GENUINEDUMPS

Linux Foundation

CKA

Certified Kubernetes Administrator

QUESTION & ANSWERS

Question #:1

List all the pods showing name and namespace with a json path expression

See the solution below.

Explanation

kubectl get pods -o=jsonpath="{.items[*]['metadata.name',

'metadata.namespace']}"

Question #:2

Annotate the pod with name=webapp

See the solution below.

Explanation

kubectl annotate pod nginx-dev-pod name=webapp

kubectl annotate pod nginx-prod-pod name=webapp

// Verify

kubectl describe po nginx-dev-pod | grep -i annotations

kubectl describe po nginx-prod-pod | grep -i annotations

Question #:3

Create a redis pod, and have it use a non-persistent storage

Note: In exam, you will have access to kubernetes.io site,

Refer : https://kubernetes.io/docs/tasks/configure-pod-container/configurevolume-storage/

See the solution below.

Explanation

apiVersion: v1

kind: Pod

metadata:

name: redis

spec:

containers:

- name: redis

image: redis

volumeMounts:

- name: redis-storage

mountPath: /data/redis

ports:

- containerPort: 6379

volumes:

- name: redis-storage

emptyDir: {}

Question #:4

Get all the pods with label “env”

See the solution below.

Explanation

kubectl get pods -L env

Question #:5

Create a NetworkPolicy which denies all ingress traffic

See the solution below.

Explanation

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: default-deny

spec:

podSelector: {}

policyTypes:

- Ingress

Question #:6

Create an nginx pod with container Port 80 and it should only receive traffic only it checks the endpoint / on port 80 and verify and delete the pod.

See the solution below.

Explanation

kubectl run nginx --image=nginx --restart=Never --port=80 --

dry-run -o yaml > nginx-pod.yaml

// add the readinessProbe section and create

vim nginx-pod.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

run: nginx

name: nginx

spec:

containers:

- image: nginx

name: nginx

ports:

- containerPort: 80

readinessProbe:

httpGet:

path: /

port: 80

restartPolicy: Never

kubectl apply -f nginx-pod.yaml

// verify

kubectl describe pod nginx | grep -i readiness

kubectl delete po nginx

Question #:7

Create a busybox pod and add “sleep 3600” command

See the solution below.

Explanation

kubectl run busybox --image=busybox --restart=Never -- /bin/sh -c

"sleep 3600"

Question #:8

Create a ETCD backup of kubernetes cluster

Note : You don’t need to memorize command, refer -

https://kubernetes.io/docs/tasks/administer-cluster/configureupgrade-etcd/ during exam

See the solution below.

Explanation

ETCDCTL_API=3 etcdctl --endpoints=[ENDPOINT] --cacert=[CA CERT]

--cert=[ETCD SERVER CERT] --key=[ETCD SERVER KEY] snapshot save

[BACKUP FILE NAME]

In exam, cluster setup is done with kubeadm , this means ETCD

used by the kubernetes cluster is coming from static pod.

kubectl get pod -n kube-system

kubectl describe pod etcd-master -n kube-system

You can locate the information on

endpoint: — advertise-client-urls=https://172.17.0.15:2379

ca certificate: — trusted-cafile=/etc/kubernetes/pki/etcd/ca.crt

server certificate : — certfile=/etc/kubernetes/pki/etcd/server.crt

key: — key-file=/etc/kubernetes/pki/etcd/server.key

To Create backup

export ETCDCTL_API=3

(or)

ETCDCTL_API=3 etcdctl ETCDCTL_API=3 etcdctl --

endpoints=https://172.17.0.15:2379 --

cacert=/etc/kubernetes/pki/etcd/ca.crt --

cert=/etc/kubernetes/pki/etcd/server.crt --

key=/etc/kubernetes/pki/etcd/server.key snapshot save etcdsnapshot.db

//Verify

ETCDCTL_API=3 etcdctl --write-out=table snapshot status

snapshot.db

Question #:9

Install a kubernetes cluster with one master and one workerusing kubeadm

See the solution below.

Explanation

This is a straightforward question, you need to install kubernetes cluster using kubeadm with one master and

one worker.

Installation is considered success once both master and worker

nodes become available.

Refer : https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/

Question #:10

Deploy a pod with image=redis on a node with label disktype=ssd

See the solution below.

Explanation

// Get list of nodeskubectl get nodes//Get node with the label disktype=ssdkubectl get no -l disktype=ssd// Create a sample yaml filekubectl run node-redis --generator=run-pod/v1 --image=redis --dryrun -o yaml > test-redis.yaml// Edit test-redis.yaml file and add nodeSelectorvim test-redis.yamlapiVersion: v1kind: Podmetadata:name: redisspec:nodeSelector:disktype: ssdcontainers:- name: node-redisimage: redisimagePullPolicy: IfNotPresentkubectl apply -f test-redis.yaml/ // VerifyK kubectl get po -o wide