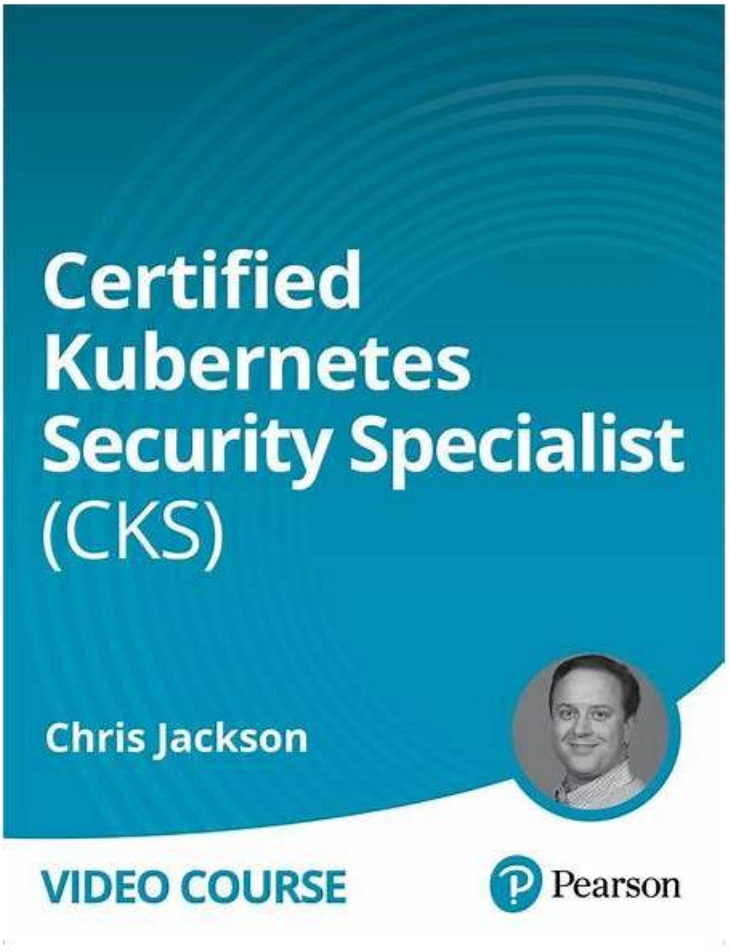
CKS認證題庫：Certified Kubernetes Security Specialist (CKS)考試最新發布|更新的CKS測試題庫



P.S. Fast2test在Google Drive上分享了免費的、最新的CKS考試題庫：https://drive.google.com/open?id=1f0uchBjSXNkETwORoOrptB6nhku9YDwg

作為一名專業的IT人員，如何證明自己的能力，加強自己在公司的地位，獲得Linux Foundation CKS認證可以提高你的IT技能，以獲得更好的工作機會。快登錄Fast2test網站吧！這里有大量的學習資料試題和答案，是滿足嚴格質量標準的考試題庫，涵蓋所有的Linux Foundation CKS考試知識點。客戶成功購買我們的CKS題庫資料之后，都將享受一年的免費更新服務，一年之內，如果您購買的CKS學習資料更新了，我們將免費發送最新版本的到您的郵箱。

Linux Foundation CKS Exam Overview:

Certification Vendor:	The Linux Foundation
Exam Name:	Certified Kubernetes Security Specialist
Exam Number:	CKS
Exam Duration:	120 minutes
Exam Format:	Performance-based hands-on command line tasks
Related Certifications:	CKA (Certified Kubernetes Administrator)
Real Exam Qty:	15-20
Exam Price:	$395 USD
Certificate Validity Period:	2 years

Available Languages:	English
Passing Score:	66%
Sample Questions:	Linux Foundation CKS Sample Questions
Exam Way:	Online proctored exam (remote) or at a testing center
Pre Condition:	CKA (Certified Kubernetes Administrator) certification is required before taking CKS
Official Syllabus URL:	https://training.linuxfoundation.org/certification/certified-kubernetes-security-specialist/

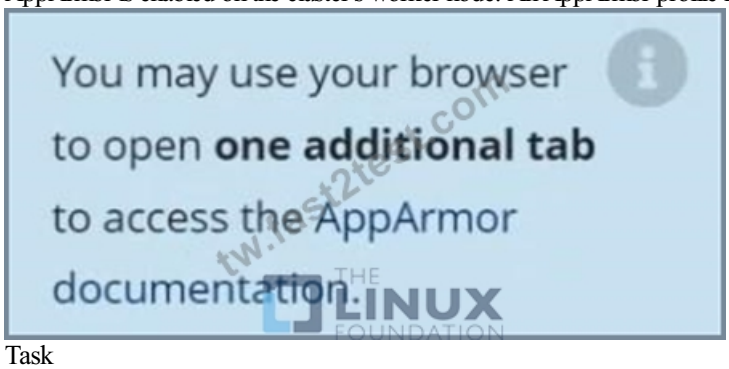
有效的CKS認證題庫 |第一次嘗試輕鬆學習並通過考試和專業的Linux Foundation Certified Kubernetes Security Specialist (CKS)

當你嘗試了我們提供的關於Linux Foundation CKS認證考試的部分考題及答案,你可以對我們Fast2test做出選擇了，我們會100%為你提供方便以及保障。請記住能讓你100%通過Linux Foundation CKS認證考試的就是我們的Fast2test。

Linux Foundation CKS（Certified Kubernetes Security Specialist）考試是一項認證計畫，旨在評估和驗證個人在保護基於容器的應用和Kubernetes平台方面的專業知識。該考試針對負責保護Kubernetes集群並確保符合行業公認的安全標準的專業人員。CKS認證計畫是一個供所有具有良好Kubernetes和其安全原則理解的IT專業人員參加的中立計畫。

最新的 Kubernetes Security Specialist CKS 免費考試真題 (Q52-Q57):

問題 #52
Context
AppArmor is enabled on the cluster's worker node. An AppArmor profile is prepared, but not enforced yet.



Task
On the cluster's worker node, enforce the prepared AppArmor profile located at /etc/apparmor.d/nginx_apparmor.
Edit the prepared manifest file located at /home/candidate/KSSH00401/nginx-pod.yaml to apply the AppArmor profile.
Finally, apply the manifest file and create the Pod specified in it.

答案：

解題說明：

```
candidate@cli:~$ kubectl config use-context KSSH00401
Switched to context "KSSH00401".
candidate@cli:~$ ssh kssh00401-worker1
Warning: Permanently added '10.240.86.172' (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@kssh00401-worker1:~# head /etc/apparmor.d/nginx_apparmor
#include <tunables/global>

profile nginx-profile-2 flags=(attach_disconnected,mediate_deleted) {
 #include <abstractions/base>
  network inet tcp,
  network inet udp,
  network inet icmp,

  deny network raw,

root@kssh00401-worker1:~# apparmor_parser -q /etc/apparmor.d/nginx_apparmor
root@kssh00401-worker1:~# exit
logout
Connection to 10.240.86.172 closed.
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
```



```
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-p
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
~
```

```
candidate@cli:~$ vim KSSH00401/nginx-pod.yaml
candidate@cli:~$ kubectl create -f KSSH00401/nginx-pod.yaml
pod/nginx-pod created
candidate@cli:~$ cat KSSH00401/nginx-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  annotations:
    container.apparmor.security.beta.kubernetes.io/nginx-pod: localhost/nginx-profile-2
spec:
  containers:
  - name: nginx-pod
    image: nginx:1.19.0
    ports:
    - containerPort: 80
```


問題 #53
You have a Kubernetes cluster that runs a sensitive application called "banking-app" in a Deployment The application needs access to a private registry to pull container images. You want to ensure that the "banking-app" container only communicates with the private registry and no other external networks. How can you use NetworkPolicy to enforce this network security restriction?

答案：

解題說明：
Solution (Step by Step) :
1. Create a NetworkPolicy tor the Private Registry: You'll create a NetworkPolicy that allows the "banking-app" container to communicate with the private registry but blocks access to all other external networks.

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: banking-app-registry-access
  namespace: default  # Your application's namespace
spec:
  podSelector:
    matchLabels:
      app: banking-app
  policyTypes:
    - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: banking-app
    - ipBlock:
        cidr: "172.17.0.0/16"  # Replace with your private registry's CIDR
        except: []
```


'podSelectors: This defines which pods are affected by the policy. - 'policyTypeS: This specifies the type of traffic that the policy governs (Ingress in this case). - 'ingress': Defines the allowed incoming traffic. - 'trom': Specifies the source of allowed traffic. - spodSeIector': Allows traffic from other pods with the "banking-app" label. - 'ipBlock': Allows traffic from a specific CIDR range. - 'cidr': Replace '172.17.0.0/16' with the actual CIDR of your private registry. - 'except': Optional for excluding specific IP addresses or ranges. 2 Apply the NetworkPolicy: Apply the YAML file to your cluster: bash kubectl apply -f banking-app-registry-access.yaml 3. Verify NetworkPolicy: After applying the policy, run: bash kubectl get networkpolicy -n default # Replace 'default' with your namespace You should see your new "banking-app-registry-access" NetworkPolicy listed. 4. Test the Policy: - Try to access external networks from within the "banking-app" container. - You should observe that the container is unable to connect to any external services except the private registry. - Make sure your application can still pull images from the private registry. 5. Additional Considerations: - Egress Traffic: You might need to define a separate NetworkPolicy for 'Egress' traffic if you want to allow the "banking-app" to communicate with specific internal services. - Detailed Controls: You can add more specific rules to the

'ingress' section to allow specific ports or protocols from the private registry.

問題 #54
You are managing a Kubernetes cluster with a deployment named 'database-deployment' running 3 replicas of a PostgreSQL database container. You need to implement a security policy that restricts the database pods from accessing the internet, allowing them to only communicate with each other and with specific external services. The allowed external services include a dedicated monitoring service at 'monitoring-example-com:8080' and a logging service at 'logging-example-com:514'. Additionally, you want to enforce this policy using NetworkPolicy.

答案：

解題說明：
Solution (Step by Step) :
1. Create a NetworkPolicy for database pods:
- Create a YAML file named "database-networkpolicy.yamr with the following contents:

```
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
```

```
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 100.64.0.0/10 # Allow communication to monitoring service
    except:
    - to:
        - ipBlock:
            cidr: 100.64.0.0/12 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 172.16.0.0/12
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
```

```yaml
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 10.128.0.0/14 # Allow communication to logging service
    except:
    - to:
        - ipBlock:
            cidr: 10.128.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0/14
    except:
    - to:
        - ipBlock:
            cidr: 192.168.0.0/16 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.99.0/24
- ipBlock:
    cidr: 172.17.0.0/24
- ipBlock:
    cidr: 10.0.0.0/24
- ipBlock:
    cidr: 100.64.0.0/10 # Allow communication to monitoring service
    except:
    - to:
        - ipBlock:
            cidr: 100.64.0.0/12 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 172.16.0.0/12
    except:
    - to:
        - ipBlock:
            cidr: 172.16.0.0/14 # Exclude a specific range for security reasons (optional)
- ipBlock:
    cidr: 192.168.0.0
```

問題 #55
SIMULATION
Create a network policy named allow-np, that allows pod in the namespace staging to connect to port 80 of other pods in the same namespace.
Ensure that Network Policy:-
1. Does not allow access to pod not listening on port 80.
2. Does not allow access from Pods, not in namespace staging.

答案：

解題說明：
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: network-policy
spec:
podSelector: {} #selects all the pods in the namespace deployed
policyTypes:
- Ingress
ingress:
- ports: #in input traffic allowed only through 80 port only
- protocol: TCP
port: 80

問題 #56
You are managing a Kubernetes cluster With a critical application deployed as a Deployment. You need to ensure that only authorized users can access the Kubernetes API to manage this application's resources. Describe the security measures you would implement to restrict access to the Kubernetes API and ensure only authorized users can manage this specific Deployment

答案：

解題說明：
Solution (Step by Step):
1. Create a Service Account:
- Create a dedicated Service Account for the application:

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: critical-app-sa
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: critical-app-role
rules:
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["get", "list", "watch", "update", "patch", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: critical-app-rolebinding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: critical-app-role
subjects:
- kind: ServiceAccount
  name: critical-app-sa
  namespace: default
```


2. Configure the Deployment to use the Service Account - Update the Deployment YAML to specify the Service Account:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: critical-app
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: critical-app
    spec:
      serviceAccountName: critical-app-sa
      containers:
      - name: critical-app-container
        image: example/critical-app:latest
        # ... other container configurations ...
```


3. Use RBAC (Role-Based Access Control): - Create a Role and RoleBinding for the Service Account:

4. Apply the Changes: - Apply the YAML configurations using 'kubectl apply -f critical-app.yaml' 5. Restrict Access to the Kubernetes API: - Use kubeconfig files to limit access to the API for specific users or groups. - Use tools like 'kubectl auth can-i' to verify access permissions. 6. Limit Access to Specific Namespaces: - By default, Service Accounts have access to resources in their respective namespaces. - To further restrict access, create a Namespace and limit the Service Account's permissions within that namespace. By implementing these measures, you create a secure environment where only authorized users (through the Service Account) can manage the critical application's Deployment

問題 #57

.....

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, 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, Disposable vapes

P.S. Fast2test在Google Drive上分享了免費的、最新的CKS考試題庫：https://drive.google.com/open?id=1f0uchBjSXNkETwORoOrptB6nhku9YDwg