Linux Foundation

CKA

Certified Kubernetes Administrator

QUESTION & ANSWERS

QUESTION 1

Create PersistentVolume named task-pv-volume with storage 10Gi, access modes ReadWriteMany, storageClassName manual, and volume at /mnt/data and Create a PersistentVolumeClaim of at least 3Gi storage and access mode ReadWriteOnce and verify

Correct Answer: See the solution below.

Explanation/Reference:

```
vim task-pv-volume.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
name: task-pv-volume
labels:
type: local
spec:
storageClassName: manual
capacity:
storage: 10Gi
accessModes:
- ReadWriteMany
hostPath:
path: "/mnt/data"
kubectl apply -f task-pv-volume.yaml
//Verify
kubectl get pv
vim task-pvc-volume.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: task-pv-claim
spec:
storageClassName: manual
accessModes:
- ReadWriteMany
resources:
requests:
storage: 3Gi
kubectl apply -f task-pvc-volume.yaml
//Verify
Kuk kubectl get pvc
```

QUESTION 2

Allow traffic from all the pods in "web" namespace and from pods
with label "type=monitoring" to the pods matching label "app: db"

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl create namespace web
kubectl label namespace/web app=web
vim web-allow-all-ns-monitoring.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: web-allow-all-ns-monitoring
namespace: default
spec:
podSelector:
matchLabels:
app: db
ingress:
- from:
- namespaceSelector:
matchLabels:
app: web
podSelector:
matchLabels:
type: monitoring
k kubectl apply -f web-allow-all-ns-monitoring.yaml
```

QUESTION 3

Get list of persistent volumes and persistent volume claim in
the cluster

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl get pv kubectl get pvc
```

QUESTION 4

Create a NetworkPolicy which denies all ingress traffic

Correct Answer: See the solution below.

Explanation/Reference:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: default-deny
spec:
podSelector: {}
policyTypes:
- Ingress
```

QUESTION 5

List pod logs named “frontend” and search for the pattern “started” and write it to a file “/opt/error-logs”

Correct Answer: See the solution below.

Explanation/Reference:

```
Kubectl logs frontend | grep -i “started” > /opt/error-logs
```

QUESTION 6

Check logs of each container that “busyboxpod-{1,2,3}”

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl logs busybox -c busybox-container-1
kubectl logs busybox -c busybox-container-2
kubectl logs busybox -c busybox-container-3
```

QUESTION 7

Undo the deployment to the previous version 1.17.1 and verify Image has the previous version

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl rollout undo deploy webapp
kubectl describe deploy webapp | grep Image
```

QUESTION 8

Create an nginx pod and list the pod with different levels of verbosity

Correct Answer: See the solution below.

Explanation/Reference:

```
// create a pod
kubectl run nginx --image=nginx --restart=Never --port=80
// List the pod with different verbosity
kubectl get po nginx --v=7
kubectl get po nginx --v=8
kubectl get po nginx --v=9
```

QUESTION 9

Create the service as type NodePort with the port 32767 for the nginx pod with the pod selector app: my-nginx

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl run nginx --image=nginx --restart=Never --
```

```
labels=app=nginx --port=80 --dry-run -o yaml > nginx-pod.yaml
```

QUESTION 10

Create the nginx pod with version 1.17.4 and expose it on port 80

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl run nginx --image=nginx:1.17.4 --restart=Never --port=80
```

QUESTION 11

Create a Pod nginx and specify both CPU, memory requests and limits together and verify.

Correct Answer: See the solution below.

Explanation/Reference:

```
kubectl run nginx-request --image=nginx --restart=Always --dryrun -o yaml > nginx-request.yml
// add the resources section and create
apiVersion: v1
kind: Pod
metadata:
labels:
run: nginx
name: nginx-request
spec:
containers:
- image: nginx
name: nginx
resources:
requests:
memory: "100Mi"
cpu: "0.5"
limits:
memory: "200Mi"
cpu: "1"
restartPolicy: Always
k kubectl apply -f nginx-request.yaml
// Verify
```

Kubectl top po