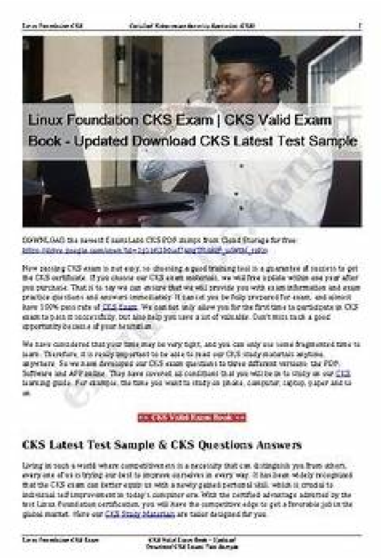
Linux Foundation CKS 100% Correct Answers | Valid CKS Exam Dumps



Linux Foundation CKS Certified Kubernetes Security Specialist (CKS) 1

Linux Foundation CKS Exam | CKS Valid Exam Book - Updated Download CKS Latest Test Sample

DOWNLOAD the newest ExamsLabs CKS PDF dumps from Cloud Storage for free: [illegible]

Now passing CKS exam is not easy, so choosing a good training tool is a guarantee of success to get the CKS certificate. If you choose our CKS exam materials, we will free update within one year after you purchase. That is to say we can ensure that we will provide you with exam information and exam practice questions and answers immediately. It can let you be fully prepared for exam, and almost have 100% pass rate of CKS Exam. We can not only allow you for the first time to participate in CKS exam to pass it successfully, but also help you save a lot of valuable. Don't miss such a good opportunity because of your hesitation.

We have considered that your time may be very tight, and you can only use some fragmented time to learn. Therefore, it is really important to be able to read our CKS study materials anytime, anywhere. So we have developed our CKS exam questions to three different versions: the PDF, Software and APP online. They have covered all conditions that you will be in to study as our CKS learning guide. For example, the time you want to study on phone, computer, laptop, paper and so on.

>> CKS Valid Exam Book <<

CKS Latest Test Sample & CKS Questions Answers

Living in such a world where competitiveness is a necessity that can distinguish you from others, every one of us is trying our best to improve ourselves in every way. It has been widely recognized that the CKS exam can better equip us with a newly gained personal skill, which is crucial to individual self-improvement in today's computer era. With the certified advantage admitted by the test Linux Foundation certification, you will have the competitive edge to get a favorable job in the global market. Here our CKS Study Materials are tailor designed for you.

Linux Foundation CKS Exam CKS Valid Exam Book - Updated Download CKS Latest Test dumps

P.S. Free & New CKS dumps are available on Google Drive shared by TestKingFree: https://drive.google.com/open?id=1cAdIWWv0NxA4_9Zu8kvXwNgR-KI4uwFD

As we all know, it is a must for all of the candidates to pass the exam if they want to get the related CKS certification which serves as the best evidence for them to show their knowledge and skills. If you want to simplify the preparation process, here comes a piece of good news for you. We will bring you integrated CKS Exam Materials to the demanding of the ever-renewing exam, which will be of great significance for you to keep pace with the times.

Linux Foundation CKS (Certified Kubernetes Security Specialist) Certification Exam is an essential certification program for professionals seeking to validate their knowledge and skills in securing Kubernetes clusters. Certified Kubernetes Security Specialist (CKS) certification exam covers a wide range of security topics and is vendor-neutral, making it a valuable credential for professionals working in a variety of industries. CKS Exam is rigorous and performance-based, ensuring that certified professionals possess the necessary knowledge and skills to secure Kubernetes environments effectively.

>> Linux Foundation CKS 100% Correct Answers <<

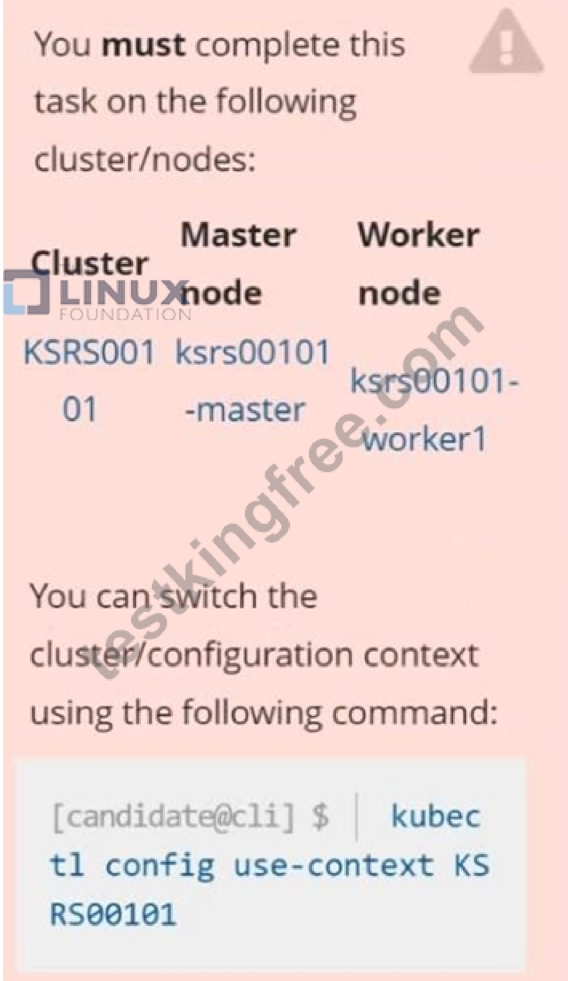
Free PDF CKS 100% Correct Answers | Easy To Study and Pass Exam at first attempt & Updated CKS: Certified Kubernetes Security Specialist

(CKS)

It is carefully edited and reviewed by our experts. The design of the content conforms to the examination outline. Through the practice of our CKS study materials, you can grasp the intention of the examination organization accurately. The number of its test questions is several times of the traditional problem set, which basically covers all the knowledge points to be mastered in the exam. You only need to review according to the content of our CKS Study Materials, no need to refer to other materials. With the help of our CKS study materials, your preparation process will be relaxed and pleasant.

Linux Foundation Certified Kubernetes Security Specialist (CKS) Sample Questions (Q25-Q30):

NEW QUESTION # 25
SIMULATION

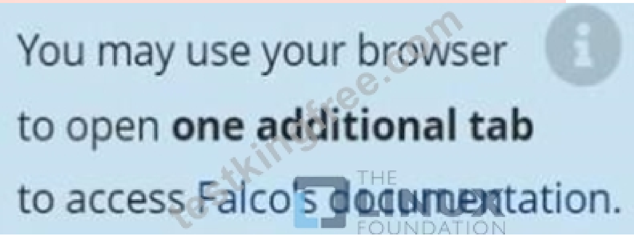


You **must** complete this task on the following cluster/nodes:

Cluster	Master node	Worker node
KSRS00101	ksrs00101-master	ksrs00101-worker1

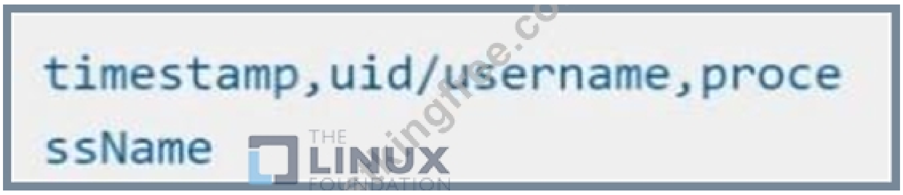
You can switch the cluster/configuration context using the following command:

```
[candidate@cli] $ kubectl config use-context KSRS00101
```



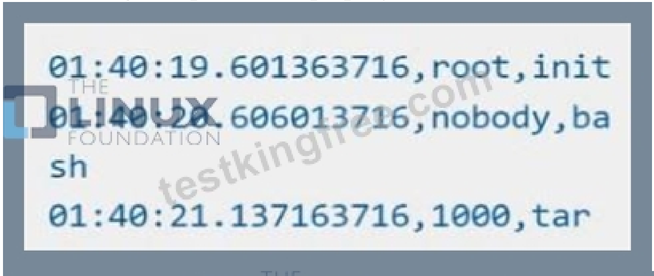
You may use your browser to open **one additional tab** to access Falco's documentation.

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:

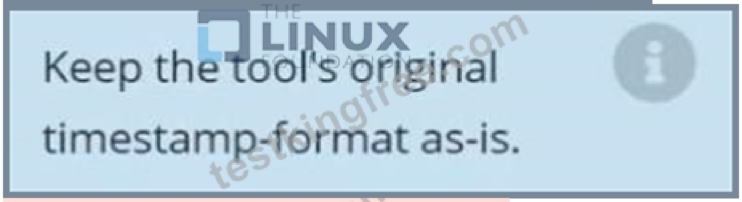


```
timestamp,uid/username,proce
ssName
```

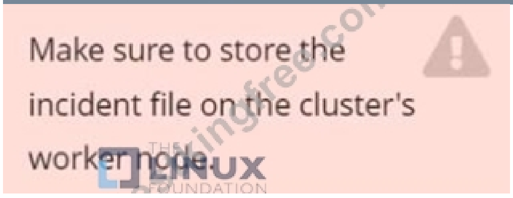
The following example shows a properly formatted incident file:



```
01:40:19.601363716,root,init
01:40:20.606013716,nobody,ba
sh
01:40:21.137163716,1000,tar
```



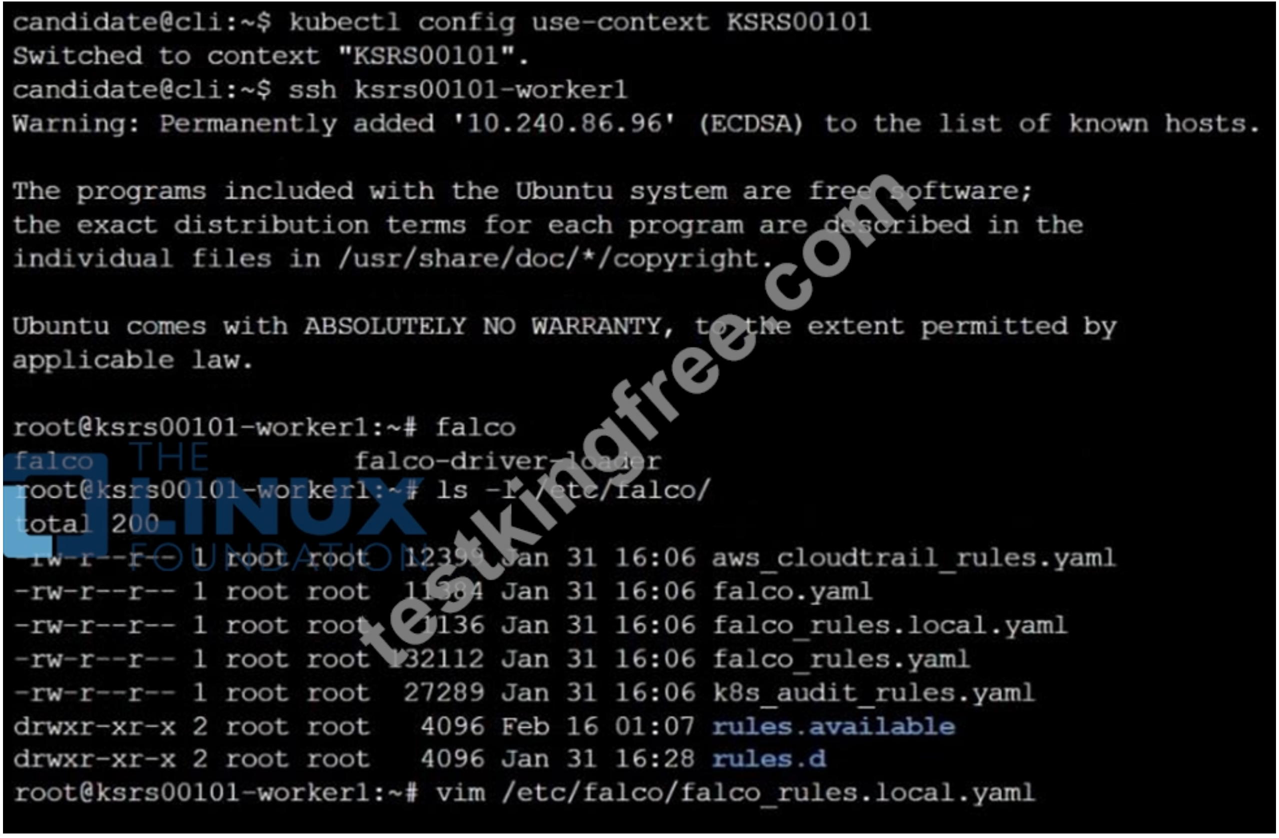
Keep the tool's original timestamp-format as-is.



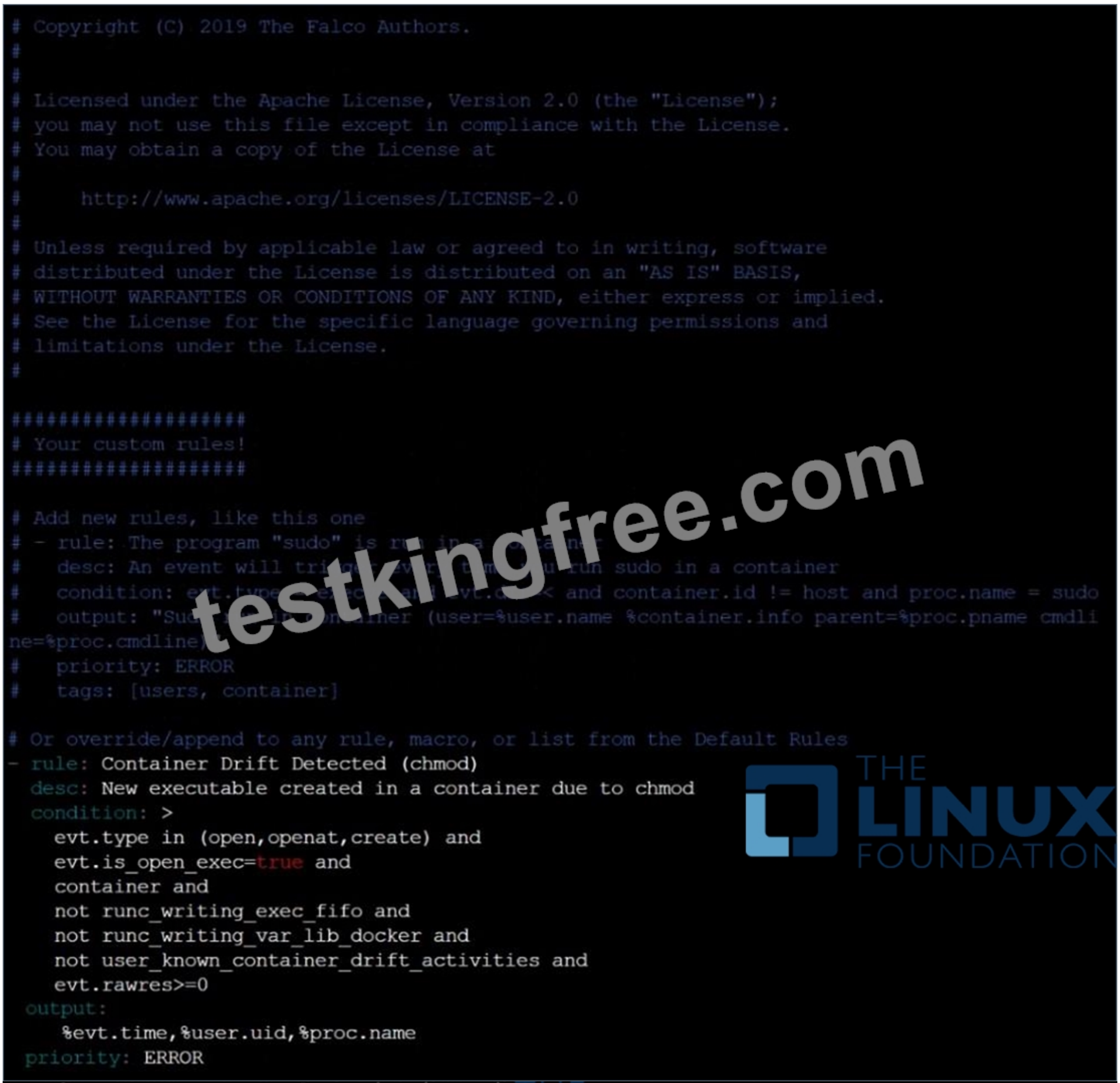
Make sure to store the incident file on the cluster's worker node.

Answer:

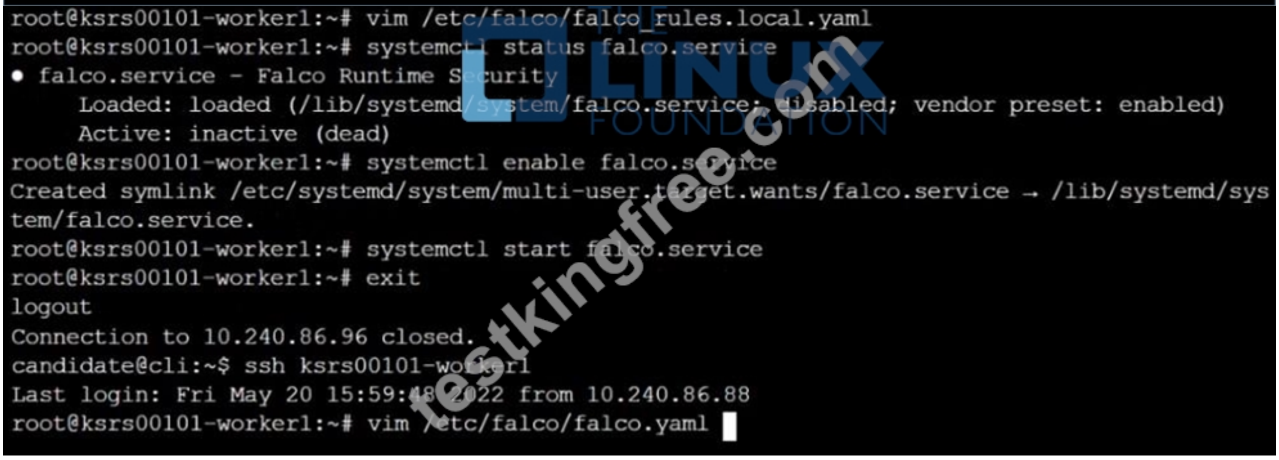
Explanation:
See explanation below
Explanation:



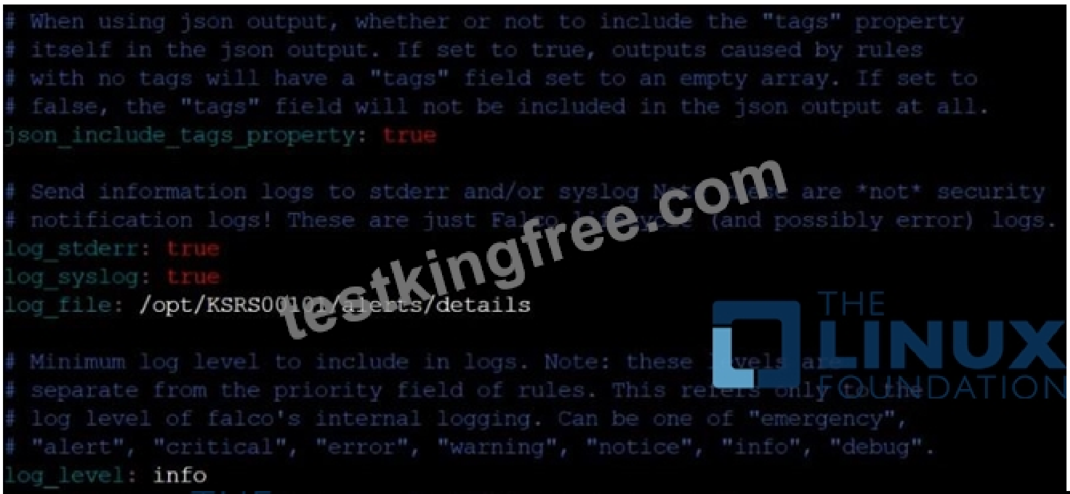
```
candidate@cli:~$ kubectl config use-context KSRS00101
Switched to context "KSRS00101".
candidate@cli:~$ ssh ksrs00101-worker1
Warning: Permanently added '10.240.86.96' (ECDSA) to the list of known hosts.

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@ksrs00101-worker1:~# falco
falco                falco-driver-loader
root@ksrs00101-worker1:~# ls -l /etc/falco/
total 200
-rw-r--r-- 1 root root  12399 Jan 31 16:06 aws_cloudtrail_rules.yaml
-rw-r--r-- 1 root root  11384 Jan 31 16:06 falco.yaml
-rw-r--r-- 1 root root   1136 Jan 31 16:06 falco_rules.local.yaml
-rw-r--r-- 1 root root 132112 Jan 31 16:06 falco_rules.yaml
-rw-r--r-- 1 root root  27289 Jan 31 16:06 k8s_audit_rules.yaml
drwxr-xr-x 2 root root   4096 Feb 16 01:07 rules.available
drwxr-xr-x 2 root root   4096 Jan 31 16:28 rules.d
root@ksrs00101-worker1:~# vim /etc/falco/falco_rules.local.yaml
```



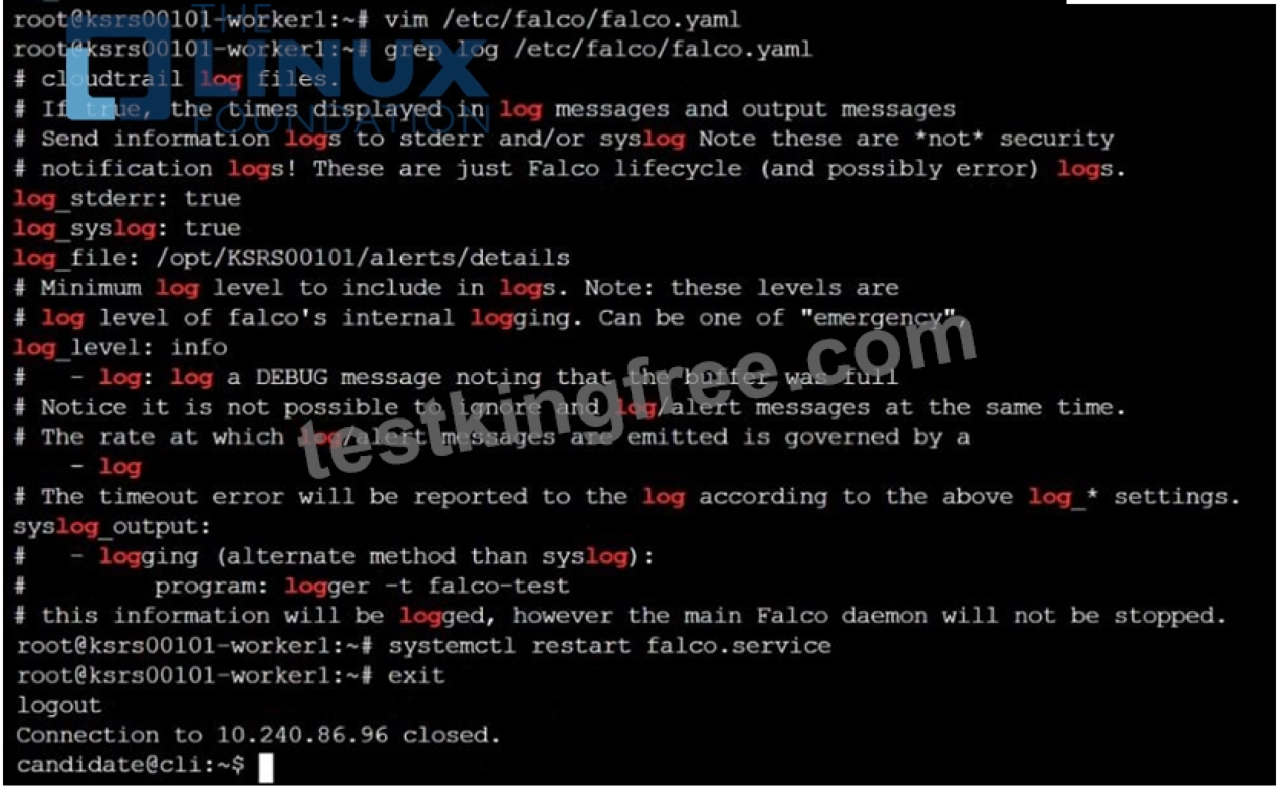
```
# Copyright (C) 2019 The Falco Authors.
#
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

####################
# Your custom rules!
####################

# Add new rules, like this one
# - rule: The program "sudo" is run in a container
#   desc: An event will trigger every time you run sudo in a container
#   condition: evt.type = execve and evt.dir=< and container.id != host and proc.name = sudo
#   output: "Sudo run in container (user=%user.name %container.info parent=%proc.pname cmdli
ne=%proc.cmdline)"
#   priority: ERROR
#   tags: [users, container]

# Or override/append to any rule, macro, or list from the Default Rules
- rule: Container Drift Detected (chmod)
  desc: New executable created in a container due to chmod
  condition: >
    evt.type in (open,openat,create) and
    evt.is_open_exec=true and
    container and
    not runc_writing_exec_fifo and
    not runc_writing_var_lib_docker and
    not user_known_container_drift_activities and
    evt.rawres>=0
  output:
     %evt.time,%user.uid,%proc.name
  priority: ERROR
```



```
root@ksrs00101-worker1:~# vim /etc/falco/falco_rules.local.yaml
root@ksrs00101-worker1:~# systemctl status falco.service
● falco.service - Falco Runtime Security
     Loaded: loaded (/lib/systemd/system/falco.service; disabled; vendor preset: enabled)
     Active: inactive (dead)
root@ksrs00101-worker1:~# systemctl enable falco.service
Created symlink /etc/systemd/system/multi-user.target.wants/falco.service → /lib/systemd/sys
tem/falco.service.
root@ksrs00101-worker1:~# systemctl start falco.service
root@ksrs00101-worker1:~# exit
logout
Connection to 10.240.86.96 closed.
candidate@cli:~$ ssh ksrs00101-worker1
Last login: Fri May 20 15:59:48 2022 from 10.240.86.88
root@ksrs00101-worker1:~# vim /etc/falco/falco.yaml
```



```
# When using json output, whether or not to include the "tags" property
# itself in the json output. If set to true, outputs caused by rules
# with no tags will have a "tags" field set to an empty array. If set to
# false, the "tags" field will not be included in the json output at all.
json_include_tags_property: true

# Send information logs to stderr and/or syslog Note these are *not* security
# notification logs! These are just Falco lifecycle (and possibly error) logs.
log_stderr: true
log_syslog: true
log_file: /opt/KSRS00101/alerts/details

# Minimum log level to include in logs. Note: these levels are
# separate from the priority field of rules. This refers only to the
# log level of falco's internal logging. Can be one of "emergency",
# "alert", "critical", "error", "warning", "notice", "info", "debug".
log_level: info
```



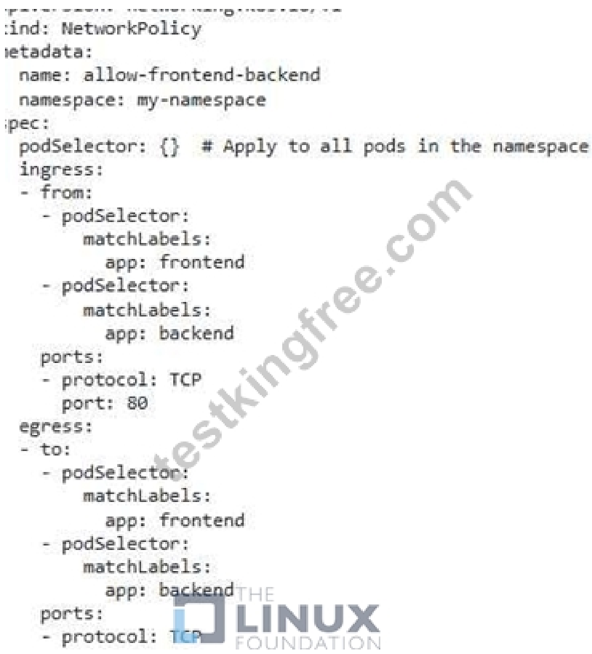
```
root@ksrs00101-worker1:~# vim /etc/falco/falco.yaml
root@ksrs00101-worker1:~# grep log /etc/falco/falco.yaml
# cloudtrail log files.
# If true, the times displayed in log messages and output messages
# Send information logs to stderr and/or syslog Note these are *not* security
# notification logs! These are just Falco lifecycle (and possibly error) logs.
log_stderr: true
log_syslog: true
log_file: /opt/KSRS00101/alerts/details
# Minimum log level to include in logs. Note: these levels are
# log level of falco's internal logging. Can be one of "emergency",
log_level: info
#   - log: log a DEBUG message noting that the buffer was full
# Notice it is not possible to ignore and log/alert messages at the same time.
# The rate at which log/alert messages are emitted is governed by a
    - log
# The timeout error will be reported to the log according to the above log_* settings.
syslog_output:
#   - logging (alternate method than syslog):
#         program: logger -t falco-test
# this information will be logged, however the main Falco daemon will not be stopped.
root@ksrs00101-worker1:~# systemctl restart falco.service
root@ksrs00101-worker1:~# exit
logout
Connection to 10.240.86.96 closed.
candidate@cli:~$
```

NEW QUESTION # 26

You need to implement a secure network policy that allows communication only between specific pods within a namespace. For example, you want to allow communication between pods that have the label 'app=frontend' and pods that have the label 'app=backend', but block all other communication within the namespace.

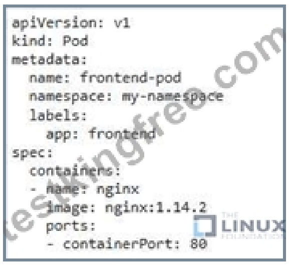
Answer:

Explanation:
Solution (Step by Step) :
1. Create a NetworkPolicy:
- Define a NetworkP01icy that allows communication between 'frontend' and 'backend' pods, but blocks other communication within the namespace.



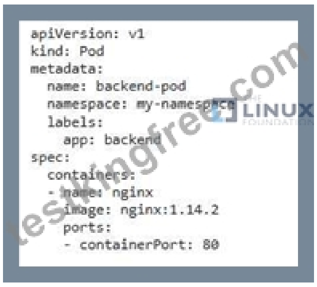
```
piVersion: networking.k8s.io/v1
ind: NetworkPolicy
etadata:
  name: allow-frontend-backend
  namespace: my-namespace
pec:
  podSelector: {}  # Apply to all pods in the namespace
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: frontend
    - podSelector:
        matchLabels:
          app: backend
    ports:
    - protocol: TCP
      port: 80
  egress:
  - to:
    - podSelector:
        matchLabels:
          app: frontend
    - podSelector:
        matchLabels:
          app: backend
    ports:
    - protocol: TCP
```

2. Create a Frontend Pod: - Create a Pod with the label 'app=frontend'.



```
apiVersion: v1
kind: Pod
metadata:
  name: frontend-pod
  namespace: my-namespace
  labels:
    app: frontend
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

3. Create a Backend Pod: - Create a Pod With the label 'app=backend'.



```
apiVersion: v1
kind: Pod
metadata:
  name: backend-pod
  namespace: my-namespace
  labels:
    app: backend
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

4. Apply the YAML files: - Apply the created YAML files using ' kubectl apply -f 5. Verify the Network Policy: - Try to connect from the 'frontend-pod' to the 'backend-pod' (e.g., using ' kubectl exec -it frontend-pod bash' and 'curl backend-pod:80')- It should succeed. - Try to connect from the 'frontend-pod' to another pod in the namespace that doesn't have the Sapp-backend' label. This connection should be blocked.

NEW QUESTION # 27

You must complete this task on the following cluster/nodes:
Cluster: trace
Master node: master
Worker node: worker1
You can switch the cluster/configuration context using the following command:
[desk@cli] $ kubectl config use-context trace
Given: You may use Sysdig or Falco documentation.
Task:
Use detection tools to detect anomalies like processes spawning and executing something weird frequently in the single container belonging to Pod tomcat.
Two tools are available to use:
1. falco
2. sysdig
Tools are pre-installed on the worker1 node only.

Analyse the container's behaviour for at least 40 seconds, using filters that detect newly spawning and executing processes.
Store an incident file at /home/cert_masters/report, in the following format:
[timestamp],[uid],[processName]
Note: Make sure to store incident file on the cluster's worker node, don't move it to master node.

Answer:

Explanation:

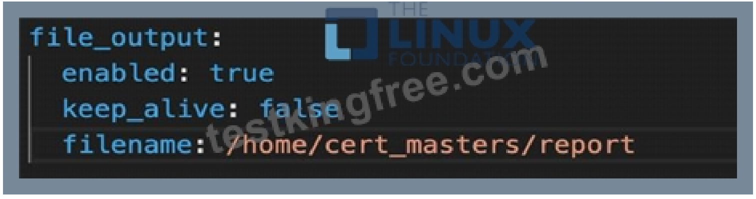
```
$vim /etc/falco/falco_rules.local.yaml
- rule: Container Drift Detected (open+create)
desc: New executable created in a container due to open+create
condition: >
evt.type in (open,openat,creat) and
evt.is_open_exec=true and
container and
not runc_writing_exec_fifo and
not runc_writing_var_lib_docker and
not user_known_container_drift_activities and
evt.rawres>=0
output: >
%evt.time,%user.uid,%proc.name # Add this/Refer falco documentation
priority: ERROR
$kill -1 <PID of falco>
```

Explanation

```
[desk@cli] $ ssh node01
[node01@cli] $ vim /etc/falco/falco_rules.yaml
```

search for Container Drift Detected & paste in falco_rules.local.yaml

```
[node01@cli] $ vim /etc/falco/falco_rules.local.yaml
- rule: Container Drift Detected (open+create)
desc: New executable created in a container due to open+create
condition: >
evt.type in (open,openat,creat) and
evt.is_open_exec=true and
container and
not runc_writing_exec_fifo and
not runc_writing_var_lib_docker and
not user_known_container_drift_activities and
evt.rawres>=0
output: >
%evt.time,%user.uid,%proc.name # Add this/Refer falco documentation
priority: ERROR
[node01@cli] $ vim /etc/falco/falco.yaml
```



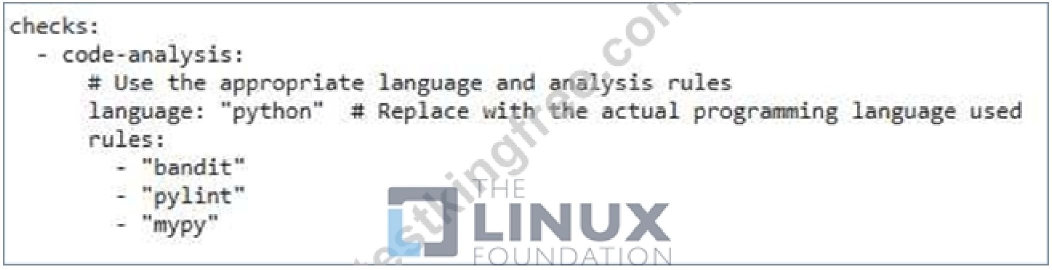
```
file_output:
  enabled: true
  keep_alive: false
  filename: /home/cert_masters/report
```

NEW QUESTION # 28
You are working on a Kubernetes cluster and need to analyze the security posture of a user workload running within a container image. The image is built from a Dockeffile tnat pulls code from a public GitHub repository. You need to identify potential security vulnerabilities in the codebase using a static analysis tool.

Answer:

Explanation:
Solution (Step by Step) :
1. Identify the Code Repository:
- Access the public GitHub repository where the source code for the user workload resides.

2. Install KubeLinter:
- Use 'pip install kube-linter' to install KubeLinter on your machine.
3. Configure KubeLinter:
- Create a configuration file for KubeLinter (e.g., 'kube-linter.yaml') with the following content:



```
checks:
  - code-analysis:
      # Use the appropriate language and analysis rules
      language: "python"  # Replace with the actual programming language used
      rules:
        - "bandit"
        - "pylint"
        - "mypy"
```

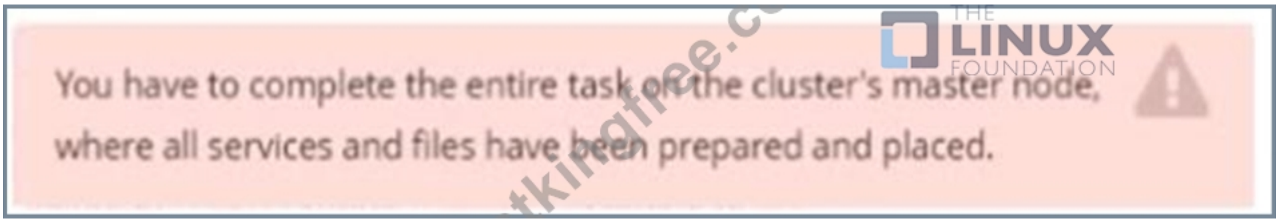
4. Run KubeLinter. - Execute the following command to analyze the source code: bash kube-linter --config kube-linter.yaml --path - Replace '/path/to/your/repository/' with the actual path to the GitHub repository's codebase. 5. Analyze the Results: - KubeLinter will output a report highlighting potential security vulnerabilities, coding best practices violations, and other issues detected in the codebase. - Review the findings carefully and prioritize remediation actions based on the severity and impact of the vulnerabilities.

NEW QUESTION # 29

Context

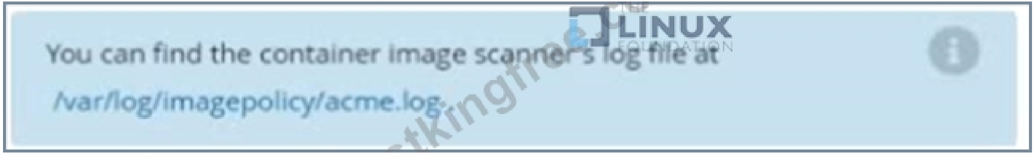
A container image scanner is set up on the cluster, but it's not yet fully integrated into the cluster s configuration. When complete, the container image scanner shall scan for and reject the use of vulnerable images.

Task



You have to complete the entire task on the cluster's master node, where all services and files have been prepared and placed.

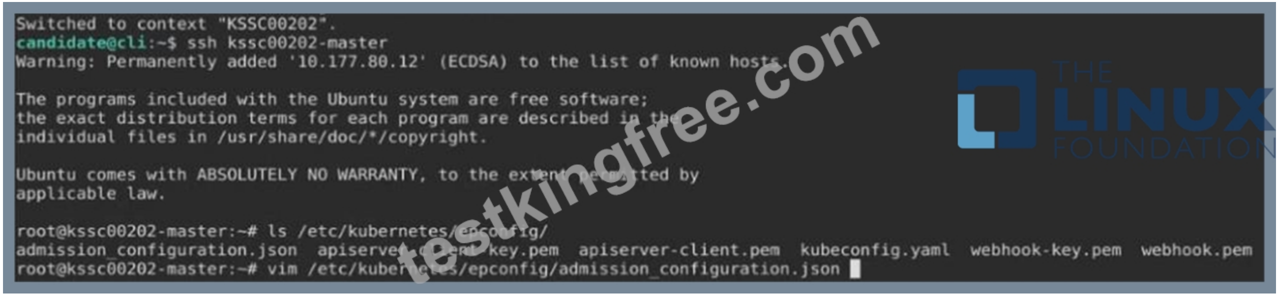
Given an incomplete configuration in directory /etc/kubernetes/epconfig and a functional container image scanner with HTTPS endpoint https://wakanda.local:8081 /image_policy :
1. Enable the necessary plugins to create an image policy
2. Validate the control configuration and change it to an implicit deny
3. Edit the configuration to point to the provided HTTPS endpoint correctly Finally, test if the configuration is working by trying to deploy the vulnerable resource /root/KSSC00202/vulnerable-resource.yml.



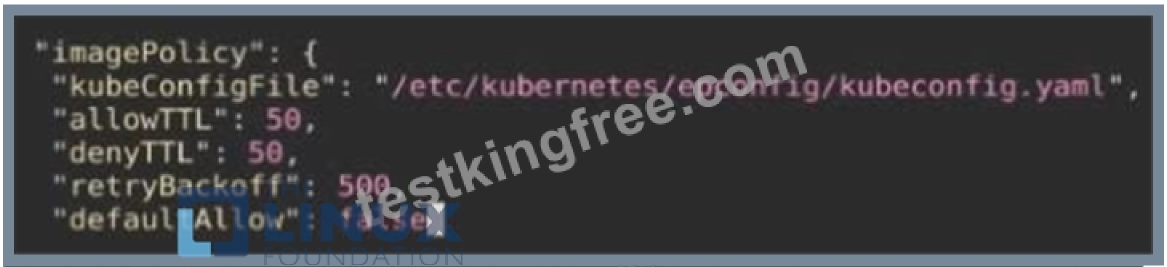
You can find the container image scanner's log file at /var/log/imagepolicy/acme.log.

Answer:

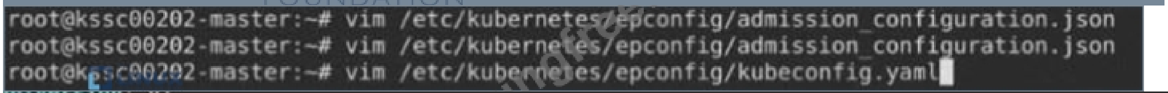
Explanation:



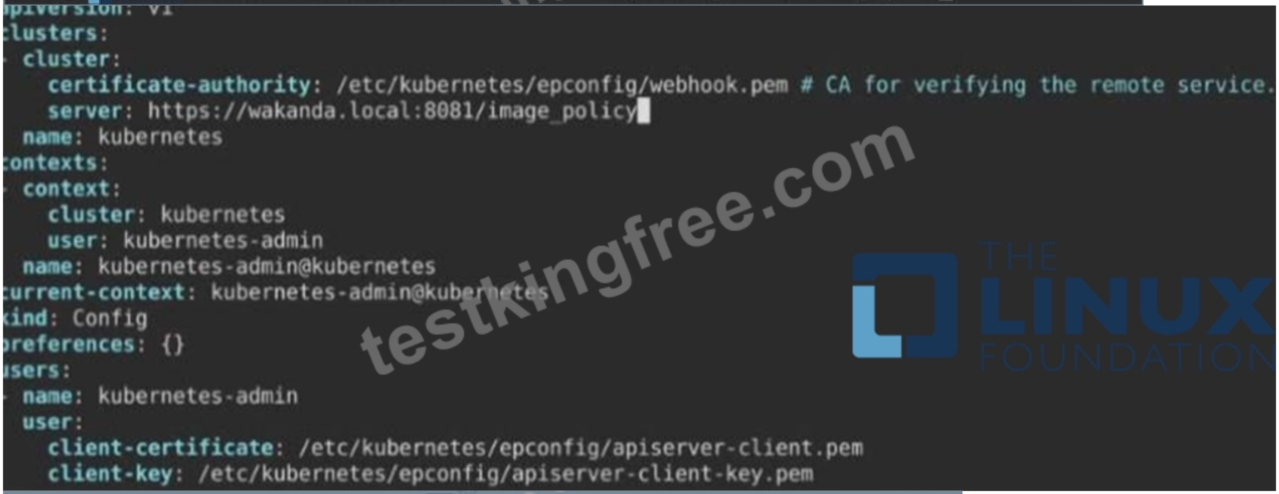
```
Switched to context "KSSC00202".
candidate@cli:~$ ssh kssc00202-master
Warning: Permanently added '10.177.80.12' (ECDSA) to the list of known hosts.

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@kssc00202-master:~# ls /etc/kubernetes/epconfig/
admission_configuration.json  apiserver-client-key.pem  apiserver-client.pem  kubeconfig.yaml  webhook-key.pem  webhook.pem
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/admission_configuration.json
```



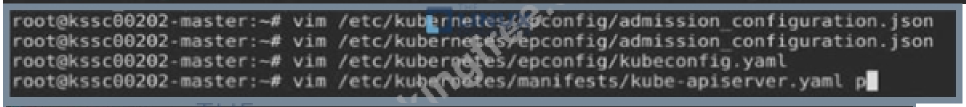
```
"imagePolicy": {
  "kubeConfigFile": "/etc/kubernetes/epconfig/kubeconfig.yaml",
  "allowTTL": 50,
  "denyTTL": 50,
  "retryBackoff": 500,
  "defaultAllow": false
```



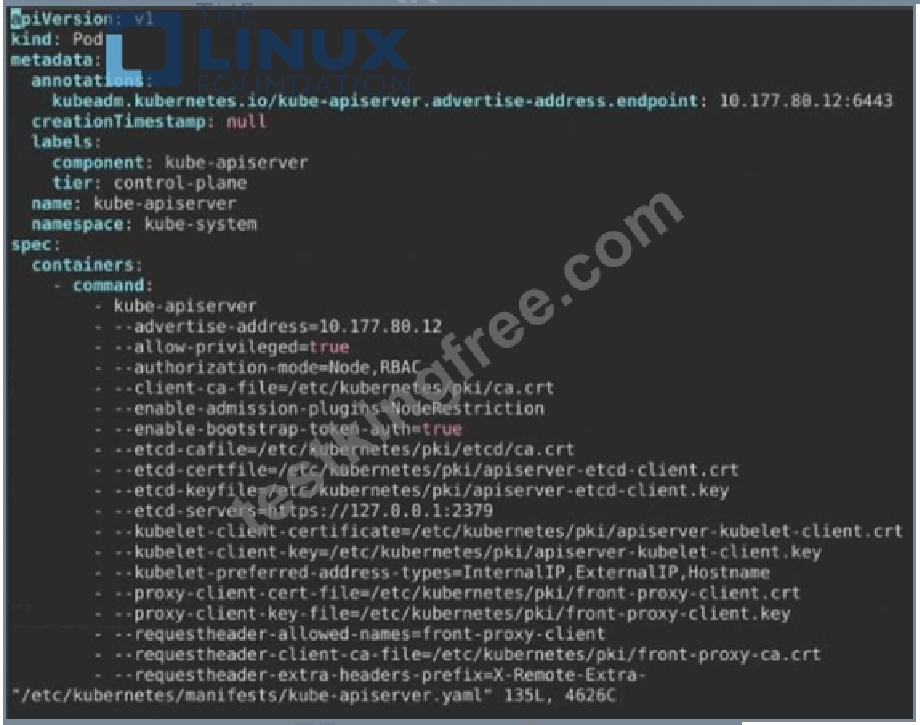
```
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/admission_configuration.json
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/admission_configuration.json
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/kubeconfig.yaml
```



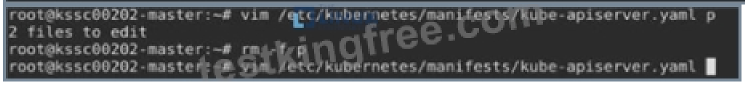
```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/epconfig/webhook.pem # CA for verifying the remote service.
    server: https://wakanda.local:8081/image_policy
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate: /etc/kubernetes/epconfig/apiserver-client.pem
    client-key: /etc/kubernetes/epconfig/apiserver-client-key.pem
```



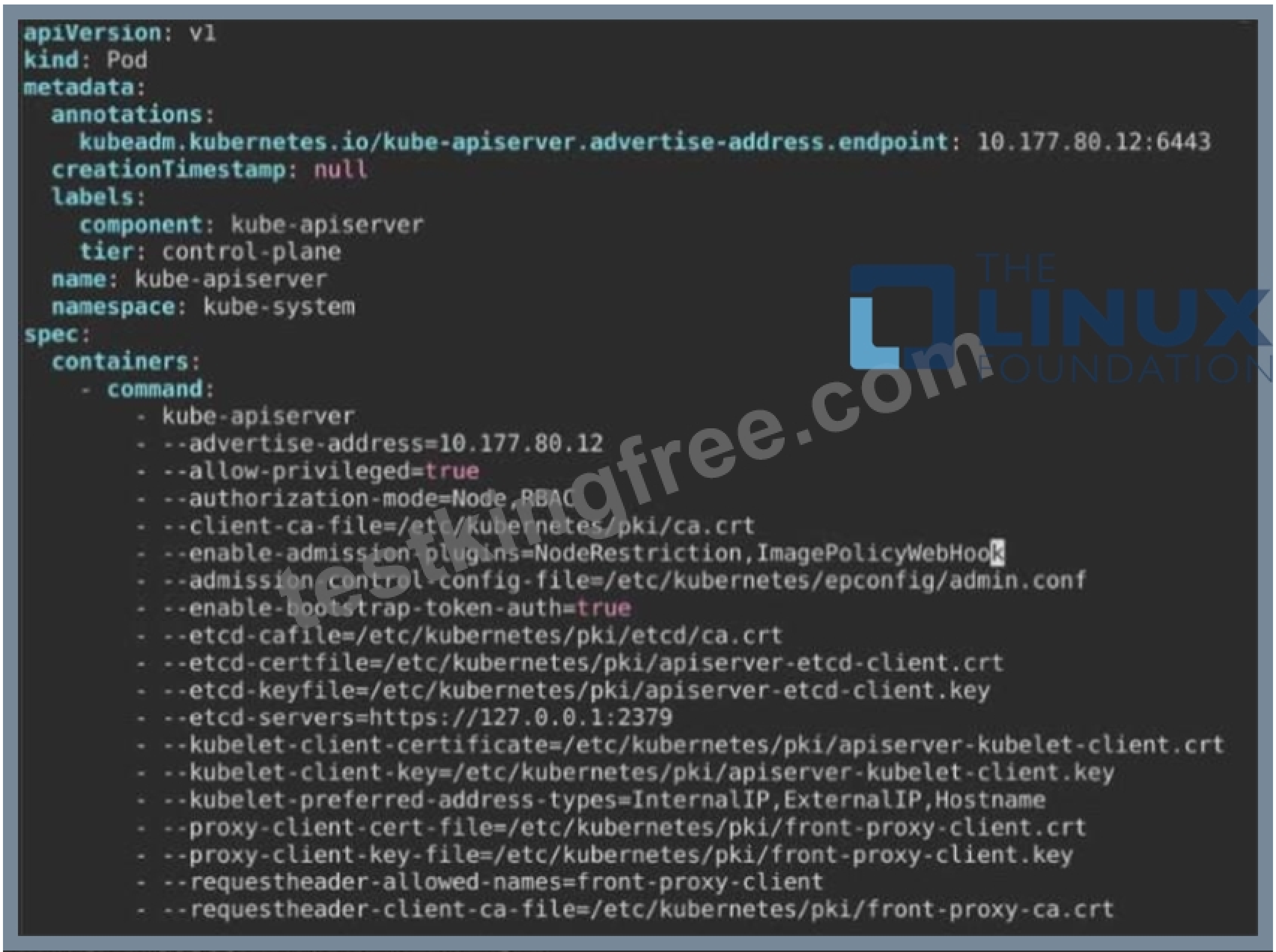
```
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/admission_configuration.json
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/admission_configuration.json
root@kssc00202-master:~# vim /etc/kubernetes/epconfig/kubeconfig.yaml
root@kssc00202-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml p
```



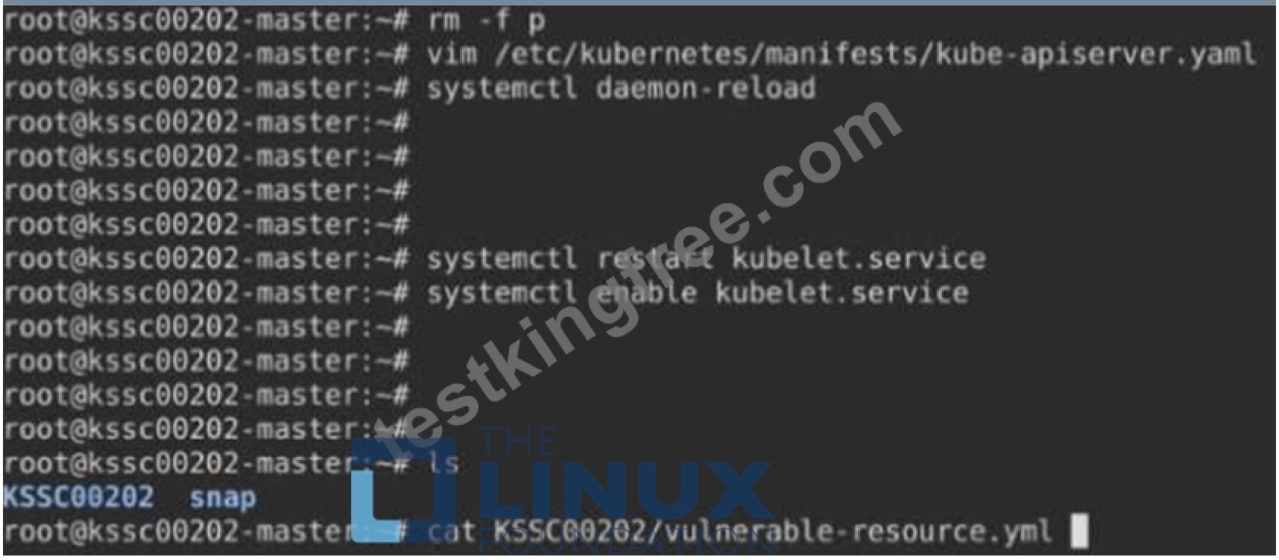
```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.177.80.12:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=10.177.80.12
    - --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-
"/etc/kubernetes/manifests/kube-apiserver.yaml" 135L, 4626C
```



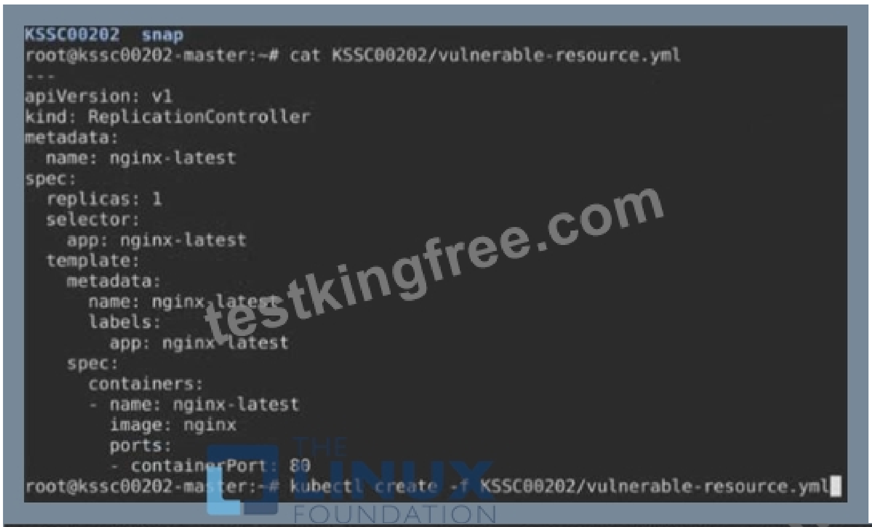
```
root@kssc00202-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml p
2 files to edit
root@kssc00202-master:~# rm -f p
root@kssc00202-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
```



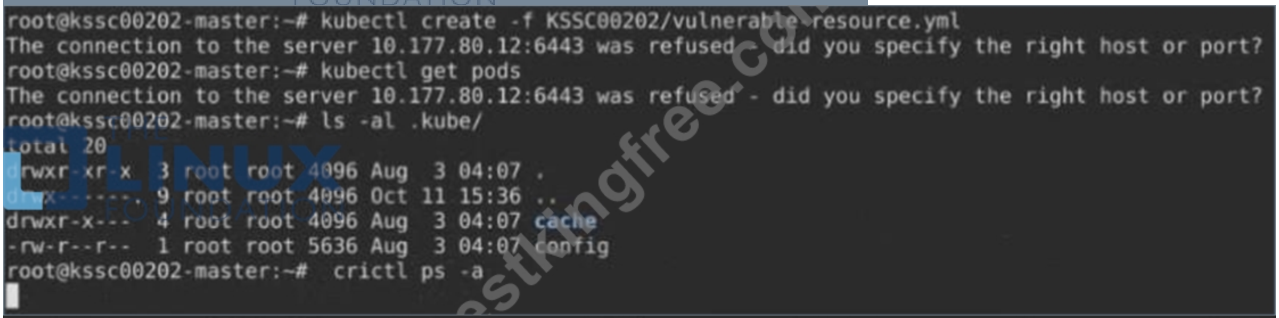
```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.177.80.12:6443
  creationTimestamp: null
  labels:
    component: kube-apiserver
    tier: control-plane
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=10.177.80.12
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-plugins=NodeRestriction,ImagePolicyWebHook
    - --admission-control-config-file=/etc/kubernetes/epconfig/admin.conf
    - --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
```



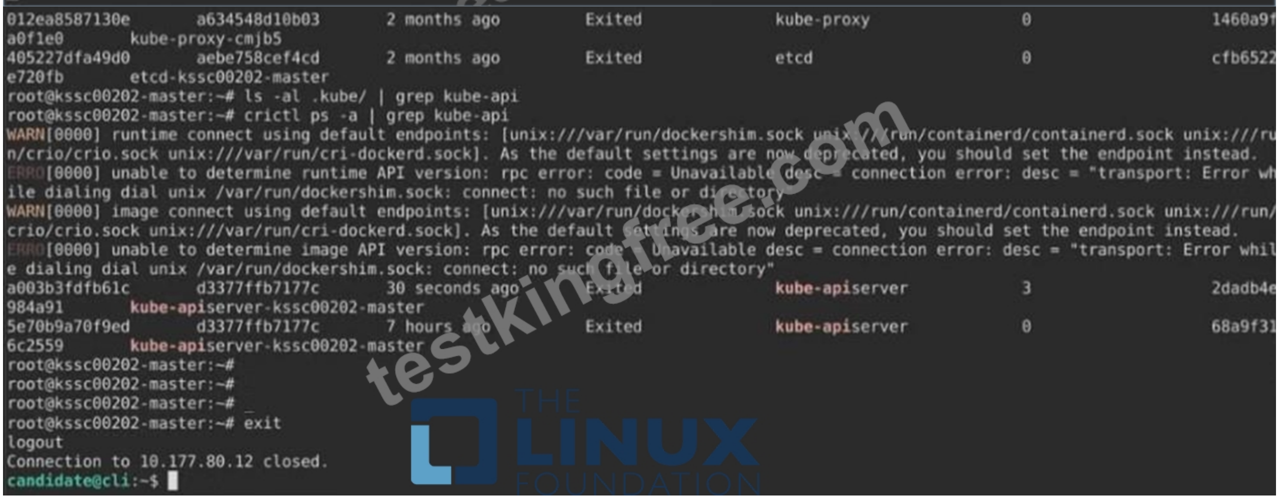
```
root@kssc00202-master:~# rm -f p
root@kssc00202-master:~# vim /etc/kubernetes/manifests/kube-apiserver.yaml
root@kssc00202-master:~# systemctl daemon-reload
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~# systemctl restart kubelet.service
root@kssc00202-master:~# systemctl enable kubelet.service
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~# ls
KSSC00202  snap
root@kssc00202-master:~# cat KSSC00202/vulnerable-resource.yml
```



```
KSSC00202  snap
root@kssc00202-master:~# cat KSSC00202/vulnerable-resource.yml
---
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-latest
spec:
  replicas: 1
  selector:
    app: nginx-latest
  template:
    metadata:
      name: nginx-latest
      labels:
        app: nginx-latest
    spec:
      containers:
      - name: nginx-latest
        image: nginx
        ports:
        - containerPort: 80
root@kssc00202-master:~# kubectl create -f KSSC00202/vulnerable-resource.yml
```



```
root@kssc00202-master:~# kubectl create -f KSSC00202/vulnerable-resource.yml
The connection to the server 10.177.80.12:6443 was refused - did you specify the right host or port?
root@kssc00202-master:~# kubectl get pods
The connection to the server 10.177.80.12:6443 was refused - did you specify the right host or port?
root@kssc00202-master:~# ls -al .kube/
total 20
drwxr-xr-x  3 root root 4096 Aug  3 04:07 .
drwx------. 9 root root 4096 Oct 11 15:36 ..
drwxr-x---  4 root root 4096 Aug  3 04:07 cache
-rw-r--r--  1 root root 5636 Aug  3 04:07 config
root@kssc00202-master:~#  crictl ps -a
█
```



```
012ea8587130e       a634548d10b03       2 months ago        Exited              kube-proxy              0                   1460a9f
a0f1e0       kube-proxy-cmjb5
405227dfa49d0       aebe758cef4cd       2 months ago        Exited              etcd                    0                   cfb6522
e720fb       etcd-kssc00202-master
root@kssc00202-master:~# ls -al .kube/ | grep kube-api
root@kssc00202-master:~# crictl ps -a | grep kube-api
WARN[0000] runtime connect using default endpoints: [unix:///var/run/dockershim.sock unix:///run/containerd/containerd.sock unix:///ru
n/crio/crio.sock unix:///var/run/cri-dockerd.sock]. As the default settings are now deprecated, you should set the endpoint instead.
ERRO[0000] unable to determine runtime API version: rpc error: code = Unavailable desc = connection error: desc = "transport: Error wh
ile dialing dial unix /var/run/dockershim.sock: connect: no such file or directory"
WARN[0000] image connect using default endpoints: [unix:///var/run/dockershim.sock unix:///run/containerd/containerd.sock unix:///run/
crio/crio.sock unix:///var/run/cri-dockerd.sock]. As the default settings are now deprecated, you should set the endpoint instead.
ERRO[0000] unable to determine image API version: rpc error: code = Unavailable desc = connection error: desc = "transport: Error whil
e dialing dial unix /var/run/dockershim.sock: connect: no such file or directory"
a003b3fdfb61c       d3377ffb7177c       30 seconds ago      Exited              kube-apiserver          3                   2dadb4e
984a91       kube-apiserver-kssc00202-master
5e70b9a70f9ed       d3377ffb7177c       7 hours ago         Exited              kube-apiserver          0                   68a9f31
6c2559       kube-apiserver-kssc00202-master
root@kssc00202-master:~#
root@kssc00202-master:~#
root@kssc00202-master:~# _
root@kssc00202-master:~# exit
logout
Connection to 10.177.80.12 closed.
candidate@cli:~$ █
```

NEW QUESTION # 30

......

TestKingFree provide you with the comprehensive Linux Foundation CKS Exam information to help you to succeed. Our training materials are the latest study materials which bring by experts. We help you achieve your success. You can get the most detailed and accurate exam questions and answers from us. Our Training Tools are updated in a timely manner in accordance with the changing of Exam Objectives. In fact, the success is not far away, go down along with TestKingFree, then you will come to the road to success.

Valid CKS Exam Dumps: https://www.testkingfree.com/Linux-Foundation/CKS-practice-exam-dumps.html

- New CKS Exam Pattern □ CKS Key Concepts □ New CKS Exam Pattern □ Search for 《 CKS 》 and easily obtain a free download on ➤ www.pdfvce.com □ □CKS Practice Exam
- Certification CKS Torrent □ CKS Practice Exam Questions ☂ CKS Exam Certification Cost □ Simply search for [CKS] for free download on ▶ www.examdiscuss.com ◀ □Valid Test CKS Bootcamp
- Quiz 2026 Linux Foundation CKS: Certified Kubernetes Security Specialist (CKS) Latest 100% Correct Answers □ Easily obtain free download of □ CKS □ by searching on 「 www.pdfvce.com 」 □CKS New Cram Materials
- 2026 The Best CKS: Certified Kubernetes Security Specialist (CKS) 100% Correct Answers □ Open website ☀ www.troytecdumps.com □☀□ and search for □ CKS □ for free download □CKS Exam Certification Cost
- Three Easy-to-Use Formats of Pdfvce Linux Foundation CKS Exam Questions □ Open ⇛ www.pdfvce.com ⇚ enter □ CKS □ and obtain a free download □CKS Valid Test Testking
- Online CKS Bootcamps □ Valid Dumps CKS Free □ CKS New Cram Materials □ Download 【 CKS 】 for free by simply searching on ✔ www.vceengine.com □✔□ □Reliable CKS Study Notes
- Certification CKS Torrent □ Valid CKS Exam Forum □ CKS Valid Test Question □ Download ➽ CKS □ for free by simply entering ▹ www.pdfvce.com ◃ website □CKS Valid Test Question
- 2026 The Best CKS: Certified Kubernetes Security Specialist (CKS) 100% Correct Answers □ Open □ www.examdiscuss.com □ enter ▹ CKS ◃ and obtain a free download □New CKS Exam Pattern
- ekadantha.in, bbs.t-firefly.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, bbs.t-firefly.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, bbs.chaken.net.cn, bbs.t-firefly.com, train2growup.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.thingstogetme.com, Disposable vapes

BONUS!!! Download part of TestKingFree CKS dumps for free: https://drive.google.com/open?id=1cAdIWWv0NxA4_9Zu8kvXwNgR-KI4uwFD