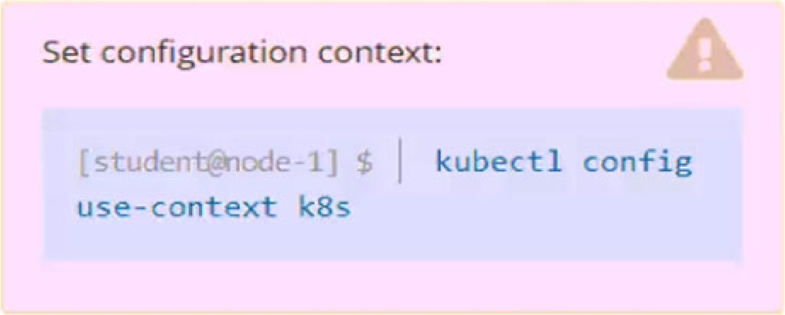
EXAMTOPICS
- Expert Verified, Online, **Free**.

Question #1 *Topic 1*

SIMULATION -



Set configuration context:

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

Context -

You have been asked to create a new ClusterRole for a deployment pipeline and bind it to a specific ServiceAccount scoped to a specific namespace.

Task -

Create a new ClusterRole named deployment-clusterrole, which only allows to create the following resource types:

✑ Deployment

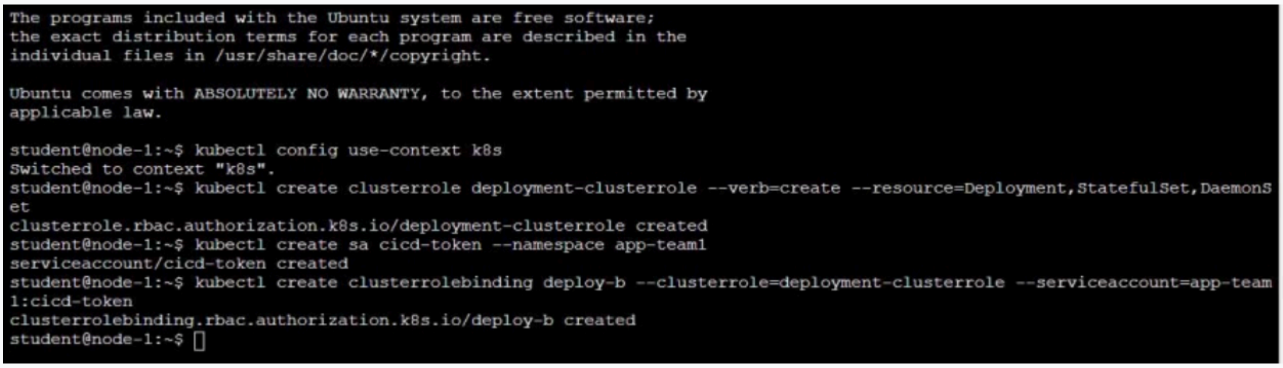
✑ Stateful Set

✑ DaemonSet

Create a new ServiceAccount named cicd-token in the existing namespace app-team1.

Bind the new ClusterRole deployment-clusterrole to the new ServiceAccount cicd-token, limited to the namespace app-team1.

Suggested Answer:



```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl create clusterrole deployment-clusterrole --verb=create --resource=Deployment,StatefulSet,DaemonS
et
clusterrole.rbac.authorization.k8s.io/deployment-clusterrole created
student@node-1:~$ kubectl create sa cicd-token --namespace app-team1
serviceaccount/cicd-token created
student@node-1:~$ kubectl create clusterrolebinding deploy-b --clusterrole=deployment-clusterrole --serviceaccount=app-team
1:cicd-token
clusterrolebinding.rbac.authorization.k8s.io/deploy-b created
student@node-1:~$ ▯
```

⊟ **schlagzeuger1** Highly Voted 👍 2 years ago

I would suggest a role binding instead of the clusterrolebinding exposed in the solution as:

$ k create rolebinding deploy-b -n app-team1 --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token

With this, we scope resource creation to the namespace app-team1 as stated in the excercise.

To check, simply issue commands:

$ k auth can-i create deployment -n app-team1 --as system:serviceaccount:app-team1:cicd-token

==> yes

$ k auth can-i create deployment -n default --as system:serviceaccount:app-team1:cicd-token

==> no

upvoted 41 times

⊟ **137eceb** 1 month, 2 weeks ago

clusterRoles are not bound to a namespace. If we wanted to bind the pemissions to a specific namspace, we would create a role and not a clusterrole.

So, clusterrolebinding is correct.

upvoted 1 times

Sukon_Desknot 1 year, 10 months ago

The question specifically asked for clusterRole.

upvoted 3 times

dirkdirkdirk 1 year, 9 months ago

Yes, but not clusterRoleBinding.

upvoted 7 times

sonixrw 1 year, 5 months ago

"limited to the namespace app-team" means roleBinding also fine. Are the question on exam really in this broken english?

upvoted 3 times

spocknimoy 1 year, 5 months ago

make sense

upvoted 1 times

memoor 1 year, 3 months ago

```
root@master-node-1:~# kubectl create clusterrole deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets -o yaml --dry-run=client | kubectl apply -f -
clusterrole.rbac.authorization.k8s.io/deployment-clusterrole configured
root@master-node-1:~# kubectl create serviceaccount cicd-token -n app-team1
serviceaccount/cicd-token created
root@master-node-1:~# kubectl create clusterrolebinding deployment-clusterrolebinding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token --namespace=app-team1 -o yaml --dry-run=client | kubectl apply -f -
clusterrolebinding.rbac.authorization.k8s.io/deployment-clusterrolebinding created
root@master-node-1:~# kubectl auth can-i create deployment -n app-team1 --as system:serviceaccount:app-team1:cicd-token
yes
root@master-node-1:~# kubectl auth can-i create daemonsets --namespace app-team1 --as=system:serviceaccount
no
```

upvoted 2 times

Vihar112 Highly Voted 1 year, 1 month ago

Setting Configuration Context:

```
kubectl config use-context k8s
```

Creating the ClusterRole:

```
kubectl create clusterrole deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets -n app-team1
```

Creating the ServiceAccount:

```
kubectl create serviceaccount cicd-token -n app-team1
```

Binding the ClusterRole to the ServiceAccount:

To bind the ClusterRole to the ServiceAccount in a specific namespace, you'll use a RoleBinding:

```
kubectl create rolebinding deployment-clusterrole-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

upvoted 11 times

nahid0002 Most Recent 3 weeks, 1 day ago

```
kubectl create clusterrole deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets

kubectl create serviceaccount cicd-token -n app-team1

kubectl create rolebinding deployment-rolebinding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token --namespace=app-team1
```

upvoted 1 times

noahsark 1 month ago

killer_sh_lab:

```
k create ns app-team1
k create clusterrole deployment-clusterrole --verb=create --resource=Deployment,StatefulSet,DaemonSet
k create sa cicd-token -n=app-team1
k create rolebinding rb --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n=app-team1
```

Checks:

```
k auth can-i create deployment -n=app-team1 --as system:serviceaccount:app-team1:cicd-token
=> yes

k auth can-i create deployment -n=default --as system:serviceaccount:app-team1:cicd-token
⇒ no
```

upvoted 1 times

⊟ **14b2b2e** 2 months ago

If I used a clusterrolebind here as per the answer will it still be correct?

upvoted 1 times

⊟ **Saransundar** 7 months, 3 weeks ago

First needs to create clusterrole:

```
$ kubectl create clusterrole deployment-clusterrole --verb=create --resource=dployments,statefulsets,daemonsets
```

Step:2 create service account

```
kubectl create sa cicd-token -n app-team1
```

step:3 Create rolebinding to clusterrole for specific namespace

```
$kubectl create rolebinding deployment-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

step:4 test the role and actions

```
$kubectl auth can-i create deployment --as=system:serviceaccount:app-team1:cicd-token -n app-team1
```

upvoted 5 times

⊟ **ProfXsamson** 8 months ago

Alternatively, a RoleBinding can reference a ClusterRole and bind that ClusterRole to the namespace of the RoleBinding. If you want to bind a ClusterRole to all the namespaces in your cluster, you use a ClusterRoleBinding.

upvoted 1 times

⊟ **BABU97** 9 months, 1 week ago

dont fall for this! create clusterrole and follow instructions given! you can specify the namespace when your creating a clusterrolebdinding just as you have been asked in the question 'limited to the namespace app-team, also dont forget to create the serviceaccount on the same namespace app-team1

upvoted 1 times

⊟ **mKrishna** 10 months, 3 weeks ago

```
k create clusterrole deployment-clusterrole -n app-team1 --resource=deployment,statefulset,daemonset --verb=create

k create serviceaccount cicd-token -n app-team1

k create clusterrolebinding rb-deployment-clusterrole --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

upvoted 2 times

⊟ **sandip_k8s** 1 year, 1 month ago

```
k create rolebinding deployments,statefulsets,daemonsets --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

upvoted 1 times

⊟ **Samm1** 1 year, 3 months ago

The question is structured this way:

```
kubectl create ns app-team1. #ns already exist
kubectl create sa cicd-token -n app-team1

kubectl api-resources # to verify the resources names
kubectl create clusterrole deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets
kubectl create rolebinding deployment-role-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token --namespace=app-team1
kubectl auth can-i create deployments --as=system:serviceaccount:app-team1:cicd-token -n app-team1
```

upvoted 2 times

spocknimoy 1 year, 5 months ago

Clusterrolebinding or rolebinding ? Some confusing answers

upvoted 1 times

orangelemons 1 year, 4 months ago

the question clearly states to create a clusterrole but never mentioned using a clusterrolebinding, instead it states to limit the binding to the namespace app-team1. So, it should be rolebinding.

upvoted 1 times

ghsotq 1 year, 5 months ago

```
kubectl create rolebinding deployment-clusterrole-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

upvoted 1 times

Magodi 1 year, 5 months ago

```
controlplane $ k create ns app-team1
namespace/app-team1 created
controlplane $ k create sa -n app-team1 cicd-token
serviceaccount/cicd-token created
controlplane $ k create clusterrole deployment-clusterrole --verb=create --resource=deploy,sts,ds
clusterrole.rbac.authorization.k8s.io/deployment-clusterrole created
controlplane $ k create clusterrolebinding deployment-clusterrole --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-tokenclusterrolebinding.rbac.authorization.k8s.io/deployment-clusterrole created
controlplane $
controlplane $ k auth can-i create sts --as=system:serviceaccount:default:cicd-token
no
controlplane $ k auth can-i create sts --as=system:serviceaccount:app-team1:cicd-token
yes
controlplane $
```

upvoted 3 times

Khaled_Rashwan 1 year, 9 months ago

Create the ClusterRole:

```
kubectl create clusterrole deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets
```

Create a new ServiceAccount:

```
kubectl create serviceaccount -n app-team1 cicd-token
```

Bind the new ClusterRole "deployment-clusterrole" to the new ServiceAccount:

```
kubectl create clusterrolebinding cicd-token-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1
```

upvoted 2 times

RD2022 1 year, 6 months ago

kubectl create clusterrolebinding cicd-token-binding --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token -n app-team1 - will not work as there is no namespace (-n) option for clusterrolebinding - if you do create a CRB it will give CR permissions to the user for the whole cluster

upvoted 2 times

Nirms 1 year, 10 months ago

```
1. k create clusterrole deployment-clusterrole --verb=create --resource=Deployment,StatefulSet,DaemonSet
2. k create sa cicd-token -n app-team1
3. k create rolebinding deploy-b -n app-team1 --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token
4. k auth can-i create deployment -n app-team1 --as system:serviceaccount:app-team1:cicd-token
5. k auth can-i create deployment --as system:serviceaccount:app-team1:cicd-token
```

upvoted 2 times

angdatabase 1 year, 10 months ago

```
k create rolebinding deploy-b -n app-team1 --clusterrole=deployment-clusterrole --serviceaccount=app-team1:cicd-token
```

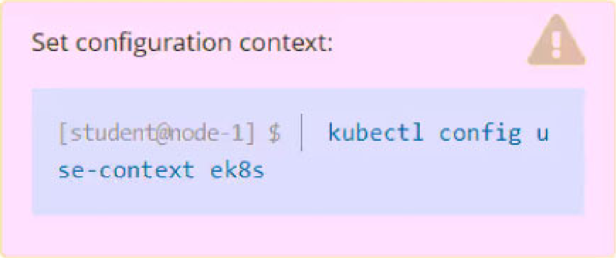
=======

This is Currect

upvoted 1 times

Question #2 *Topic 1*

SIMULATION -



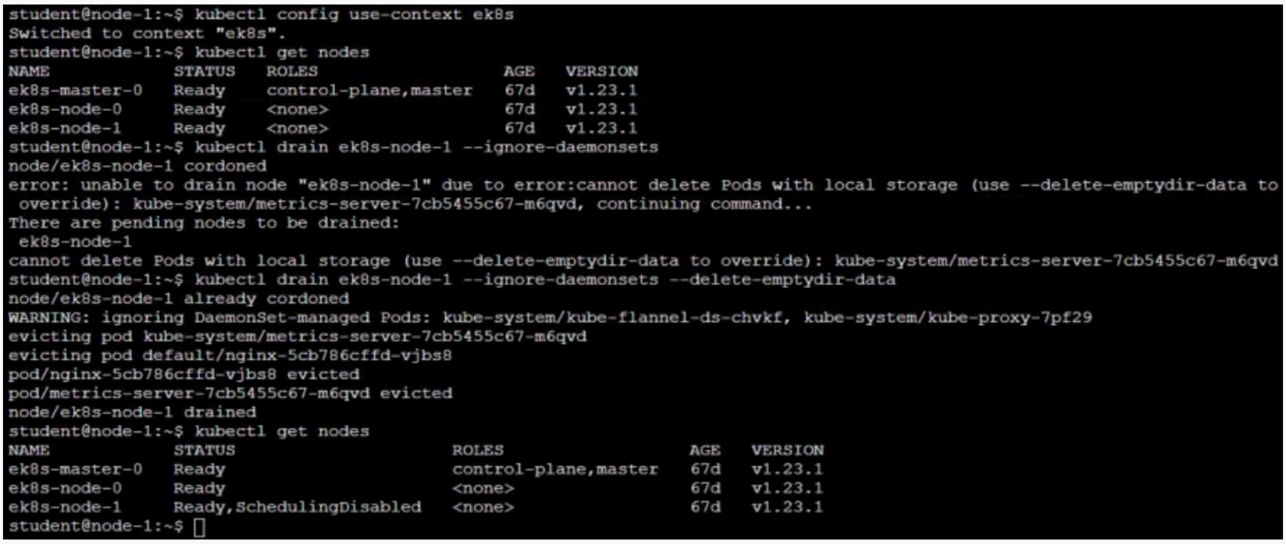
Set configuration context:

```
[student@node-1] $ | kubectl config u
se-context ek8s
```

Task -

Set the node named ek8s-node-0 as unavailable and reschedule all the pods running on it.

Suggested Answer:



```
student@node-1:~$ kubectl config use-context ek8s
Switched to context "ek8s".
student@node-1:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE   VERSION
ek8s-master-0   Ready    control-plane,master   67d   v1.23.1
ek8s-node-0     Ready    <none>                 67d   v1.23.1
ek8s-node-1     Ready    <none>                 67d   v1.23.1
student@node-1:~$ kubectl drain ek8s-node-1 --ignore-daemonsets
node/ek8s-node-1 cordoned
error: unable to drain node "ek8s-node-1" due to error:cannot delete Pods with local storage (use --delete-emptydir-data to
 override): kube-system/metrics-server-7cb5455c67-m6qvd, continuing command...
There are pending nodes to be drained:
 ek8s-node-1
cannot delete Pods with local storage (use --delete-emptydir-data to override): kube-system/metrics-server-7cb5455c67-m6qvd
student@node-1:~$ kubectl drain ek8s-node-1 --ignore-daemonsets --delete-emptydir-data
node/ek8s-node-1 already cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-chvkf, kube-system/kube-proxy-7pf29
evicting pod kube-system/metrics-server-7cb5455c67-m6qvd
evicting pod default/nginx-5cb786cffd-vjbs8
pod/nginx-5cb786cffd-vjbs8 evicted
pod/metrics-server-7cb5455c67-m6qvd evicted
node/ek8s-node-1 drained
student@node-1:~$ kubectl get nodes
NAME            STATUS                     ROLES                  AGE   VERSION
ek8s-master-0   Ready                      control-plane,master   67d   v1.23.1
ek8s-node-0     Ready                      <none>                 67d   v1.23.1
ek8s-node-1     Ready,SchedulingDisabled   <none>                 67d   v1.23.1
student@node-1:~$ 
```

Meliodas12 Highly Voted 1 year, 9 months ago

Do you really need to uncordon it? The task is only to mark the node unschedulable and reschedule all the running pods on it. It should be done by issuing:

# drain node <node_name> --ignore-daemonsets

upvoted 16 times

femijohn123 1 year, 3 months ago

But the command will only mark the node schedulingDisabled how do we now reschedule the pods?

upvoted 1 times

importme 9 months ago

drain will safely evict all of the pods from a node, if there are daemon set-managed pods, drain will not proceed without using --ignore-daemonsets flag

upvoted 1 times

VivekSolutionArchitect 1 year, 1 month ago

The key thing here is to reschedule all POD's, if cluster has only single node and the controlplane then POD's won't get scheduled to controlplane. In this case either taints will need to be removed from controlplane or need to add tolerations to POD so that they can be scheduled on controlplane.

upvoted 2 times

Portman 1 year, 1 month ago

You are absolutely right. In my case, there where 3 nodes (1 master and 2 worker) so just cordon and then drain (i know drain alone can do the job, but cordon first and then drain is more complete) the worker node

upvoted 2 times

gtsvetko Highly Voted 1 year, 8 months ago k cordon ek8s-node-0

k drain ek8s-node-0 --delete-local-data --ignore-daemonsets --force

upvoted 12 times

⊟ **Stunomatic** 1 year, 1 month ago

i think you are right because its asking to rescheduling all pods which are running on ek8s-node-0 so when we cordon it pods will automatically rescheduling on another node.

upvoted 2 times

⊟ **Devch0801** 4 months, 2 weeks ago

When we cordon the node it only marks node as unschedulable. It doesn't automatically reschedule the pods to another node.

upvoted 2 times

⊟ **nahid0002** **Most Recent** 3 weeks, 1 day ago

kubectl cordon ek8s-node-1

kubectl drain ek8s-node-1 --ignore-daemonsets --force

upvoted 1 times

⊟ **HaiNgo** 4 months ago

We have 2 statements.

- drain ek8s-node-0 as unavailable
- kubectl drain ek8s-node-0 —ignore-daemonsets
- reschedule all the pods running on it
- kubectl uncordon ek8s-node-0
- kubectl drain ek8s-master-0 —ignore-daemonsets
- kubectl drain ek8s-node-1 —ignore-daemonsets

upvoted 1 times

⊟ **charlilec** 6 months, 2 weeks ago

should we edit all pods one by one and write tolerations same as taint on ek8s-node-0?? after drain??

upvoted 1 times

⊟ **Kk14** 7 months, 2 weeks ago

Drain marks the node as unschedulable and also evict pods on the node. While, cordon, only marks the node as unschedulable. Hence, kubectl drain <node name> --ignore-daemonsets, would work in this case and also questions refers to only pods and not all objects.

upvoted 2 times

⊟ **anhayg** 8 months ago

This question can be tricky with pods that are NOT created as part of replicaSets. Then the pods will be terminated only and not rescheduled on other nodes. That needs to be checked before draining the node.

upvoted 2 times

⊟ **Natraj007** 8 months, 2 weeks ago

# Mark the node as unschedulable

kubectl cordon ek8s-node-0

# Delete all pods running on the node

kubectl delete pods --all --grace-period=0 --force --field-selector spec.nodeName=ek8s-node-0

upvoted 1 times

⊟ **Stargazer11** 10 months, 3 weeks ago

No need to uncordon it. Draining the node will evict the nodes and mark it unschedulable.

# k drain <node_name> --ignore-daemonsets

# k get nodes

# k get pods -o wide (make sure existing pods are on other nodes)

upvoted 3 times

⊟ **botsokui** 1 year, 4 months ago

don't need to run cordon if you're going to drain it. Drain will do it then evict all the pods in the node.

I'm confused the question didn't say we have to uncordon it eventually, why some comments are saying that we need to uncordon the node?

upvoted 6 times

⊟ **real111** 1 year, 4 months ago

As i read this question i see that there are two actions required - make unavailable and reschedule. To make unavailable (or unschedulable) we need to cordon it and then drain, no?

upvoted 2 times

⊟ **real111** 1 year, 9 months ago

would this be correct:

K get nodes

K drain ek8s-node-0 –ignore-demosets

Mark the node as unschedulable:

kubectl cordon ek8s-node-0

Delete the node:

kubectl delete node ek8s-node-0

kubectl get pods -o wide

upvoted 3 times

⊟ **spocknimoy** 1 year, 5 months ago

I guess You Shouldnt delete node

upvoted 5 times

⊟ **Hamiltonian** 2 years ago

dont forget to uncordon the node so that rescheudling can occur

upvoted 4 times

⊟ **iiiaz** 1 year, 4 months ago

The task does not asks to make it back available! I guess the exam checker will compare the LAST state of the node when ending the exam. If the node is uncordoned, how can the exam checker know that at some moment the node had been unscheduable.

I do not see the point of uncordon it.

upvoted 4 times

⊟ **not4me** 2 years ago

It doesn’t matter, focus on how to solve it instead of node names

upvoted 1 times

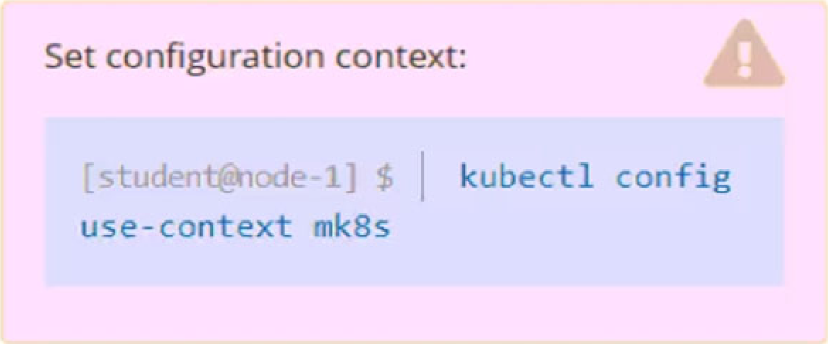
⊟ **gcpengineer** 2 years, 1 month ago

isnt this should be node-0?

upvoted 7 times

Question #3 *Topic 1*

SIMULATION -



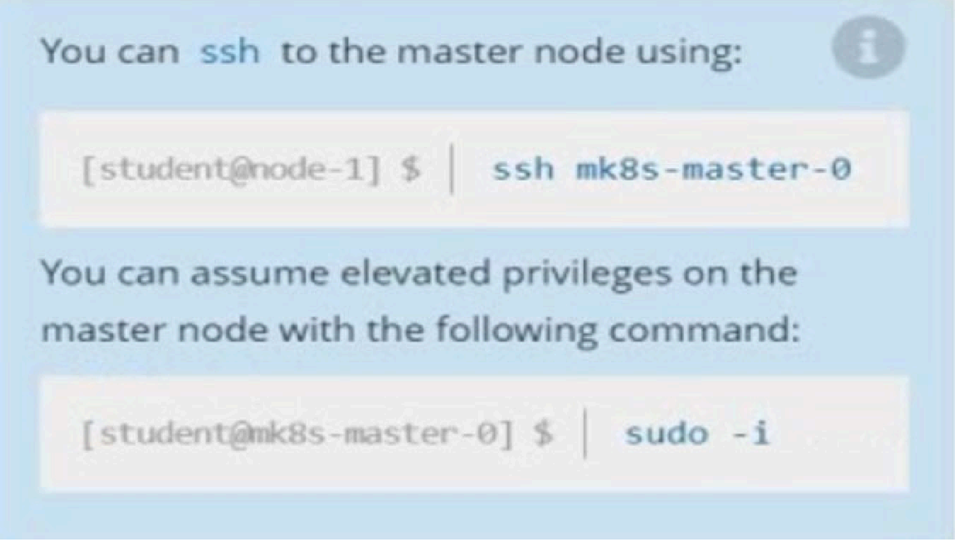
Set configuration context:

```
[student@node-1] $ | kubectl config
use-context mk8s
```

Task -

Given an existing Kubernetes cluster running version 1.22.1, upgrade all of the Kubernetes control plane and node components on the master node only to version 1.22.2.

Be sure to drain the master node before upgrading it and uncordon it after the upgrade.



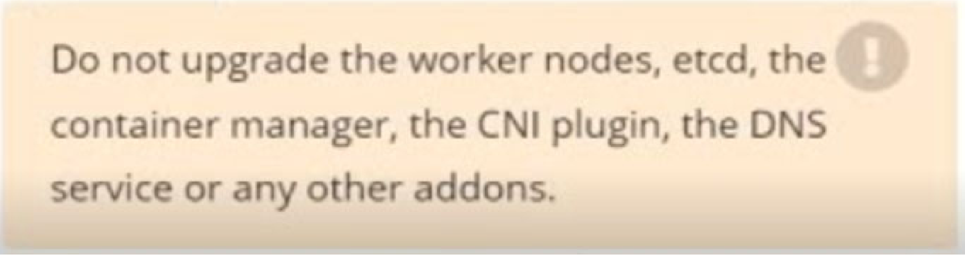
You can ssh to the master node using:

```
[student@node-1] $ | ssh mk8s-master-0
```

You can assume elevated privileges on the master node with the following command:

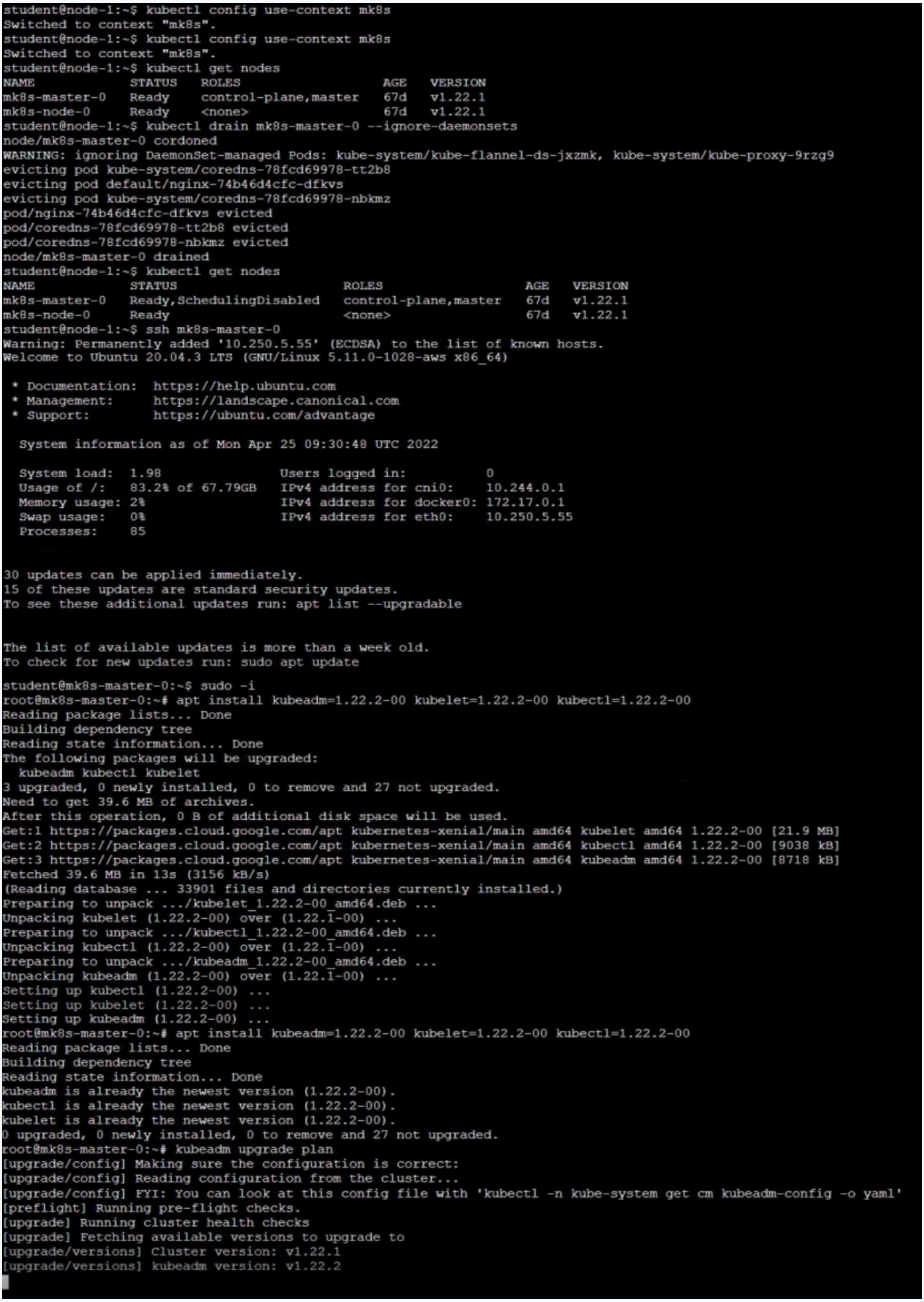
```
[student@mk8s-master-0] $ | sudo -i
```

You are also expected to upgrade kubelet and kubectl on the master node.

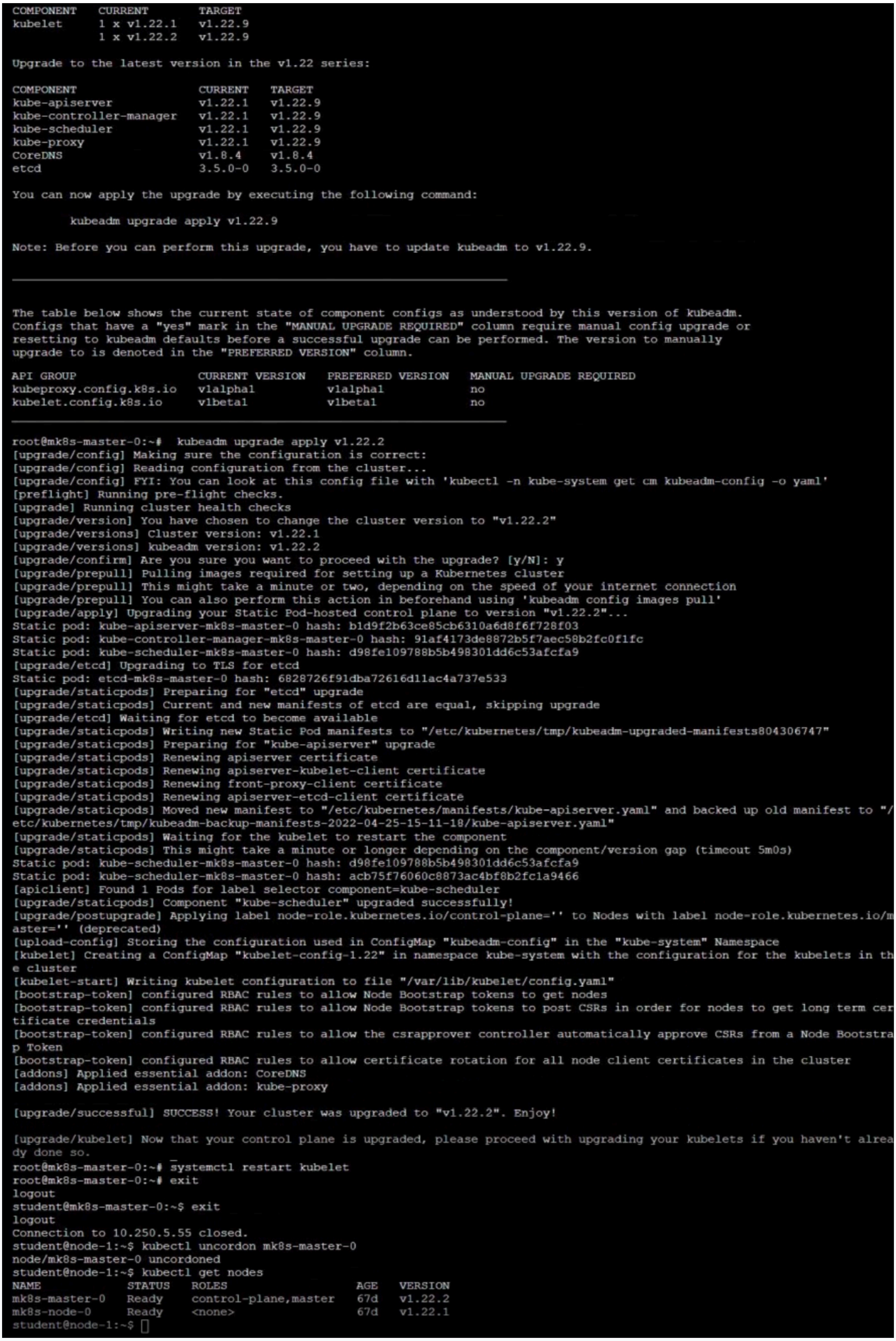


Do not upgrade the worker nodes, etcd, the container manager, the CNI plugin, the DNS service or any other addons.

Suggested Answer:



```
student@node-1:~$ kubectl config use-context mk8s
Switched to context "mk8s".
student@node-1:~$ kubectl config use-context mk8s
Switched to context "mk8s".
student@node-1:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE   VERSION
mk8s-master-0   Ready    control-plane,master   67d   v1.22.1
mk8s-node-0     Ready    <none>                 67d   v1.22.1
student@node-1:~$ kubectl drain mk8s-master-0 --ignore-daemonsets
node/mk8s-master-0 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-jxzmk, kube-system/kube-proxy-9rzg9
evicting pod kube-system/coredns-78fcd69978-tt2b8
evicting pod default/nginx-74b46d4cfc-dfkvs
evicting pod kube-system/coredns-78fcd69978-nbkmz
pod/nginx-74b46d4cfc-dfkvs evicted
pod/coredns-78fcd69978-tt2b8 evicted
pod/coredns-78fcd69978-nbkmz evicted
node/mk8s-master-0 drained
student@node-1:~$ kubectl get nodes
NAME            STATUS                     ROLES                  AGE   VERSION
mk8s-master-0   Ready,SchedulingDisabled   control-plane,master   67d   v1.22.1
mk8s-node-0     Ready                      <none>                 67d   v1.22.1
student@node-1:~$ ssh mk8s-master-0
Warning: Permanently added '10.250.5.55' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Apr 25 09:30:48 UTC 2022

  System load:  1.98               Users logged in:          0
  Usage of /:   83.2% of 67.79GB   IPv4 address for cni0:    10.244.0.1
  Memory usage: 2%                 IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                 IPv4 address for eth0:    10.250.5.55
  Processes:    85


30 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

student@mk8s-master-0:~$ sudo -i
root@mk8s-master-0:~# apt install kubeadm=1.22.2-00 kubelet=1.22.2-00 kubectl=1.22.2-00
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  kubeadm kubectl kubelet
3 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
Need to get 39.6 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.22.2-00 [21.9 MB]
Get:2 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.22.2-00 [9038 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.22.2-00 [8718 kB]
Fetched 39.6 MB in 13s (3156 kB/s)
(Reading database ... 33901 files and directories currently installed.)
Preparing to unpack .../kubelet_1.22.2-00_amd64.deb ...
Unpacking kubelet (1.22.2-00) over (1.22.1-00) ...
Preparing to unpack .../kubectl_1.22.2-00_amd64.deb ...
Unpacking kubectl (1.22.2-00) over (1.22.1-00) ...
Preparing to unpack .../kubeadm_1.22.2-00_amd64.deb ...
Unpacking kubeadm (1.22.2-00) over (1.22.1-00) ...
Setting up kubectl (1.22.2-00) ...
Setting up kubelet (1.22.2-00) ...
Setting up kubeadm (1.22.2-00) ...
root@mk8s-master-0:~# apt install kubeadm=1.22.2-00 kubelet=1.22.2-00 kubectl=1.22.2-00
Reading package lists... Done
Building dependency tree
Reading state information... Done
kubeadm is already the newest version (1.22.2-00).
kubectl is already the newest version (1.22.2-00).
kubelet is already the newest version (1.22.2-00).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@mk8s-master-0:~# kubeadm upgrade plan
[upgrade/config] Making sure the configuration is correct:
[upgrade/config] Reading configuration from the cluster...
[upgrade/config] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[preflight] Running pre-flight checks.
[upgrade] Running cluster health checks
[upgrade] Fetching available versions to upgrade to
[upgrade/versions] Cluster version: v1.22.1
[upgrade/versions] kubeadm version: v1.22.2
```



```
COMPONENT   CURRENT       TARGET
kubelet     1 x v1.22.1   v1.22.9
            1 x v1.22.2   v1.22.9

Upgrade to the latest version in the v1.22 series:

COMPONENT                 CURRENT    TARGET
kube-apiserver            v1.22.1    v1.22.9
kube-controller-manager   v1.22.1    v1.22.9
kube-scheduler            v1.22.1    v1.22.9
kube-proxy                v1.22.1    v1.22.9
CoreDNS                   v1.8.4     v1.8.4
etcd                      3.5.0-0    3.5.0-0

You can now apply the upgrade by executing the following command:

        kubeadm upgrade apply v1.22.9

Note: Before you can perform this upgrade, you have to update kubeadm to v1.22.9.

_____________________________________________________________________


The table below shows the current state of component configs as understood by this version of kubeadm.
Configs that have a "yes" mark in the "MANUAL UPGRADE REQUIRED" column require manual config upgrade or
resetting to kubeadm defaults before a successful upgrade can be performed. The version to manually
upgrade to is denoted in the "PREFERRED VERSION" column.

API GROUP                 CURRENT VERSION   PREFERRED VERSION   MANUAL UPGRADE REQUIRED
kubeproxy.config.k8s.io   v1alpha1          v1alpha1            no
kubelet.config.k8s.io     v1beta1           v1beta1             no
_____________________________________________________________________

root@mk8s-master-0:~#  kubeadm upgrade apply v1.22.2
[upgrade/config] Making sure the configuration is correct:
[upgrade/config] Reading configuration from the cluster...
[upgrade/config] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[preflight] Running pre-flight checks.
[upgrade] Running cluster health checks
[upgrade/version] You have chosen to change the cluster version to "v1.22.2"
[upgrade/versions] Cluster version: v1.22.1
[upgrade/versions] kubeadm version: v1.22.2
[upgrade/confirm] Are you sure you want to proceed with the upgrade? [y/N]: y
[upgrade/prepull] Pulling images required for setting up a Kubernetes cluster
[upgrade/prepull] This might take a minute or two, depending on the speed of your internet connection
[upgrade/prepull] You can also perform this action in beforehand using 'kubeadm config images pull'
[upgrade/apply] Upgrading your Static Pod-hosted control plane to version "v1.22.2"...
Static pod: kube-apiserver-mk8s-master-0 hash: b1d9f2b63ce85cb6310a6d8f6f728f03
Static pod: kube-controller-manager-mk8s-master-0 hash: 91af4173de8872b5f7aec58b2fc0f1fc
Static pod: kube-scheduler-mk8s-master-0 hash: d98fe109788b5b498301dd6c53afcfa9
[upgrade/etcd] Upgrading to TLS for etcd
Static pod: etcd-mk8s-master-0 hash: 6828726f91dba72616d11ac4a737e533
[upgrade/staticpods] Preparing for "etcd" upgrade
[upgrade/staticpods] Current and new manifests of etcd are equal, skipping upgrade
[upgrade/etcd] Waiting for etcd to become available
[upgrade/staticpods] Writing new Static Pod manifests to "/etc/kubernetes/tmp/kubeadm-upgraded-manifests804306747"
[upgrade/staticpods] Preparing for "kube-apiserver" upgrade
[upgrade/staticpods] Renewing apiserver certificate
[upgrade/staticpods] Renewing apiserver-kubelet-client certificate
[upgrade/staticpods] Renewing front-proxy-client certificate
[upgrade/staticpods] Renewing apiserver-etcd-client certificate
[upgrade/staticpods] Moved new manifest to "/etc/kubernetes/manifests/kube-apiserver.yaml" and backed up old manifest to "/
etc/kubernetes/tmp/kubeadm-backup-manifests-2022-04-25-15-11-18/kube-apiserver.yaml"
[upgrade/staticpods] Waiting for the kubelet to restart the component
[upgrade/staticpods] This might take a minute or longer depending on the component/version gap (timeout 5m0s)
Static pod: kube-scheduler-mk8s-master-0 hash: d98fe109788b5b498301dd6c53afcfa9
Static pod: kube-scheduler-mk8s-master-0 hash: acb75f76060c8873ac4bf8b2fc1a9466
[apiclient] Found 1 Pods for label selector component=kube-scheduler
[upgrade/staticpods] Component "kube-scheduler" upgraded successfully!
[upgrade/postupgrade] Applying label node-role.kubernetes.io/control-plane='' to Nodes with label node-role.kubernetes.io/m
aster='' (deprecated)
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config-1.22" in namespace kube-system with the configuration for the kubelets in th
e cluster
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term cer
tificate credentials
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstra
p Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.22.2". Enjoy!

[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't alrea
dy done so.
root@mk8s-master-0:~# systemctl restart kubelet
root@mk8s-master-0:~# exit
logout
student@mk8s-master-0:~$ exit
logout
Connection to 10.250.5.55 closed.
student@node-1:~$ kubectl uncordon mk8s-master-0
node/mk8s-master-0 uncordoned
student@node-1:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE   VERSION
mk8s-master-0   Ready    control-plane,master   67d   v1.22.2
mk8s-node-0     Ready    <none>                 67d   v1.22.1
student@node-1:~$ 
```

⊟ **AWS_cert2023** Highly Voted 11 months, 2 weeks ago

--etcd-upgrade Perform the upgrade of etcd. (default true)

The default etcd upgrade is true, so disable it with

```
sudo kubeadm upgrade apply v1.** --etcd-upgrade=false
```

upvoted 7 times

⊟ **TONNI2001** Highly Voted 👍 1 year, 7 months ago

not exactly like in the exam. but it helps. do not relay on dumps only. i recommend to study with kodekloud and the corresponding udemy course.

upvoted 5 times

⊟ **noahsark** Most Recent ⊙ 1 month ago

# practice lightninglab1 in Mumshad udemy / kodekloud.

upvoted 1 times

⊟ **Ayxanus0610** 9 months, 2 weeks ago

```
$ kubectl config use-config eks8
$ kubectl cordon ek8s-node-1
$ kubectl drain ek8s-node-1 --delete-local-data --ignore-daemonsets --force
```

This is much more understandable solution for me

upvoted 1 times

⊟ **real111** 1 year, 4 months ago

Would those steps be right?

1. k cordon nodename;
2. k drain --ignore-daemonsets nodename;
3. ssh into node;
4. swapoff -a;
5. then find swap line in /etc/fstab and comment it out;
6. run apt-mark unhold kubeadm kubelet kubectl && \
apt-get update && apt-get install -y kubeadm=1.2xx-00 kubelet=1.2x.x-00 kubectl=1.2x.x-00&& \ apt-mark hold kubeadm kubelet kubectl
7. sudo systemctl daemon-reload
8. sudo systemctl restart kubelet
9. exit ssh
10.k uncordon node

upvoted 4 times

⊟ **mrallrounder123453656** 1 year, 6 months ago

we need to set --etcd-upgrade=false, question says so not upgrade the etcd

upvoted 4 times

⊟ **Issaitani** 1 year, 5 months ago

Looking at the upgade logs i can see that etcd is still the same version, but i think you have a point because the --etcd-upgrade flag is set to true by default

upvoted 2 times

⊟ **Alencar_07** 12 months ago

I didn't understand the issue that way.
It says to update all components and etcd is one of them.
"Update all Kubernetes node and control plane components on the master node to version 1.22.2 only."

upvoted 2 times

⊟ **zain1258** 5 months, 1 week ago

I was also confused about it. But if you see the last picture in question, it clearly says not to upgrade etcd and Addons as well. So I think following is the command to upgrade the cluster

```
kubeadm upgrade apply v1.22.2 --etcd-upgrade=false
```

Now the question is how we can skip CoreDNS upgrade as I don't see any option to skip it

upvoted 1 times

⊟ **Frank_sinaatra** 1 month ago

```
kubeadm upgrade apply v1.**** --skip-phases=addon/coredns --etcd-upgrade=false
```

the --skip-phases can be used for any addon, assuming you call it properly.

upvoted 1 times

⊟ **gtsvetko** 1 year, 8 months ago

I has this question on the exam, you can follow the steps below to complete it, just ensure that you are ssh-ing from the right node:

https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/#upgrading-control-plane-nodes

upvoted 5 times

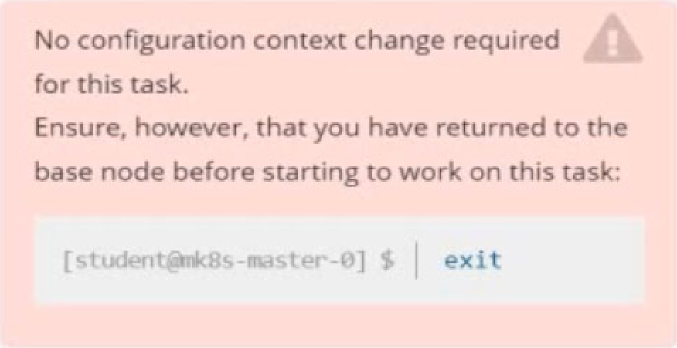
⊟ **phidelics** 1 year, 8 months ago

are this questions exactly thesame in exams?

upvoted 7 times

Question #4 *Topic 1*

SIMULATION -



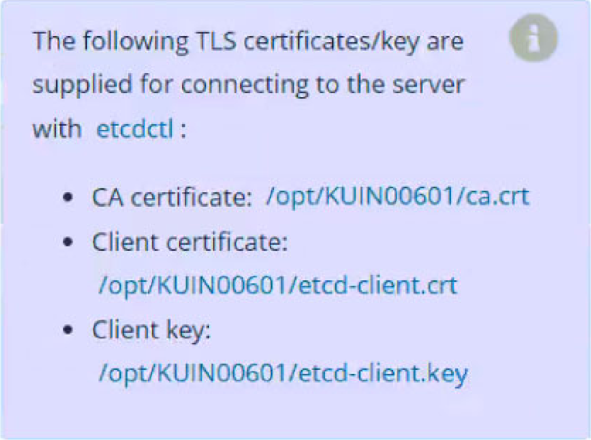
No configuration context change required for this task.

Ensure, however, that you have returned to the base node before starting to work on this task:

```
[student@mk8s-master-0] $ | exit
```

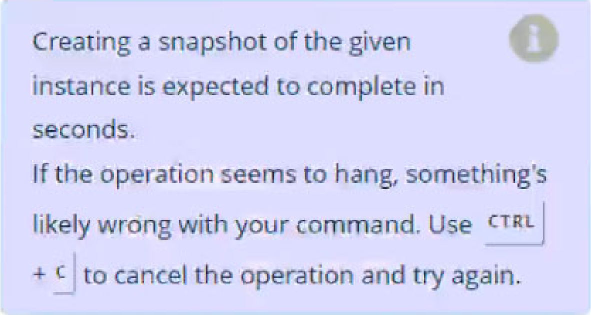
Task -

First, create a snapshot of the existing etcd instance running at https://127.0.0.1:2379, saving the snapshot to /var/lib/backup/etcd-snapshot.db.



The following TLS certificates/key are supplied for connecting to the server with etcdctl :

- CA certificate: /opt/KUIN00601/ca.crt
- Client certificate: /opt/KUIN00601/etcd-client.crt
- Client key: /opt/KUIN00601/etcd-client.key

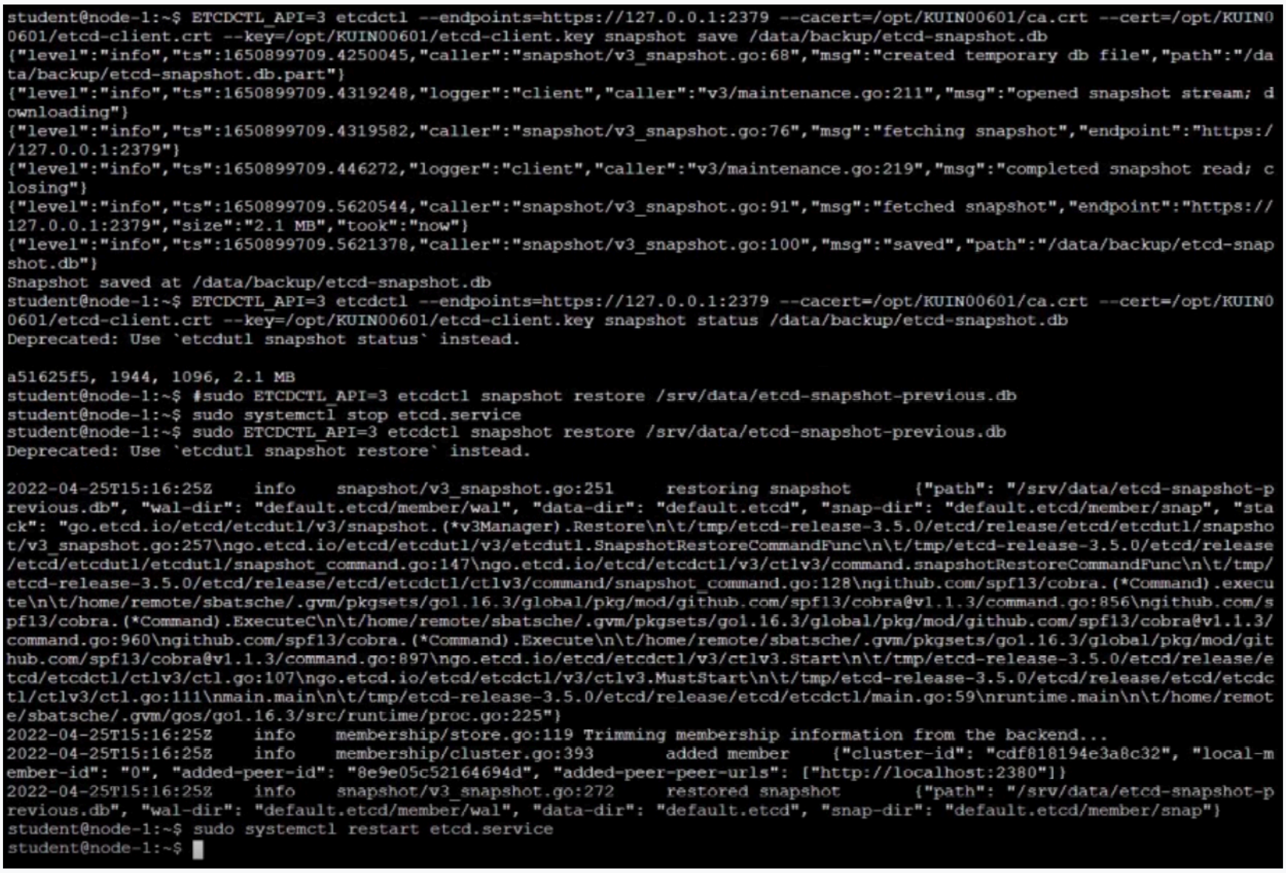


Creating a snapshot of the given instance is expected to complete in seconds.

If the operation seems to hang, something's likely wrong with your command. Use CTRL + C to cancel the operation and try again.

Next, restore an existing, previous snapshot located at /var/lib/backup/etcd-snapshot-previous.db.

Suggested Answer:



```
student@node-1:~$ ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN0
0601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot save /data/backup/etcd-snapshot.db
{"level":"info","ts":1650899709.4250045,"caller":"snapshot/v3_snapshot.go:68","msg":"created temporary db file","path":"/da
ta/backup/etcd-snapshot.db.part"}
{"level":"info","ts":1650899709.4319248,"logger":"client","caller":"v3/maintenance.go:211","msg":"opened snapshot stream; d
ownloading"}
{"level":"info","ts":1650899709.4319582,"caller":"snapshot/v3_snapshot.go:76","msg":"fetching snapshot","endpoint":"https:/
/127.0.0.1:2379"}
{"level":"info","ts":1650899709.446272,"logger":"client","caller":"v3/maintenance.go:219","msg":"completed snapshot read; c
losing"}
{"level":"info","ts":1650899709.5620544,"caller":"snapshot/v3_snapshot.go:91","msg":"fetched snapshot","endpoint":"https://
127.0.0.1:2379","size":"2.1 MB","took":"now"}
{"level":"info","ts":1650899709.5621378,"caller":"snapshot/v3_snapshot.go:100","msg":"saved","path":"/data/backup/etcd-snap
shot.db"}
Snapshot saved at /data/backup/etcd-snapshot.db
student@node-1:~$ ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN0
0601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot status /data/backup/etcd-snapshot.db
Deprecated: Use `etcdutl snapshot status` instead.

a51625f5, 1944, 1096, 2.1 MB
student@node-1:~$ #sudo ETCDCTL_API=3 etcdctl snapshot restore /srv/data/etcd-snapshot-previous.db
student@node-1:~$ sudo systemctl stop etcd.service
student@node-1:~$ sudo ETCDCTL_API=3 etcdctl snapshot restore /srv/data/etcd-snapshot-previous.db
Deprecated: Use `etcdutl snapshot restore` instead.

2022-04-25T15:16:25Z    info    snapshot/v3_snapshot.go:251     restoring snapshot      {"path": "/srv/data/etcd-snapshot-p
revious.db", "wal-dir": "default.etcd/member/wal", "data-dir": "default.etcd", "snap-dir": "default.etcd/member/snap", "sta
ck": "go.etcd.io/etcd/etcdutl/v3/snapshot.(*v3Manager).Restore\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdutl/snapsho
t/v3_snapshot.go:257\ngo.etcd.io/etcd/etcdutl/v3/etcdutl.SnapshotRestoreCommandFunc\n\t/tmp/etcd-release-3.5.0/etcd/release
/etcd/etcdutl/etcdutl/snapshot_command.go:147\ngo.etcd.io/etcd/etcdctl/v3/ctlv3/command.snapshotRestoreCommandFunc\n\t/tmp/
etcd-release-3.5.0/etcd/release/etcd/etcdctl/ctlv3/command/snapshot_command.go:128\ngithub.com/spf13/cobra.(*Command).execu
te\n\t/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobra@v1.1.3/command.go:856\ngithub.com/s
pf13/cobra.(*Command).ExecuteC\n\t/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/github.com/spf13/cobra@v1.1.3/
command.go:960\ngithub.com/spf13/cobra.(*Command).Execute\n\t/home/remote/sbatsche/.gvm/pkgsets/go1.16.3/global/pkg/mod/git
hub.com/spf13/cobra@v1.1.3/command.go:897\ngo.etcd.io/etcd/etcdctl/v3/ctlv3.Start\n\t/tmp/etcd-release-3.5.0/etcd/release/e
tcd/etcdctl/ctlv3/ctl.go:107\ngo.etcd.io/etcd/etcdctl/v3/ctlv3.MustStart\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdc
tl/ctlv3/ctl.go:111\nmain.main\n\t/tmp/etcd-release-3.5.0/etcd/release/etcd/etcdctl/main.go:59\nruntime.main\n\t/home/remot
e/sbatsche/.gvm/gos/go1.16.3/src/runtime/proc.go:225"}
2022-04-25T15:16:25Z    info    membership/store.go:119 Trimming membership information from the backend...
2022-04-25T15:16:25Z    info    membership/cluster.go:393       added member    {"cluster-id": "cdf818194e3a8c32", "local-m
ember-id": "0", "added-peer-id": "8e9e05c52164694d", "added-peer-peer-urls": ["http://localhost:2380"]}
2022-04-25T15:16:25Z    info    snapshot/v3_snapshot.go:272     restored snapshot       {"path": "/srv/data/etcd-snapshot-p
revious.db", "wal-dir": "default.etcd/member/wal", "data-dir": "default.etcd", "snap-dir": "default.etcd/member/snap"}
student@node-1:~$ sudo systemctl restart etcd.service
student@node-1:~$ █
```

⊟ **rajusai** Highly Voted 👍 1 year, 10 months ago

In the real exam you need restore from /var/lib/backup/etcd-snapshot-previous.db and there will be a permission issue, to fix this you need to be a root user and change owner permission then you need to restore db backup

upvoted 16 times

⊟ **MYOM** 1 year, 1 month ago

take note ppl!

upvoted 2 times

⊟ **rakeshsjadhav** 1 year, 4 months ago

Cd /etc/Kubernetes/manifest---- to check etcd yaml . Any scenario where this manifest file was not located ?

upvoted 1 times

⊟ **BOD007** 1 year, 2 months ago

If its an external etcd-server you will not see the manifest.

upvoted 1 times

⊟ **137eceb** 1 month, 2 weeks ago

it is not external

upvoted 1 times

⊟ **BOD007** 1 year, 2 months ago

If etcd is stacked, meaning it is running as a pod on the master node... the manifest file should be there. I strongly doubt the exam will setup an external etcd server as it will take a lot more effort to complete that task.

Personally, i will start by checking ;

- If there is an etcd pod on the master-node
- Describe the api-server pod to see the etcd-server address(localhost or remote)
- If local, then business as usual
- If remote you will need some additional steps to fully restore, including ssh into the etcd server and modifying the --data-dir at systemd/system/etcd.service file (A lot more headache)

upvoted 3 times

⊟ **schlagzeuger1** Highly Voted 👍 2 years ago

The solution does not fully solve the excercise, because the restore operation creates a default.etcd directory in the current directory where the utility ectdctl is called. Inside the newly created defaultl.etcd directory, it's a subdirectory called "member" containing the actual backup. That location is for sure not the one that is configured at etcd start. First is advisable to find out where the ETCD data is (--data-dir flag at service start). To effectively apply the backup, for instance, after stopping the service, we should move the targeted "member" folder in --data-dir location, let's name it $DATA_DIR_PATH, and decorate the restore operation with the flag data-dir set to $DATA_DIR_PATH:

```
# #-- SERVICE ETCD STOPPED --
...
# mv $DATA_DIR_PATH/member $SOME_OTHER_LOCATION
# ETCDCTL_API=3 etcdctl --data-dir $DATA_DIR_PATH snapshot restore /var/lib/backup/etcd-snapshot-previous.db
# #-- SERVICE ETCD STARTED --
```

upvoted 12 times

flapa83 1 year, 10 months ago

This is only possible if ETCD is created as a systemd service and not as a pod. in most cases etcd will be created as a static pod, you cant utilize systemd service for that and you obviously cant have DATA_DIR_PATH as an environmental variable

upvoted 2 times

flapa83 1 year, 10 months ago

setting DATA_DIR_PATH is ok, not an issue, ignore initial comment on env

upvoted 1 times

asorin 2 years ago

indeed, or you can repoint in the manifest file of etcd the data-dir to the restored directory

upvoted 8 times

noahsark Most Recent 1 month ago

# practice backup and restore 2 lab in Mumshad udemy / kodekloud.

upvoted 1 times

Aishu610 2 months, 4 weeks ago

Hello,
Can someone please explain the steps with commands to change to be a root user?
Thank you

upvoted 1 times

[Removed] 3 months, 3 weeks ago

tip for this question ,change the user to root after you will be able to do it

upvoted 1 times

cipheronix 5 months, 2 weeks ago

Backup:

```
etcdctl --endpoints 127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt
--key=/opt/KUIN00601/etcd-client.key snapshot save /var/lib/backup/etcd-snapshot.db
```

Restore:

```
mkdir -p /var/lib/new-etcd/
etcdctl snapshot restore --data-dir=/var/lib/new-etcd/ /var/lib/backup/etcd-snapshot-previous.db
```

Edit manifest:

```
vi /etc/kuberenetes/manifest/etcd.yaml

#change hostpath
- hostPath:
path: /var/lib/new-etcd/
type: DirectoryOrCreate
name: etcd-data
```

upvoted 4 times

Vish0211 7 months, 3 weeks ago

Do we need to update the Volume.hostPath in file - /etc/kubernetes/manifests/etcd.yaml post restore ?

upvoted 2 times

abilalzengin 9 months, 2 weeks ago

Solution is here > https://www.youtube.com/watch?v=Onb85cQl1jc

upvoted 6 times

Ayxanus0610 9 months, 2 weeks ago

```
$ ETCDCTL_API=3 etcdctl --endpoints 127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot save /var/lib/backup/etcd-snapshot.db


$ ETCDCTL_API=3 etcdctl --endpoints 127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot-previous.db
```

This is more effective one you can use

upvoted 2 times

LavaPup 1 year, 1 month ago

Wondering if no one has noticed the path in question? It clearly says to take the snapshot under /var/lib/backup/ but it seems everyone is okay with /etc/data/

Any hints/help?

upvoted 5 times

DSK 1 year, 1 month ago

Why do we need to stop etcd service?

upvoted 1 times

Portman 1 year, 1 month ago

I guess that it is always a good practice, so that you make sure nothing is writing on ETCD while performing the restore.

upvoted 1 times

Anky1090 1 year, 2 months ago

If you see that the question simply says to restore a backup. Doesn't mention any data directory.
There's a default.etcd directory that gets created if you restore this in the current working directory in the exam.

Remember, if the etcd doesn't come back up the way it's expected, you may loose onto resources in the K8s cluster. There are around 9-10 questions to be performed in that context. You need to treat this question very carefully and not mess with the database by changing the manifest file.

upvoted 6 times

mellohello 1 year, 5 months ago

```
#backup
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot save /etc/data/etcd-snapshot.db

#restore
ETCDCTL_API=3 etcdctl --endpoints="https://127.0.0.1:2379" --cacert=/opt/KUIN000601/ca.crt --cert=/opt/KUIN000601/etcd-client.crt --key=/opt/KUIN000601/etcd-client.key snapshot restore /var/lib/backup/etcd-snapshot-previoys.db
```

upvoted 8 times

mrallrounder123453656 1 year, 6 months ago

do we need to run these backup/restored commands from the master node?

upvoted 1 times

sTeVe86 1 year, 7 months ago

https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/
to cover the etcd backup and restore

upvoted 1 times

Nicky88 1 year, 8 months ago

You may have 2 etcd instances. One running in the cluster itself and the second one running outside the kubernetes cluster. They are not asking you to change context so DO NOT restore in the kubernets cluster.
Follow these steps outside the cluster:
1.Execute “member list” and “snapshot status” to check hash
2.systemctl stop etcd
3.restore another backup using same certs, endpoint, and different dir
4.chown -R etcd:etcd /DIR_YOU_RESTORE
5.change dir in the service file

6.system daemon-reload

7.systermctl start etcd

8.systemctl status etcd

9.member list - to check you have different hash

upvoted 7 times

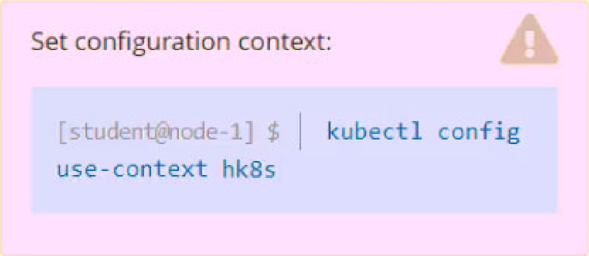
⊟ **Nicky88** 1 year, 8 months ago

Any update about this task? Is there any step-by-step guide?

upvoted 2 times

Question #5 *Topic 1*

SIMULATION -



Set configuration context:

```
[student@node-1] $ | kubectl config
use-context hk8s
```

Task -

Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace fubar.

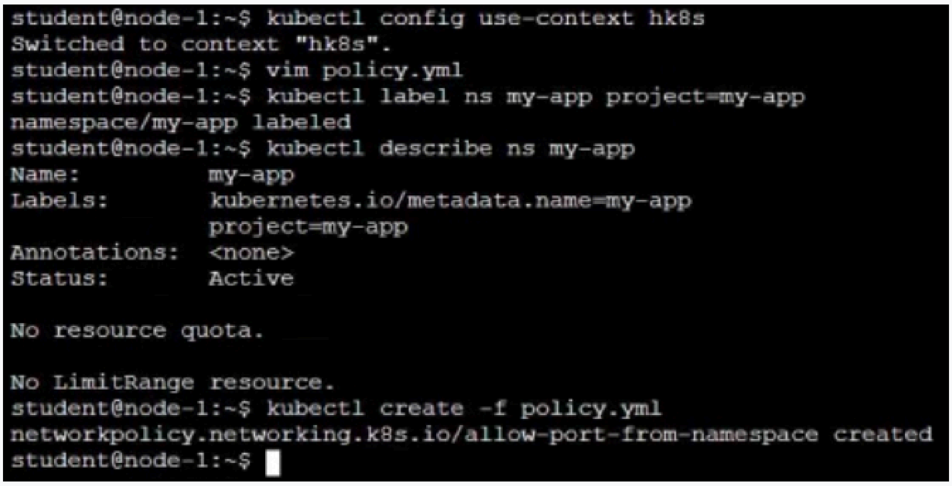
Ensure that the new NetworkPolicy allows Pods in namespace internal to connect to port 9000 of Pods in namespace fubar.

Further ensure that the new NetworkPolicy:

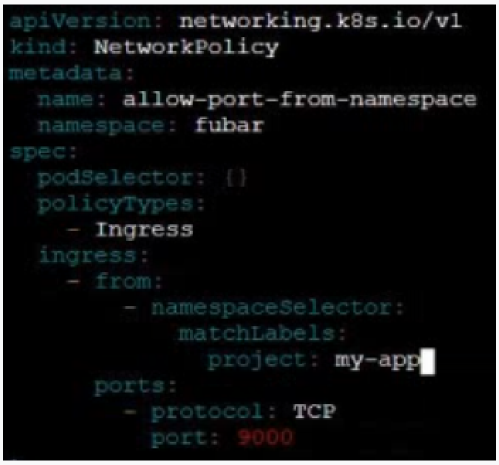
✑ does not allow access to Pods, which don't listen on port 9000

✑ does not allow access from Pods, which are not in namespace internal

Suggested Answer:



```
student@node-1:~$ kubectl config use-context hk8s
Switched to context "hk8s".
student@node-1:~$ vim policy.yml
student@node-1:~$ kubectl label ns my-app project=my-app
namespace/my-app labeled
student@node-1:~$ kubectl describe ns my-app
Name:         my-app
Labels:       kubernetes.io/metadata.name=my-app
              project=my-app
Annotations:  <none>
Status:       Active

No resource quota.

No LimitRange resource.
student@node-1:~$ kubectl create -f policy.yml
networkpolicy.networking.k8s.io/allow-port-from-namespace created
student@node-1:~$ 
```



```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-port-from-namespace
  namespace: fubar
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              project: my-app
      ports:
        - protocol: TCP
          port: 9000
```

TemitopeWalker Highly Voted 2 years ago

I think this asnwer is wrong the solution should be

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: allow-port-from-namespace

namespace: fubar

spec:

```
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
- namespaceSelector:
matchLabels:
kubernetes.io/metadata.name: internal
ports:
- protocol: TCP
port: 9000
```

upvoted 54 times

pentium2000 Highly Voted 1 year, 11 months ago

For this question, we should create a label for "internal" namespace in further YAML.

```
# k label ns internal tier=internal

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
- namespaceSelector:
matchLabels:
tier: internal
ports:
- protocol: TCP
port: 9000
```

upvoted 12 times

Jibbajabba 1 year ago

Don't think you need to create a label specifically unless you need to work with multiple namespaces

"The Kubernetes control plane sets an immutable label kubernetes.io/metadata.name on all namespaces, the value of the label is the namespace name.

While NetworkPolicy cannot target a namespace by its name with some object field, you can use the standardized label to target a specific namespace."

I suppose that implies you CAN but you don't HAVE TO.

upvoted 6 times

noahsark Most Recent 1 month ago

killer_sh_lab:

part1

```
# Changing to port 80 for test purposes
# From internal to fubar
# netpol is in fubar
# ingress is from internal

k create ns fubar --labels=’name=fubar’
k run nginx -n=fubar --image nginx --port 80
k create ns internal --labels=’name=internal’
k run nginx2 -n=internal --image nginx --port 80 --labels=’name=internal’
```

upvoted 1 times

Pi_otR 9 months, 4 weeks ago

Due to this part: "- does not allow access from Pods, which are not in namespace internal" -means that even pods in namespace fubar should not be able to reach other pods in same namespace.

I would suggest to do following :

```
----
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: {} # Selects all Pods in the `fubar` namespace
policyTypes:
- Ingress
- Egress
ingress:
- from:
- namespaceSelector:
matchLabels:
name: internal
ports:
- protocol: TCP
port: 9000
```

this way Egress is specified but due to fact nothing is defined pod in same NSs are not able to communicate.

upvoted 2 times

fonte 4 months, 1 week ago

No need for that... the policy already restricts the traffic to the internal ns.

Tested it and even another pod in the fubar ns cannot reach the other pods listening port 9000.

upvoted 2 times

Alencar_07 12 months ago

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: # Selects Pods in the namespace where the NetworkPolicy is applied
matchLabels: {}

policyTypes:
- Ingress

ingress:
- from:
- namespaceSelector: # Allow traffic only from Pods in the 'internal' namespace
matchLabels:
name: internal
ports:
- protocol: TCP
port: 9000 # Allow connections to port 9000

egress:
- to:
- namespaceSelector: # Allow traffic only to Pods in the 'fubar' namespace
matchLabels:
```

```
name: fubar
ports:
- protocol: TCP
port: 9000 # Allow connections to port 9000
```

upvoted 1 times

Stunomatic 1 year, 1 month ago

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
- namespaceSelector:
matchLabels:
name: internal
ports:
- protocol: TCP
port: 9000
```

upvoted 1 times

aloshari 1 year, 1 month ago

I think we need to check my-app labels first to match it,

upvoted 1 times

Shenannigan 1 year, 3 months ago

Tested locally and this worked for me
Used Nginx Pod with port set to 9000 in the fubar namespace
Used Alpine Pod image alpine/curl in the internal namespace for testing
exec into the Alpine Pod and run the command:
curl (your nginx pod IP seperated by dashes).fubar.pod.cluster.local:9000

Policy:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
- namespaceSelector:
matchLabels:
kubernetes.io/metadata.name: "internal"
ports:
- protocol: TCP
port: 9000
```

upvoted 4 times

VivekSolutionArchitect 1 year, 1 month ago

It doesn't work for me when I use port 9000 for nginx, however port 80 works fine. Not sure if I am doing something incorrectly.

upvoted 2 times

didorins 1 year, 5 months ago

I still fail to understand this question. Do they want me to create a policy that allows only traffic on port 9000 from namespace internal (x2 ingress) or do they want to create a network policy to restrict incoming traffic, so that only pods FROM (ingress) internal namespace are allowed and pods TO (egress) port 9000 ?

upvoted 1 times

Nurbol 1 year, 5 months ago

To one who wonder where this from: kubernetes.io/metadata.name: internal, run: k get ns internal --show-labels

upvoted 3 times

sonixrw 1 year, 5 months ago

Should we also add deny any any and add NP to access port 9000 in ns foobar, from internal?

upvoted 1 times

ahmedovelshan 1 year, 6 months ago

Maybe this?

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector:
matchLabels:
- namespaceSelector:
matchExpressions:
- key: namespace
operator: In
Values: ["fubar"]
policyTypes:
- Ingress
- Egress
ingress:
- from:
- namespaceSelector:
matchExpressions:
- key: namespace
operator: In
Values: ["internal"]
ports:
- protocol: TCP
port: 9000
```

upvoted 3 times

kopper2019 1 year, 7 months ago

using this I get an error so I had to use label, at least practicing not in exam yet

kubernetes.io/metadata.name: internal

upvoted 2 times

dayody 1 year, 3 months ago

me too I got an error using it

upvoted 1 times

kopper2019 1 year, 7 months ago

I was using

kubernetes.io/metadata.name=echo instead of kubernetes.io/metadata.name: echo

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: test-network-policy
```

```
namespace: my-app
spec:
podSelector: {}
policyTypes:
- Ingress
ingress:
- from:
- namespaceSelector:
matchLabels:
kubernetes.io/metadata.name: echo
ports:
- protocol: TCP
port: 9000
```

upvoted 2 times

Sylzys 1 year, 7 months ago

Is there a template during the exam or do we have to write it all from scratch?

upvoted 2 times

ramon712 1 year, 8 months ago

Sorry, I disagree with :

```
matchLabels:
kubernetes.io/metadata.name: internal
```

I suggest :

```
ingress:
- from:
- namespaceSelector:
matchLabels:
items[0].metadata.namespace: internal # from query kubectl get po with jsonpath
```

What do you think ?

upvoted 2 times

ramon712 1 year, 8 months ago

I made an error. So, the answer from Kubernetes's document :
kubernetes.io/metadata.name
Example: kubernetes.io/metadata.name: "mynamespace"

Used on: Namespaces
The Kubernetes API server (part of the control plane) sets this label on all namespaces. The label value is set to the name of the namespace.
You can't change this label's value.
This is useful if you want to target a specific namespace with a label selector.

upvoted 1 times

rajusai 1 year, 10 months ago

They have asked us for namespace internal, hence following is the correct under matchlabels kubernetes.io/metadata.name: internal

upvoted 2 times

Steve122 1 year, 10 months ago

no magic: (this policy is ns scoped so no need any labelling on ns)
tested, works

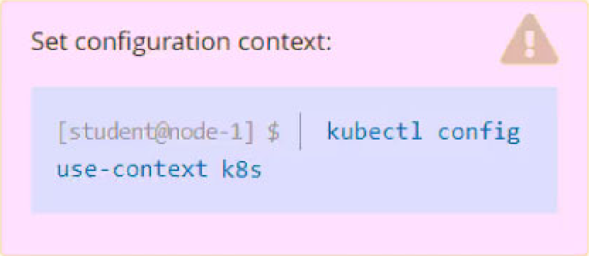
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: allow-port-from-namespace
namespace: fubar
spec:
podSelector: {}
policyTypes:
```

```
- Ingress
ingress:
- from:
ports:
- protocol: TCP
port: 9000
```

upvoted 1 times

Question #6 *Topic 1*

SIMULATION -



Set configuration context:

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

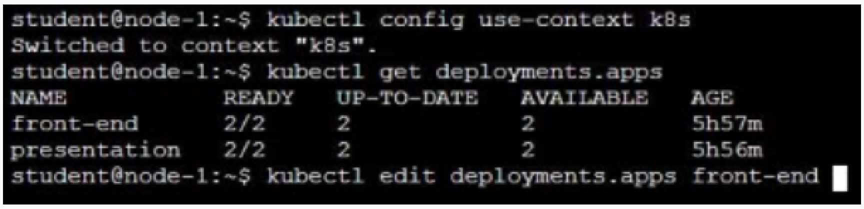
Task -

Reconfigure the existing deployment front-end and add a port specification named http exposing port 80/tcp of the existing container nginx.

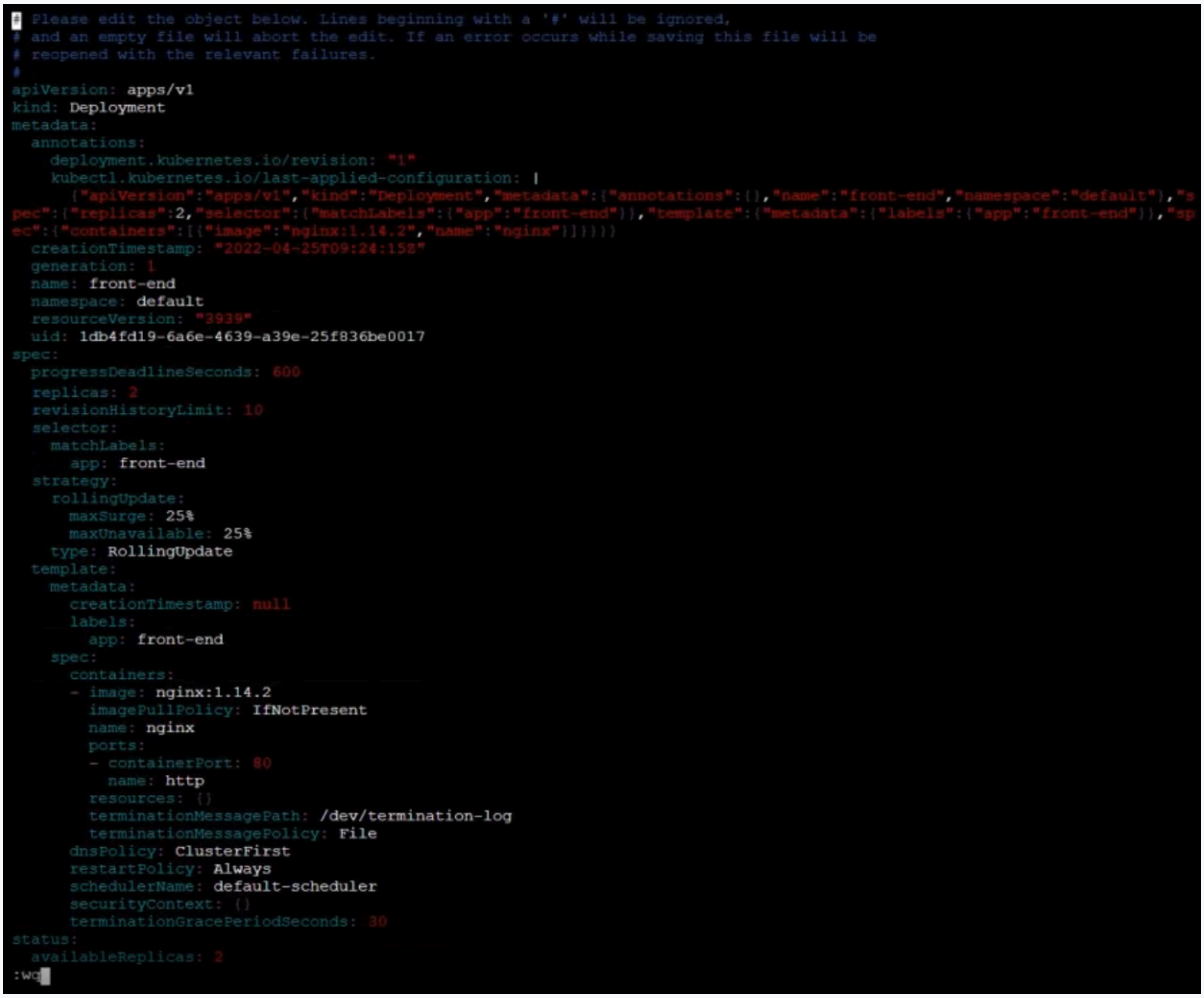
Create a new service named front-end-svc exposing the container port http.

Configure the new service to also expose the individual Pods via a NodePort on the nodes on which they are scheduled.

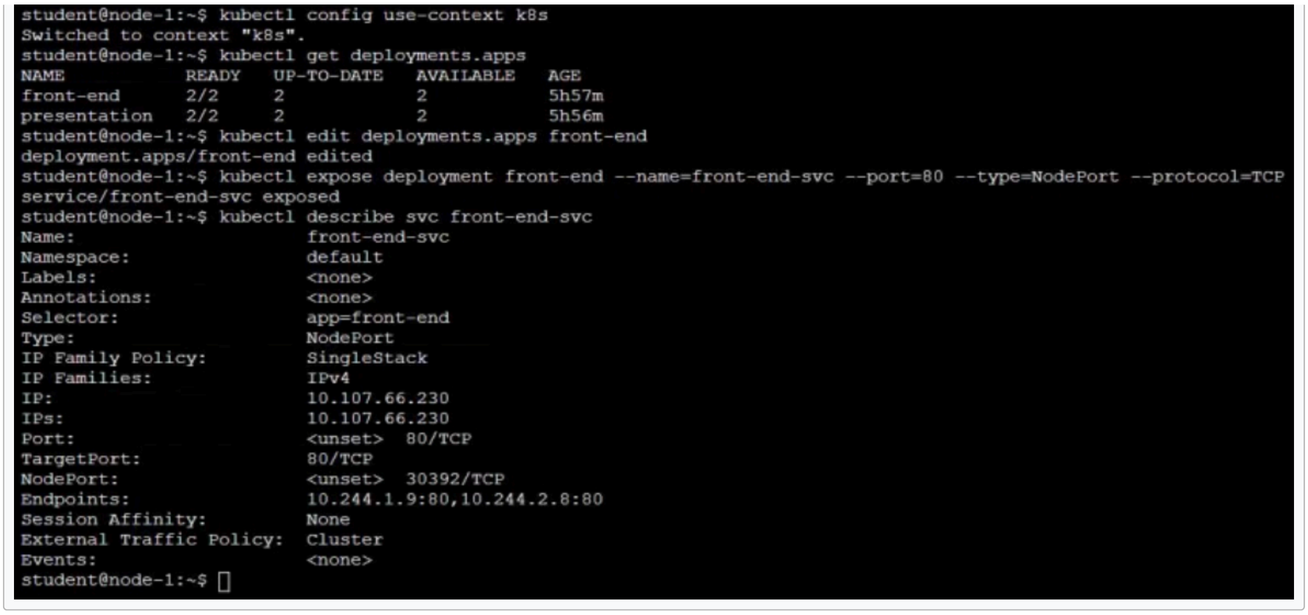
Suggested Answer:



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
front-end      2/2     2            2           5h57m
presentation   2/2     2            2           5h56m
student@node-1:~$ kubectl edit deployments.apps front-end
```



```
# 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"
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"name":"front-end","namespace":"default"},"s
pec":{"replicas":2,"selector":{"matchLabels":{"app":"front-end"}},"template":{"metadata":{"labels":{"app":"front-end"}},"sp
ec":{"containers":[{"image":"nginx:1.14.2","name":"nginx"}]}}}}
  creationTimestamp: "2022-04-25T09:24:15Z"
  generation: 1
  name: front-end
  namespace: default
  resourceVersion: "3939"
  uid: 1db4fd19-6a6e-4639-a39e-25f836be0017
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: front-end
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: front-end
    spec:
      containers:
      - image: nginx:1.14.2
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
        - containerPort: 80
          name: http
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
status:
  availableReplicas: 2
:wq
```



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
front-end      2/2     2            2           5h57m
presentation   2/2     2            2           5h56m
student@node-1:~$ kubectl edit deployments.apps front-end
deployment.apps/front-end edited
student@node-1:~$ kubectl expose deployment front-end --name=front-end-svc --port=80 --type=NodePort --protocol=TCP
service/front-end-svc exposed
student@node-1:~$ kubectl describe svc front-end-svc
Name:                     front-end-svc
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=front-end
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.107.66.230
IPs:                      10.107.66.230
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30392/TCP
Endpoints:                10.244.1.9:80,10.244.2.8:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
student@node-1:~$ 
```

Hamiltonian Highly Voted 2 years ago

if you add port, name, and protocol in the deployment spec, then you only need to run:

kubectl expose deployment front-ed--type=NodePort --name=front-end-svc

upvoted 16 times

yorkicurke Highly Voted 2 months ago

for reconfiguring part ;

#kubectl edit deployment front-end

then add the following in the container part;

containers:

- name: nginx-dep-cont

image: nginx:1.14.2

ports:

- name: http

containerPort: 80

protocol: TCP

and to Finally expose it;

## kubectl expose deployment front-end --name front-end-svc --target-port http --type=NodePort

extra steps;

# kubectl describe svc front-end-svc

## curl PICK-ANY-IP-FROM-ENDPOINTS:80

in my case

# curl 192.168.133.201:80

Tested!

upvoted 7 times

137eceb Most Recent 1 month, 2 weeks ago

tcp is default. It will add it automatically.

upvoted 1 times

Elvi13 6 months ago

after editing the existing ,kubectl expose deployment front-end --name=front-end-svc --port=80 --target-port=80 --type=NodePort

upvoted 1 times

Ayxanus0610 9 months, 2 weeks ago

kubectl config use-context k8s

kubectl expose deployment front-end --port=80 --target-port=80 --protocol=TCP --type=NodePort --name=front-end-svc

with this solution it is much more easier

upvoted 3 times

AWS_cert2023 11 months, 2 weeks ago

1 kubectl edit deploy front-end

```
ports:
- containerPort: 80
name: http
```

2

```
apiVersion: v1
kind: Service
metadata:
name: svc1
spec:
selector:
app: nginx
ports:
- name: name-of-service-port
protocol: TCP
port: 80
targetPort: http
```

3

```
apiVersion: v1
kind: Service
metadata:
name: svc2
type: NodePort
selector:
app: nginx
ports:
- port: 80
targetPort: http
nodePort: 30007
```

upvoted 1 times

Alencar_07 12 months ago

which this one makes you think more than you need to answer.

When asked to export (http) you can put a flag - name: http : example:

```
ports:
- container port: 80
name: http
```

However, when you create the service and export port 80 from it.

upvoted 1 times

Samm1 1 year, 3 months ago

The question as I understood:

```
kubectl create deployment front-end --image=nginx --replicas=1 --dry-run=client -oyaml > dep1.yaml
kubectl apply -f dep1.yaml
vi dep1.yaml # manually add the name: http and containerPort: 80
kubectl expose deployment front-end --target-port=http --name=front-end-svc --type=ClusterIP/NodePort --dry-run=client -oyaml>svc1.yaml
kubectl apply -f svc1.yaml
kubectl describe svc front-end-svc
```

upvoted 2 times

real111 1 year, 4 months ago

I dont realy understand if we need to do two or three actions- 1.reconfigure the yaml file - add ports. 2 create sa and 3. edit service adding type Nodeport or is it ok just reconfigure yaml and run k expose deployment front-end --name front-end-svc --type NodePort --port 80 --target-port HTTP ?

upvoted 2 times

iiiaz 1 year, 4 months ago

This is what I asked too. Search here for my user "iiiaz". But I think for your comment at action 2 is not sa (service account) but svc (service). I guess there are 3 actions but can LinuxFoundation tell if you first done action 2 (create service) and then action 3 (change this service type to NodePort)? As you can directly create a service type NodePort.

upvoted 1 times

iiiaz 1 year, 4 months ago

Who can explain the difference between these tasks:

"T1. Create a new service named front-end-svc exposing the container port http.

T2. Configure the new service to also expose the individual Pods via a NodePort on the nodes on which they are scheduled."

Is it T1: first expose the deployment as default (service type ClusterIP is created) on port http?

Then, T2, edit the service from type ClusterIP to type NodePort?

I see the solutions mentioned here go directly to task 2. Quite confusing question.

upvoted 1 times

mrallrounder123453656 1 year, 6 months ago

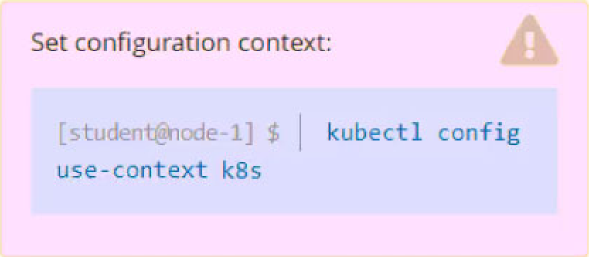
This. is more align with the question

```
k expose deployment front-end --name front-end-svc --type NodePort --port 80 --target-port HTTP
```

upvoted 7 times

Question #7 *Topic 1*

SIMULATION -



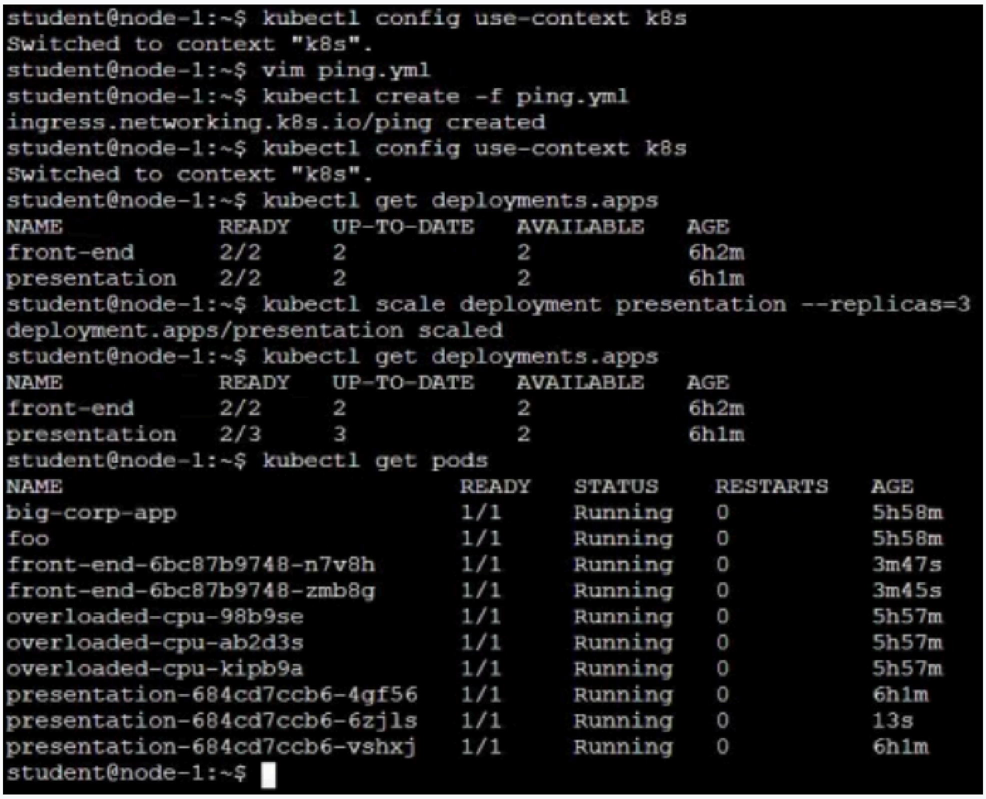
Set configuration context:

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

Task -

Scale the deployment presentation to 3 pods.

Suggested Answer:



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ vim ping.yml
student@node-1:~$ kubectl create -f ping.yml
ingress.networking.k8s.io/ping created
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
front-end      2/2     2            2           6h2m
presentation   2/2     2            2           6h1m
student@node-1:~$ kubectl scale deployment presentation --replicas=3
deployment.apps/presentation scaled
student@node-1:~$ kubectl get deployments.apps
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
front-end      2/2     2            2           6h2m
presentation   2/3     3            2           6h1m
student@node-1:~$ kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
big-corp-app                     1/1     Running   0          5h58m
foo                              1/1     Running   0          5h58m
front-end-6bc87b9748-n7v8h       1/1     Running   0          3m47s
front-end-6bc87b9748-zmb8g       1/1     Running   0          3m45s
overloaded-cpu-98b9se            1/1     Running   0          5h57m
overloaded-cpu-ab2d3s            1/1     Running   0          5h57m
overloaded-cpu-kipb9a            1/1     Running   0          5h57m
presentation-684cd7ccb6-4gf56    1/1     Running   0          6h1m
presentation-684cd7ccb6-6zjls    1/1     Running   0          13s
presentation-684cd7ccb6-vshxj    1/1     Running   0          6h1m
student@node-1:~$ █
```

noahsark 1 month ago

killer_sh_lab:

k create deploy presentation --image=nginx --dry-run=client -o=yaml > presentation.yml

k apply -f presentation.yml

k scale deploy presentation --replicas=3

upvoted 1 times

sreehp00 6 months, 3 weeks ago

dont forget to mention out the concerned namespace:

controlplane ~ ✖ k scale --replicas=3 deployment/webapp-video -n app-space

deployment.apps/webapp-video scaled

upvoted 2 times

cp2323 1 year ago

kubectl scale --replicas=3 deployment/presentation

upvoted 3 times

junaid84u 1 year, 10 months ago

use below command to scale the deployment

kubvectl scale deployment_name --replicas=3

upvoted 2 times

sTeVe86 1 year, 7 months ago

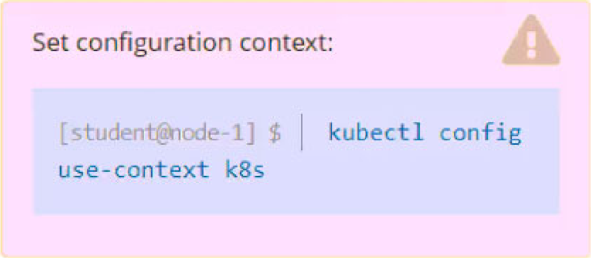
You must specify the deployment to scale up or down.

```
kubectl scale deployment <deployment_name> --replicas=3
kubectl scale deployment/<deployment_name> --replicas=3
```

upvoted 14 times

Question #8 *Topic 1*

SIMULATION -



Set configuration context:

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

Task -

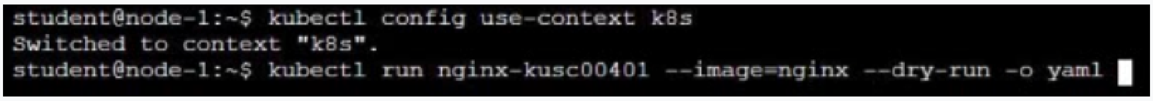
Schedule a pod as follows:

✑ Name: nginx-kusc00401

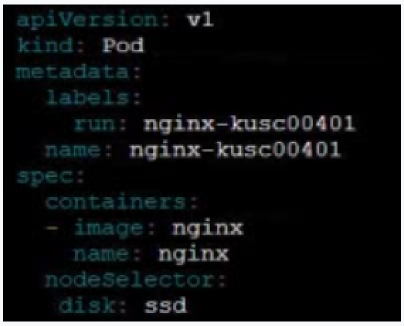
✑ Image: nginx

✑ Node selector: disk=ssd

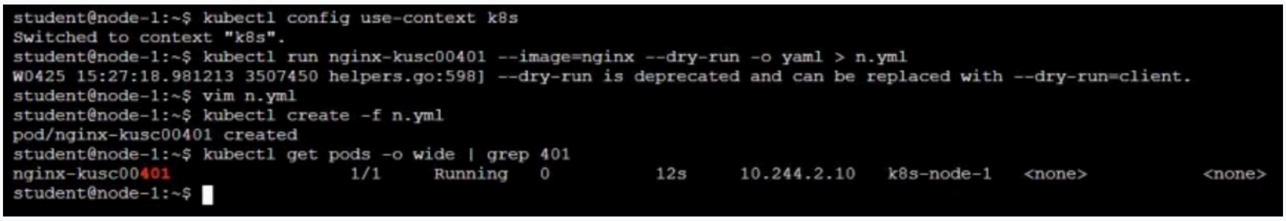
Suggested Answer:



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl run nginx-kusc00401 --image=nginx --dry-run -o yaml
```



```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-kusc00401
  name: nginx-kusc00401
spec:
  containers:
  - image: nginx
    name: nginx
  nodeSelector:
   disk: ssd
```



```
student@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
student@node-1:~$ kubectl run nginx-kusc00401 --image=nginx --dry-run -o yaml > n.yml
W0425 15:27:18.981213 3507450 helpers.go:598] --dry-run is deprecated and can be replaced with --dry-run=client.
student@node-1:~$ vim n.yml
student@node-1:~$ kubectl create -f n.yml
pod/nginx-kusc00401 created
student@node-1:~$ kubectl get pods -o wide | grep 401
nginx-kusc00401                 1/1     Running   0          12s     10.244.2.10    k8s-node-1    <none>           <none>
student@node-1:~$
```

k8s_psg Highly Voted 1 year, 10 months ago

@pentium2000 it's not disk=ssd, it should be disk: ssd

upvoted 13 times

pentium2000 Highly Voted 1 year, 11 months ago

https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/

apiVersion: v1

kind: Pod

metadata:

name: nginx-kusc00401

spec:

containers:

- name: nginx-kusc00401

image: nginx

```
nodeSelector:
disk=ssd
```

upvoted 6 times

ScrewOnPrem 1 year, 9 months ago

```
nodeSelector:
disktype: ssd
```

https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/

In above url itself it is clearly mentioned.

upvoted 4 times

siriseniaws Most Recent 10 months, 1 week ago

Hers first we have to do the node label.

```
kubectl label no node02 disk=ssd
```

then execute below code

```
root@master:~# cat nodeSelector.yml
apiVersion: v1
kind: Pod
metadata:
labels:
run: nginx-kusc00401
name: nginx-kusc00401
spec:
containers:
- image: nginx
name: nginx
nodeSelector:
disk: ssd
root@master:~#
```

then only pod will be in running state.

```
root@master:~# kubectl get po -o wide | grep 401
nginx-kusc00401 1/1 Running 0 7m52s 10.0.2.128 node02 <none> <none>
root@master:~#
```

upvoted 1 times

cp2323 1 year ago

first label the node or check if the label exist on the node

```
kubectl get nodes --show-labels | grep -i disk=ssd
```

if NOT exist then run - kubectl label node node01 disk=ssd

then

```
apiVersion: v1
kind: Pod
metadata:
name: nginx-kusc00401
spec:
containers:
- name: nginx
image: nginx
imagePullPolicy: IfNotPresent
nodeSelector:
disk: ssd
```

upvoted 2 times

LavaPup 1 year, 1 month ago

For some reason, the code provided by Pentium or ScrewOnPrem didn;t work for me until I added the labels under metadata

upvoted 1 times

bp339 1 year, 5 months ago

```
apiVersion: v1
kind: Pod
```

```
metadata:
name: nginx-kusc00401
spec:
containers:
- name: nginx
image: nginx
imagePullPolicy: IfNotPresent
nodeSelector:
disk: ssd
```

this works for me

upvoted 1 times

kopper2019 1 year, 7 months ago

```
root@cka-master1:~ # kubectl label nodes cka-node1 disk=spinning

root@cka-master1:~ # kubectl get nodes --show-labels

root@cka-master1:~ # vim nodeselector.yaml
apiVersion: v1
kind: Pod
metadata:
name: nginx-kusc00401
spec:
containers:
- name: nginx
image: nginx
imagePullPolicy: IfNotPresent
nodeSelector:
disk: spinning

#
root@cka-master1:~ # kubectl apply -f nodeselector.yaml
```

upvoted 2 times

dkjwr 1 year, 7 months ago

The "nodeselector" arguement is looking a a label on the node. Use-

```
kubectl get nodes --show-labels
```

to check what that label actually is.

The examples using "disktype" from the docs are using that because that is the label name that has been attached to the node.

P.S. It's a ":" not a "=" ->

```
disk: ssd
```

upvoted 4 times

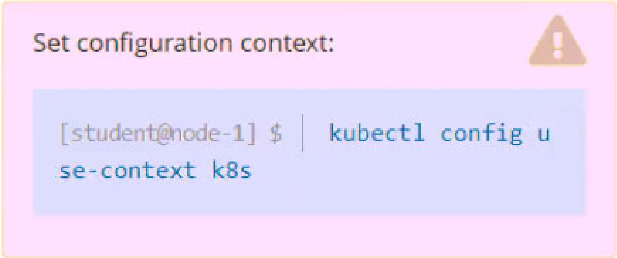
junaid84u 1 year, 10 months ago

nodeSelector is the good and right path .. please don;t use nodeaffinity as its different

upvoted 2 times

Question #9 *Topic 1*

SIMULATION -



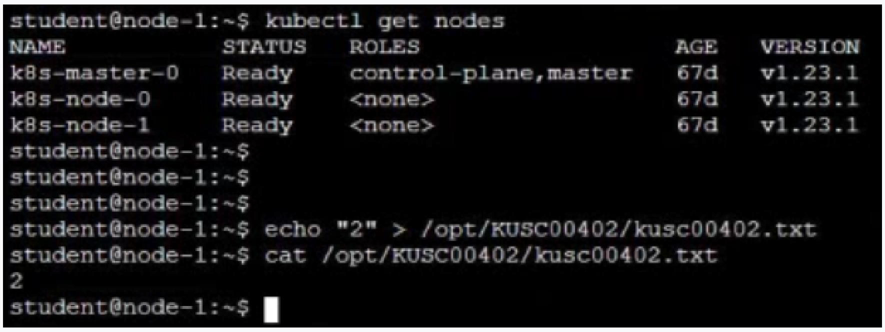
Set configuration context:

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

Task -

Check to see how many nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUSC00402/kusc00402.txt.

Suggested Answer:



```
student@node-1:~$ kubectl get nodes
NAME           STATUS   ROLES                  AGE   VERSION
k8s-master-0   Ready    control-plane,master   67d   v1.23.1
k8s-node-0     Ready    <none>                 67d   v1.23.1
k8s-node-1     Ready    <none>                 67d   v1.23.1
student@node-1:~$
student@node-1:~$
student@node-1:~$
student@node-1:~$ echo "2" > /opt/KUSC00402/kusc00402.txt
student@node-1:~$ cat /opt/KUSC00402/kusc00402.txt
2
student@node-1:~$ █
```

⊟ **SijoTharakan** Highly Voted 2 years, 1 month ago

kubectl get nodes -o=custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taints[*].value,TaintEffect:.spec.taints[*].effect

Reference:
https://kubernetes.io/docs/reference/kubectl/cheatsheet/#interacting-with-nodes-and-cluster

upvoted 19 times

⊟ **leebug** 1 year, 5 months ago

Thanks for the link to the reference!

upvoted 3 times

⊟ **frankja2** Highly Voted 1 year, 1 month ago

are you drunk or on drugs? It says "NOT INCLUDING" but every answer is like getting those

upvoted 6 times

⊟ **noahsark** Most Recent 1 month ago

killer_sh_lab:

k get no

k describe no node01 | g taint

# keep it simple, check each node, there's only a few nodes.

# count number of nodes accordingly not including nodes tainted NoSchedule, sample below:

echo “2” > /opt/KUSC00402/kusc00402.txt

upvoted 1 times

⊟ **137eceb** 1 month, 2 weeks ago

k describe nodes | grep Taint | grep -v NoSchedule| wc -l > filename.txt

upvoted 2 times

⊟ **Archanakaviya** 1 month, 4 weeks ago

echo $(k get nodes --no-headers | grep 'Ready' | grep -v 'NoSchedule' | wc -l) > opt/KUSC00402/kusc00402.txt

upvoted 1 times

⊟ **[Removed]** 4 months, 1 week ago

kubectl get nodes --no-headers | grep -v 'NoSchedule' | grep -c ' Ready ' > /opt/KUSC00402/kusc00402.txt

upvoted 2 times

zanhsieh 7 months, 1 week ago

k8s jsonpath doesn't support multiple filter conditions like:

```
...conditions[?(@.type=="Ready" && @.status=="True")].type
...conditions[?(@.type=="Ready" AND @.status=="True")].type
```

So have to play around with extra grep as below. This shall handle node status Ready.

```
k get no --no-headers -o='custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taints[*].value,TaintEffect:.spec.taints[*].effect,Status:.status.conditio
(@.type=="Ready")].status' | grep -v NoSchedule | grep True > /opt/KUSC00402/kusc00402.txt
```

upvoted 1 times

plamennfs 8 months ago

The easiest solution and smart solution is :

```
k describe nodes | grep -i taint | grep none -c > /opt/KUSC00402/kusc00402.txt
```

upvoted 1 times

wwwmmm 3 months, 1 week ago

I tested, this one works

upvoted 1 times

thanhnv142 8 months, 3 weeks ago

step 1: kubectl get nodes - o yaml | grep taint

step 2: check the tainted and export to the file

upvoted 2 times

fc146fc 9 months, 3 weeks ago

```
kubectl get nodes –no-headers -o='custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taints[*].value,TaintEffect:.spec.taints[*].effect' | grep -v NoSchedule | wc -l > /opt/KUSC00402/kusc00402.txt
```

upvoted 2 times

Einthu 1 year, 2 months ago

```
k get nodes -o jsonpath='{range.items[*]}{.metadata.name}{"\t"}{.spec.taints[?(.effect == "NoSchedule")]}{"\n"}{end}'
```

upvoted 2 times

femijohn123 1 year, 2 months ago

```
k get nodes -o yaml --no-headers | grep -i taint | wc -l #this will show the tainted nodes
```

upvoted 1 times

Samm1 1 year, 3 months ago

From my understanding of the question:

-c command will count the occurrence, for example of 2 node with 1 tainted node

```
kubectl get nodes -o=custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taints[*].value,TaintEffect:.spec.taints[*].effect| grep -c NoSchedule > to1.txt
cat to1.txt. # 1
```

upvoted 2 times

Shenannigan 1 year, 3 months ago

I was able to accomplish this with the following command:

```
k get nodes --no-headers -o custom-columns=Name:.metadata.name,Taint:.spec.taints[*].effect,Ready:'{.status.conditions[?(@.reason == "KubeletReady")].status}' | grep -v NoSchedule | wc -l
```

you can remove the | wc -l to see the output

upvoted 1 times

Shenannigan 1 year, 3 months ago

modified to pull type instead of True/False value so you get the ready output

```
k get nodes --no-headers -o custom-columns=Name:.metadata.name,Taint:.spec.taints[*].effect,Ready:'{.status.conditions[?(@.reason == "KubeletReady")].type}' | grep -v NoSchedule
```

upvoted 2 times

Shepardos 1 year, 3 months ago

```
kubectl describe node | grep -ie Ready -ie taint
```

upvoted 2 times

mellohello 1 year, 5 months ago

```
controlplane $ vim /opt/KUSC00402/kusc00402.txt
```

write the number of the nodes, then save the doc!

```
controlplane $ cat /opt/KUSC00402/kusc00402.txt
```

upvoted 2 times

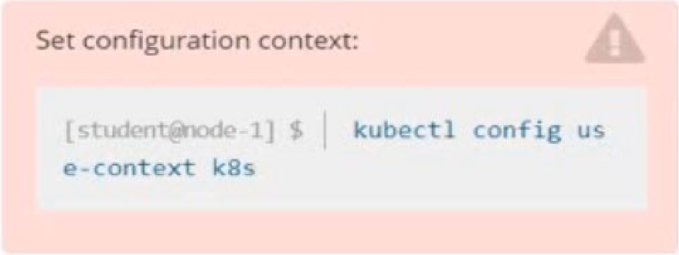
mellohello 1 year, 6 months ago

```
kubectl describe node | grep -i taint > /opt/KUSC00402/kusc00402.txt
```

upvoted 1 times

Question #10 *Topic 1*

SIMULATION -



Set configuration context:

```
[student@node-1] $ | kubectl config us
e-context k8s
```

Task -

Schedule a Pod as follows:

✑ Name: kucc8

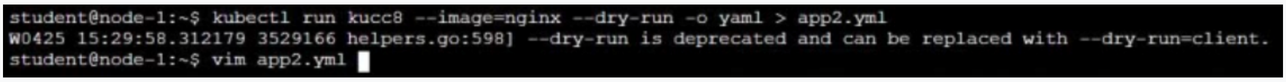
✑ App Containers: 2

✑ Container Name/Images:

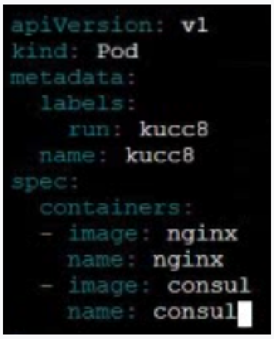
- nginx

- consul

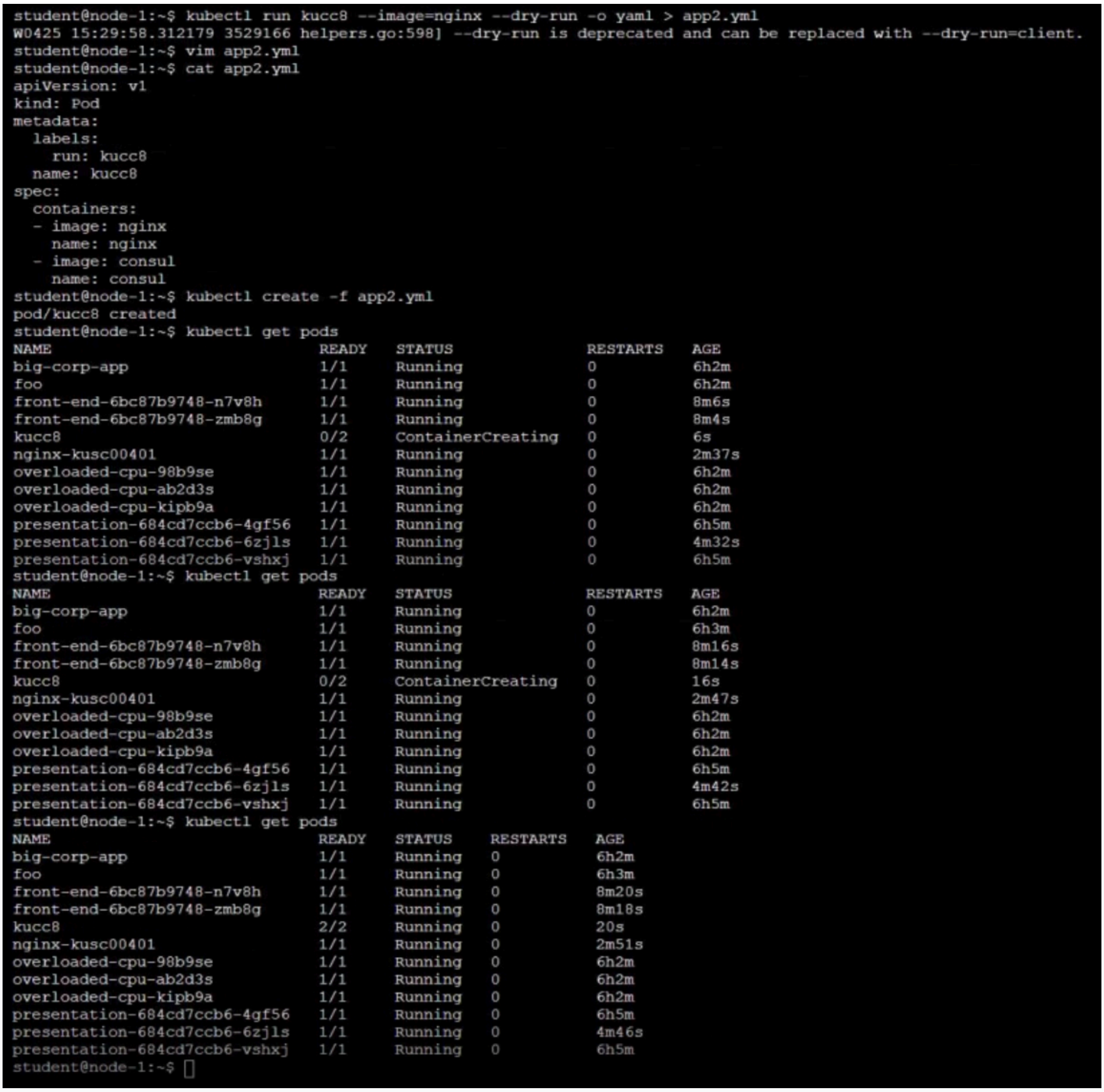
Suggested Answer:



```
student@node-1:~$ kubectl run kucc8 --image=nginx --dry-run -o yaml > app2.yml
W0425 15:29:58.312179 3529166 helpers.go:598] --dry-run is deprecated and can be replaced with --dry-run=client.
student@node-1:~$ vim app2.yml
```



```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: kucc8
  name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: consul
    name: consul
```



```
student@node-1:~$ kubectl run kucc8 --image=nginx --dry-run -o yaml > app2.yml
W0425 15:29:58.312179 3529166 helpers.go:598] --dry-run is deprecated and can be replaced with --dry-run=client.
student@node-1:~$ vim app2.yml
student@node-1:~$ cat app2.yml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: kucc8
  name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: consul
    name: consul
student@node-1:~$ kubectl create -f app2.yml
pod/kucc8 created
student@node-1:~$ kubectl get pods
NAME                              READY   STATUS              RESTARTS   AGE
big-corp-app                      1/1     Running             0          6h2m
foo                               1/1     Running             0          6h2m
front-end-6bc87b9748-n7v8h        1/1     Running             0          8m6s
front-end-6bc87b9748-zmb8g        1/1     Running             0          8m4s
kucc8                             0/2     ContainerCreating   0          6s
nginx-kusc00401                   1/1     Running             0          2m37s
overloaded-cpu-98b9se             1/1     Running             0          6h2m
overloaded-cpu-ab2d3s             1/1     Running             0          6h2m
overloaded-cpu-kipb9a             1/1     Running             0          6h2m
presentation-684cd7ccb6-4gf56     1/1     Running             0          6h5m
presentation-684cd7ccb6-6zjls     1/1     Running             0          4m32s
presentation-684cd7ccb6-vshxj     1/1     Running             0          6h5m
student@node-1:~$ kubectl get pods
NAME                              READY   STATUS              RESTARTS   AGE
big-corp-app                      1/1     Running             0          6h2m
foo                               1/1     Running             0          6h3m
front-end-6bc87b9748-n7v8h        1/1     Running             0          8m16s
front-end-6bc87b9748-zmb8g        1/1     Running             0          8m14s
kucc8                             0/2     ContainerCreating   0          16s
nginx-kusc00401                   1/1     Running             0          2m47s
overloaded-cpu-98b9se             1/1     Running             0          6h2m
overloaded-cpu-ab2d3s             1/1     Running             0          6h2m
overloaded-cpu-kipb9a             1/1     Running             0          6h2m
presentation-684cd7ccb6-4gf56     1/1     Running             0          6h5m
presentation-684cd7ccb6-6zjls     1/1     Running             0          4m42s
presentation-684cd7ccb6-vshxj     1/1     Running             0          6h5m
student@node-1:~$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
big-corp-app                      1/1     Running   0          6h2m
foo                               1/1     Running   0          6h3m
front-end-6bc87b9748-n7v8h        1/1     Running   0          8m20s
front-end-6bc87b9748-zmb8g        1/1     Running   0          8m18s
kucc8                             2/2     Running   0          20s
nginx-kusc00401                   1/1     Running   0          2m51s
overloaded-cpu-98b9se             1/1     Running   0          6h2m
overloaded-cpu-ab2d3s             1/1     Running   0          6h2m
overloaded-cpu-kipb9a             1/1     Running   0          6h2m
presentation-684cd7ccb6-4gf56     1/1     Running   0          6h5m
presentation-684cd7ccb6-6zjls     1/1     Running   0          4m46s
presentation-684cd7ccb6-vshxj     1/1     Running   0          6h5m
student@node-1:~$ 
```

sejar Highly Voted 11 months, 3 weeks ago

hashicorp/consul:latest seems the latest image

upvoted 11 times

caco0516 Highly Voted 1 year, 1 month ago

Got this error with consul image :

image "consul": rpc error: code = NotFound desc = failed to pull and unpack image "docker.io/library/consul:latest": failed to resolve reference "docker.io/library/consul:latest": docker.io/library/consul:latest: not found

Warning Failed 12s (x2 over 27s) kubelet Error: ErrImagePull

Normal BackOff 0s (x2 over 27s) kubelet Back-off pulling image "consul"

Warning Failed 0s (x2 over 27s) kubelet Error: ImagePullBackOff

upvoted 6 times

noahsark Most Recent 1 month ago

killer_sh_lab:

k run kucc8 --image=nginx --dry-run -o=yaml > app2.yml

# edit

apiVersion: v1

kind: Pod

metadata:

labels:

```
run: kucc8
name: kucc8
spec:
containers:
- image: nginx
name: nginx
- image: hashicorp/consul:latest
name: consul
```

```
k apply -f app2.yml
```

upvoted 1 times

cp2323 1 year ago

yeah getting error with consul image, guessing this is quite old question

upvoted 1 times

namesgeo 1 year, 1 month ago

```
kubectl run kucc8 --image=nginx --dry-run=client > kucc8.yml
--------------
vim kucc8.yml like this
---------------
apiVersion: v1
kind: Pod
metadata:
name: kucc8
spec:
containers:
- name: nginx
image: nginx
- name: consul
image: consul
-------------------
kubectl create -f kucc8.yml
```

upvoted 4 times

bp339 1 year, 5 months ago

```
apiVersion: v1
kind: Pod
metadata:
name: kucc8
spec:
containers:
- name: nginx
image: nginx
- name: consul
image: consul
```

upvoted 3 times

mrallrounder123453656 1 year, 6 months ago

```
apiVersion: v1
kind: Pod
metadata:
creationTimestamp: null
labels:
run: kucc8
name: kucc8
spec:
containers:
- image: nginx
```

name: kucc8

- image: consul

name: consul

upvoted 2 times

⊟ **sonixrw** 1 year, 5 months ago

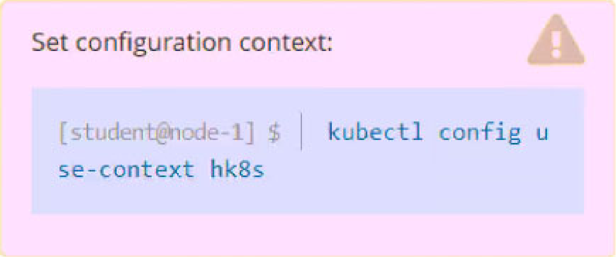
- image: nginx

name: nginx

upvoted 2 times

Question #11 *Topic 1*

SIMULATION -



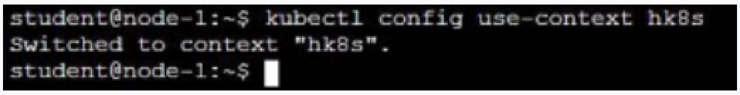
Set configuration context:

```
[student@node-1] $ | kubectl config use-context hk8s
```

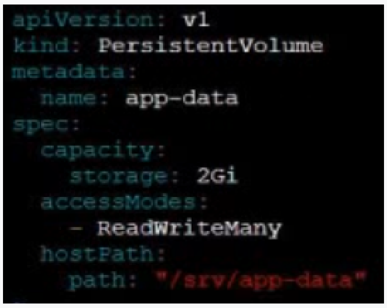
Task -

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app- data.

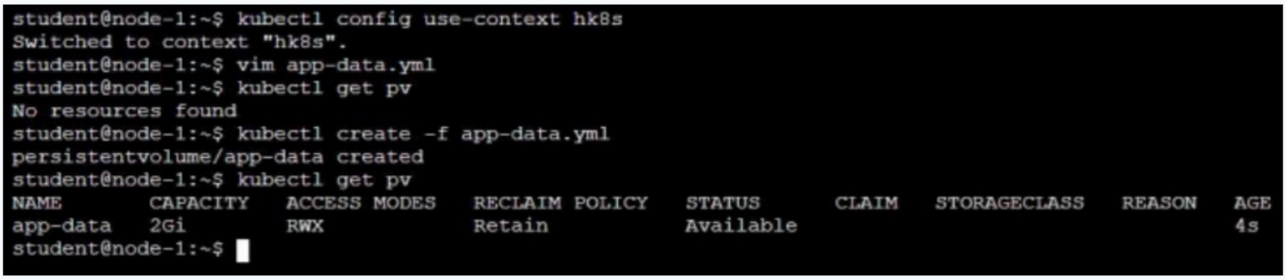
Suggested Answer:



```
student@node-1:~$ kubectl config use-context hk8s
Switched to context "hk8s".
student@node-1:~$
```



```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/srv/app-data"
```



```
student@node-1:~$ kubectl config use-context hk8s
Switched to context "hk8s".
student@node-1:~$ vim app-data.yml
student@node-1:~$ kubectl get pv
No resources found
student@node-1:~$ kubectl create -f app-data.yml
persistentvolume/app-data created
student@node-1:~$ kubectl get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
app-data   2Gi        RWX            Retain           Available                                   4s
student@node-1:~$
```

pentium2000 Highly Voted 1 year, 11 months ago

apiVersion: v1

kind: PersistentVolume

metadata:

name: app-data

spec:

capacity:

storage: 2Gi

accessModes:

- ReadOnlyMany

hostPath:

path: "/srv/app-data"

upvoted 21 times

noahsark Most Recent 1 month ago

killer_sh_lab:

vim app-data.yml

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: app-data
spec:
capacity:
storage: 2Gi
accessModes:
- ReadOnlyMany
hostPath:
path: "/srv/app-data"


k get pv
k apply -f app-data.yml
```

upvoted 1 times

Archanakaviya 1 month, 4 weeks ago

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: app-data
spec:
capacity:
storage: 2Gi
accessModes:
- ReadOnlyMany
hostPath:
path: /srv/app-data
```

upvoted 1 times

BABU97 9 months, 1 week ago

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: app-data
spec:
capacity:
storage: 2Gi
accessModes:
- ReadWriteMany
hostPath:
path: "/srv/app-data"
~
```

upvoted 2 times

fc146fc 9 months, 3 weeks ago

```
Vi app-data.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
name: app-data
spec:
capacity:
storage: 2Gi

accessModes:
- ReadWriteMany
hostPath:
```