At the top, there are two buttons: 'Readme' and 'Web Terminal'. The 'THE LINUX FOUNDATION' logo is in the top right. The terminal content displays a YAML pod specification for a pod named 'hungry-bear'. The specification includes metadata, a spec with a volume named 'workdir', and two containers: 'checker' and 'create'. The 'checker' container runs a shell script that sleeps for 100,000 seconds if a file 'calm.txt' is not present. The 'create' container runs a shell script that creates the file 'calm.txt'. The terminal ends with a prompt ':wC'.

```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
  - name: workdir
    emptyDir: {}
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ -f /workdir/calm.txt ];
      then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
  - name: create
    image: alpine
    command: ["/bin/sh", "-c", "touch /workdir/calm.txt"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
:wC
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\4 D.JPG



The screenshot shows a web terminal interface with a dark background. At the top, there are two buttons: 'Readme' and 'Web Terminal'. The 'THE LINUX FOUNDATION' logo is in the top right. The terminal content shows a series of commands and their outputs. The user runs 'vim ds.yaml', then 'k create -f ds.yaml', which outputs 'daemonset.apps/ds-kusc00201 created'. Then the user runs 'k get ds', which outputs a table of daemon sets. Finally, the user runs 'vim /opt/KUCC00108/pod-spec-KUCC00108.yaml', 'k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml', which outputs 'pod/hungry-bear created', and ends with a prompt.

```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201   2        2        2      2           2          <none>         4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~#
```

NEW QUESTION 6

Get IP address of the pod – "nginx-dev"

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
Kubect1 get po -o wide Using JsonPath
kubect1 get pods -o=jsonpath='{range items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'
```

NEW QUESTION 7

Get list of all the pods showing name and namespace with a jsonpath expression.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubectl get pods -o=jsonpath="{.items[*]['metadata.name'
, 'metadata.namespace']}"
```

NEW QUESTION 8

Create a namespace called 'development' and a pod with image nginx called nginx on this namespace.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubectl create namespace development
kubectl run nginx --image=nginx --restart=Never -n development
```

NEW QUESTION 9

Create a pod with environment variables as var1=value1. Check the environment variable in pod

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubectl run nginx --image=nginx --restart=Never --env=var1=value1
# then
kubectl exec -it nginx -- env
# or
kubectl exec -it nginx -- sh -c 'echo $var1'
# or
kubectl describe po nginx | grep value1
```

NEW QUESTION 10

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubect1 get pods --sort-by=.metadata.name
```

NEW QUESTION 10

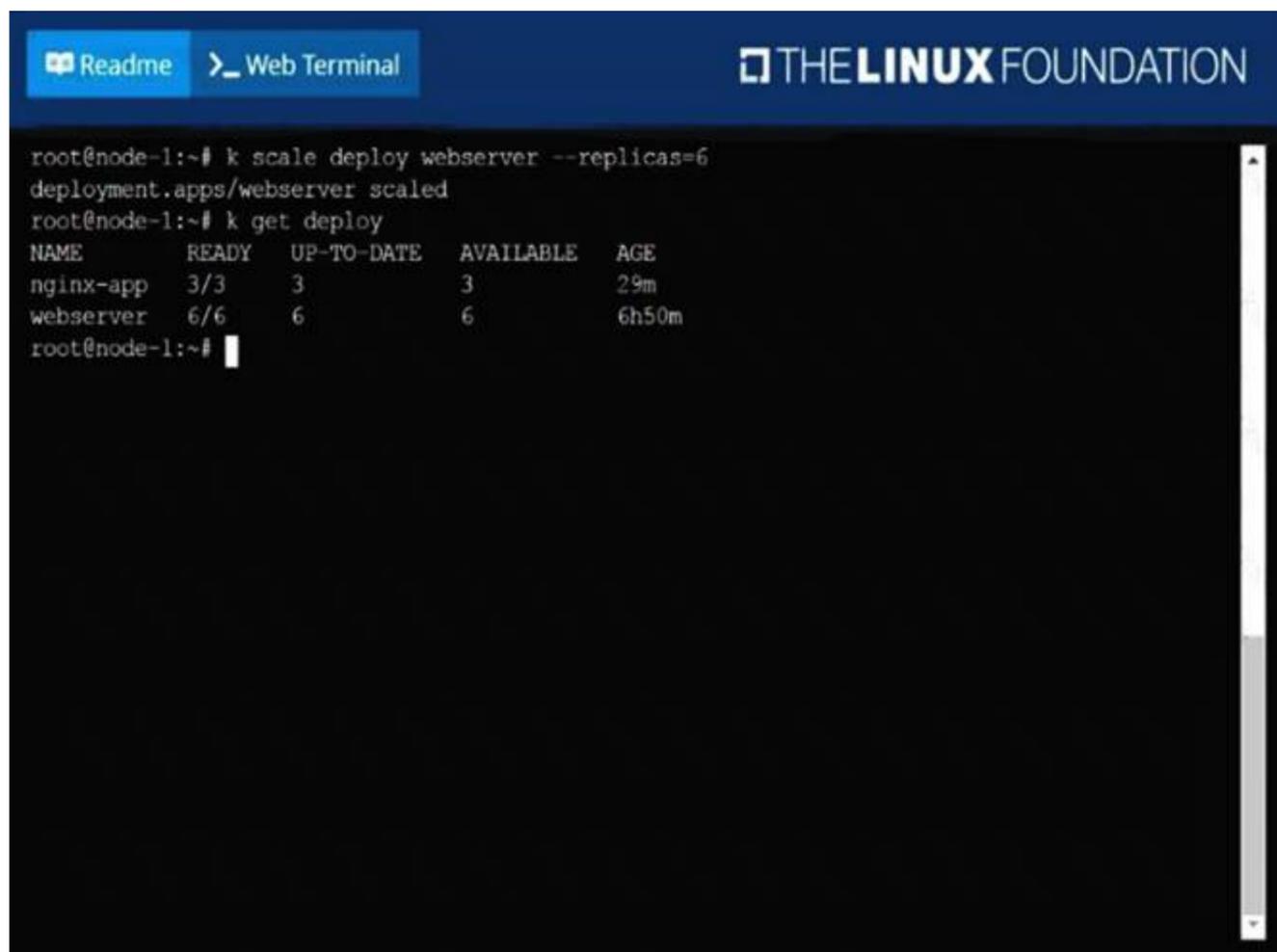
Scale the deployment webserver to 6 pods.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\14 B.JPG
```



```
root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app     3/3     3             3           29m
webserver     6/6     6             6           6h50m
root@node-1:~#
```

NEW QUESTION 11

Create a deployment as follows:

- > Name: nginx-random
- > Exposed via a service nginx-random
- > Ensure that the service & pod are accessible via their respective DNS records
- > The container(s) within any pod(s) running as a part of this deployment should use the nginx Image

Next, use the utility nslookup to look up the DNS records of the service & pod and write the output to /opt/KUNW00601/service.dns and /opt/KUNW00601/pod.dns respectively.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 C.JPG


```
Readme Web Terminal THE LINUX FOUNDATION

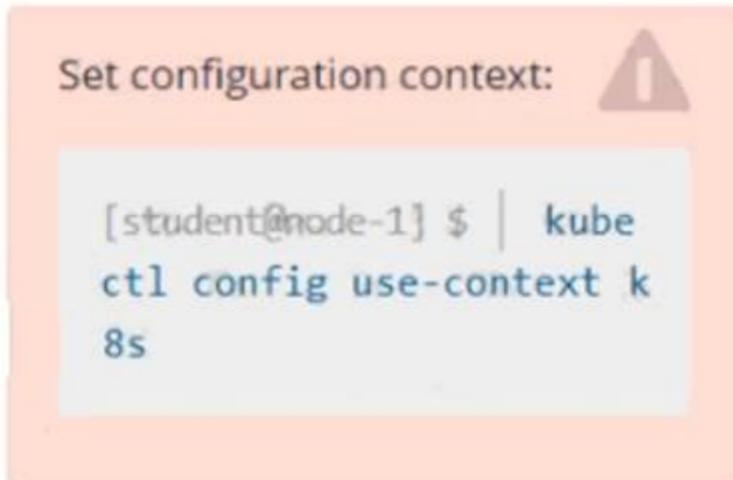
root@node-1:~# k create deploy nginx-random --image=nginx
deployment.apps/nginx-random created
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80
service/nginx-random exposed
root@node-1:~# vim dns.yaml
root@node-1:~# k create -f dns.yaml
pod/busybox1 created
root@node-1:~# k get po -o wide | grep nginx-random
nginx-random-6d5766bbdc-ptzv2 1/1 Running 0 103s 10.244.2.16 k8s-node-1 <none> <none>
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: nginx-random
Address 1: 10.111.37.132 nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random > /opt/KUNW00601/service.dns
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name: 10-244-2-16.default.pod
Address 1: 10.244.2.16 10-244-2-16.nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod > /opt/KUNW00601/pod.dns
```

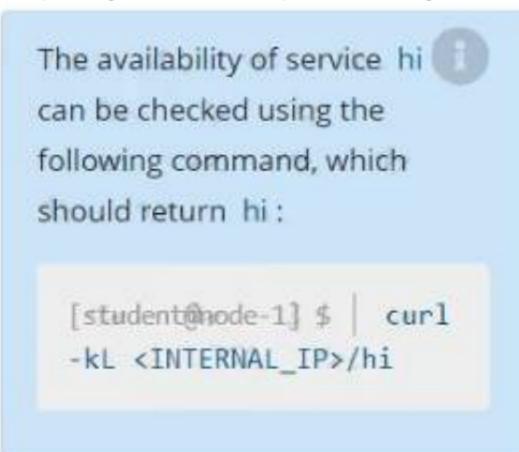
NEW QUESTION 12

Score: 7%

**Task**

Create a new nginx Ingress resource as follows:

- Name: ping
- Namespace: ing-internal
- Exposing service hi on path /hi using service port 5678



- A. Mastered
- B. Not Mastered

Answer: A**Explanation:**

Solution:

vi ingress.yaml

#

apiVersion: networking.k8s.io/v1 kind: Ingress

```
metadata: name: ping
namespace: ing-internal spec:
rules:
- http:
paths:
- path: /hi pathType: Prefix backend: service:
name: hi port:
number: 5678
#
kubectl create -f ingress.yaml
```

NEW QUESTION 15

Create a busybox pod and add "sleep 3600" command

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

```
kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c "sleep 3600"
```

NEW QUESTION 16

Set the node named ek8s-node-1 as unavailable and reschedule all the pods running on it.

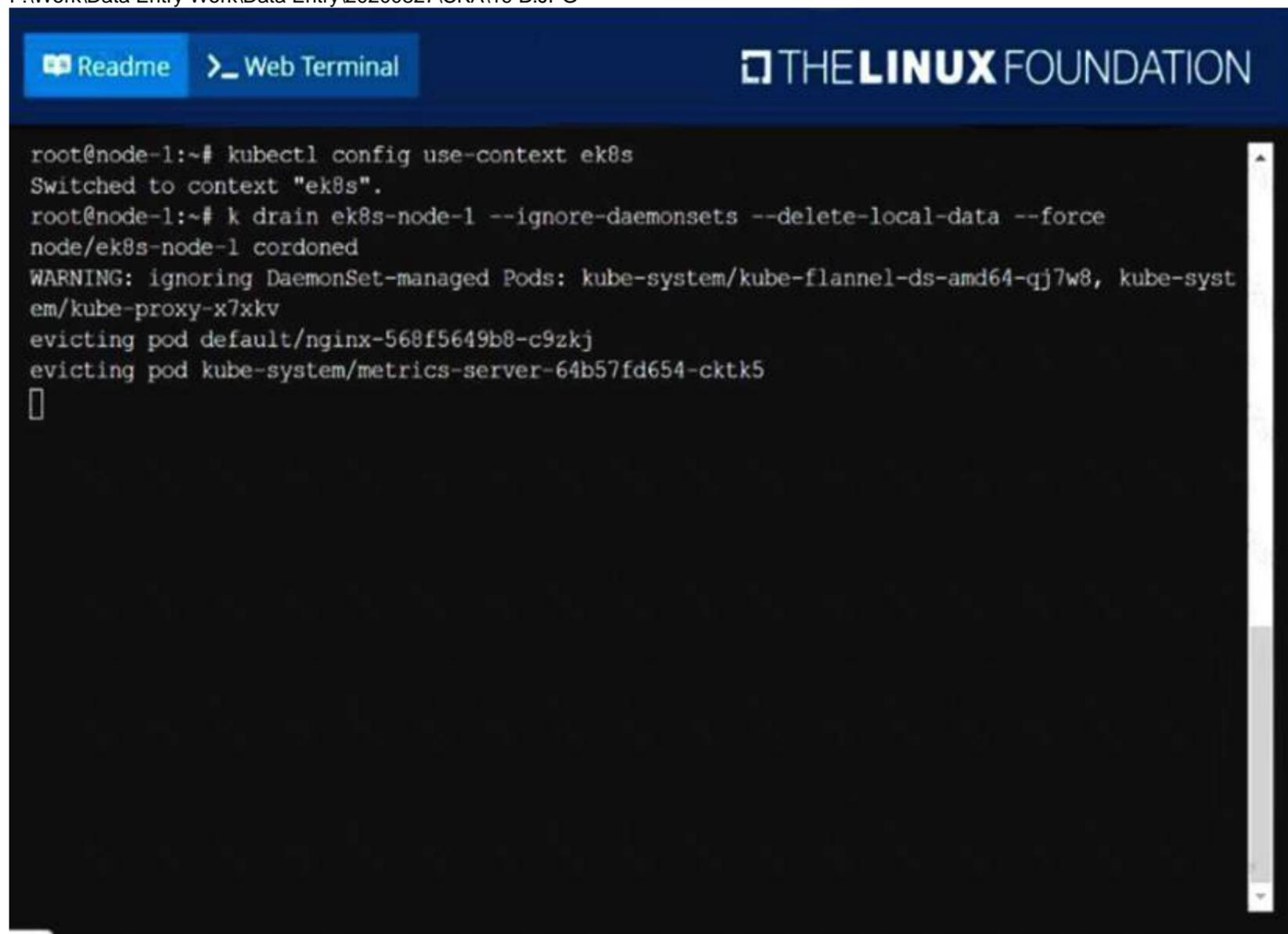
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\19 B.JPG



```
root@node-1:~# kubectl config use-context ek8s
Switched to context "ek8s".
root@node-1:~# k drain ek8s-node-1 --ignore-daemonsets --delete-local-data --force
node/ek8s-node-1 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-amd64-qj7w8, kube-system/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zky
evicting pod kube-system/metrics-server-64b57fd654-cktk5
[]
```

NEW QUESTION 17

Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place.

Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

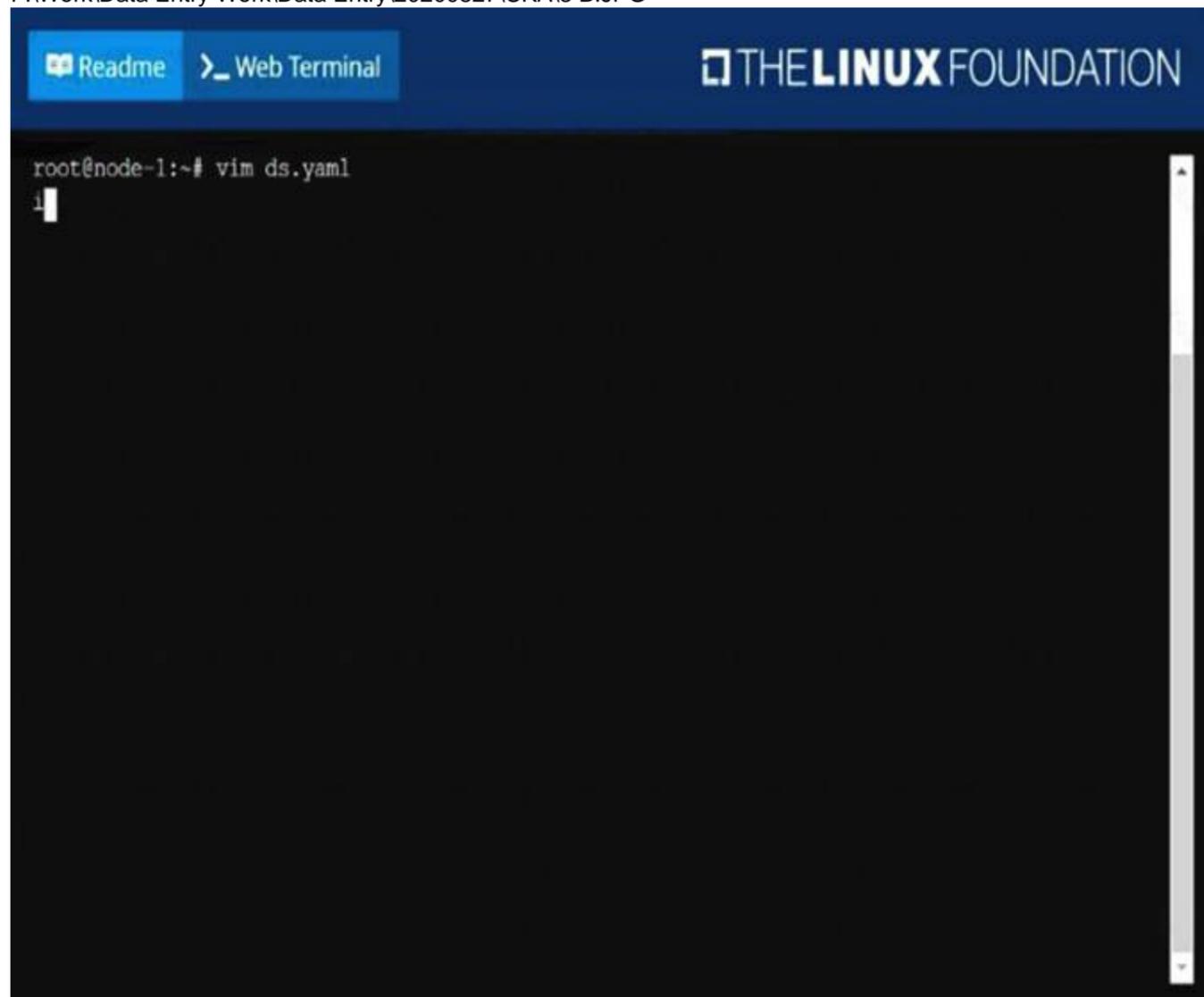
- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

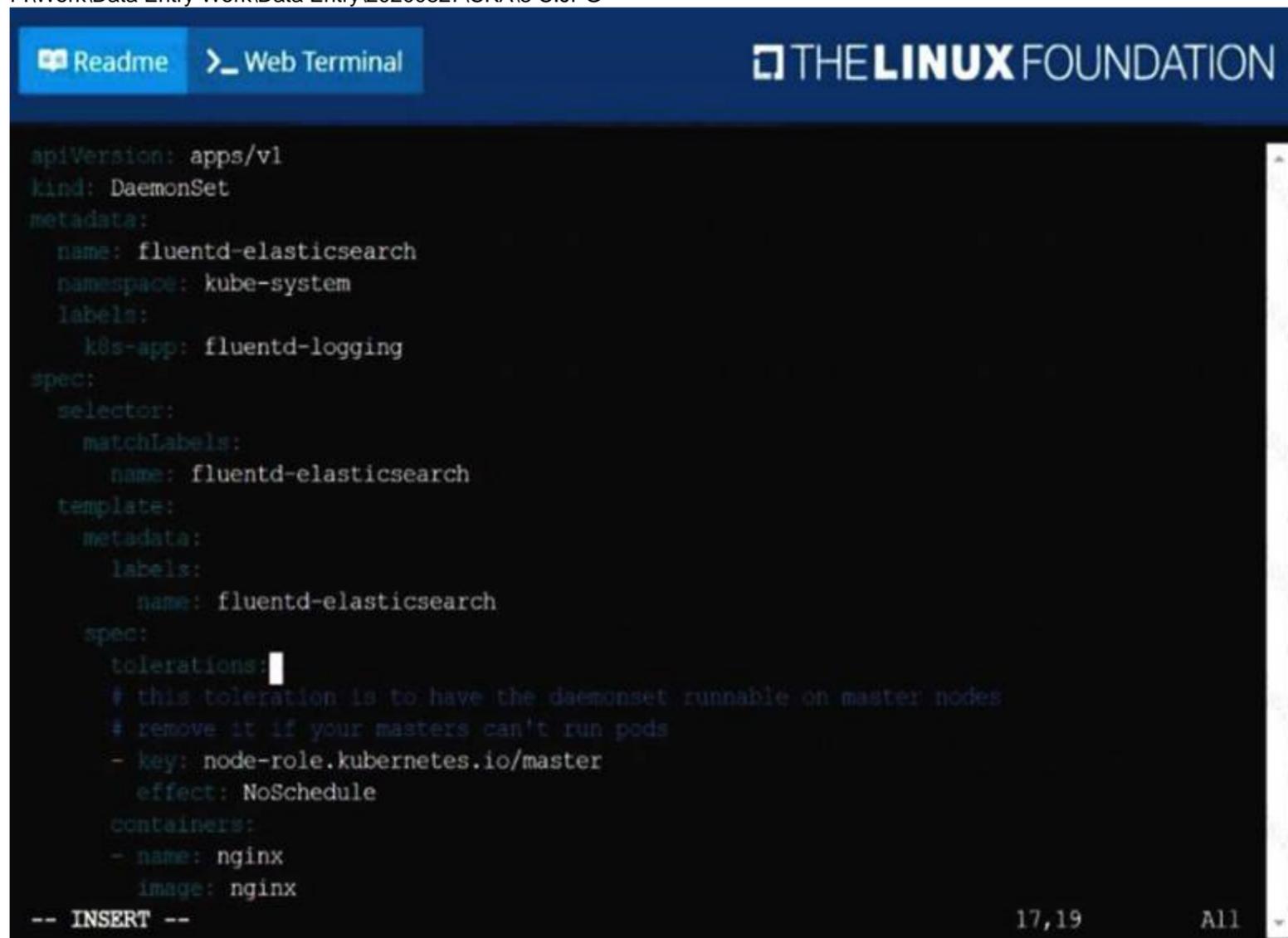
solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 B.JPG



```
root@node-1:~# vim ds.yaml
i
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 C.JPG



```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      tolerations:
        # this toleration is to have the daemonset runnable on master nodes
        # remove it if your masters can't run pods
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      containers:
        - name: nginx
          image: nginx
-- INSERT --
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 D.JPG

```

Readme Web Terminal THE LINUX FOUNDATION
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ds-kusc00201
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
      - name: nginx
        image: nginx
~
~
~
~
~
~
~
~
~
~
:WG

```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\3 E.JPG

```

Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201   2         2         2       2            2          <none>          4s
root@node-1:~#

```

NEW QUESTION 18

Print pod name and start time to "/opt/pod-status" file

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubect1 get pods -o=jsonpath='{range items[*]}{.metadata.name}\t{.status.podIP}\n}{end}'

NEW QUESTION 23

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany. The type of volume is hostPath and its location is /srv/app-data.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution
Persistent Volume
A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don't belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app on Kubernetes need not know the underlying infrastructure. When the developer needs a certain amount of persistent storage for their application, the system administrator configures the cluster so that they consume the PersistentVolume provisioned in an easy way.

Creating Persistent Volume

kind: PersistentVolume
apiVersion: v1
metadata: name: app-data
spec: capacity: # defines the capacity of PV we are creating storage: 2Gi #the amount of storage we are trying to claim accessModes: # defines the rights of the volume we are creating - ReadWriteMany hostPath: path: "/srv/app-data" # path to which we are creating the volume

Challenge

> Create a Persistent Volume named app-data, with access mode ReadWriteMany, storage classname shared, 2Gi of storage capacity and the host path /srv/app-data.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /srv/app-data
  storageClassName: shared
```

"app-data.yaml" 12L, 194C

* 2. Save the file and create the persistent volume. Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl create -f pv.yaml
persistentvolume/pv created
```

* 3. View the persistent volume.

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
app-data	2Gi	RWX	Retain	Available		shared		31s

> Our persistent volume status is available meaning it is available and it has not been mounted yet. This status will change when we mount the persistentVolume to a persistentVolumeClaim.

PersistentVolumeClaim

In a real ecosystem, a system admin will create the PersistentVolume then a developer will create a PersistentVolumeClaim which will be referenced in a pod. A PersistentVolumeClaim is created by specifying the minimum size and the access mode they require from the persistentVolume.

Challenge

> Create a Persistent Volume Claim that requests the Persistent Volume we had created above. The claim should request 2Gi. Ensure that the Persistent

Volume Claim has the same storageClassName as the persistentVolume you had previously created.

kind: PersistentVolumeapiVersion: v1metadata: name:app-data spec:

accessModes: - ReadWriteMany resources:

requests: storage: 2Gi

storageClassName: shared

* 2. Save and create the pvc

njerry191@cloudshell:~ (extreme-clone-265411)\$ kubectl create -f app-data.yaml persistentvolumeclaim/app-data created

* 3. View the pvc Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pvc
NAME      STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS
pv        Bound   pv       512m      RWX            shared
```

* 4. Let's see what has changed in the pv we had initially created.

Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411)$ kubectl get pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM      STORAGECLASS  REASON  AGE
pv        512m      RWX            Retain          Bound   default/pv  shared        16m
```

Our status has now changed from available to bound.

* 5. Create a new pod named myapp with image nginx that will be used to Mount the Persistent Volume Claim with the path /var/app/config.

Mounting a Claim

apiVersion: v1kind: Podmetadata: creationTimestamp: null name: app-dataspec: volumes: - name:congigpvc persistentVolumeClaim: claimName: app-data

containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data " name: configpvc

NEW QUESTION 24

Create a pod with image nginx called nginx and allow traffic on port 80

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

kubectl run nginx --image=nginx --restart=Never --port=80

NEW QUESTION 25

From the pod label name=cpu-utilizer, find pods running high CPU workloads and write the name of the pod consuming most CPU to the file /opt/KUTR00102/KUTR00102.txt (which already exists).

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\16 B.JPG



Context

You have been asked to create a new ClusterRole for a deployment pipeline and bind it to a specific ServiceAccount scoped to a specific namespace.

Task

Create a new ClusterRole named deployment-clusterrole, which only allows to create the following resource types:

- Deployment
- StatefulSet
- DaemonSet

Create a new ServiceAccount named cicd-token in the existing namespace app-team1.

Bind the new ClusterRole deployment-clusterrole to the new ServiceAccount cicd-token, limited to the namespace app-team1.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Task should be complete on node k8s -1 master, 2 worker for this connect use command

```
[student@node-1] > ssh k8s
```

```
kubectl create clusterrole deployment-clusterrole --verb=create
```

```
--resource=deployments,statefulsets,daemonsets
```

```
kubectl create serviceaccount cicd-token --namespace=app-team1
```

```
kubectl create rolebinding deployment-clusterrole --clusterrole=deployment-clusterrole
```

```
--serviceaccount=default:cicd-token --namespace=app-team1
```

NEW QUESTION 30

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CKA Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CKA Product From:

<https://www.2passeasy.com/dumps/CKA/>

Money Back Guarantee

CKA Practice Exam Features:

- * CKA Questions and Answers Updated Frequently
- * CKA Practice Questions Verified by Expert Senior Certified Staff
- * CKA Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * CKA Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year