Boost Security in Kubernetes with CIS Security Controls and Benchmarks

Ader Fu <ader.ydfu@gmail.com>

KCD Taipei 2024, Aug. 4, 2024

KCD

GROWING CLOUD NATIVE TOGETHER

About me

@ydFu

ader.ydfu@gmail.com

Name:

- **Ader Fu, a.k.a. Yao-De Fu** has extensive experience in DevOps.

Co-organizer:

- Cloud Native Taiwan User Group (https://www.facebook.com/groups/cloudnative.tw)
- Kubernetes Community Day(KCD Taiwan 2022&2023, KCD Taipei 2024)

Organization:

- Member: **kubernetes, kubernetes-sigs, istio**
- Teams: **Kubernetes Special Interest Group (SIG) Docs**

Community role:

- **cncf/glossary approver**(https://github.com/cncf/glossary)
- **kubernetes/website approver and reviewer**(https://github.com/kubernetes/website)

Certificate:

- **1st CNCF Kubestronauts in Taiwan** (https://www.cncf.io/training/kubestronaut/)
- **CKA, CKAD, CKS, KCNA and KCSA**
- **Prometheus Certified Associate (PCA)**

Outline

- Center for Internet Security (CIS) introduction
- Applying CIS benchmarks in Kubernetes
- Leveraging CNCF Landscape’s security and compliance projects

kubernetes

Center for Internet Security (CIS)

What is system hardening?

A process of fortifying system configuration settings to safeguard against attack vectors—methods of gaining unauthorized access to computer or network systems—thereby reducing an organization's attack surface, which encompasses all possible points where an unauthorized user can access a system and extract data, to mitigate the risk of cyber-attacks.

Advantage:

- Reduce the risk of being attacked
- Improve system stability
- Meet compliance requirements

kubernetes

Configuration compliance scan

Configuration compliance scan is a Security Configuration Assessment (SCA) to verify if an asset's (Servers, Networking devices, IOT, Database, etc.) present configuration is based on best practice standards.

- Known System Hardening Configuration Standards

1. Center for Internet Security (CIS) **- CIS Benchmarks**(https://www.cisecurity.org/cis-benchmarks-overview)
2. National Institute of Standards and Technology (NIST) – **SP 800-123(General Server Security) and 800-53B(Security and Privacy Control)** (https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-123.pdf) & (https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.80053r5.pdf)
3. European Network and Information Security Agency (ENISA) (https://www.enisa.europa.eu)

CIS introduction

The Center for Internet Security, Inc. (CIS®)

- CIS are a community-driven nonprofit, responsible for the CIS Controls® and CIS Benchmarks™, globally recognized best practices for securing IT systems and data.
- MS-ISAC and EI-ISAC are the key resources for cyber threat prevention, protection, response, and recovery for the nation's state, local, tribal, and territorial (SLTT) governments and elections communities, respectively

The CIS Vision

Leading the global community to secure our ever-changing connected world.

The CIS Mission

Our mission is to make the connected world a safer place by developing, validating, and promoting timely best practice solutions that help people, businesses, and governments protect themselves against pervasive cyber threats.

What Types of IT Systems Do CIS Benchmarks Cover?

Cloud Providers

Alibaba Cloud
Amazon Web Services
Google Cloud Computing Platform
Microsoft 365
Microsoft Azure…

Desktop Software

Google Chrome
Microsoft Office
Microsoft Web Browser
Mozilla Firefox
Safari Browser
Zoom…

DevSecOps Tools

Software Supply Chain Security

DEV SEC OPS

Mobiles Devices

Apple iOS
Google Android

Multi Function Print Devices

Print Devices

Network Devices

Check Point Firewall
Cisco
F5
Fortinet
Juniper
Palo Alto Networks
pfSense Firewall…

Operating System

Apple macOS
Azure Linux
Bottlerocket
CentOS Linux
Debian Family Linux
Microsoft Windows Server…

Server Software

Apache HTTP Server
Apache Tomcat
BIND
Docker
Kubernetes
MariaDB
Microsoft IIS…

Critical Security Controls

CIS Controls

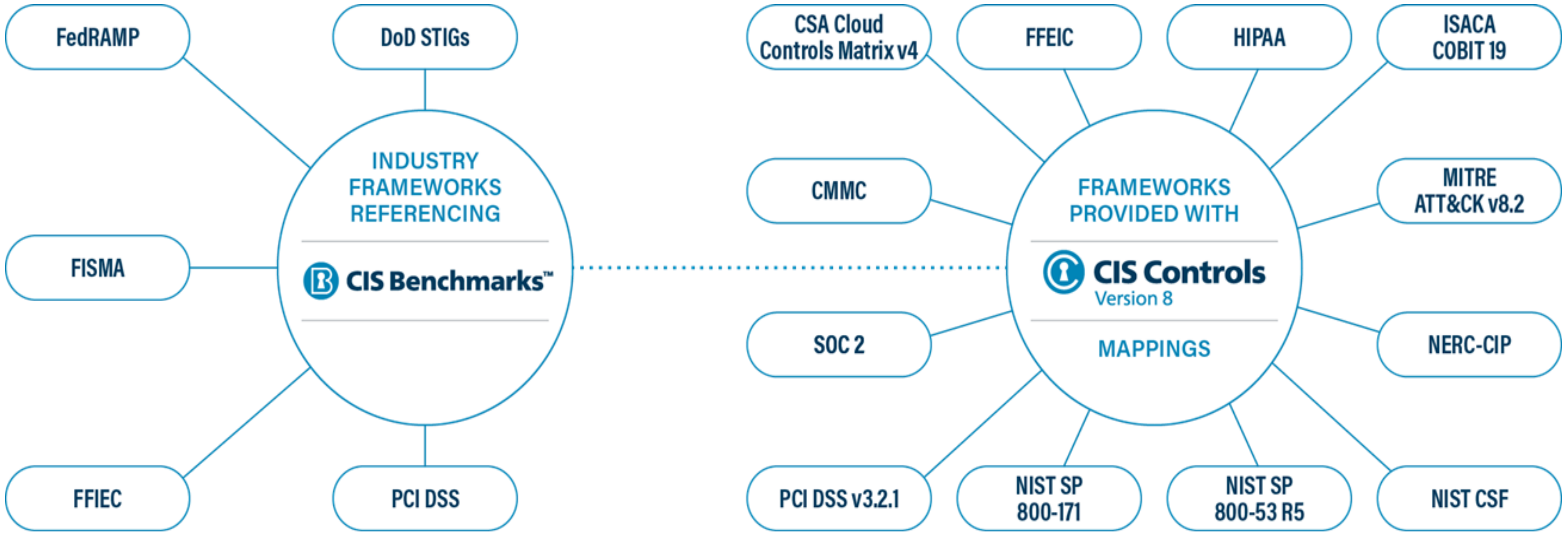
CONTROL 01	Inventory and Control of Enterprise Assets	5 Safeguards	IG1 2/5	IG2 4/5	IG3 5/5
CONTROL 02	Inventory and Control of Software Assets	7 Safeguards	IG1 3/7	IG2 6/7	IG3 7/7
CONTROL 03	Data Protection	14 Safeguards	IG1 6/14	IG2 12/14	IG3 14/14
CONTROL 04	Secure Configuration of Enterprise Assets and Software	12 Safeguards	IG1 7/12	IG2 11/12	IG3 12/12
CONTROL 05	Account Management	6 Safeguards	IG1 4/6	IG2 6/6	IG3 6/6
CONTROL 06	Access Control Management	8 Safeguards	IG1 5/8	IG2 7/8	IG3 8/8
CONTROL 07	Continuous Vulnerability Management	7 Safeguards	IG1 4/7	IG2 7/7	IG3 7/7
CONTROL 08	Audit Log Management	12 Safeguards	IG1 3/12	IG2 11/12	IG3 12/12
CONTROL 09	Email and Web Browser Protections	7 Safeguards	IG1 2/7	IG2 6/7	IG3 7/7
CONTROL 10	Malware Defenses	7 Safeguards	IG1 3/7	IG2 7/7	IG3 7/7
CONTROL 11	Data Recovery	5 Safeguards	IG1 4/5	IG2 5/5	IG3 5/5
CONTROL 12	Network Infrastructure Management	8 Safeguards	IG1 1/8	IG2 7/8	IG3 8/8
CONTROL 13	Network Monitoring and Defense	11 Safeguards	IG1 0/11	IG2 6/11	IG3 11/11
CONTROL 14	Security Awareness and Skills Training	9 Safeguards	IG1 8/9	IG2 9/9	IG3 9/9
CONTROL 15	Service Provider Management	7 Safeguards	IG1 1/7	IG2 4/7	IG3 7/7
CONTROL 16	Applications Software Security	14 Safeguards	IG1 0/14	IG2 11/14	IG3 14/14
CONTROL 17	Incident Response Management	9 Safeguards	IG1 3/9	IG2 8/9	IG3 9/9
CONTROL 18	Penetration Testing	5 Safeguards	IG1 0/5	IG2 3/5	IG3 5/5

What are the CIS Benchmarks?

- 100+ CIS Benchmarks covering 25 vendor product families (e.g., AWS Foundations, AWS End User Compute, Amazon EKS)
- Recognized by industry frameworks – DISA STIGs, FISMA, FedRAMP, PCI DSS
- Community developed – CIS members, SMEs, security community experts, and technology vendors
- Prescriptive instruction
  - ➢ Step-by-step list to apply configurations
  - ➢ Rationale on why the configuration is recommended
  - ➢ Impact the configuration will make
  - ➢ Mapped to CIS Controls

CIS Benchmarks™ kubernetes

Scope of Impact of CIS Benchmarks and CIS Controls



kubernetes

Government Configuration Baseline(GCB)

- 政府組態基準(Government Configuration Baseline，以下簡稱GCB)目的在於規範資通訊終端設備(如：個人電腦、伺服器主機及網通設備)的一致性安全設定(如：密碼長度、更新期限等)，以降低成為駭客入侵管道，進而引發資安事件之疑慮。
- 國家資通安全研究社(核心業務-資安防護)

1 發展一致的安全組態設定

2 提升政府機關資通訊終端設備使用安全性

GCB 對應法規與落地場域

資通安全責任等級分級辦法	規定項目：政府組態基準(GCB)	
	A、B級公務機關	初次受核定或等級變更後之一年內，依主管機關公告之項目，完成政府組態基準導入作業，並持續維運。
	A、B級特定非公務機關	特定非公務機關之中央目的事業主管機關得視實際需求，於符合本辦法規定之範圍內，另行訂定其所管特定非公務機關之資通安全應辦事項。

CIS Benchmarks 與 GCB 的關係

- 以 Red Hat Enterprise Linux 8 TWGCB-01-008(v1.0)-政府組態基準說明文件。
- 共計 292 項規定，分基本項目(243 項)，防火牆(18 項)以及 SSH 服務(31 項)。
- 主要參考下列二項國際知名資訊安全標準:
  - ➢ **CIS (Center for Internet Security) RHEL 8 Benchmark**
  - ➢ DISA (Defense Information Systems Agency) RHEL 8 STIG
  - ➢ TWGCB for RHEL 5

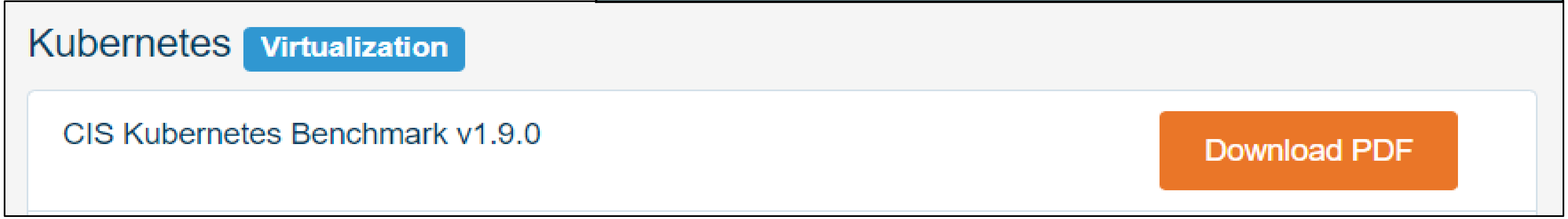
項次	項目	項數	合計
1	基本項目 (系統設定與維護、網路設定、日誌與稽核、SELinux…)	243	292
2	Firewalld 防火牆組態基準	5	
3	Nftables 防火牆組態基準	7	
4	Iptables 防火牆組態基準	6	
5	SSH 伺服器組態基準	31	

Applying CIS benchmarks in Kubernetes

Covers the classification of Kubernetes

- The Types of Kubernetes in CIS Benchmarks is in **Server Software/Kubernetes**
- Category links: https://www.cisecurity.org/benchmark/kubernetes

Covers the classification of Kubernetes
Red Hat OpenShift Container Platform
Google Kubernetes Engine (GKE)
Amazon Elastic Kubernetes Service (EKS)
Oracle Cloud Infrastructure Container Engine for Kubernetes (OKE)
Azure Kubernetes Service (AKS)
Alibaba Cloud Container Service For Kubernetes (ACK)
Kubernetes


Kubernetes Virtualization

CIS Kubernetes Benchmark v1.9.0

CIS Benchmark Implementation Guide

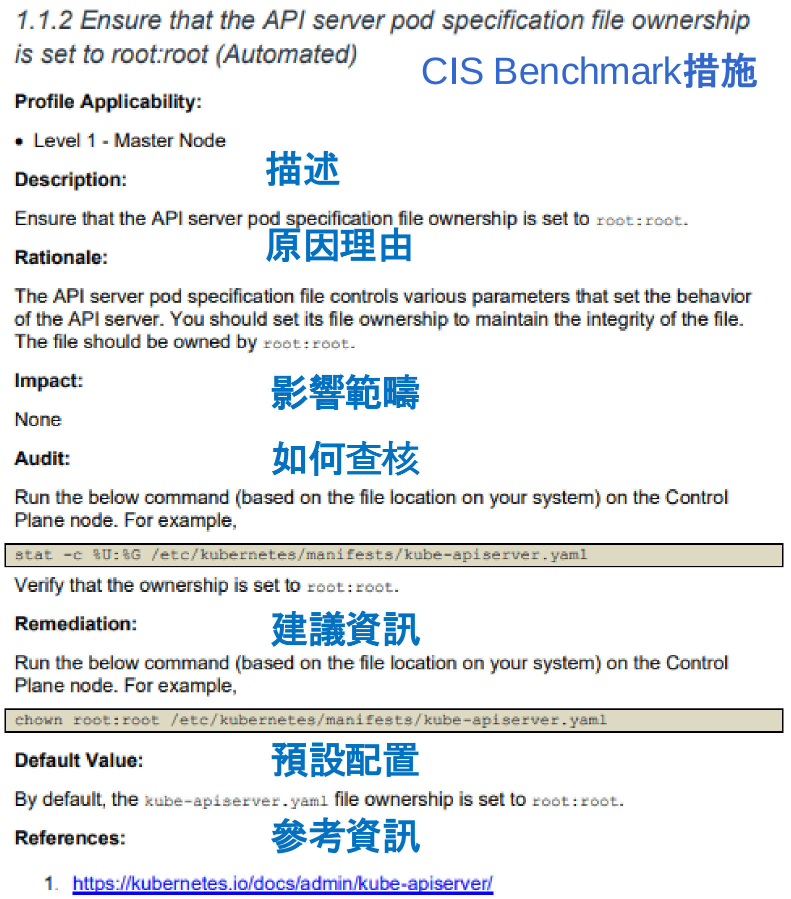
CIS. Center for Internet Security®

CIS Benchmarks™

CIS Kubernetes Benchmark

v1.9.0 - 03-25-2024

Source:
https://www.cisecurity.org/benchmark/kubernetes



CIS Benchmark措施

1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated)

Profile Applicability:

- Level 1 - Master Node

Description: 描述

Ensure that the API server pod specification file ownership is set to `root:root`.

Rationale: 原因理由

The API server pod specification file controls various parameters that set the behavior of the API server. You should set its file ownership to maintain the integrity of the file. The file should be owned by `root:root`.

Impact: 影響範疇

None

Audit: 如何查核

Run the below command (based on the file location on your system) on the Control Plane node. For example,

```
stat -c %U:%G /etc/kubernetes/manifests/kube-apiserver.yaml
```

Verify that the ownership is set to `root:root`.

Remediation: 建議資訊

Run the below command (based on the file location on your system) on the Control Plane node. For example,

```
chown root:root /etc/kubernetes/manifests/kube-apiserver.yaml
```

Default Value: 預設配置

By default, the `kube-apiserver.yaml` file ownership is set to `root:root`.

References: 參考資訊

1. https://kubernetes.io/docs/admin/kube-apiserver/

CIS benchmarks Level

Level 1

- More basic
- Smaller service impact
- For small & medium businesses
- For organizations that don't handle PII or other sensitive data

Level 2

- More comprehensive
- Can impact business functions
- For enterprises & critical infrastructure like energy, communications, healthcare, finance, etc.

CIS Benchmarks™

Applying CIS benchmarks in Kubernetes

1. Control Plane Components
2. Etcd
3. Control Plane Configuration
4. Worker Nodes
5. Policies

CIS Benchmarks™

Source: https://www.cisecurity.org/benchmark/kubernetes

1.1 Control Plane Node Configuration Files

1.1.1 Ensure that the API server pod specification file permissions are set to 600 or more restrictive (Automated)
1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated)
1.1.3 Ensure that the controller manager pod specification file permissions are set to 600 or more restrictive (Automated)
1.1.4 Ensure that the controller manager pod specification file ownership is set to root:root (Automated)
1.1.5 Ensure that the scheduler pod specification file permissions are set to 600 or more restrictive (Automated)
1.1.6 Ensure that the scheduler pod specification file ownership is set to root:root (Automated)
1.1.7 Ensure that the etcd pod specification file permissions are set to 600 or more restrictive (Automated)
1.1.8 Ensure that the etcd pod specification file ownership is set to root:root (Automated)
1.1.9 Ensure that the Container Network Interface file permissions are set to 600 or more restrictive (Manual)
1.1.10 Ensure that the Container Network Interface file ownership is set to root:root (Manual)

CIS Benchmarks™

1.1 Control Plane Node Configuration Files

1.1.11 Ensure that the etcd data directory permissions are set to 700 or more restrictive (Automated)
1.1.12 Ensure that the etcd data directory ownership is set to etcd:etcd (Automated)
1.1.13 Ensure that the default administrative credential file permissions are set to 600 (Automated)
1.1.14 Ensure that the default administrative credential file ownership is set to root:root (Automated)
1.1.15 Ensure that the scheduler.conf file permissions are set to 600 or more restrictive (Automated)
1.1.16 Ensure that the scheduler.conf file ownership is set to root:root (Automated)
1.1.17 Ensure that the controller-manager.conf file permissions are set to 600 or more restrictive (Automated)
1.1.18 Ensure that the controller-manager.conf file ownership is set to root:root (Automated)
1.1.19 Ensure that the Kubernetes PKI directory and file ownership is set to root:root (Automated)
1.1.20 Ensure that the Kubernetes PKI certificate file permissions are set to 600 or more restrictive (Manual)
1.1.21 Ensure that the Kubernetes PKI key file permissions are set to 600 (Manual)

CIS Benchmarks™

1.2 API Server

1.2.1 Ensure that the --anonymous-auth argument is set to false (Manual)
1.2.2 Ensure that the --token-auth-file parameter is not set (Automated)
1.2.3 Ensure that the DenyServiceExternalIPs is set (Manual)
1.2.4 Ensure that the --kubelet-client-certificate and --kubeletclient-key arguments are set as appropriate (Automated)
1.2.5 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Automated)
1.2.6 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)
1.2.7 Ensure that the --authorization-mode argument includes Node (Automated)
1.2.8 Ensure that the --authorization-mode argument includes RBAC (Automated)
1.2.9 Ensure that the admission control plugin EventRateLimit is set (Manual)
1.2.10 Ensure that the admission control plugin AlwaysAdmit is not set (Automated)

CIS Benchmarks™

1.2 API Server

1.2.11 Ensure that the admission control plugin AlwaysPullImages is set (Manual)
1.2.12 Ensure that the admission control plugin ServiceAccount is set (Automated)
1.2.13 Ensure that the admission control plugin NamespaceLifecycle is set (Automated)
1.2.14 Ensure that the admission control plugin NodeRestriction is set (Automated)
1.2.15 Ensure that the --profiling argument is set to false (Automated)
1.2.16 Ensure that the --audit-log-path argument is set (Automated)
1.2.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Automated)
1.2.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Automated)
1.2.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Automated)
1.2.20 Ensure that the --request-timeout argument is set as appropriate (Manual)

CIS Benchmarks™

1.2 API Server

1.2.21 Ensure that the --service-account-lookup argument is set to true (Automated)
1.2.22 Ensure that the --service-account-key-file argument is set as appropriate (Automated)
1.2.23 Ensure that the --etcd-certfile and --etcd-keyfile arguments are set as appropriate (Automated)
1.2.24 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Automated)
1.2.25 Ensure that the --client-ca-file argument is set as appropriate (Automated)
1.2.26 Ensure that the --etcd-cafile argument is set as appropriate (Automated)
1.2.27 Ensure that the --encryption-provider-config argument is set as appropriate (Manual)
1.2.28 Ensure that encryption providers are appropriately configured (Manual)
1.2.29 Ensure that the API Server only makes use of Strong Cryptographic Ciphers (Manual)

CIS Benchmarks™

1.3 Controller Manager

1.3.1 Ensure that the --terminated-pod-gc-threshold argument is set as appropriate (Manual)

1.3.2 Ensure that the --profiling argument is set to false (Automated)

1.3.3 Ensure that the --use-service-account-credentials argument is set to true (Automated)

1.3.4 Ensure that the --service-account-private-key-file argument is set as appropriate (Automated)

1.3.5 Ensure that the --root-ca-file argument is set as appropriate (Automated)

1.3.6 Ensure that the RotateKubeletServerCertificate argument is set to true (Automated)

1.3.7 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)

CIS Benchmarks™

1.4 Scheduler

1.4.1 Ensure that the --profiling argument is set to false (Automated)

1.4.2 Ensure that the --bind-address argument is set to 127.0.0.1 (Automated)

CIS Benchmarks™

2. Etcd

2.1 Ensure that the --cert-file and --key-file arguments are set as appropriate (Automated)
2.2 Ensure that the --client-cert-auth argument is set to true (Automated)
2.3 Ensure that the --auto-tls argument is not set to true (Automated)
2.4 Ensure that the --peer-cert-file and --peer-key-file arguments are set as appropriate (Automated)
2.5 Ensure that the --peer-client-cert-auth argument is set to true (Automated)
2.6 Ensure that the --peer-auto-tls argument is not set to true (Automated)
2.7 Ensure that a unique Certificate Authority is used for etcd (Manual)

CIS Benchmarks™

3. Control Plane Configuration

- 3.1 Authentication and Authorization

3.1.1 Client certificate authentication should not be used for users (Manual)

3.1.2 Service account token authentication should not be used for users (Manual)

3.1.3 Bootstrap token authentication should not be used for users (Manual)

- 3.2 Logging

3.2.1 Ensure that a minimal audit policy is created (Manual)

3.2.2 Ensure that the audit policy covers key security concerns (Manual)

4.1 Worker Node Configuration Files

4.1.1 Ensure that the kubelet service file permissions are set to 600 or more restrictive (Automated)
4.1.2 Ensure that the kubelet service file ownership is set to root:root (Automated)
4.1.3 If proxy kubeconfig file exists ensure permissions are set to 600 or more restrictive (Manual)
4.1.4 If proxy kubeconfig file exists ensure ownership is set to root:root (Manual)
4.1.5 Ensure that the --kubeconfig kubelet.conf file permissions are set to 600 or more restrictive (Automated)
4.1.6 Ensure that the --kubeconfig kubelet.conf file ownership is set to root:root (Automated)
4.1.7 Ensure that the certificate authorities file permissions are set to 600 or more restrictive (Manual)

CIS Benchmarks™

4.2 Kubelet

4.2.1 Ensure that the --anonymous-auth argument is set to false (Automated)
4.2.2 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Automated)
4.2.3 Ensure that the --client-ca-file argument is set as appropriate (Automated)
4.2.4 Verify that the --read-only-port argument is set to 0 (Manual)
4.2.5 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Manual)
4.2.6 Ensure that the --make-iptables-util-chains argument is set to true (Automated)
4.2.7 Ensure that the --hostname-override argument is not set (Manual)

CIS Benchmarks™

4.2 Kubelet & 4.3 kube-proxy

- 4.2 Kubelet

4.2.8 Ensure that the eventRecordQPS argument is set to a level which ensures appropriate event capture (Manual)
4.2.9 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Manual)
4.2.10 Ensure that the --rotate-certificates argument is not set to false (Automated)
4.2.11 Verify that the RotateKubeletServerCertificate argument is set to true (Manual)
4.2.12 Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers (Manual)
4.2.13 Ensure that a limit is set on pod PIDs (Manual)

- 4.3 kube-proxy

4.3.1 Ensure that the kube-proxy metrics service is bound to localhost (Automated)

CIS Benchmarks™

5. Policies

- 5.1 RBAC and Service Accounts

5.1.1 Ensure that the cluster-admin role is only used where required (Manual)
5.1.2 Minimize access to secrets (Manual)
5.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual)
5.1.4 Minimize access to create pods (Manual)
5.1.5 Ensure that default service accounts are not actively used. (Manual)
5.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual)
5.1.7 Avoid use of system:masters group (Manual)
5.1.8 Limit use of the Bind, Impersonate and Escalate permissions in the Kubernetes cluster (Manual)
5.1.9 Minimize access to create persistent volumes (Manual)
5.1.10 Minimize access to the proxy sub-resource of nodes (Manual)
5.1.11 Minimize access to the approval sub-resource of certificatesigningrequests objects (Manual)

CIS Benchmarks™

5. Policies

- 5.2 Pod Security Standards

5.2.1 Ensure that the cluster has at least one active policy control mechanism in place (Manual)
5.2.2 Minimize the admission of privileged containers (Manual)
5.2.3 Minimize the admission of containers wishing to share the host process ID namespace (Manual)
5.2.4 Minimize the admission of containers wishing to share the host IPC namespace (Manual)
5.2.5 Minimize the admission of containers wishing to share the host network namespace (Manual)
5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Manual)
5.2.7 Minimize the admission of root containers (Manual)

CIS Benchmarks™

Policies

- Pod Security Standards

5.2.8 Minimize the admission of containers with the NET_RAW capability (Manual)

5.2.9 Minimize the admission of containers with added capabilities (Manual)

5.2.10 Minimize the admission of containers with capabilities assigned (Manual)

5.2.11 Minimize the admission of Windows HostProcess Containers (Manual)

5.2.12 Minimize the admission of HostPath volumes (Manual)

5.2.13 Minimize the admission of containers which use HostPorts (Manual)

Policies

- 5.3 Network Policies and CNI

5.3.1 Ensure that the CNI in use supports Network Policies (Manual)

5.3.2 Ensure that all Namespaces have Network Policies defined (Manual)

- 5.4 Secrets Management

5.4.1 Prefer using secrets as files over secrets as environment variables (Manual)

5.4.2 Consider external secret storage (Manual)

- 5.5 Extensible Admission Control

5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)

CIS Benchmarks™

5. Policies

- 5.7 General Policies

5.7.1 Create administrative boundaries between resources using namespaces (Manual)
5.7.2 Ensure that the seccomp profile is set to docker/default in your pod definitions (Manual)
5.7.3 Apply Security Context to Your Pods and Containers (Manual)
5.7.4 The default namespace should not be used (Manual)

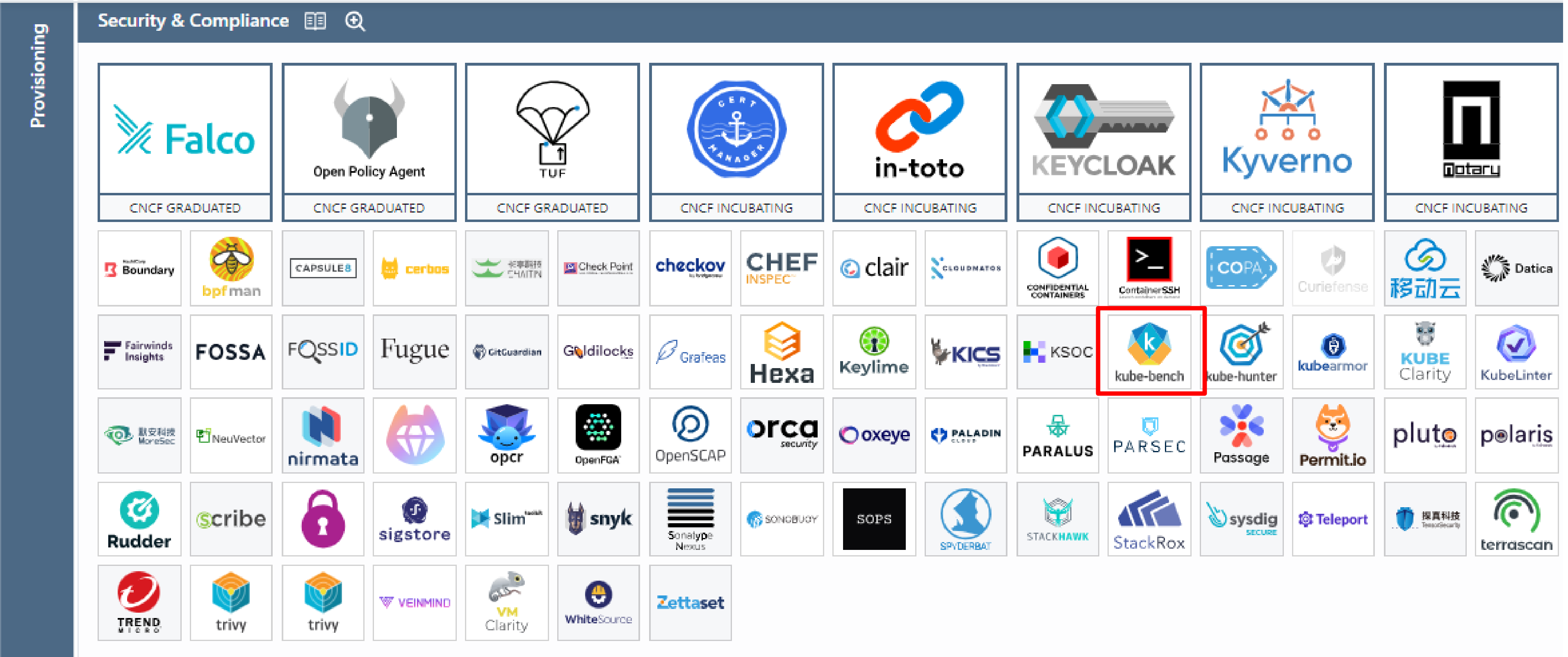
CIS Benchmarks™

Leveraging CNCF Landscape’s security and compliance projects

CNCF landscape:Security & Compliance

CLOUD NATIVE LANDSCAPE

Source: https://landscape.cncf.io/



Provisioning

Security & Compliance

Kube-bench: Kubernetes Security Assessment Tool

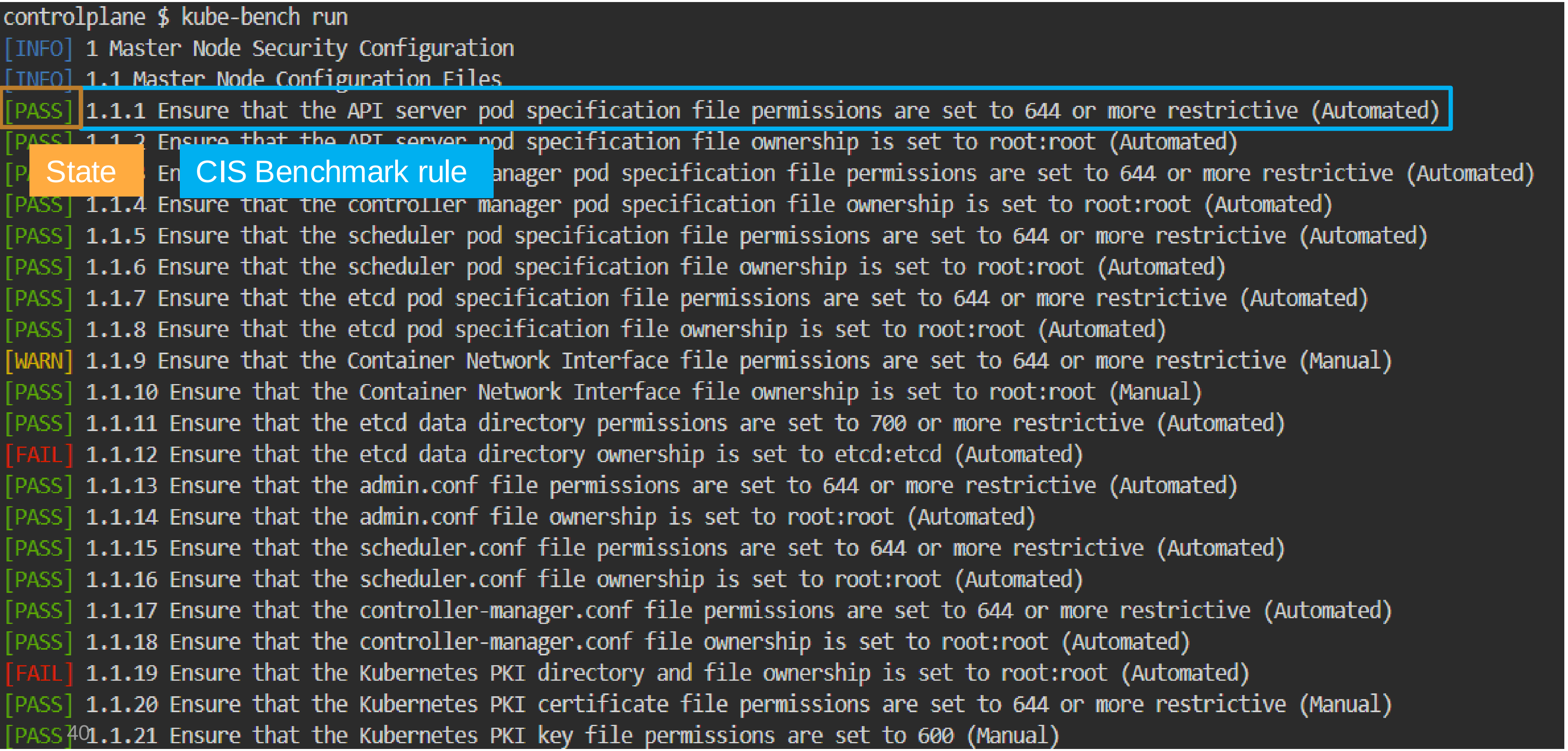
- **Overview:**
  - Kube-bench is an open-source tool for assessing the security of Kubernetes clusters. It runs checks based on the Center for Internet Security (CIS) Kubernetes benchmark and was developed in GoLang by Aqua Security.

- **Key Features:**
  - Security Reconnaissance: Supports both self-managed Kubernetes clusters and clusters managed by cloud providers (e.g., AWS, Azure, GKE).
  - Test Rule Configuration: Configured via YAML files, allowing easy updates.

- **CIS Kubernetes Benchmark:**
  - Follows best practices to ensure Kubernetes cluster security.

- **Reporting and Alerts:**
  - Supports report generation and alert creation based on JSON format scan scripts, with customizable policies.

Running the CIS Benchmark using kube-bench



```
controlplane $ kube-bench run
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 Master Node Configuration Files
[PASS] 1.1.1 Ensure that the API server pod specification file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.2 Ensure that the API server pod specification file ownership is set to root:root (Automated)
[P         En                   anager pod specification file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.4 Ensure that the controller manager pod specification file ownership is set to root:root (Automated)
[PASS] 1.1.5 Ensure that the scheduler pod specification file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.6 Ensure that the scheduler pod specification file ownership is set to root:root (Automated)
[PASS] 1.1.7 Ensure that the etcd pod specification file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.8 Ensure that the etcd pod specification file ownership is set to root:root (Automated)
[WARN] 1.1.9 Ensure that the Container Network Interface file permissions are set to 644 or more restrictive (Manual)
[PASS] 1.1.10 Ensure that the Container Network Interface file ownership is set to root:root (Manual)
[PASS] 1.1.11 Ensure that the etcd data directory permissions are set to 700 or more restrictive (Automated)
[FAIL] 1.1.12 Ensure that the etcd data directory ownership is set to etcd:etcd (Automated)
[PASS] 1.1.13 Ensure that the admin.conf file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.14 Ensure that the admin.conf file ownership is set to root:root (Automated)
[PASS] 1.1.15 Ensure that the scheduler.conf file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.16 Ensure that the scheduler.conf file ownership is set to root:root (Automated)
[PASS] 1.1.17 Ensure that the controller-manager.conf file permissions are set to 644 or more restrictive (Automated)
[PASS] 1.1.18 Ensure that the controller-manager.conf file ownership is set to root:root (Automated)
[FAIL] 1.1.19 Ensure that the Kubernetes PKI directory and file ownership is set to root:root (Automated)
[PASS] 1.1.20 Ensure that the Kubernetes PKI certificate file permissions are set to 644 or more restrictive (Manual)
[PASS 1.1.21 Ensure that the Kubernetes PKI key file permissions are set to 600 (Manual)
```

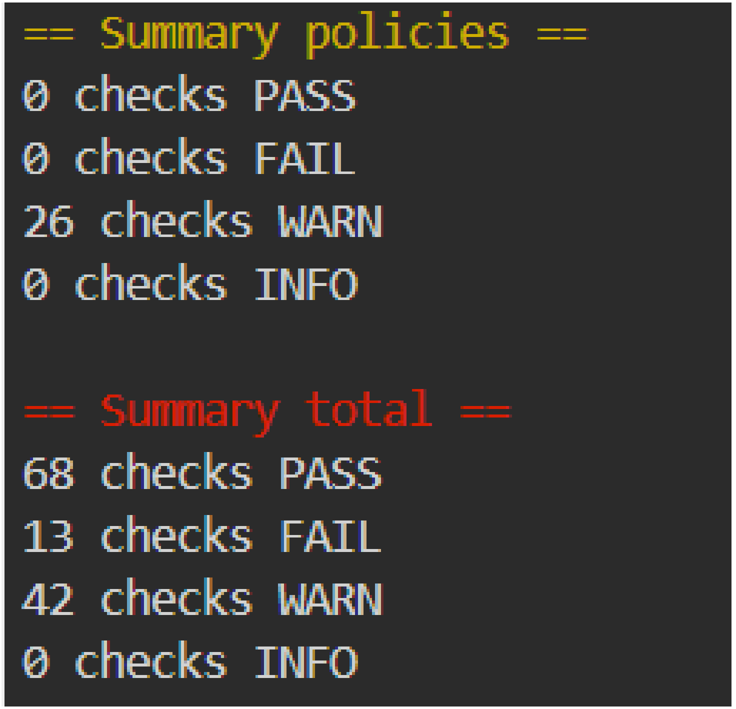
Running the CIS Benchmark using kube-bench

Type	Result	Staute
manual		[WARN]
manual	通過	[PASS]
manual	失敗	[FAIL]
automated	通過	[PASS]
automated	失敗	[FAIL]
skip		[INFO]

[PASS]/[FAIL]: Execution success or failure
[WARN]: Requires further confirmation
[INFO]: No further action needed

kubernetes

Running the CIS Benchmark using kube-bench



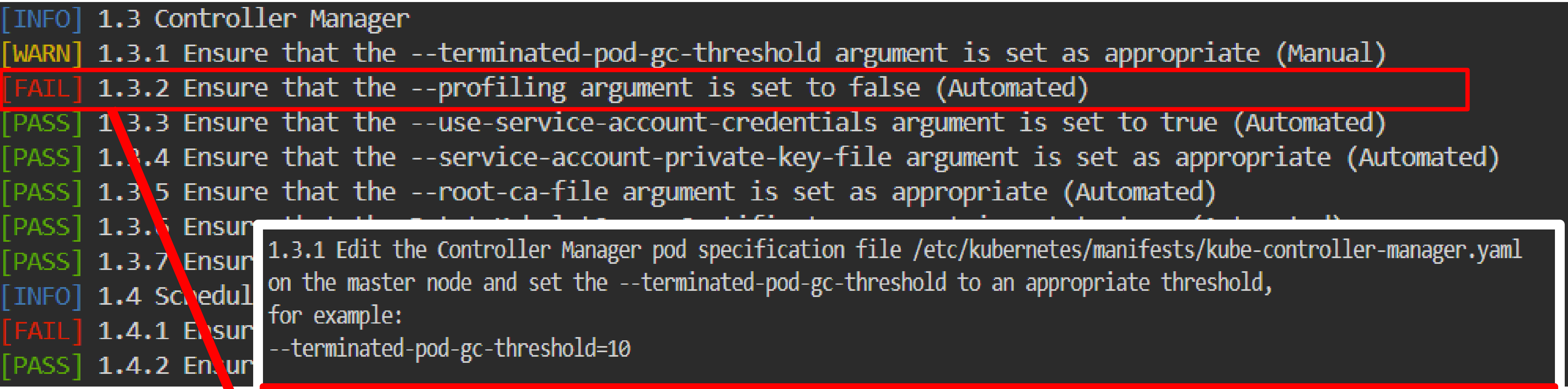
```
== Summary policies ==
0 checks PASS
0 checks FAIL
26 checks WARN
0 checks INFO

== Summary total ==
68 checks PASS
13 checks FAIL
42 checks WARN
0 checks INFO
```

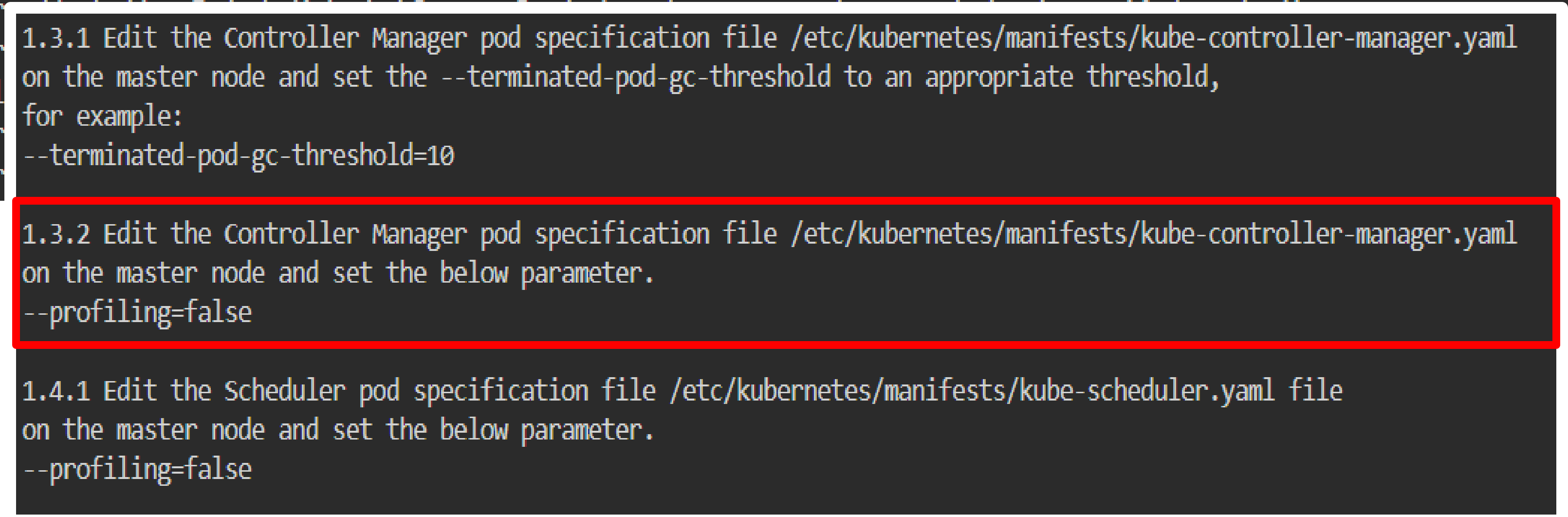
Summary #section

Summary total

Running the CIS Benchmark using kube-bench



```
[INFO] 1.3 Controller Manager
[WARN] 1.3.1 Ensure that the --terminated-pod-gc-threshold argument is set as appropriate (Manual)
[FAIL] 1.3.2 Ensure that the --profiling argument is set to false (Automated)
[PASS] 1.3.3 Ensure that the --use-service-account-credentials argument is set to true (Automated)
[PASS] 1.3.4 Ensure that the --service-account-private-key-file argument is set as appropriate (Automated)
[PASS] 1.3.5 Ensure that the --root-ca-file argument is set as appropriate (Automated)
[PASS] 1.3.6 Ensur
[PASS] 1.3.7 Ensur
[INFO] 1.4 Schedul
[FAIL] 1.4.1 Ensur
[PASS] 1.4.2 Ensur
```



```
1.3.1 Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml
on the master node and set the --terminated-pod-gc-threshold to an appropriate threshold,
for example:
--terminated-pod-gc-threshold=10

1.3.2 Edit the Controller Manager pod specification file /etc/kubernetes/manifests/kube-controller-manager.yaml
on the master node and set the below parameter.
--profiling=false

1.4.1 Edit the Scheduler pod specification file /etc/kubernetes/manifests/kube-scheduler.yaml file
on the master node and set the below parameter.
--profiling=false
```

Remediations Steps

Thanks!

註冊報名抽 CKA 證照優惠卷 /
iThome 2024 活動入場卷