CKS$^{Q\&As}$

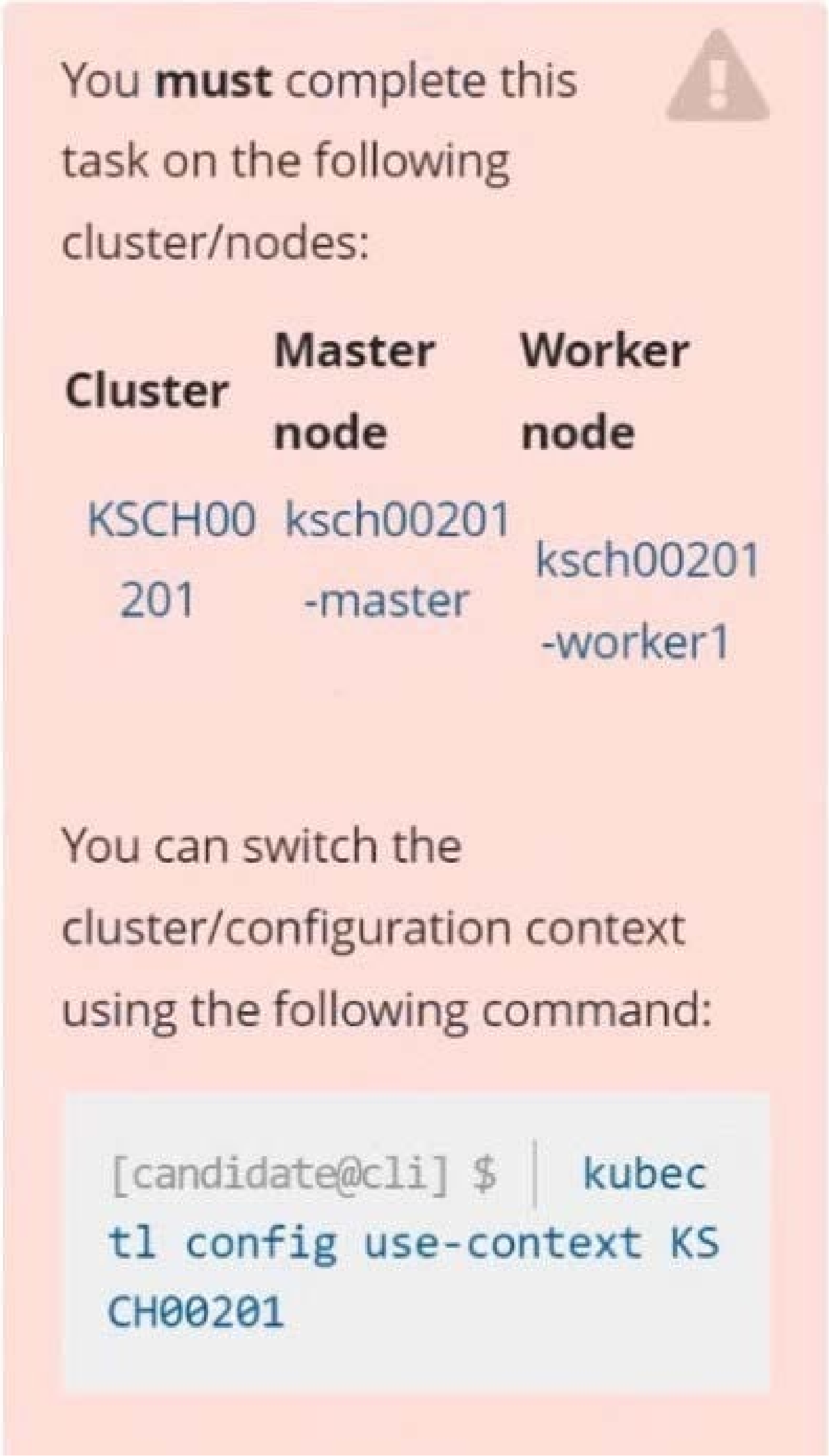
Certified Kubernetes Security Specialist (CKS) Exam

Pass Linux Foundation CKS Exam with 100% Guarantee

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

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



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

Cluster	Master node	Worker node
KSCH00201	ksch00201-master	ksch00201-worker1

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

```
[candidate@cli] $ kubectl config use-context KSCH00201
```

Task

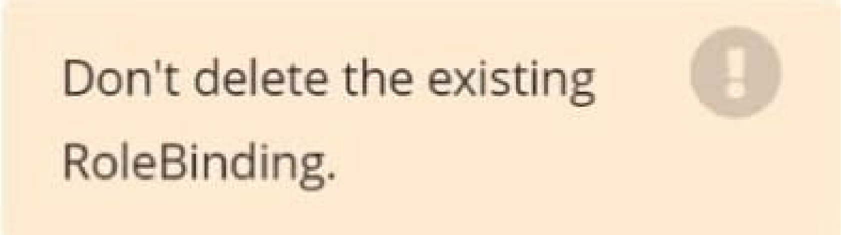
Given an existing Pod named web-pod running in the namespace security.

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.

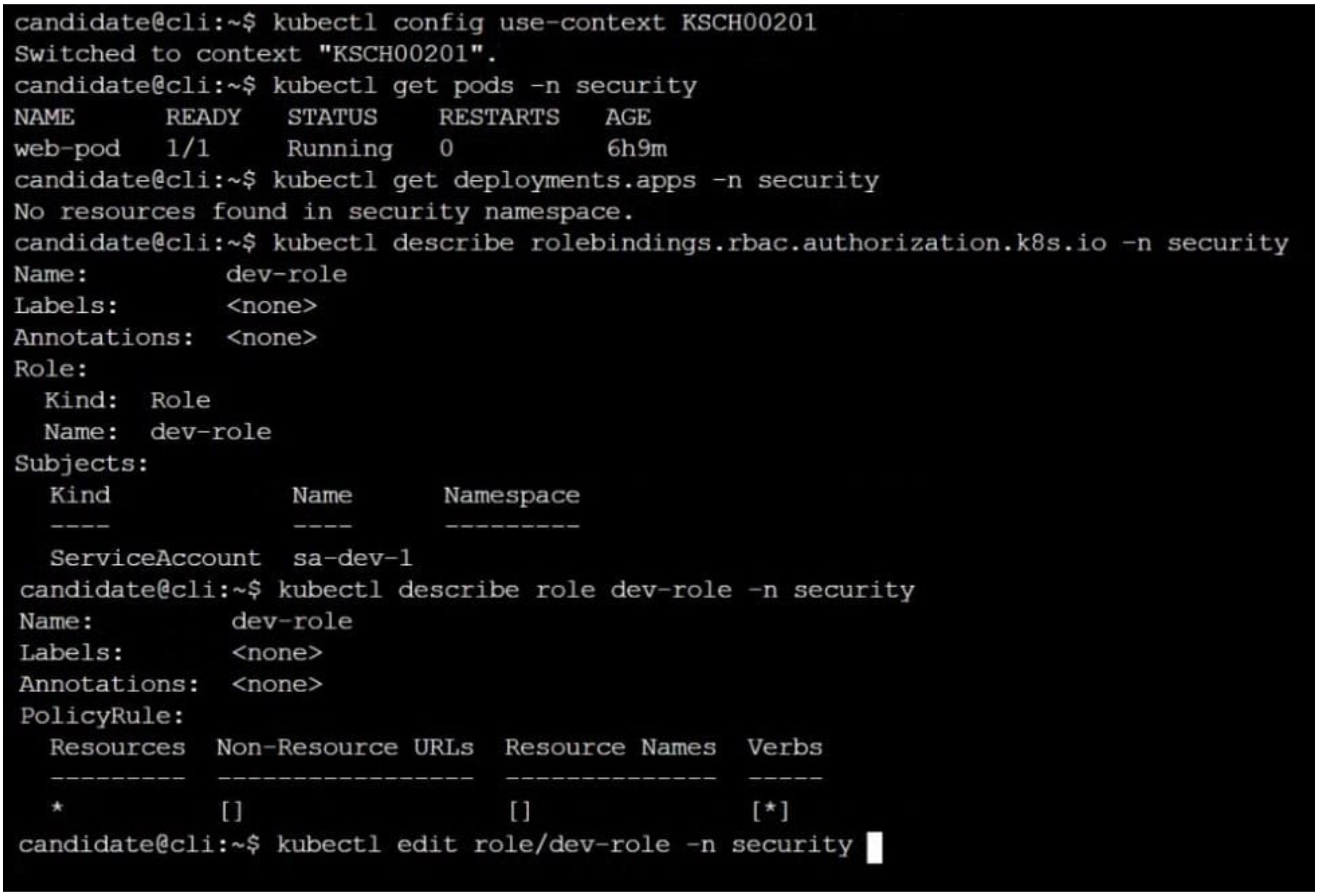


Don't delete the existing RoleBinding.

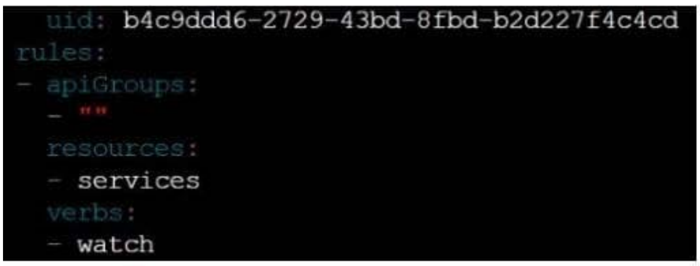
A. See the explanation below

B. PlaceHolder

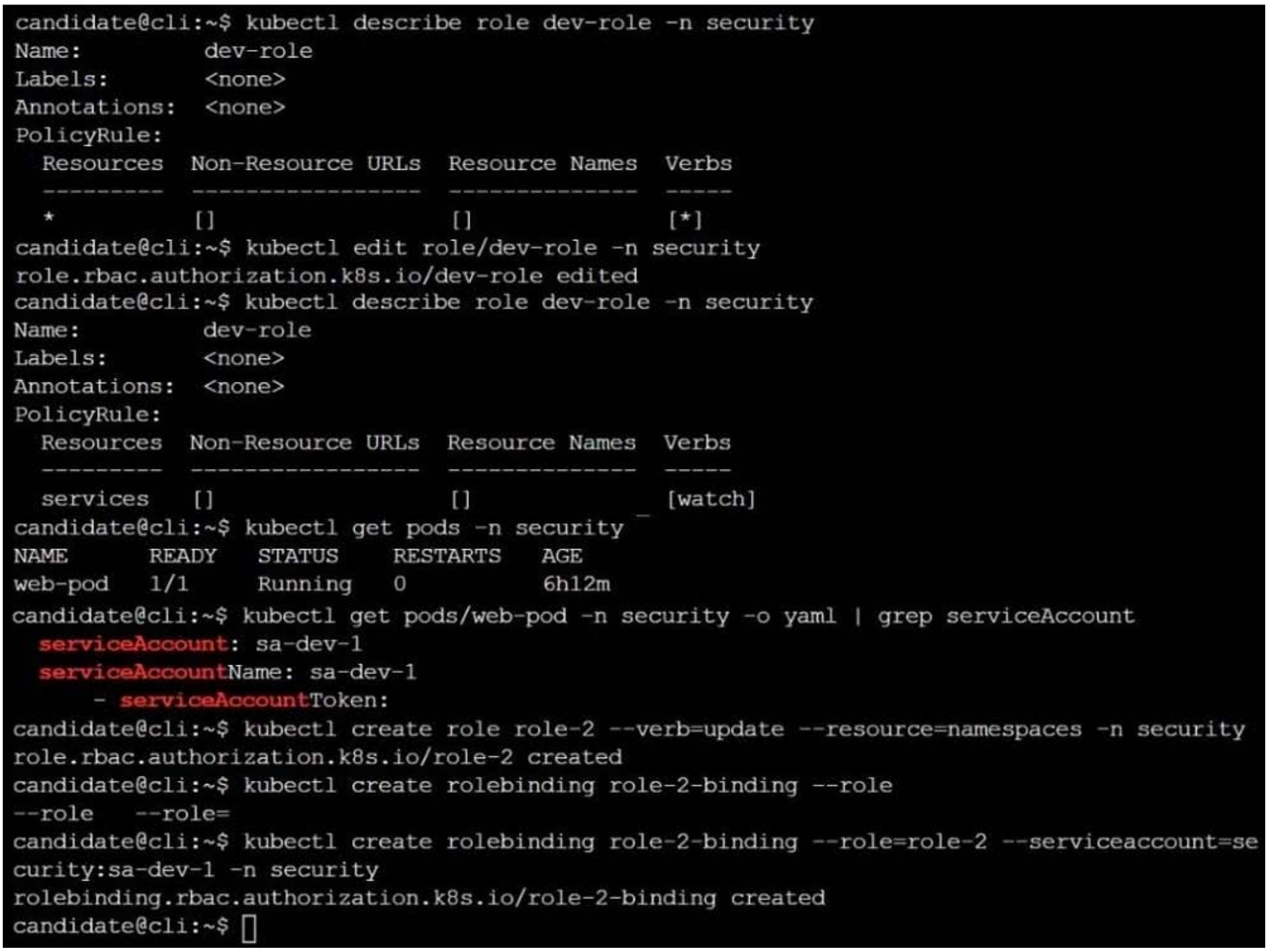
Correct Answer: A



```
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
```



```
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 2

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.

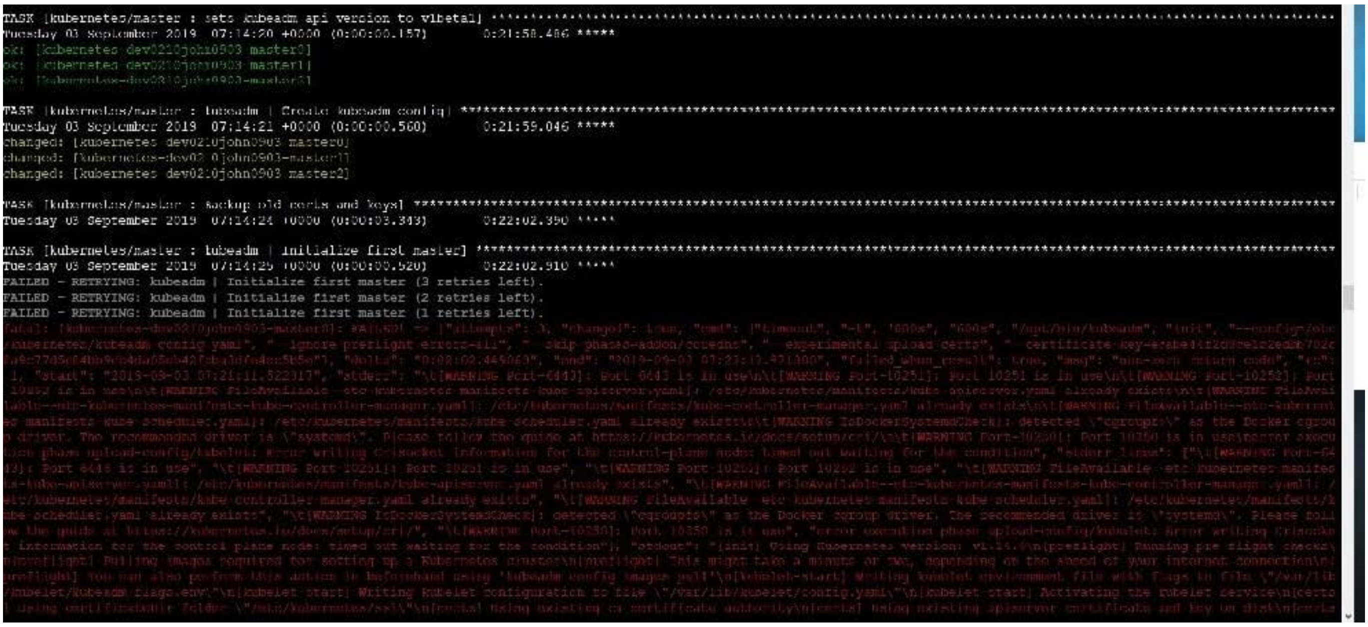
Finally, test the configuration by deploying the pod having the image tag as latest.

A. See explanation below.

B. PlaceHolder

Correct Answer: A

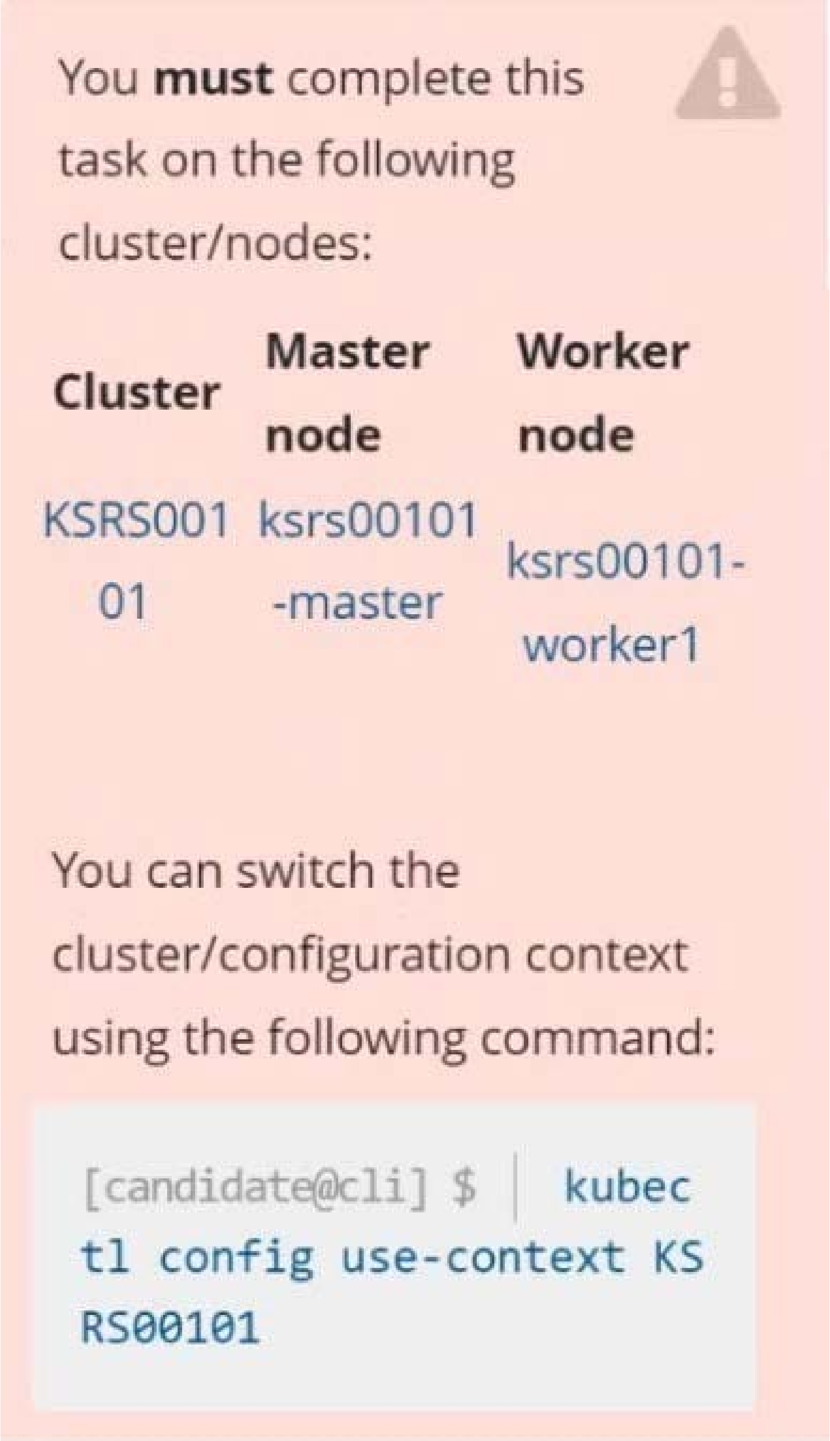
ssh-add ~/.ssh/tempprivate eval "$(ssh-agent -s)" cd contrib/terraform/aws vi terraform.tfvars terraform init terraform apply -var-file=credentials.tfvars ansible-playbook -i ./inventory/hosts ./cluster.yml -e ansible_ssh_user=core -e bootstrap_os=coreos -b --become-user=root --flush-cache -e ansible_user=core



```
TASK [kubernetes/master : sets kubeadm api version to v1beta1] *************************************************************************
Tuesday 03 September 2019  07:14:20 +0000 (0:00:00.157)       0:21:58.486 *****
ok: [kubernetes-dev0210john0903-master0]
ok: [kubernetes-dev0210john0903-master1]
ok: [kubernetes-dev0210john0903-master2]

TASK [kubernetes/master : kubeadm | Create kubeadm config] ****************************************************************************
Tuesday 03 September 2019  07:14:21 +0000 (0:00:00.560)       0:21:59.046 *****
changed: [kubernetes-dev0210john0903-master0]
changed: [kubernetes-dev0210john0903-master1]
changed: [kubernetes-dev0210john0903-master2]

TASK [kubernetes/master : Backup old certs and keys] **********************************************************************************
Tuesday 03 September 2019  07:14:24 +0000 (0:00:03.343)       0:22:02.390 *****

TASK [kubernetes/master : kubeadm | Initialize first master] **************************************************************************
Tuesday 03 September 2019  07:14:25 +0000 (0:00:00.520)       0:22:02.910 *****
FAILED - RETRYING: kubeadm | Initialize first master (3 retries left).
FAILED - RETRYING: kubeadm | Initialize first master (2 retries left).
FAILED - RETRYING: kubeadm | Initialize first master (1 retries left).
fatal: [kubernetes-dev0210john0903-master0]: FAILED! => {"attempts": 3, "changed": true, "cmd": ["timeout", "-k", "600s", "600s", "/opt/bin/kubeadm", "init", "--config=/etc
/kubernetes/kubeadm-config.yaml", "--ignore-preflight-errors=all", "--skip-phases=addon/coredns", "--experimental-upload-certs", "--certificate-key=[illegible]
[illegible]"], "delta": "0:02:02.449063", "end": "2019-09-03 07:23:13.971300", "failed_when_result": true, "msg": "non-zero return code", "rc":
 1, "start": "2019-09-03 07:21:11.522317", "stderr": "\t[WARNING Port-6443]: Port 6443 is in use\n\t[WARNING Port-10251]: Port 10251 is in use\n\t[WARNING Port-10252]: Port
 10252 is in use\n\t[WARNING FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists\n\t[WARNING FileAvai
lable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /etc/kubernetes/manifests/kube-controller-manager.yaml already exists\n\t[WARNING FileAvailable--etc-kubernet
es-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/kube-scheduler.yaml already exists\n\t[WARNING IsDockerSystemdCheck]: detected \"cgroupfs\" as the Docker cgrou
p driver. The recommended driver is \"systemd\". Please follow the guide at https://kubernetes.io/docs/setup/cri/\n\t[WARNING Port-10250]: Port 10250 is in use\nerror execu
tion phase upload-config/kubelet: Error writing Crisocket information for the control-plane node: timed out waiting for the condition", "stderr_lines": ["\t[WARNING Port-64
43]: Port 6443 is in use", "\t[WARNING Port-10251]: Port 10251 is in use", "\t[WARNING Port-10252]: Port 10252 is in use", "\t[WARNING FileAvailable--etc-kubernetes-manifes
ts-kube-apiserver.yaml]: /etc/kubernetes/manifests/kube-apiserver.yaml already exists", "\t[WARNING FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml]: /
etc/kubernetes/manifests/kube-controller-manager.yaml already exists", "\t[WARNING FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml]: /etc/kubernetes/manifests/k
ube-scheduler.yaml already exists", "\t[WARNING IsDockerSystemdCheck]: detected \"cgroupfs\" as the Docker cgroup driver. The recommended driver is \"systemd\". Please foll
ow the guide at https://kubernetes.io/docs/setup/cri/", "\t[WARNING Port-10250]: Port 10250 is in use", "error execution phase upload-config/kubelet: Error writing Crisocke
t information for the control-plane node: timed out waiting for the condition"], "stdout": "[init] Using Kubernetes version: v1.14.[illegible]\n[preflight] Running pre-flight checks\
n[preflight] Pulling images required for setting up a Kubernetes cluster\n[preflight] This might take a minute or two, depending on the speed of your internet connection\n[
preflight] You can also perform this action in beforehand using 'kubeadm config images pull'\n[kubelet-start] Writing kubelet environment file with flags to file \"/var/lib
/kubelet/kubeadm-flags.env\"\n[kubelet-start] Writing kubelet configuration to file \"/var/lib/kubelet/config.yaml\"\n[kubelet-start] Activating the kubelet service\n[certs
] Using certificateDir folder \"/etc/kubernetes/pki\"\n[certs] Using existing ca certificate authority\n[certs] Using existing apiserver certificate and key on disk\n[certs
```

QUESTION 3

CORRECT TEXT

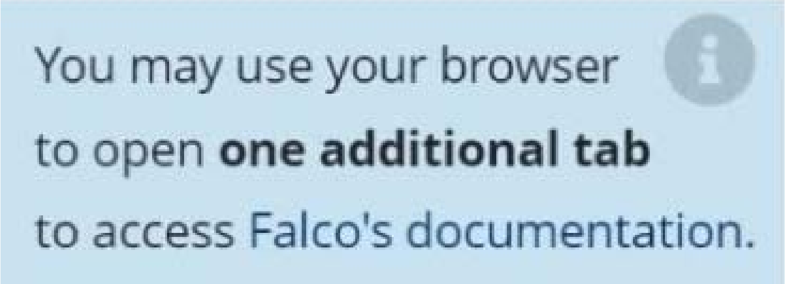


You **must** complete this task on the following cluster/nodes:

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:

1.

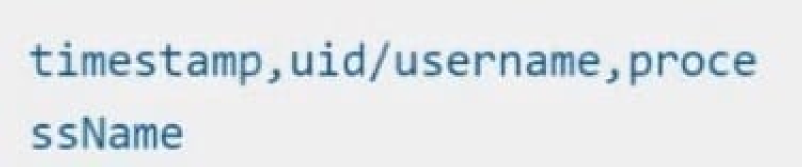
sysdig

2.

falco

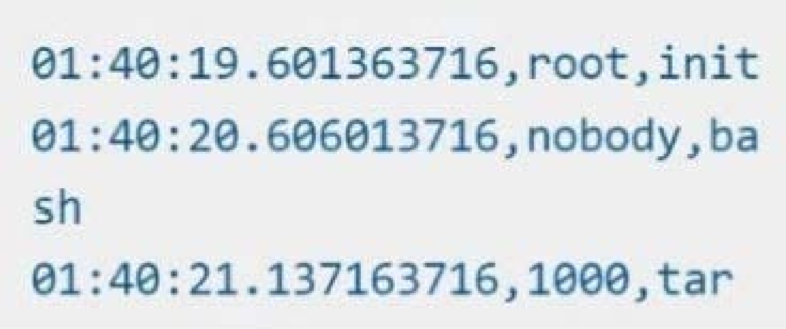
Using the tool of your choice (including any non pre-installed tool), analyze the container\\'s behavior for at least 30 seconds, using filters that detect newly spawning and executing processes. Store an incident file at /opt/KSRS00101/alerts/

details, containing the detected incidents, one per line, in the following format:

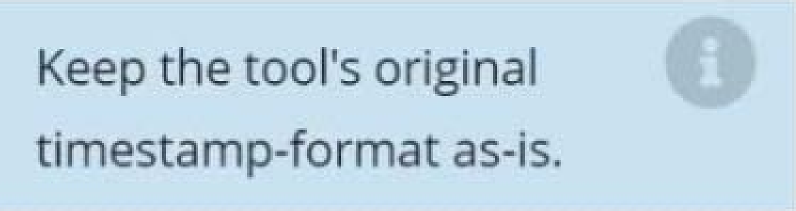


```
timestamp,uid/username,processName
```

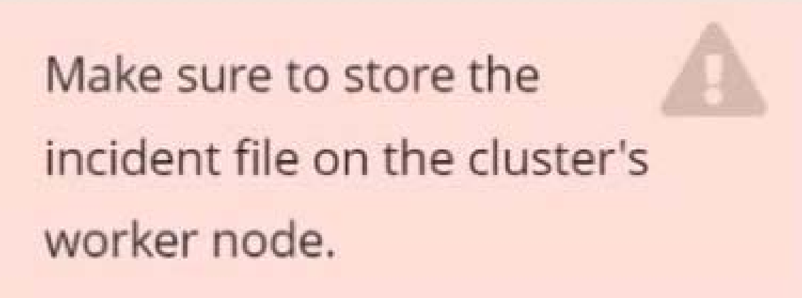
The following example shows a properly formatted incident file:



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



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

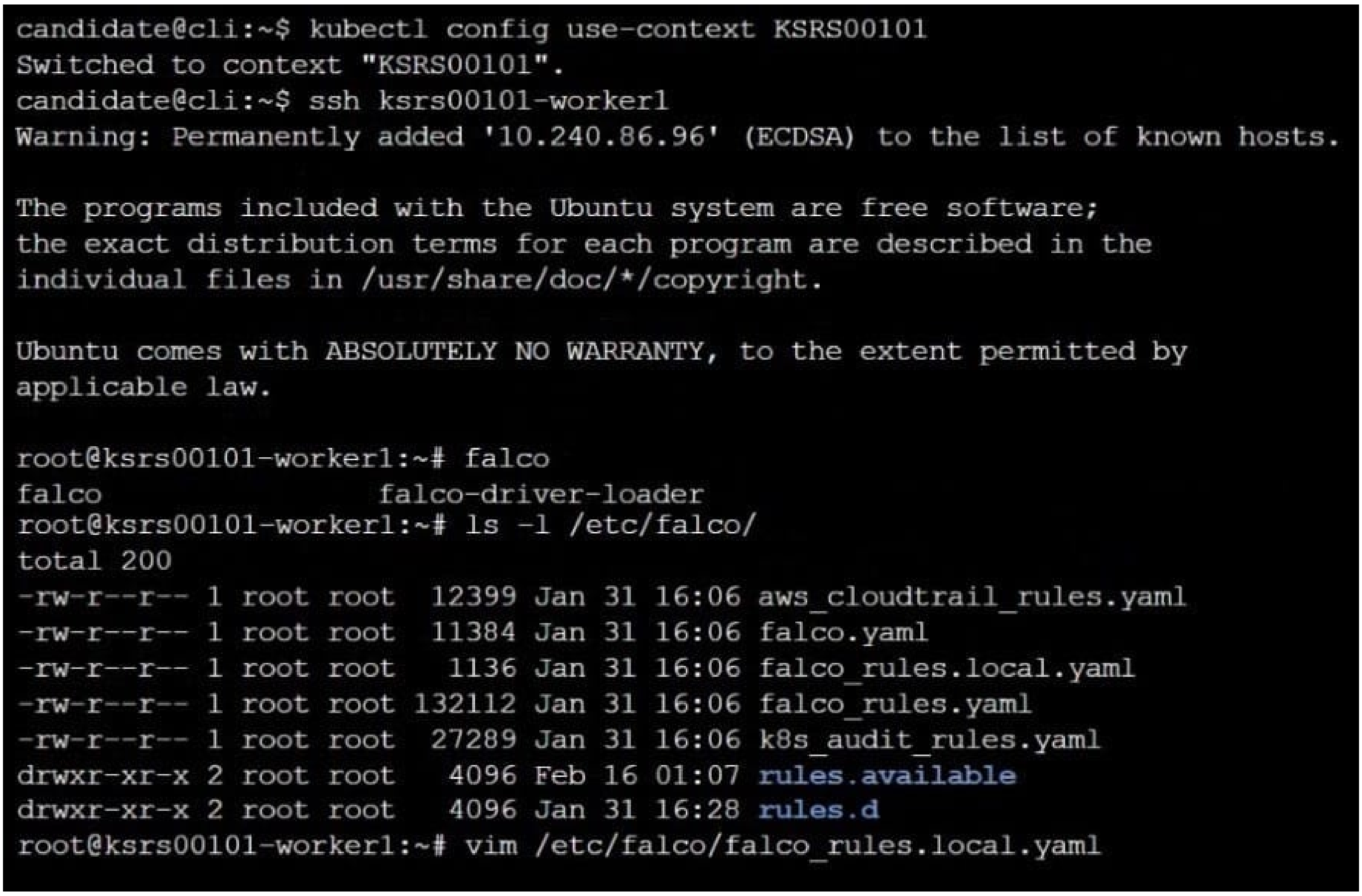


Make sure to store the incident file on the cluster's worker node.

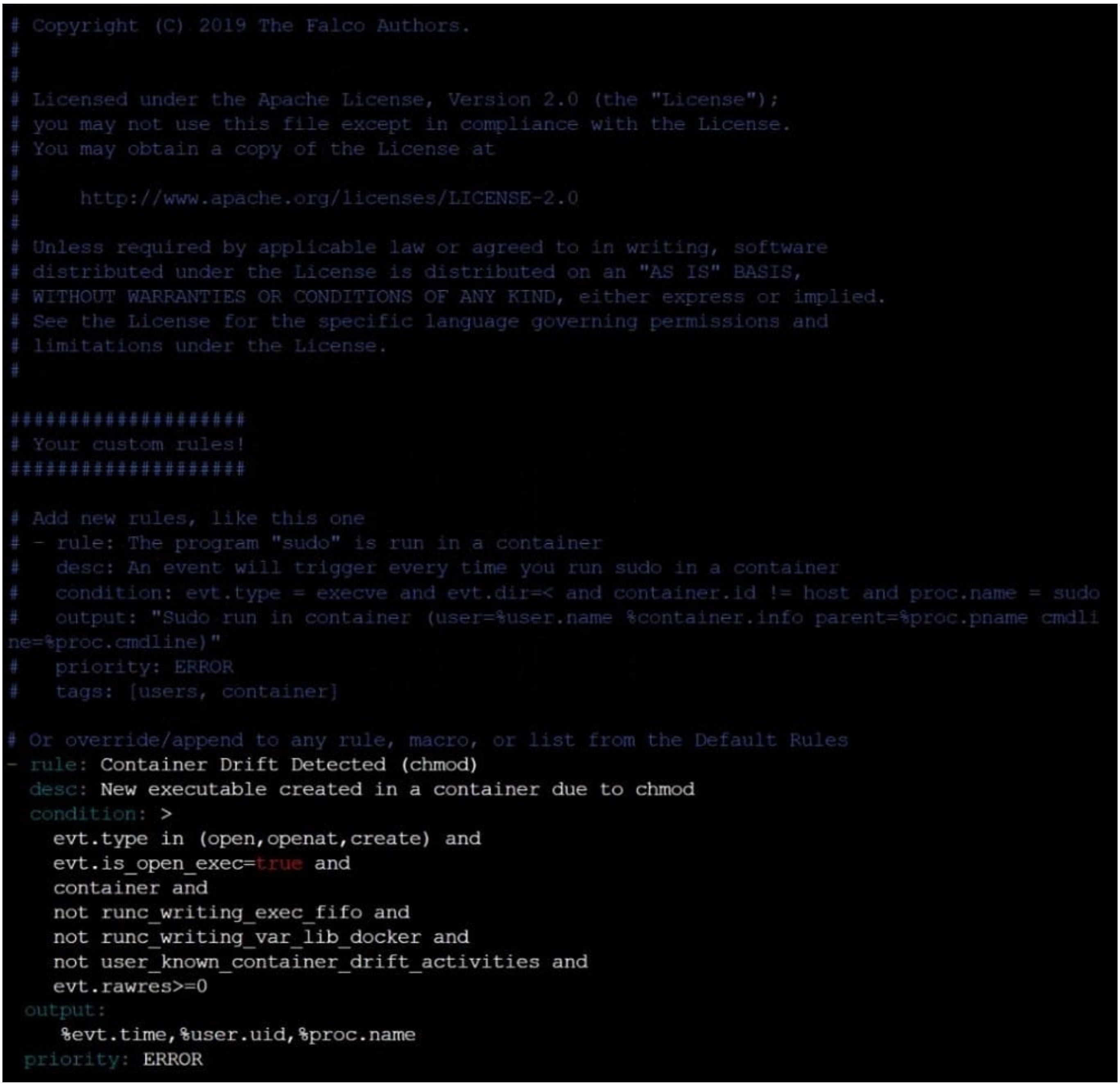
A. See the explanation below:

B. PlaceHolder

Correct Answer: A

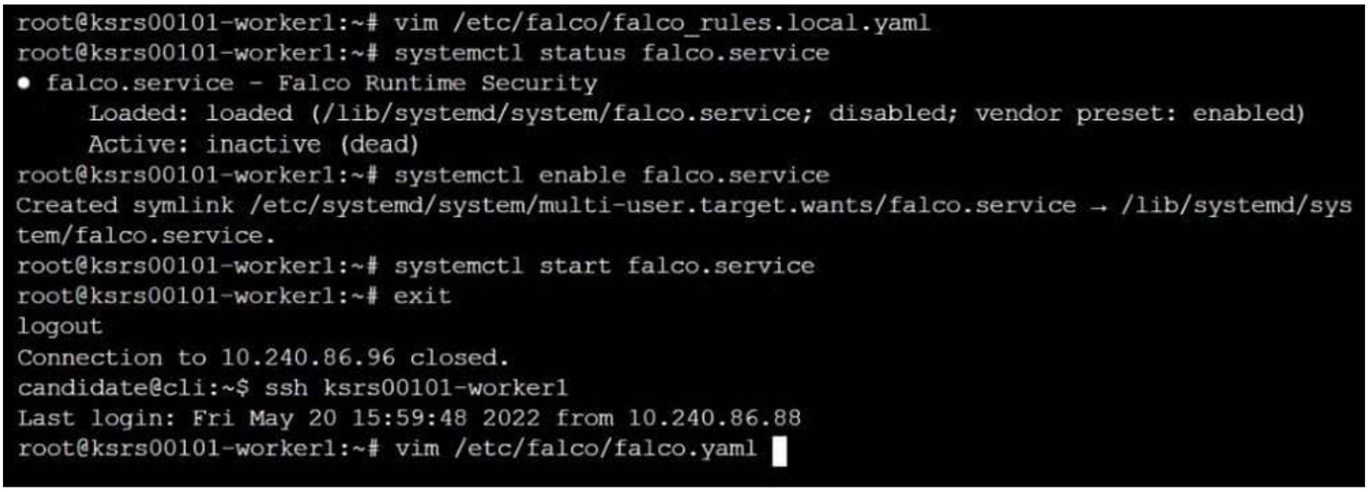


```
candidate@cli:~$ kubectl config use-context KSRS00101
Switched to context "KSRS00101".
candidate@cli:~$ ssh ksrs00101-worker1
Warning: Permanently added '10.240.86.96' (ECDSA) to the list of known hosts.

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
```

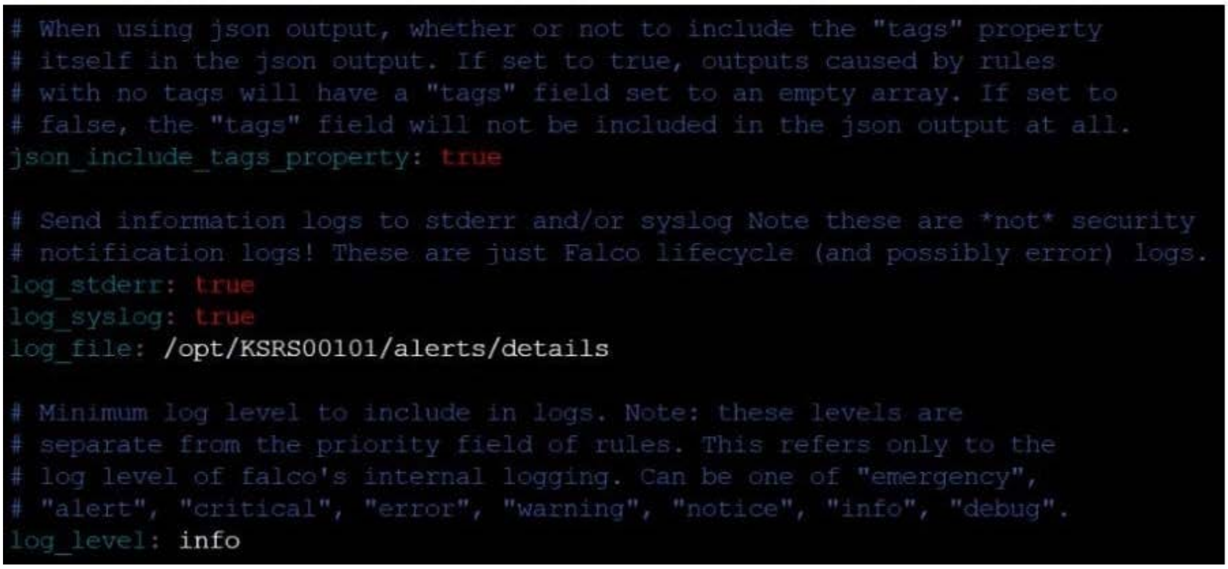


```
# Copyright (C) 2019 The Falco Authors.
#
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable 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
```

Text

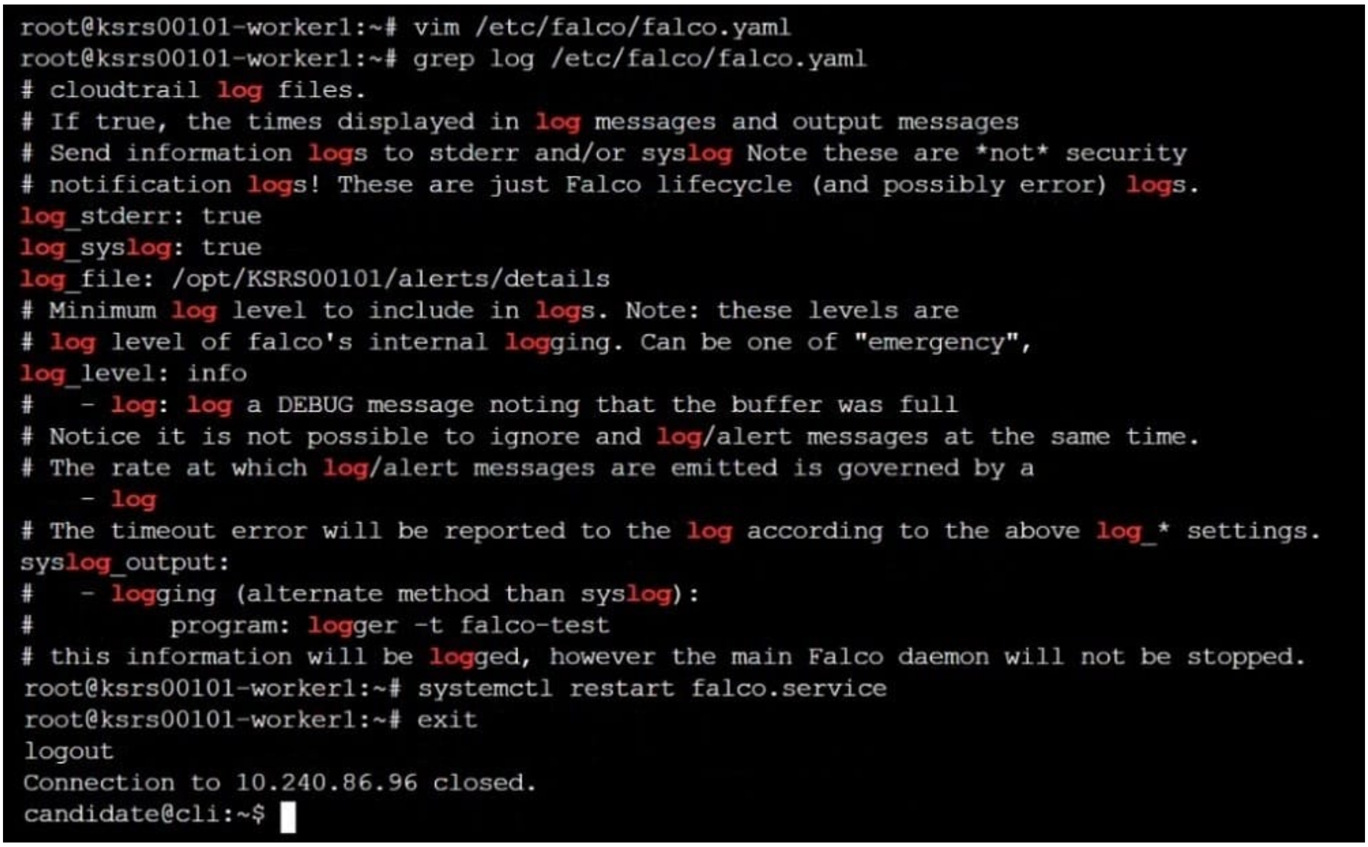


```
root@ksrs00101-worker1:~# vim /etc/falco/falco_rules.local.yaml
root@ksrs00101-worker1:~# systemctl status falco.service
● falco.service - Falco Runtime Security
     Loaded: loaded (/lib/systemd/system/falco.service; disabled; vendor preset: enabled)
     Active: inactive (dead)
root@ksrs00101-worker1:~# 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
```

Text



```
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:~$ 
```

QUESTION 4

Create a new ServiceAccount named backend-sa in the existing namespace default, which has the capability to list the pods inside the namespace default.

Create a new Pod named backend-pod in the namespace default, mount the newly created sa backend-sa to the pod, and Verify that the pod is able to list pods.

Ensure that the Pod is running.

A. See the below:

B. PlaceHolder

Correct Answer: A

A service account provides an identity for processes that run in a Pod.

When you (a human) access the cluster (for example, using kubectl), you are authenticated by the apiserver as a particular User Account (currently this is usually admin, unless your cluster administrator has customized your cluster). Processes in containers inside pods can also contact the apiserver. When they do, they are authenticated as a particular Service Account (for example, default).

When you create a pod, if you do not specify a service account, it is automatically assigned the default service account in the same namespace. If you get the raw json or yaml for a pod you have created (for example, kubectl get pods/ -o yaml), you can see the spec.serviceAccountName field has been automatically set. You can access the API from inside a pod using automatically mounted service account credentials, as described in Accessing the Cluster. The API permissions of the service account depend on the authorization plugin and policy in use. In version 1.6+, you can opt out of automounting API credentials for a service account by setting automountServiceAccountToken: false on the service account:

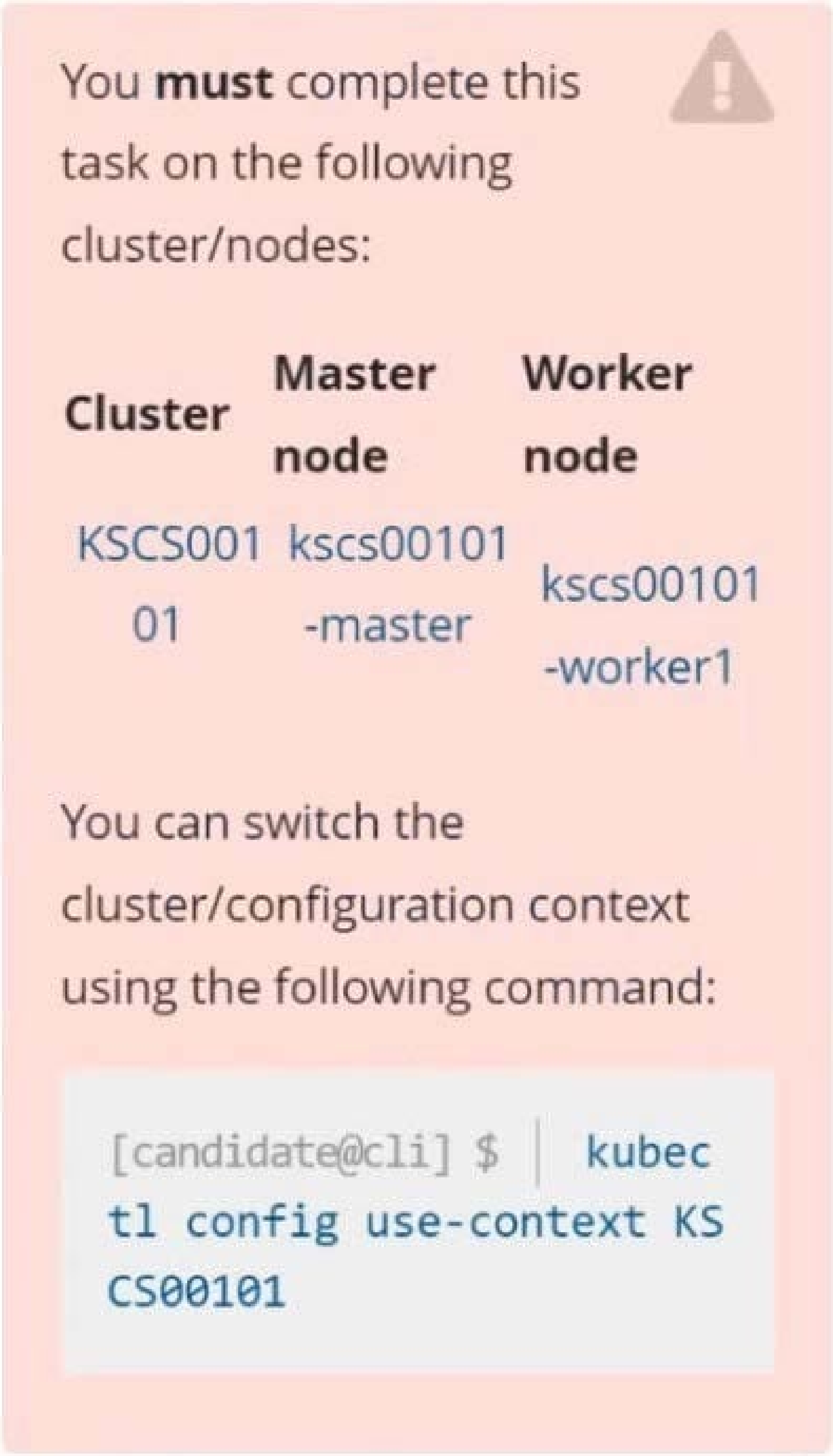
apiVersion: v1 kind: ServiceAccount metadata: name: build-robot automountServiceAccountToken: false

In version 1.6+, you can also opt out of automounting API credentials for a particular pod: apiVersion: v1 kind: Pod metadata: name: my-pod spec: serviceAccountName: build-robot automountServiceAccountToken: false

The pod spec takes precedence over the service account if both specify a automountServiceAccountToken value.

QUESTION 5

CORRECT TEXT Context



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

Cluster	Master node	Worker node
KSCS00101	kscs00101-master	kscs00101-worker1

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

```
[candidate@cli] $ kubectl config use-context KSCS00101
```

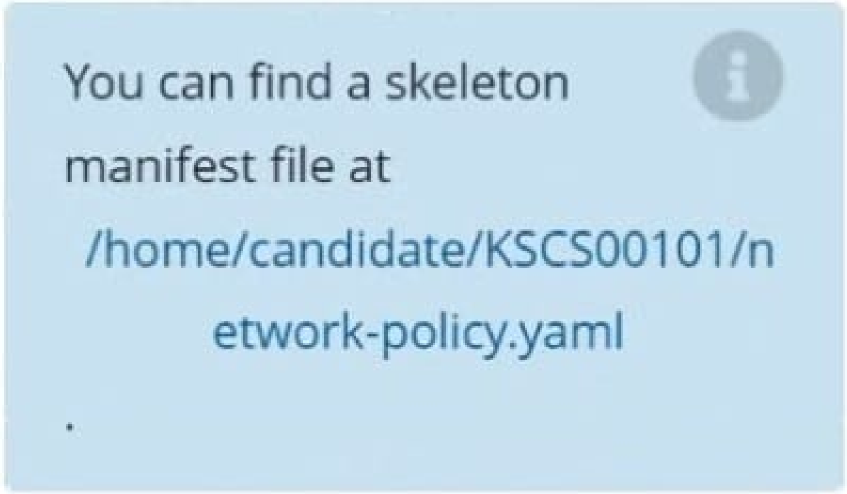
A default-deny NetworkPolicy avoids to accidentally expose a Pod in a namespace that doesn\\'t have any other NetworkPolicy defined.

Task

Create a new default-deny NetworkPolicy named defaultdeny in the namespace testing for all traffic of type Egress.

The new NetworkPolicy must deny all Egress traffic in the namespace testing.

Apply the newly created default-deny NetworkPolicy to all Pods running in namespace testing.



You can find a skeleton manifest file at /home/candidate/KSCS00101/network-policy.yaml

.

A. See explanation below.

B. PlaceHolder

Correct Answer: A