

Exam Questions CKAD

Certified Kubernetes Application Developer (CKAD) Program

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



NEW QUESTION 1

Exhibit:



Context

It is always useful to look at the resources your applications are consuming in a cluster. Task

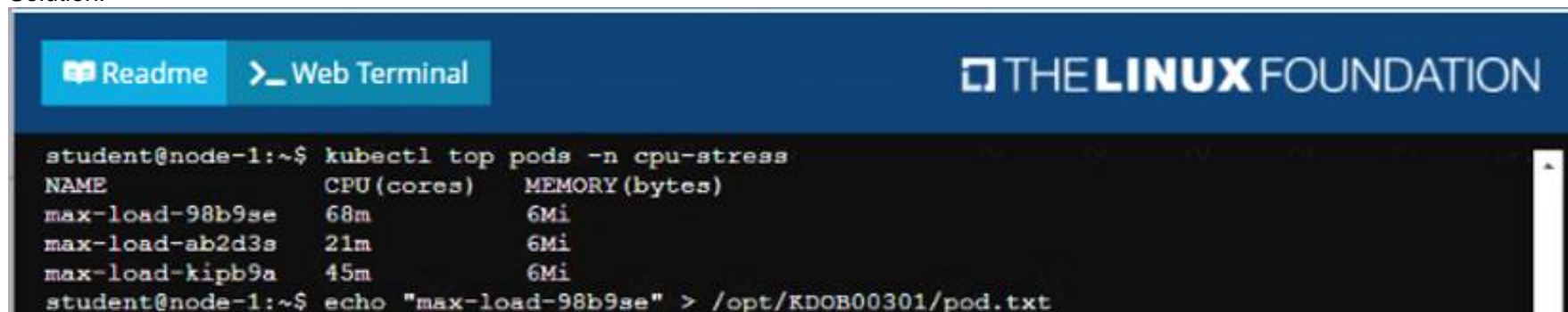
- From the pods running in namespacecpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030I/pod.txt, which has already been created.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:



NEW QUESTION 2

Exhibit:



Task

Create a new deployment for running.nginx with the following parameters;

- Run the deployment in the kdpd00201 namespace. The namespace has already been created
 - Name the deployment frontend and configure with4replicas
 - Configure the pod with a container image of lfccncl/nginx:1.13.7
 - Set an environment variable of NGINX PORT=8080and also expose that port for the container above Answer:
- See the solution below.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Readme Web Terminal THE LINUX FOUNDATION

```
student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
```

Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: api
  name: api
  namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        resources: {}
status: {}


~
"nginx_deployment.yml" 25L, 421C 4,1 All
```

Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: api
  name: api
  namespace: kdpd00201
spec:
  replicas: 4
  selector:
    matchLabels:
      app: api
  template:
    metadata:
      labels:
        app: api
    spec:
      containers:
      - image: lfcncf/nginx:1.13.7-alpine
        name: nginx
        ports:
        - containerPort: 8080
      env:
      - name: NGINX_PORT
        value: "8080"

~
23,8 All
```


Readme
Web Terminal



```

student@node-1:~$ kubectl create deployment api --image=lfcncf/nginx:1.13.7-alpine --replicas=4
-n kdpd00201 --dry-run=client -o yaml > nginx_deployment.yml
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create nginx_deployment.yml
Error: must specify one of -f and -k

error: unknown command "nginx_deployment.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_deployment.yml
error: error validating "nginx_deployment.yml": error validating data: ValidationError(Deployment.spec.template.spec): unknown field "env" in io.k8s.api.core.v1.PodSpec; if you choose to ignore these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_deployment.yml
student@node-1:~$ kubectl create -f nginx_deployment.yml
deployment.apps/api created
student@node-1:~$ kubectl get pods -n kdpd00201
NAME                READY   STATUS    RESTARTS   AGE
api-745677f7dc-7hnmv 1/1     Running   0           13s
api-745677f7dc-9q5vp 1/1     Running   0           13s
api-745677f7dc-fd4gk 1/1     Running   0           13s
api-745677f7dc-mbnpc 1/1     Running   0           13s
student@node-1:~$

```

NEW QUESTION 3

Exhibit:



Context

You sometimes need to observe a pod's logs, and write those logs to a file for further analysis. Task

Please complete the following;

- Deploy the counter pod to the cluster using the provided YAMLSpec file at /opt/KDOB00201/counter.yaml
- Retrieve all currently available application logs from the running pod and store them in the file /opt/KDOB00201/log_Output.txt, which has already been created

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

student@node-1:~$ kubectl create -f /opt/KDOB00201/counter.yaml
pod/counter created
student@node-1:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
counter             1/1     Running   0           10s
liveness-http       1/1     Running   0           6h45m
nginx-101            1/1     Running   0           6h46m
nginx-configmap      1/1     Running   0           107s
nginx-secret         1/1     Running   0           7m21s
poller              1/1     Running   0           6h46m
student@node-1:~$ kubectl logs counter
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$

```

```

student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat /opt/KDOB00201/log_output.txt

```

The screenshot shows a web terminal window with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active, and the terminal output shows a list of 24 hexadecimal strings, each preceded by a number from 1 to 24. The strings are: 1: 2b305101817ae25ca60ae46510fb6d11, 2: 3648cf2eae95ab680dba8f195f891af4, 3: 65c8bbd4dbf70bf81f2a0984a3a44ede, 4: 40d3a9c8e46f5533bb4828fbe5c8d038, 5: 390442d2530a90c3602901e3fe999ac8, 6: b71d95187417e139effb33af77681040, 7: 66a8e55a6491e756d2d0549ad6ab90a7, 8: ff2b3d583b64125d2f9129c443bb37ff, 9: b6c6a12b6e77944ed8baaaf6c242dae4, 10: bfcc9a894a0604fc4b814b37d0a200a4, 11: 5493cd16a1790a5fb9512b0c9d4c5dd1, 12: 03f169e93e6143438e6dfe4ecb3cc9ed, 13: 764b37fe611373c42d0b47154041f6eb, 14: 1a56fbc1896b0ee6394136166281839e, 15: ecc492eb17715de090c47345a98d98d3, 16: 7974a6bec0fb44b6b8bbfc71aa3fbe74, 17: 9ae01bef01748b12cc9f97a5f9f72cd6, 18: 23fb22ee34d4272e4c9e005f1774515f, 19: ec7e1a5d314da9a0ad45d53be5a7acae, 20: 0bccdd8ee02cd42029e8162cd1c1197c, 21: d6851ea43546216b95bcb81ced997102, 22: 7ed9a38ea8bf0d86206569481442af44, 23: 29b8416ddc63dbfcb987ab3c8198e9fe, 24: 1f2062001df51a108ab25010f506716f. The terminal prompt is 'student@node-1:~\$'.

NEW QUESTION 4

Exhibit:



Task

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

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

The screenshot shows a web terminal window with a dark background. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The 'Web Terminal' tab is active, and the terminal output shows the command 'kubectl run nginx-resources -n pod-resources --image=nginx --dry-run=client -o yaml > nginx_resources.yml' being entered. The prompt is 'student@node-1:~\$'. The command is followed by 'vim nginx_'. The terminal prompt is 'student@node-1:~\$'.

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
"nginx_resources.yml" 16L, 289C
1,1
All

```

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources:
      requests:
        cpu: 200m
        memory: "1Gi"
~
~
~
~
~
~
~
~
~
~
-- INSERT --
15,22
All

```

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

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

```

Readme
Web Terminal
THE **LINUX** FOUNDATION

```

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

```

NEW QUESTION 5

Exhibit:



Context

A pod is running on the cluster but it is not responding. Task

The desired behavior is to have Kubemetes restart the pod when an endpoint returns an HTTP 500 on the

/healthz endpoint. The service, probe-pod, should never send traffic to the pod while it is failing. Please complete the following:

- The application has an endpoint, /started, that will indicate if it can accept traffic by returning an HTTP 200. If the endpoint returns an HTTP 500, the application has not yet finished initialization.
- The application has another endpoint /healthz that will indicate if the application is still working as expected by returning an HTTP 200. If the endpoint returns an HTTP 500 the application is no longer responsive.
- Configure the probe-pod pod provided to use these endpoints
- The probes should use port 8080

A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:

apiVersion:v1 kind:Pod metadata: labels: test:liveness

name:liveness-exec

spec: containers:

-name:liveness

image:k8s.gcr.io/busybox args:

- /bin/sh

- -c

- touch/tmp/healthy;sleep30;rm-rf/tmp/healthy;sleep600

livenessProbe: exec: command:

- cat

- /tmp/healthy initialDelaySeconds:5 periodSeconds:5

In the configuration file, you can see that the Pod has a single Container. The periodSeconds field specifies that the kubelet should perform a liveness probe every 5 seconds. The initialDelaySeconds field tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the command cat /tmp/healthy in the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy. If the command returns a non-zero value, the kubelet kills the container and restarts it.

When the container starts, it executes this command:

/bin/sh -c"touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600"

For the first 30 seconds of the container's life, there is a /tmp/healthy file. So during the first 30 seconds, the command cat /tmp/healthy returns a success code.

After 30 seconds, cat /tmp/healthy returns a failure co

Create the Pod:

kubectl apply -f <https://k8s.io/examples/pods/probe/exec-liveness.yaml> Within 30 seconds, view the Pod events:

kubectl describe pod liveness-exec

The output indicates that no liveness probes have failed yet:

FirstSeen LastSeen Count From SubobjectPath Type Reason Message

24s 24s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "k8s.gcr.io/busybox" 23s 23s 1 {kubelet worker0} spec.containers{liveness}

Normal Pulled Successfully pulled image

"k8s.gcr.io/busybox"

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]

23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e

After 35 seconds, view the Pod events again: kubectl describe pod liveness-exec

At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

FirstSeen LastSeen Count From SubobjectPath Type Reason Message

37s 37s 1 {default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0

36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "k8s.gcr.io/busybox" 36s 36s 1 {kubelet worker0} spec.containers{liveness}

Normal Pulled Successfully pulled image

"k8s.gcr.io/busybox"

36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]

36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e

2s 2s 1 {kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open '/tmp/healthy': No such file or directory

Wait another 30 seconds, and verify that the container has been restarted: kubectl get pod liveness-exec

The output shows that RESTARTS has been incremented: NAME READY STATUS RESTARTS AGE

liveness-exec 1/1 Running 1 1m

NEW QUESTION 6

Exhibit:



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

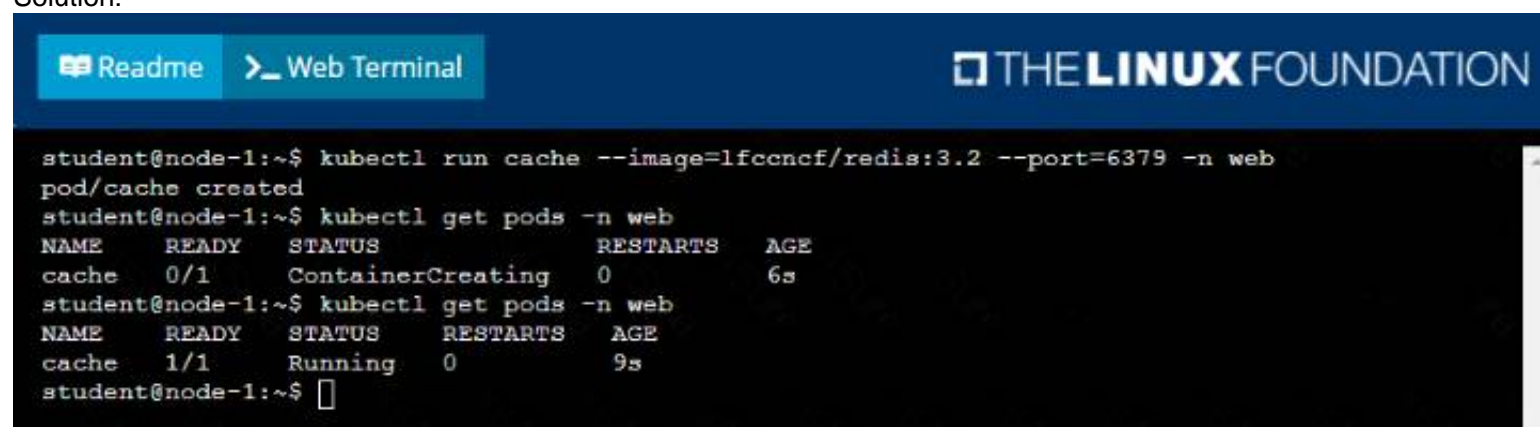
A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:



NEW QUESTION 7

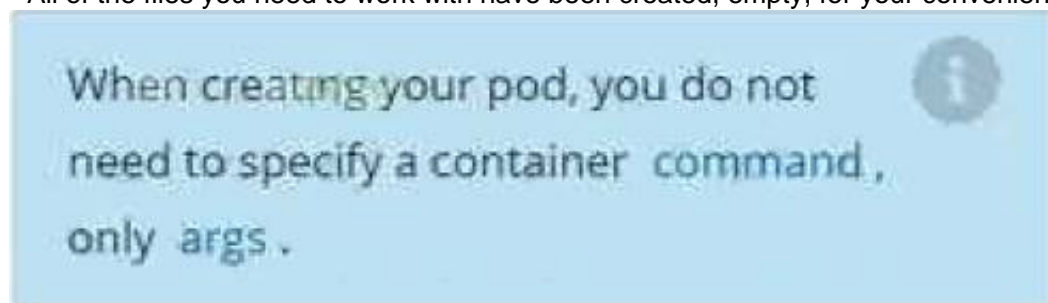
Context

Anytime a team needs to run a container on Kubernetes they will need to define a pod within which to run the container.

Task

Please complete the following:

- Create a YAML formatted pod manifest /opt/KDPD00101/pod1.yml to create a pod named app1 that runs a container named app1cont using image lfcncf/arg-output with these command line arguments: -lines 56 -F
- Create the pod with the kubectl command using the YAML file created in the previous step
- When the pod is running display summary data about the pod in JSON format using the kubectl command and redirect the output to a file named /opt/KDPD00101/out1.json
- All of the files you need to work with have been created, empty, for your convenience



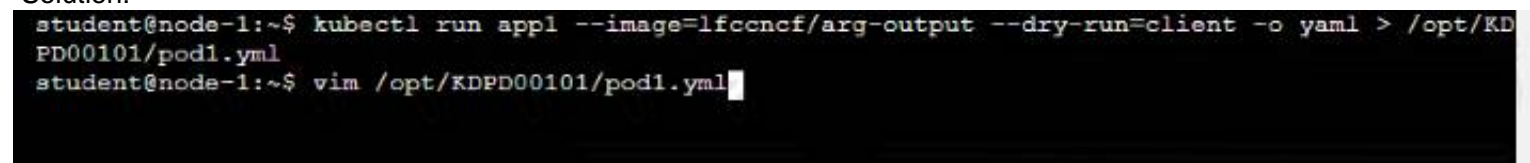
A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:



The image shows a terminal window with a dark background and light blue text. At the top, there are two tabs: 'Readme' and 'Web Terminal'. The top right corner features the 'THE LINUX FOUNDATION' logo. The main content of the terminal is a Kubernetes manifest file. The manifest defines a Pod with a container named 'appl' using the 'lfcncf/arg-output' image. The terminal also shows a cursor at the end of the 'args' line. The bottom right corner of the terminal displays '11,30' and 'All'.

```
pod/app1 created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
app1          0/1     ContainerCreating   0           5s
counter       1/1     Running             0           4m44s
liveness-http 1/1     Running             0           6h50m
nginx-101     1/1     Running             0           6h51m
nginx-configmap 1/1     Running             0           6m21s
nginx-secret  1/1     Running             0           11m
poller        1/1     Running             0           6h51m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
app1          1/1     Running   0           26s
counter       1/1     Running   0           5m5s
liveness-http 1/1     Running   0           6h50m
nginx-101     1/1     Running   0           6h51m
nginx-configmap 1/1     Running   0           6m42s
nginx-secret  1/1     Running   0           12m
poller        1/1     Running   0           6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```

```

Readme Web Terminal

nginx-configmap 1/1 Running 0 6m2
nginx-secret 1/1 Running 0 11m
poller 1/1 Running 0 6h5
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json >

```

```

Readme Web Terminal THE LINUX FOUNDATION

poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json > /opt/KDPD00101/out1.json
student@node-1:~$
student@node-1:~$

```

NEW QUESTION 8

Exhibit:



Context

A project that you are working on has a requirement for persistent data to be available. Task

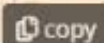
To facilitate this, perform the following tasks:

- 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 PersistentVolume.
- Create a PersistentVolumeClaim 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 PersistentVolumeClaim 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:

```

Readme Web Terminal THE LINUX FOUNDATION

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

```

```

Readme Web Terminal THE LINUX FOUNDATION

* 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:~$

```

```

Readme Web Terminal THE LINUX FOUNDATION

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

```


Readme

Web Terminal

THE LINUX FOUNDATION

```

-- INSERT --
0,1
All

```

Readme

Web Terminal

THE LINUX FOUNDATION

```

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

```

Readme

Web Terminal

THE LINUX FOUNDATION

```

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                STORAGECLASS  AGE
task-pv-volume      1Gi       RWO           Retain          Bound   default/task-pv-claim  storage        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

```

ReadmeWeb Terminal

THE LINUX FOUNDATION

```
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
```

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

ReadmeWeb Terminal

THE LINUX FOUNDATION

```
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 9

Exhibit:



Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount. Task
Please complete the following:

- Create a ConfigMap named another-config containing the key/value pair: key4/value3
- start a pod named nginx-configmap containing a single container using the nginx image, and mount the key you just created into the pod under directory /also/a/path

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:


Solution:

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

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
    - image: nginx
      name: nginx-configmap
      volumeMounts:
        - name: myvol
          mountPath: /also/a/path
  volumes:
    - name: myvol
      configMap:
        name: another-config
```

```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME          DATA   AGE
another-config 1       5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml ^C
student@node-1:~$ mv nginx_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$
```


Readme
Web Terminal



```

student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.con
ainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create -f nginx_configmap.yml
pod/nginx-configmap created
student@node-1:~$ kubectl get pods
NAME                READY   STATUS             RESTARTS   AGE
liveness-http       1/1     Running            0           6h44m
nginx-101            1/1     Running            0           6h45m
nginx-configmap      0/1     ContainerCreating  0           5s
nginx-secret         1/1     Running            0           5m39s
poller               1/1     Running            0           6h44m
student@node-1:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
liveness-http       1/1     Running   0           6h44m
nginx-101            1/1     Running   0           6h45m
nginx-configmap      1/1     Running   0           8s
nginx-secret         1/1     Running   0           5m42s
poller               1/1     Running   0           6h45m
student@node-1:~$ l

```

NEW QUESTION 10

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

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

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

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

Money Back Guarantee

CKAD Practice Exam Features:

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