

Exam Session - Cert Prep: Certified Kubernetes Administrator (CKA) Exam

 cloudacademy.com/quiz/exam/3761541/results

#1

An intern in your team is trying to create a serviceAccount but he is having trouble getting the process to work. To be able to help him, you ask him to show you how he is creating the account. His process looks like this:

```
$ cat > /tmp/serviceaccount.yaml <<EOF
apiVersion: v1
metadata: name: build-robot
EOF
$ kubectl create -f /tmp/serviceaccount.yaml
```

How could the intern fix the problem?

✗

The kubectl line should have the "-t serviceAccount" parameter

✗

He should use kubelet instead of kubectl

✓

He is missing a line with the following string, right below the apiVersion line: kind: ServiceAccount

✗

The file should be in JSON format, not in YAML format

Explanation

You can create serviceAccounts like this:

```
$ cat > /tmp/serviceaccount.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: build-robot
EOF
$ kubectl create -f /tmp/serviceaccount.yaml
serviceaccounts/build-robot
```

 <https://kubernetes.io/docs/user-guide/service-accounts/>

#2

You need to update operating system packages, including Kubernetes packages, on the servers running your Kubernetes cluster. You have created a plan defining the order and how many nodes are allowed to be updated at a time. You need to make sure that pods are gracefully scheduled onto new nodes when a node is updated. Which command will you use before updating a node, and which command will you use after updates are complete? (Choose 2 answers)



`kubectldrain <node_name>`



`kubectluncordon <node_name>`



`kubectlcordon <node_name>`



`sudo kubeadm upgrade node config`

Explanation

To update operating system packages on a node in a Kubernetes cluster, you should first drain the node using `kubectldrain <node_name>`. Draining gracefully evicts pods running on the node by sending a TERM (terminate) signal to the main process in each container in the pods. It also prevents any new pods from being scheduled onto the node so you can update the system. Once the updates are complete, you can allow pods to be scheduled onto the node by issuing `kubectluncordon <node_name>`.

[!\[\]\(6a9b39b98eb945faa14c645ec99e4eaa_img.jpg\)/lab/create-manage-kubernetes-cluster-scratch/upgrading-kubernetes-clusters-kubeadm/](#)

[Covered in this lecture](#)

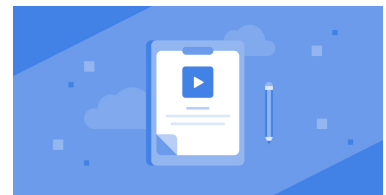
[Create K8s API Resources](#)

[Course:Deploying A Cloud Native Application into Kubernetes](#)

3m



#3



What is the core grouping primitive in Kubernetes?



Key-value pair



Namespace



Annotation



Label selector

Explanation

Unlike names and UIDs, labels do not provide uniqueness, so in general, we expect many objects to carry the same label(s). The label selector is the core grouping primitive in Kubernetes, and allows the the client/user to identify a set of objects.

 <https://kubernetes.io/docs/user-guide/labels/>

#4

Your DevOps manager has asked you to manually create a service account named jenkins. Which command could you use?



kubectl config set-credentials -name jenkins -create



kubelet config set-credentials -name jenkins -create -i yaml



kubectl create serviceaccount jenkins



kubelet create serviceaccount jenkins -i yaml

Explanation

Service account bearer tokens are perfectly valid to use outside the cluster and can be used to create identities for long standing jobs that wish to talk to the Kubernetes API. To manually create a service account, simply use the `kubectl create serviceaccount (NAME)` command. This creates a service account in the current namespace and an associated secret. For example:

```
$ kubectl create serviceaccount jenkins
```

<https://kubernetes.io/docs/admin/authentication/>

#5

You have created a service for two nginx replicas using the `kubectl create -f` command, with the following yml: `nginx-svc.yml` `apiVersion: v1` `kind: Service` `metadata: name: my-nginx` `labels: run: my-nginx` `spec: ports: - port: 80` `protocol: TCP` `selector: run: my-nginx` What is another approach you could take to accomplish the same result? (Assume that you have already created a deployment named "my-nginx" with these specifications.)

✗

Use the Kubernetes API to create the two replicas (via the `createReplicatedService` command)

✗

Use the Kubernetes API to create the two replicas (via the `createService` command)

✓

Run the command `kubectl expose deployment my-nginx`

✗

Run the command `kubectl createService my-nginx --replicas=2`

Explanation

You can create a Service for your two nginx replicas with `kubectl expose`:

This is equivalent to **`kubectl create -f`** the following yml:

<https://kubernetes.io/docs/user-guide/connecting-applications/>

Covered in this lecture

kubectl Tips

Course:Administering Kubernetes Clusters

17m



#6



You've been asked to debug a pod that is crashing. What command would be particularly helpful during this process?

✗

`kubectl create -f`



kubectl get nodes -o



kubectl logs



docker pull

Explanation

What to do when a pod is crashing or otherwise unhealthy?

First, take a look at the logs of the current container:

```
$ kubectl logs ${POD_NAME} ${CONTAINER_NAME}
```

If your container has previously crashed, you can access the previous container's crash log with:

```
$ kubectl logs --previous ${POD_NAME} ${CONTAINER_NAME}
```

Alternately, you can run commands inside that container with exec:

```
$ kubectl exec ${POD_NAME} -c ${CONTAINER_NAME} -- ${CMD} ${ARG1} ${ARG2} ... ${ARGN}
```

 <https://kubernetes.io/docs/user-guide/debugging-pods-and-replication-controllers/>

Covered in this lecture

Networking

Course:Kubernetes Patterns for Application Developers

11m



#7



What is one way through which users can access the Kubernetes API?



kubectl



Java API



ssh



gRPC requests

Explanation

Users access the API using kubectl, client libraries, or by making REST requests. Both human users and Kubernetes service accounts can be authorized for API access.

 <https://kubernetes.io/docs/admin/accessing-the-api/>

#8

What are normal users within Kubernetes?



Users managed by an outside service



Users stored in a file of usernames and passwords



Regular system users




Users managed by the Kubernetes API

Explanation

All Kubernetes clusters have two categories of users: service accounts managed by Kubernetes, and normal users.

Normal users are assumed to be managed by an outside, independent service. An admin distributing private keys, a user store like Keystone or Google Accounts, even a file with a list of usernames and passwords. In this regard, *Kubernetes does not have objects which represent normal user accounts*. Regular users cannot be added to a cluster through an API call.

 <https://kubernetes.io/docs/admin/authentication/>

#9

Your CTO has requested that you start backing up your Kubernetes cluster state on a daily basis. You prefer to use Kubernetes to accomplish the task. What kind of Kubernetes resource will you use, and what command will run in a container to backup the data?



Deployment and rsync



CronJob and etcdctl



CronJob and rsync



Deployment and etcdctl

Explanation

Kubernetes CronJob resources can run a job on a periodic schedule. The etcdctl command can connect to secured etcd clusters and backup the cluster state using the snapshot save command.

 [/lab/create-manage-kubernetes-cluster-scratch/backing-and-restoring-kubernetes-clusters/](#)

#10

What is a Kubernetes cluster service discovery/DNS provider that can be configured automatically by kubeadm?



CoreDNS



OpenDNS



BIND



GDDNS

Explanation

Kubeadm will install CoreDNS from version 1.11, when CoreDNS reach general availability (GA).

 [/lab/create-manage-kubernetes-cluster-scratch/upgrading-kubernetes-clusters-kubeadm/](#)

#11

_____ is responsible for implementing a form of virtual IP for Services of type other than ExternalName.



kube-proxy



kubectl



kubeadm



minikube

Explanation

Every node in a Kubernetes cluster runs a kube-proxy. kube-proxy is responsible for implementing a form of virtual IP for Services of type other than ExternalName. In Kubernetes v1.0 the proxy was purely in userspace. In Kubernetes v1.1 an iptables proxy was added, but was not the default operating mode. Since Kubernetes v1.2, the iptables proxy is the default.

 [https://kubernetes.io/docs/concepts/services-networking/service/#headless-services](#)

#12

Your company is using pods to co-locate a content management system with a data loader and a local cache manager. A new team member suggests running these programs in a single Docker container instead. You explain to him that Kubernetes recommend using pods because it _____.



allows individual containers to be versioned, rebuilt, and redeployed independently



reduces latency when serving user requests



allows software dependencies to be tightly coupled



prevents scheduler and node failures

Explanation

Running multiple programs in a single (Docker) container is not recommended for the reasons below.

1. Transparency: Making the containers within the pod visible to the infrastructure enables the infrastructure to provide services to those containers, such as process management and resource monitoring. This facilitates a number of conveniences for users.
2. Decoupling software dependencies: The individual containers may be versioned, rebuilt and redeployed independently. Kubernetes may even support live updates of individual containers someday.
3. Ease of use: Users don't need to run their own process managers, worry about signal and exit-code propagation, etc.
4. Efficiency: Because the infrastructure takes on more responsibility, containers can be lighter weight.

Affinity-based co-scheduling of containers is also not recommended because although this approach would provide co-location, it would not provide most of the benefits of pods, such as resource sharing, IPC, guaranteed fate sharing, and simplified management.

 <https://kubernetes.io/docs/user-guide/pods/>

#13

You have decided to use Calico as your CNI network plugin for a new Kubernetes cluster. The decision to use Calico was primarily motivated by your requirements for using network policies. You are using kubeadm to create the cluster. Which command-line option must you specify to kubeadm init?



--pod-network-cidr



--service-cidr

✗


--ignore-preflight-errors

✗

--apiserver-advertise-address

Explanation

The --pod-network-cidr option specifies the range of IP addresses for the pod network. The addresses must match the Calico IP Pools to have Network Policy function correctly.

 [/lab/create-manage-kubernetes-cluster-scratch/installing-kubeadm-and-its-dependencies/](#)
#14

What is the difference between user accounts and service accounts?

✗

User accounts are namespaced. Service accounts are intended to be global.

✗

Service accounts are synchronized to complex business processes. User account creation is more lightweight.

✓

User accounts are for humans. Service accounts are for processes, which run in pods.

✗

Service accounts are associated with shared team roles. User accounts are for individual processes.

Explanation

Kubernetes distinguished between the concept of a user account and a service accounts for a number of reasons:

- User accounts are for humans. Service accounts are for processes, which run in pods.
- User accounts are intended to be global. Names must be unique across all namespaces of a cluster. Service accounts are namespaced.

- Typically, a cluster's User accounts might be synced from a corporate database, where new user account creation requires special privileges and is tied to complex business processes. Service account creation is intended to be more lightweight, allowing cluster users to create service accounts for specific tasks (i.e., principle of least privilege).
- Auditing considerations for humans and service accounts may differ.
- A config bundle for a complex system may include definition of various service accounts for components of that system. Because service accounts can be created ad-hoc and have namespaced names, such config is portable.

 <https://kubernetes.io/docs/admin/service-accounts-admin/>

Covered in this lecture

Service Accounts

Course:Kubernetes Patterns for Application Developers

5m



#15



What does it mean to say that Kubernetes supports automatic resource bin packing?



It automatically places containers based on their resource requirements and other constraints.



It scales applications up and down based on CPU usage.



It automatically mounts the storage system of your choice.



It gives containers their own IP addresses and a single DNS name for a set of containers, and can load-balance across them.

Explanation

Kubernetes supports automatic bin packing, which means that it automatically places containers based on their resource requirements and other constraints.

 </blog/what-is-kubernetes/>

#16

You have been notified that the Kubernetes cluster that you administer has been behaving strangely. A team member has told you they suspect one of the cluster's master components is at fault. How can you quickly check on the status of the master components?



Check the output of `systemctl status` for each component



Check the output of `kubectl get componentstatuses`



Check the output of `kubeadm upgrade plan`



Check the output of `kubectl get node $master_node`

Explanation

The output of `kubectl get componentstatuses` tells you the status of the cluster's default scheduler, controller manager, and etcd data store. The status of API server is implicitly tested because `kubectl` will output an error message if the API server cannot be reached.



[/lab/create-manage-kubernetes-cluster-scratch/initializing-kuberernetes-master-node/](#)

#17

You administer a multi-node production Kubernetes cluster running on EC2 instances in AWS. A developer has informed you that a single pod application works correctly when using Minikube even if the pod is deleted and rescheduled, but appears to lose all of its data when the pod is rescheduled in production. What is a possible explanation for this?



The pod uses a `HostPath` persistent volume



The pod uses a persistent volume with `ReadWriteOnce` access mode



The pod's container is not running in privileged mode



The pod uses a `tmpfs` (RAM-backed filesystem) volume

Explanation

The HostPath persistent volume type should only be used on single node clusters, such as Minikube. If the pod is scheduled onto a different node the persistent volume will not find the data at the original node's host path.

The access mode doesn't effect the data's persistence.

Privileged mode doesn't impact the ability of a pod to mount a volume after being rescheduled.

A tmpfs volume could explain why data is deleted when the pod is rescheduled, but because the pod data persists on Minikube it can't be using a tmpfs volume.

[🔗/course/introduction-to-kubernetes/volumes/](/course/introduction-to-kubernetes/volumes/)

Covered in this lecture

Volumes

Course:Introduction to Kubernetes

13m



#18



When using Kubernetes, why is it recommended that you use labels?

✗

To attach arbitrary non-identifying metadata, for retrieval by API clients such as tools, libraries, etc.

✗

To use in environments where many users are spread across multiple teams, or projects.

✓

To enable users to map their own organizational structures onto system objects in a loosely coupled fashion, without requiring clients to store these mappings.

✗

To model an application-specific “logical host”.

Explanation

Labels enable users to map their own organizational structures onto system objects in a loosely coupled fashion, without requiring clients to store these mappings.

Service deployments and batch processing pipelines are often multi-dimensional entities (e.g., multiple partitions or deployments, multiple release tracks, multiple tiers, multiple micro-services per tier). Management often requires cross-cutting operations, which breaks encapsulation of strictly hierarchical representations, especially rigid hierarchies determined by the infrastructure rather than by users.

 <https://kubernetes.io/docs/user-guide/labels/>

#19

Which are the two primary modes of finding a Service with Kubernetes?

✗

API Calls and LDAP

✓

Environment variables and DNS

✗


Selectors and DHCP

✗

Labels and Radius

Explanation

Kubernetes supports two primary modes of finding a Service: environment variables and DNS. The Kubernetes DNS server is the only way to access services of type ExternalName.

 <https://kubernetes.io/docs/user-guide/services/>

Covered in this lecture

Rolling Updates and Rollbacks

Course: Introduction to Kubernetes

8m



#20



You have been pre-creating persistent volumes for a Kubernetes cluster, but as demand increases, you prefer to have persistent volumes automatically created whenever a persistent volume claim requests a persistent volume. What do you need to do to enable this kind of dynamic provisioning of persistent volumes in the cluster?



Create a StorageClass and ensure PersistentVolumeClaim resources refer to it



Ensure PersistentVolume resources are of type dynamic



Mount PersistentVolume resources using the dynamic access mode



Ensure the reclaim policy of PersistentVolume resources is set to automatic

Explanation

Dynamic provisioning of persistent volumes is made possible by creating storage classes and referring to a specific storage class in a persistent volume claim.

There is no dynamic type for PersistentVolume resources.

There is no dynamic access mode for PersistentVolume resources. Access mode relates to how many times a volume can be mounted for writing and reading.

There is no automatic reclaim policy for PersistentVolume resources. Possible values are retain, recycle, or delete. The reclaim policy relates to what happens to the PersistentVolume after it is released by a PersistentVolumeClaim.



[/lab/deploy-a-stateful-application-in-a-kubernetes-cluster/deploying-stateful-application-kubernetes-cluster/](#)

#21

Your DevOps manager wants you to implement horizontal pod Autoscaling, so that Kubernetes automatically scales the number of pods, based on observed CPU utilization. Which requirement is necessary for the autoscaler to work?



Configure the --horizontal-pod-autoscaler-sync-period to any value greater than 1 minute.



Deploy heapster monitoring or Metrics Server.



Shutdown the kubelet service.



Upgrade your pods to the latest version.

Explanation

The autoscaler is implemented as a control loop. It periodically queries CPU utilization for the pods it targets. (The period of the autoscaler is controlled by `--horizontal-pod-autoscaler-sync-period` flag of controller manager. The default value is 30 seconds). Then, it compares the arithmetic mean of the pod's CPU utilization with the target and adjust the number of replicas if needed.

CPU utilization is the recent CPU usage of a pod divided by the sum of CPU requested by the pod's containers. Please note that if some of the pod's containers do not have CPU request set, CPU utilization for the pod will not be defined and the autoscaler will not take any action. Further details of the autoscaling algorithm are given here.

The autoscaler uses metrics from heapster or Metrics Server to collect CPU utilization. Therefore, it is required to deploy heapster monitoring in your cluster for autoscaling to work.

The autoscaler accesses corresponding replication controller, deployment or replica set by scale sub-resource. Scale is an interface which allows to dynamically set the number of replicas and to learn the current state of them. More details on scale sub-resource can be found here.



<https://kubernetes.io/docs/user-guide/horizontal-pod-autoscaling/>

#22

Which command can you use to check whether a Service exists?



```
wget -q0- hostnames
```



```
kubectl get svc hostnames
```



```
echo $HOSTNAMES_SERVICE_HOST
```



```
nslookup hostnames.default.svc.cluster.local
```

Explanation

To check whether a Service actually exists, you can execute the following command:

```
$ kubectl get svc hostnames  
Error from server: service "hostnames" not found
```

<https://kubernetes.io/docs/user-guide/debugging-services/>

#23

Service accounts within Kubernetes are associated with _____.

✗

objects representing key pairs

✗

usernames representing "normal" users

✗

registered client certificates

✓

pods running in the clusters

Explanation

All Kubernetes clusters have two categories of users: service accounts managed by Kubernetes, and normal users.

In contrast, service accounts are users managed by the Kubernetes API. They are bound to specific namespaces, and created automatically by the API server or manually through API calls. Service accounts are tied to a set of credentials stored as **Secrets**, which are mounted into pods that allow cluster processes to talk to the Kubernetes API.

<https://kubernetes.io/docs/admin/authentication/>

Covered in this lecture

Serving - Creating and Configuring

Course: Introduction to Knative

2m



#24



You have a five Linux EC2 instances running in AWS. You are using kubeadm to create a single master cluster with four worker nodes. The output of `kubectl get nodes` only displays the master and three worker nodes. Which of the following commands could result in the final worker node being displayed in the list of nodes?



`kubeadm join`



`kubectl get nodes --all-namespaces`



`systemctl restart kubelet`



`kubectl uncordon`

Explanation

The join command initializes and joins a worker node to the cluster. The command should be executed on the instance that does not appear in the output of `kubectl get nodes`.

[/lab/create-manage-kubernetes-cluster-scratch/joining-worker-node-kubernetes-cluster/](#)

#25

To update a service without an outage, kubectl supports what is called ____.



high availability (HA) deployment



rolling update



automatic-hot failover



automatic-warm failover

Explanation

To update a service without an outage, kubectl supports what is called 'rolling update', which updates one pod at a time, rather than taking down the entire service at the same time. Note that kubectl rolling-update only supports Replication Controllers.

 <https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/>

Covered in this lecture

Rolling Updates and Rollbacks

Course: Introduction to Kubernetes

8m



#26



A colleague has asked for your recommendation on how/where to record or store client library information that is useful for debugging purposes, like name, version, and build info. What would you recommend?



Store it in a namespace.



Store it the pod logs.



Assign adequate annotations.



Define adequate labels.

Explanation

Annotations are useful to be able to attach arbitrary non-identifying metadata, for retrieval by API clients such as tools, libraries, etc. This information may be large, may be structured or unstructured, may include characters not permitted by labels, etc. Such information would not be used for object selection and therefore doesn't belong in labels.

Examples of such information include:

- fields managed by a declarative configuration layer, to distinguish them from client- and/or server-set default values and other auto-generated fields, fields set by auto-sizing/auto-scaling systems, etc., in order to facilitate merging

- build/release/image information (timestamps, release ids, git branch, PR numbers, image hashes, registry address, etc.)
- pointers to logging/monitoring/analytics/audit repos
- client library/tool information (e.g. for debugging purposes – name, version, build info)
- other user and/or tool/system provenance info, such as URLs of related objects from other ecosystem components
- lightweight rollout tool metadata (config and/or checkpoints)
- phone/pager number(s) of person(s) responsible, or directory entry where that info could be found, such as a team website

 <https://kubernetes.io/docs/user-guide/annotations/>

#27

You have noticed that you cannot connect to a service, and then begin a debugging process through which you notice the following: The connection is immediately dropped There are endpoints in the endpoints list The pod's containerPort matches up with the Service's containerPort What is a likely cause for this problem?



The right ports are not exposed.



An error in the pod description was silently ignored when the pod was created.



There was a failure to pull the image (due to, for example, using an incorrect image name).



The proxy can't contact the pod.

Explanation

If you can connect to the service, but the connection is immediately dropped, and there are endpoints in the endpoints list, it's likely that the proxy can't contact your pods.

There are three things to check:

- Are your pods working correctly? Look for restart count, and debug pods
- Can you connect to your pods directly? Get the IP address for the Pod, and try to connect directly to that IP
- Is your application serving on the port that you configured? Kubernetes doesn't do port remapping, so if your application serves on 8080, the containerPort field needs to be 8080.

<https://kubernetes.io/docs/user-guide/application-troubleshooting/>

Covered in this lecture

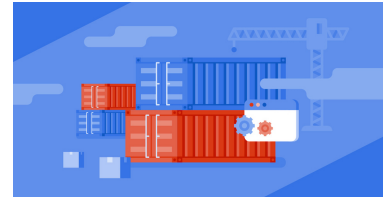
Deployments

Course: Introduction to Kubernetes

11m



#28



Which Kubernetes ServiceType leads to no proxying of any kind being set up in your cluster?



ClusterIP



NodePort



LoadBalancer



ExternalName

Explanation

Kubernetes ServiceTypes allow you to specify what kind of service you want. The default is ClusterIP. Type values and their behaviors are:

- ClusterIP: Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster. This is the default ServiceType.
- NodePort: Exposes the service on each Node's IP at a static port (the NodePort). A ClusterIP service, to which the NodePort service will route, is automatically created. You'll be able to contact the NodePort service, from outside the cluster, by requesting <NodeIP>:<NodePort>.
- LoadBalancer: Exposes the service externally using a cloud provider's load balancer. NodePort and ClusterIP services, to which the external load balancer will route, are automatically created.
- ExternalName: Maps the service to the contents of the externalName field (e.g. foo.bar.example.com), by returning a CNAME record with its value. No proxying of any kind is set up. This requires version 1.7 or higher of kube-dns.

<https://kubernetes.io/docs/concepts/services-networking/service/#headless-services>

#29

You would like to start sending traffic to a pod only when a probe succeeds. What is the best type of probe in this situation?



Use a LivenessProbe.



Specify a LivenessProbe and a RestartPolicy of Always or OnFailure.



Specify a ReadinessProbe.



Use a ReadinessProbe and a RestartPolicy of Always or OnFailure.

Explanation

If the process in your container is able to crash on its own whenever it encounters an issue or becomes unhealthy, you do not necessarily need a liveness probe; the kubelet will automatically perform the correct action in accordance with the RestartPolicy when the process crashes.

If you'd like your container to be killed and restarted if a probe fails, then specify a LivenessProbe and a RestartPolicy of Always or OnFailure.

If you'd like to start sending traffic to a pod only when a probe succeeds, specify a ReadinessProbe. In this case, the ReadinessProbe may be the same as the LivenessProbe, but the existence of the ReadinessProbe in the spec means that the pod will start without receiving any traffic and only start receiving traffic once the probe starts succeeding.

If a container wants the ability to take itself down for maintenance, you can specify a ReadinessProbe that checks an endpoint specific to readiness, which is different than the LivenessProbe.



<https://kubernetes.io/docs/user-guide/pod-states/>

#30

An intern had been given admin access to the Kubernetes cluster that you administer. They accidentally modified and deleted several Kubernetes resources. What command can you use to roll back the cluster to an earlier state?



etcdctl snapshot restore



kubectrl rollback



kubeadm reset



kubectrl restore

Explanation

The etcdctl snapshot restore command restores an etcd snapshot to an etcd directory, in effect restoring the state of the cluster to the condition it was in at the time the backup was created.



[/lab/create-manage-kubernetes-cluster-scratch/backing-and-restoring-kubernetes-clusters/](#)