aws partnercast

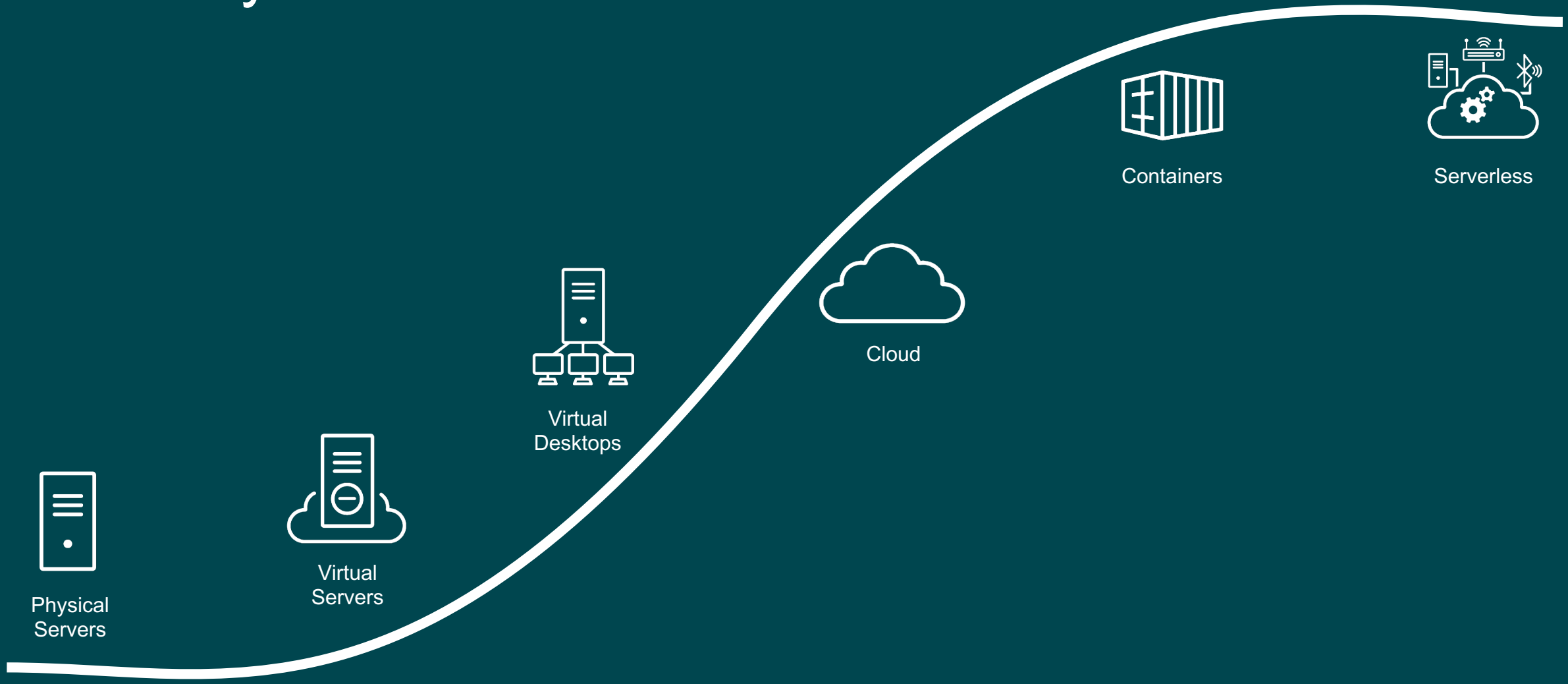
Container Security

Tejas Sheth
Sr. Security Solutions architect

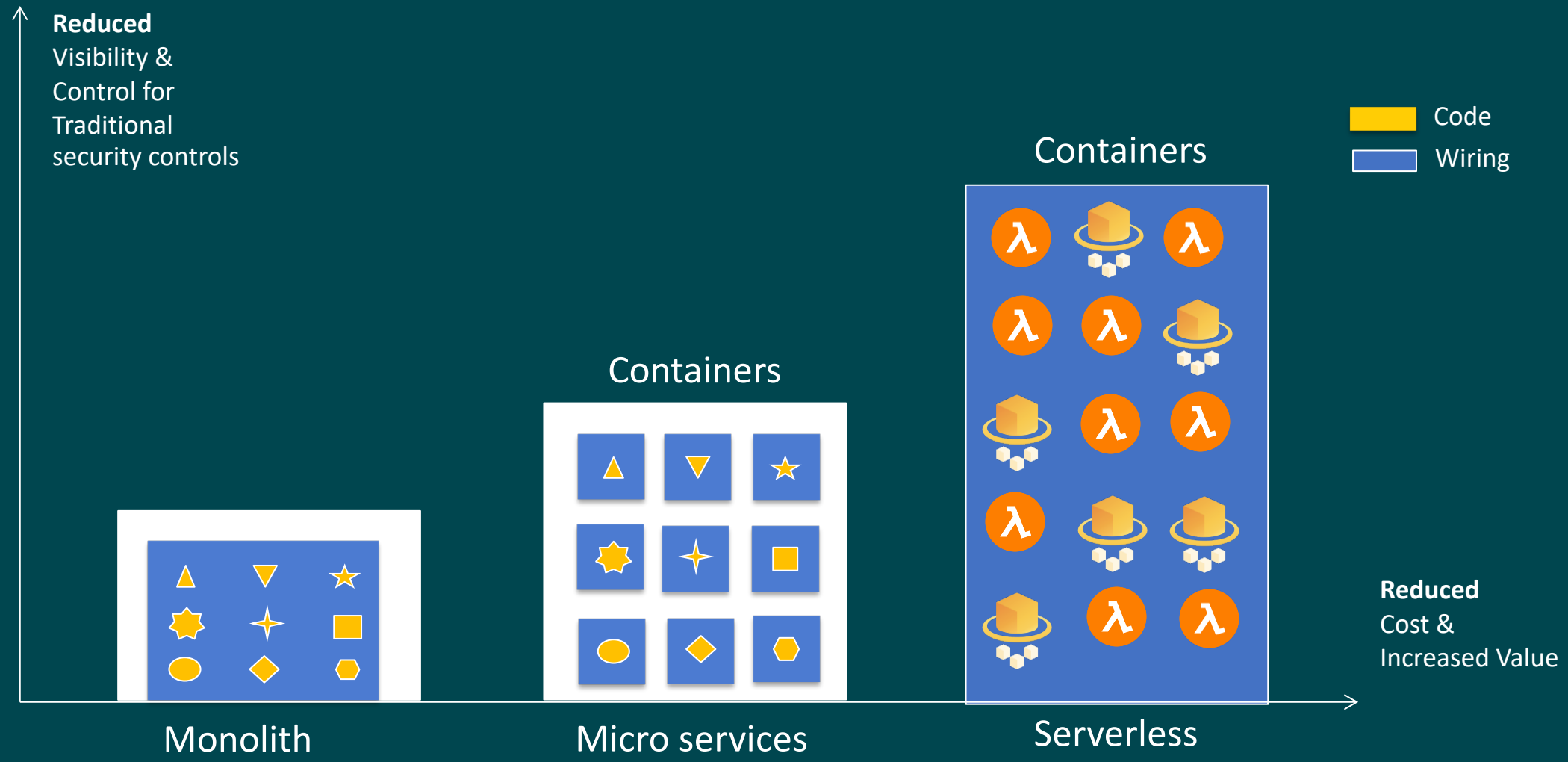
Objective

- Container security Threat vectors
- Container security controls
  - Preventive controls
    - Risk detection and remediation at early stage for SDLC
    - Risk remediation tread offs
    - Best practices
  - Protective controls
    - Threat mitigation at edge
    - Threat mitigation at VPC level
    - Threat mitigation at container level
  - Threat detection and response controls
- Dive deep on each control with container security best practice
- Security automation use cases

Journey to abstraction



Abstraction Tread-off



“Gartner estimates that 90% of global organizations will be running containerized applications in production by 2026”

[1] Gartner, The Innovation Leader’s Guide to Navigating the Cloud-Native Container Ecosystem, Arun Chandrasekaran, Wataru Katsurashima, August 18, 2021.

aws

AWS managed container services

Management

Deployment, scheduling, scaling, and management of containerized applications

Amazon Elastic Container Service

Amazon Elastic Kubernetes Service (EKS)

Hosting

Where the containers run

Amazon EC2

AWS Fargate

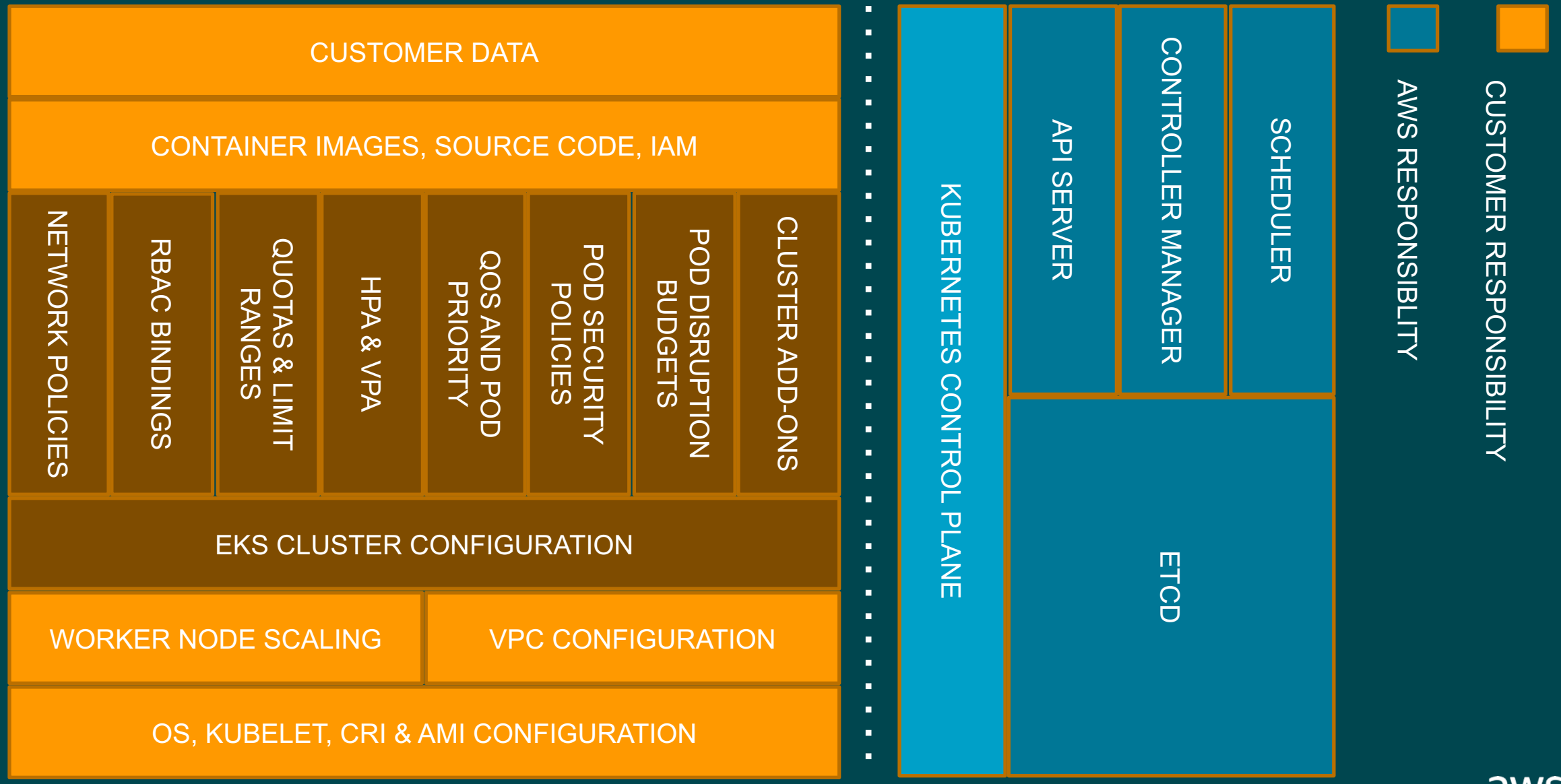
Image registry

Container image repository

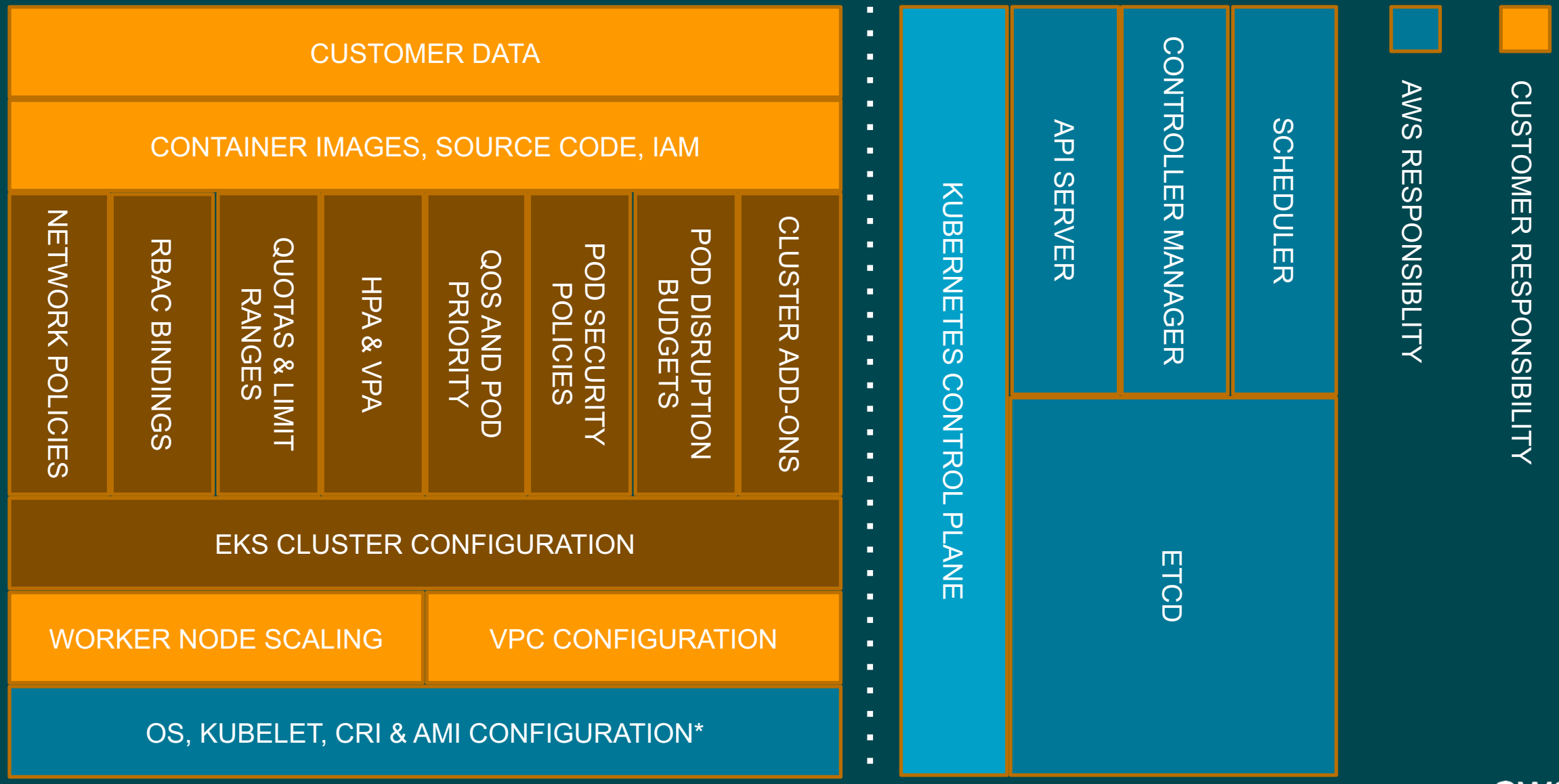
Amazon Elastic Container Registry

aws

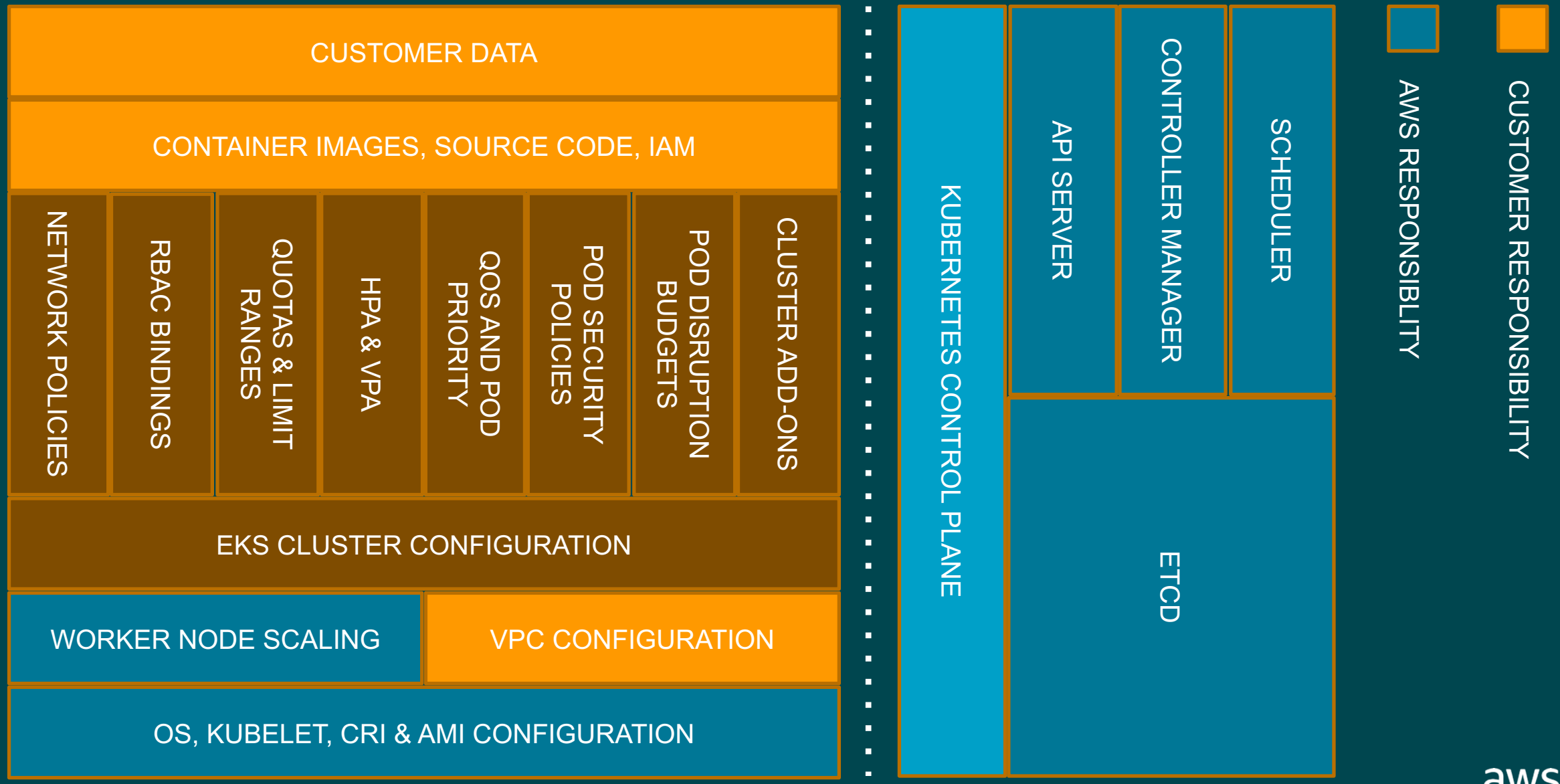
Amazon EKS with self-managed workers



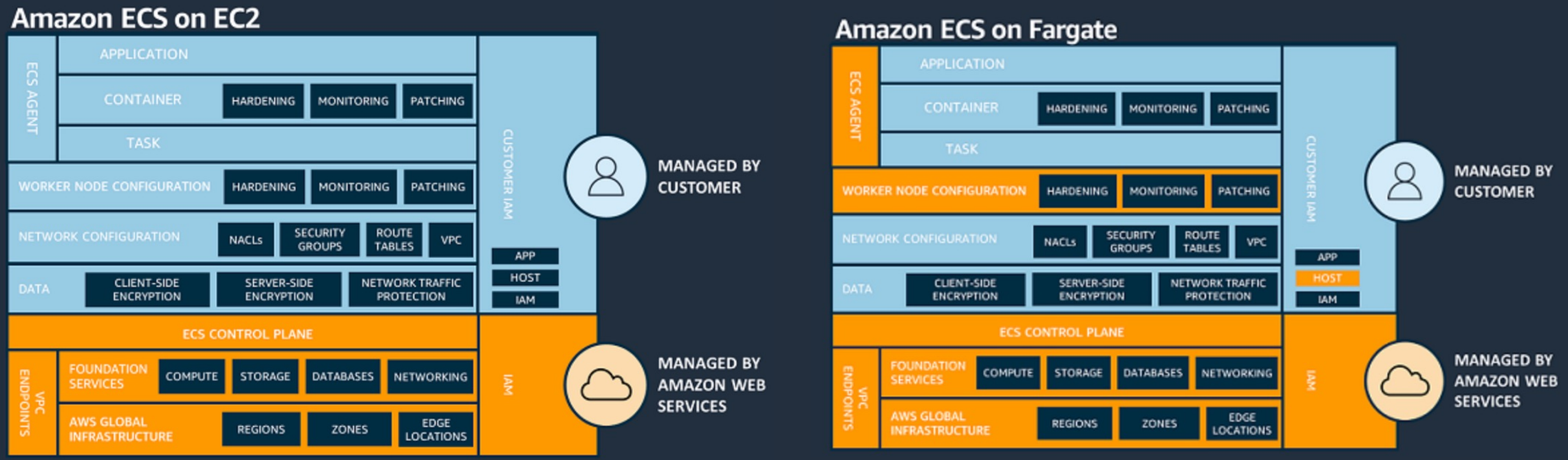
Amazon EKS with managed node groups



Amazon EKS on AWS Fargate



AWS ECS shared responsibility model



Container risks

01 Vulnerable application code

02 Poorly configured containers and pods

03 Insecure networking

04 Malware

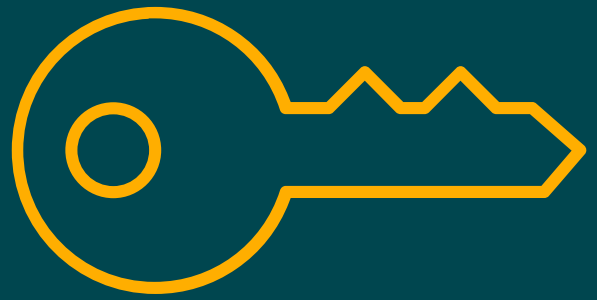
05 Unauthorized behavior

06 Secrets exposure

Are you working on a current AWS opportunity for Container security?

- Yes
- No

Security controls



Preventative

Detective

Risk mitigation techniques

Preventive controls

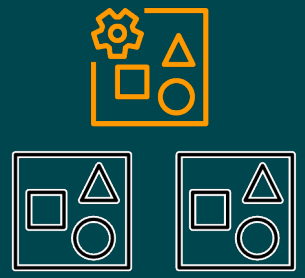
Protective controls

Threat Detection and Response

Preventive controls

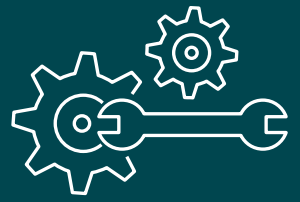
Container security foundations

Preventive controls

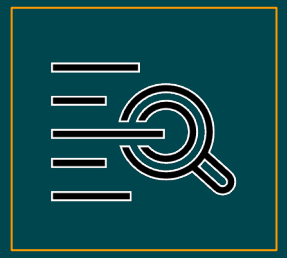


Adopt a multi account architecture

Implement the principle of least privilege



Automate tasks to save time and reduce risk

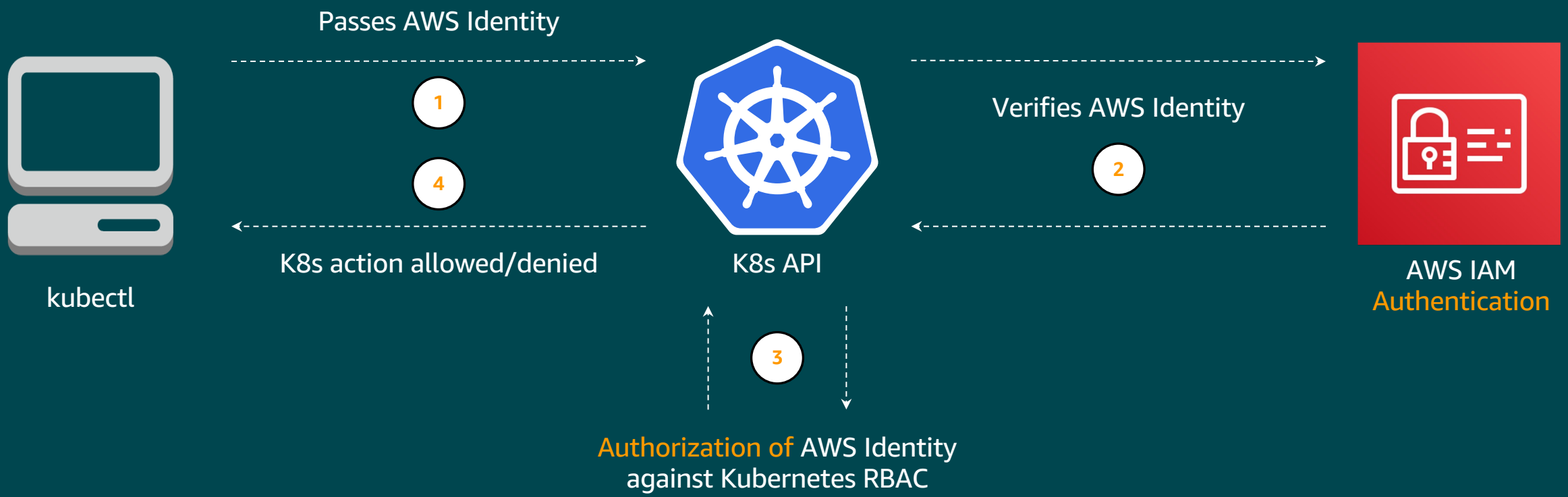


Establish a centralized logging account

Protect data in transit and at rest

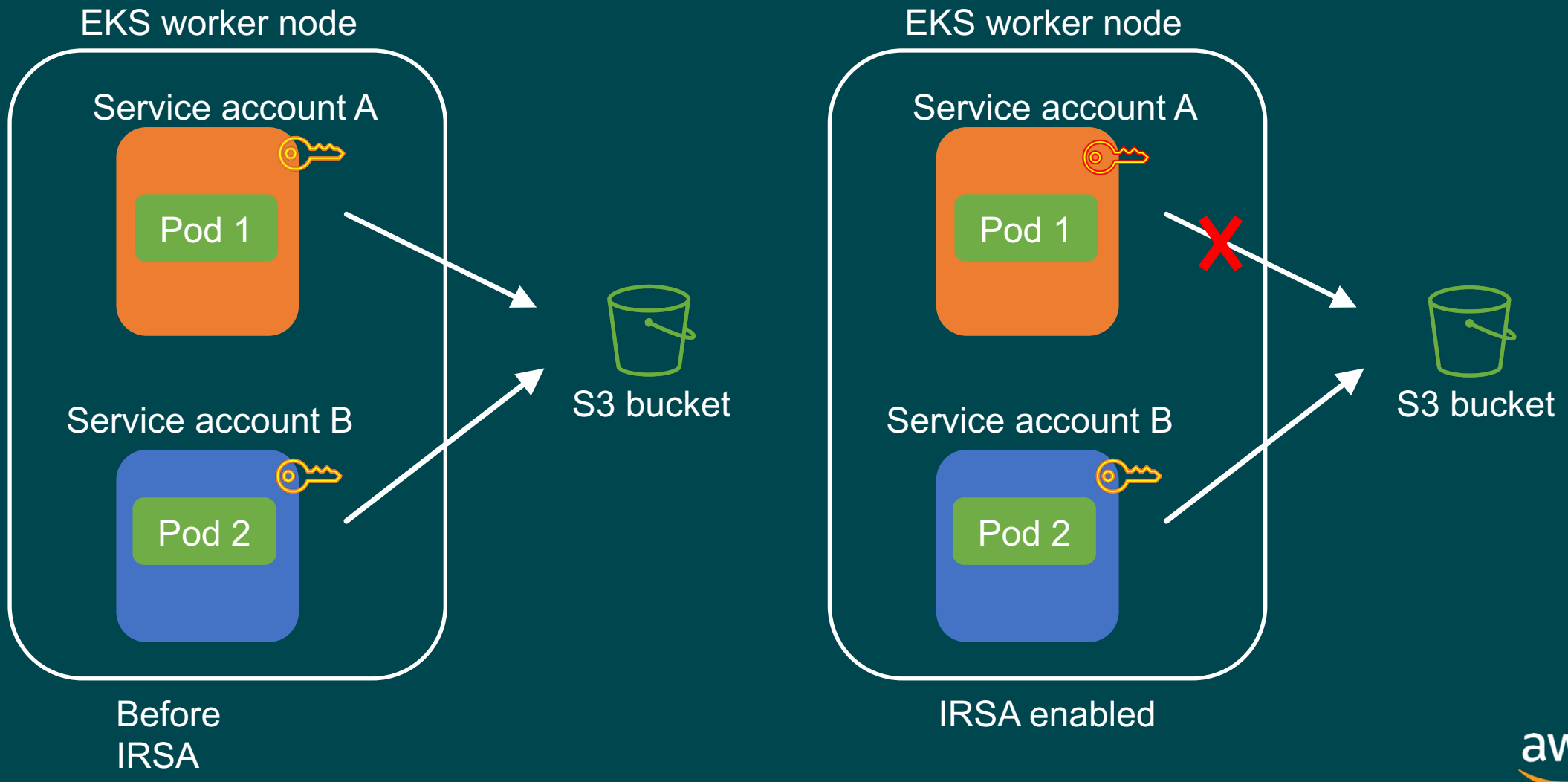
Apply security at all layers

Amazon EKS: IAM Authentication + kubectl



Container IAM – Best practices

ENABLE PERMISSION PER POD INSTEAD OF PER NODE



Network Security

VPC

- No changes to VPC (non-dedicated)
- Traffic routed through ENI to task

Security Group

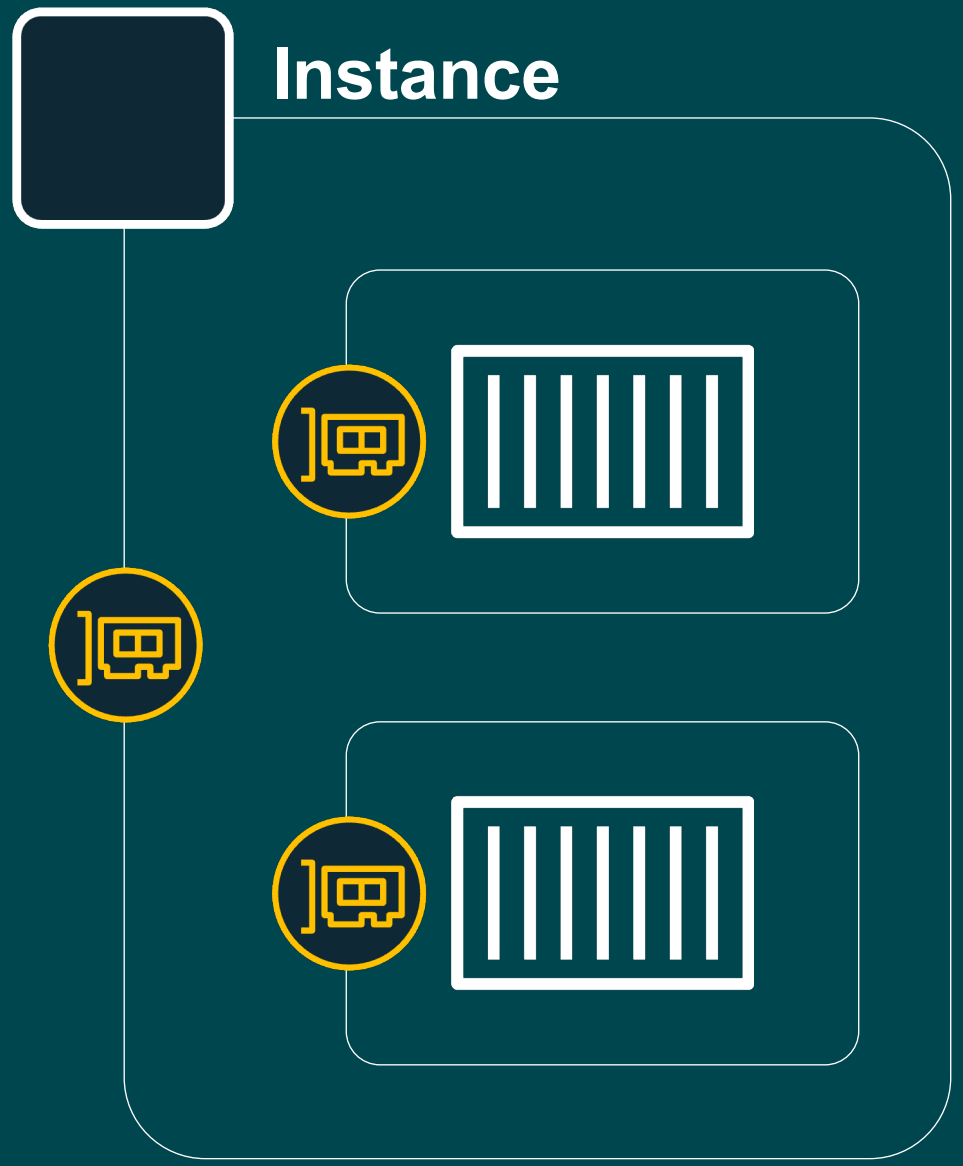
- Isolated security group per task
- Task-level specification

Task Network

- Uses CNI plugin architecture
- Isolated network namespace per task

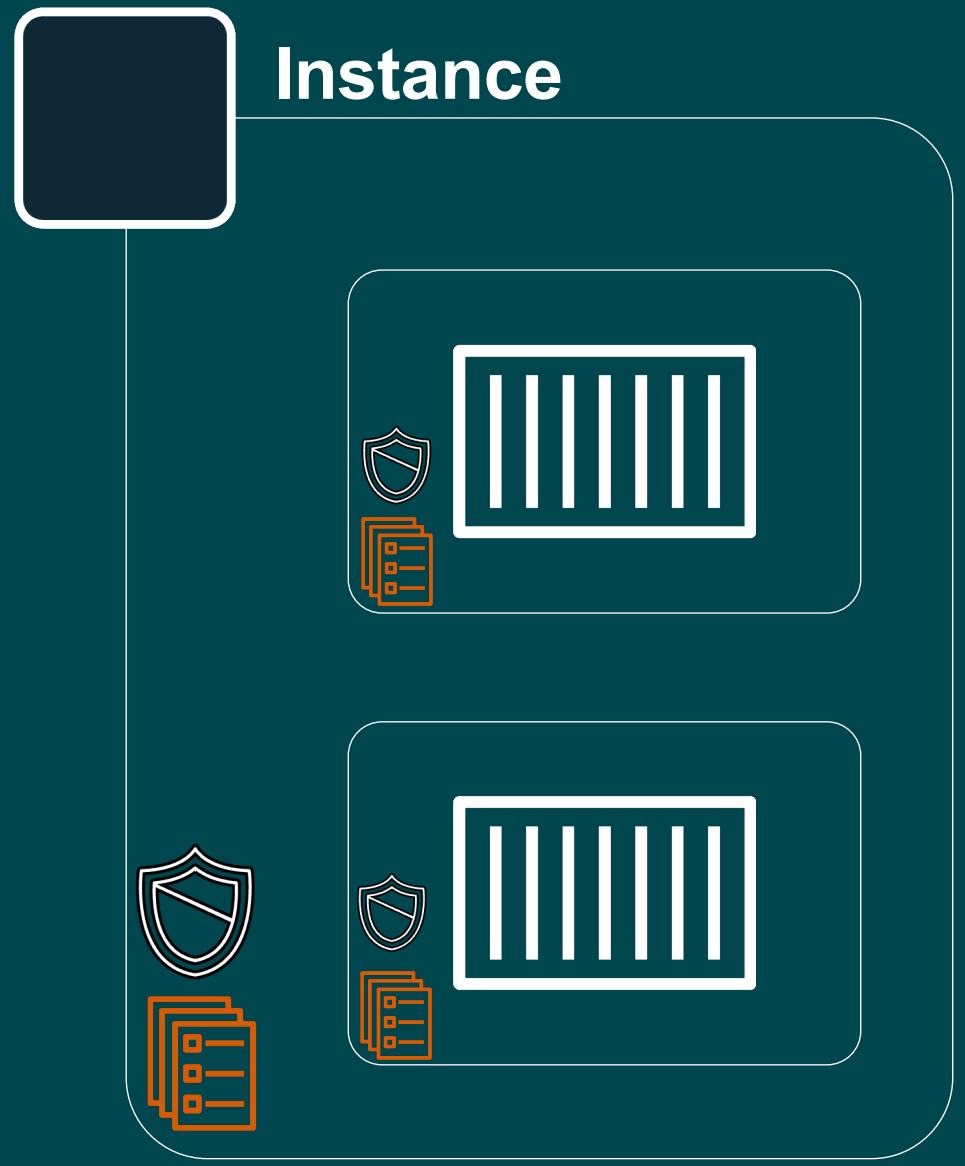
PrivateLink

- Pull images from ECR within VPC



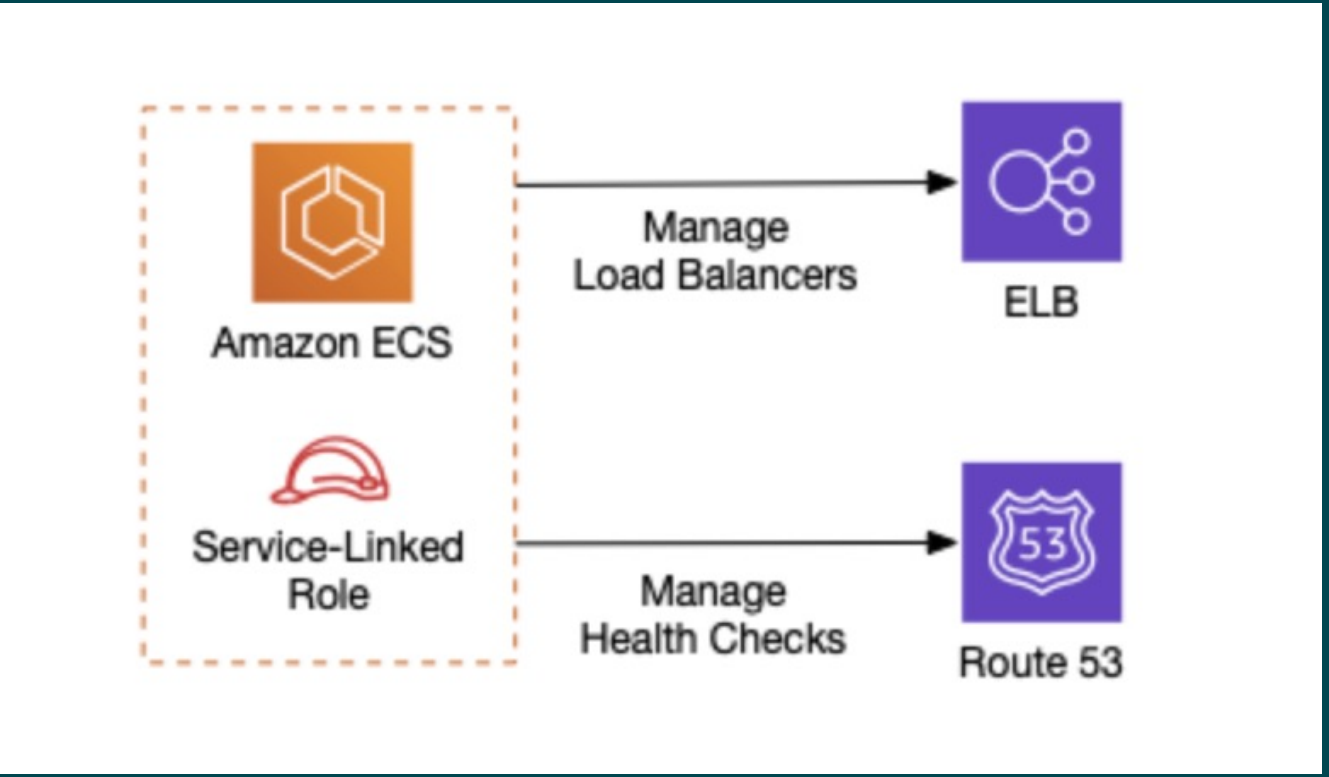
K8S Security hardening

- Define clusters based on workloads
- Kubernetes Node Affinity
- Kubernetes Pod limits quotas
- Kubernetes Network policies
- Node hardening SELinux
- Secure computing (Seccomp) to prevent unwanted system calls from container



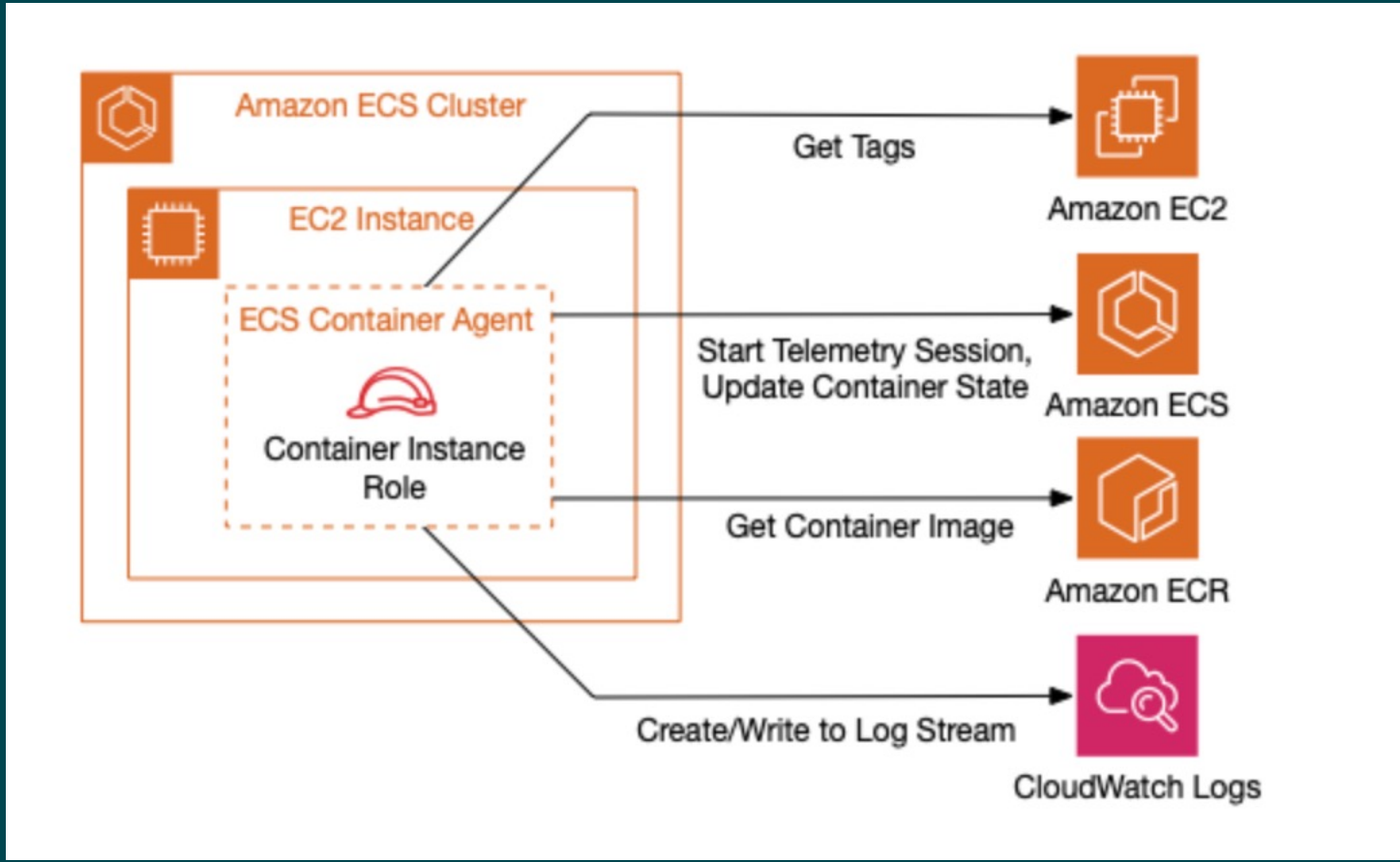
Service-Linked Role

Preventive Controls



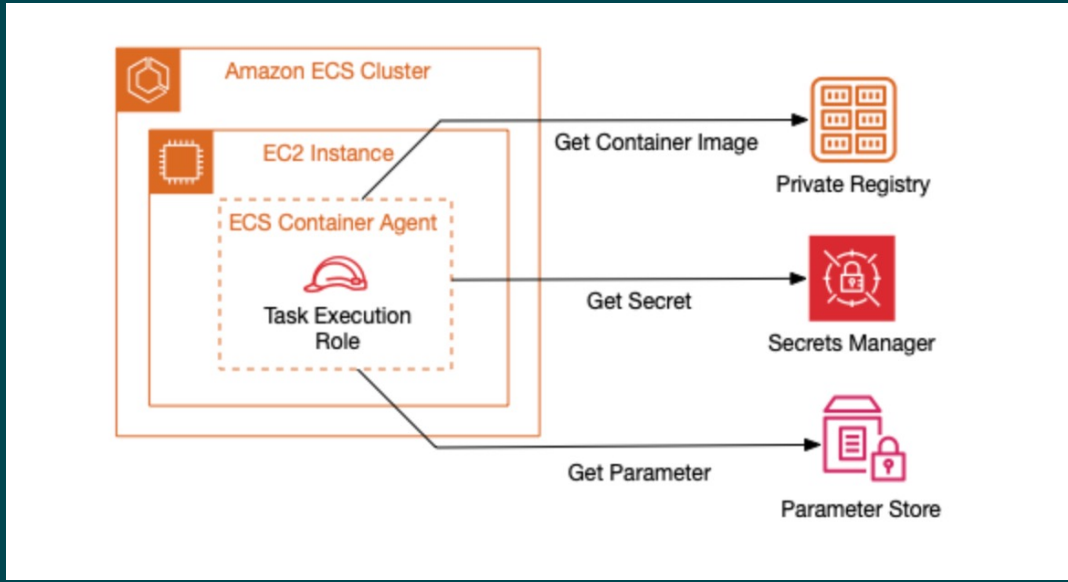
Container Instance Role

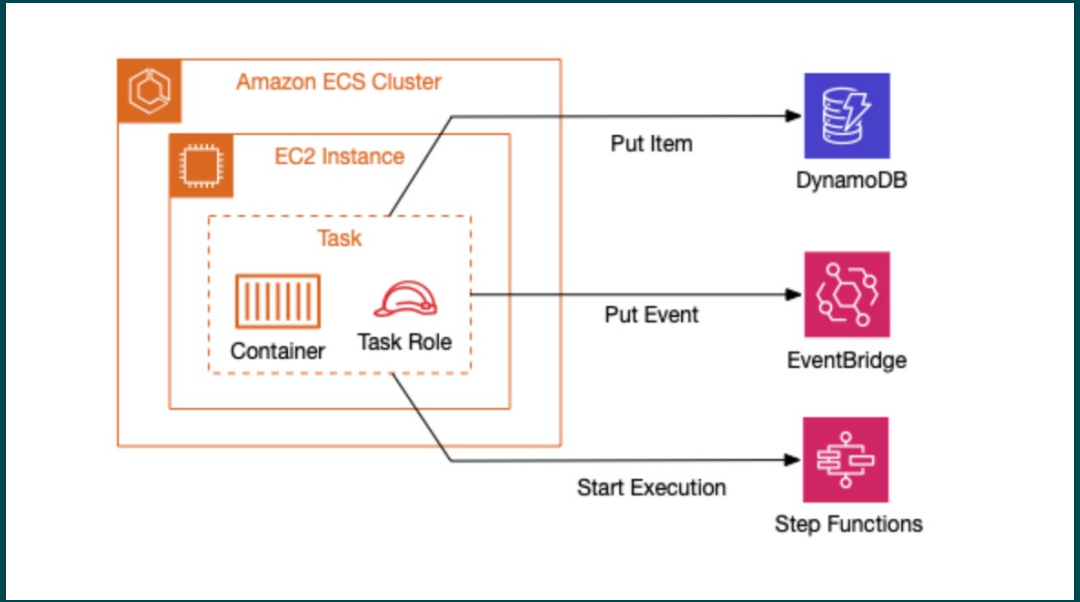
Preventive Controls



Task Execution Role

Preventive Controls

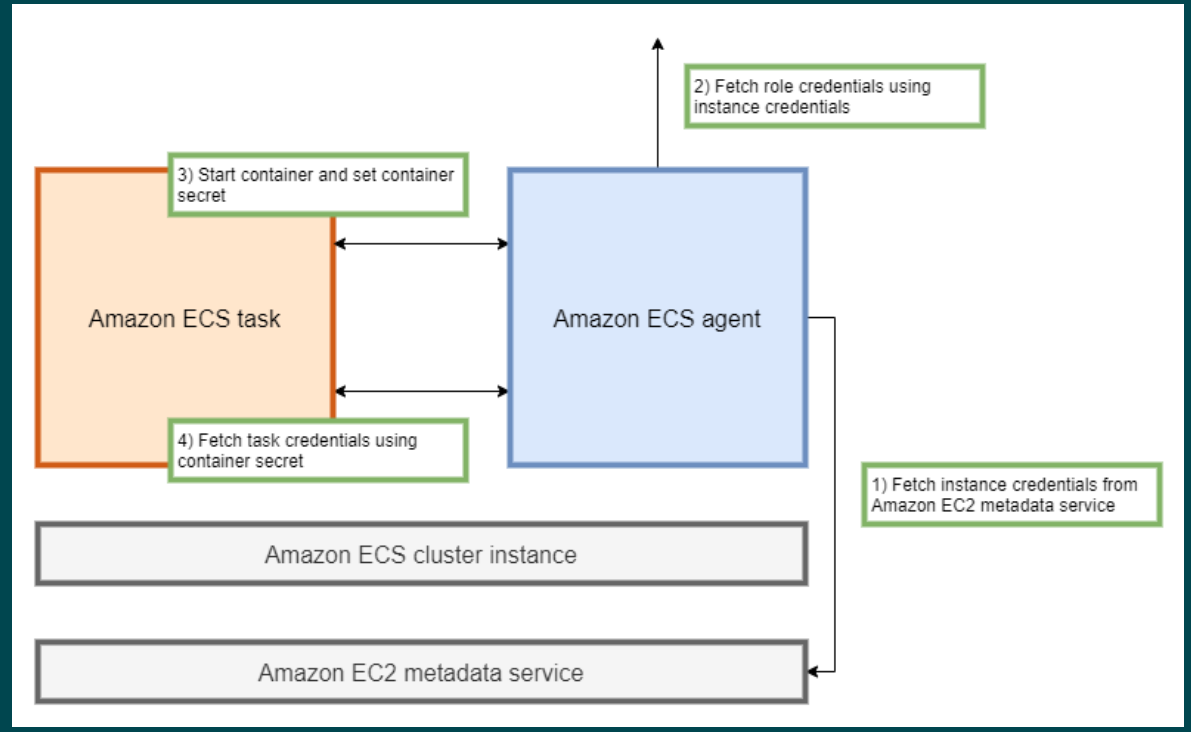




ECS Metadata services access with task role

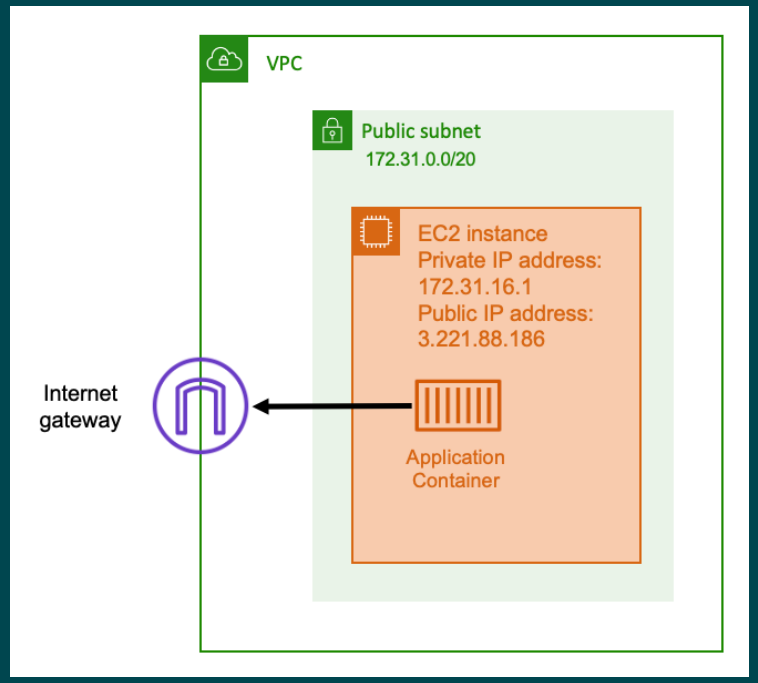
Preventive Controls

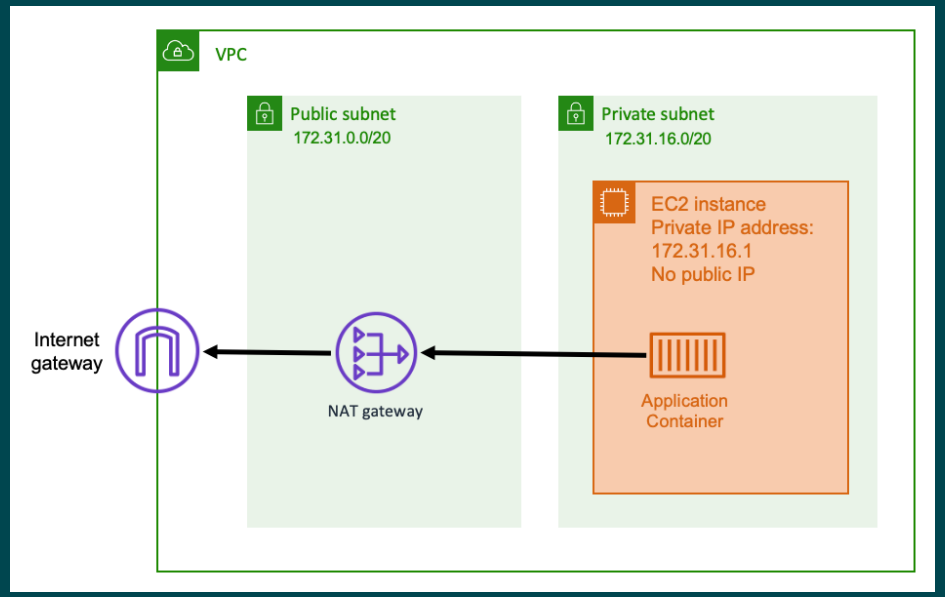
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



Connecting to the internet

Preventive Controls





Public Subnet

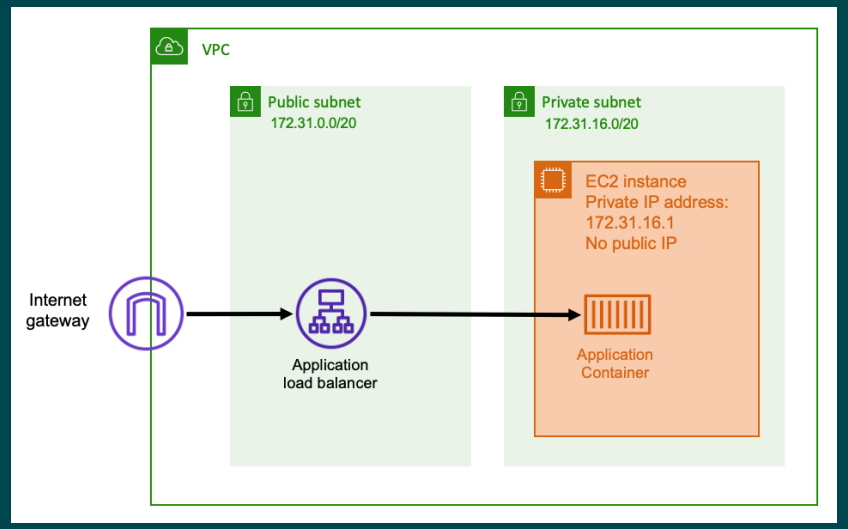
- Pay CLOSE ATTENTION to your security group and firewall rules. Make sure SSH port 22 is not open!
- Use case: public application with requirements of large amounts of bandwidth or minimal latency
- E.g., video streaming, gaming services

Private Subnet + NAT Gateway

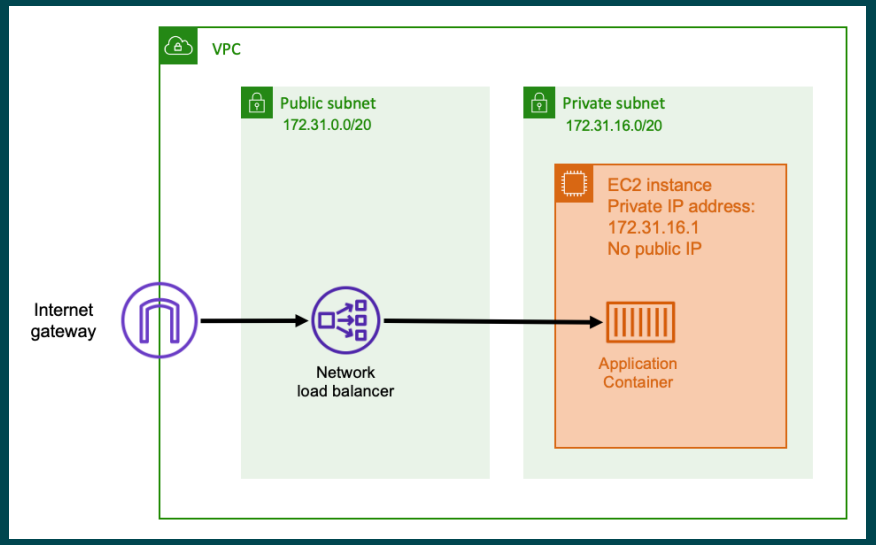
- Use case: when the application containers should be protected from external access
- E.g.: payment processing, sensitive data
- NAT gateway hourly charge applies

aws

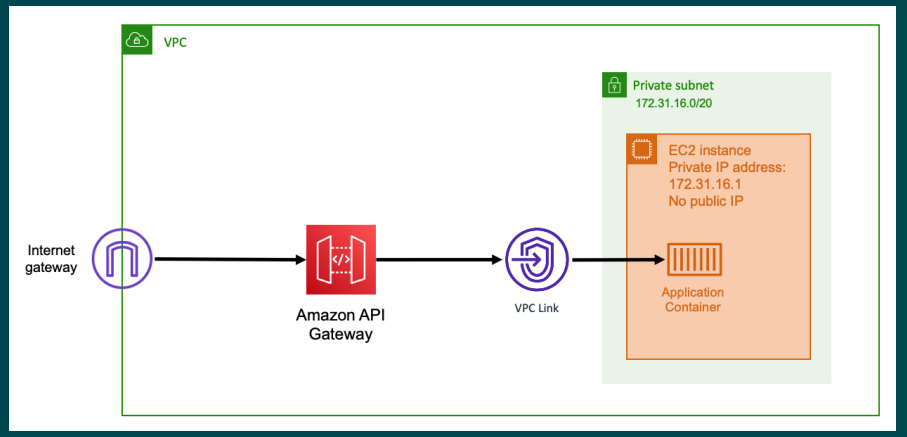
Receiving inbound connection from the internet



Application Load Balancer - for public HTTP services



Network Load Balancer – for non-HTTP protocols

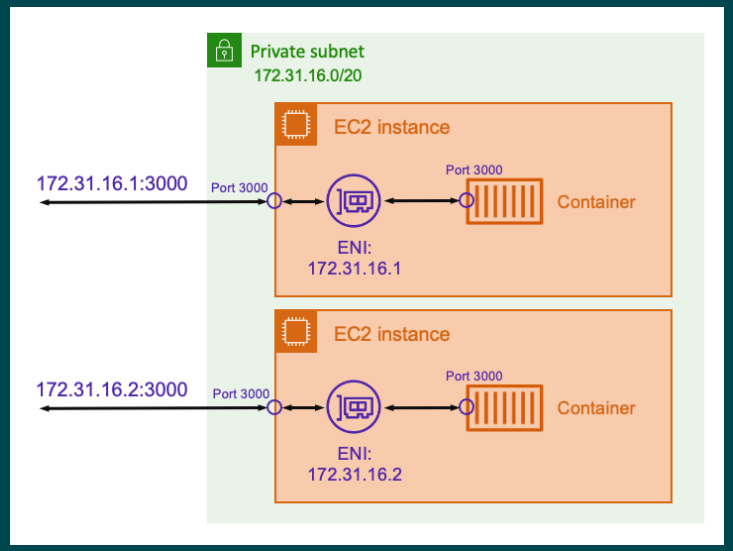


Amazon API Gateway - for HTTP applications with sudden bursts or low request volumes

aws

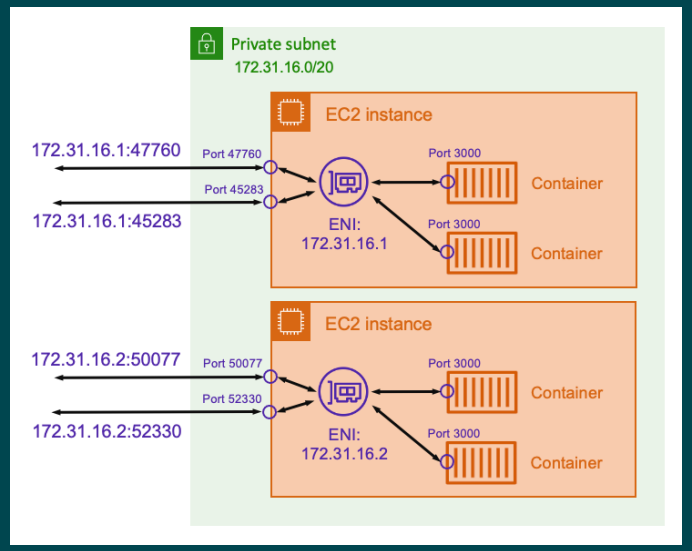
Choosing a network mode

Preventive Controls



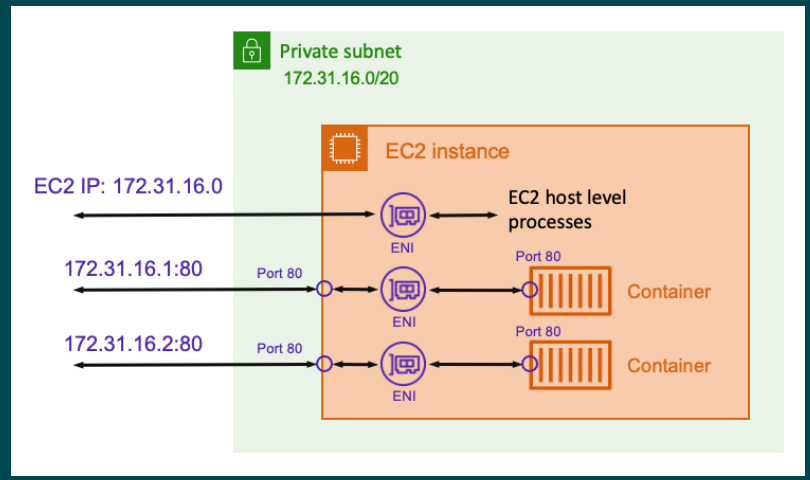
`host` mode

- container networking directly tied to the underlying host
- cannot run more than 1 task on an instance
- cannot remap port
- NOT RECOMMENDED



`bridge` mode

- a virtual network bridge between host and container networking
- supports dynamic port mapping
- difficult to lock down service to service communication
- not supported by ECS Fargate

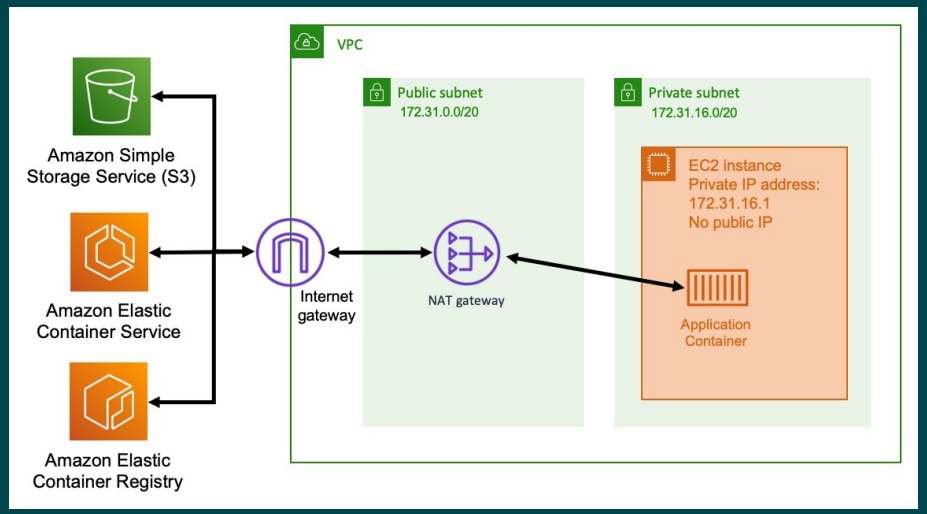


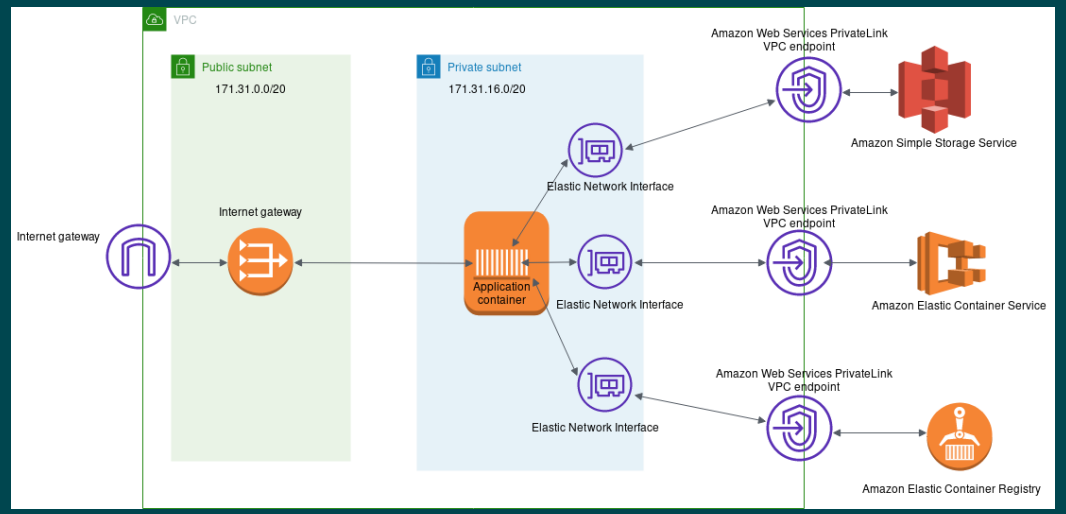
`awsvpc` mode

- each task gets its own IP
- tasks can bind to the same port
- better security group granularity
- IPv6 dual stack VPC compatible
- required when using Fargate
- considerations:
  - increase task density by ENI trunking
  - prevent IP address exhaustion

Connecting to AWS services from inside the VPC

Preventive Controls





NAT Gateway

- go through the public internet
- disadvantages
  - no control over outbound destinations
  - NAT gateway bandwidth limit

VPC Endpoint (AWS PrivateLink)

- Common
- private connectivity: traffic doesn't leave the AWS network

https://docs.aws.amazon.com/config/latest/developerguide/ecs-containers-nonprivileged.html

aws

IAM best practice with ECS

Preventive Controls

- Block access to EC2 Metadata

sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface docker+ --destination 192.0.2.0/32 --jump DROP

- Use “awsvpc” mode

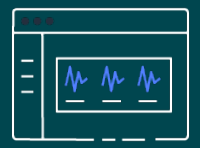
- Use IAM access advisor to refine roles

aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/ExamplePolicy1

- Monitor AWS Cloudtrail for any suspicious activity

Amazon Inspector

Preventive Controls



Gain centralized visibility

- Environment coverage
- High impact findings
- Resources by finding severity

One-click continuous monitoring

- Automatic discovery of resources
- Monitors throughout the resource life-cycle

Prioritize with contextualized scoring

- Inspector Risk score
- Security metrics
- Customized views

Centrally manage at scale

- AWS Organizations
- Package vulnerability, Network reachability
- Environment coverage

Automate and take actions

- Management APIs
- Detailed findings in Amazon EventBridge
- Security Hub integration

Prioritize vulnerability remediation

Types of vulnerability

Remotely exploitable

Remote Command/Code Execution
SQL Injection
XSS
CSRF
Partial DDoS

Locally exploitable

Buffer overflow
Privilege escalation
Arbitrary execution
File and memory view as root
System-wide DoS

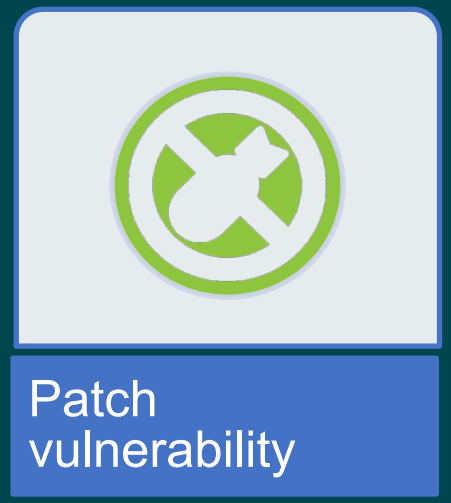
Vulnerability management

Reduce attack surface

Patch vulnerability

Security Tools

- Choose base image based wisely
  - Ubuntu vs Alpine (choose based on least vulnerability found)
  - Footprint and community response
- Remove Bash (if not required)
- Use apt-get command with --no-install-recommends
- Build image from scratch with least amount of packages only required by OS
- Use low level build tools like “buildah” for least attack surface

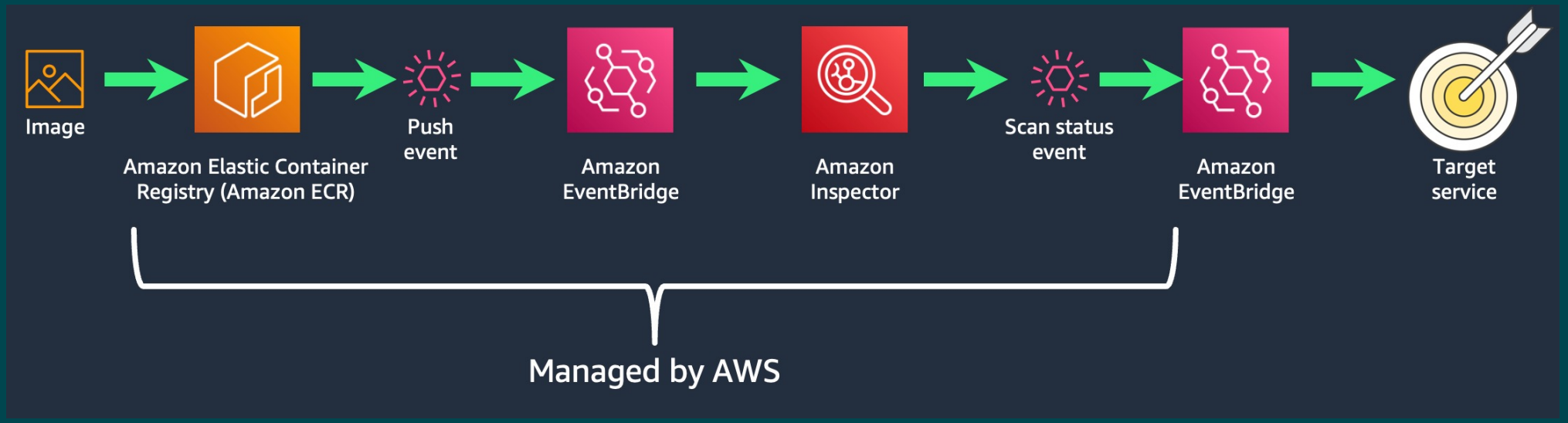


- Choose way to deploy application
  - Package manager
    - Use apt-get, yum, pip, npm etc. to update packages
  - Install from source
    - Most flexible and complex way to update dependent packages.

- Protect at Host, container and application
  - IPS (intrusion prevention system) for virtual patch
  - Malware detection
  - Support not just for Kubernetes but all containers
  - Admission control to reduce attack surface
  - RASP for protecting application inside out

Prevent vulnerable image to be deployed in production

Preventive Controls



- Use Amazon Inspector to find vulnerabilities in container images in elastic container registry and operating systems hosting containers on Amazon EC2
- Inspector leverages a continuous vulnerability scanning model to enable customers to support automated vulnerability analysis across all workloads

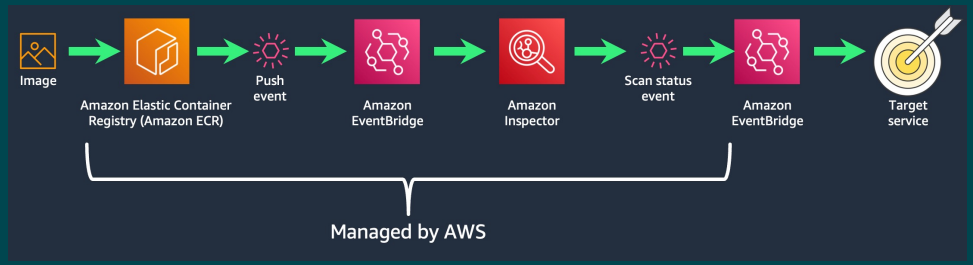
Other measures during build time

Preventive controls

Add Static code analysis in CI/CD pipeline

Add Container image scanning in CI/CD pipeline

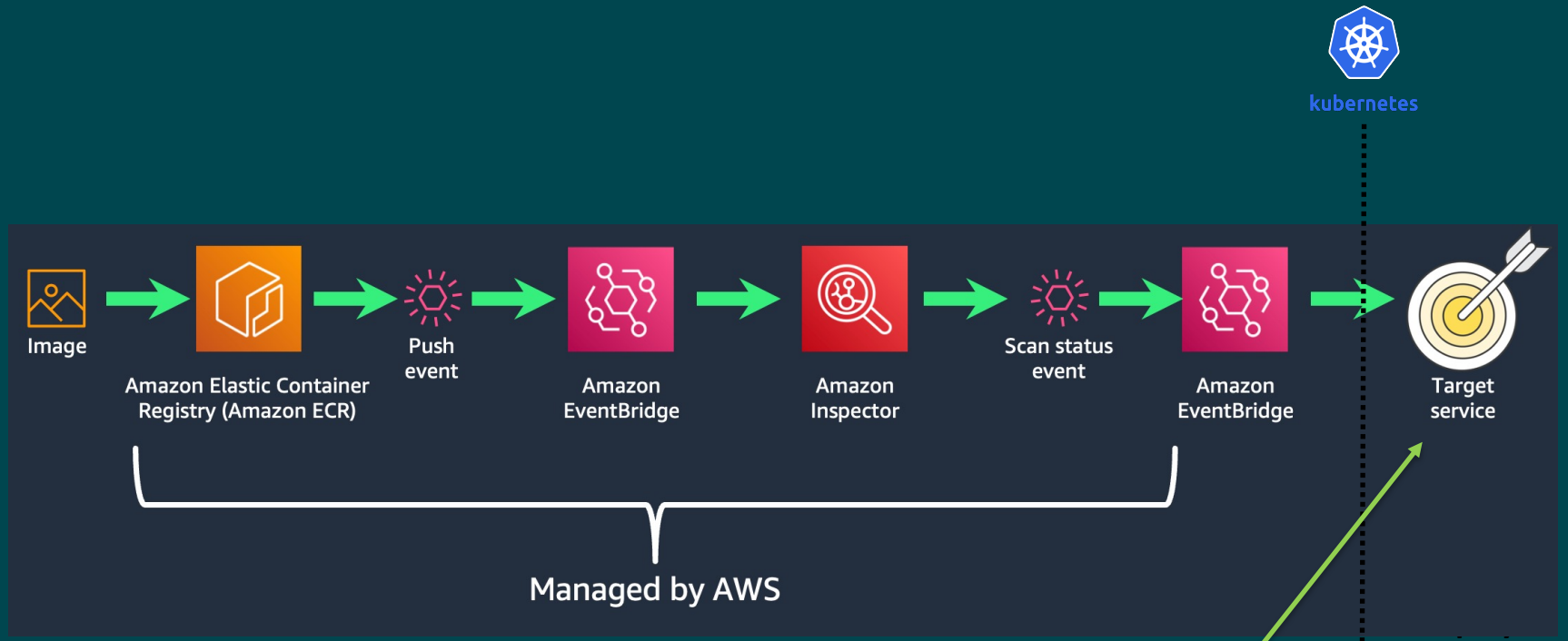
- Find Secrets stored in Image
- Verify if all images have “update” statement.
- Secure DevOps tool chain
- Vulnerabilities risk at each layer in image
- Check for malicious payload in image
- Check for compliance in image (hardening best practice)
- Sign approved images at pipeline



Ensuring your Images are safe for runtime

Escape from DevOps pipeline

Preventive controls



```
dmathieu/> kubectl get pods --watch
NAME                                   READY   STATUS              RESTARTS   AGE
static-app-deployment-584d9757f5-4mxfq   1/1     Running             0          2m
static-app-deployment-584d9757f5-bhwhr   1/1     Running             0          1m
static-app-deployment-584d9757f5-fm4f4   1/1     Running             0          1m
static-app-deployment-584d9757f5-fwknf   1/1     Running             0          2m
static-app-deployment-584d9757f5-gbdtj   1/1     Running             0          1m
static-app-deployment-584d9757f5-n7fj4   1/1     Running             0          2m
static-app-deployment-584d9757f5-pnjgs   1/1     Running             0          1m
static-app-deployment-584d9757f5-r689q   1/1     Running             0          1m
static-app-deployment-584d9757f5-tctk5   1/1     Running             0          56s
static-app-deployment-584d9757f5-zwt7x   1/1     Running             0          1m
static-app-deployment-5f6959d966-8lbqz   0/1     Pending             0          0s
static-app-deployment-5f6959d966-8lbqz   0/1     Pending             0          0s
static-app-deployment-5f6959d966-8lbqz   0/1     ContainerCreating   0          0s
```

Curse of privileged containers

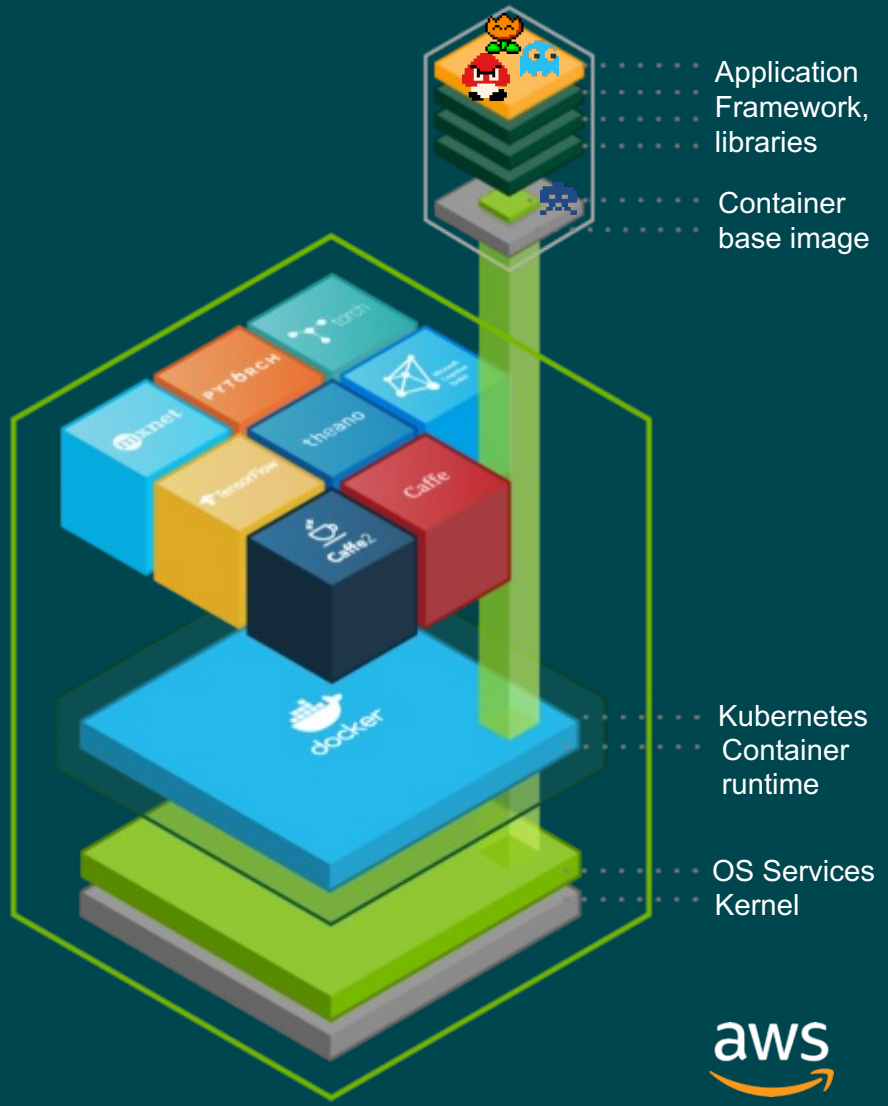
Preventive Controls

Container that is owned by Root

Vulnerable unpatched container platform

Container running in Host namespace

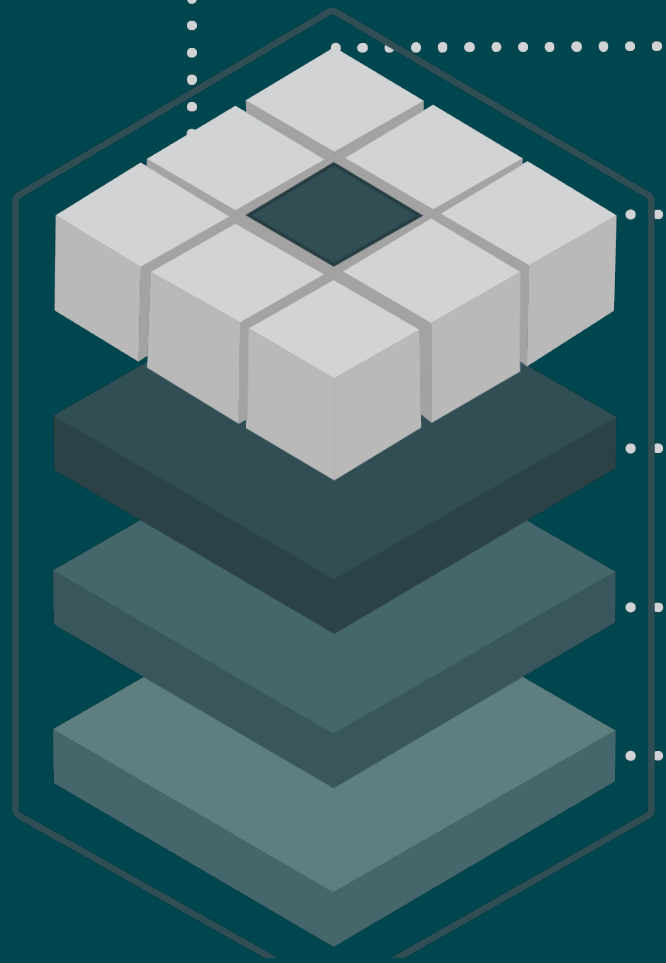
Containers with privilege escalation enabled in K8S



aws

Admission Control policy at runtime

Preventive Controls



- Policy for prevent non-compliant runtime configuration
- Policy for approved source of container registry
- Container image signing for approved SBOM
- Read-only container filesystem
- Container node Security posture (e.g. CIS benchmark)
- Select Container Optimized AMI

Do your customer ask you if their Container workload is secured and what are the ways to protect this containers ?

- Always
- Sometimes
- Never

Encryption in Transit using ACM PCA

Preventive Controls

ACM Private CA Kubernetes Connector

- Easily automate and configure end-to-end encryption for Amazon Elastic Kubernetes Service (EKS)
- Generate and manage private certificates for TLS and mTLS for Kubernetes clusters
- Simplify regulatory and compliance requirements, such as PCI-DSS and HIPAA, for data in-transit

Ingress Controllers and Load Balancers

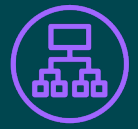
Ingress controllers are a way for you to intelligently route HTTP/S traffic that emanates from outside the cluster to services running inside the cluster.

Ingresses are often fronted by a Network Load Balancer (NLB) or an Application Load Balancer (ALB)

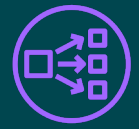
Encrypted traffic can be terminated at different places within the network:

- The pod – Supports end-to-end encryption, but places additional burden on your Pod to establish TLS handshakes
- The load balancer – Reduces Pod compute cycles needed to establish TLS handshake
- Ingress resource – Reduces Pod compute cycles needed to establish TLS handshake

Elastic Load Balancing



Application Load Balancer



Network Load Balancer

aws

Encrypting Kubernetes Secrets

Preventive Controls

There are two primary options for protecting secrets in EKS:

1. **Use AWS KMS for envelope encryption of Kubernetes secrets**

2. **[Recommended] Use an external secrets management service – AWS Secrets Manager or a partner solution from the AWS Marketplace**

AWS KMS

AWS Secrets Manager

aws

Secrets on EKS: Envelope Encryption

Envelope Encryption on KMS – Key features

- The KMS plugin for Kubernetes allows you to encrypt your secrets with a unique data key generated by AWS KMS
- The data key is then encrypted using a KMS key which is non-exportable and cannot leave the hardware security modules that back KMS
- KMS keys are rotated on a recurring schedule (if rotation is enabled)
- Using the KMS plugin for Kubernetes, all secrets are stored (encrypted) in etcd and can only be decrypted by the Kubernetes API server

Secrets on EKS: AWS Secret Manager

Allows for automated secrets rotation

Kubernetes doesn't automatically rotate secrets. If you have to rotate secrets, consider using an external secret store, e.g. AWS Secrets Manager.

Improved security over default Kubernetes secrets management

AWS Secrets Manager offers features such as fine-grained access controls, encryption by default (cannot be disabled), and automatic rotation of secrets. None of these capabilities are available with default Kubernetes secrets management.

Secret Store CSI Driver for direct integration with AWS services

The Secret Store CSI Driver is a community project that uses the CSI driver model to fetch secrets from external secret stores.

The AWS provider for the Secret Store CSI Driver supports:

- AWS Secrets Manager & AWS Systems Manager Parameter Store
- Rotation of expired secrets
- Synchronization of AWS Secrets Manager secrets to Kubernetes Secrets - synchronization can be useful when you need to reference a secret as an environment variable instead of reading them from a volume

Additional best practices for secrets

Use separate namespaces as a way to isolate secrets from different applications

If you have secrets that cannot be shared between applications, create a separate namespace for those applications.

For example: corp/secrets/accounting vs. corp/secrets/engineering

Use volume mounts instead of environment variables

The values of environment variables can unintentionally appear in logs. Secrets mounted as volumes are automatically removed from the node when the pod is deleted.

Audit the use of Kubernetes Secrets

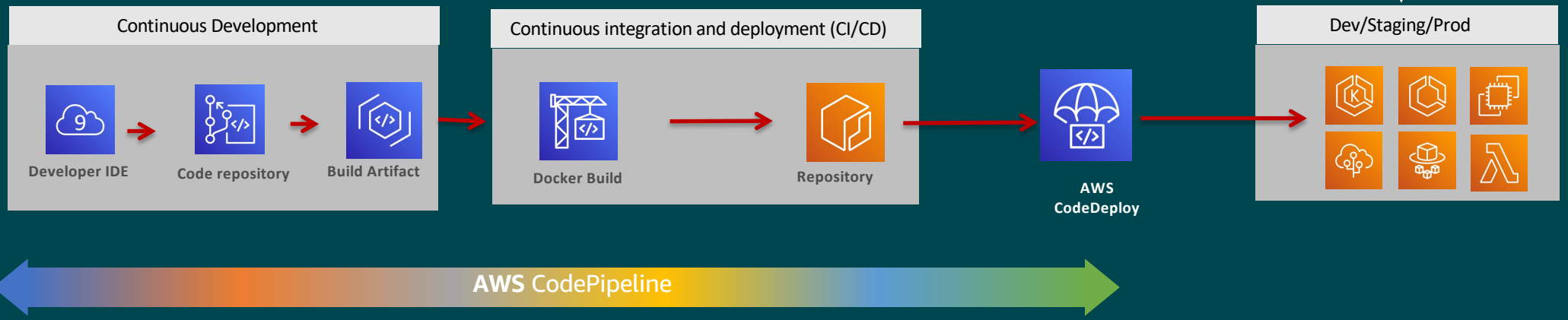
Turn on EKS audit logging and create a CloudWatch metrics filter and alarm to alert you when a secret is used

Preventative

Amazon CodeGuru
AWS Cloud9 IDE
AWS CodeArtifact
AWS CodeCommit

Amazon Inspector
AWS CodeBuild
AWS ECR

- AWS Certificate manager
- AWS Key Management service
- AWS Secrets manager
- AWS IAM
- AWS Systems manager
- AWS PrivateLink

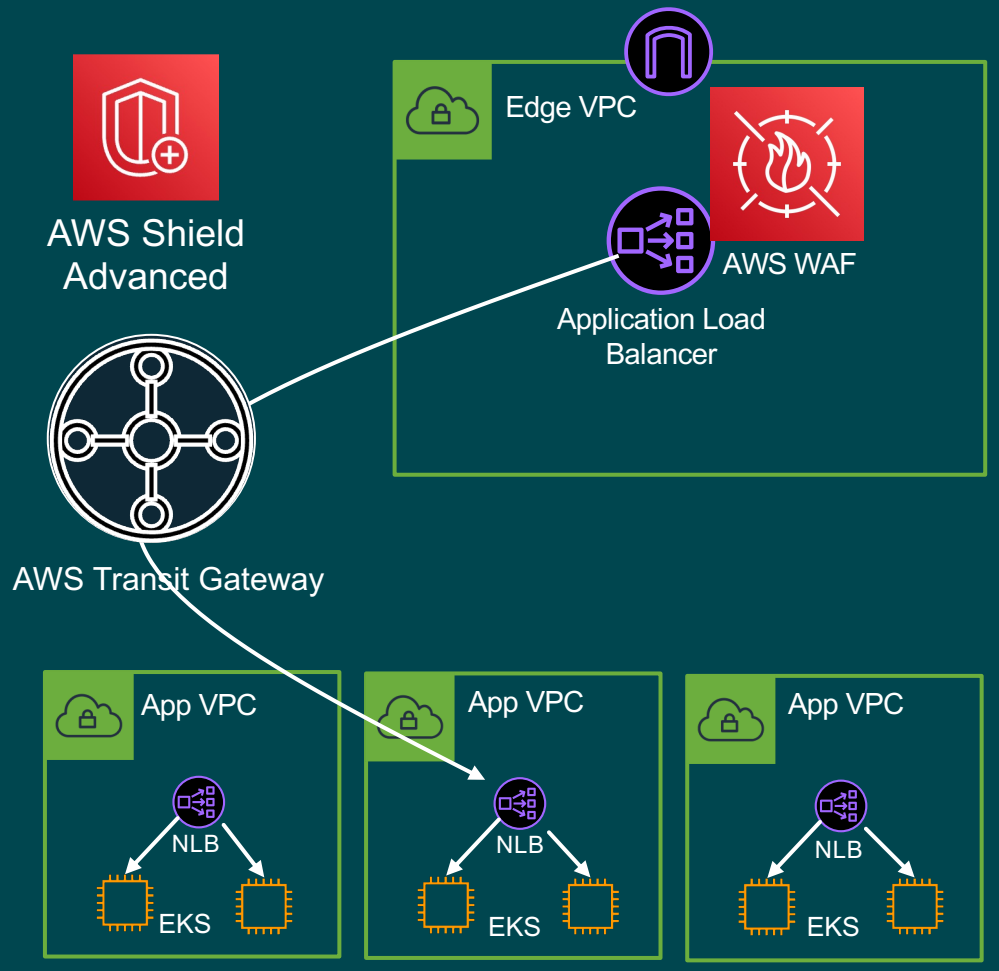


Demo

Protective controls

Protect container-based applications from attacks

Edge Protection with AWS Shield Advanced and AWS WAF

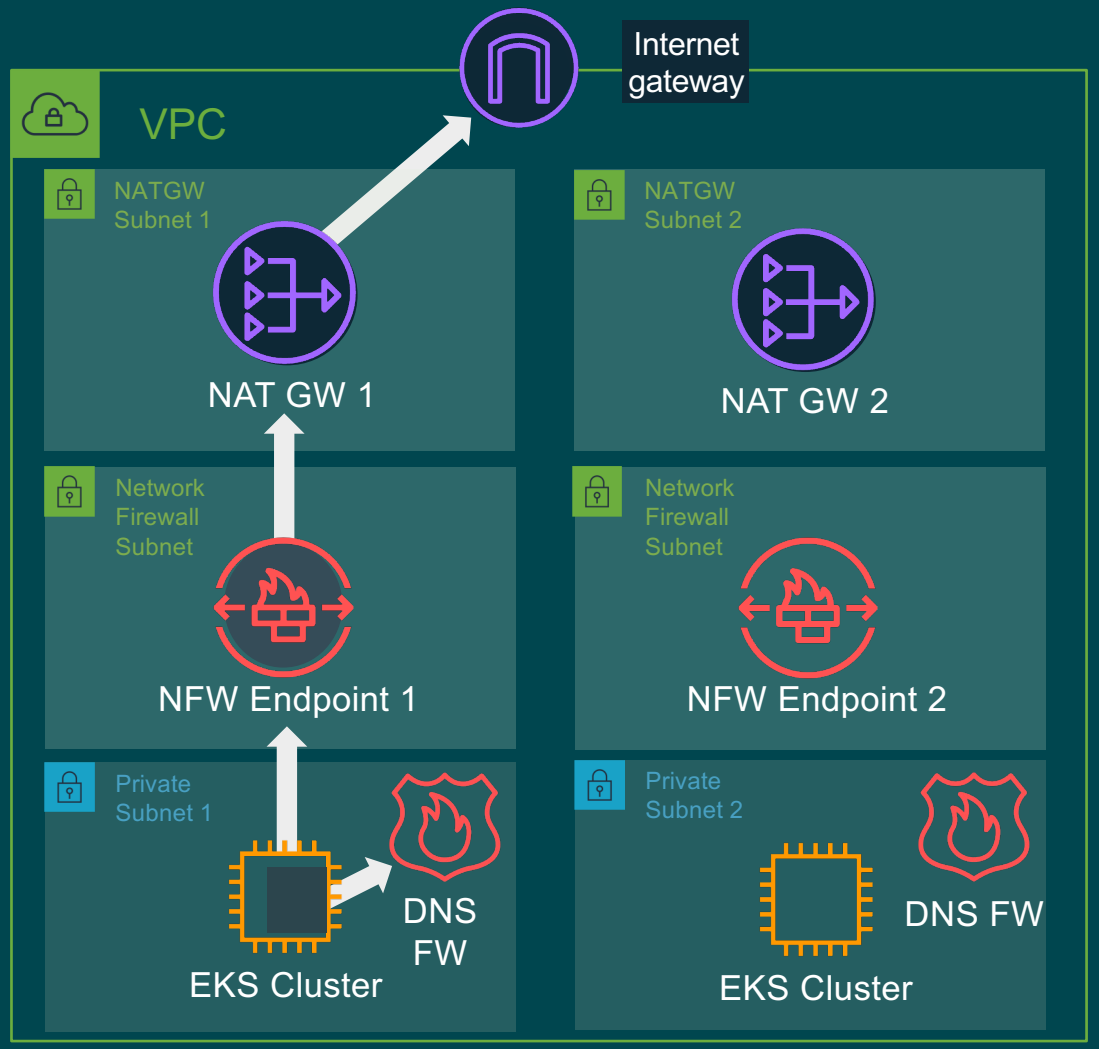


- Stop DDoS attacks with AWS Shield Advanced and Automatic Application Layer(L7) Mitigation
- Protect against OWASP top 10 web attack such as SQL injection and Cross-Site Scripting
- Gain additional visibility and control over malicious bot traffic
- Support for centralized (shown) and distributed application architectures

aws

Mitigate software supply-chain risk

Amazon Route 53 Resolver DNS Firewall and AWS Network Firewall

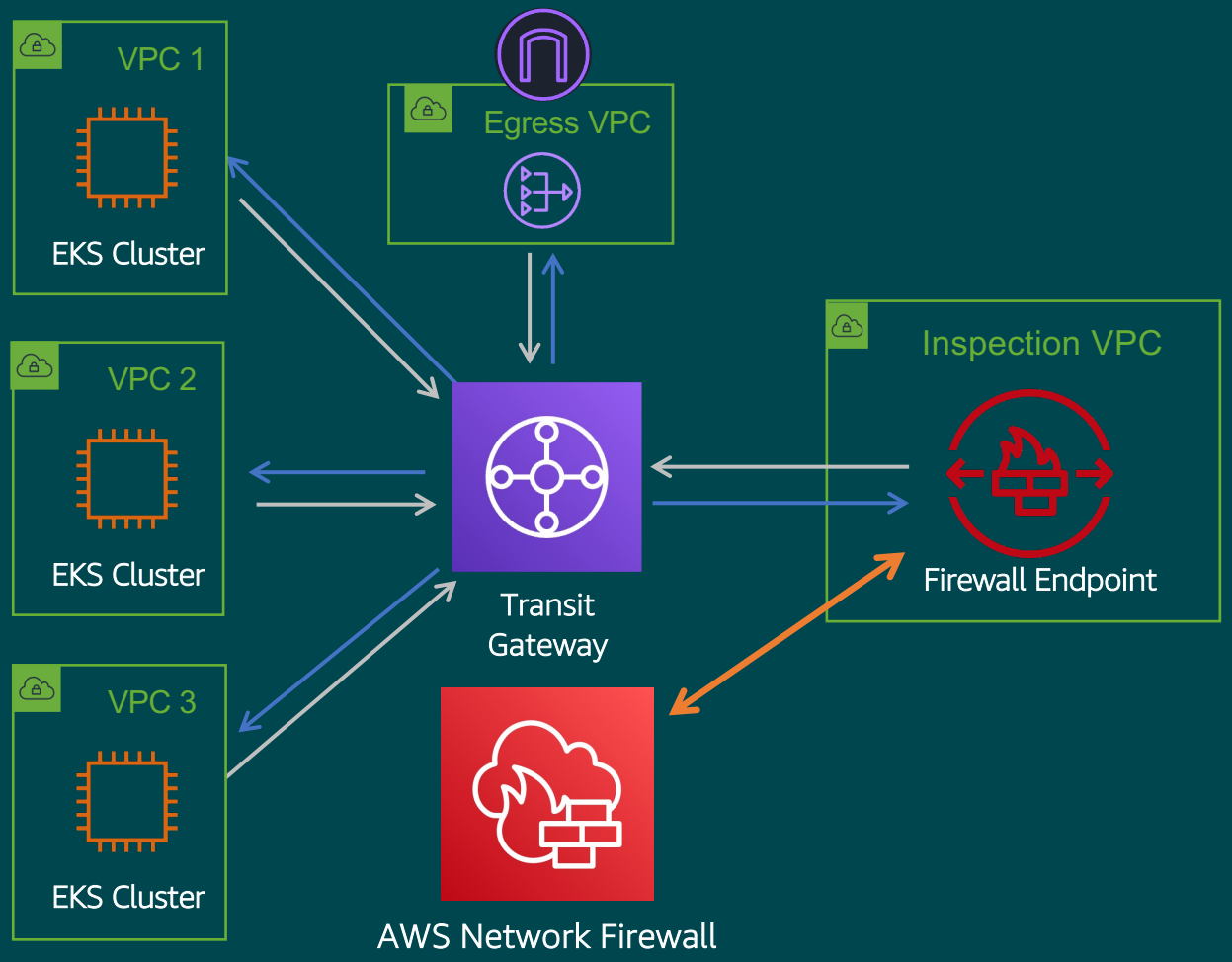


- Detect and block unexpected network egress communications from malicious code
- Limit network communications to trusted code and container repositories
- Add network protocol detection and enforcement to egress controls
- Support for centralized or distributed (shown) deployment models

aws

Block malicious network activity

AWS Network Firewall Managed Threat Signatures



- AWS Network Firewall includes 16 threat signature categories including coin mining, web attacks and emerging events
- Full visibility into signature content to selectively choose what traffic to block or alert against
- Inspect traffic in centralized (shown) or distributed deployment models

Demo

Threat Detection and Response

THREAT Detection

Continuously check your AWS accounts and workloads to detect threats and verify compliance based on industry standards.

AWS Detective

Makes it easy to analyze, investigate, and quickly identify the root cause of potential security issues or suspicious activities

AWS GuardDuty

Threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3

AWS Inspector

Automated security assessment service that helps improve the security and compliance of applications deployed on AWS

Amazon GuardDuty

Data sources

VPC flow logs

DNS logs

Login events

CloudTrail events

S3 data plane events

EKS runtime & control plane logs

Security analytics

Machine learning

Threat intelligence

Continuous learning

Security findings

Actionable

Accurate

Contextual

Integration

Amazon Detective

AWS Security Hub

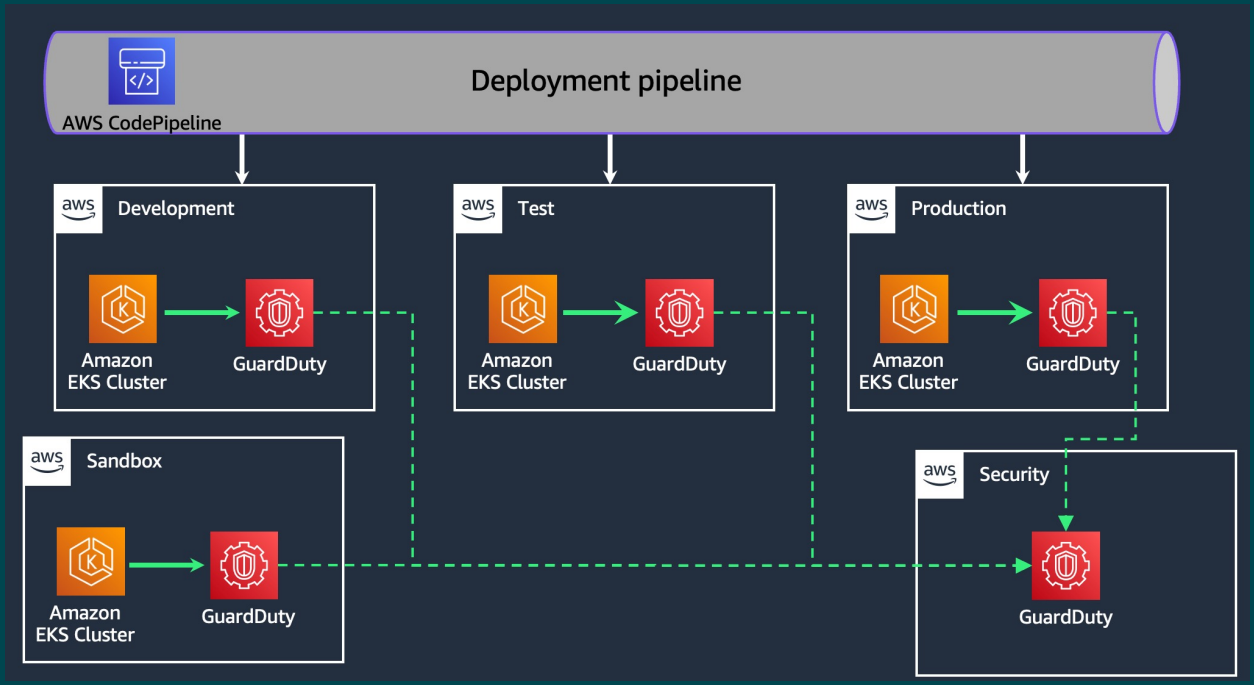
Amazon EventBridge

AWS Partner Network

aws

Detect malicious activity running in container workloads

Threat Detection with Amazon GuardDuty



- Review Amazon EKS specific activity, such as pod volume patterns and container service user activity, including divergent behavior within and across EKS clusters
- Investigate security findings with EKS clusters, such as cryptocurrency mining, unintentional admin privilege exposure, container misconfigurations that allow access to underlying EC2 nodes, or behavioral patterns common to compromised container clusters

GuardDuty for Kubernetes Finding Types

Policy

- Exposed dashboard
- Admin access to default service account
- Anonymous access granted

Malicious access

- Data discovery, exfiltration, or modification from:
  - Tor
  - Successful anonymous access
  - Malicious IPs

Suspicious behavior

- Execution in Kubernetes system pod
- Container with sensitive mount
- Privilege container

- GuardDuty immediately begins to analyze Kubernetes data sources from your Amazon EKS clusters and monitors them for malicious and suspicious activity.

GuardDuty aligns findings using the MITRE ATT&CK framework

Credential Access Defense Evasion Discovery Impact

Privilege Escalation Policy Execution Persistence

Amazon GuardDuty Malware Protection

- **GuardDuty Malware Protection** – a fully managed malware detection service supports detection of malicious software by scanning Amazon Elastic Block Storage (EBS) and container workloads running on Amazon EC2.

➡ Detects malware (trojans, worms, rootkits, crypto miners, bots, etc.) and generates a new finding in GuardDuty

➡ Scans Elastic Block Storage (EBS) for malware on Amazon EC2 instance and container workloads that are exhibiting suspicious behavior

- *Automatic - malware scanning triggered on GuardDuty findings*
- *Agentless - no security software required to install or maintain*

aws

GuardDuty Malware Protection

DELIVERS AGENTLESS DETECTION OF MALWARE ON AWS WORKLOADS

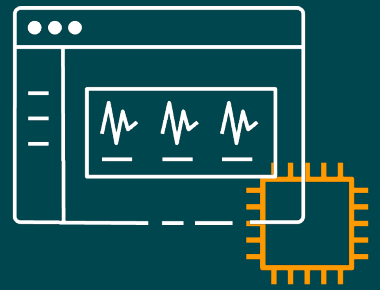
Single-click organization-wide malicious file detection

Centralized monitoring, automation, and investigation

Contextualized findings to validate suspicious behavior

No agents to install, update, or maintain

Container aware



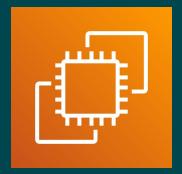
No performance impact or hidden Amazon EC2 costs for scanning

aws

GuardDuty Malware Protection: Coverage

Amazon EC2 coverage

Amazon EC2 instances



Amazon ECS

Amazon EKS

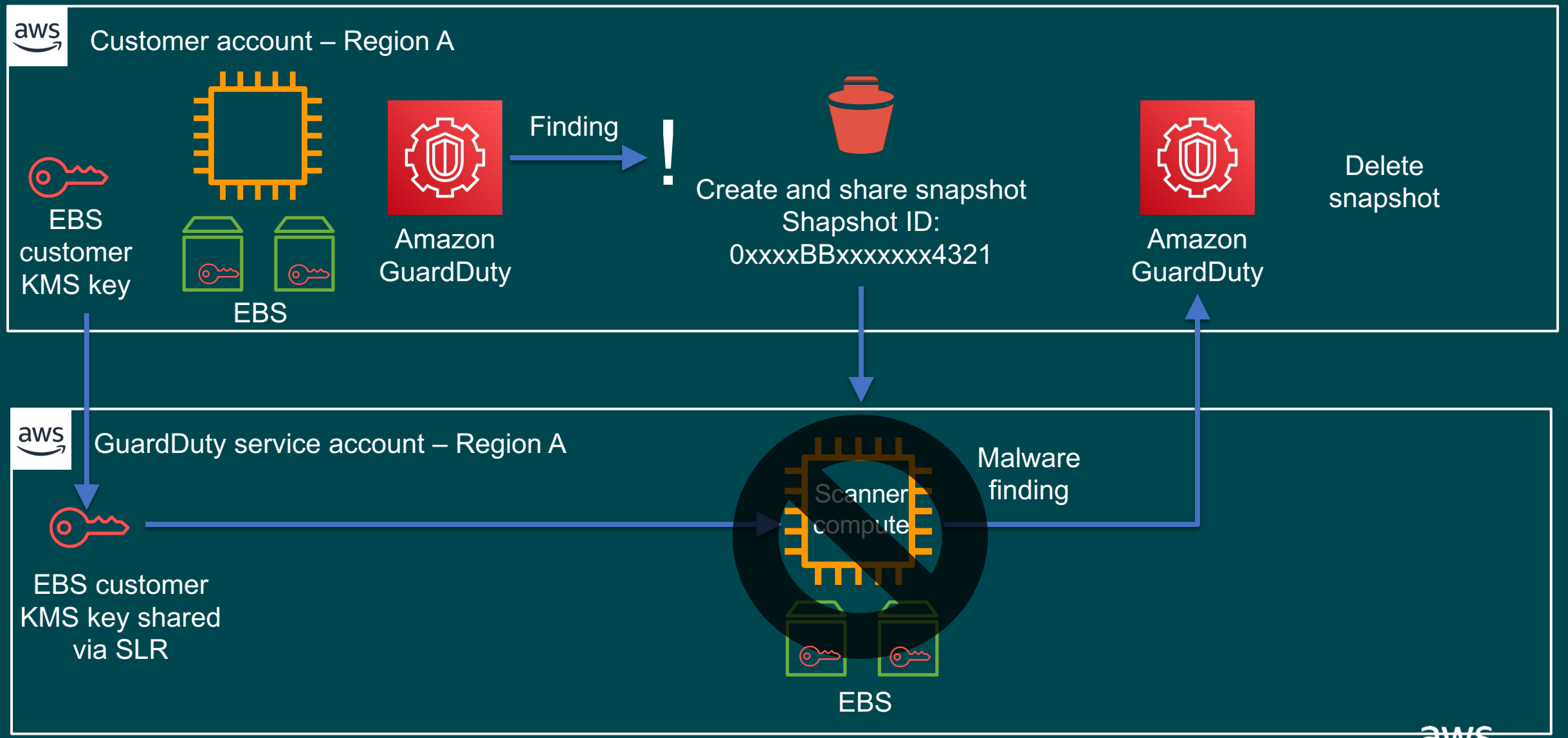
Self-managed containers on Amazon EC2

When does GuardDuty initiate a malware scan?

- Malware scans are automatically triggered when GuardDuty detects a potentially compromised Amazon EC2 instance to identify malware that may be causing the activity
- It only scans an EC2 instance once every 24 hours, irrespective of multiple GuardDuty findings observed on it

aws

How does it work?



Investigate events

Investigate security issues in container workloads

Identify the root-cause of malicious or suspicious behavior with Amazon Detective

- Review Amazon EKS specific activity, such as pod volume patterns and container service user activity, including divergent behavior within and across EKS clusters
- Investigate security findings with EKS clusters, such as cryptocurrency mining, unintentional admin privilege exposure, container misconfigurations that allow access to underlying EC2 nodes, or behavioral patterns common to compromised container clusters

Amazon Detective Use Case Benefits

INVESTIGATE ISSUES FASTER AND WITH LESS EFFORT

1) Finding / Alert triage

Accelerate triage and avoid unnecessary escalations

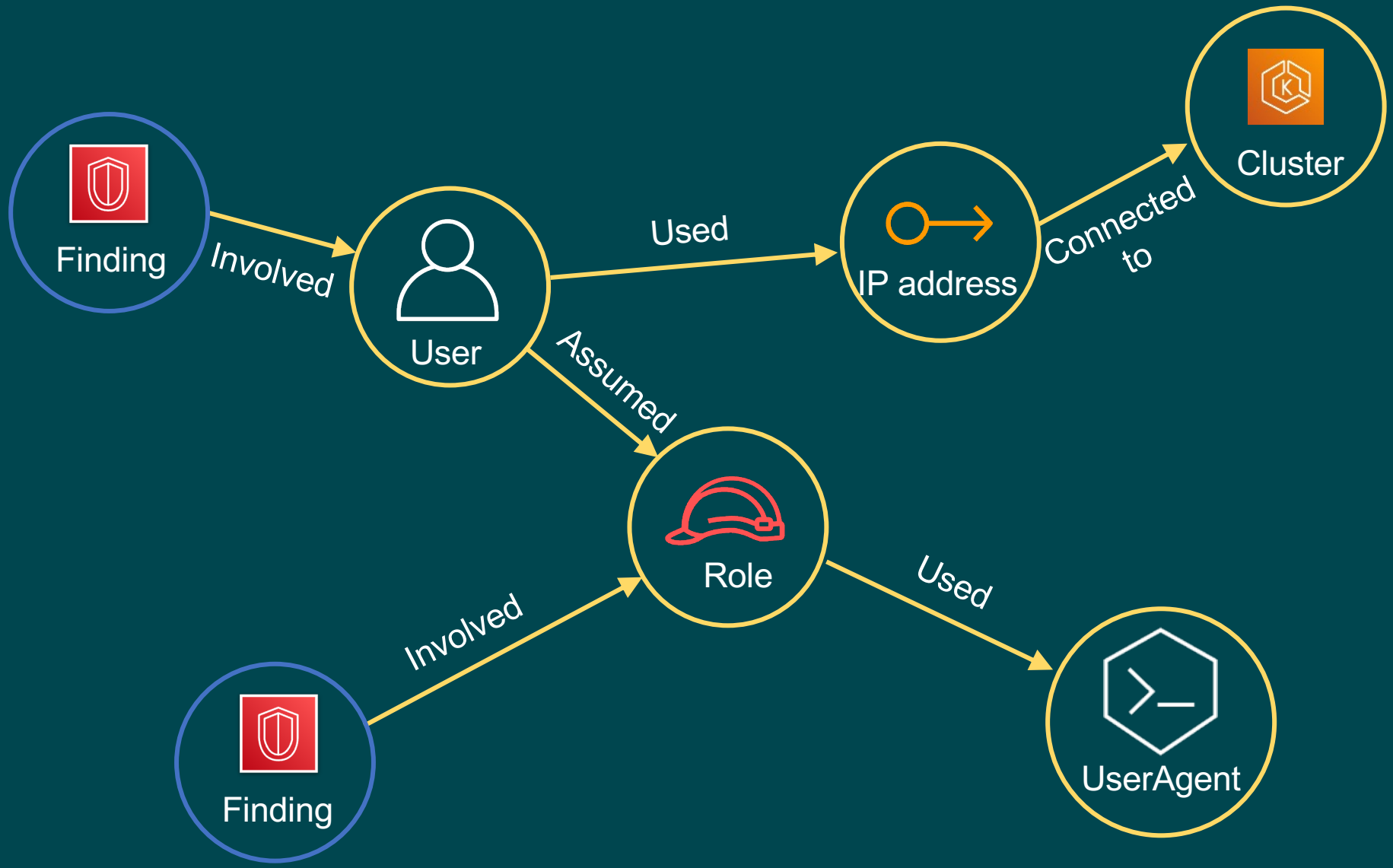
2) Incident Investigation

Improve context and surface correlated behavior

3) Threat Hunting

Simplify data collection, aggregation and pivoting

Amazon Detective - Security behavior graph



Amazon Detective support for EKS

- **Amazon Detective** security investigations for Amazon Elastic Kubernetes Service (Amazon EKS) clusters to quickly analyze, investigate, and identify the root-cause of malicious or suspicious behavior that represents potential threats to container workloads.

→ Review Amazon EKS specific activity, such as pod volume patterns and container service user activity, including divergent behavior within and across EKS clusters

→ Investigate security findings with their EKS clusters, such as cryptocurrency mining, unintentional admin privilege exposure, container misconfigurations that allow access to underlying EC2 nodes, or behavioral patterns common to compromised container clusters.

Demo

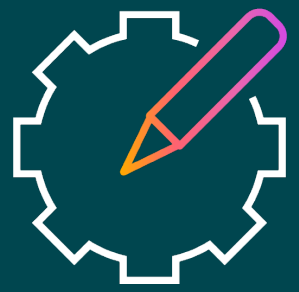
Do you have customers asking for container runtime protection?

- Yes
- No

Continuous monitoring with automated response & remediation

Security Hub – response & remediation

Enable **automated remediation** for high-severity configuration findings



Use **custom actions** to invoke runbooks for automated response

Use **partner integrations** to consolidate and normalize security findings

Integrate with **ticketing and workflow** tools

aws

Simple and Scalable Security Monitoring

Scale existing services

customers use same console, findings, and experience

Amazon GuardDuty

Amazon Detective

Amazon Inspector

Simple and easy deployment

One-click enables container support
AWS Orgs assures environment-wide enablement
(new customers have support on by default)

GuardDuty EKS Protection
GuardDuty monitors EKS cluster via Kubernetes audit logs (KAL)

GuardDuty Malware Protection
At launch is aware of containers running on EC2

Detective EKS Investigation
Detective is container-aware and continuously aggregates KAL into graph model and analytics. Pivot from GuardDuty console to immediately investigate findings for root cause analysis.

Inspector ECR Support
Inspector scans ECR images on push and continually for software vulnerability management—integrates with ECR console for easy builder communication

Continuous monitoring

Centralization of security findings scales and automates operations

Automate Response
Initiate automated remediation, invoke runbooks, integrate with ticketing and workflow tools

AWS Security Hub

Automate and reduce risk with integrated services

Comprehensive set of APIs
and security tools

Continuous monitoring
and protection

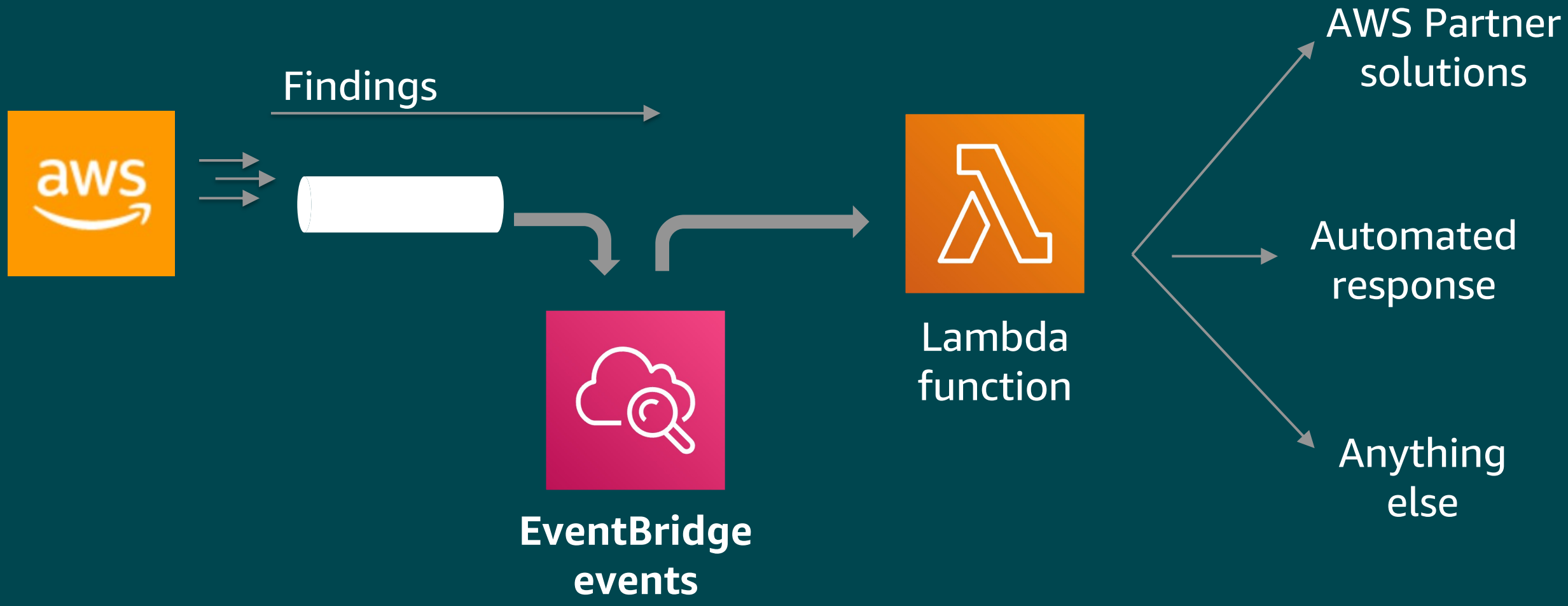
Threat remediation
and response

Operational efficiencies to
focus on critical issues

Securely deploy business
critical applications

aws

Threat response: High-level playbook



Threat response: Services

Lambda

Run code for virtually any kind of application or backend service – zero administration

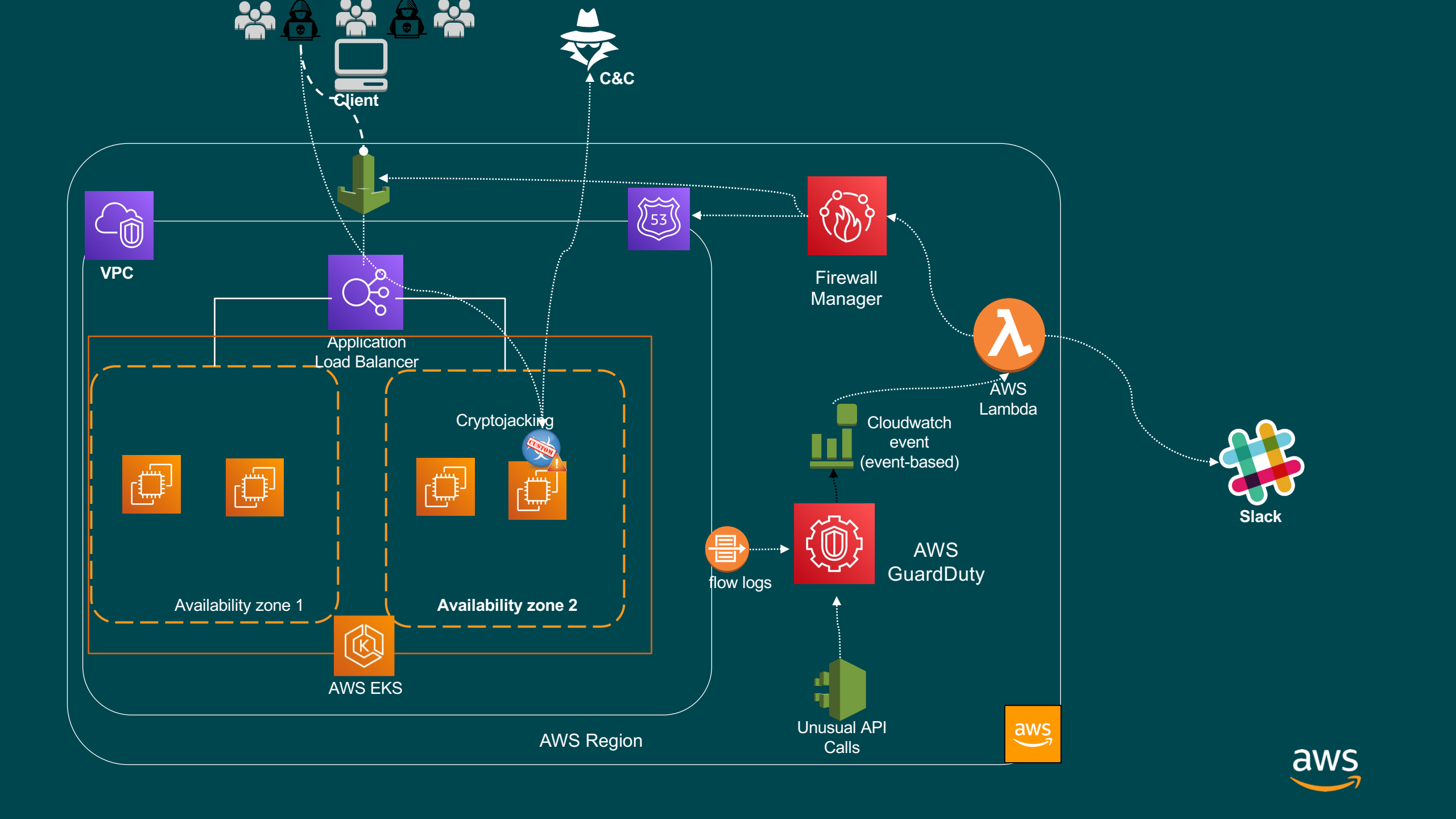
Systems Manager

AWS Step Functions workflows

Amazon VPC

AWS WAF

and others

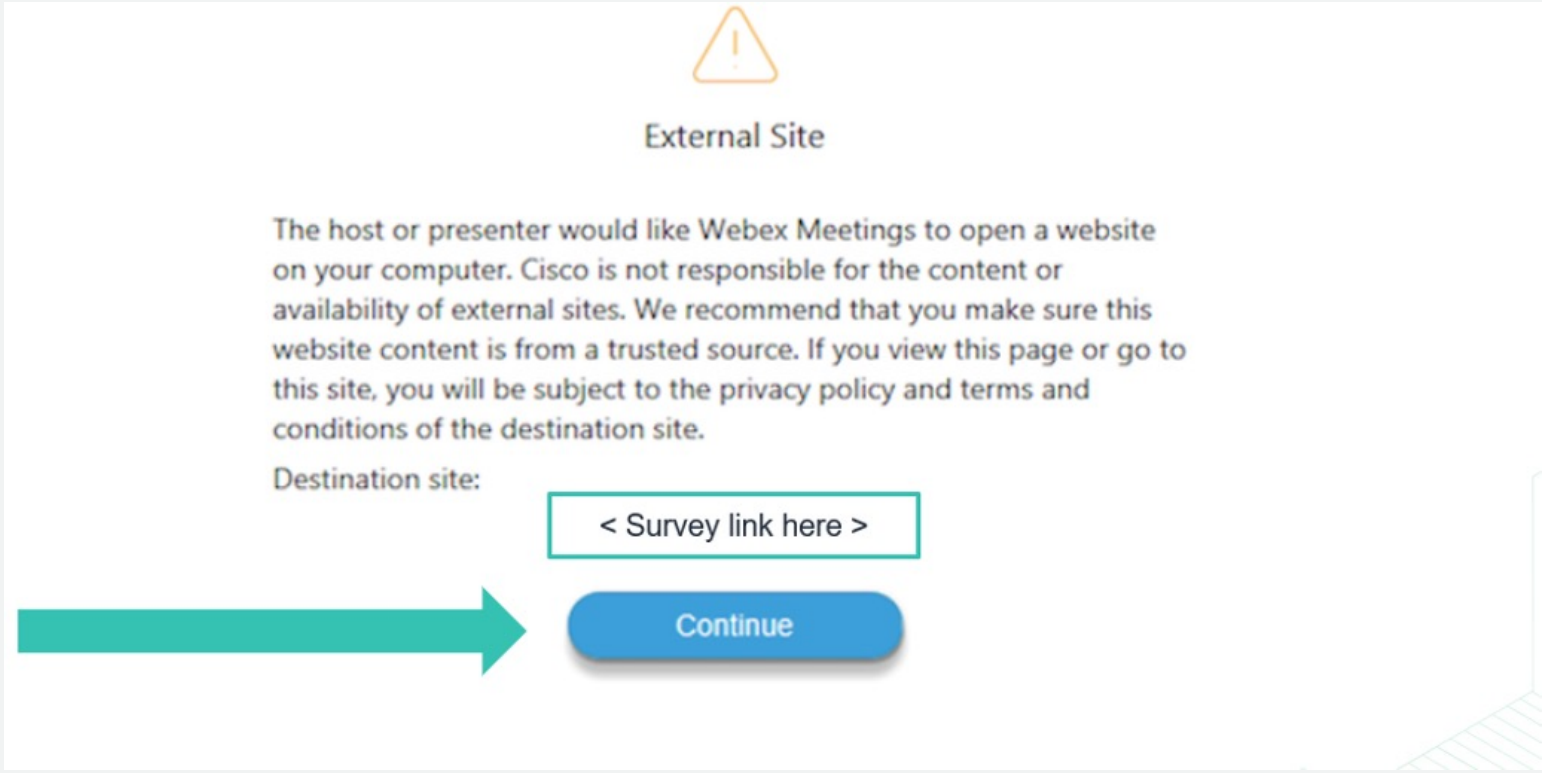

Client
C&C
VPC
53
Firewall Manager
Application Load Balancer
AWS Lambda
Cryptojacking
CUSTOM
Cloudwatch event (event-based)
Slack
flow logs
AWS GuardDuty
Availability zone 1
Availability zone 2
AWS EKS
Unusual API Calls
AWS Region
aws

Q&A

aws partnercast

Exit Survey

We appreciate your feedback! Please complete the survey at the end of this session. You will be automatically redirected via this screen:



aws partnercast

Thank you!

Please join us again for another PartnerCast session

https://aws.amazon.com/partners/training/partnercast/