DBCLOUD
EXAMS

QUESTIONS & ANSWERS

verified



Technology Provider: The Linux Foundation

Exam: Certified Kubernetes Security Specialist (CKS)

Number of questions in the database: 72

Exam Version: NOV.2026

Question #1

A container image scanner is set up on the cluster.

Given an incomplete configuration in the directory

/etc/kubernetes/confcontrol and a functional container image scanner with HTTPS endpoint https://test-server.local.8081/image_policy

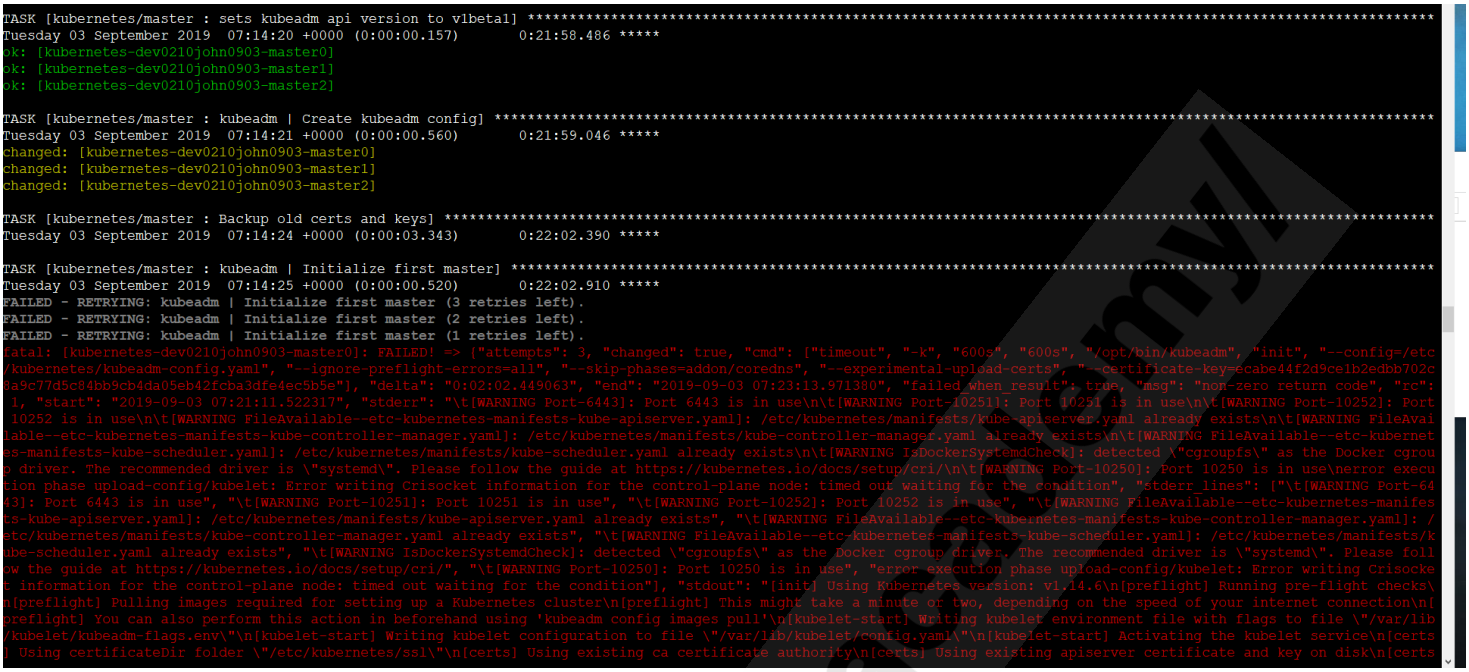
1. Enable the admission plugin.

2. Validate the control configuration and change it to implicit deny.

Finally, test the configuration by deploying the pod having the image tag as latest.

Explanation:
ssh-add ~/.ssh/tempprivate
eval '$(ssh-agent -s)'
cd contrib/terraform/aws
vi terraform.tfvars
terraform init
terraform apply -var-file=credentials.tfvars
ansible-playbook -i ./inventory/hosts ./cluster.yml -e ansible_ssh_user=core -e bootstrap_os=coreos -b --become-user=root --flush-cache -e ansible_user=core

https://dbcloud.academy/



Question #2

Context

A Role bound to a Pod's ServiceAccount grants overly permissive permissions. Complete the following tasks to reduce the set of permissions.

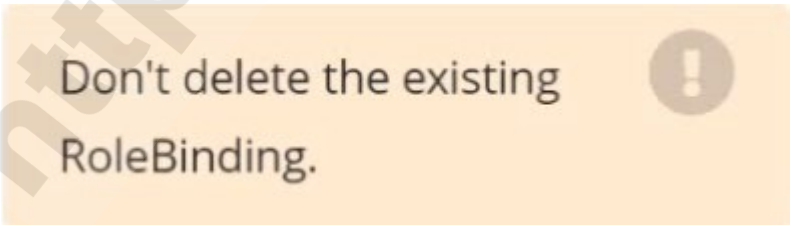
Task

Given an existing Pod named web-pod running in the namespace security.

Edit the existing Role bound to the Pod's ServiceAccount sa-dev-1 to only allow performing watch operations, only on resources of type services.

Create a new Role named role-2 in the namespace security, which only allows performing update operations, only on resources of type namespaces.

Create a new RoleBinding named role-2-binding binding the newly created Role to the Pod's ServiceAccount.



https://dbcloud.academy/

Explanation:

```
candidate@cli:~$ kubectl config use-context KSCH00201
Switched to context "KSCH00201".
candidate@cli:~$ kubectl get pods -n security
NAME      READY   STATUS    RESTARTS   AGE
web-pod   1/1     Running   0          6h9m
candidate@cli:~$ kubectl get deployments.apps -n security
No resources found in security namespace.
candidate@cli:~$ kubectl describe rolebindings.rbac.authorization.k8s.io -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  dev-role
Subjects:
  Kind            Name      Namespace
  ----            ----      ---------
  ServiceAccount  sa-dev-1
candidate@cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  *          []                 []              [*]
candidate@cli:~$ kubectl edit role/dev-role -n security
```

```
  uid: b4c9ddd6-2729-43bd-8fbd-b2d227f4c4cd
rules:
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - watch
```

https://dbcloud.academy/

```
candidate@cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  *          []                 []              [*]
candidate@cli:~$ kubectl edit role/dev-role -n security
role.rbac.authorization.k8s.io/dev-role edited
candidate@cli:~$ kubectl describe role dev-role -n security
Name:         dev-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  services   []                 []              [watch]
candidate@cli:~$ kubectl get pods -n security
NAME      READY   STATUS    RESTARTS   AGE
web-pod   1/1     Running   0          6h12m
candidate@cli:~$ kubectl get pods/web-pod -n security -o yaml | grep serviceAccount
  serviceAccount: sa-dev-1
  serviceAccountName: sa-dev-1
      - serviceAccountToken:
candidate@cli:~$ kubectl create role role-2 --verb=update --resource=namespaces -n security
role.rbac.authorization.k8s.io/role-2 created
candidate@cli:~$ kubectl create rolebinding role-2-binding --role
--role   --role=
candidate@cli:~$ kubectl create rolebinding role-2-binding --role=role-2 --serviceaccount=se
curity:sa-dev-1 -n security
rolebinding.rbac.authorization.k8s.io/role-2-binding created
candidate@cli:~$
```

Question #3

Context

The kubeadm-created cluster's Kubernetes API server was, for testing purposes, temporarily configured to allow unauthenticated and unauthorized access granting the anonymous user duster-admin access.

Task

Reconfigure the cluster's Kubernetes API server to ensure that only authenticated and authorized REST requests are allowed.

Use authorization mode Node,RBAC and admission controller NodeRestriction.

Cleaning up, remove the ClusterRoleBinding for user system:anonymous.

https://dbcloud.academy/

All kubectl configuration contexts/files were also configured to use the unauthenticated and unauthorized access. You don't have to change that, but be aware that kubectl 's configuration will stop working, once you've completed securing the cluster.

You can use the cluster's original kubectl configuration file /etc/kubernetes/admin.conf , located on the cluster's master node, to ensure that authenticated and authorized requests are still allowed.

Explanation:

https://dbcloud.academy/

```
candidate@cli:~$ kubectl config use-context KSCH00101
Switched to context "KSCH00101".
candidate@cli:~$ ssh ksch00101-master
Warning: Permanently added '10.240.86.190' (ECDSA) to the list of known hosts.

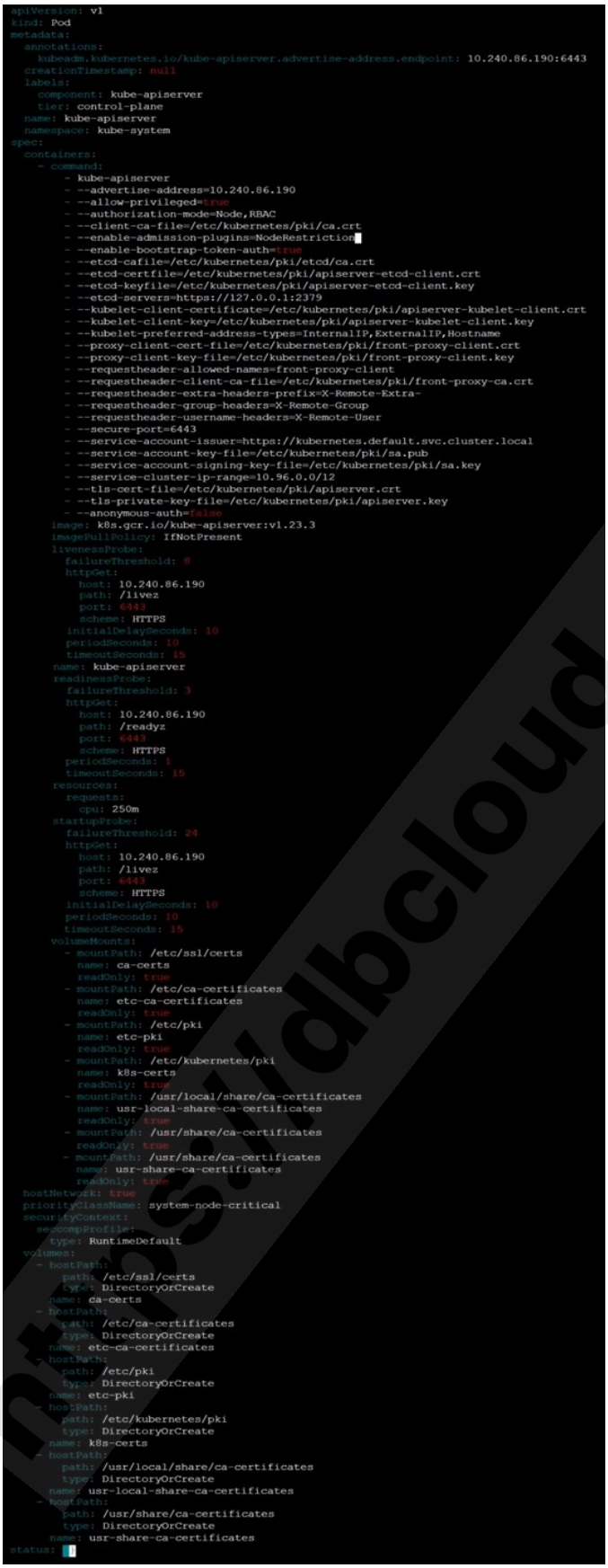
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.190
        - --allow-privileged=true
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=AlwaysAdmit
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
"/etc/kubernetes/manifests/kube-apiserver.yaml" 128L, 4343C                1,1           Top
```

https://dbcloud.academy/


root@ksch00101-master:~# cat /etc/kubernetes/admin.conf
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJB
Z01CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1ESXhO
akF3TlRVeE9Wb1hEVE15TURJeE5EQXdOVFV4T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdE
UVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlgwCm9LeUYvTGNmYTIvNzNZTktkSFdZU3JUaUx0QStr
N01qTXpRZllzM2ttNGllalpoM0tZc3YlbUdpN0UyQ2tYc0MKUnhlLlNiZnBDMzlla2k5V3hOSHc5eTM0OEtXUVE3VXBL
UmZRdXVxdlAlWXdDZkordlJmWGNGTXQxLzRNQVhWLwpkdjZ5YWRKSitPeFFSVjZlaHFBZHR0M3FtOFdVcW84UE5JT1E0
OEc3WWhnRUg5RHU3SFdkMS8raXVkSjNOMXl6CnNISEdtYklsWENSbEcydFV0M2RScDczSnRIS1JjS2tnMGxYM3FWS1Uy
QmJRblBmK01wb0VlTXFGcmZvcWVaVWcKYlBKK3ROVmZIM1JLTkhVUnYydVJIa3ZZc2JrclhUMW8rMXFNNHZrYnFNMHlq
KzNxTUtiSyt5V3dzUTlBYUVPMApUdXR4UUd1TFp3OUE3TjZZeTFVQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURB
Z0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZEcUlwLzdYbzZaNkJNVjVEK2w3bFZPcGpBOWlN
QlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBSlNWNm9wNGgxYkNv
eGZLRUZ4bwoxaVlHUFlnMlhhOTN0WEZlTTY3RnA2NkdqUEc5SXBONnNHUnRnWVlyd0Mya1BDeFVOb2IySWtUQ1FNbDV3
cWRHCkdPS2JwVVp6Smc3Y0dyS2E3R1pZWVNyVUVGRWhyd2xZWXNGME56aFBoZVcwcHJjcWtSdXN1bm55SG5YNGVOMUoK
N1NzbGZYTjJIdVFJd1VIRGl5L0JsL1ZWRmZNZnRxOGF0Z0pYSFZGTmlVcDRpNXlJTXFRNTB4ZjVqcnF1WFRmVwpVdmJq
ZjEyOThXVTk3QkxHcDdRZE9QYWVKU051UStlVkMrdnpVZ2tVQVNjclVsc24xcThPNnBRbjV3TjNxdUVrCm5zQk9pckxS
c2k2alN3UlhLbGcvangvcitqd0dTc0xwWUxDZTlxa1FraTdCSVRJT1N3ejd3c2hzbERuNzBFY0IKa0VBPQotLS0tLUVO
RCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://10.240.86.190:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ01
ocEdQcDB4Zk9JbkYxaGJwcTh5YlBUMGx1Tm5VNjBiSUpxRXVKckxJbEtXClNVa1hlVkYzNkl0ZHc1ZU1OT2JxK1haaHd
hY2JURVZCMlVDVURsbDgzdG5teFQyVXJmY0pUQmhLTCtZTFAvcWYKdjdXR3BwQ1ZXNnhVZGFibGNuUklIMnpleUVJTEt
Tck5XbUQ0TzZsMU13blZOOVJzQ2RXTkV3VGNZRHdoUTd2OQpGcExKL3hiSDdUTzkwY1RFd1Iwazl3cFVYdllkdklSXN
MRkYwL3F2bDA3U3lxbGplOEI1SnNpQlhCU1ZxbS9wCmNUUSs3SnZlbmdaZzlkOWdZaVJVdFFTcHBONkx4UnhkSzNKMGR
BK240SWxFZEtHRWh3TE00d0tMaldERG9scHgKYzB3WHkwVXBORGZ6UUxuRUFzVUJsbDRCQ3VkdW5QNVVDN2FuS3dJREF
RQUJBb0lCQURWRkZNSVRqYnNySTZTTwpQOGM0MTByN3RWZ251cXJVS202dHRnZWtXOWdlSlpvMnZyb3RsbG9qOGFRamF
0MTZnaEUwOXdZd2xMSDhId0tLCklMb2NrZnFCUytlOWo1Zm1FWGxYTG00cElCVDFRbGFJQlJRMDRyQ0JZbHdCN1VFbVB
lWjhuQ3lmR2JYTC9HM2wKcXBYTDVKdzJqcVh2MXdzcWsrdWNCRk0zZ0FYZk5YZkhlRExnV0VyNXRZRlF4VXo5UFFHOD1
pcDY1OTBkYnB1SApOMnU2NGk4UTgldk83OFVIT1c2eUFZU1loZVdha093RDFwZzNPdkhxV3FhbnV1MnlrOWxaUUR0WW5
2MytBeU5DCnloN1RaRHluZ01ZdEptbDFTQ01TNEpSR2d4NXNwaCtKOC9XOGx0Ri9wMWZxbTA0bXZSRndxU3M2Y1JCQ2Z
PVVcKbFV1MGxLRUNnWUVBNWJzT01vVzFBVndjTmJsc0pSVDNURkI2OVlxbDRYcnZRR0FZY3BhdktENnd5VmtEOTV1QQp
SaXVRS1NNKzY4REtBVm1pYllpaThJemExTkdqdC9JZDUwTGVoNklaRVg2enVpK0g3d1BSbVd6SE9ueWNmU2FmClVQMEF
RL0RiM2lCNWJQTmJHYXNkaDNIb2JvR0NSSHZmTFFXY2tYbUVXM2ZudVlIRlJLZ2xlTEVDZllFQTVUdysKTEVTV1BESFF
mamNBN0htNmdsMndGRjdCUG1sSGdaYVVRN25Eb3ZvRmMxalBMRWVCMWJ6OHJNWld1eGdmaHN0OQpMZ0xSUDBXdkJWdlJ
sVTdMTmFLTlVzRmkxU2dvaWZsS01ZWkMyZmpLWTYlRFE3YUUzcTdnVis4U2pIZHpoclhCCkVQclAvWXQ3S0QrbFBMZmh
aNXNKZWFtelY3b3gveno5Y0s0U0ZKc0NnWUJ4OVk2VzFydHBoMFcvS05JS3V4SEoKMjRxRFQxbmlObE9FdmFhakFUaTJ
ZQkxXYnIvWERsNWRjTEs4bFcxYkNYR3JwY2s3U0xKN1hZUVlXajQ2dTNJMgpEQ2ZUWlFiRWRQTzNBbWtPR2ZqWmdPcDd
pdUVkL0JDLzNpRkprcXVlenNFdFdMTHlVcjM5T0hZeWl6QVJ4Tmo2CnZuUGlma000Rkl6d3F2MHVoN0x1blFLQmdBT1Z
XZTRZM1RwbzJ3aEswbmVkMllsMXhVNjJoZ2JiVHcvaVdhdVcKY3ZMV3dlZUlmd0Q4MVRWL2R3a29KVEM1VEJRUXQzUkk
xRFFnVmtnMHFwcUtOOGhDNGQwM05MRzIwTWdZMk94WgpjSFZzK2J4elYwVVB6V1RUbEMyVEsyamhmOHVRcndzSktxY2N
OU0ZEczZwclhocThsV213Znd3aWlBR1hLSFJRCkE3RkxBb0dCQUx3NW8rbHFVZ3hHQlpKdy9ElRGek5TekQreVd6Um8
1c2ZEc2x6a2FvY0pHbEx2MUNndEVIc3QKeG5HMTlIYStSM1M3cDRtei9LeDJYMFRzaTZzUzVwWlR5WEx5STF5azh2TUZ
rR1dacjRmeVhXV2t3SjZlVEl1YwpyWFl3TWM5VFlDUGZrSFJaTm9XR1hZV3BkeTJBOXZCbFlScHZsQVZoenU2T1VZQ2w
5b2ZpCi0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml



```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.240.86.190:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
    - command:
        - kube-apiserver
        - --advertise-address=10.240.86.190
        - --allow-privileged=true
        - --authorization-mode=Node,RBAC
        - --client-ca-file=/etc/kubernetes/pki/ca.crt
        - --enable-admission-plugins=NodeRestriction
        - --enable-bootstrap-token-auth=true
        - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
        - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
        - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
        - --etcd-servers=https://127.0.0.1:2379
        - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
        - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
        - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
        - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
        - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
        - --requestheader-allowed-names=front-proxy-client
        - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
        - --requestheader-extra-headers-prefix=X-Remote-Extra-
        - --requestheader-group-headers=X-Remote-Group
        - --requestheader-username-headers=X-Remote-User
        - --secure-port=6443
        - --service-account-issuer=https://kubernetes.default.svc.cluster.local
        - --service-account-key-file=/etc/kubernetes/pki/sa.pub
        - --service-account-signing-key-file=/etc/kubernetes/pki/sa.key
        - --service-cluster-ip-range=10.96.0.0/12
        - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
        - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
        - --anonymous-auth=false
      image: k8s.gcr.io/kube-apiserver:v1.23.3
      imagePullPolicy: IfNotPresent
      livenessProbe:
        failureThreshold: 8
        httpGet:
          host: 10.240.86.190
          path: /livez
          port: 6443
          scheme: HTTPS
        initialDelaySeconds: 10
        periodSeconds: 10
        timeoutSeconds: 15
      name: kube-apiserver
      readinessProbe:
        failureThreshold: 3
        httpGet:
          host: 10.240.86.190
          path: /readyz
          port: 6443
          scheme: HTTPS
        periodSeconds: 1
        timeoutSeconds: 15
      resources:
        requests:
          cpu: 250m
      startupProbe:
        failureThreshold: 24
        httpGet:
          host: 10.240.86.190
          path: /livez
          port: 6443
          scheme: HTTPS
        initialDelaySeconds: 10
        periodSeconds: 10
        timeoutSeconds: 15
      volumeMounts:
        - mountPath: /etc/ssl/certs
          name: ca-certs
          readOnly: true
        - mountPath: /etc/ca-certificates
          name: etc-ca-certificates
          readOnly: true
        - mountPath: /etc/pki
          name: etc-pki
          readOnly: true
        - mountPath: /etc/kubernetes/pki
          name: k8s-certs
          readOnly: true
        - mountPath: /usr/local/share/ca-certificates
          name: usr-local-share-ca-certificates
          readOnly: true
        - mountPath: /usr/share/ca-certificates
          readOnly: true
        - mountPath: /usr/share/ca-certificates
          name: usr-share-ca-certificates
          readOnly: true
  hostNetwork: true
  priorityClassName: system-node-critical
  securityContext:
    seccompProfile:
      type: RuntimeDefault
  volumes:
    - hostPath:
        path: /etc/ssl/certs
        type: DirectoryOrCreate
      name: ca-certs
    - hostPath:
        path: /etc/ca-certificates
        type: DirectoryOrCreate
      name: etc-ca-certificates
    - hostPath:
        path: /etc/pki
        type: DirectoryOrCreate
      name: etc-pki
    - hostPath:
        path: /etc/kubernetes/pki
        type: DirectoryOrCreate
      name: k8s-certs
    - hostPath:
        path: /usr/local/share/ca-certificates
        type: DirectoryOrCreate
      name: usr-local-share-ca-certificates
    - hostPath:
        path: /usr/share/ca-certificates
        type: DirectoryOrCreate
      name: usr-share-ca-certificates
status: {}
```

https://dbcloud.academy/

```
root@ksch00101-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@ksch00101-master:~# systemctl daemon-reload
sroot@ksch00101-master:~# systemctl restart kubelet.service
root@ksch00101-master:~# kubectl get nodes
error: You must be logged in to the server (Unauthorized)
root@ksch00101-master:~# exit
logout
Connection to 10.240.86.190 closed.
candidate@cli:~$ kubectl get nodes
NAME               STATUS   ROLES                  AGE   VERSION
ksch00101-master   Ready    control-plane,master   93d   v1.23.3
ksch00101-worker1  Ready    <none>                 93d   v1.23.3
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                       READY   STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                    1/1     Running   1 (7h2m ago)    93d
coredns-64897985d-rr7sd                    1/1     Running   1 (7h2m ago)    93d
etcd-ksch00101-master                      1/1     Running   1 (7h2m ago)    93d
kube-apiserver-ksch00101-master            0/1     Running   0               24s
kube-controller-manager-ksch00101-master   1/1     Running   3 (42s ago)     93d
kube-flannel-ds-llktn                      1/1     Running   1 (93d ago)     93d
kube-flannel-ds-q9vnl                      1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                           1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                           1/1     Running   1 (93d ago)     93d
kube-scheduler-ksch00101-master            1/1     Running   3 (42s ago)     93d
candidate@cli:~$ kubectl get pod -n kube-system
NAME                                       READY   STATUS    RESTARTS        AGE
coredns-64897985d-7pnhm                    1/1     Running   1 (7h2m ago)    93d
coredns-64897985d-rr7sd                    1/1     Running   1 (7h2m ago)    93d
etcd-ksch00101-master                      1/1     Running   1 (7h2m ago)    93d
kube-apiserver-ksch00101-master            0/1     Running   0               30s
kube-controller-manager-ksch00101-master   1/1     Running   3 (48s ago)     93d
kube-flannel-ds-llktn                      1/1     Running   1 (93d ago)     93d
kube-flannel-ds-q9vnl                      1/1     Running   1 (93d ago)     93d
kube-proxy-2c4ht                           1/1     Running   1 (93d ago)     93d
kube-proxy-pmmbc                           1/1     Running   1 (93d ago)     93d
kube-scheduler-ksch00101-master            1/1     Running   3 (48s ago)     93d
candidate@cli:~$ kubectl get clusterrolebindings.rbac.authorization.k8s.io | grep anon
system:anonymous                                       ClusterRole/cluster-admin
                                                 7h1m
candidate@cli:~$ kubectl delete clusterrolebindings.rbac.authorization.k8s.io/system:anonymo
us
clusterrolebinding.rbac.authorization.k8s.io "system:anonymous" deleted
```

Question #4

Two tools are pre-installed on the cluster's worker node:

Using the tool of your choice (including any non pre-installed tool), analyze the container's behavior for at least 30 seconds, using filters that detect newly spawning and executing processes.

Store an incident file at /opt/KSRS00101/alerts/details, containing the detected incidents, one per line, in the following format:

```
timestamp,uid/username,proce
ssName
```

The following example shows a properly formatted incident file:

https://dbcloud.academy/