CKAD Q&As

Certified Kubernetes Application Developer (CKAD) Program

Pass Linux Foundation CKAD Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.certbus.com/ckad.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

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 lfccncf/nginx:1.13.7 Set an environment variable of NGINX__PORT=8080 and also expose that port for the container above

A. Please check explations

B. Place Holder

Correct Answer: AB

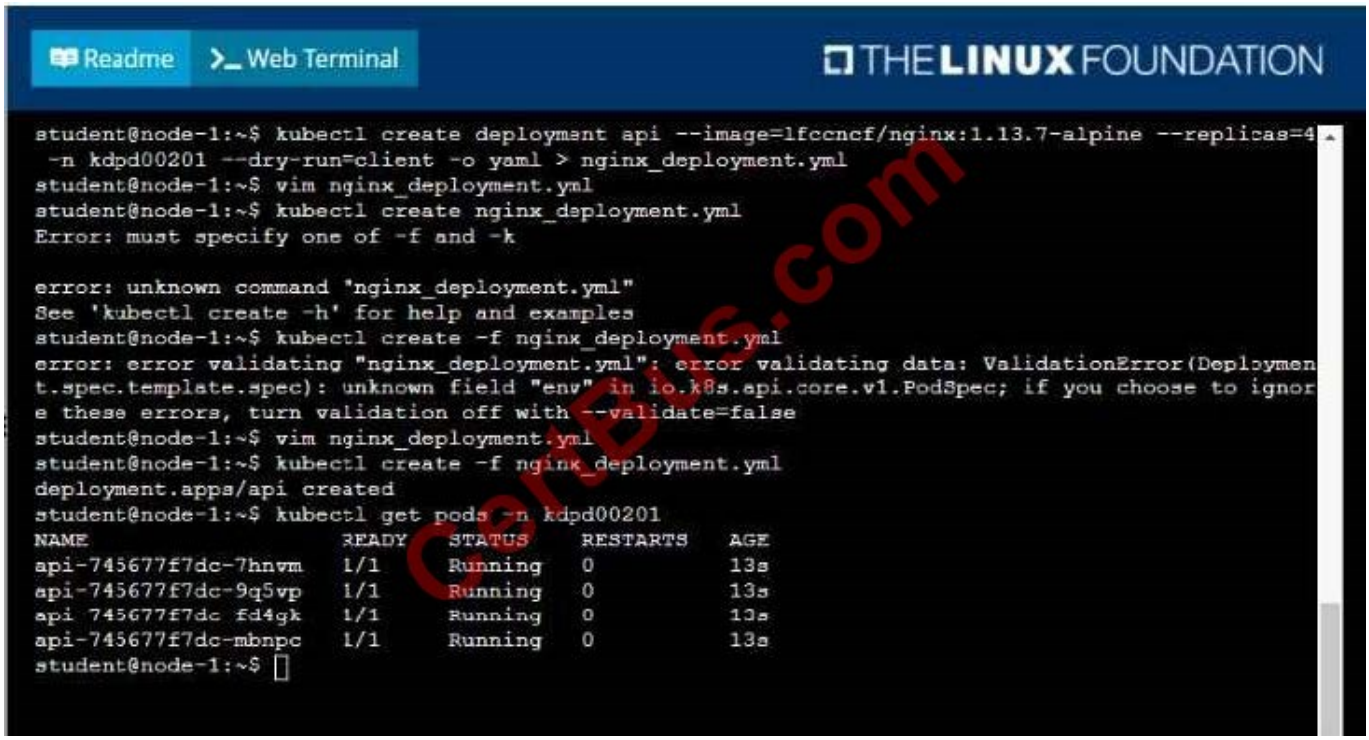
Solution:

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: lfccncf/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: lfccncf/nginx:1.13.7-alpine
        name: nginx
        ports:
        - containerPort: 8080
        env:
        - name: NGINX_PORT
          value: "8080"
~
                                                    23,8          All
```



QUESTION 2

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 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. Please check explations

B. Place Holder

Correct Answer: AB

Solution:

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

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: "2020-10-09T08:53:38Z"
  generation: 1
  labels:
    app: nginx
  name: kdsn00101-deployment
  namespace: kdsn00101
  resourceVersion: "4186"
  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
```

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
```

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/podl.yml to create a pod named app1 that runs a container named app1cont using image lfccncf/arg-output with these command line arguments: -lines 56 -F Create the pod with the kubect1 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 kubect1 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. Please check explations

B. Place Holder

Correct Answer: AB

Solution:

```
student@node-1:~$ kubectl run app1 --image=lfccncf/arg-output --dry-run=client -o yaml > /opt/KD
PD00101/pod1.yml
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: app1
  name: app1
spec:
  containers:
  - image: lfccncf/arg-output
    name: app1
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"/opt/KDPD00101/pod1.yml" 15L, 242C                                  3,1           All
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: app1
  name: app1
spec:
  containers:
  - image: lfccncf/arg-output
    name: app1
    args: ["--lines","56","-F"]
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                     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:~$
```

QUESTION 4

Exhibit:

Set configuration context:

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

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.

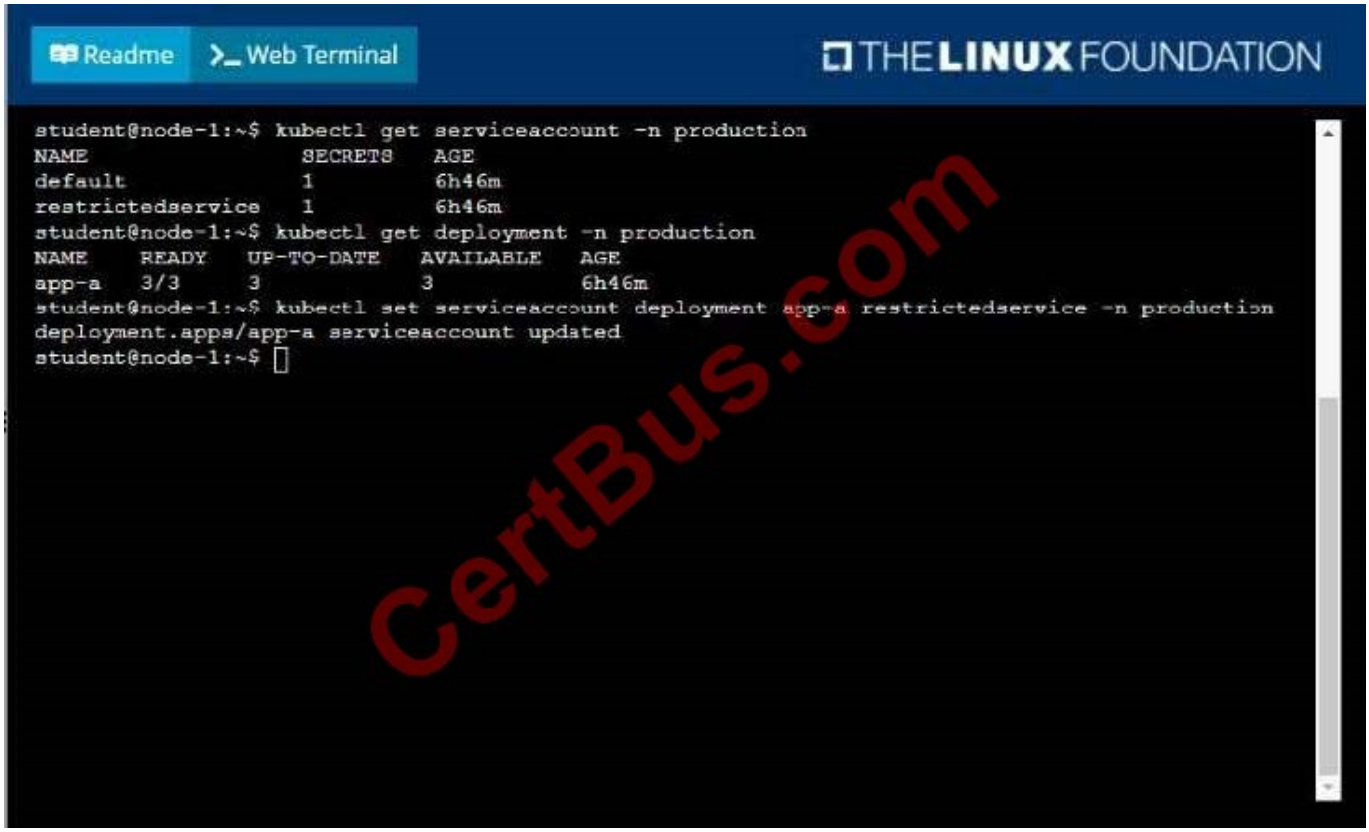
The service account has already been created.

A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:



QUESTION 5

Exhibit:

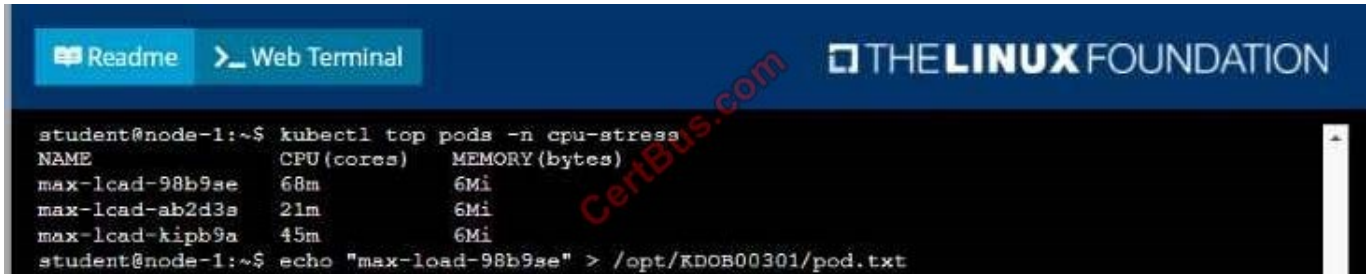
Context It is always useful to look at the resources your applications are consuming in a cluster. Task From the pods running in namespace cpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030l/pod.txt, which has already been created.

A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:



QUESTION 6

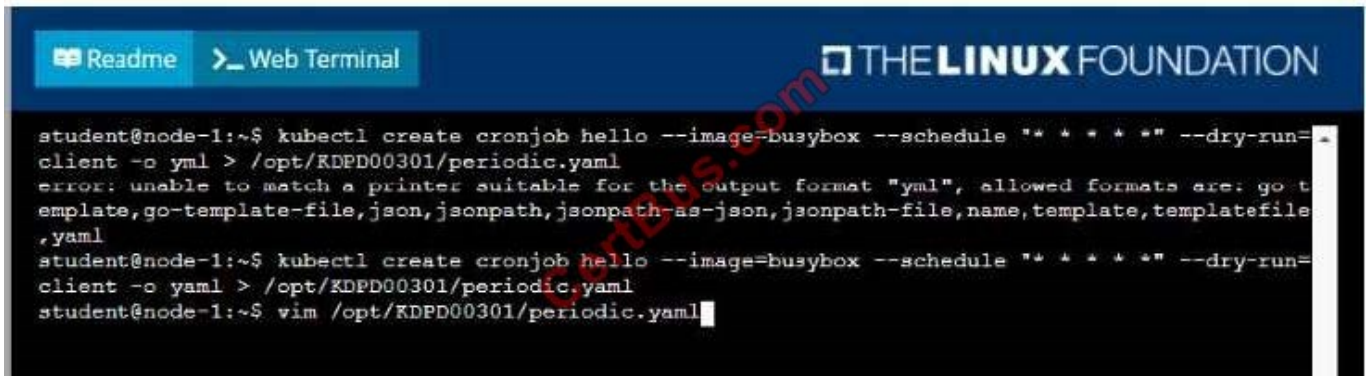
Exhibit: Context Developers occasionally need to submit pods that run periodically. Task Follow the steps below to create a pod that will start at a predetermined time and]which runs to completion only once each time it is started: Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml 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

A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:



Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
          - image: busybox
            name: hello
            args: ["/bin/sh","-c","date"]
          restartPolicy: Never
  schedule: '*/1 * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow
~
~
~
~
~
~
~
                                                    19,26         All
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
student@node-1:~$ kubectl create -f /opt/KDPD00301/periodic.yaml
cronjob.batch/hello created
student@node-1:~$ kubectl get cronjob
NAME    SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello   */1 * * * *   False     0        <none>          6s
student@node-1:~$
```

QUESTION 7

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 sk8
s-node-0
```

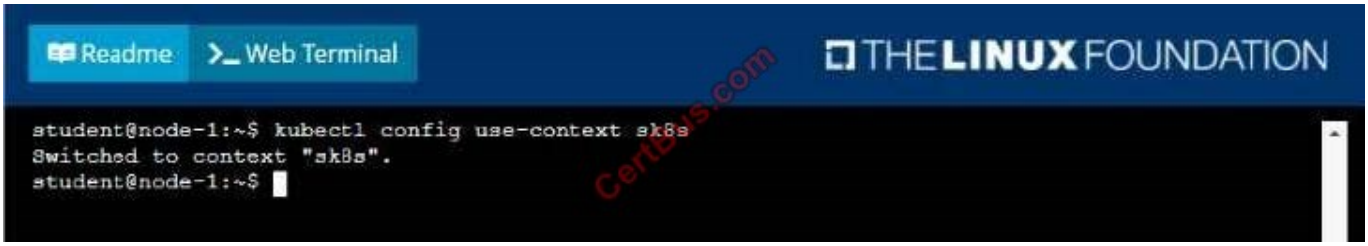
Ensure that you return to the base node (with hostname node-1) once you have completed your work on sk8s-node-0 copy

A. Please check explations

B. Place Holder

Correct Answer: AB

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                   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
```

Readme | Web 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
~
~
~
~
~
~
~
~
~
                                                    17,32          All
```

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

Readme | Web 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:~$
```

QUESTION 8

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. Please check explations

B. Place Holder

Correct Answer: AB

Pending

QUESTION 9

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. Please check explations

B. Place Holder

Correct Answer: AB

Solution:

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:~$
```

QUESTION 10

Exhibit:

Set configuration context:

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


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. Please check explations

B. Place Holder

Correct Answer: AB

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-nginx3800858182-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

QUESTION 11

Exhibit:

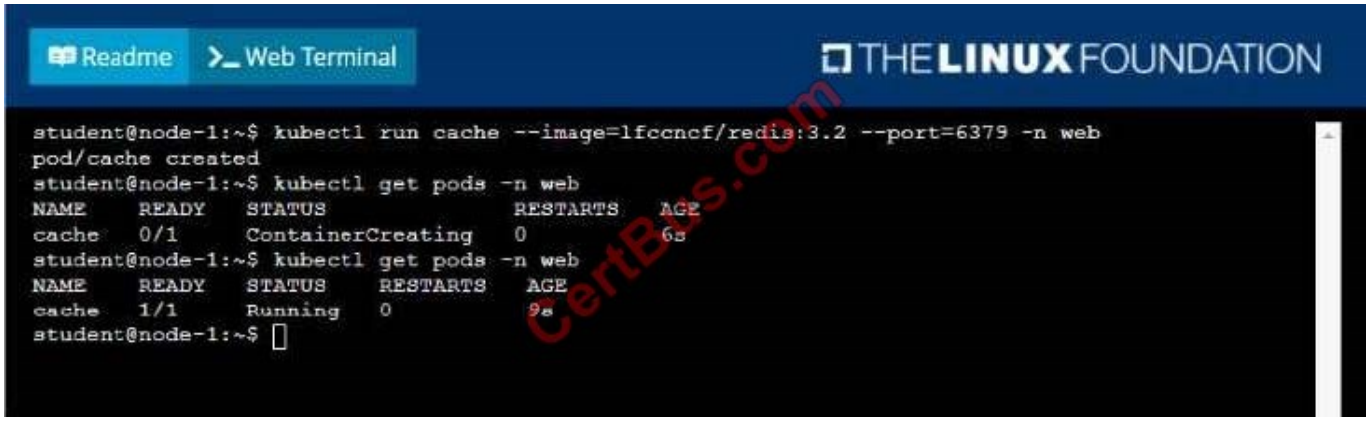
Context A web application requires a specific version of redis to be used as a cache. Task Create a pod with the following characteristics, and leave it running when complete: The pod must run in the web namespace. The namespace has already been created The name of the pod should be cache Use the lfccncf/redis image with the 3.2 tag Expose port 6379

A. Please check explations

B. Place Holder

Correct Answer: AB

Solution:



QUESTION 12

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. Please check explations

B. Place Holder

Correct Answer: AB

Solution: 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 nonzero 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 code. 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

```yaml
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; sleep 30; rm -rf /tmp/healthy; sleep 600
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
```