

## CKAD Dumps

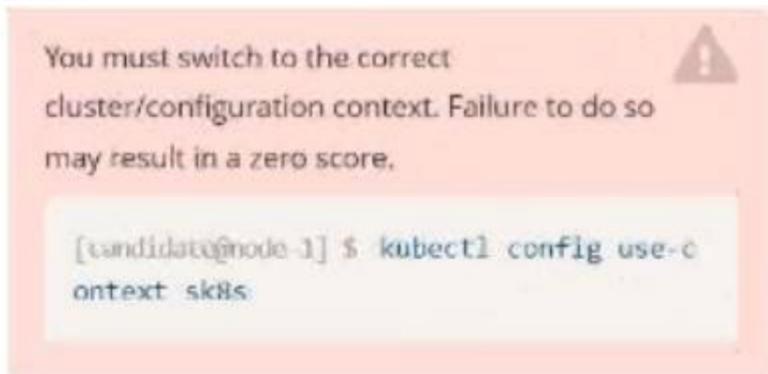
### Certified Kubernetes Application Developer (CKAD) Program

<https://www.certleader.com/CKAD-dumps.html>

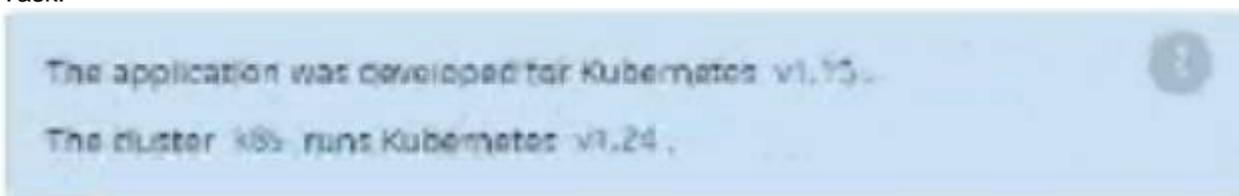


NEW QUESTION 1

Exhibit:



Task:



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
candidate@node-1:~$ kubectl config use-context sk8s  
Switched to context "sk8s".  
candidate@node-1:~$ vim ~/credible-mite/www.yaml
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help  
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: {}
```

Text Description automatically generated

```

File Edit View Terminal Tabs Help
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-fl7j             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 2**

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:

```

Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl run cache --image=lfcncf/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 3**

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

When creating your pod, you do not need to specify a container command, only args.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

**Solution:**

```
student@node-1:~$ kubectl run appl --image=lfcncf/arg-output --dry-run=client -o yaml > /opt/KDPD00101/pod1.yml
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```

Readme
THE **LINUX** FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: appl
  name: appl
spec:
  containers:
  - image: lfcncf/arg-output
    name: appl
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}

~/opt/KDPD00101/pod1.yml 15L, 242C 3,1 All
```

Readme
THE **LINUX** FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: appl
  name: appl
spec:
  containers:
  - image: lfcncf/arg-output
    name: appl
    args: ["--lines", "56", "--f"]

~/opt/KDPD00101/pod1.yml 11,30 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 4**

Exhibit:



Task:

A Dockerfile has been prepared at `~/humane-stork/build/Dockerfile`

Multiple image builders and tools have been pre-installed at 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

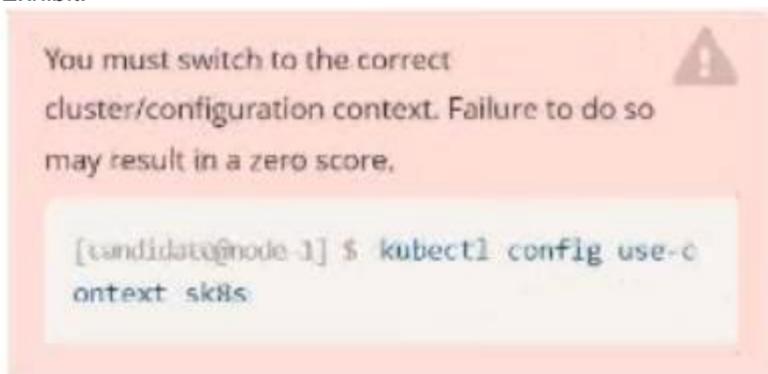
**Explanation:**

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 5**

Exhibit:



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:

```
File Edit View Terminal Tabs Help
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

```
File Edit View Terminal Tabs Help
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 --
```

```
File Edit View Terminal Tabs Help
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 6**

Exhibit:



Context

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

Task

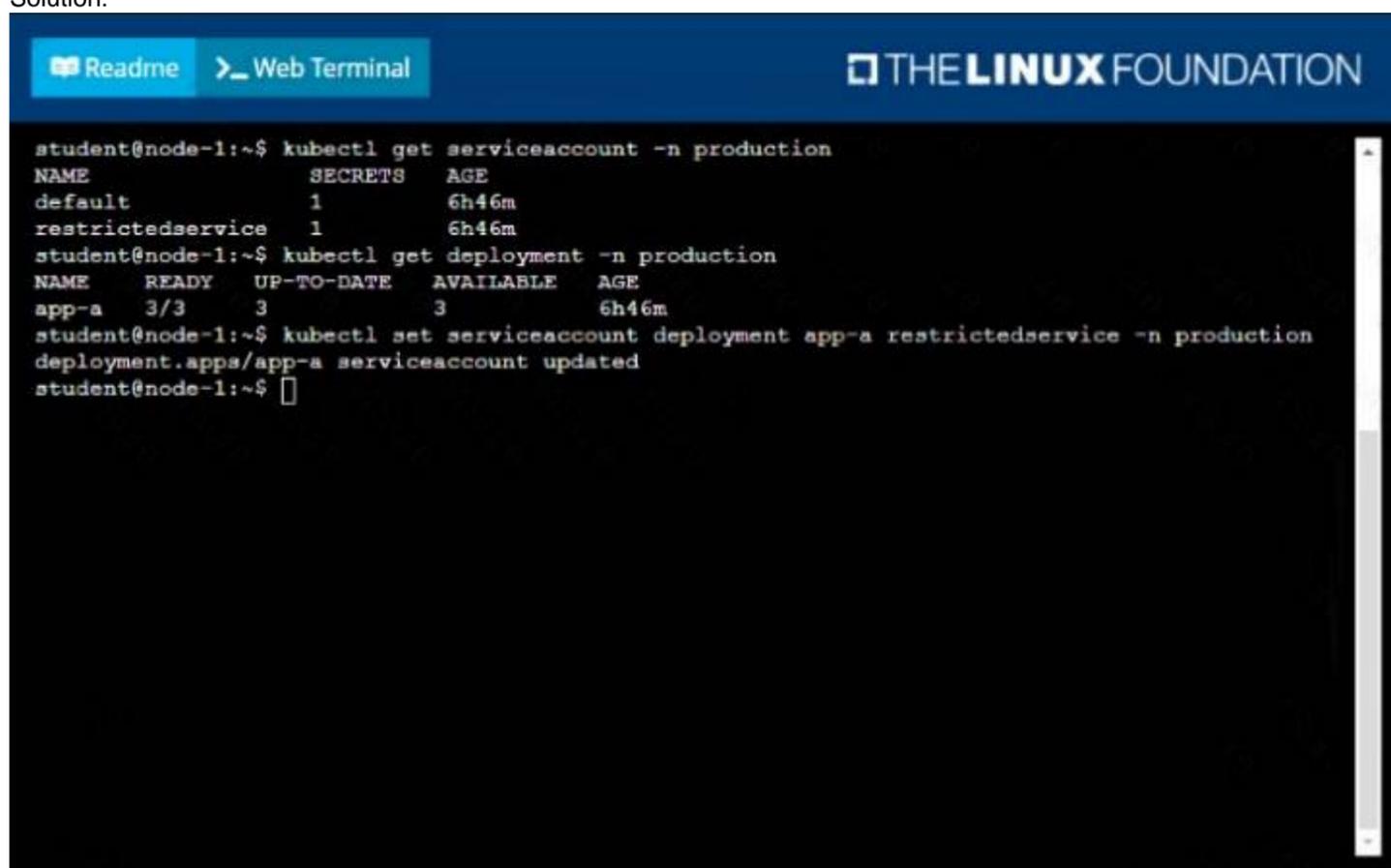
Update the app-a deployment in the production namespace to run as the restrictedservice service account. The service account has already been created.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:



**NEW QUESTION 7**

Exhibit:



Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port 90. As this port changes to 5050 an additional container needs to be added to the poller pod which adapts the container to connect to this new port. This should be realized as an ambassador container within the pod.

Task

- Update the nginxsvc service to serve on port 5050.
- Add an HAProxy container named haproxy bound to port 90 to the poller pod and deploy the enhanced pod. Use the image haproxy and inject the configuration located at /opt/KDMC00101/haproxy.cfg, with a ConfigMap named haproxy-config, mounted into the container so that haproxy.cfg is available at /usr/local/etc/haproxy/haproxy.cfg. Ensure that you update the args of the poller container to connect to localhost instead of nginxsvc so that the connection is correctly proxied to the new service endpoint. You must not modify the port of the endpoint in poller's args . The spec file used to create the initial poller pod is available in /opt/KDMC00101/poller.yaml

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

apiVersion: apps/v1 kind: Deployment metadata:

name: my-nginx spec:

selector:

matchLabels: run: my-nginx replicas: 2 template: metadata: labels:

run: my-nginx spec: containers:

- name: my-nginx image: nginx ports:

- containerPort: 90

This makes it accessible from any node in your cluster. Check the nodes the Pod is running on: `kubectl apply -f ./run-my-nginx.yaml`

`kubectl get pods -l run=my-nginx -o wide`

NAME READY STATUS RESTARTS AGE IP NODE

my-nginx-3800858182-jr4a2 1/1 Running 0 13s 10.244.3.4 kubernetes-minion-905m my-nginx-3800858182-kna2y 1/1 Running 0 13s 10.244.2.5 kubernetes-

minion-ljyd Check your pods' IPs:

`kubectl get pods -l run=my-nginx -o yaml | grep podIP` podIP: 10.244.3.4

podIP: 10.244.2.5

**NEW QUESTION 8**

Exhibit:



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 a deployment in namespace `kdsn00101` a service that accomplishes the following:

- Exposes the service on TCP port 8080
- is mapped to the pods defined by the specification of `kdsn00101-deployment`
- Is of type `NodePort`
- Has a name of `cherry`

- A. Mastered
- B. Not Mastered

**Answer: A**

**Explanation:**

Solution:



```

Readme Web Terminal THE LINUX FOUNDATION
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/kubect1-edit-d4y5r.yaml" 70L, 1957C 1,1 Top

```

```

Readme Web Terminal THE LINUX FOUNDATION
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 9**

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 with 4 replicas
- Configure the pod with a container image of lfcncf/nginx:1.13.7
- Set an environment variable of NGINX PORT=8080 and also expose that port for the container above

- A. Mastered
- B. Not Mastered

Answer: A

**Explanation:**

Solution:

```

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
    
```

```

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
    
```

```

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 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
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(Deploymen
t.spec.template.spec): unknown field "env" in io.k8s.api.core.v1.PodSpec; if you choose to ignor
e 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-7hnvm 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 10**

.....

## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your CKAD Exam with Our Prep Materials Via below:**

<https://www.certleader.com/CKAD-dumps.html>