CLOUD NATIVE
COMPUTING FOUNDATION

CKAD

Certified Kubernetes

Application Developer

E EXAMKILLER

Help Pass Your Exam At First Try

Cloud Native Computing Foundation

Exam CKAD

Certified Kubernetes Application Developer (CKAD) Program

Version: 5.2

[Total Questions: 33]

Question No : 1 CORRECT TEXT

Context

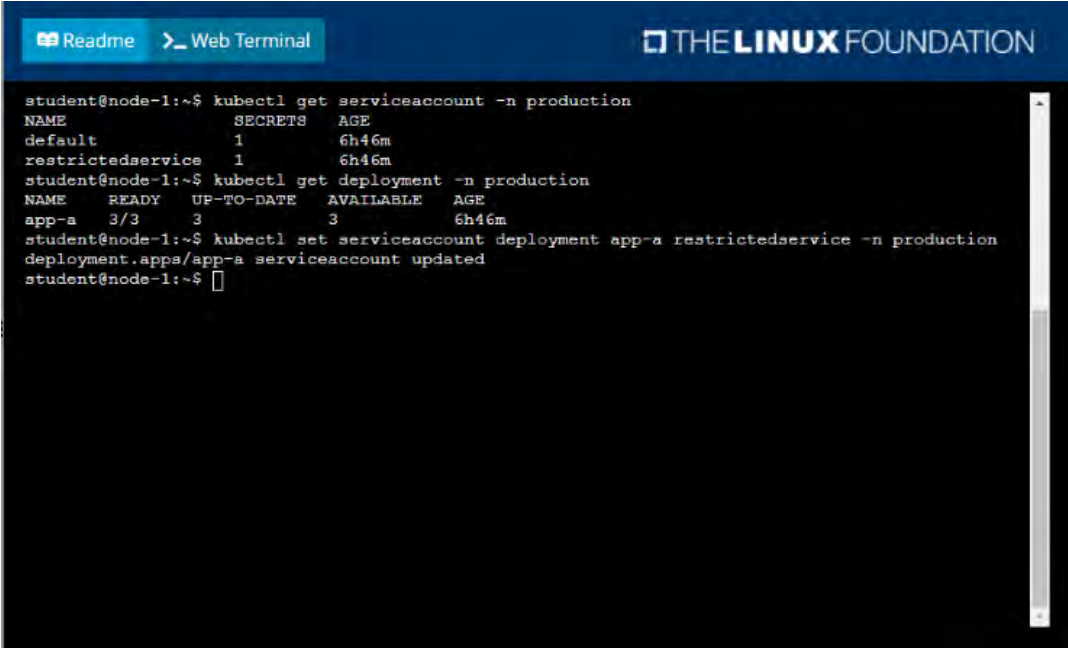Your application's namespace requires a specific service account to be used.

Task

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

Answer: See the solution below.

Explanation:
Solution:

Question No : 2 CORRECT TEXT

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:

1) Create a secret named app-secret in the default namespace containing the following single key-value pair:

Key3: value1

2) Create a Pod named ngnix secret in the default namespace.Specify a single container using the nginx:stable image.

Add an environment variable named BEST_VARIABLE consuming the value of the secret key3.

Answer: See the solution below.
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
```

```
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:~$
```

Text

Description automatically generated

Question No : 3 CORRECT TEXT

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 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.

Answer: 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
```

```
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 with medium confidence

```
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>
```
Text

Description automatically generated

Question No : 4 CORRECT TEXT

Set configuration context:

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

Context

As a Kubernetes application developer you will often find yourself needing to update a running application.
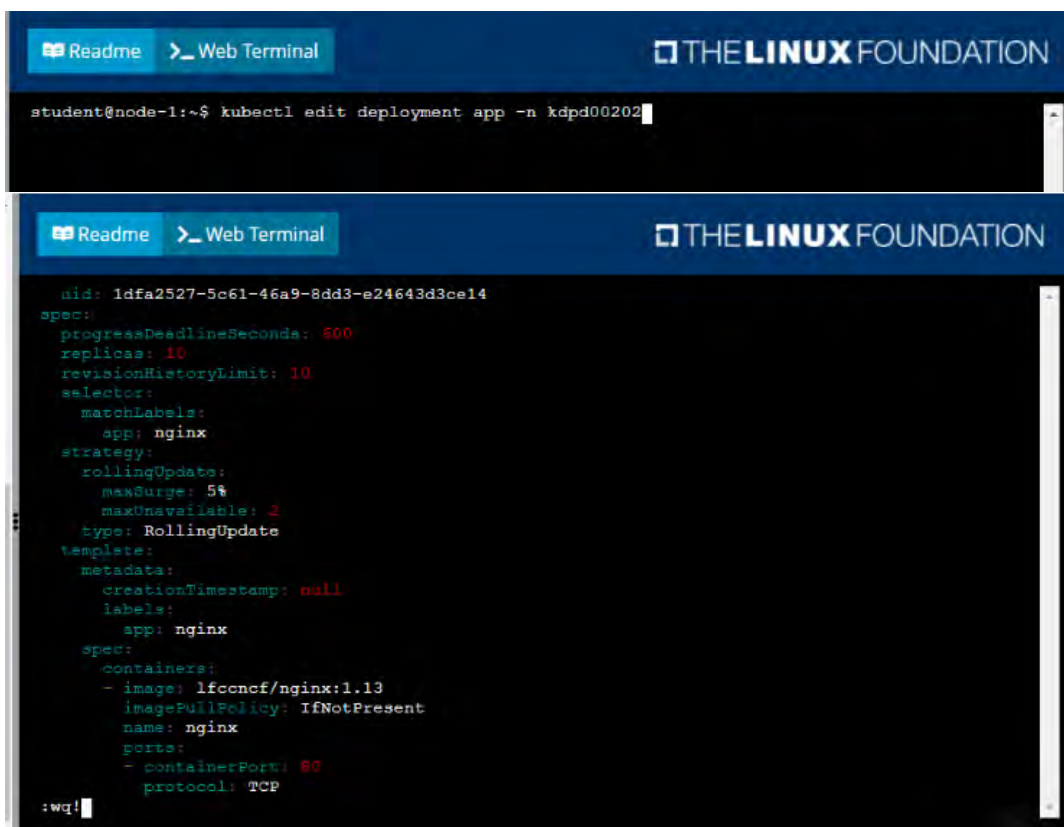
Task

Please complete the following:

• Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%

• Perform a rolling update of the web1 deployment, changing the lfccncf/ngmx image version to 1.13

• Roll back the app deployment to the previous version
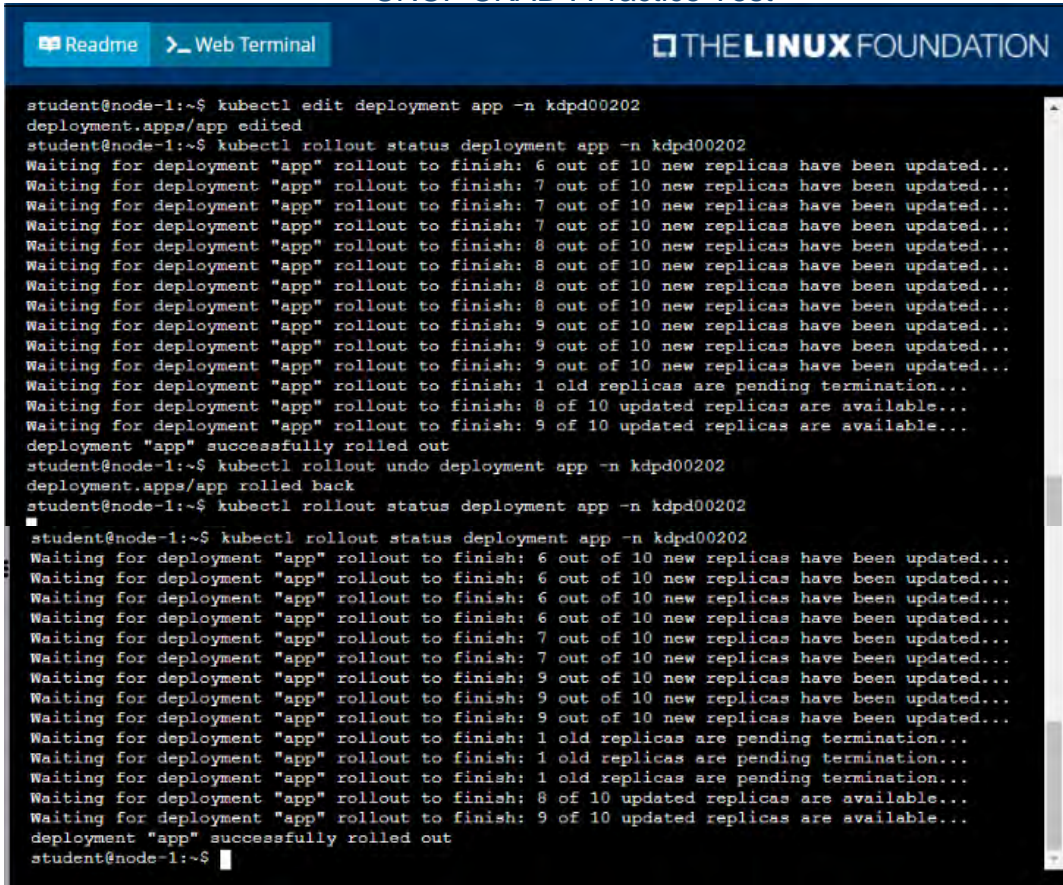
Answer: See the solution below.

Explanation:
Solution:

```
student@node-1:~$ kubectl edit deployment app -n kdpd00202
deployment.apps/app edited
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$ kubectl rollout undo deployment app -n kdpd00202
deployment.apps/app rolled back
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$
```

Question No : 5 CORRECT TEXT

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 lag 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 a t-/human stork/macque 3.0 tar

Answer: 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/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$ 
```

```
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/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/
```

```
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/builds 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/builds sudo docker build -t macaque:1.0 .
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/builds sudo docker save macaque:3.0 > ~/humane-stork/macaque-3.0.tar
```

Question No : 6 CORRECT TEXT

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 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

Answer: See the solution below.

Explanation:

Solution:

Text

Description automatically generated

Text

Description automatically generated

Question No : 7 CORRECT TEXT

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

Answer: See the solution below.

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:~$
```

Question No : 8 CORRECT TEXT

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:
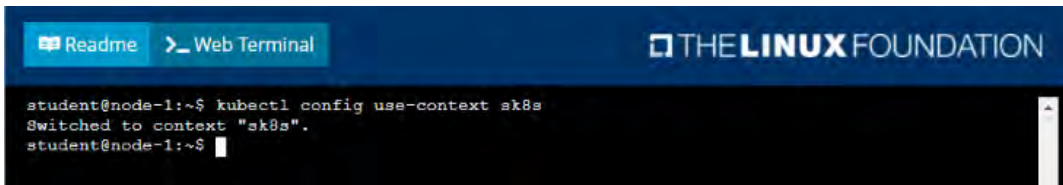
```
[student@node-1] $   ssh sk8
s-node-0
```

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

Answer: See the solution below.

Explanation:
Solution:

📖 Readme   >_ Web Terminal   □ THE **LINUX** FOUNDATION

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

```
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:~$
```

Question No : 9 CORRECT TEXT

Set configuration context:

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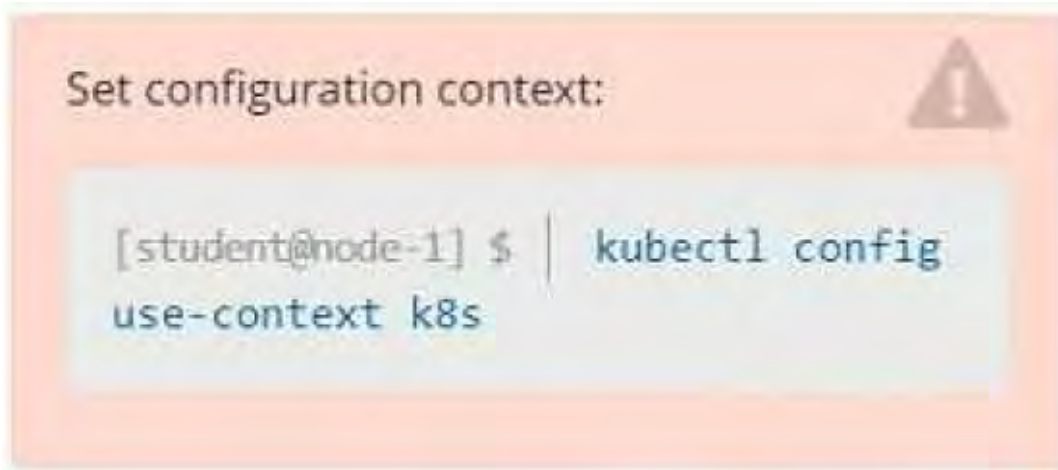
Context

Developers occasionally need to submit pods that run periodically.

Task

Follow the steps below to create a pod that will start at a predetermined time and]which runs to completion only once each time it is started:

• Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that runs the following shell command: date in a single busybox container. The command should run every minute and must complete within 22 seconds or be terminated oy Kubernetes. The Cronjob namp and container name should both be hello

• Create the resource in the above manifest and verify that the job executes successfully at least once

Answer: See the solution below.
Explanation:
Solution:

Question No : 10 CORRECT TEXT