CKA MOCK EXAM 5.0

time:2 hour

**submission :output in form of screenshoots or in a file ..(choice is yours)**

**QUESTION 1**

1. There are various Pods in all namespaces. Write a command into **/opt/course/6/find\_pods.sh** which lists all Pods sorted by their **AGE** .

2. Write a second command into **/opt/course/6/find\_pods\_uid.sh** which lists all Pods sorted by field **metadata.uid** .

Use kubectl sorting for both commands

**Answer:**

Search “kubectl Quick Reference”

<https://kubernetes.io/docs/reference/kubectl/quick-reference/>

Related command, **kubectl get events --sort-by=.metadata.creationTimestamp**

Create folder **/opt/course/6/** and create a shell file for **find\_pods.sh**

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated  
A computer screen shot of a browser window

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**QUESTION 2**

1. Create a yaml file called **nginx-dp.yaml** for a deployment of four replicas of nginx, listening on the container's port 80.
2. They should have the labels **role=webserver** and **app=nginx**
3. The deployment should be named **nginx-dp**.
4. Expose the deployment with a load balancer and use a curl statement on the IP address of the load balancer to export the output to a file titled output.txt.

**Answers:**

Find **API Overview** page, then find **Kubernetes API reference** hyperlink

<https://kubernetes.io/docs/reference/using-api/>

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.30/#replicaset-v1-apps>

Use **Using a Service to Expose Your App** page to expose deployment, <https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>

Search deployment script to get the related YAML script

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 3**

Scale the deployment you just made, **nginx-dp,** down to 2 replicas

**Answers:**

Edit the script for **nginx-dp**

Change the replica numbers to “2”

Reapply the script

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

**QUESTION 4**

1. Create a yaml file called **db-secret.yaml** for a secret called **db-user-pass**.
2. The secret should have two fields: a username and password.
3. The username should be **master\_admin** and the password should be **imamazing**.

**Answers:**

Find a YAML script for creating secret on **Quick Reference (Creating objects)** page or **Secret** page

<https://kubernetes.io/docs/reference/kubectl/quick-reference/#creating-objects>

<https://kubernetes.io/docs/concepts/configuration/secret/>

Change “data” to “stringData” if you don’t want to encode the string in base64

A yellow and black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**QUESTION 5**

1. Create a configmap named **mustard** with two values **mustard=yellow** and

# mayo=white.

1. Create a pod with the latest busybox image running a sleep for 1 hour, where you have two environment variables taken from the above configmap and show them in the container.

**Answers:**

Find a **ConfigMap** page to create a configmap command and how to use it in the pod

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#create-a-configmap>

or **kubectl create configmap** page

<https://kubernetes.io/docs/reference/kubectl/generated/kubectl_create/kubectl_create_configmap/>

A screenshot of a computer

Description automatically generated

Go to **Define container environment variables using ConfigMap data** section on how to create a pod using ConfigMap data

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#define-container-environment-variables-using-configmap-data>

Create a script to run the pod

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Alternative:**

<https://gist.github.com/alekc/48753c2d18e87fbccde2ec908a2e19b0#use-configmaps-and-secrets-to-configure-applications>

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**QUESTION 6**

1. Create any pod of your choice
2. Perform a backup of the etcd data store.
3. Delete the pod you created
4. Perform a restore of the etcd data store.

**Answers:**

Find a **Backing up an etcd cluster** in **Operating etcd clusters for Kubernetes** page

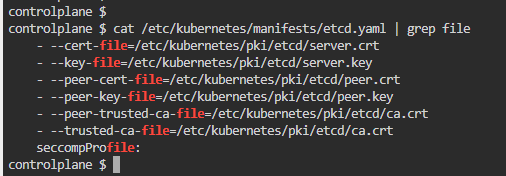
<https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/#backing-up-an-etcd-cluster>

Run a basic pod

A screenshot of a computer

Description automatically generated

Get path to required files



Modify command for creating an etcd snapshot

A screenshot of a computer

Description automatically generated

Verify the snapshot

A screenshot of a computer program

Description automatically generated

Delete the pod

A screenshot of a computer program

Description automatically generated

Restore the etcd snapshot

A screenshot of a computer

Description automatically generated

Change etcd config to use new directory

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**QUESTION 7**

1. Create a new service account with the name **podviewer**.
2. Grant this Service account access to list all Pods in the cluster by creating an appropriate cluster role called **podviewer-role** and ClusterRoleBinding called **podviewer-role-binding**.

**Answers:**

Find Using RBAC Authorization page, go to Command-line utilities & ServiceAccount permissions

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#command-line-utilities>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#service-account-permissions>

Create a service account

A screenshot of a computer screen

Description automatically generated

Create a clusterrole and ClusterRoleBinding

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 8**

Create a pod with an **Init Container**

Configure a Pod that runs two containers.

The first container should create the **/ data/speedracer.txt**.

The second container should only start once this file has been created. The second container run the **sleep 3600** command as its task.

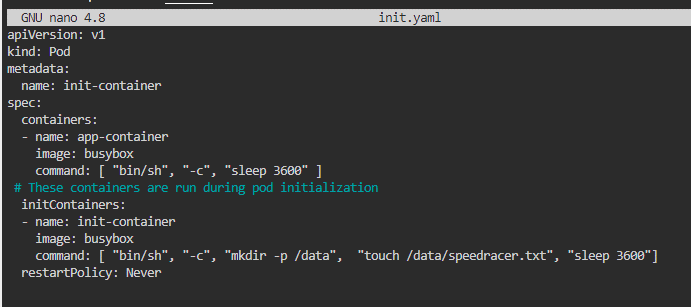
**Answers:**

Go to **Configure Pod Initialization**

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-initialization/>

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#define-container-environment-variables-using-configmap-data>

<https://kubernetes.io/docs/concepts/workloads/pods/sidecar-containers/#pod-sidecar-containers>



Apply the YAML script

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 9**

Create a new user **"joaquin".**

Grant him access to the cluster.

User **"joaquin"** should have permission to create, list, get, update and delete pods.

The private key exists at location:

**/root/joaquin/.key** and csr at **/root/joaquin.csr**

**Answers:**

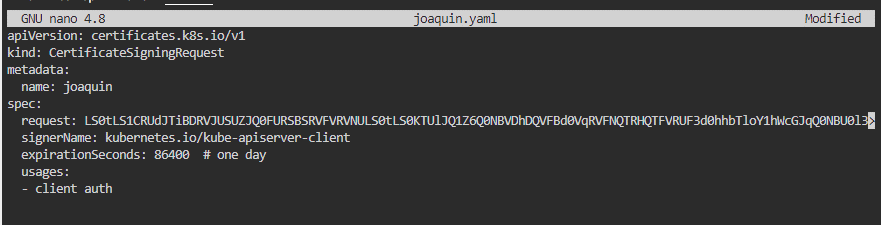
Search **“CSR”**, go to **Certificates and Certificate Signing Requests** page and go to **How to issue a certificate for a user** section

<https://kubernetes.io/docs/reference/access-authn-authz/certificate-signing-requests/#normal-user>

Follow the step on the section, but change role to **clusterrole** and rolebinding to **clusterrolebinding**

A screenshot of a computer program

Description automatically generated



A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**QUESTION 10**

1. Create a YAML file to define a pod named test-pod that uses the **busybox:latest** image.
2. Ensure the pod is scheduled to run only on the node named **node01** (if not using

Killercoda, run on any node you have).

**Answers:**

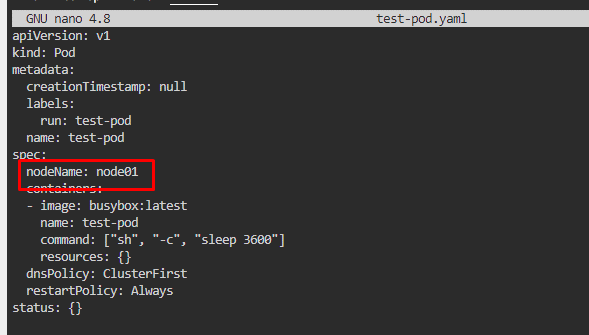
Search --dry-run in quick reference list

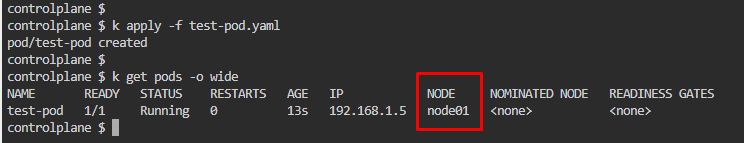
<https://kubernetes.io/docs/reference/kubectl/quick-reference/#interacting-with-running-pods>

A screen shot of a computer

Description automatically generated

Add **nodeName** to ensure the pod run on **node01**





**QUESTION 11**

Create a secret named **danish-secret** with the following key value pairs, then verify that the secret was created with the correct data

# username=admin\_dan

1. **password=Passw0rd1**

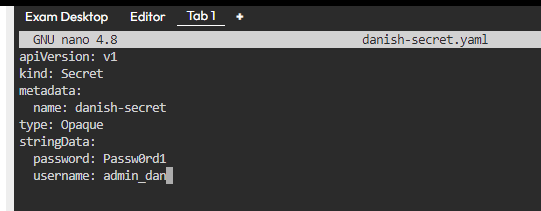
**Answers:**

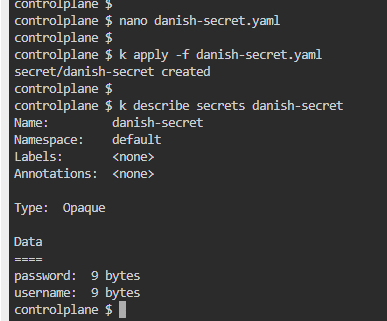
Find a YAML script for creating secret on **Quick Reference (Creating objects)** page or **Secret** page

<https://kubernetes.io/docs/reference/kubectl/quick-reference/#creating-objects>

<https://kubernetes.io/docs/concepts/configuration/secret/>

Change “data” to “stringData” if you don’t want to encode the string in base64





A screen shot of a computer code

Description automatically generated

A screen shot of a computer

Description automatically generated

**QUESTION 12**

1. In a namespace called **prototype,** user **sam** should be allowed to create and delete Pods, Deployments and StatefulSets in Namespace **prototype**
2. User **sam** should have view permissions (like the permissions of the default ClusterRole named view) only in **etcd-controlplane** and **kube-scheduler- controlplane** namespaces
3. Verify everything using **kubectl auth can-i**

**Answers:**

Search **“CSR”**, go to **Certificates and Certificate Signing Requests** page and go to **How to issue a certificate for a user** section

<https://kubernetes.io/docs/reference/access-authn-authz/certificate-signing-requests/#normal-user>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/>

Follow the step on the section

Create namespace

A screenshot of a computer

Description automatically generated

Follow the step on the section to create user

A screenshot of a computer

Description automatically generated

A screenshot of a computer

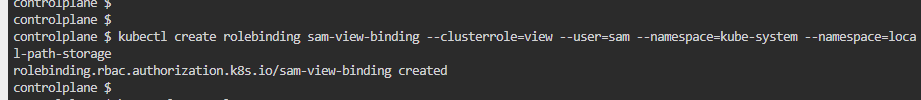
Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer code on a black background

Description automatically generated



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 13**

1. Upgrade the Cluster (Master Node)
2. Make sure to first drain both Node and make it available after upgrade.
3. Note: You may upgrade the cluster to any version you choose.

**Answers:**

Go to **Upgrading kubeadm clusters** page

<https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/>

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 14**

Create a new namespace named **mayonnaise**.

Create a new network policy named **my-policy** in the **mayonnaise** namespace Requirements:

1. Network Policy should allow pods within **mayonnaise** to connect to each other only on port 80.
2. No other ports should be allowed

2. No PODs from outside **mayonnaise** should be able to connect to any pods inside

# mayonnaise

**Answers:**

Create a namespace, **mayonnaise**

A screenshot of a computer

Description automatically generated

Go to The NetworkPolicy resource page

<https://kubernetes.io/docs/concepts/services-networking/network-policies/#networkpolicy-resource>

Copy YAML script and modify it, create a YAML file and apply it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Testing (not necessary)

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**QUESTION 15**

1. Create a nginx pod called **dns-resolver-pod** using image **nginx:latest**
2. Expose it internally with a service called **dns-res—svc**
3. Check if pod and service name are resolvable from within the cluster
4. Use the image: **busybox:1.28** for DNS lookup
5. Save the result in **/root/nginx.svc**

**Answers:**

Ref,

<https://kubernetes.io/docs/reference/kubectl/generated/kubectl_create/kubectl_create_service_clusterip/>

<https://kubernetes.io/docs/tasks/administer-cluster/dns-debugging-resolution/#create-a-simple-pod-to-use-as-a-test-environment>

Create pod with nginx image

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

**QUESTION 16**

Create a static pod named **alpine-static** on the node **node01** that uses the image

# nginx:alpine

**Answers:**

Dry run a script on controlplane, copy the script

Create a YAML file in the manifest folder at node01A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Check pod that automatically run

A screenshot of a computer screen

Description automatically generated

**QUESTION 17**

Create a Persistent Volume (PV) and a corresponding Persistent Volume Claim (PVC) and then use it in a Pod.

1.Create a Persistent Volume named my-pv with the following specifications:

Capacity: **1Gi**

Access Modes**: ReadWriteOnce**

HostPath: **/mnt/data**

Storage Class: **manual**

2.Create a Persistent Volume Claim named my-pvc with the following specifications:

Requested Storage: **1Gi**

Access Modes: **ReadWriteOnce**

Storage Class: **manual**

3.Create mount the volume at **/usr/share** Pod named **nginx-pod** that uses the **PVC my-pvc** to are**/nginx/html**.

4.Verify that the Pod is running and using the Persistent Volume.

**Answers:**

Go to **Configure a Pod to Use a PersistentVolume for Storage** page

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/>

Create a YAML file for creating PV and PVC

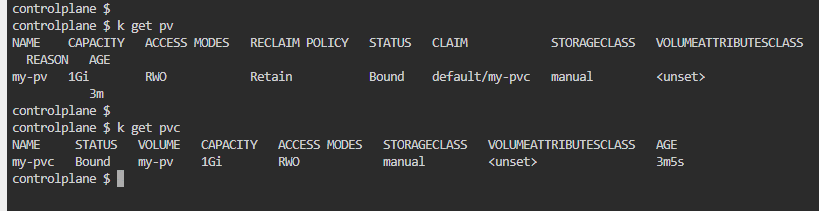
A screenshot of a computer

Description automatically generated

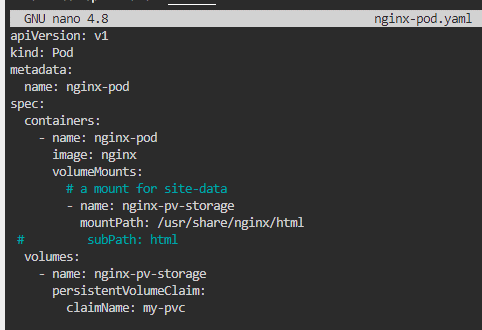
A screen shot of a computer

Description automatically generated

Check PV and PVC created



Create a YAML file for for the pod and mount the volume created



A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated