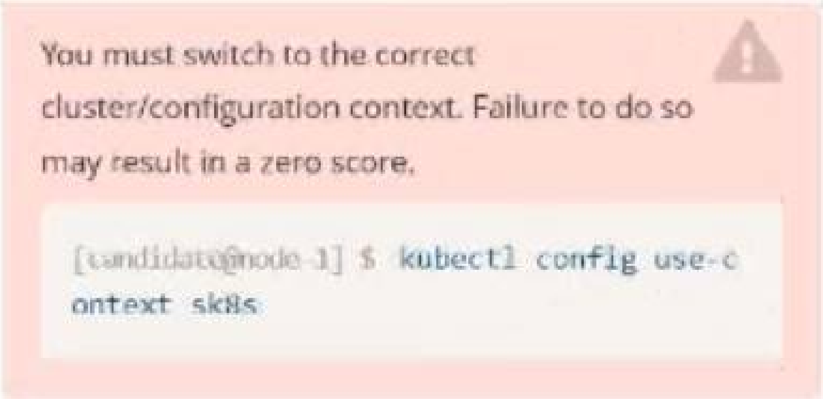
Exam Questions CKAD

Certified Kubernetes Application Developer (CKAD) Program

https://www.2passeasy.com/dumps/CKAD/

NEW QUESTION 1
Exhibit:



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

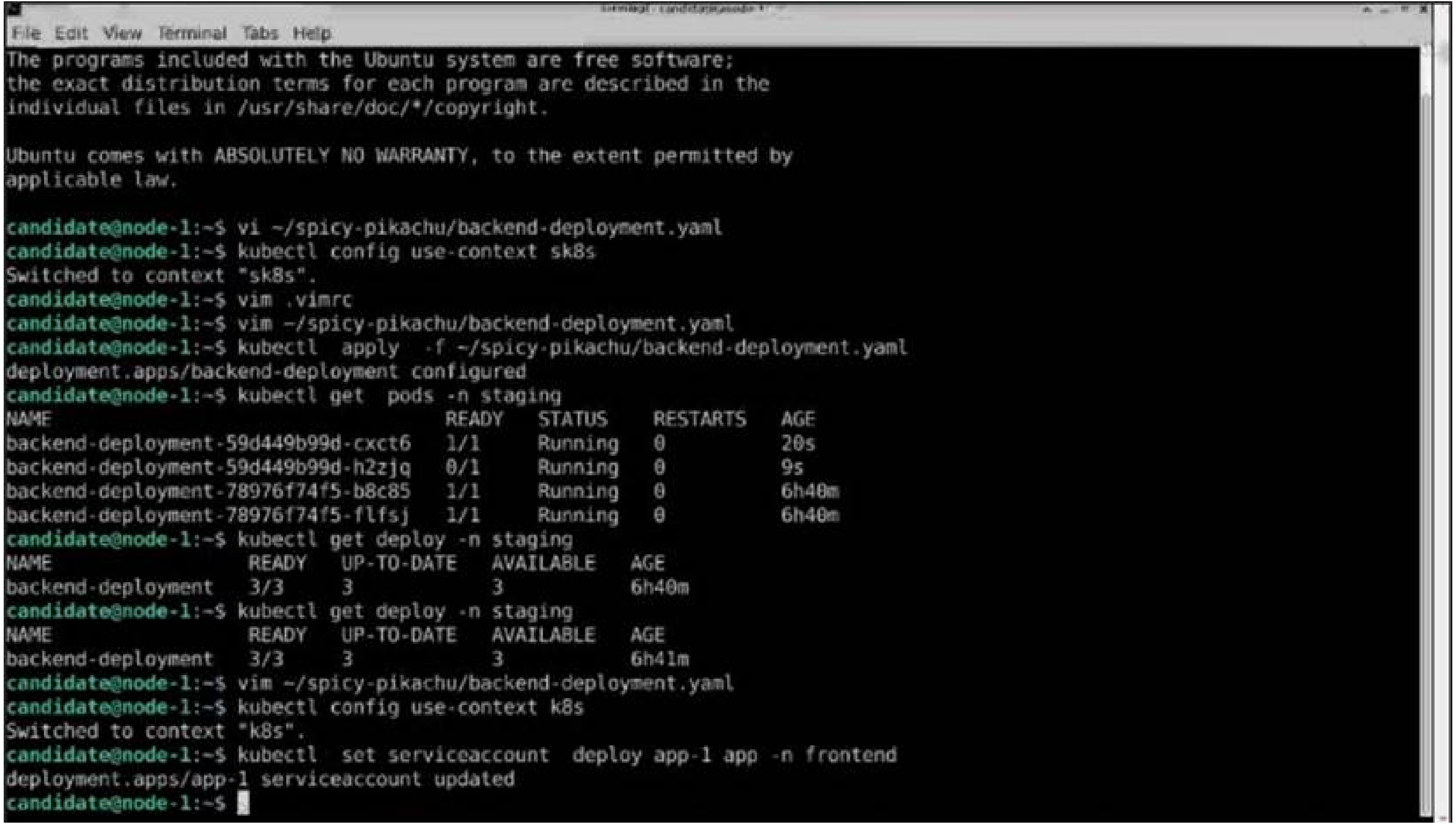
```
[candidate@node-1] $ kubectl config use-context sk8s
```

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

A. Mastered
B. Not Mastered

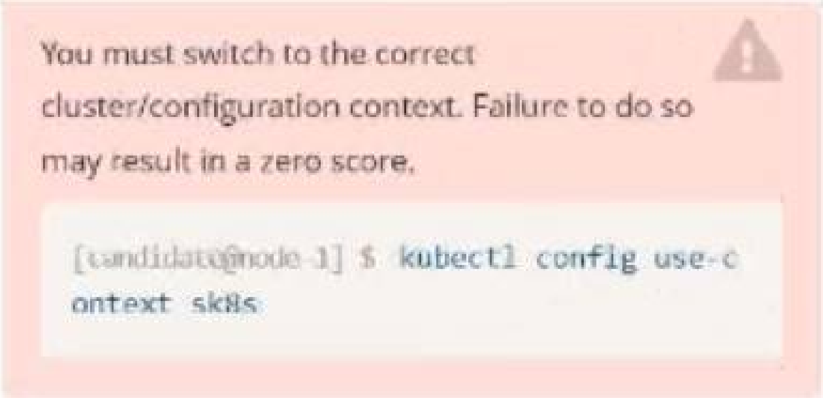
Answer: A

Explanation:
Solution:
Text Description automatically generated



```
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:~$
```

NEW QUESTION 2
Exhibit:



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

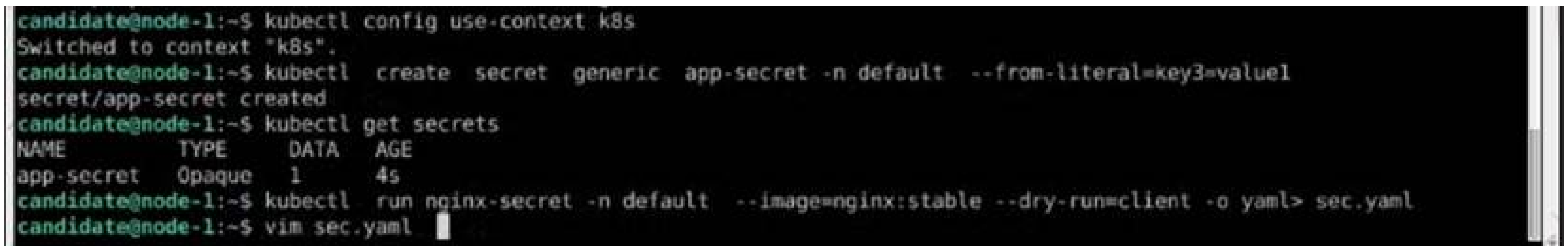
```
[candidate@node-1] $ kubectl config use-context sk8s
```

Task:
Key3: value1
Add an environment variable named BEST_VARIABLE consuming the value of the secret key3.

A. Mastered
B. Not Mastered

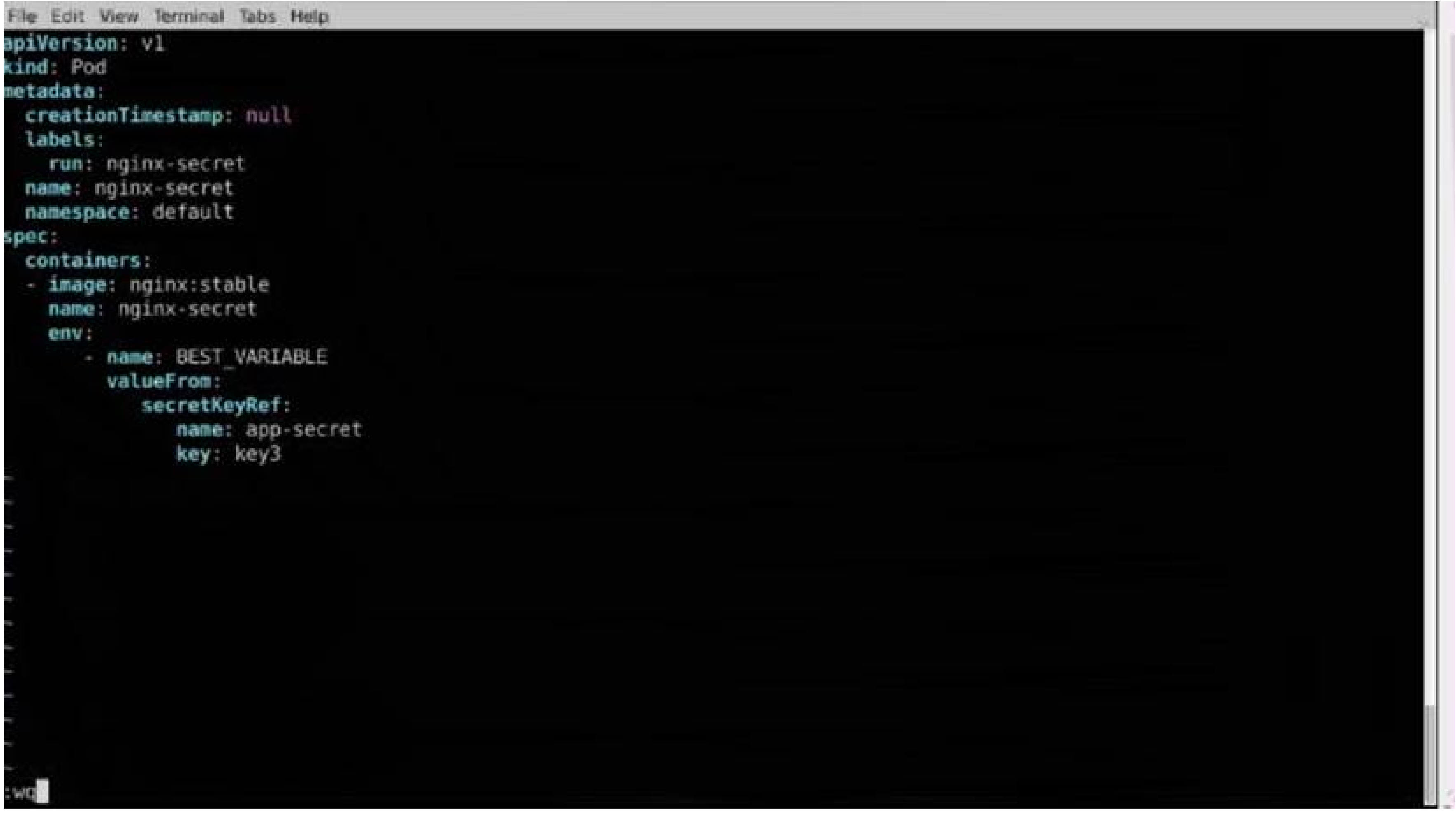
Answer: A

Explanation:
Solution:



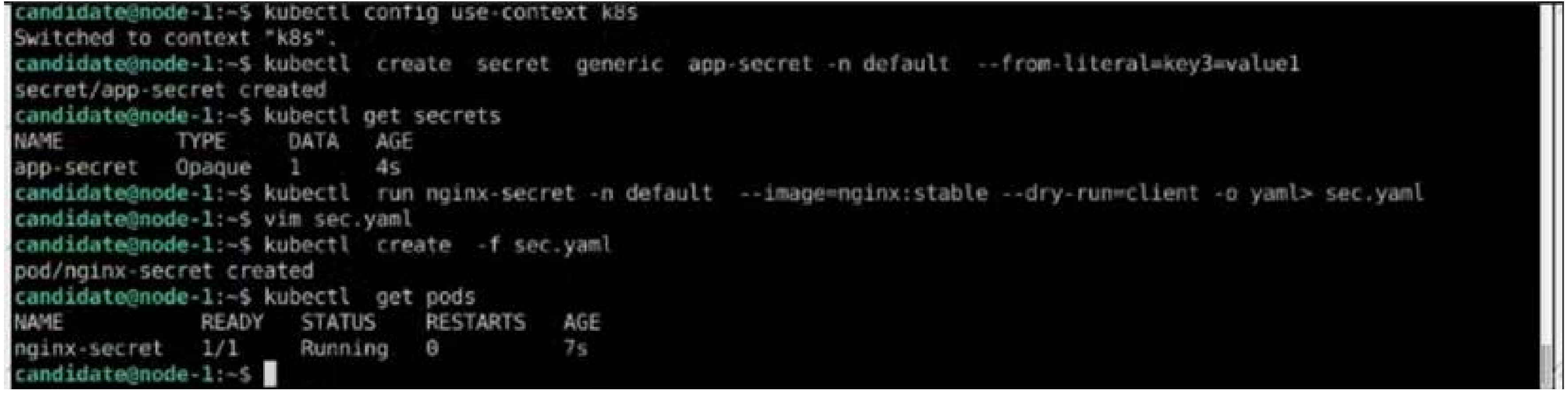
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  secret  generic  app-secret -n default  --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME         TYPE     DATA   AGE
app-secret   Opaque   1      4s
candidate@node-1:~$ kubectl  run nginx-secret -n default  --image=nginx:stable --dry-run=client -o yaml> sec.yaml
candidate@node-1:~$ vim sec.yaml
```

Text Description automatically generated



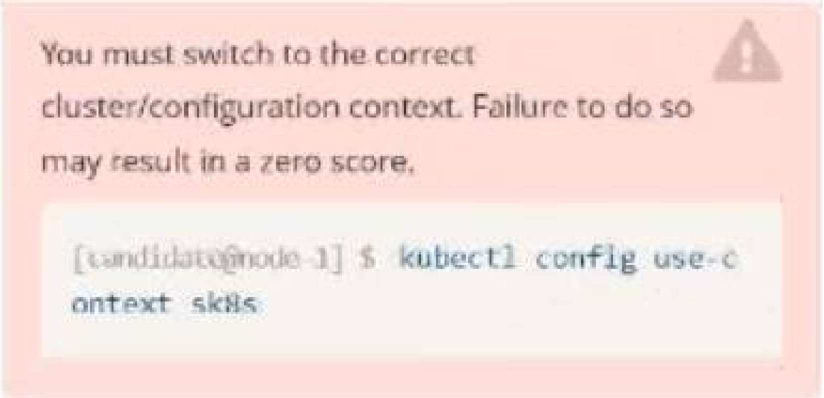
```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
  namespace: default
spec:
  containers:
  - image: nginx:stable
    name: nginx-secret
    env:
       - name: BEST_VARIABLE
         valueFrom:
            secretKeyRef:
               name: app-secret
               key: key3
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Text Description automatically generated



```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  secret  generic  app-secret -n default  --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME         TYPE     DATA   AGE
app-secret   Opaque   1      4s
candidate@node-1:~$ kubectl  run nginx-secret -n default  --image=nginx:stable --dry-run=client -o yaml> sec.yaml
candidate@node-1:~$ vim sec.yaml
candidate@node-1:~$ kubectl  create  -f sec.yaml
pod/nginx-secret created
candidate@node-1:~$ kubectl  get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-secret   1/1     Running   0          7s
candidate@node-1:~$
```

NEW QUESTION 3
Exhibit:



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-c
ontext sk8s
```

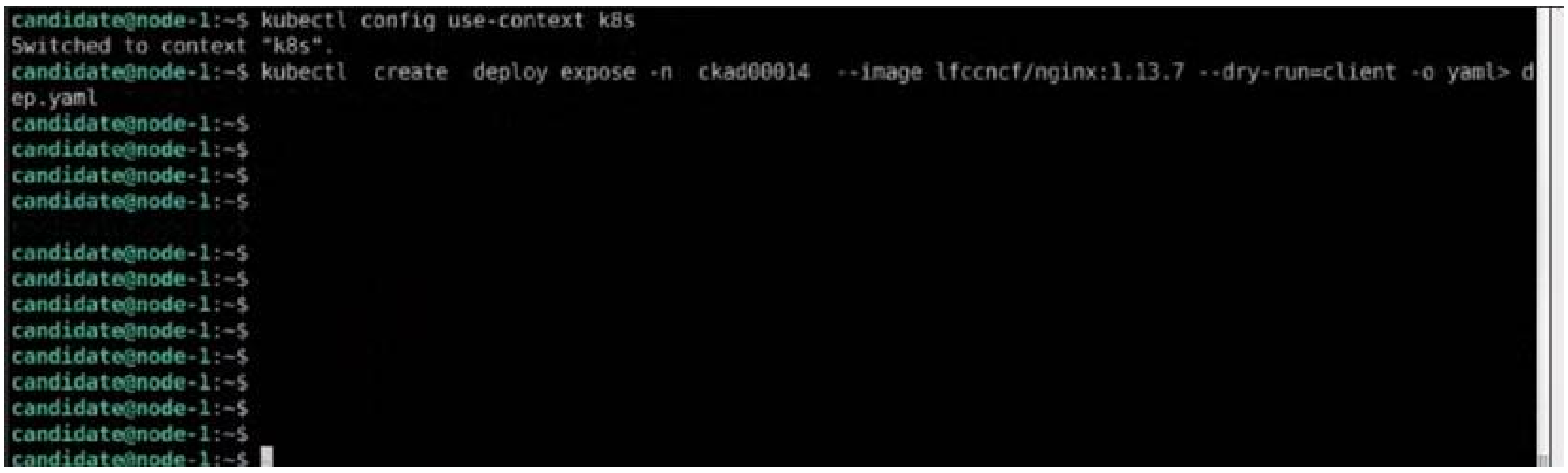
Task:
Create a Deployment named expose in the existing ckad00014 namespace running 6 replicas of a Pod. Specify a single container using the ifccncf/nginx: 1.13.7 image
Add an environment variable named NGINX_PORT with the value 8001 to the container then expose port 8001

A. Mastered
B. Not Mastered

Answer: A

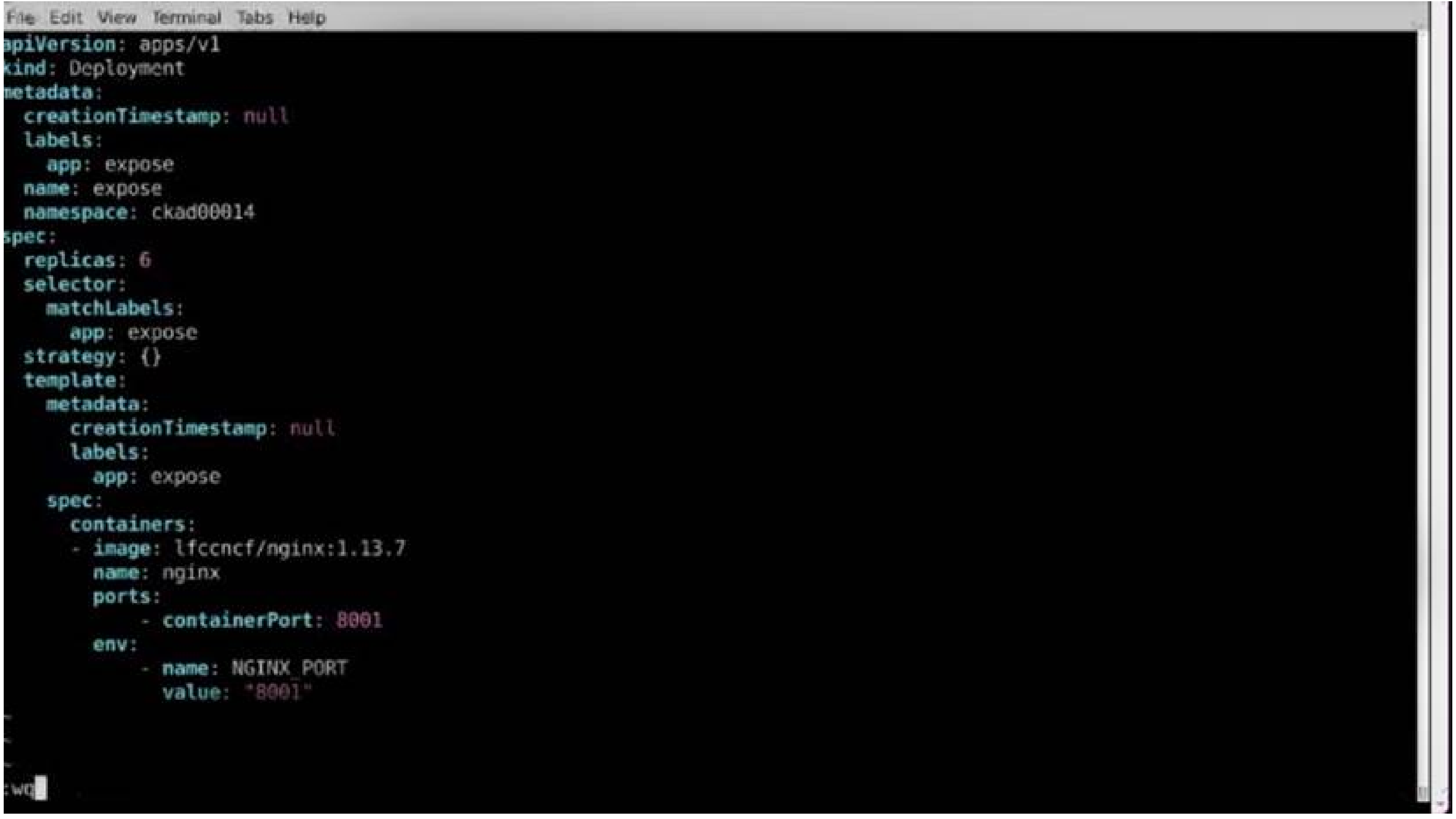
Explanation:

Solution:



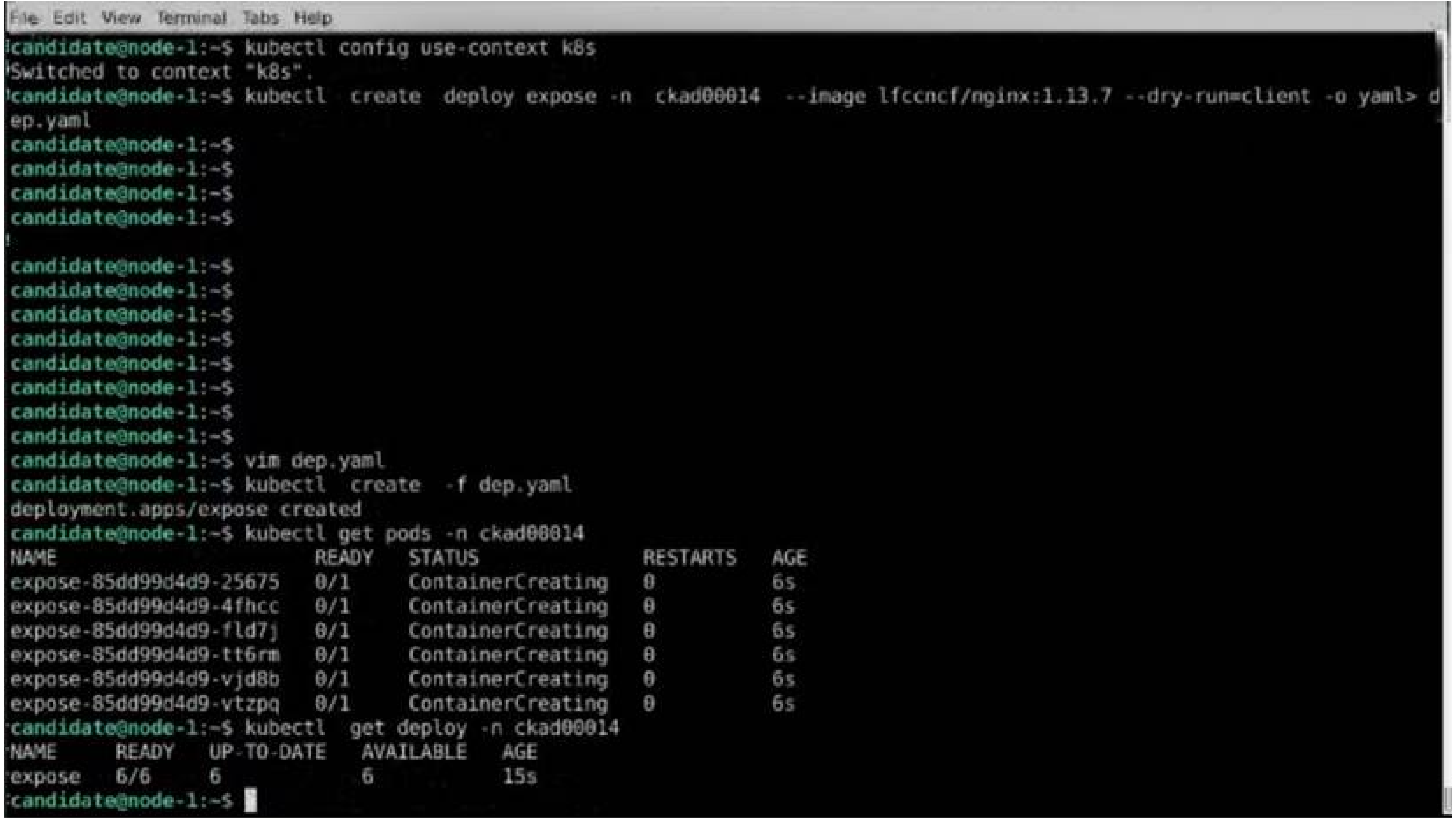
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  deploy expose -n  ckad00014  --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml> d
ep.yaml
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$

candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
```

Text Description automatically generated



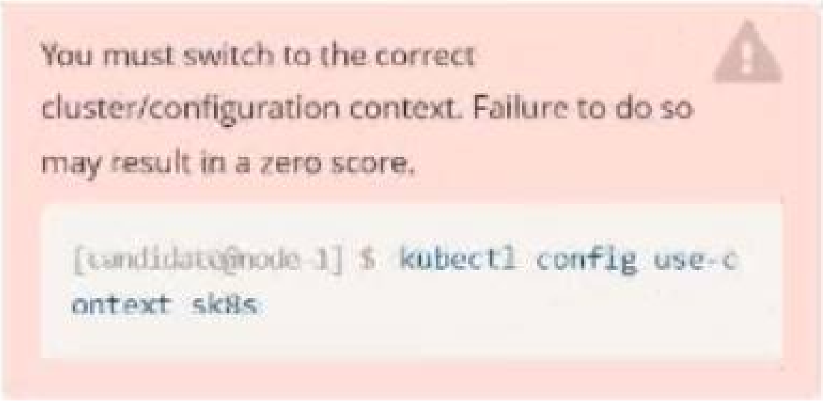
```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: expose
  name: expose
  namespace: ckad00014
spec:
  replicas: 6
  selector:
    matchLabels:
      app: expose
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: expose
    spec:
      containers:
      - image: lfccncf/nginx:1.13.7
        name: nginx
        ports:
          - containerPort: 8001
        env:
          - name: NGINX_PORT
            value: "8001"
~
~

:wq
```

Text Description automatically generated



```
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> d
ep.yaml
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$

candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$ vim dep.yaml
candidate@node-1:~$ kubectl  create  -f dep.yaml
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:~$
```

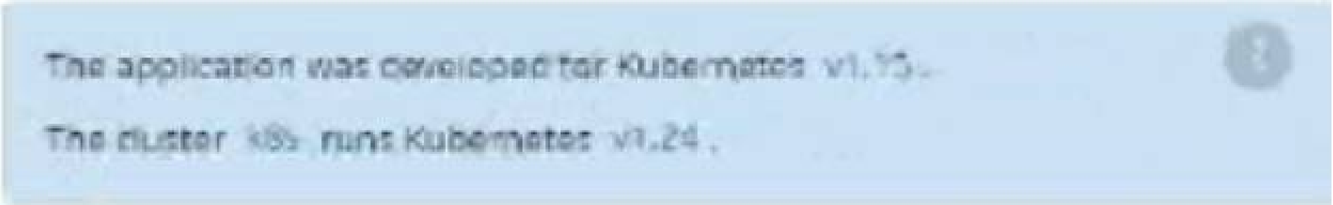
NEW QUESTION 4
Exhibit:



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-c
ontext sk8s
```

Task:



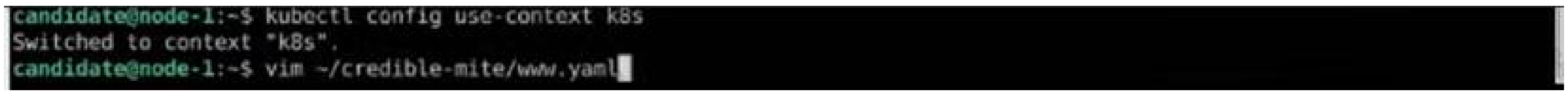
The application was developed for Kubernetes v1.15.

The cluster k8s runs Kubernetes v1.24.

A. Mastered
B. Not Mastered

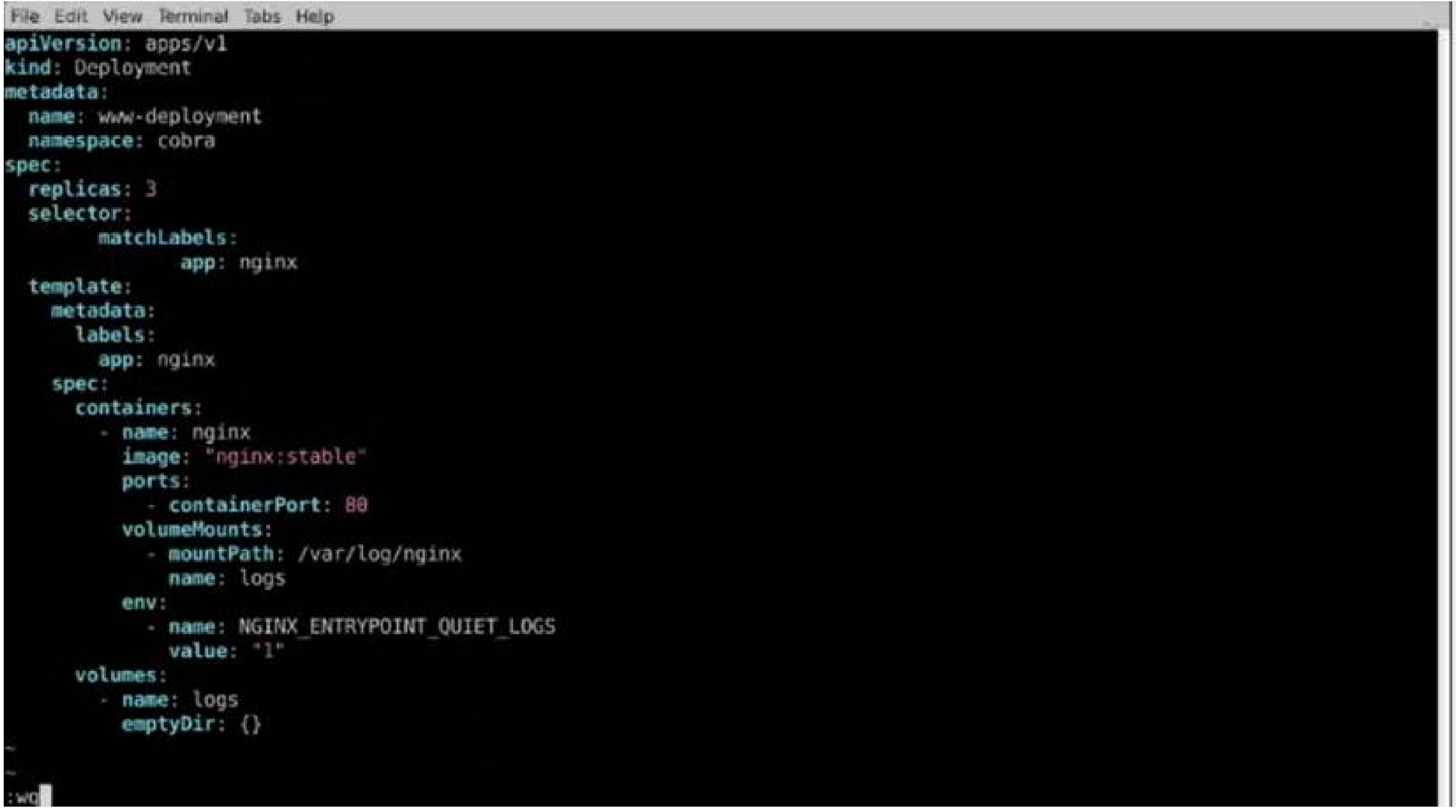
Answer: A

Explanation:
Solution:



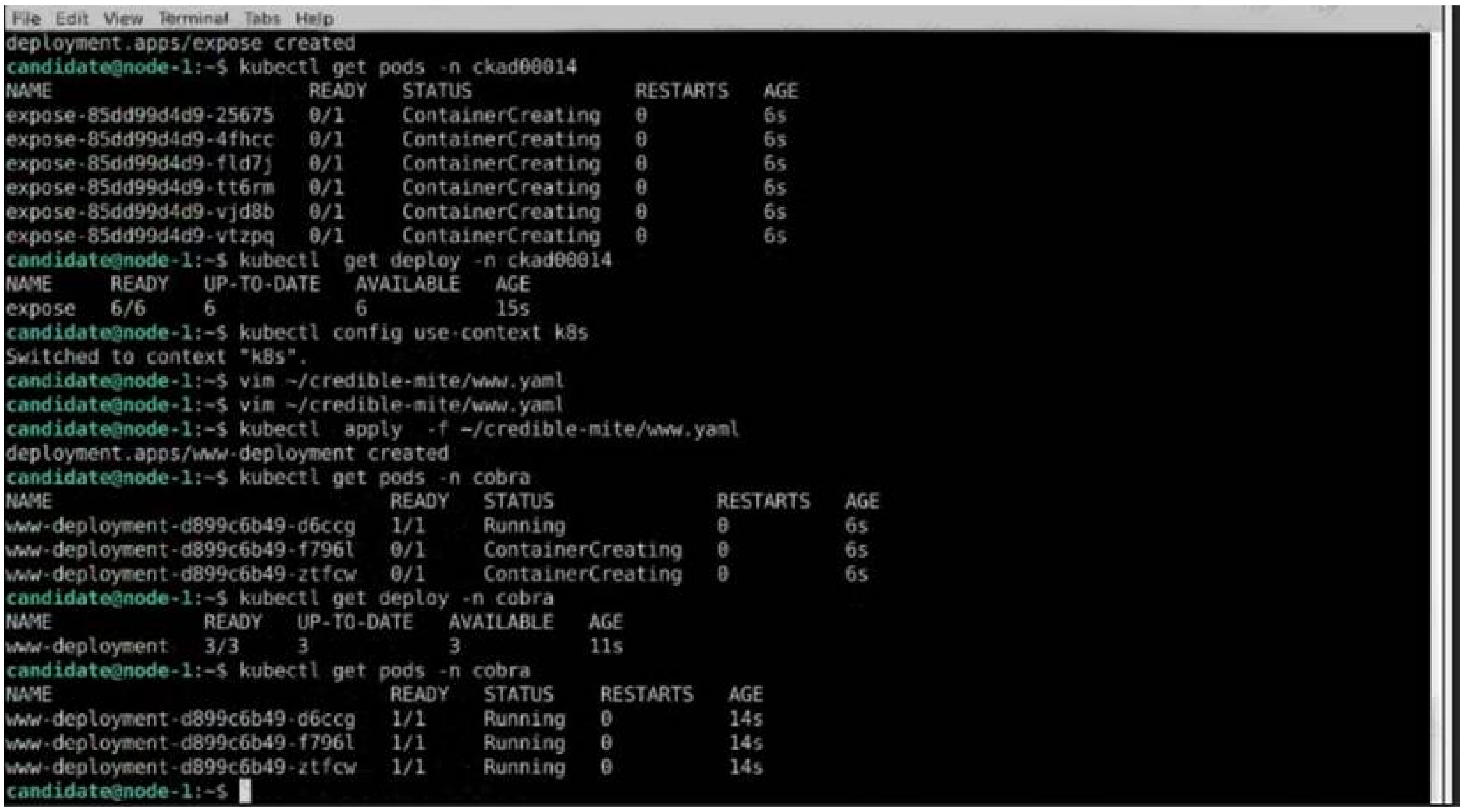
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
```

Text Description automatically generated



```
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: {}
~
~
:wq
```

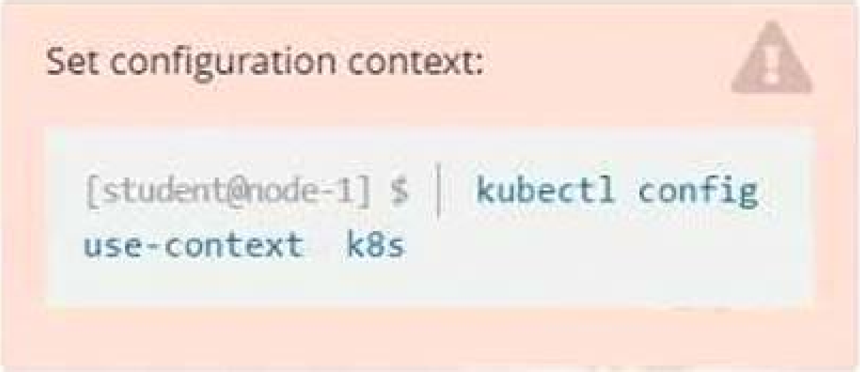
Text Description automatically generated



```
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:~$ 
```

NEW QUESTION 5

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 Ifccncf/redis image with the 3.2 tag
- Expose port 6379

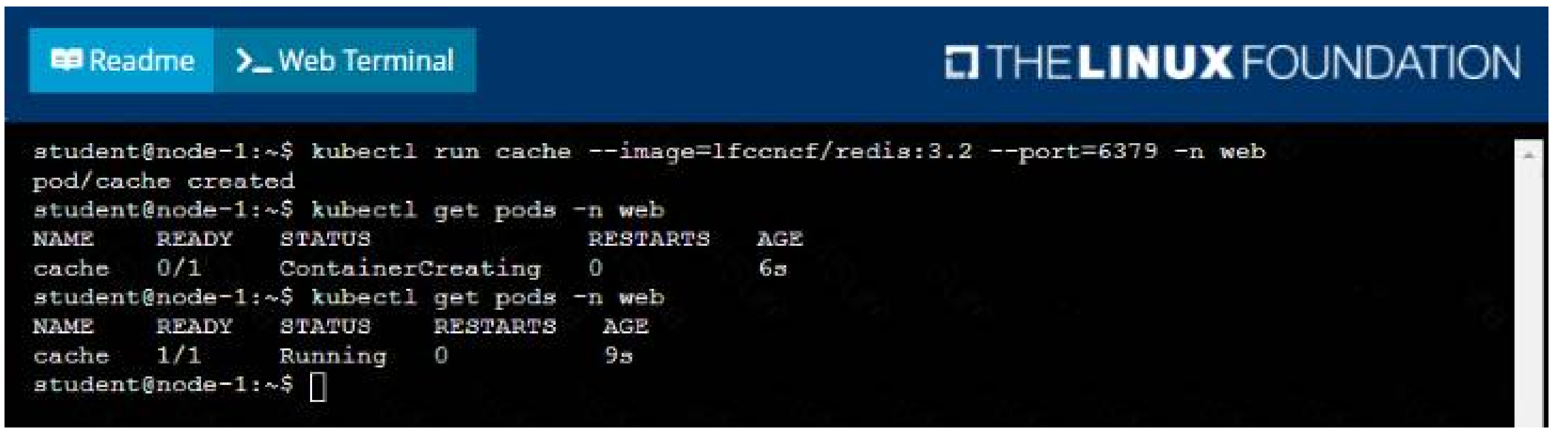
A. Mastered

B. Not Mastered

Answer: A

Explanation:

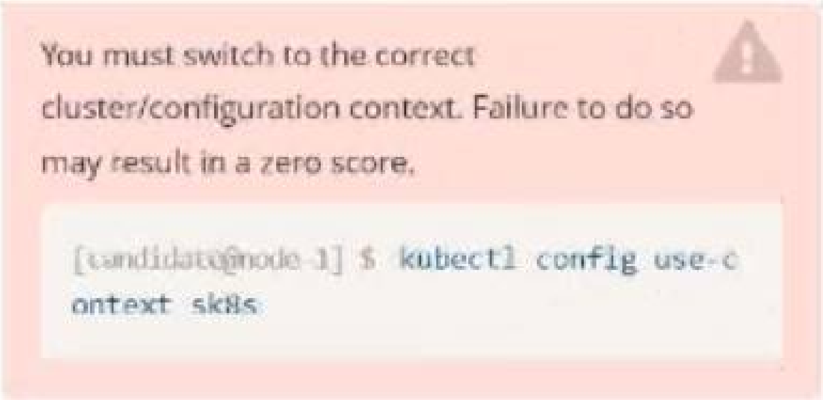
Solution:



```
student@node-1:~$ kubectl run cache --image=lfccncf/redis:3.2 --port=6379 -n web
pod/cache created
student@node-1:~$ kubectl get pods -n web
NAME    READY   STATUS              RESTARTS   AGE
cache   0/1     ContainerCreating   0          6s
student@node-1:~$ kubectl get pods -n web
NAME    READY   STATUS    RESTARTS   AGE
cache   1/1     Running   0          9s
student@node-1:~$ 
```

NEW QUESTION 6

Exhibit:



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

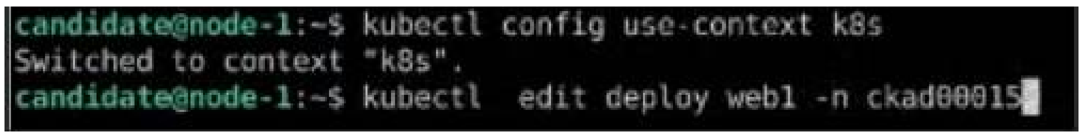
```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

Task:
1- Update the Propertunel scaling configuration of the Deployment web1 in the ckad00015 namespace setting maxSurge to 2 and maxUnavailable to 59
2- Update the web1 Deployment to use version tag 1.13.7 for the Ifconf/nginx container image. 3- Perform a rollback of the web1 Deployment to its previous version

A. Mastered
B. Not Mastered

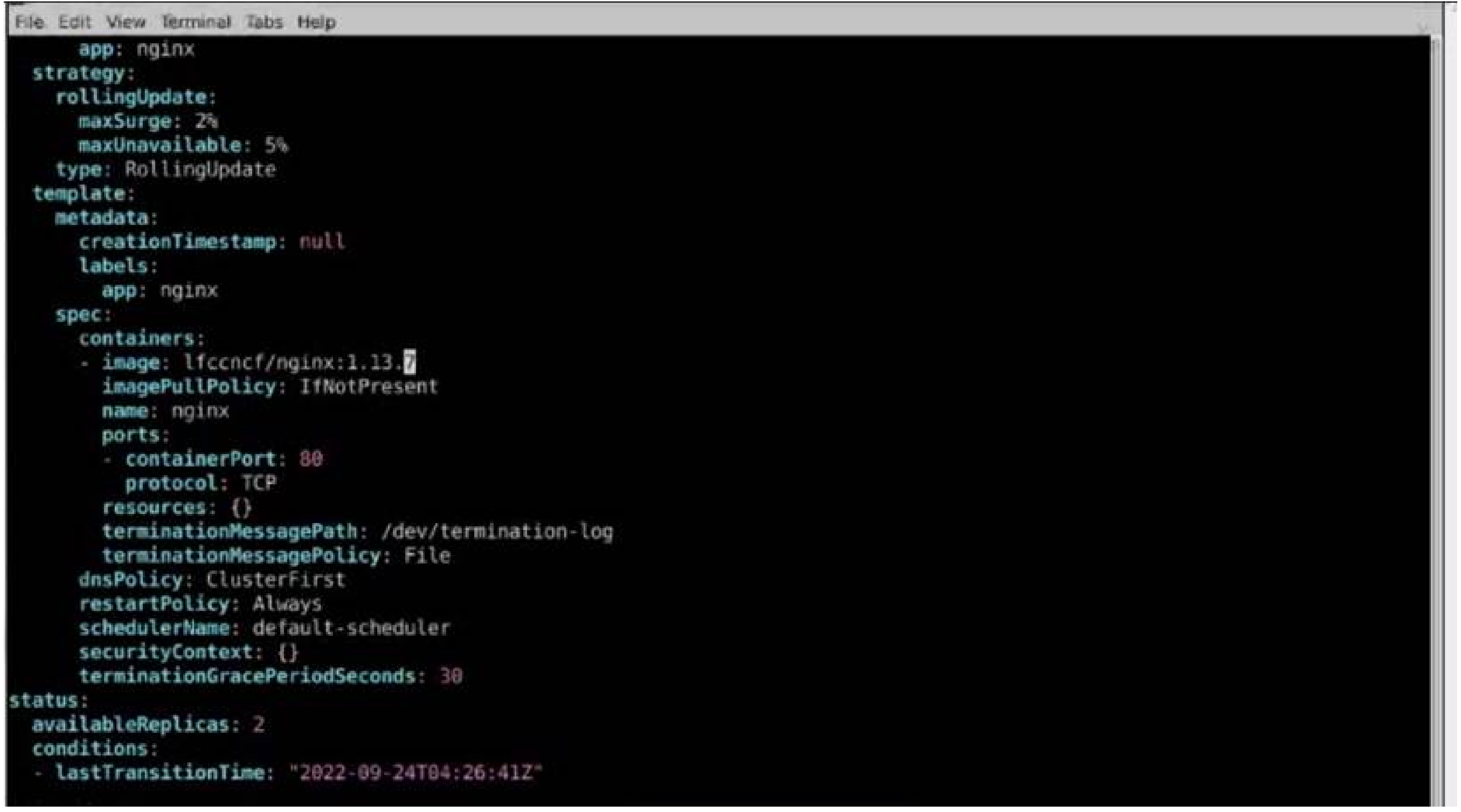
Answer: A

Explanation:
Solution:

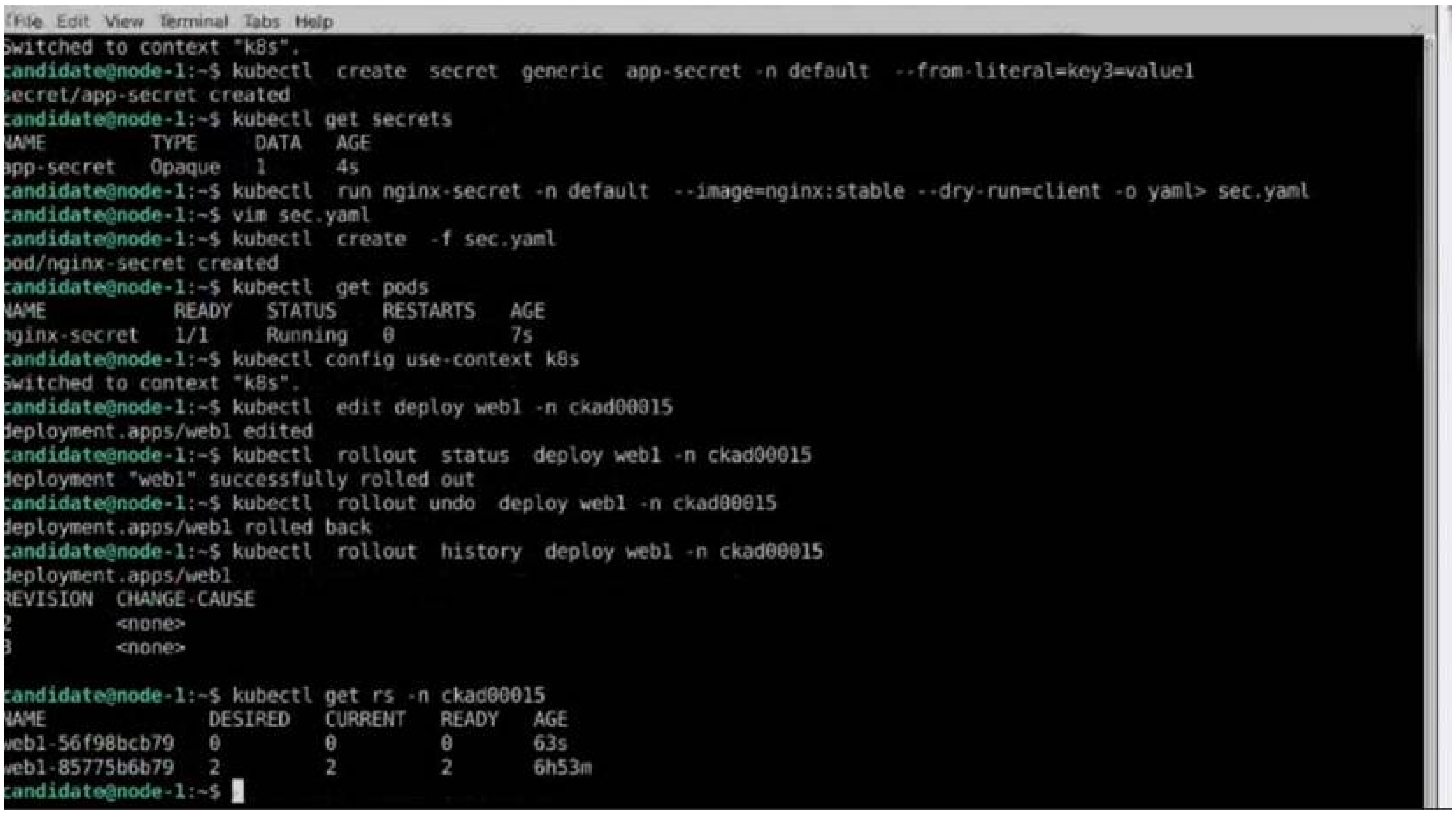


```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  edit deploy web1 -n ckad00015
```

Text Description automatically generated

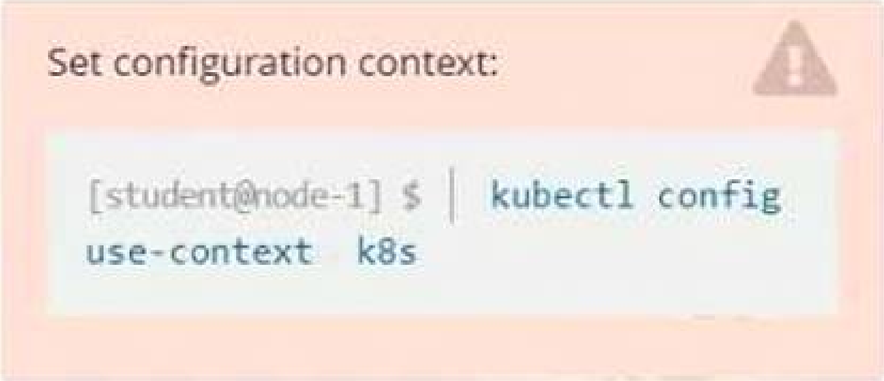


```
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 2%
      maxUnavailable: 5%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: lfccncf/nginx:1.13.7
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
        - containerPort: 80
          protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
status:
  availableReplicas: 2
  conditions:
  - lastTransitionTime: "2022-09-24T04:26:41Z"
```



```
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  secret  generic  app-secret -n default  --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME         TYPE      DATA    AGE
app-secret   Opaque    1       4s
candidate@node-1:~$ kubectl  run nginx-secret -n default  --image=nginx:stable --dry-run=client -o yaml> sec.yaml
candidate@node-1:~$ vim sec.yaml
candidate@node-1:~$ kubectl  create  -f sec.yaml
pod/nginx-secret created
candidate@node-1:~$ kubectl  get pods
NAME           READY   STATUS     RESTARTS   AGE
nginx-secret   1/1     Running    0          7s
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  edit deploy web1 -n ckad00015
deployment.apps/web1 edited
candidate@node-1:~$ kubectl  rollout  status  deploy web1 -n ckad00015
deployment "web1" successfully rolled out
candidate@node-1:~$ kubectl  rollout undo  deploy web1 -n ckad00015
deployment.apps/web1 rolled back
candidate@node-1:~$ kubectl  rollout  history  deploy web1 -n ckad00015
deployment.apps/web1
REVISION  CHANGE-CAUSE
2         <none>
3         <none>

candidate@node-1:~$ kubectl get rs -n ckad00015
NAME                DESIRED   CURRENT   READY   AGE
web1-56f98bcb79     0         0         0       63s
web1-85775b6b79     2         2         2       6h53m
candidate@node-1:~$ 
```

NEW QUESTION 7
Exhibit:



Set configuration context:

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

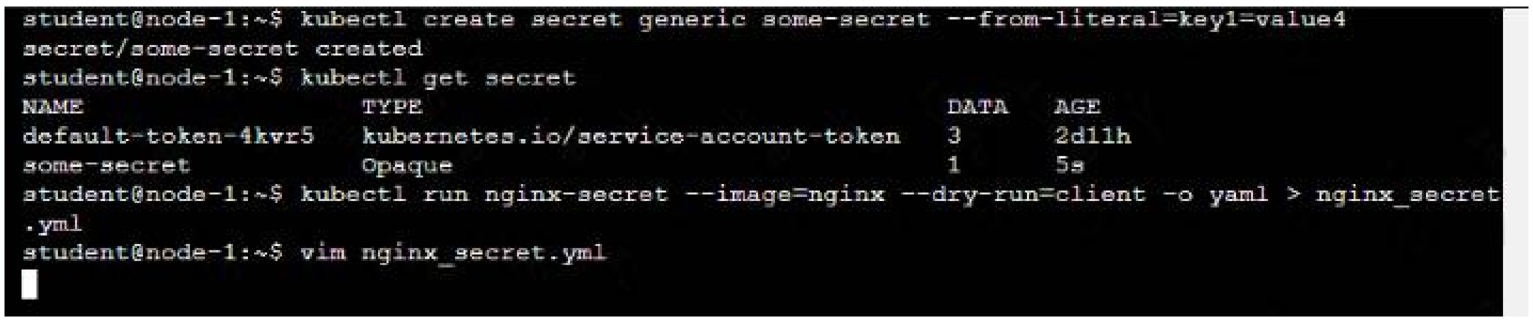
Context
You are tasked to create a secret and consume the secret in a pod using environment variables as follow:
Task
• Create a secret named another-secret with a key/value pair; key1/value4
• Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key 1, using COOL_VARIABLE as the name for the environment variable inside the pod

A. Mastered
B. Not Mastered

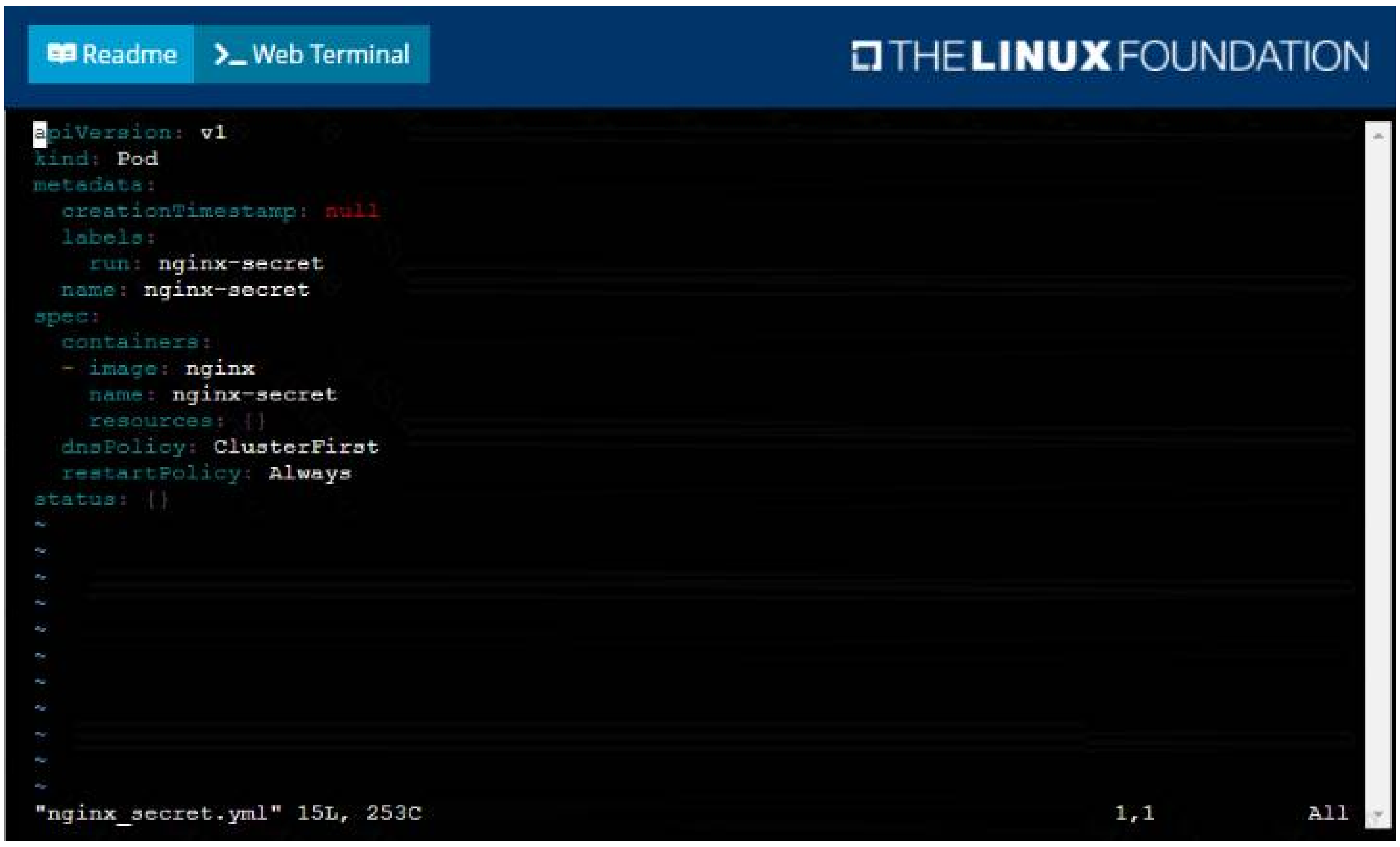
Answer: A

Explanation:

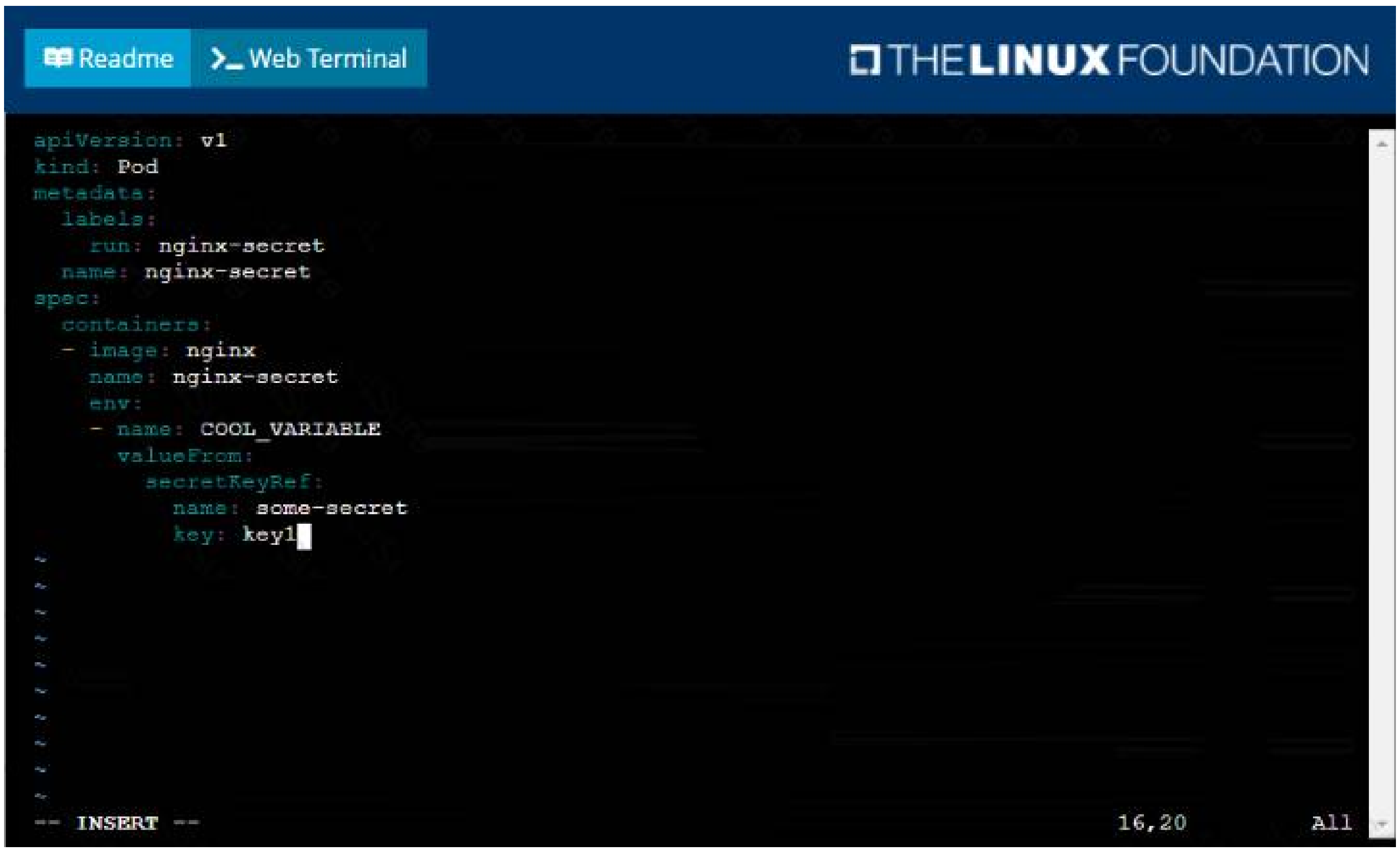
Solution:



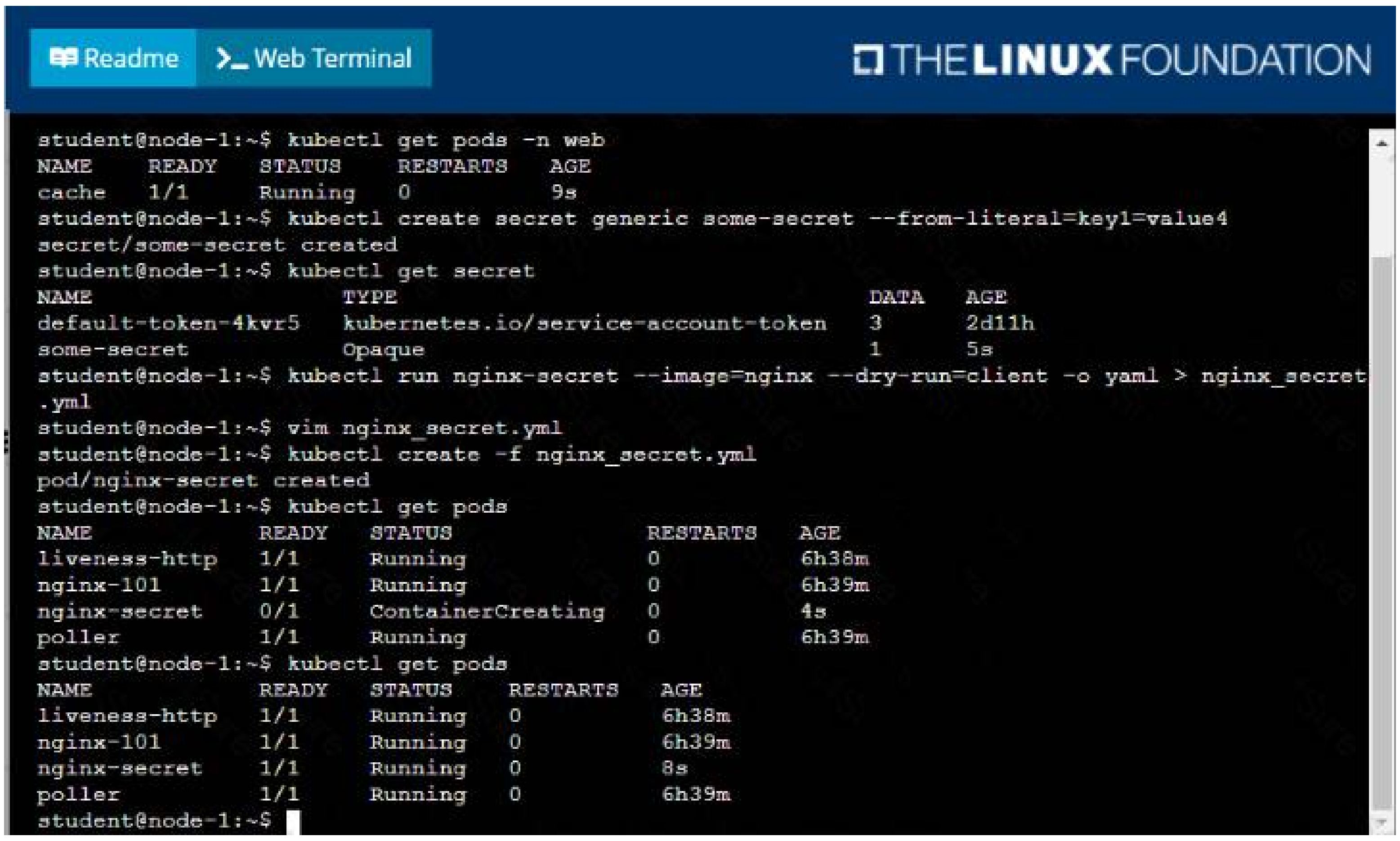
```
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-4kvr5   kubernetes.io/service-account-token   3      2d11h
some-secret           Opaque                                1      5s
student@node-1:~$ kubectl run nginx-secret --image=nginx --dry-run=client -o yaml > nginx_secret
.yml
student@node-1:~$ vim nginx_secret.yml
```



```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"nginx_secret.yml" 15L, 253C                                              1,1           All
```

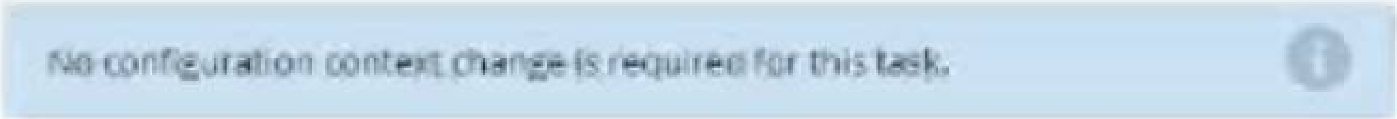


```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    env:
    - name: COOL_VARIABLE
      valueFrom:
        secretKeyRef:
          name: some-secret
          key: key1
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                               16,20         All
```

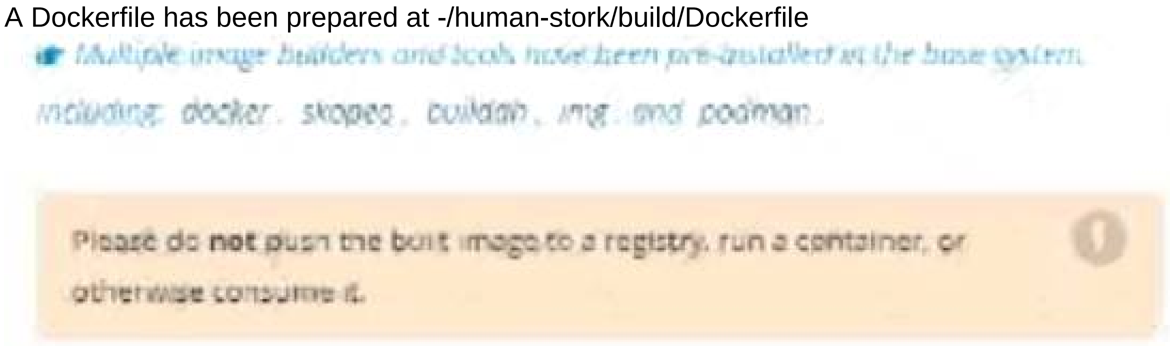


```
student@node-1:~$ kubectl get pods -n web
NAME     READY    STATUS     RESTARTS    AGE
cache    1/1      Running    0           9s
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                   TYPE                                   DATA   AGE
default-token-4kvr5    kubernetes.io/service-account-token    3      2d11h
some-secret            Opaque                                 1      5s
student@node-1:~$ kubectl run nginx-secret --image=nginx --dry-run=client -o yaml > nginx_secret
.yml
student@node-1:~$ vim nginx_secret.yml
student@node-1:~$ kubectl create -f nginx_secret.yml
pod/nginx-secret created
student@node-1:~$ kubectl get pods
NAME            READY    STATUS               RESTARTS    AGE
liveness-http   1/1      Running              0           6h38m
nginx-101       1/1      Running              0           6h39m
nginx-secret    0/1      ContainerCreating    0           4s
poller          1/1      Running              0           6h39m
student@node-1:~$ kubectl get pods
NAME            READY    STATUS     RESTARTS    AGE
liveness-http   1/1      Running    0           6h38m
nginx-101       1/1      Running    0           6h39m
nginx-secret    1/1      Running    0           8s
poller          1/1      Running    0           6h39m
student@node-1:~$
```

NEW QUESTION 8
Exhibit:


No configuration context change is required for this task.

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

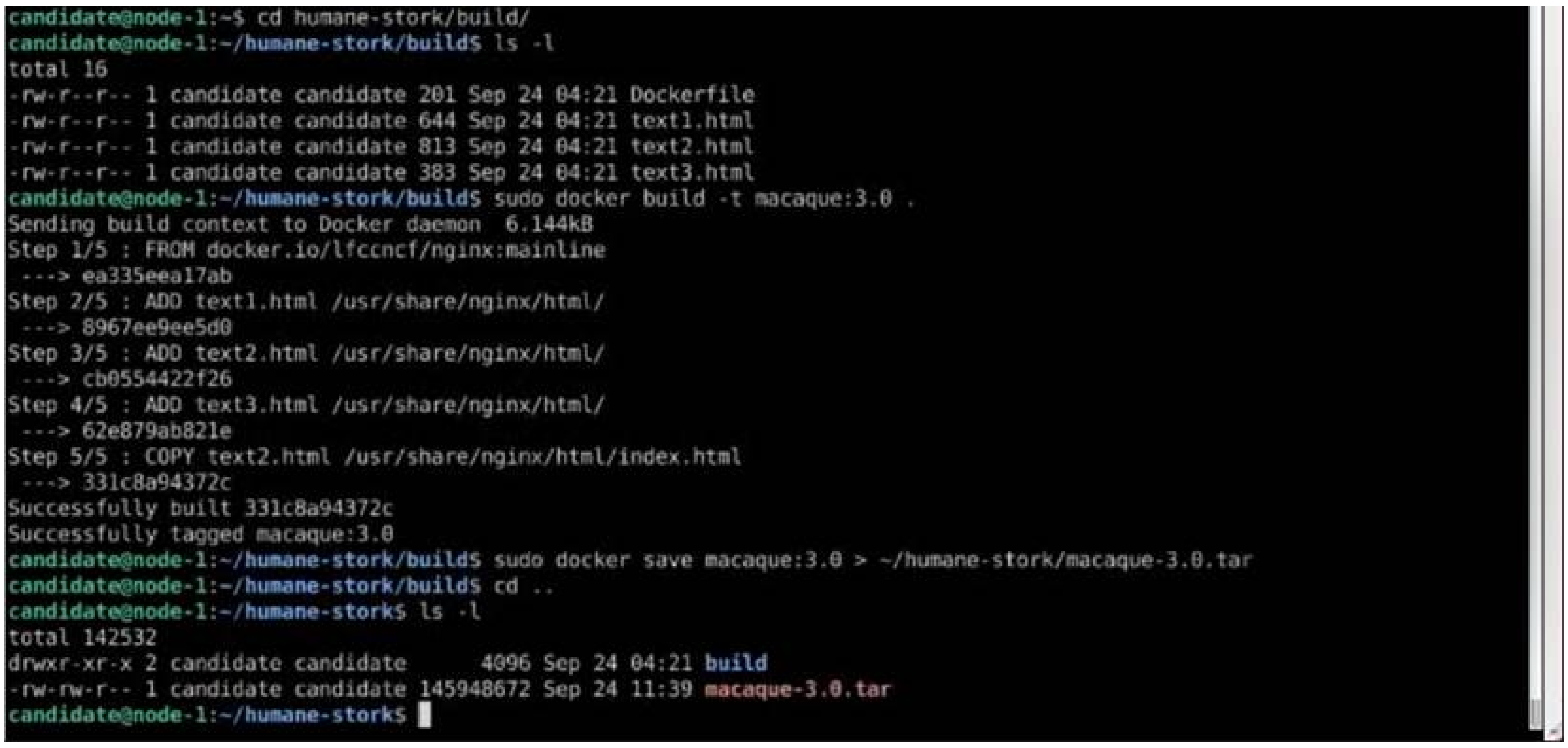

Multiple image builders and tools have been pre-installed in the base system, including: docker, skopeo, buildah, img and podman.

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

A. Mastered
B. Not Mastered

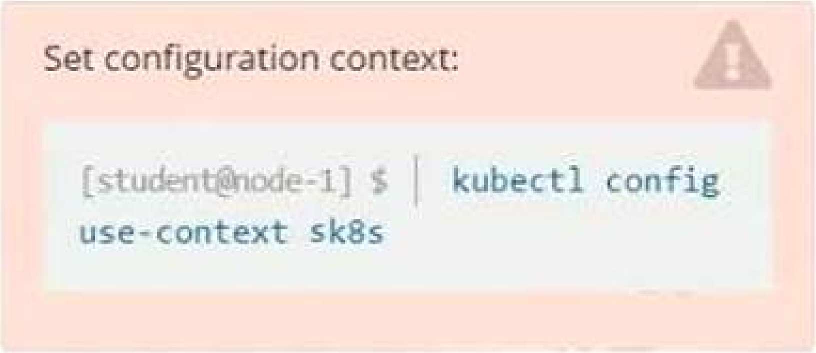
Answer: A

Explanation:
Solution:



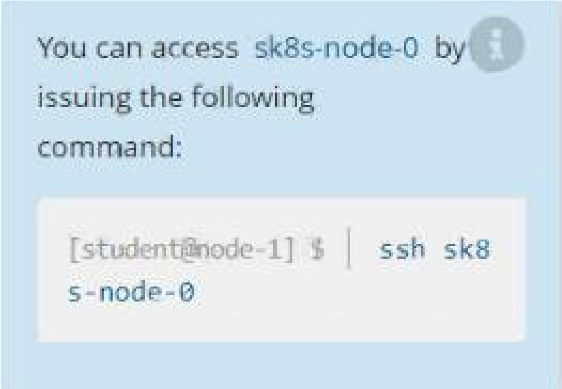
```
candidate@node-1:~$ cd humane-stork/build/
candidate@node-1:~/humane-stork/build$ ls -l
total 16
-rw-r--r-- 1 candidate candidate 201 Sep 24 04:21 Dockerfile
-rw-r--r-- 1 candidate candidate 644 Sep 24 04:21 text1.html
-rw-r--r-- 1 candidate candidate 813 Sep 24 04:21 text2.html
-rw-r--r-- 1 candidate 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/lfccncf/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$ 
```

NEW QUESTION 9
Exhibit:


Set configuration context:

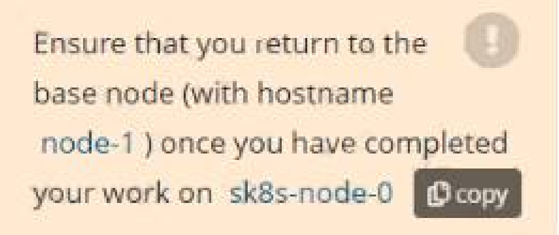
```
[student@node-1] $ | kubectl config use-context sk8s
```

Context
A project that you are working on has a requirement for persistent data to be available. Task
To facilitate this, perform the following tasks:
• Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance
• Create a PersistentVolume named task-pv-volume using hostPath and allocate 1Gi to it, specifying that the volume is at /opt/KDSP00101/data on the cluster's node. The configuration should specify the access mode of ReadWriteOnce . It should define the StorageClass name exam for the PersistentVolume , which will be used to bind PersistentVolumeClaim requests to this PersistenetVolume.
• Create a PefsissentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce
• Create a pod that uses the PersistentVolmeClaim as a volume with a label app: my-storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod



You can access sk8s-node-0 by issuing the following command:

```
[student@node-1] $ | ssh sk8s-node-0
```



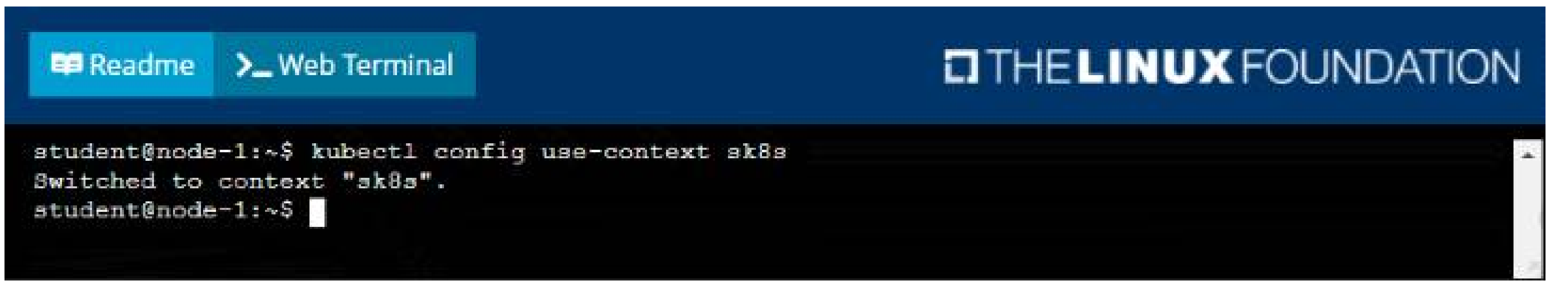
Ensure that you return to the base node (with hostname node-1) once you have completed your work on sk8s-node-0

A. Mastered
B. Not Mastered

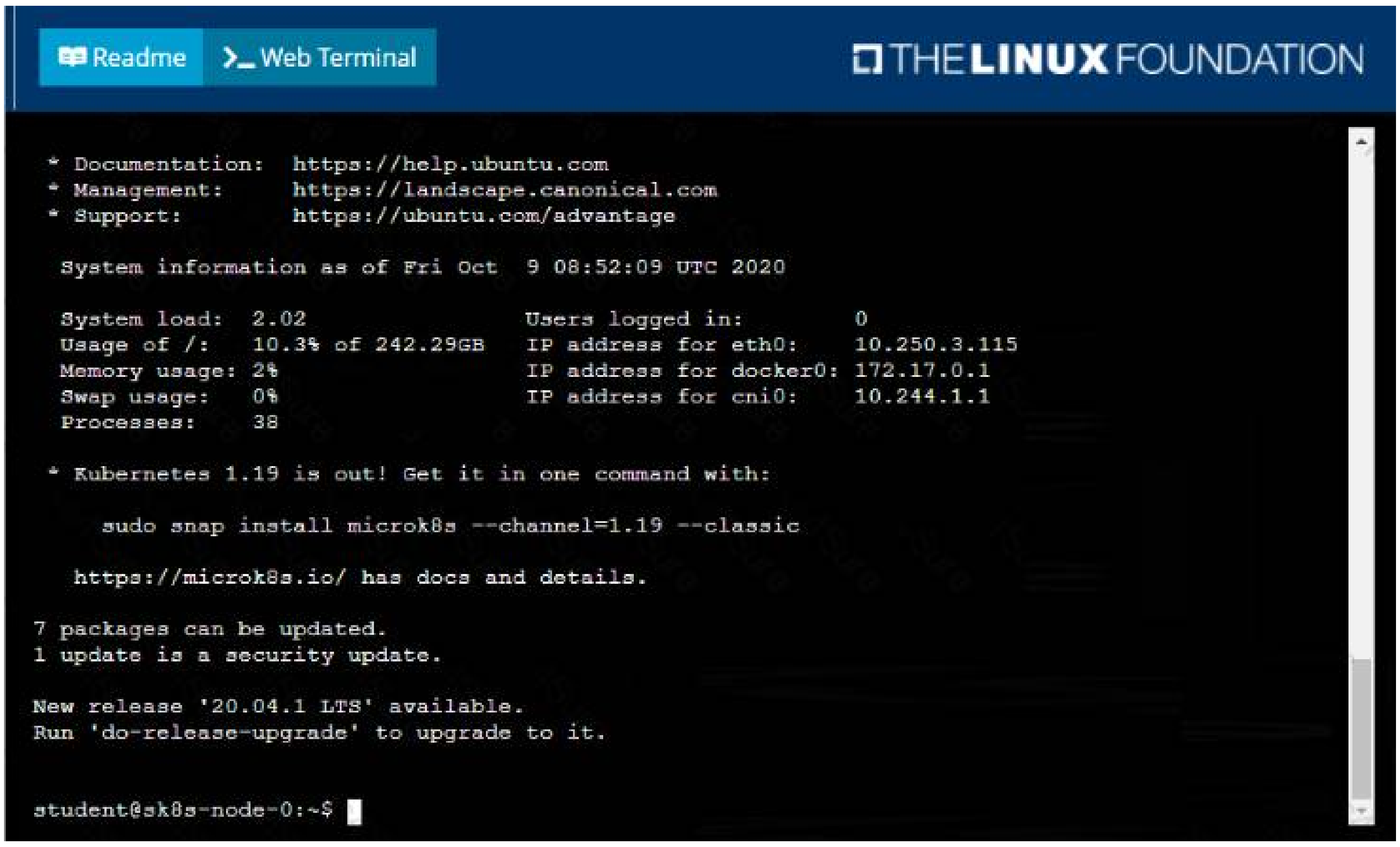
Answer: A

Explanation:

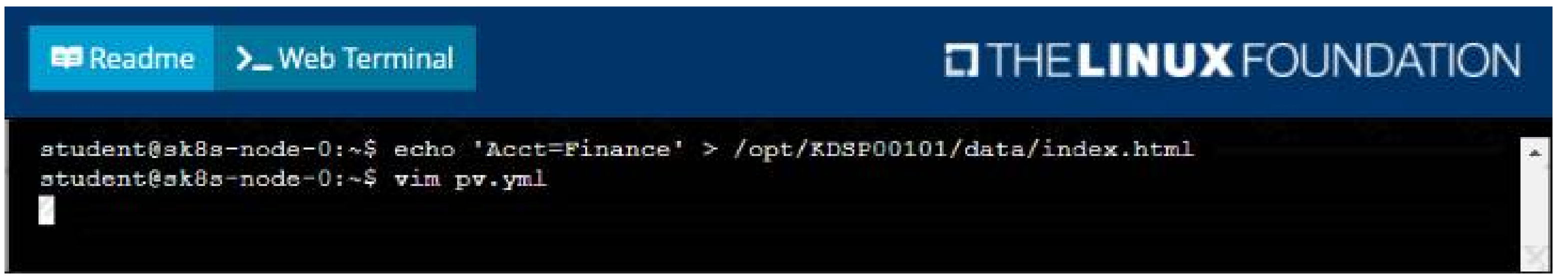
Solution:



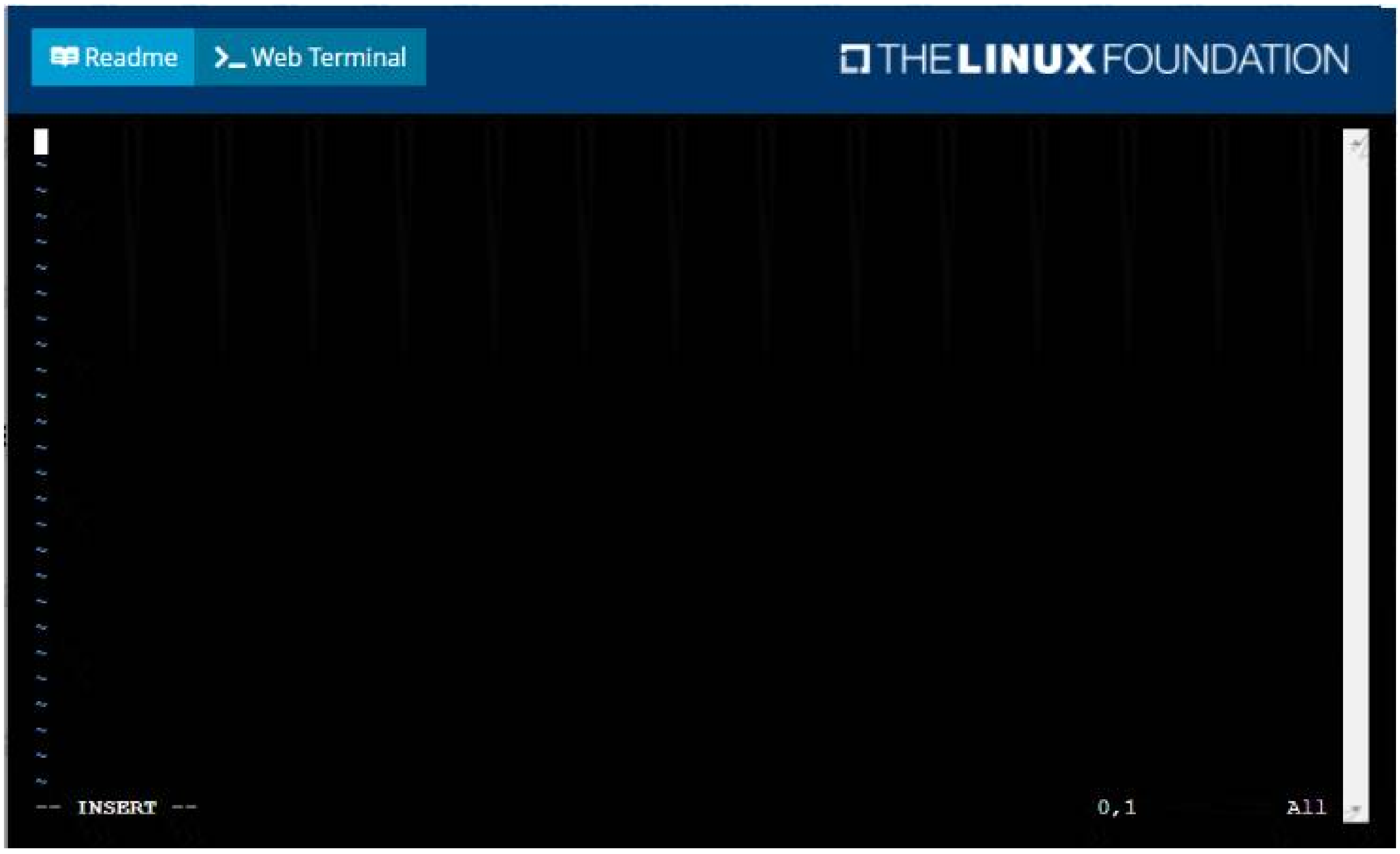
```
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$
```



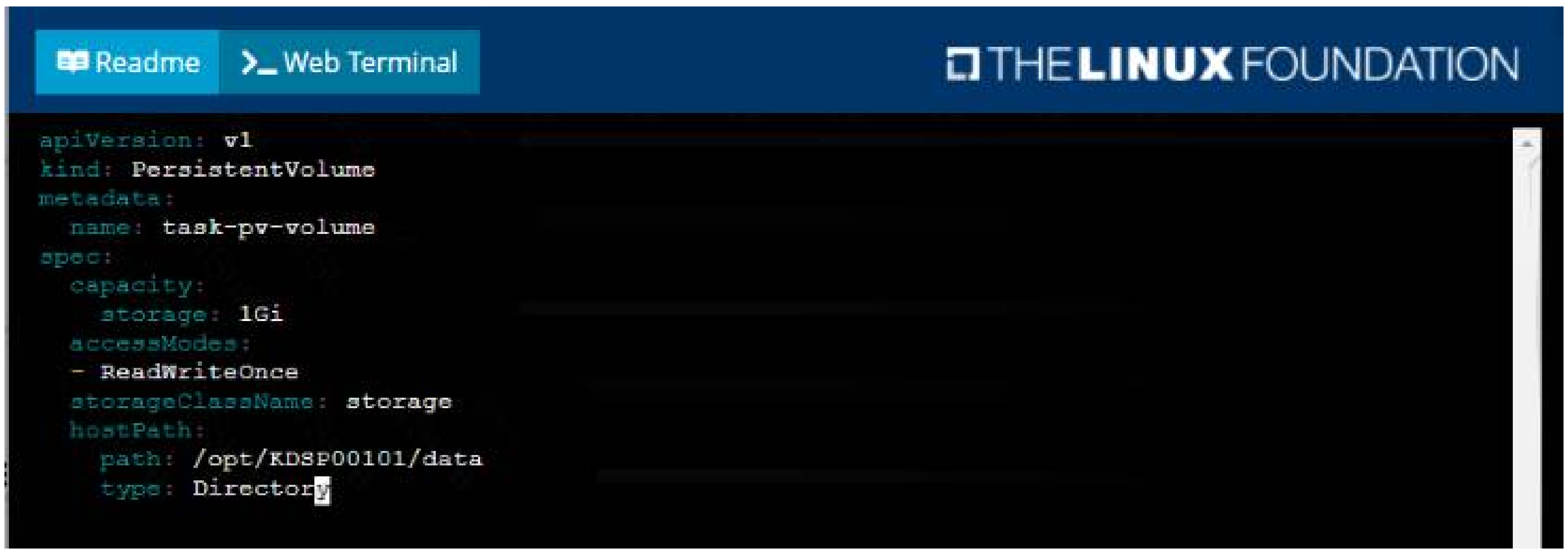
```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Oct  9 08:52:09 UTC 2020

  System load:  2.02               Users logged in:       0
  Usage of /:   10.3% of 242.29GB  IP address for eth0:   10.250.3.115
  Memory usage: 2%                 IP address for docker0: 172.17.0.1
  Swap usage:   0%                 IP address for cni0:    10.244.1.1
  Processes:    38

 * Kubernetes 1.19 is out! Get it in one command with:

     sudo snap install microk8s --channel=1.19 --classic

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

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

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


student@sk8s-node-0:~$
```



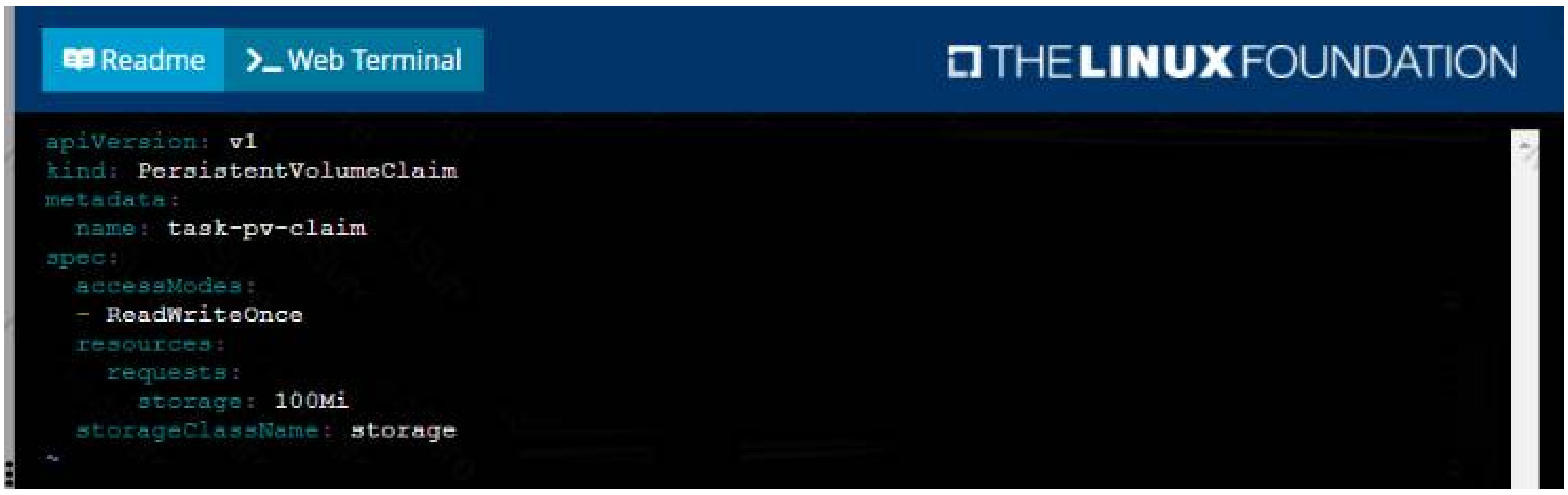
```
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml
```



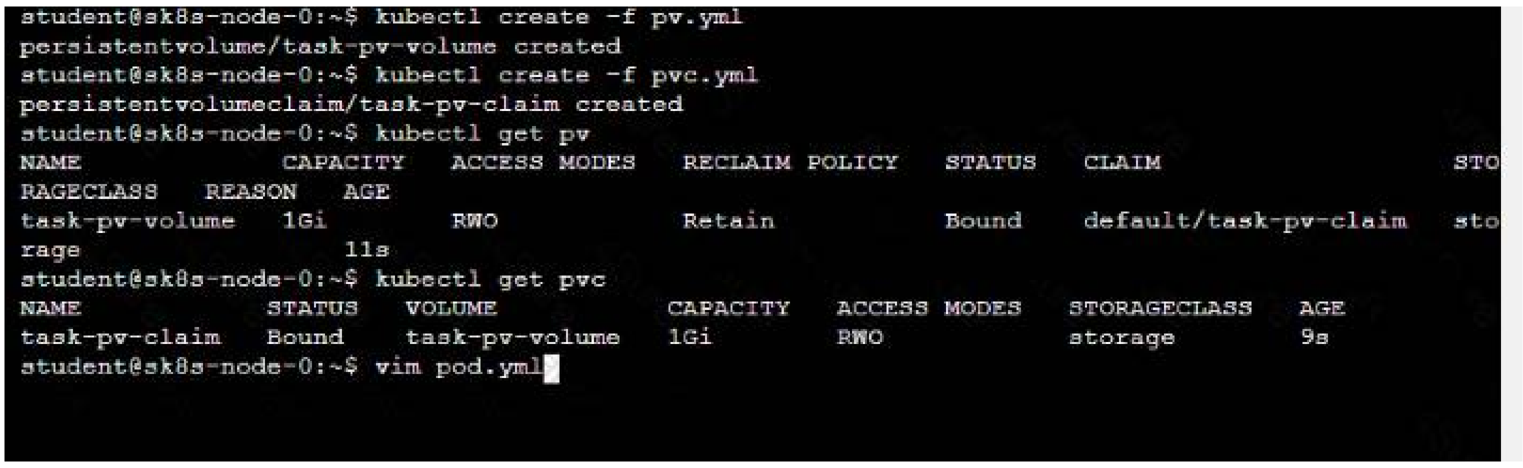
```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                        0,1           All
```



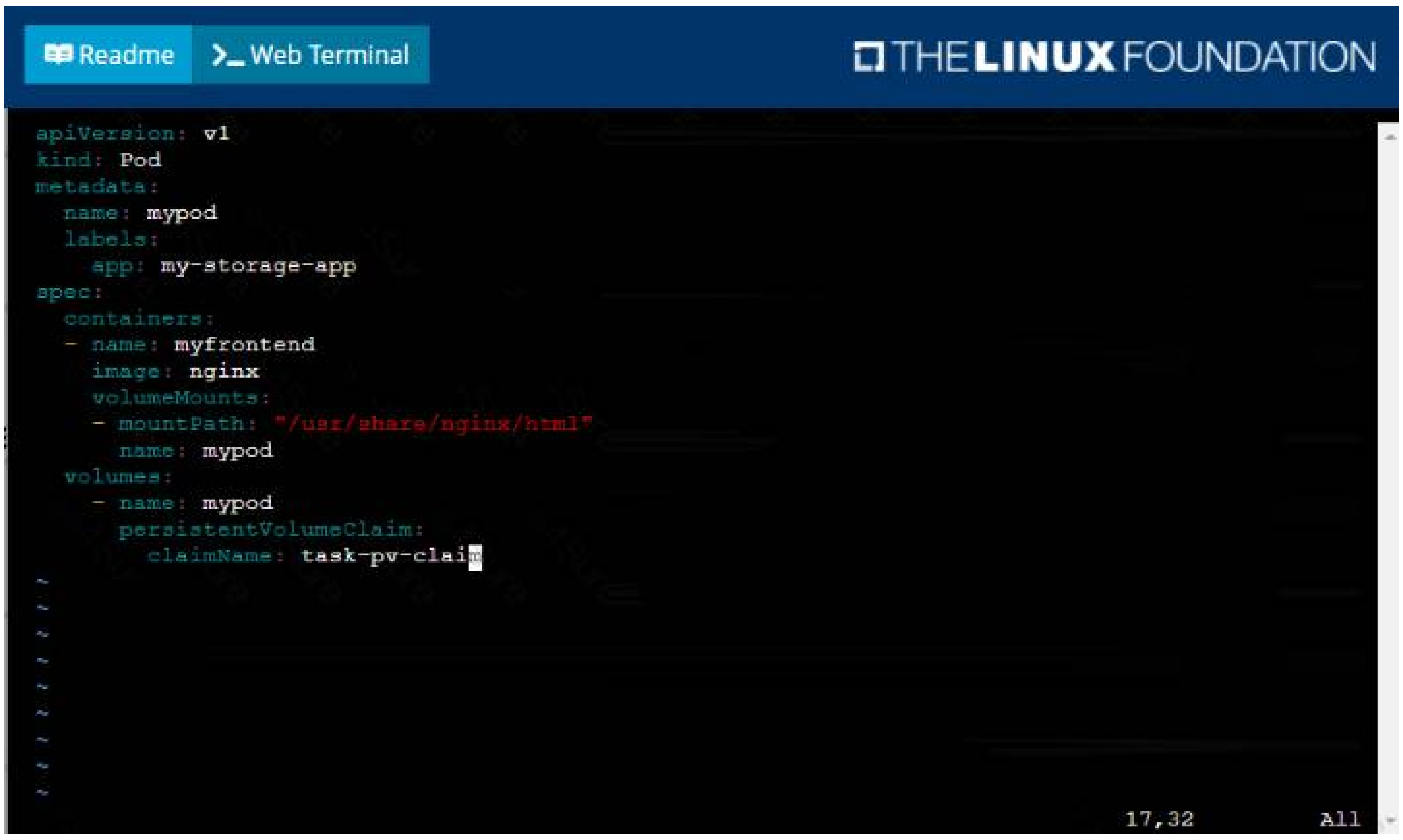
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
spec:
  capacity:
    storage: 1Gi
  accessModes:
  - ReadWriteOnce
  storageClassName: storage
  hostPath:
    path: /opt/KDSP00101/data
    type: Directory
```



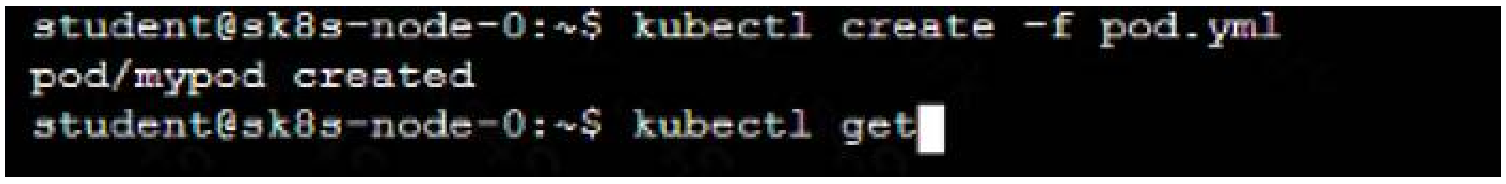
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: storage
~
```



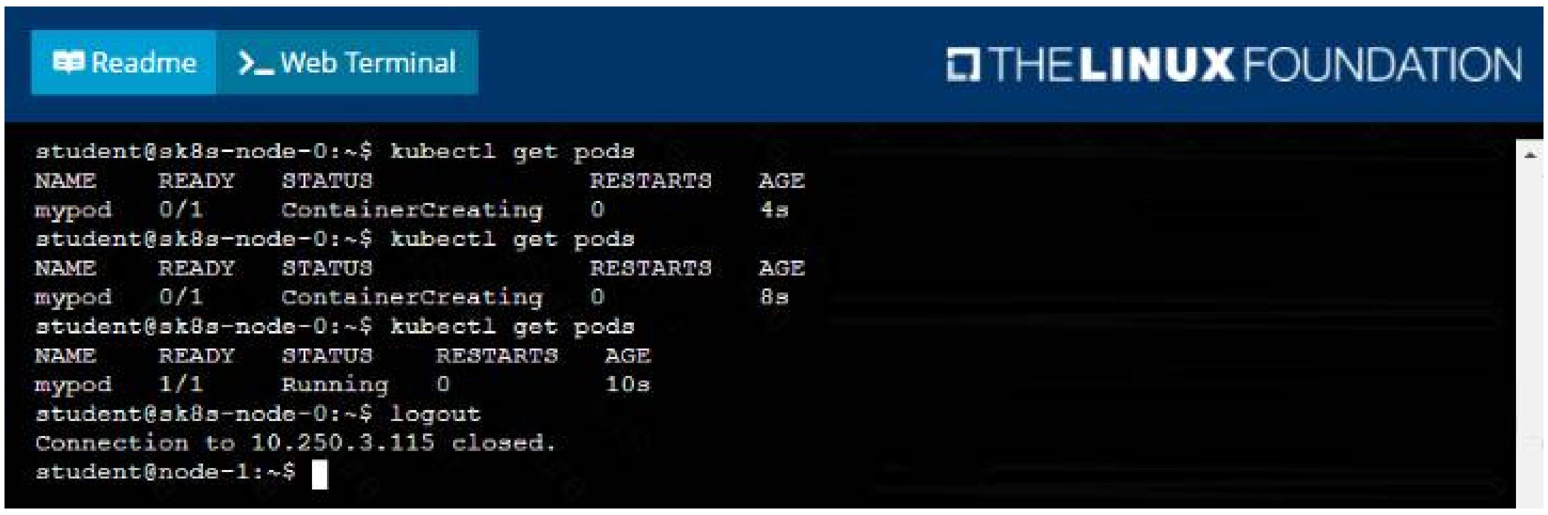
```
student@sk8s-node-0:~$ kubectl create -f pv.yml
persistentvolume/task-pv-volume created
student@sk8s-node-0:~$ kubectl create -f pvc.yml
persistentvolumeclaim/task-pv-claim created
student@sk8s-node-0:~$ kubectl get pv
NAME             CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                   STO
RAGECLASS    REASON    AGE
task-pv-volume   1Gi        RWO            Retain           Bound    default/task-pv-claim   sto
rage                   11s
student@sk8s-node-0:~$ kubectl get pvc
NAME            STATUS   VOLUME           CAPACITY   ACCESS MODES   STORAGECLASS   AGE
task-pv-claim   Bound    task-pv-volume   1Gi        RWO            storage        9s
student@sk8s-node-0:~$ vim pod.yml
```



```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: my-storage-app
spec:
  containers:
  - name: myfrontend
    image: nginx
    volumeMounts:
    - mountPath: "/usr/share/nginx/html"
      name: mypod
  volumes:
    - name: mypod
      persistentVolumeClaim:
        claimName: task-pv-claim
~
~
~
~
~
~
~
~
~
                                                                17,32          All
```

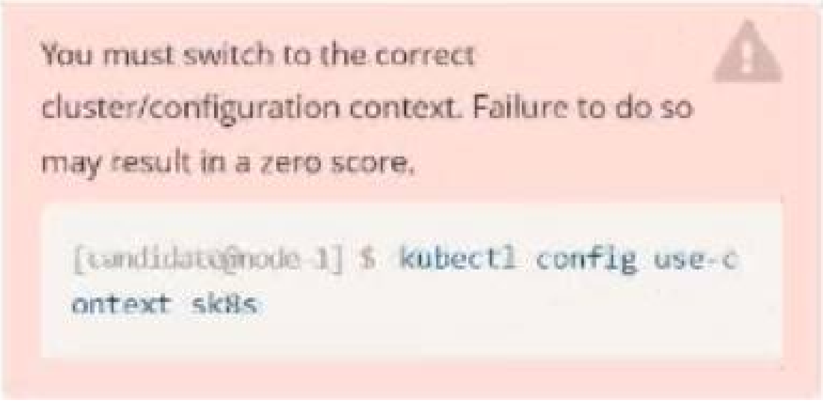


```
student@sk8s-node-0:~$ kubectl create -f pod.yml
pod/mypod created
student@sk8s-node-0:~$ kubectl get
```



```
student@sk8s-node-0:~$ kubectl get pods
NAME      READY    STATUS              RESTARTS    AGE
mypod     0/1      ContainerCreating   0           4s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY    STATUS              RESTARTS    AGE
mypod     0/1      ContainerCreating   0           8s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY    STATUS    RESTARTS    AGE
mypod     1/1      Running   0           10s
student@sk8s-node-0:~$ logout
Connection to 10.250.3.115 closed.
student@node-1:~$
```

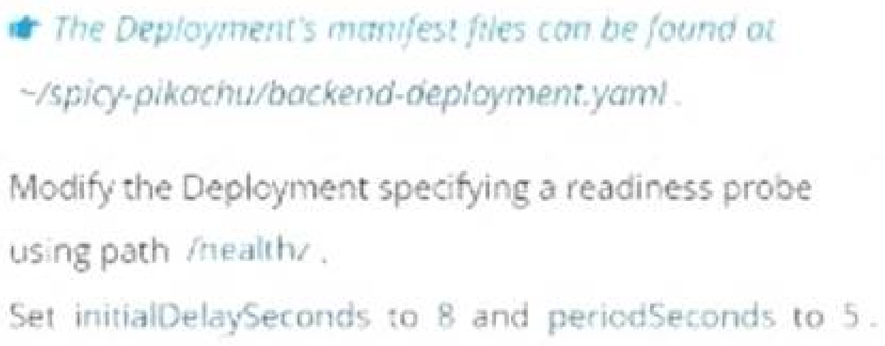
NEW QUESTION 10
Exhibit:



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-c
ontext sk8s
```

Task
A Deployment named backend-deployment in namespace staging runs a web application on port 8081.



The Deployment's manifest files can be found at ~/spicy-pikachu/backend-deployment.yaml .

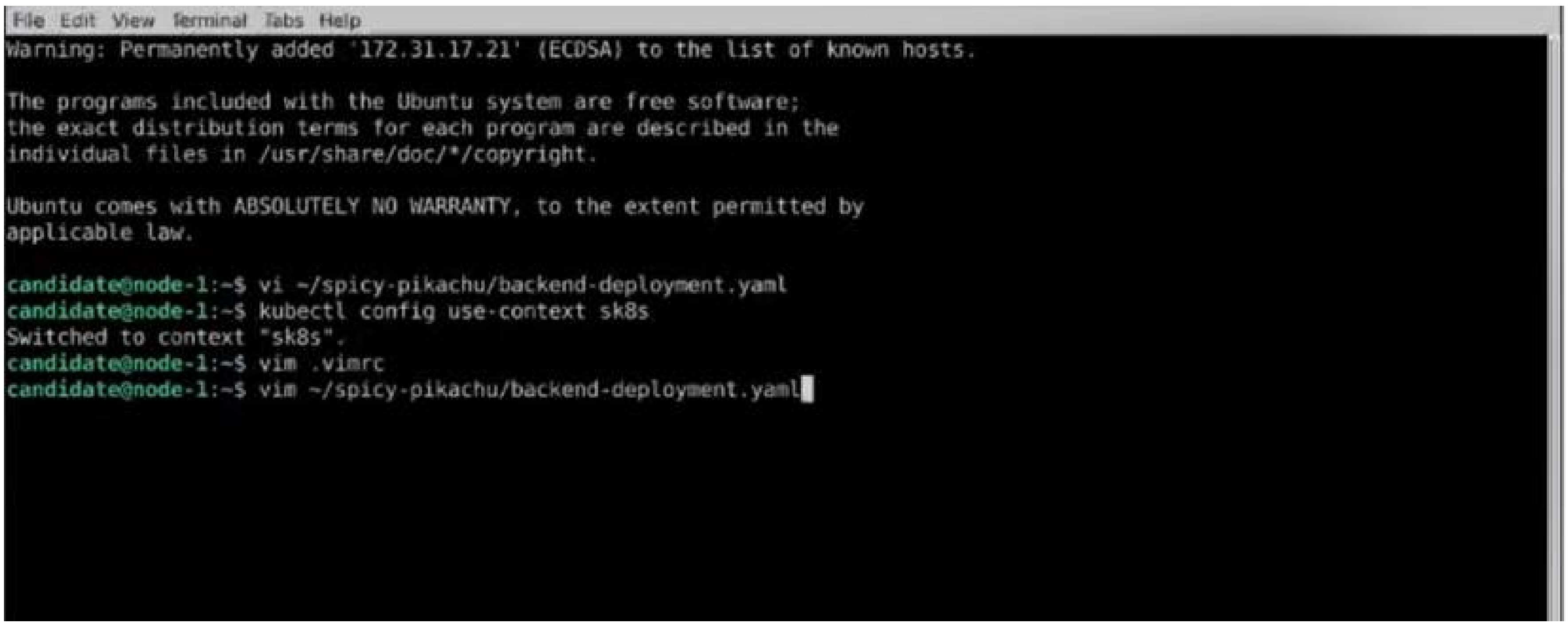
Modify the Deployment specifying a readiness probe using path /healthz .

Set initialDelaySeconds to 8 and periodSeconds to 5 .

A. Mastered
B. Not Mastered

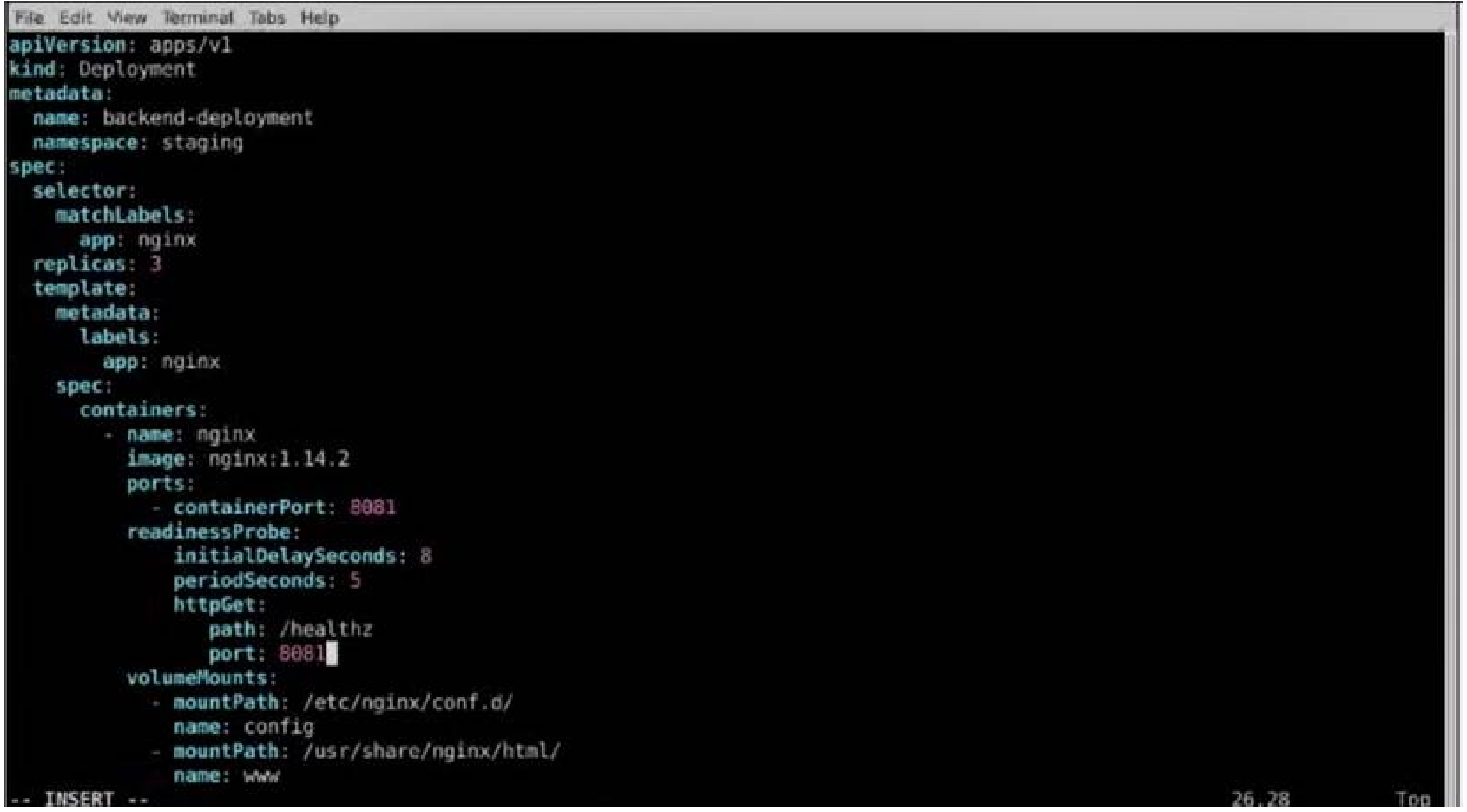
Answer: A

Explanation:
Solution:

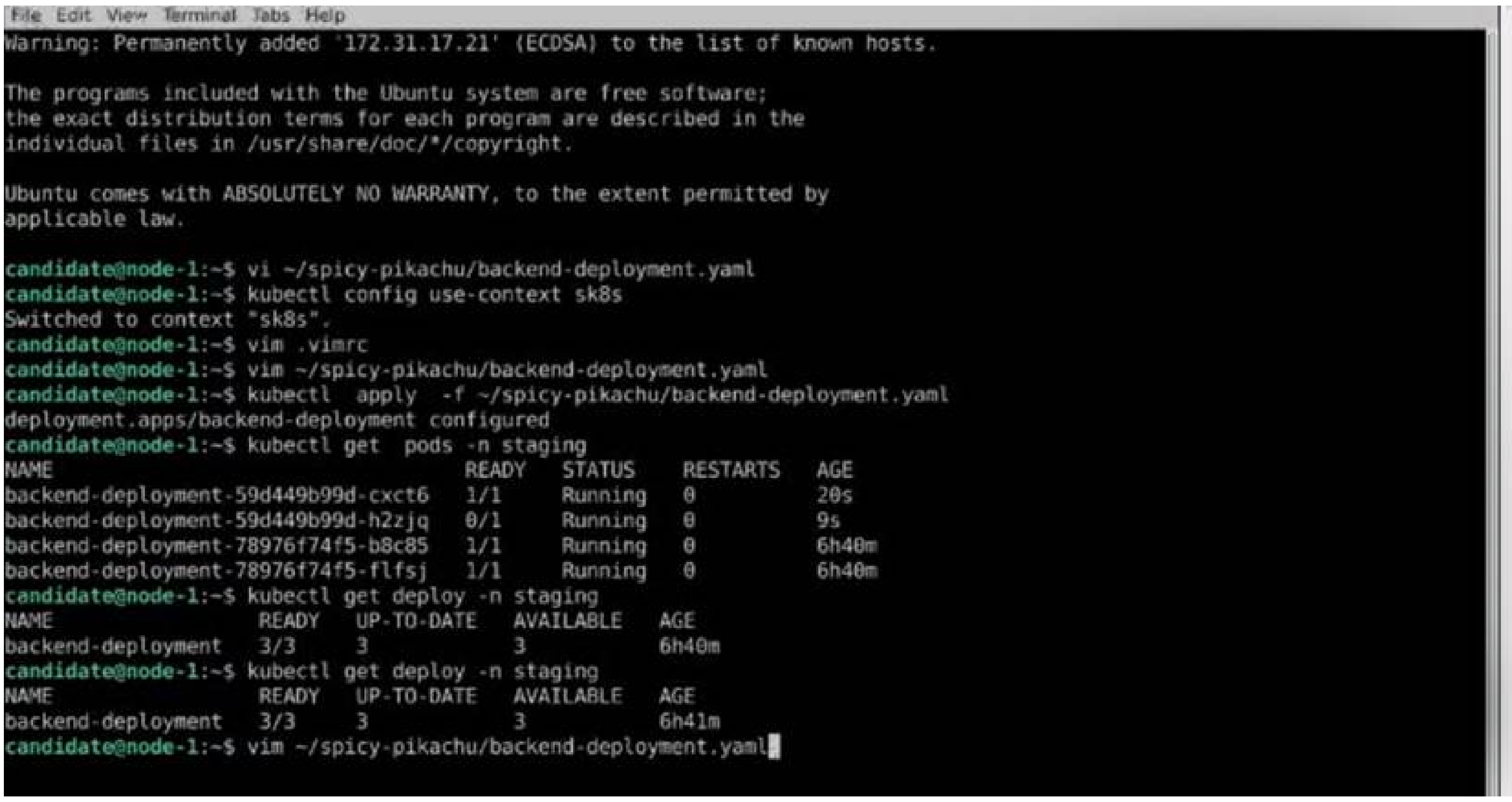


```
Warning: Permanently added '172.31.17.21' (ECDSA) to the list of known hosts.

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
```

Text Description automatically generated

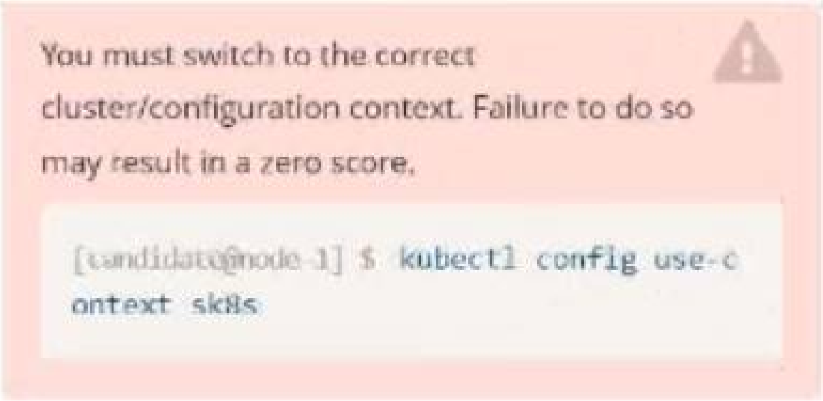


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-deployment
  namespace: staging
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 8081
          readinessProbe:
            initialDelaySeconds: 8
            periodSeconds: 5
            httpGet:
               path: /healthz
               port: 8081
          volumeMounts:
            - mountPath: /etc/nginx/conf.d/
              name: config
            - mountPath: /usr/share/nginx/html/
              name: www
-- INSERT --                                                                                  26,28          Top
```



```
Warning: Permanently added '172.31.17.21' (ECDSA) to the list of known hosts.

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
```

NEW QUESTION 10
Exhibit:



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

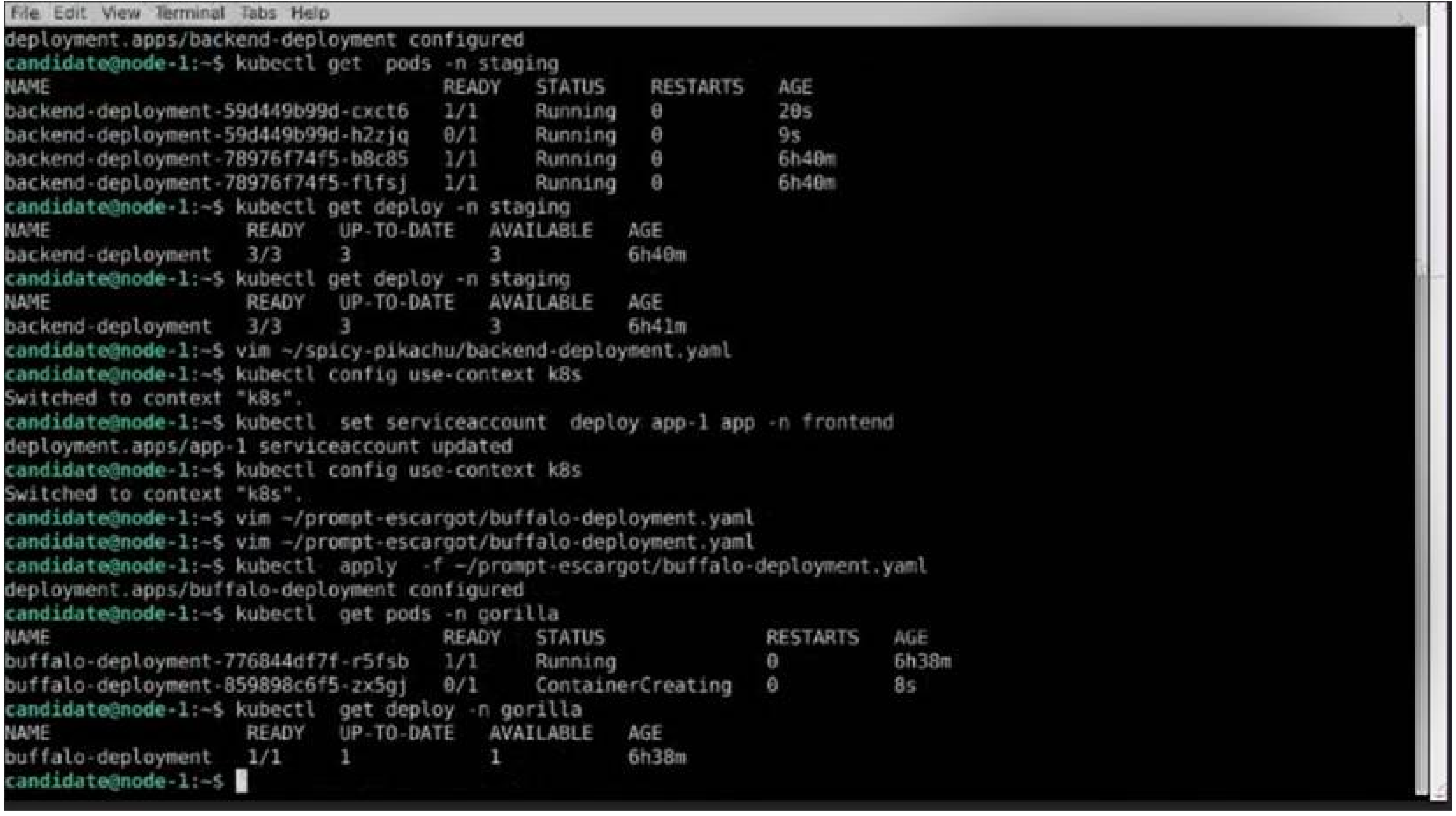
```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

Task:
A pod within the Deployment named buffale-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”
The buffalo-deployment ‘S manifest can be found at -/prompt/escargot/buffalo-deployment.yaml

A. Mastered
B. Not Mastered

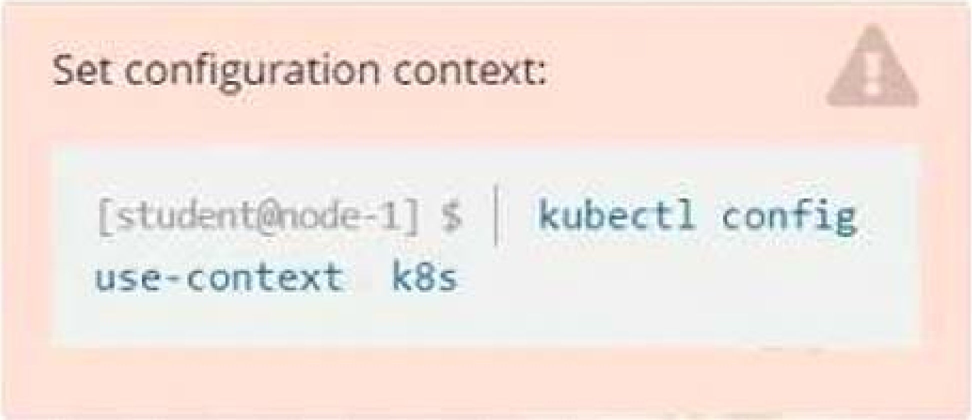
Answer: A

Explanation:
Solution:
Text Description automatically generated



```
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:~$ 
```

NEW QUESTION 12
Exhibit:



Set configuration context:

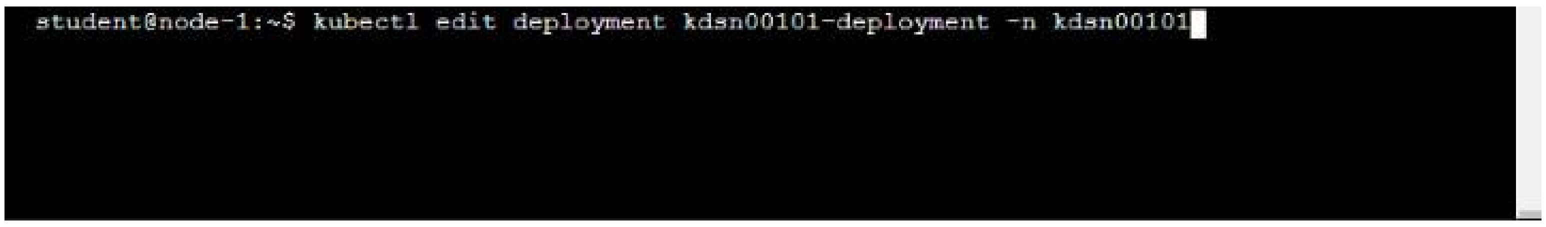
```
[student@node-1] $ | kubectl config
use-context  k8s
```

Context
You have been tasked with scaling an existing deployment for availability, and creating a service to expose the deployment within your infrastructure.
Task
Start with the deployment named kdsn00101-deployment which has already been deployed to the namespace kdsn00101 . Edit it to:
• Add the func=webFrontEnd key/value label to the pod template metadata to identify the pod for the service definition
• Have 4 replicas
Next, create ana deploy in namespace kdsn00l01 a service that accomplishes the following:
• Exposes the service on TCP port 8080
• is mapped to me pods defined by the specification of kdsn00l01-deployment
• Is of type NodePort
• Has a name of cherry

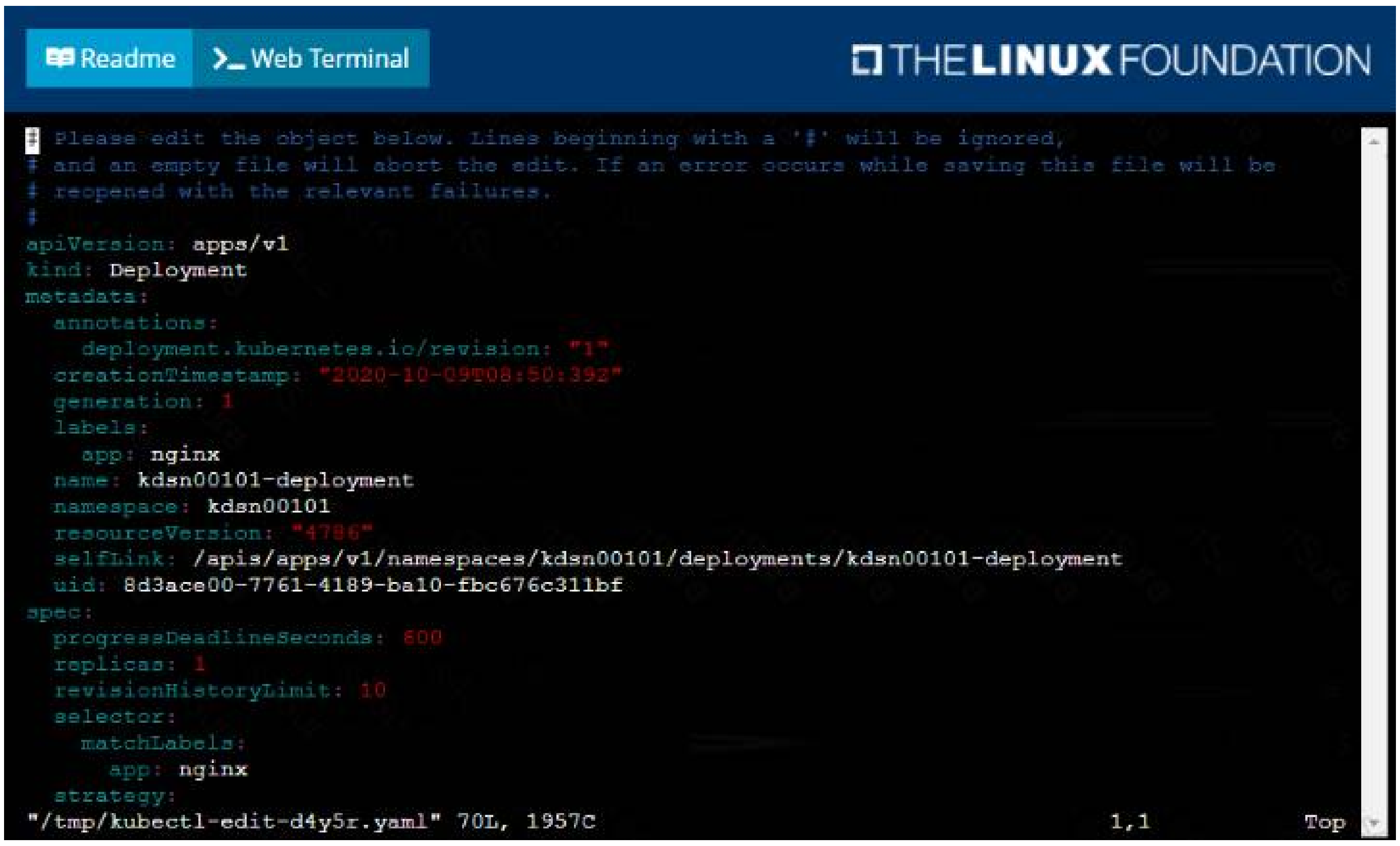
A. Mastered
B. Not Mastered

Answer: A

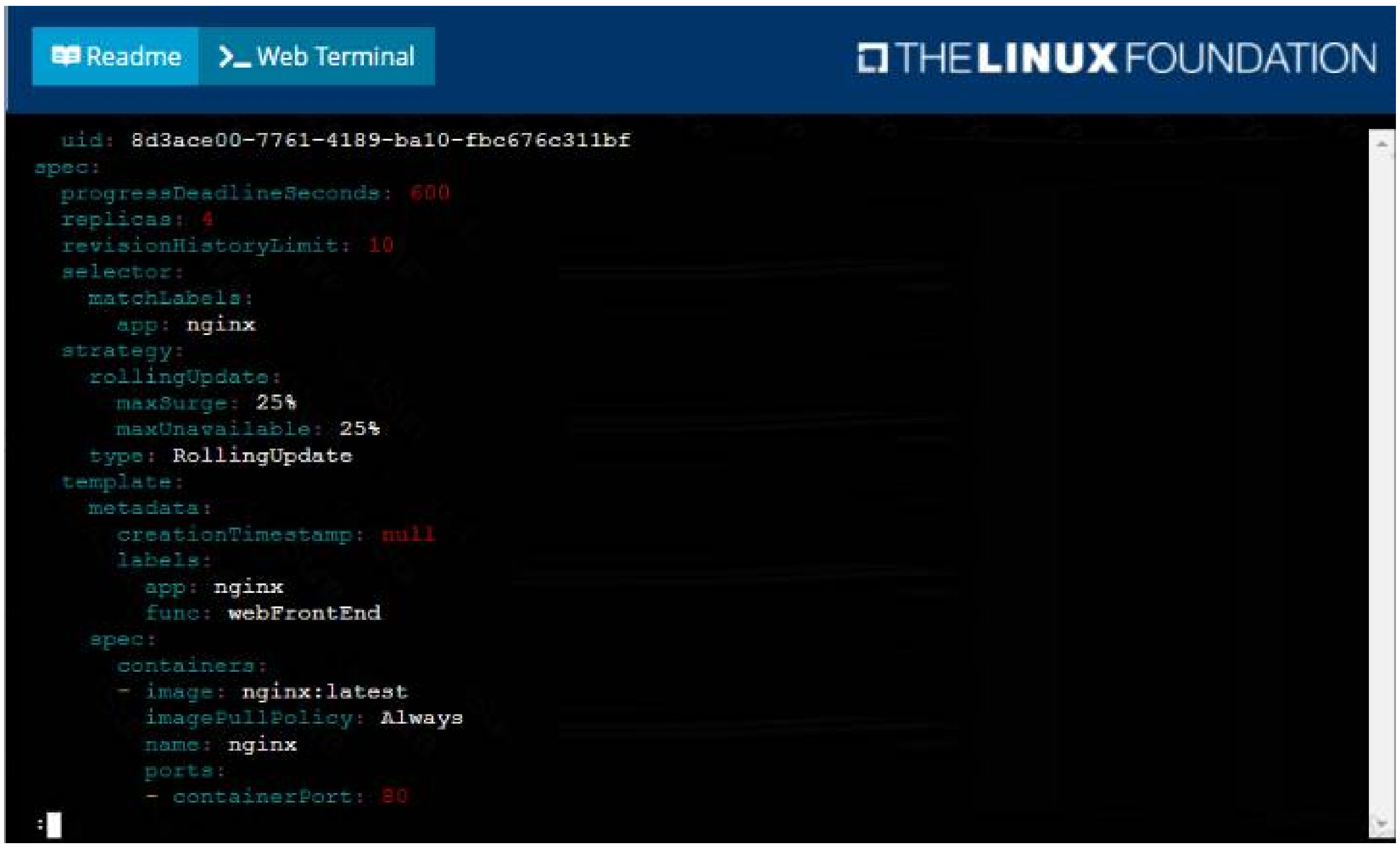
Explanation:
Solution:



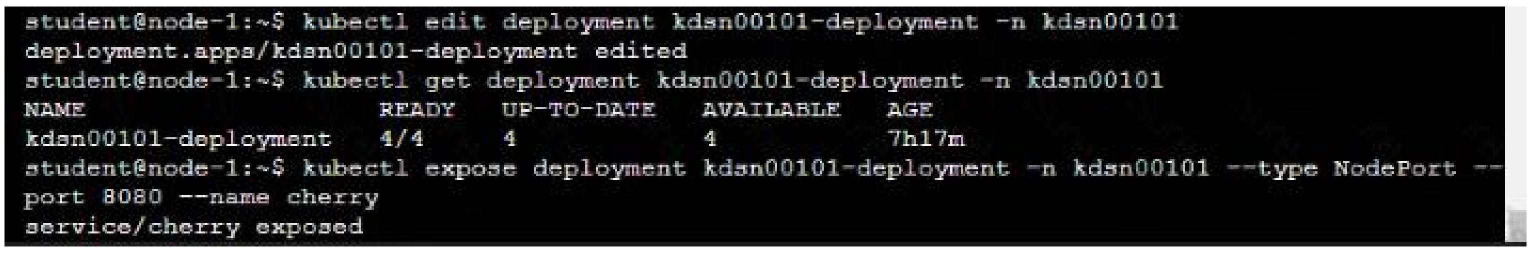
```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
```



```
# 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: "2020-10-09T08:50:39Z"
  generation: 1
  labels:
    app: nginx
  name: kdsn00101-deployment
  namespace: kdsn00101
  resourceVersion: "4786"
  selfLink: /apis/apps/v1/namespaces/kdsn00101/deployments/kdsn00101-deployment
  uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
"/tmp/kubectl-edit-d4y5r.yaml" 70L, 1957C                              1,1          Top
```

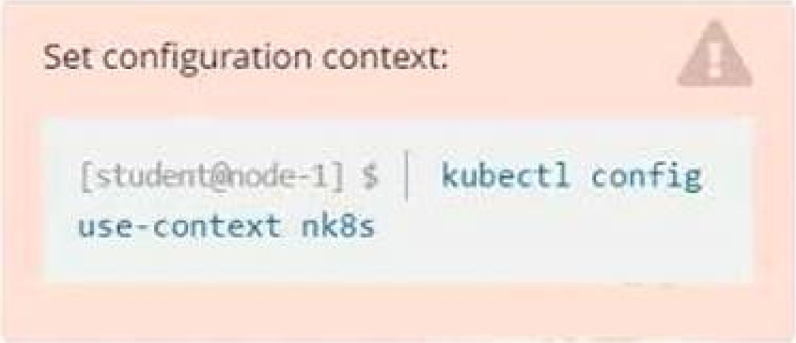


```
  uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        func: webFrontEnd
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
        - containerPort: 80
:
```



```
student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
deployment.apps/kdsn00101-deployment edited
student@node-1:~$ kubectl get deployment kdsn00101-deployment -n kdsn00101
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
kdsn00101-deployment   4/4     4            4           7h17m
student@node-1:~$ kubectl expose deployment kdsn00101-deployment -n kdsn00101 --type NodePort --
port 8080 --name cherry
service/cherry exposed
```

NEW QUESTION 13
Exhibit:

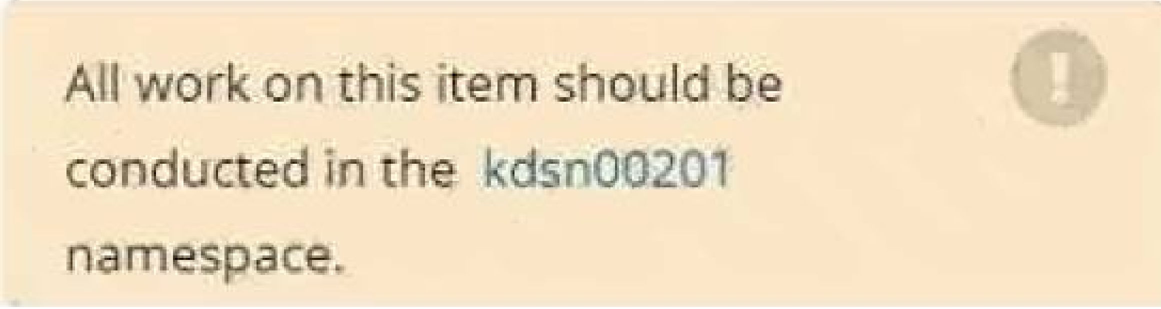


Set configuration context:

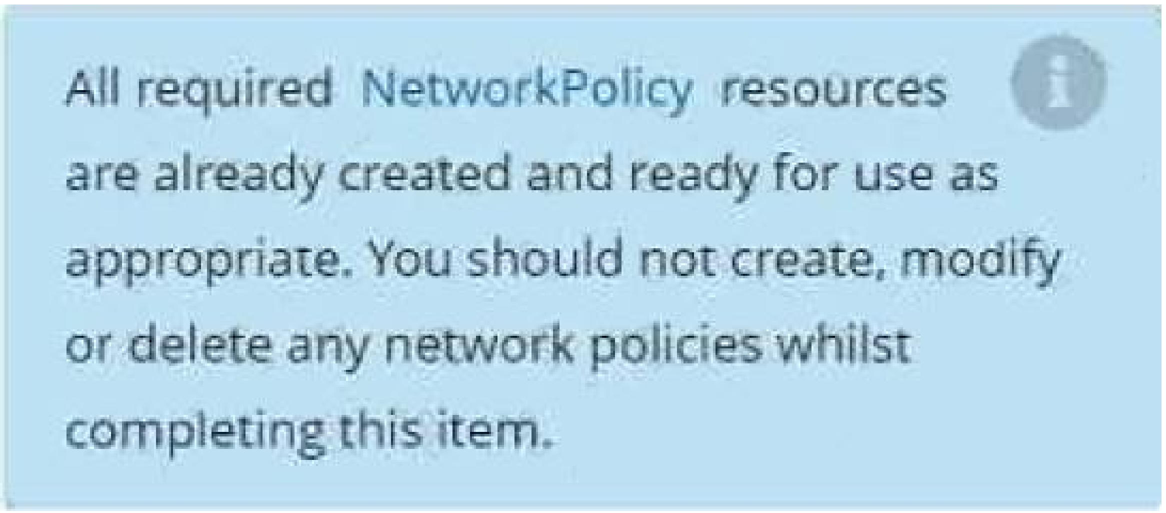
```
[student@node-1] $ | kubectl config
use-context nk8s
```

Task

You have rolled out a new pod to your infrastructure and now you need to allow it to communicate with the web and storage pods but nothing else. Given the running pod kdsn00201 -newpod edit it to use a network policy that will allow it to send and receive traffic only to and from the web and storage pods.



All work on this item should be conducted in the kdsn00201 namespace.



All required NetworkPolicy resources are already created and ready for use as appropriate. You should not create, modify or delete any network policies whilst completing this item.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
apiVersion: networking.k8s.io/v1 kind: NetworkPolicy
metadata:
name: internal-policy namespace: default spec:
podSelector: matchLabels: name: internal policyTypes:
- Egress
- Ingress ingress:
- {}
egress:
- to:
- podSelector: matchLabels: name: mysql ports:
- protocol: TCP port: 3306
- to:
- podSelector: matchLabels:
name: payroll ports:
- protocol: TCP port: 8080
- ports:
- port: 53 protocol: UDP
- port: 53 protocol: TCP

NEW QUESTION 17
......