# Mirantis CKA Exam

**Volume: 89 Questions + 17 Labs**

Question: 1
Containers are run on which of these?
Choose the correct answer:

A. Services

B. Controllers

C. Nodes

D. None of these

Answer: C

Explanation
Nodes run the pods.

Question: 2
Kubernetes changed the name of cluster members to "Nodes." What were they called before that?
Choose the correct answer:

A. Workers

B. Cogs

C. Minions

D. Slaves

Answer: C

Explanation
A lot of documentation and tutorials online still refer to worker nodes this way.

Question: 3
Unique IP addresses are assigned to:
Choose the correct answer:

A. NAT is used extensively, so unique IP addresses are irrelevant

B. Pods

C. Container Hosts

D. Containers

Answer: B

Explanation
A pod gets assigned a single IP address, regardless of how many containers make it up. This is analogous to many services running on a single virtual machine.

Question: 4
Usually, when submitting a Kubernetes API call, data is sent in which format? (Select all that apply)
Choose the 2 correct answers:

A. YAML

B. XML

C. DOC

D. JSON

Answer: AD

Explanation
If using a direct API call in an application, JSON is used. If using kubectl to submit a request, it takes YAML.

Question: 5
Which of these are not inherently created by Kubernetes?
Choose the correct answer:

A. Services

B. Nodes

C. Controllers

D. Pods

Answer: B

Explanation
Nodes are added to a cluster, and a Kubernetes object is created to reflect them, but Kubernetes itself doesn't create them.

Question: 6
Communications between the apiserver and the kubelet on the cluster nodes are used for all but which of the following?
Choose the correct answer:

A. Providing the kubelet's port-forwarding capability

B. Fetching logs for pods

C. Keep-alive xml packets

D. Attaching (through kubectl) to running pods

Answer: C

Explanation
Communications between the apiServer and the Kubelet are constantly communicating for a variety of purposes.

Question: 7
The connection between the apiserver and nodes, pods and services:
Choose the correct answer:

A. Is unencrypted and therefore unsafe to run over public networks.

B. Is always encrypted with IPSec.

C. Is always encrypted using the method configured in the .kube file.

D. Is currently encrypted with IPSec with plans to allow other encryption plugins later.

Answer: A

Explanation
It's a fairly simple process to encrypt the streams using TLS.

Question: 8
If memory is running low on a running node, which of these keys will return "True"?
Choose the correct answer:

A. OOM

B. Warning

C. MemoryPressure

D. LowMemory

Answer: C

Explanation
MemoryPressure and DiskPressure return true as a node starts to become overcommitted.

Question: 9
What does a pod represent in a Kubernetes cluster?
Choose the correct answer:

A. A running process

B. Conditions under which applications will autoscale

C. A set of rules for maintaining high availability

D. All the containers in the cluster

Answer: A

Explanation
Pods are the running containers in a Kubernetes cluster.

Question: 10

Which of these components mount volumes to containers?
Choose the correct answer:

A. kube-proxy

B. fluentd

C. kubelet

D. kube-scheduler

Answer: C

Explanation
The kubelet which runs on nodes handles moment-to-moment management of the pods on its node.

Question: 11
What is the difference between a Docker volume and a Kubernetes volume?
Choose the correct answer:

A. Proximity: In Docker, volumes can reside on the same host with their containers. In Kubernetes, they must reside on separate metal for resiliency.

B. Back-end Drivers. Docker supports more block storage types than Kubernetes does.

C. Size: Docker volumes are limited to 3TB. Kubernetes volumes are limited to 16TB.

D. Volume lifetimes. In Docker, this is loosely defined. In Kubernetes, the volume has the same lifetime as its surrounding pod.

Answer: D

Explanation
Docker volumes are not used in conjunction with Kubernetes at this time.

Question: 12
In a typical deployment, the Kubernetes Master listens on what port number?
Choose the correct answer:

A. 22

# Mirantis CKA Exam

## LAB1. Deploying Your Cluster

Solution:

First, create a master server using the "Cloudnative Kubernetes" engine. Once this machine has booted, log in to it, change the password, and then start the deployment.
K8s requires a pod network to function. We are going to use Flannel, so we need to pass in a flag to the deployment script so K8s knows how to configure itself:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

This command might take a fair amount of time to complete, possibly as much as ten minutes. Once it's complete, make note of the join command output by kubeadm init that looks like this:

```
kubeadm join --token --discovery-token-ca-cert-hash sha256:
```

You will run that command on the other nodes to allow them to join the cluster -- but not quite yet. We'll get to that soon.
Create a directory:

```
mkdir -p $HOME/.kube
```

Next, you'll move the configuration files to a location usable by your local user. if you copy and paste these commands, do so one at a time, or your sudo password prompt might cause things to go slightly wrong and you might have to will be wrong and you might have to redo it.

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

In order for your pods to communicate with one another, you'll need to install pod networking. We are going to use Flannel for our Container Network Interface (CNI) because it's easy to install and reliable. Enter this command:

```
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/v0.9.1/Documentati
on/kube-flannel.yml
```

Next, you'll check to make sure everything is coming up properly.

```
kubectl get pods --all-namespaces
```

Once the kube-dns-xxxx containers are up, your cluster is ready to accept worker nodes. Create three or so worker nodes the same way you created your master nodes -- by bringing up the "Cloudnative Kubernetes" image on your Cloud Servers tab above.
ssh to each of the other nodes in the cluster, and execute the kubeadm join command you noted earlier. You will need execute this command with root privileges, so be sure to add "sudo" to the

beginning of the command in order for it to complete correctly. Once this command is issued, you may log out of the node. Kubernetes will configure it for you from this point on.
See the video "Setting Up Your Cluster" in this course for details and a full walkthrough of the process.
On the master, you can watch the node come up by repeatedly running:

```
kubtctl get nodes
```

### LAB2. Run a Job

Applications that run to completion inside a pod are called "jobs." This is useful for doing batch processing.
Most Kubernetes objects are created using yaml. Here is some sample yaml for a job which uses perl to calculate pi to 2000 digits and then stops.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
      restartPolicy: Never
  backoffLimit: 4
```

Create this yaml file on your master node and call it "pi-job.yaml". Run the job with the command:

```
kubectl create -f pi-job.yaml
```

Task:

# Mirantis CKA Exam

1. Check the status of the job using the kubectl describe command.
2. When the job is complete, view the results by using the kubectl logs command on the appropriate pod.
3. Write yaml for a new job. Use the image "busybox" and have it sleep for 10 seconds and then complete. Run your job to be sure it works.

Solution:

1. The full command is kubectl describe job pi
2. The previous command will give you the name of the pod associated with the job, which you will need to pass into the kubectl logs command.
For example: (the precise code will vary)

```
$ kubectl describe job pi

Name:           pi

Namespace:      default

Selector:       controller-uid=7ffe0296-f7ad-11e7-8717-0abccbe536d6

Labels:         controller-uid=7ffe0296-f7ad-11e7-8717-0abccbe536d6

                job-name=pi

Annotations:

Parallelism:    1

Completions:    1

Start Time:     Fri, 13 Apr 2018 15:30:20 +0000

Pods Statuses:  0 Running / 1 Succeeded / 0 Failed

Pod Template:

  Labels:  controller-uid=7ffe0296-f7ad-11e7-8717-0abccbe536d6

           job-name=pi

  Containers:

   pi:

    Image:  perl

    Port:

    Command:
```

```
      perl
      -Mbignum=bpi
      -wle
      print bpi(2000)
    Environment:
    Mounts:
  Volumes:
Events:
  Type    Reason           Age   From           Message
  ----    ------           ----  ----           -------
  Normal  SuccessfulCreate  4m    job-controller  Created pod: pi-fmctx


$ kubectl logs pi-fmctx
```

3. The yaml could vary in a couple of ways, but here is an example solution:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox
spec:
  template:
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["sleep", "10"]
      restartPolicy: Never
  backoffLimit: 4
```

# Mirantis CKA Exam

## LAB3. Deploy a Pod

Pods usually represent running applications in a Kubernetes cluster. Here is an example of some yaml which defines a pod:

```
apiVersion: v1
kind: Pod
metadata:
  name: alpine
  namespace: default
spec:
  containers:
  - name: alpine
    image: alpine
    command:
      - sleep
      - "3600"
    imagePullPolicy: IfNotPresent
  restartPolicy: Always
```

Task:
1. Looking at the yaml, describe what the pod will do.
2. Run the pod.
3. Delete the pod.
4. Write yaml for a pod that runs the nginx image.
5. Run your yaml to ensure it functions as expected.
Delete any user pods you created during this lab.

Solution:

1. This pod will cause the alpine linux container to sleep for 3600 seconds (1 hour) and then exit. Kubernetes will then restart the pod.
2. If the yaml is named alpine.yaml then the command is kubectl create -f alpine.yaml
3. There are a few ways to accomplish this.
Use the file method: kubectl delete -f alpine.yaml
Use the object method: kubectl delete pod alpine or kubectl delete pod/alpine
4. There are many possibilities, but here is yaml that satisfies the exercise: