Kubernetes & Shared Responsibility

Understanding one’s scope of responsibility in a Kubernetes cloud service

Why are we here?

Kubernetes is

- Big
- Complex
- New

Overview

- Shared Responsibility Models
- Kubernetes (aka K8)
- Cloud Delivery Model
- Customer Surface Area
- Threat Matrix
- Hardening
- Q&A

Who Am I?

Ken Netzorg

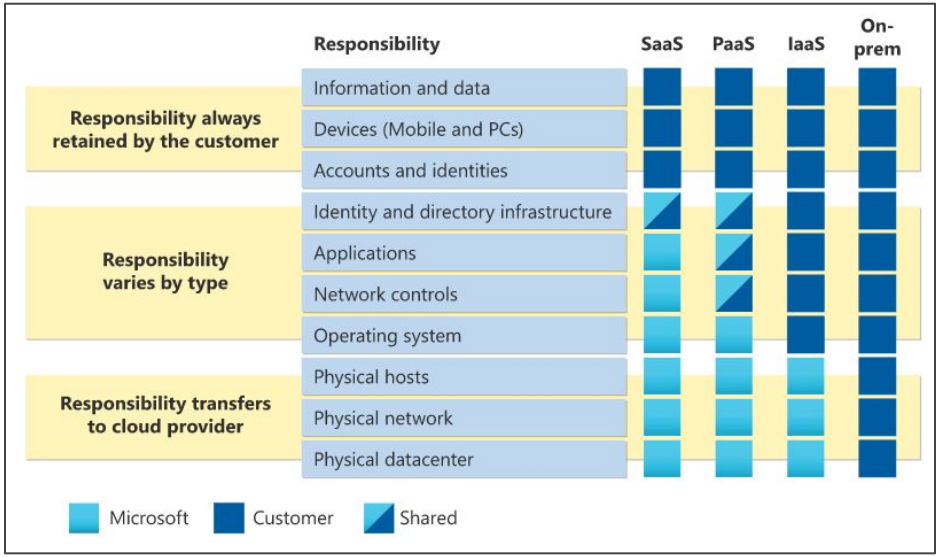
25 Years across various responsibilities ranging from network administration to DBA to C# development.

I like to solve problems, IT related.

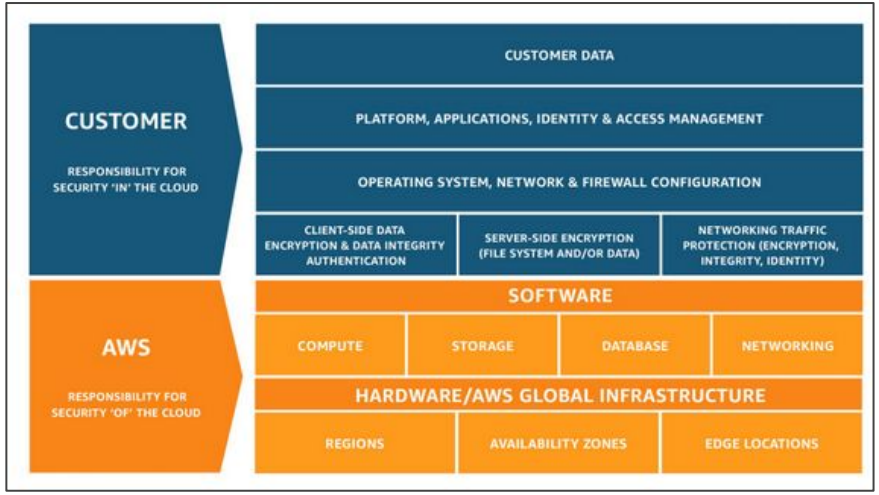
Currently the Director of Technology Operations and Security at DecisivEdge and heavily involved in Azure infrastructure, security, and DevOps.

Certs: CISA/CISSP

Shared Responsibility Models

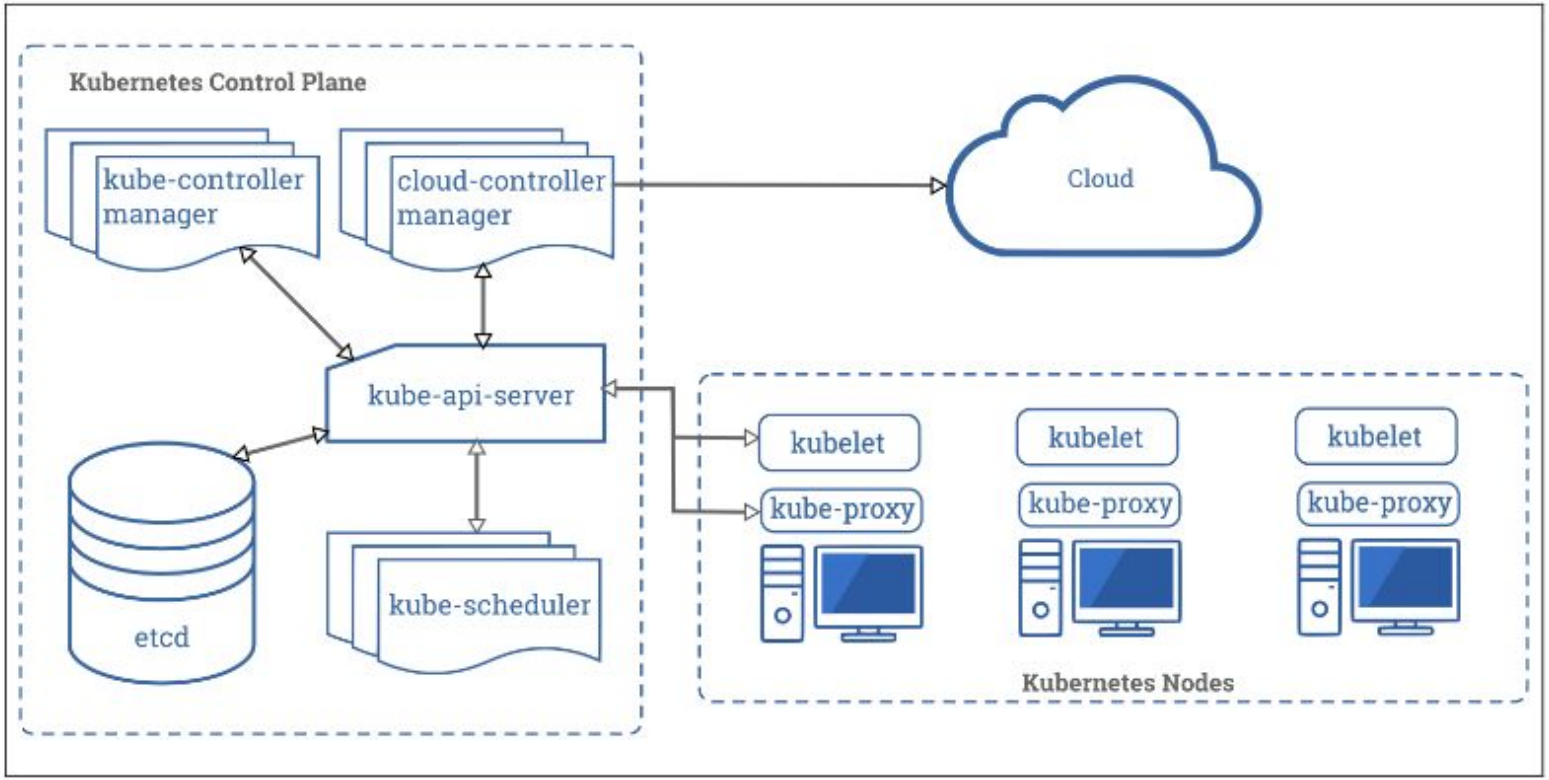


Microsoft - Azure



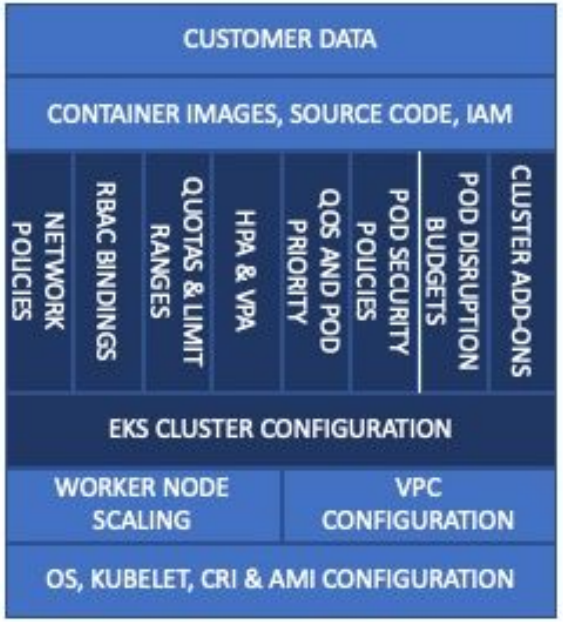
Amazon - AWS

Kubernetes Architecture



Source: NIST Kubernetes Hardening Guide Aug 2021 V1.0

Kubernetes Architecture - Nodes

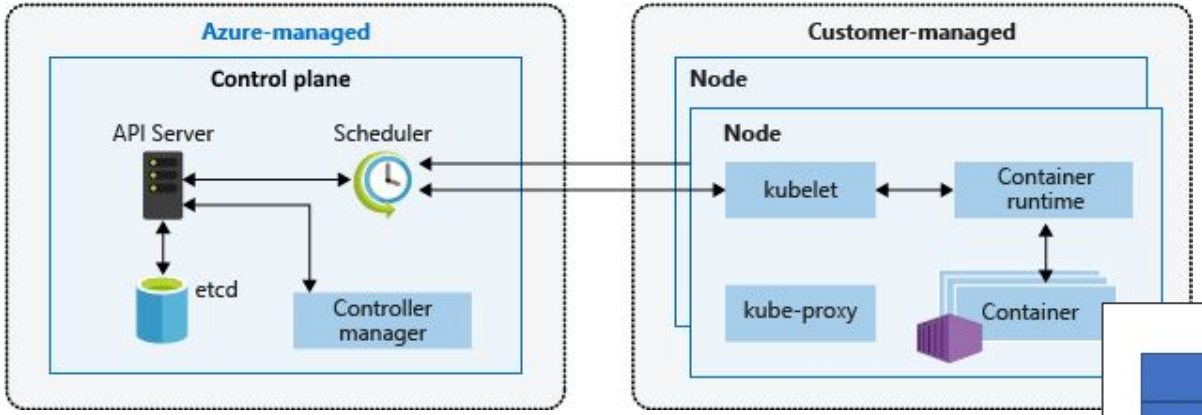


Add-Ons

- Network Policies
- DNS
- Service Mesh
- Ingress
- Etc.

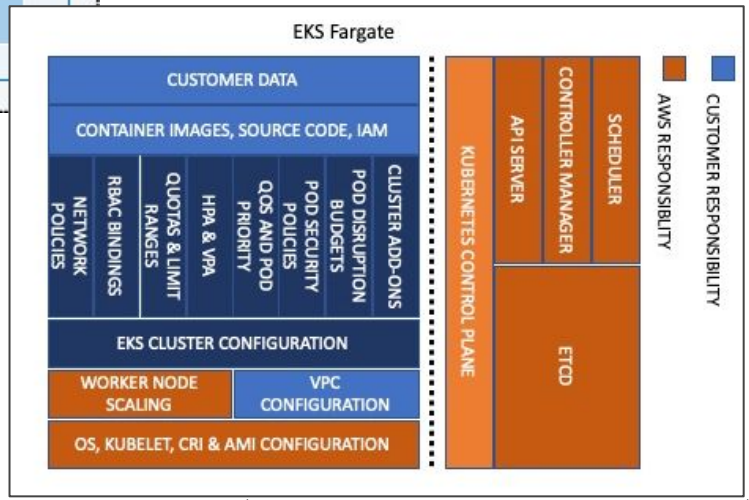
Source: AWS EKS Best Practices Guide (https://aws.github.io/aws-eks-best-practices/security/docs/)

Kubernetes In the Cloud



Microsoft - Azure

Amazon - AWS

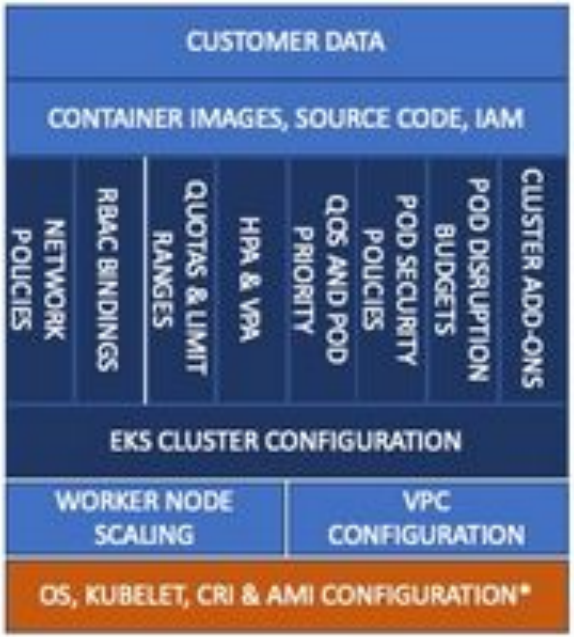


Source: AKS Docs (https://docs.microsoft.com/en-us/azure/aks/concepts-clusters-workloads)

Source: AWS EKS Best Practices Guide (https://aws.github.io/aws-eks-best-practices/security/docs/)

Customer Surface Area

- Application Security
  - Secure Coding
  - Patching
  - etc
- Access Management
- Transport Security
  - Encrypted
  - Segmented
- Backups/Redundancy
  - Alternate data locations
- Secret Management
- Patching



Kubernetes Threat Matrix (updated)

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

Source: Microsoft Security Blog (https://www.microsoft.com/security/blog/2021/03/23/secure-containerized-environments-with-updated-threat-matrix-for-kubernetes/)

Hardening Help

Sources are available, all is not lost

- NIST - Kubernetes Hardening Guidance (total platform)
- CIS - Total platform and specific provider guides
- Azure - “Day 2” Topics
- Amazon - EKS Best Practices
- Azure AKS Baseline Cluster (Azure centric/full solution concept)

Links to these documents will be in the final slide

Hardening Help - NIST

Contents

Kubernetes Hardening Guidance i
Executive summary iv
Introduction 1
- Recommendations 2
- Architectural overview 3

Threat model 5
Kubernetes Pod security 7
- "Non-root" containers and "rootless" container engines 7
- Immutable container file systems 8
- Building secure container images 8
- Pod Security Policies 10
- Protecting Pod service account tokens 11
- Hardening container engines 12

Network separation and hardening 13
- Namespaces 13
- Network policies 14
- Resource policies 14
- Control plane hardening 15
  - Etcd 16
  - Kubeconfig Files 16
- Worker node segmentation 16
- Encryption 17
- Secrets 17
- Protecting sensitive cloud infrastructure 18

Authentication and authorization 18
- Authentication 19
- Role-based access control 20

Log auditing 22
- Logging 22
  - Kubernetes native audit logging configuration 24
  - Worker node and container logging 25
  - Seccomp: audit mode 26
  - SYSLOG 27
- SIEM platforms 27
- Alerting 28
- Service meshes 29
- Fault tolerance 30
- Tools 31

Upgrading and application security practices 32

NIST main TOC

Hardening Help - CIS (Azure)

Table of Contents

Terms of Use 1
Overview 6
Intended Audience 6
Consensus Guidance 6
Typographical Conventions 7
Assessment Status 7
Profile Definitions 8
Acknowledgements 9
Recommendations 10
1 Master (Control Plane) Components 10
2 Master (Control Plane) Configuration 11
2.1 Logging 12
2.1.1 Enable audit Logs (Manual) 12
3 Worker Nodes 15
3.1 Worker Node Configuration Files 16
3.1.1 Ensure that the kubeconfig file permissions are set to 644 or more restrictive (Manual) 17
3.1.2 Ensure that the kubelet kubeconfig file ownership is set to root:root (Manual) 19
3.1.3 Ensure that the kubelet configuration file has permissions set to 644 or more restrictive (Manual) 21
3.1.4 Ensure that the kubelet configuration file ownership is set to root:root (Manual) 23
3.2 Kubelet 25
3.2.1 Ensure that the --anonymous-auth argument is set to false (Manual) 25
3.2.2 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Manual) 29
3.2.3 Ensure that the --client-ca-file argument is set as appropriate (Manual) 32
3.2.4 Ensure that the --read-only-port is secured (Manual) 35
3.2.5 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Manual) 37

3.2.6 Ensure that the --protect-kernel-defaults argument is set to true (Manual) 40
3.2.7 Ensure that the --make-iptables-util-chains argument is set to true (Manual) 43
3.2.8 Ensure that the --hostname-override argument is not set (Manual) 46
3.2.9 Ensure that the --eventRecordQPS argument is set to 0 or a level which ensures appropriate event capture (Manual) 49
3.2.10 Ensure that the --rotate-certificates argument is not set to false (Manual) 52
3.2.11 Ensure that the RotateKubeletServerCertificate argument is set to true (Manual) 55
4 Policies 58
4.1 RBAC and Service Accounts 59
4.1.1 Ensure that the cluster-admin role is only used where required (Manual) .. 59
4.1.2 Minimize access to secrets (Manual) 61
4.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual) 63
4.1.4 Minimize access to create pods (Manual) 65
4.1.5 Ensure that default service accounts are not actively used. (Manual) 67
4.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual) 69
4.2 Pod Security Policies 71
4.2.1 Minimize the admission of privileged containers (Automated) 72
4.2.2 Minimize the admission of containers wishing to share the host process ID namespace (Automated) 74
4.2.3 Minimize the admission of containers wishing to share the host IPC namespace (Automated) 76
4.2.4 Minimize the admission of containers wishing to share the host network namespace (Automated) 78
4.2.5 Minimize the admission of containers with allowPrivilegeEscalation (Automated) 80
4.2.6 Minimize the admission of root containers (Automated) 82
4.2.7 Minimize the admission of containers with the NET_RAW capability (Automated) 84
4.2.8 Minimize the admission of containers with added capabilities (Automated) 86

CIS partial TOC

Hardening Help - AWS EKS

EKS Best Practices Guides

aws/aws-eks-best-practices

EKS Best Practices Guides

Guides
- Introduction

Security
- Home
- Identity and Access Management
- Pod Security
- Multi-tenancy
- Detective Controls
- Network Security
- Data Encryption and Secrets Management
- Runtime Security
- Infrastructure Security
- Regulatory Compliance
- Incident Response and Forensics
- Image Security

Cluster Autoscaling

Reliability

Windows Containers (beta)

Introduction

Welcome to the EKS Best Practices Guides. The primary goal of this project is to offer a set of best practices for day 2 operations for Amazon EKS. We elected to publish this guidance to GitHub so we could interate quickly, provide timely and effective recommendations for variety of concerns, and easily incorporate suggestions from the broader community.

We currently have published guides for the following topics:

- Best Practices for Security
- Best Practices for Reliability
- Best Practices for Cluster Autoscaling

In the future we will be publishing best practices guidance for performance, cost optimization, and operational excellence.

Contributing

We encourage you to contribute to these guides. If you have implemented a practice that has proven to be effective, please share it with us by opening an issue or a pull request. Similarly, if you discover an error or flaw in the guidance we've already published, please submit a PR to correct it.

Table of contents

Contributing

EKS Best Practices Intro

Conclusion

- Kubernetes is complex
- Leveraging a cloud service provider eases the burden
- Understanding where the provider stops and you start is critical
- Help is readily available if you know where to look

Q & A

Contact

Ken Netzorg
knetzorg@gmail.com
zorg_the_blue (discord)

Copy of the slides will be posted to my github: zorg-the-blue

https://github.com/zorg-the-blue/talks

Document Source Links

NIST - Kubernetes Hardening Guidance (total platform)
https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/1/CTR_KUBERNETES%20HARDENING%20GUIDANCE.PDF

CIS - Total platform and specific provider guides/checklists [*Requires registration*]
https://www.cisecurity.org/benchmark/kubernetes

Azure - “Day 2” Topics
https://docs.microsoft.com/en-us/azure/architecture/operator-guides/aks/day-2-operations-guide

Amazon - EKS Best Practices
https://aws.github.io/aws-eks-best-practices/

Azure AKS Baseline Cluster (Azure centric/full solution concept)
https://github.com/mspnp/aks-secure-baseline