# New AWS security services for container threat detection

**Scott Ward**
Principal Solutions Architect
AWS, External Security Services

# Why customers adopt containers

Reduced risk

Uniform security across environment, maintained with automation

Operational efficiency

Reduced operational burden by removing undifferentiated heavy lifting

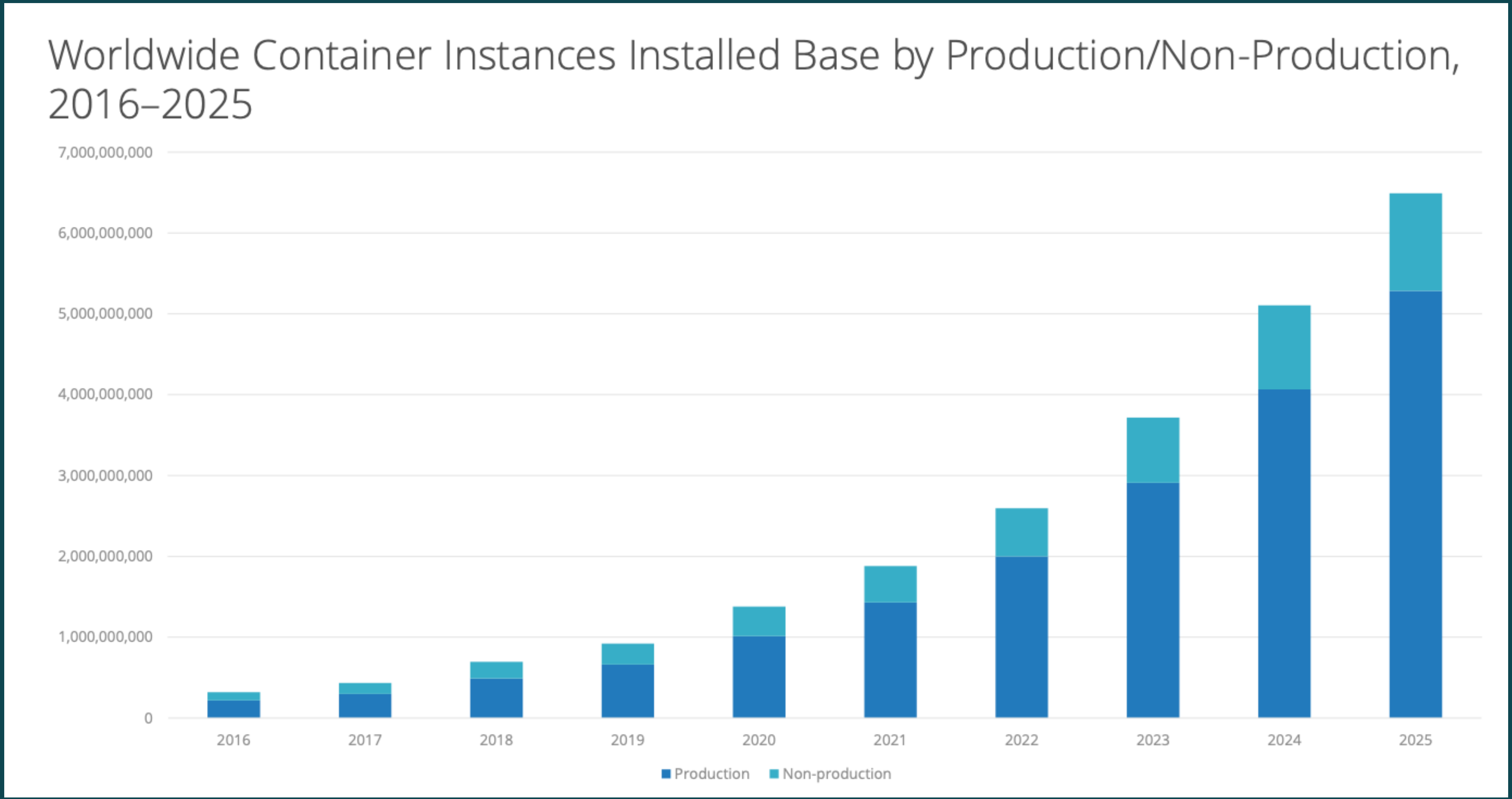
Speed

Consistent environment improves developer velocity

Agility

Automation increases speed and ease of testing and iterating

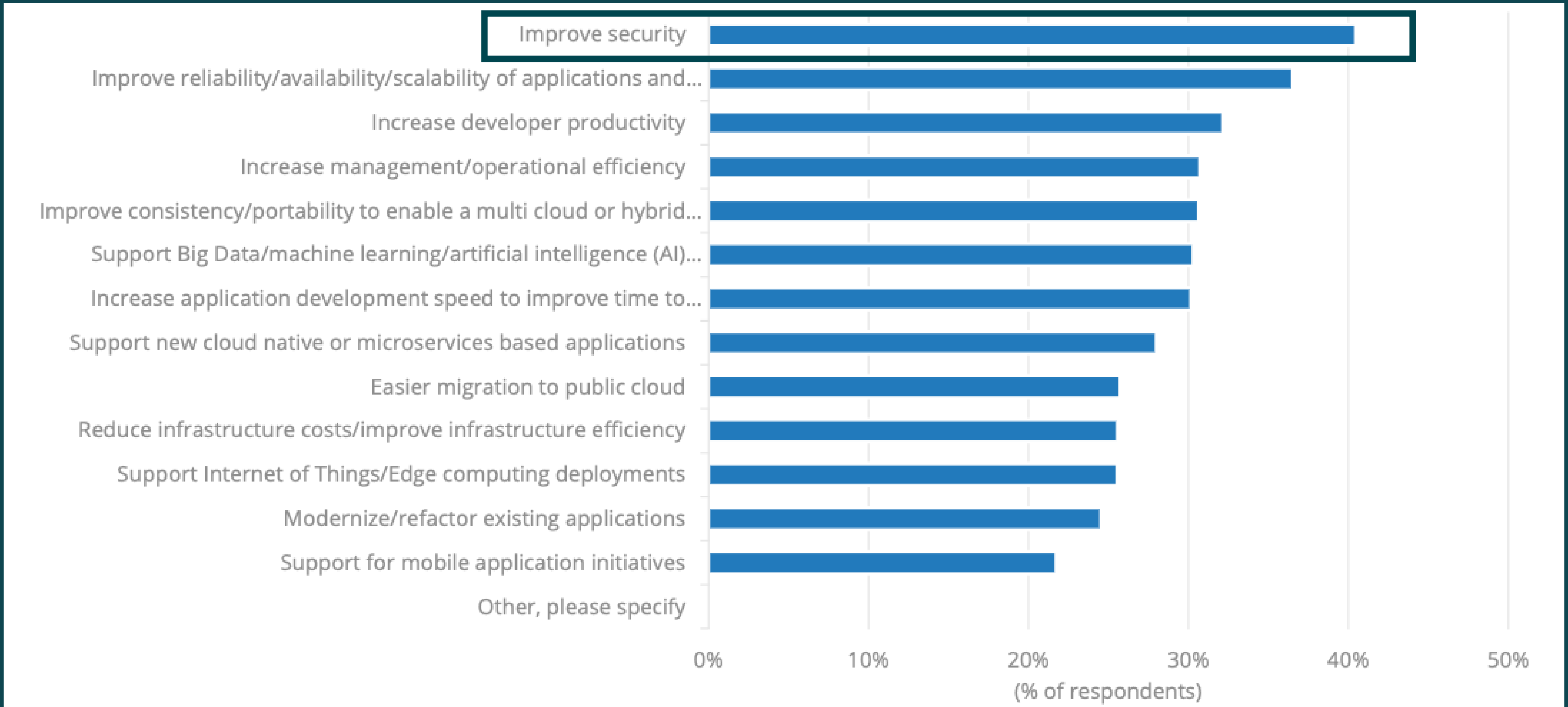
# Container deployment production/non-production



Source: Container Infrastructure Software Survey, IDC, December 2021

# Drivers to deploy containers

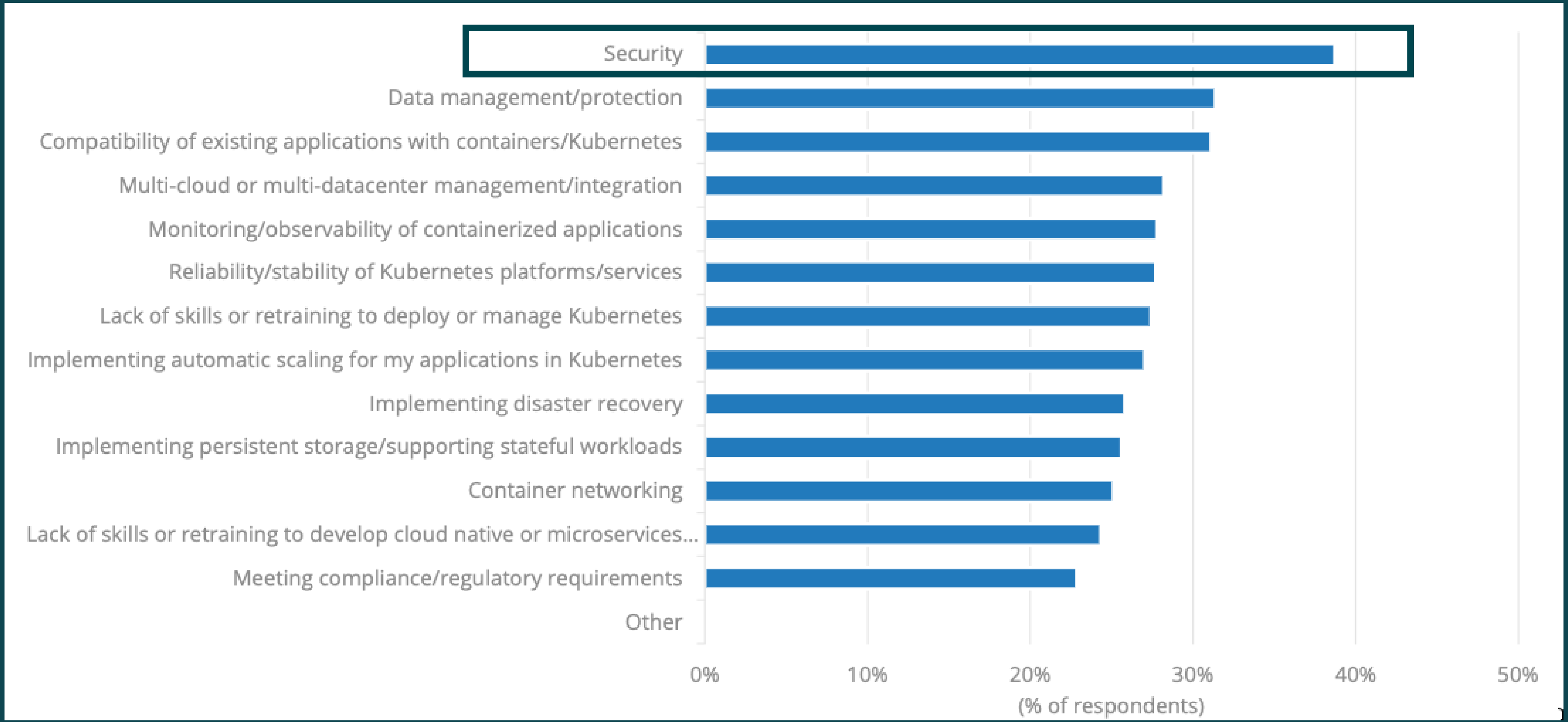
What were the primary drivers that caused your organization to initially deploy containers and Kubernetes?



Source: Container Infrastructure Software Survey, IDC, December 2021

# Drivers to deploy containers

What were the top challenges when deploying containers and Kubernetes



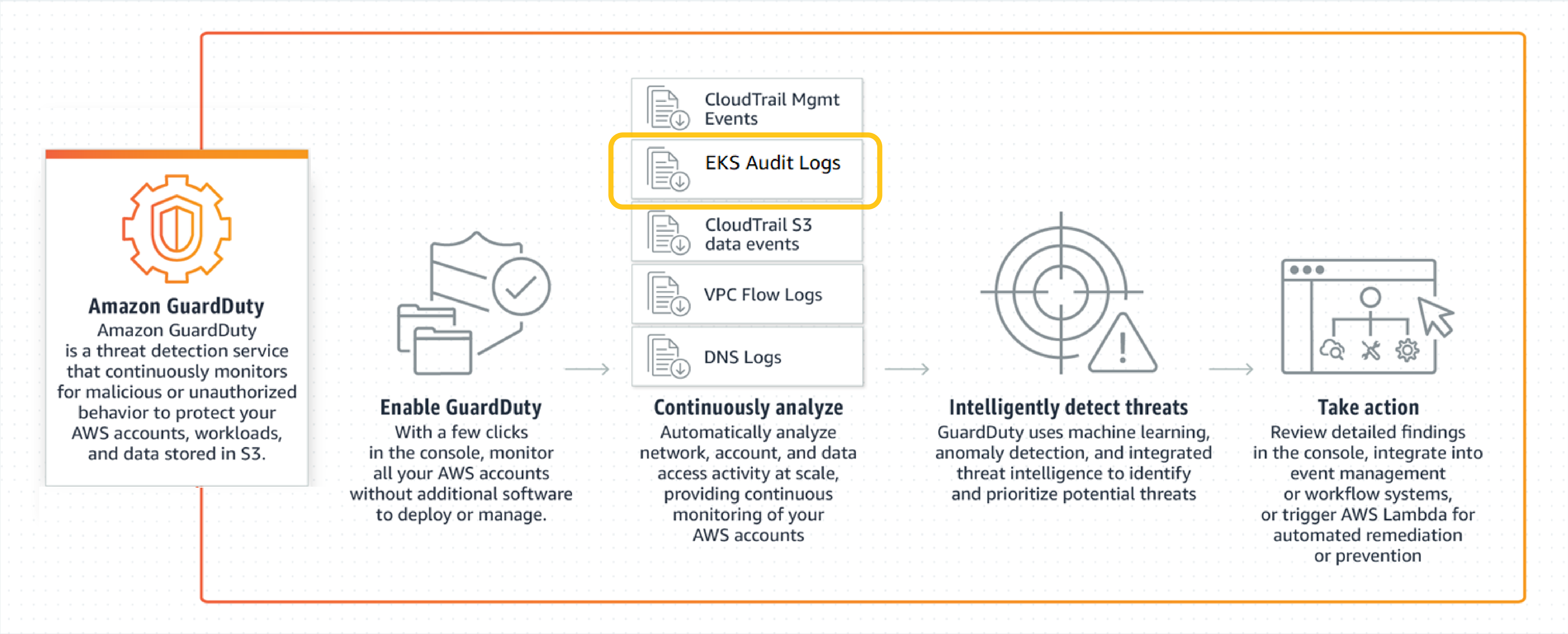
Security is a top driver and a top challenge

Source: Container Infrastructure Software Survey, IDC, December 2021

# What are best practices for securing container workloads running on AWS?

# Amazon GuardDuty for EKS Protection

# Amazon GuardDuty – expanded coverage for EKS



**Amazon GuardDuty**

Amazon GuardDuty is a threat detection service that continuously monitors for malicious or unauthorized behavior to protect your AWS accounts, workloads, and data stored in S3.

CloudTrail Mgmt Events

EKS Audit Logs

CloudTrail S3 data events

VPC Flow Logs

DNS Logs

**Enable GuardDuty**

With a few clicks in the console, monitor all your AWS accounts without additional software to deploy or manage.

**Continuously analyze**

Automatically analyze network, account, and data access activity at scale, providing continuous monitoring of your AWS accounts

**Intelligently detect threats**

GuardDuty uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats

**Take action**

Review detailed findings in the console, integrate into event management or workflow systems, or trigger AWS Lambda for automated remediation or prevention

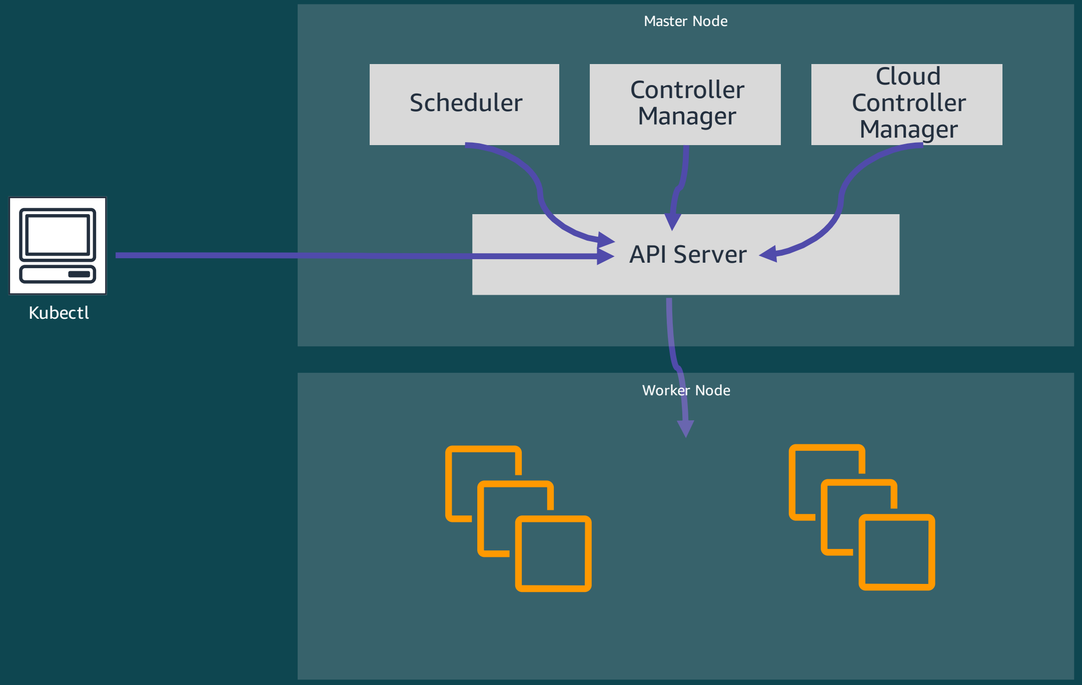
# Amazon EKS control plane API and audit logs

Kubernetes control plane API – HTTP API to query and manipulate the state of API objects in Kubernetes

Pods, Namespaces, ConfigMaps, Events

Audit logs provide information on API interactions

- What happened?
- When did it happen?
- Who initiated it?
- On what did it happen?
  - Endpoints, pods, configmap, etc.
- Where was it observed?
- From where was it initiated?
- To where was it going?



# What can GuardDuty for EKS detect with audit logs

### Credential Access

MaliciousIPCaller
MaliciousIPCaller.Custom
SuccessfulAnonymousAccess
TorIPCaller

### Defense Evasion

MaliciousIPCaller
MaliciousIPCaller.Custom
SuccessfulAnonymousAccess
TorIPCaller

### Discovery

MaliciousIPCaller
MaliciousIPCaller.Custom
SuccessfulAnonymousAccess
TorIPCaller

### Impact

MaliciousIPCaller
MaliciousIPCaller.Custom
SuccessfulAnonymousAccess
TorIPCaller

### Persistence

ContainerWithSensitiveMount
MaliciousIPCaller
MaliciousIPCaller.Custom
SuccessfulAnonymousAccess
TorIPCaller

### Policy

AdminAccessToDefaultServiceAccount
AnonymousAccessGranted
ExposedDashboard
KubeflowDashboardExposed

### Execution

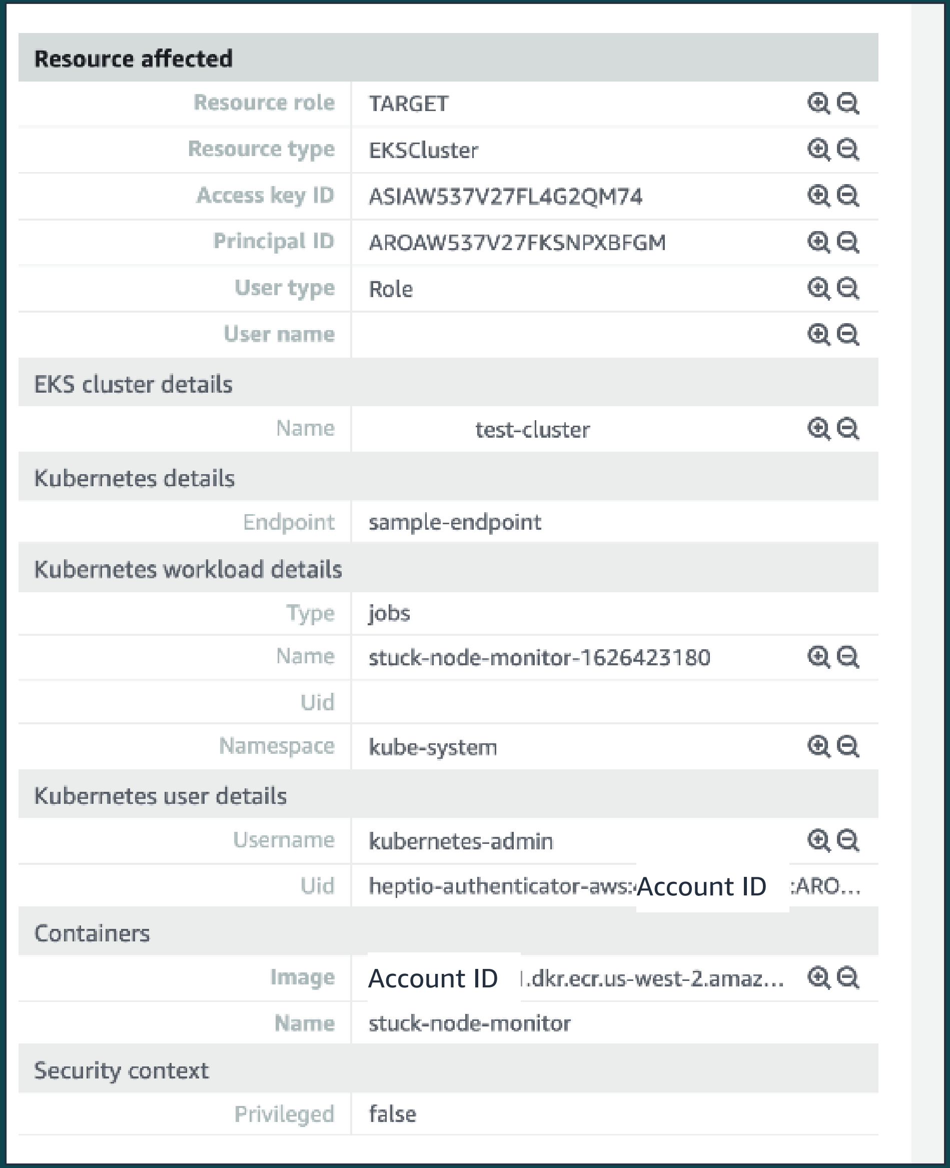
ExecInKubeSystemPod

### Privilege Escalation

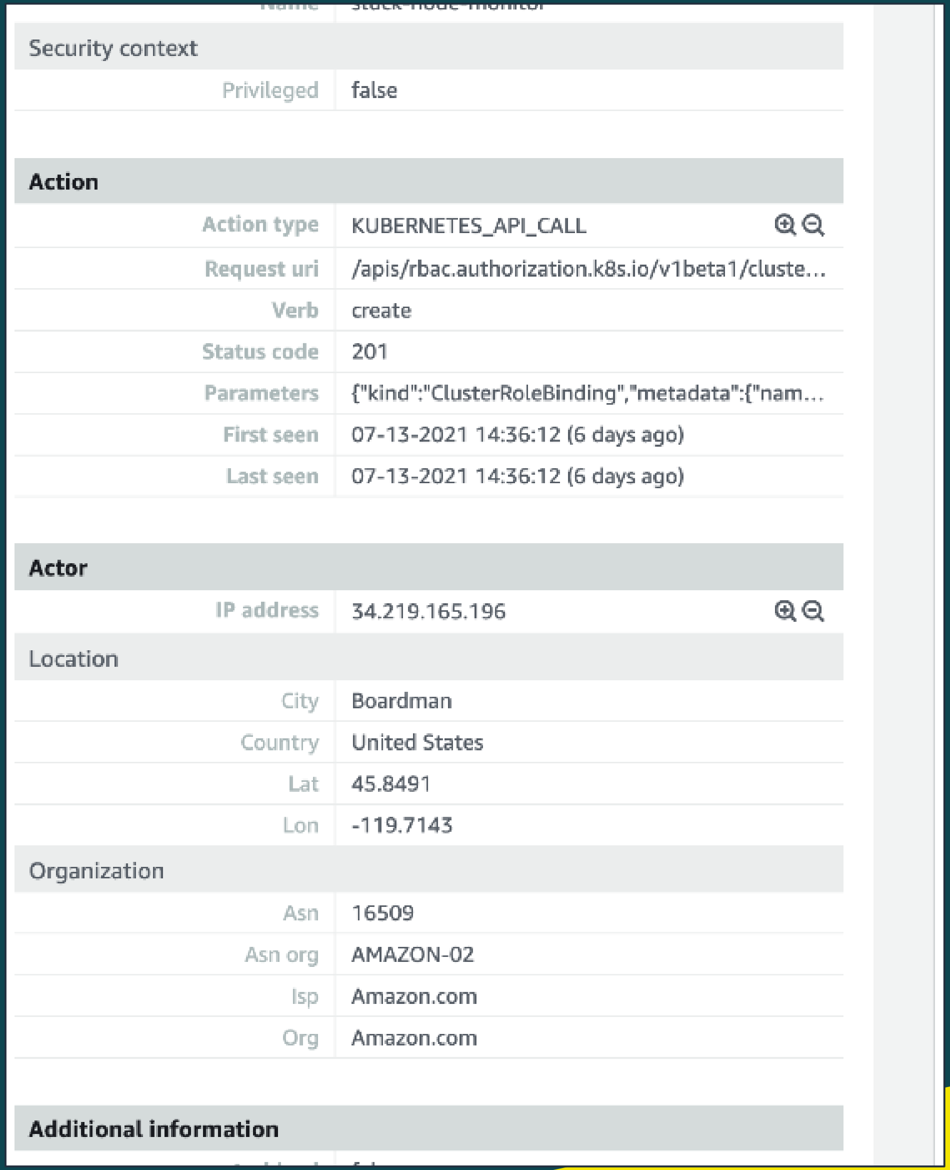
PrivilegedContainer

Currently 27 finding types

# GuardDuty EKS findings detail



| | |
|---|---|
| **Resource affected** | |
| Resource role | TARGET |
| Resource type | EKSCluster |
| Access key ID | ASIAW537V27FL4G2QM74 |
| Principal ID | AROAW537V27FKSNPXBFGM |
| User type | Role |
| User name | |
| EKS cluster details | |
| Name | test-cluster |
| Kubernetes details | |
| Endpoint | sample-endpoint |
| Kubernetes workload details | |
| Type | jobs |
| Name | stuck-node-monitor-1626423180 |
| Uid | |
| Namespace | kube-system |
| Kubernetes user details | |
| Username | kubernetes-admin |
| Uid | heptio-authenticator-aws:Account ID :ARO… |
| Containers | |
| Image | Account ID .dkr.ecr.us-west-2.amaz… |
| Name | stuck-node-monitor |
| Security context | |
| Privileged | false |



| | |
|---|---|
| Name | stuck-node-monitor |
| Security context | |
| Privileged | false |
| **Action** | |
| Action type | KUBERNETES_API_CALL |
| Request uri | /apis/rbac.authorization.k8s.io/v1beta1/cluste… |
| Verb | create |
| Status code | 201 |
| Parameters | {"kind":"ClusterRoleBinding","metadata":{"nam… |
| First seen | 07-13-2021 14:36:12 (6 days ago) |
| Last seen | 07-13-2021 14:36:12 (6 days ago) |
| **Actor** | |
| IP address | 34.219.165.196 |
| Location | |
| City | Boardman |
| Country | United States |
| Lat | 45.8491 |
| Lon | -119.7143 |
| Organization | |
| Asn | 16509 |
| Asn org | AMAZON-02 |
| Isp | Amazon.com |
| Org | Amazon.com |
| **Additional information** | |

# Remediation of GuardDuty for EKS Protection findings

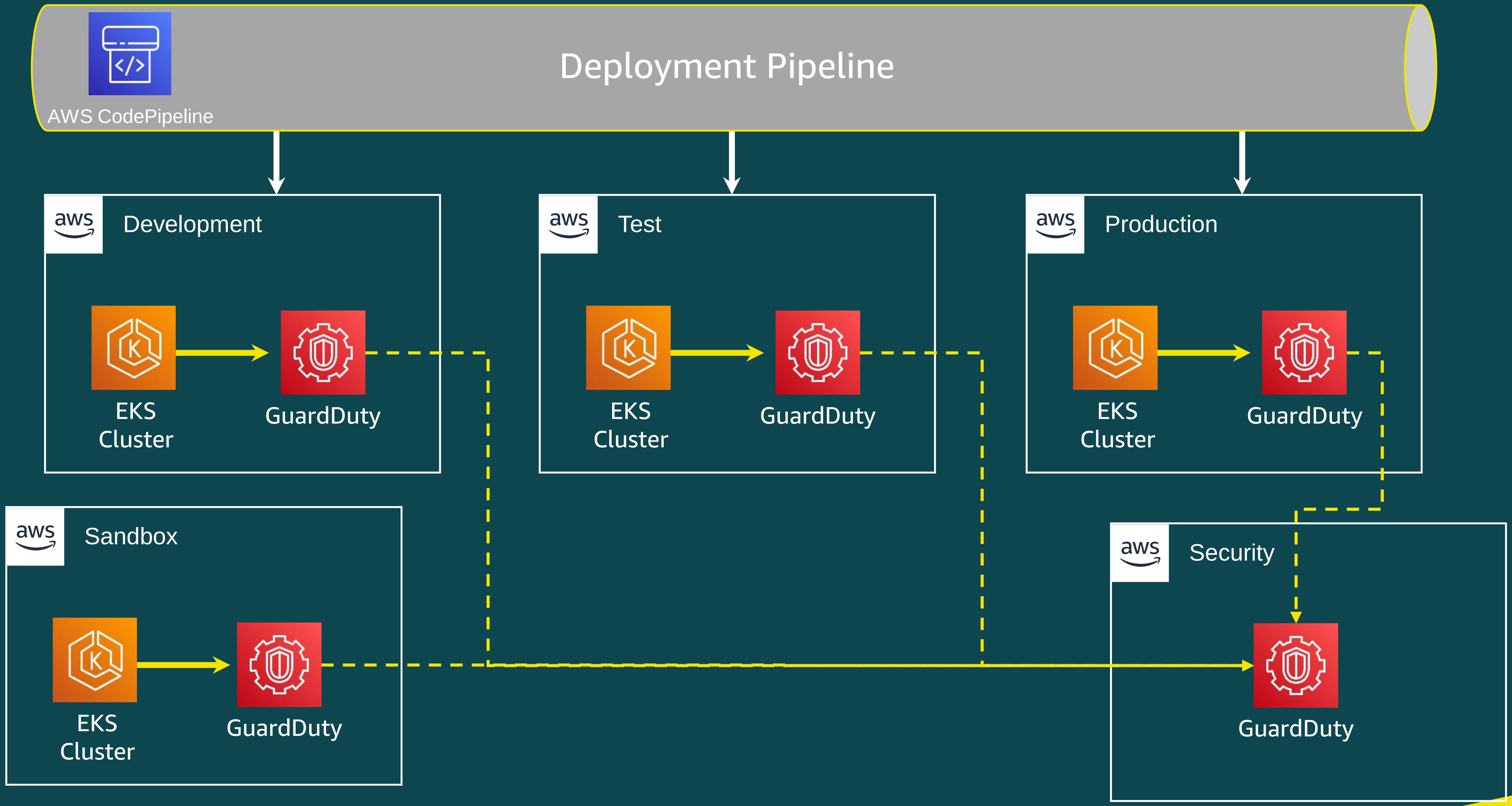
GuardDuty Kubernetes remediation guidance:
https://docs.aws.amazon.com/guardduty/latest/ug/guardduty-remediate-kubernetes.html

EKS security best practices:
https://aws.github.io/aws-eks-best-practices/security/docs/

- Make cluster API endpoint private
  - If public is a must, then white list specific CIDR IP addresses
- Review and revoke unnecessary anomalous access
- Reverse actions taken – where appropriate
- Rotate credentials and secrets of impacted users
- Isolate pods, revoke pod credentials, and gather data for forensics
- Terminate pods or nodes
- Patch container image and re-deploy

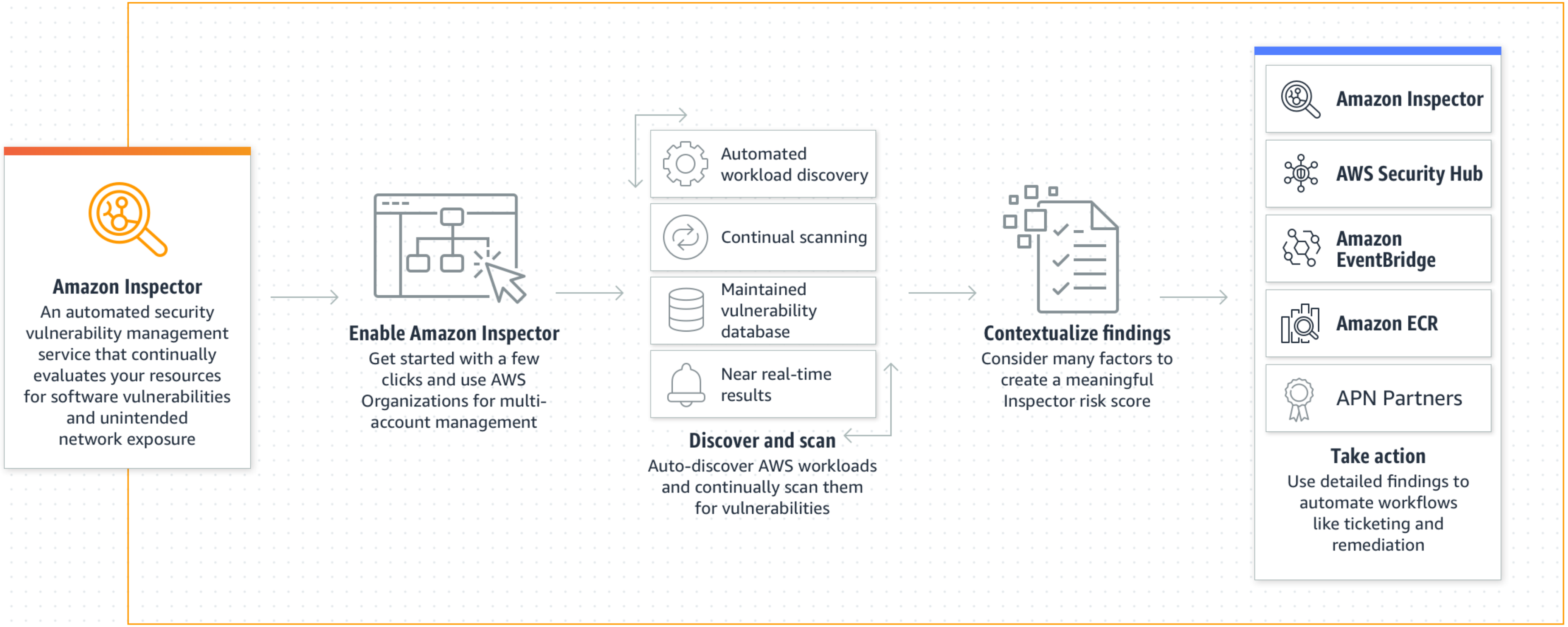
Automate wherever you can

# Where can you use GuardDuty for EKS Protection?



# Amazon Inspector

# Amazon Inspector – How it works



**Amazon Inspector**
An automated security vulnerability management service that continually evaluates your resources for software vulnerabilities and unintended network exposure

**Enable Amazon Inspector**
Get started with a few clicks and use AWS Organizations for multi-account management

- Automated workload discovery
- Continual scanning
- Maintained vulnerability database
- Near real-time results

**Discover and scan**
Auto-discover AWS workloads and continually scan them for vulnerabilities

**Contextualize findings**
Consider many factors to create a meaningful Inspector risk score

- Amazon Inspector
- AWS Security Hub
- Amazon EventBridge
- Amazon ECR
- APN Partners

**Take action**
Use detailed findings to automate workflows like ticketing and remediation

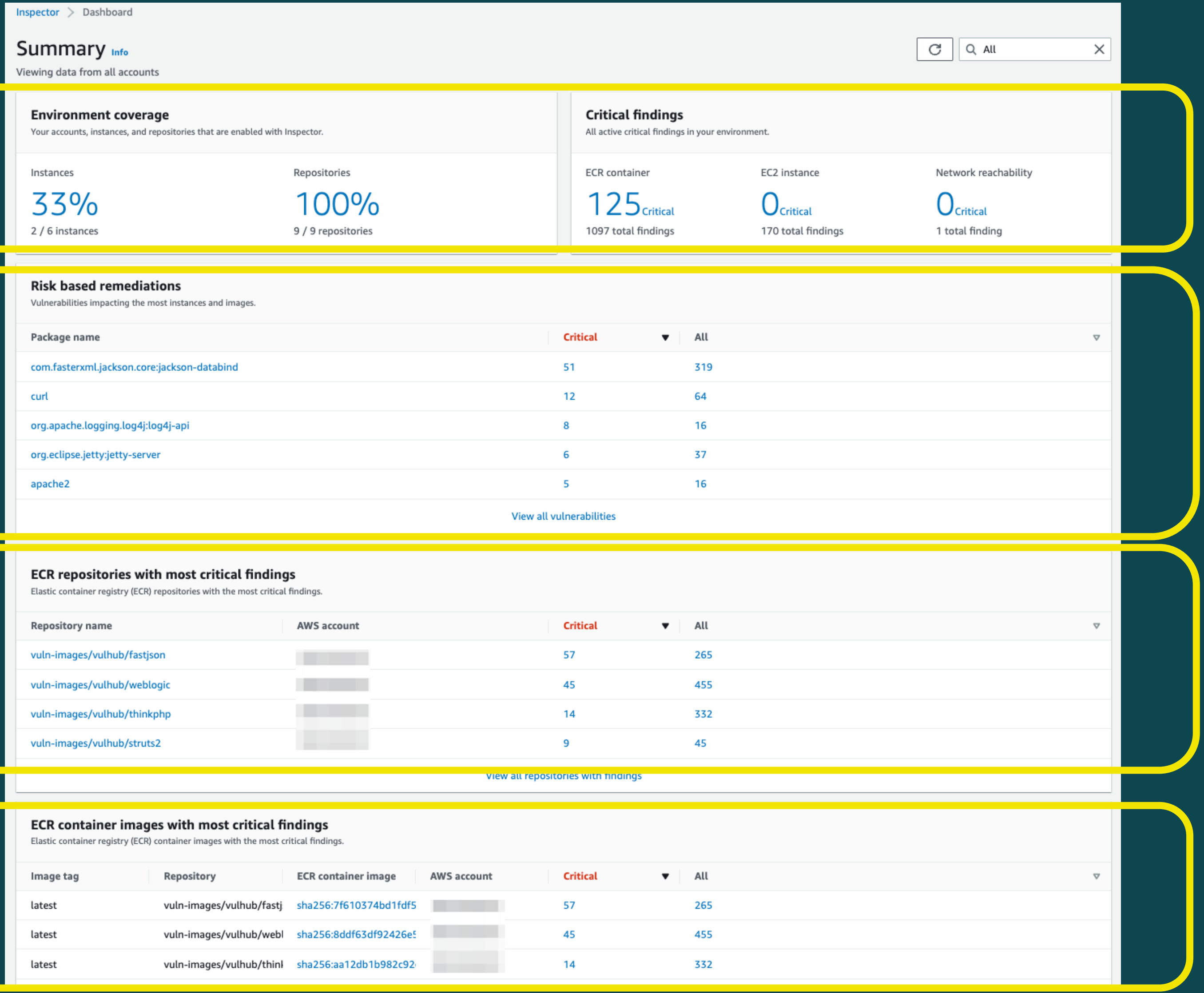
# How Inspector scans container images

- Retrieve the image from ECR
- Extract each layer of the image
- Look at OS and the installed packages
- Look through the file system for other files
- Compare against vulnerability database

# Inspector findings for container images

- Available in Inspector & Amazon ECR consoles
- Repositories configured for continuous scanning:
  - All findings closed 30 days after image was first pushed to the repo
  - No further scanning of the image occurs
- Repositories configured only for scan on-push:
  - Findings will remain open until the image is deleted
- Closed findings are deleted after 30 days
- Deleting an image closes the associated findings

# ECR scan results in Inspector



Inspector > Dashboard

## Summary Info

Viewing data from all accounts

### Environment coverage

Your accounts, instances, and repositories that are enabled with Inspector.

| Instances | Repositories |
|---|---|
| 33%<br>2 / 6 instances | 100%<br>9 / 9 repositories |

### Critical findings

All active critical findings in your environment.

| ECR container | EC2 instance | Network reachability |
|---|---|---|
| 125 Critical<br>1097 total findings | 0 Critical<br>170 total findings | 0 Critical<br>1 total finding |

### Risk based remediations

Vulnerabilities impacting the most instances and images.

| Package name | Critical ▼ | All |
|---|---|---|
| com.fasterxml.jackson.core:jackson-databind | 51 | 319 |
| curl | 12 | 64 |
| org.apache.logging.log4j:log4j-api | 8 | 16 |
| org.eclipse.jetty:jetty-server | 6 | 37 |
| apache2 | 5 | 16 |

View all vulnerabilities

### ECR repositories with most critical findings

Elastic container registry (ECR) repositories with the most critical findings.

| Repository name | AWS account | Critical ▼ | All |
|---|---|---|---|
| vuln-images/vulhub/fastjson | [redacted] | 57 | 265 |
| vuln-images/vulhub/weblogic | [redacted] | 45 | 455 |
| vuln-images/vulhub/thinkphp | [redacted] | 14 | 332 |
| vuln-images/vulhub/struts2 | [redacted] | 9 | 45 |

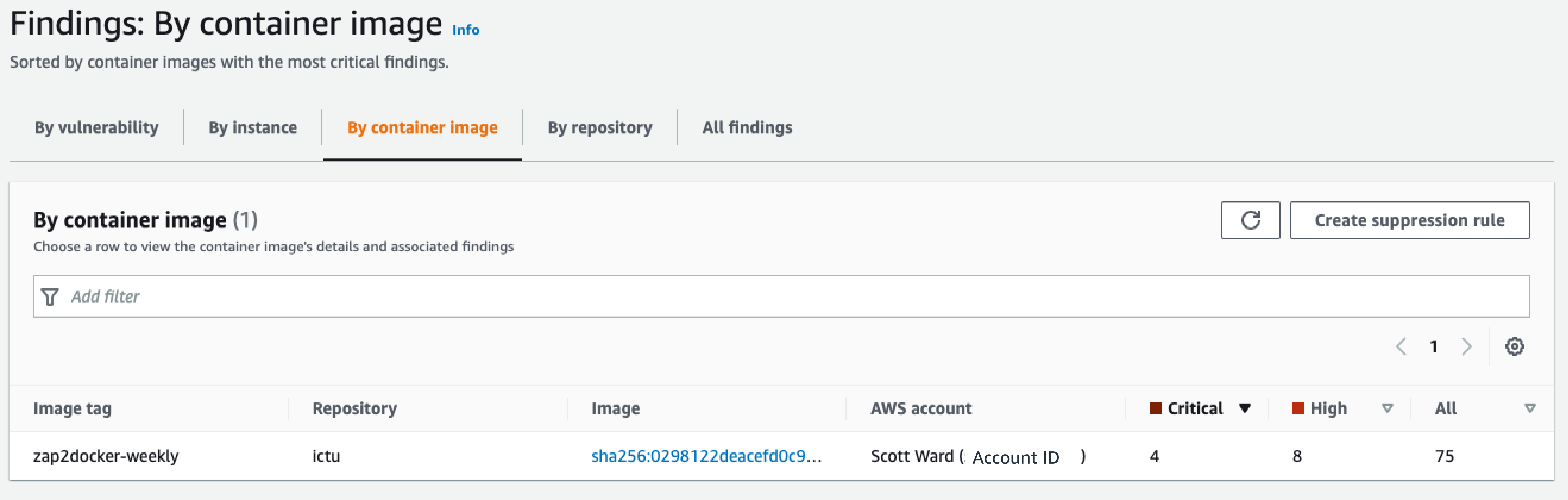
View all repositories with findings

### ECR container images with most critical findings

Elastic container registry (ECR) container images with the most critical findings.

| Image tag | Repository | ECR container image | AWS account | Critical ▼ | All |
|---|---|---|---|---|---|
| latest | vuln-images/vulhub/fastj | sha256:7f610374bd1fdf5 | [redacted] | 57 | 265 |
| latest | vuln-images/vulhub/webl | sha256:8ddf63df92426e5 | [redacted] | 45 | 455 |
| latest | vuln-images/vulhub/thinl | sha256:aa12db1b982c92 | [redacted] | 14 | 332 |

# ECR Scan results in Inspector – by container image



## Findings: By container image Info

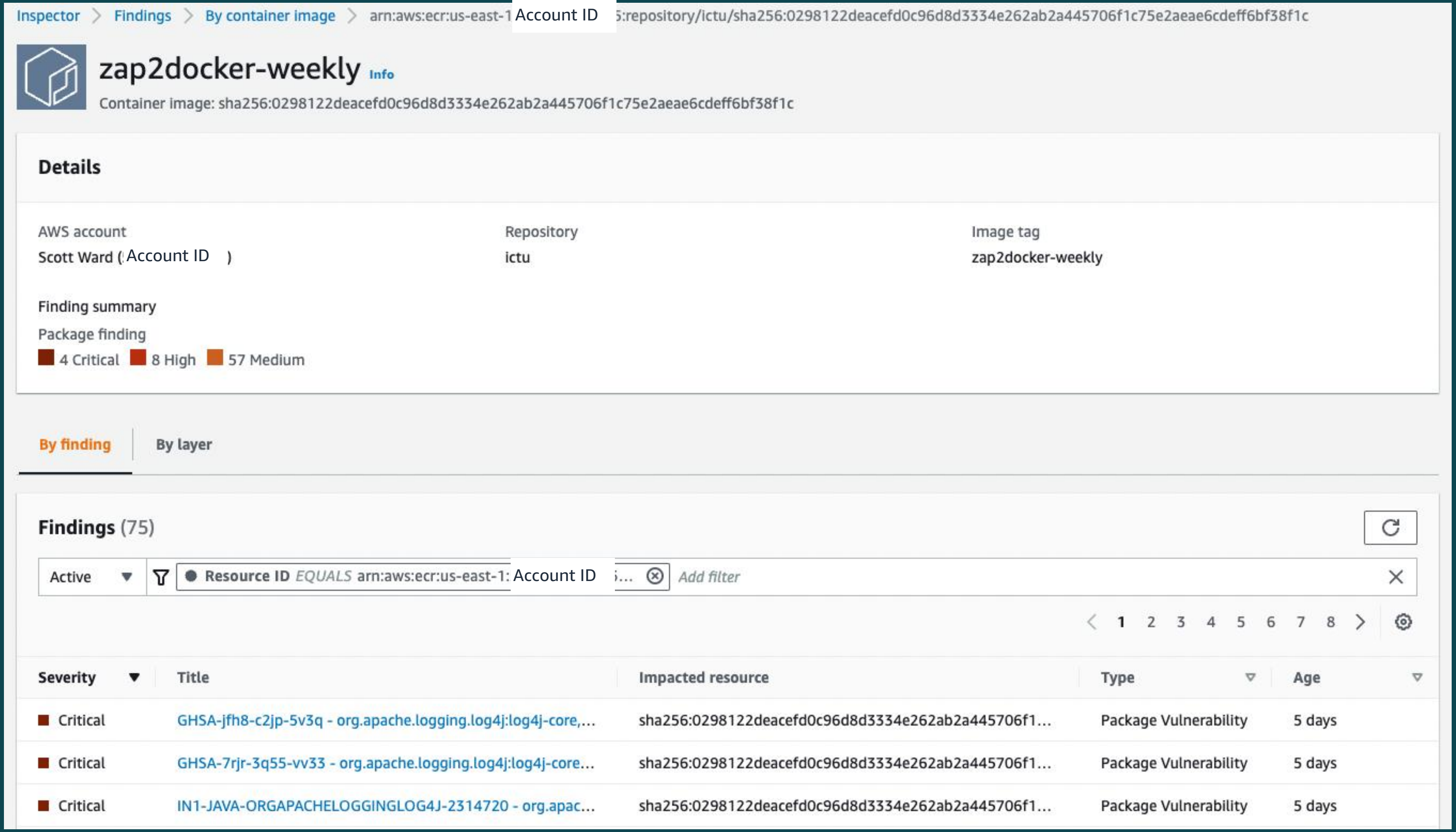
Sorted by container images with the most critical findings.

### By container image (1)

Choose a row to view the container image's details and associated findings

| Image tag | Repository | Image | AWS account | Critical | High | All |
|---|---|---|---|---|---|---|
| zap2docker-weekly | ictu | sha256:0298122deacefd0c9… | Scott Ward ( Account ID ) | 4 | 8 | 75 |

# ECR Scan results in Inspector – container image findings



Inspector > Findings > By container image > arn:aws:ecr:us-east-1 Account ID 5:repository/ictu/sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

## zap2docker-weekly Info

Container image: sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

### Details

| | | |
|---|---|---|
| AWS account | Repository | Image tag |
| Scott Ward ( Account ID ) | ictu | zap2docker-weekly |

Finding summary

Package finding

■ 4 Critical ■ 8 High ■ 57 Medium

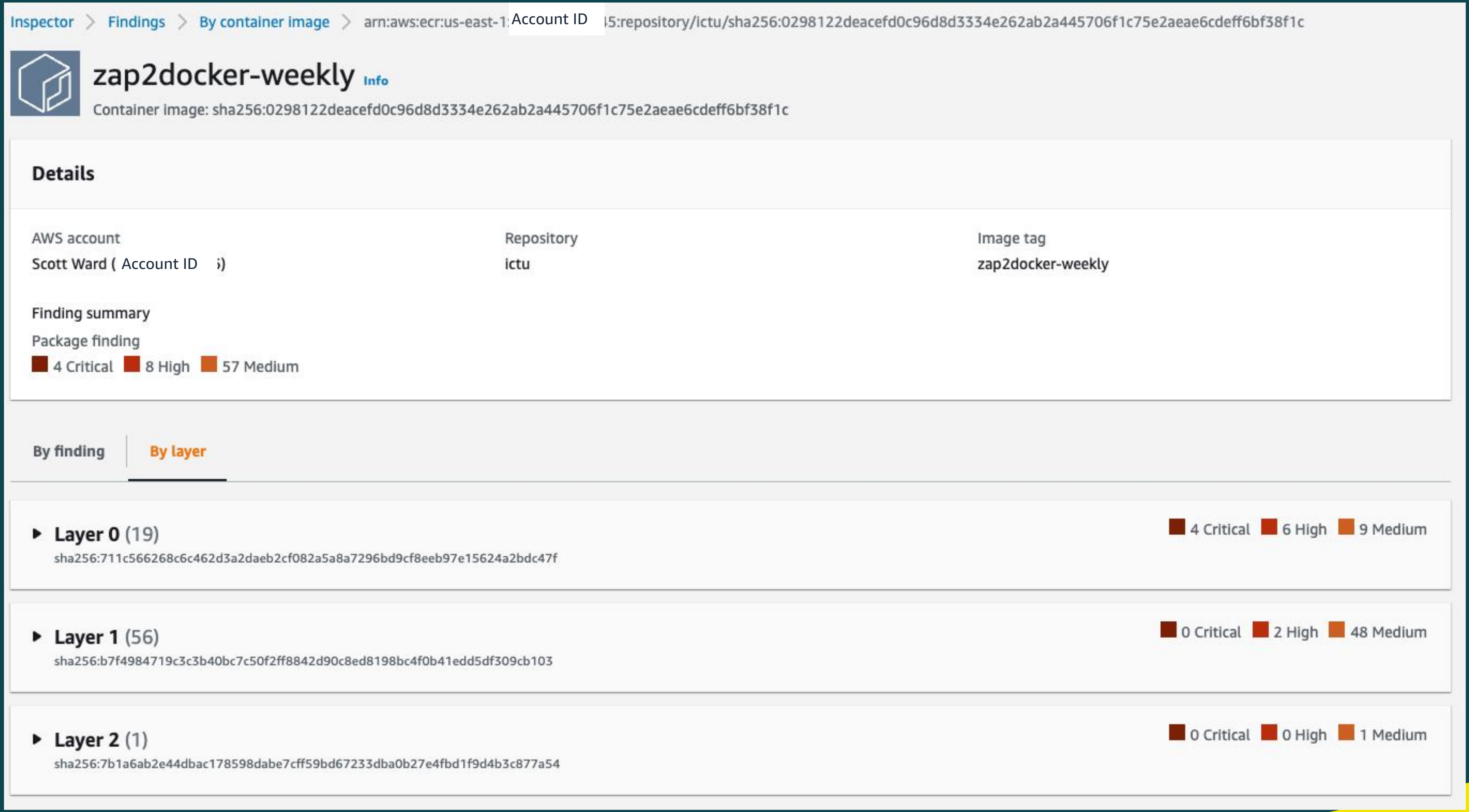
### Findings (75)

Active ▼ | Resource ID *EQUALS* arn:aws:ecr:us-east-1: Account ID 5... ⊗ | *Add filter* ✕

< 1 2 3 4 5 6 7 8 >

| Severity ▼ | Title | Impacted resource | Type | Age |
|---|---|---|---|---|
| ■ Critical | GHSA-jfh8-c2jp-5v3q - org.apache.logging.log4j:log4j-core,... | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1... | Package Vulnerability | 5 days |
| ■ Critical | GHSA-7rjr-3q55-vv33 - org.apache.logging.log4j:log4j-core... | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1... | Package Vulnerability | 5 days |
| ■ Critical | IN1-JAVA-ORGAPACHELOGGINGLOG4J-2314720 - org.apac... | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1... | Package Vulnerability | 5 days |

# ECR Scan results in Inspector – image layer findings



Inspector > Findings > By container image > arn:aws:ecr:us-east-1:Account ID 5:repository/ictu/sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

## zap2docker-weekly Info

Container image: sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

### Details

| AWS account | Repository | Image tag |
|---|---|---|
| Scott Ward ( Account ID 5) | ictu | zap2docker-weekly |

Finding summary

Package finding

■ 4 Critical ■ 8 High ■ 57 Medium

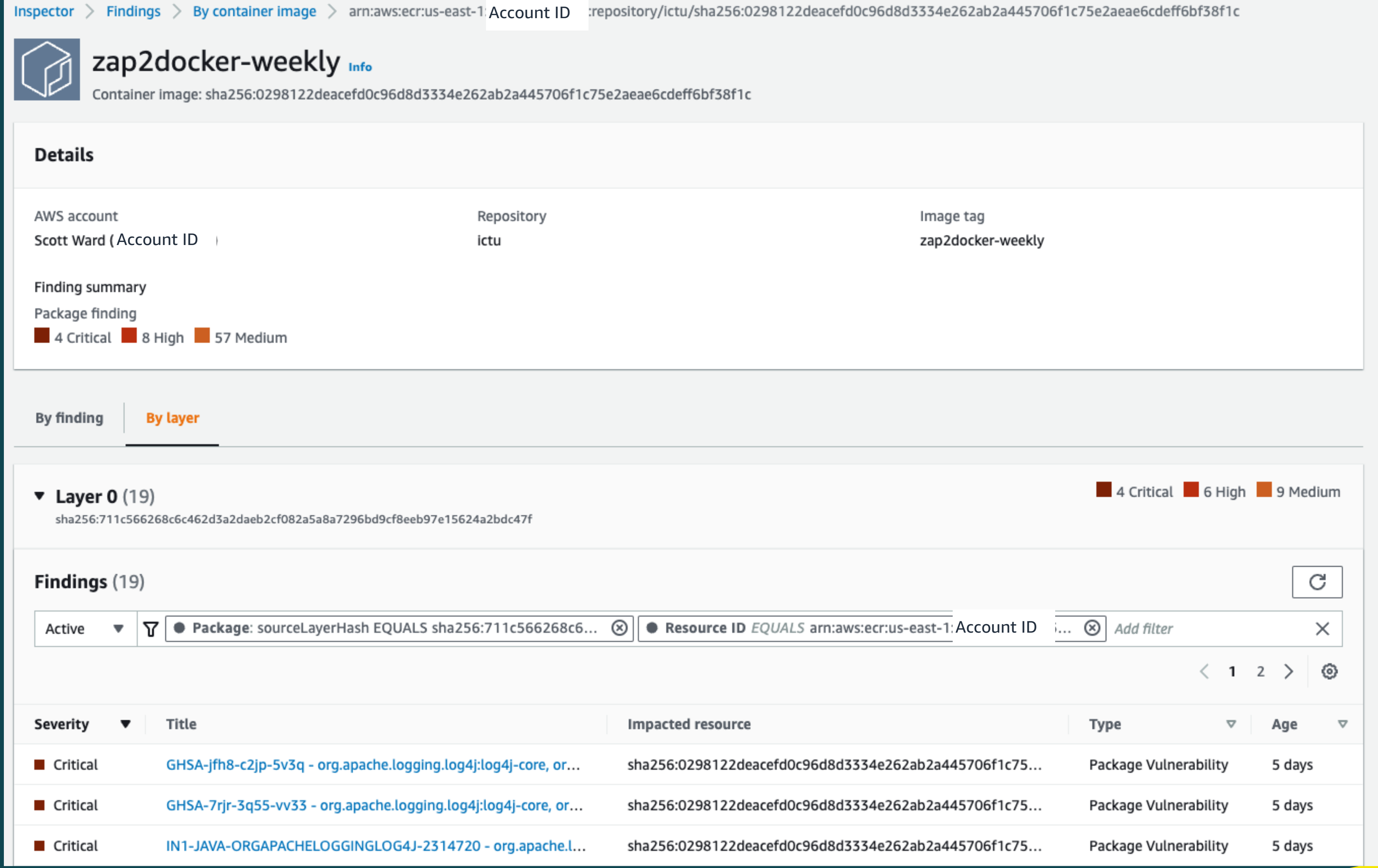
By finding | By layer

▸ **Layer 0** (19)
sha256:711c566268c6c462d3a2daeb2cf082a5a8a7296bd9cf8eeb97e15624a2bdc47f
■ 4 Critical ■ 6 High ■ 9 Medium

▸ **Layer 1** (56)
sha256:b7f4984719c3c3b40bc7c50f2ff8842d90c8ed8198bc4f0b41edd5df309cb103
■ 0 Critical ■ 2 High ■ 48 Medium

▸ **Layer 2** (1)
sha256:7b1a6ab2e44dbac178598dabe7cff59bd67233dba0b27e4fbd1f9d4b3c877a54
■ 0 Critical ■ 0 High ■ 1 Medium

# ECR Scan results in Inspector – image layer findings



Inspector > Findings > By container image > arn:aws:ecr:us-east-1 Account ID :repository/ictu/sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

## zap2docker-weekly Info

Container image: sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75e2aeae6cdeff6bf38f1c

### Details

| AWS account | Repository | Image tag |
|---|---|---|
| Scott Ward ( Account ID | ictu | zap2docker-weekly |

Finding summary

Package finding

🟥 4 Critical 🟥 8 High 🟧 57 Medium

By finding | **By layer**

▼ **Layer 0** (19)

sha256:711c566268c6c462d3a2daeb2cf082a5a8a7296bd9cf8eeb97e15624a2bdc47f

🟥 4 Critical 🟥 6 High 🟧 9 Medium

**Findings** (19)

Active ▼ | ● **Package**: sourceLayerHash EQUALS sha256:711c566268c6… ⊗ | ● **Resource ID** *EQUALS* arn:aws:ecr:us-east-1: Account ID … ⊗ | *Add filter* ✕

< 1 2 >

| Severity ▼ | Title | Impacted resource | Type | Age |
|---|---|---|---|---|
| 🟥 Critical | GHSA-jfh8-c2jp-5v3q - org.apache.logging.log4j:log4j-core, or… | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75… | Package Vulnerability | 5 days |
| 🟥 Critical | GHSA-7rjr-3q55-vv33 - org.apache.logging.log4j:log4j-core, or… | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75… | Package Vulnerability | 5 days |
| 🟥 Critical | IN1-JAVA-ORGAPACHELOGGINGLOG4J-2314720 - org.apache.l… | sha256:0298122deacefd0c96d8d3334e262ab2a445706f1c75… | Package Vulnerability | 5 days |

# Remediating Inspector container findings

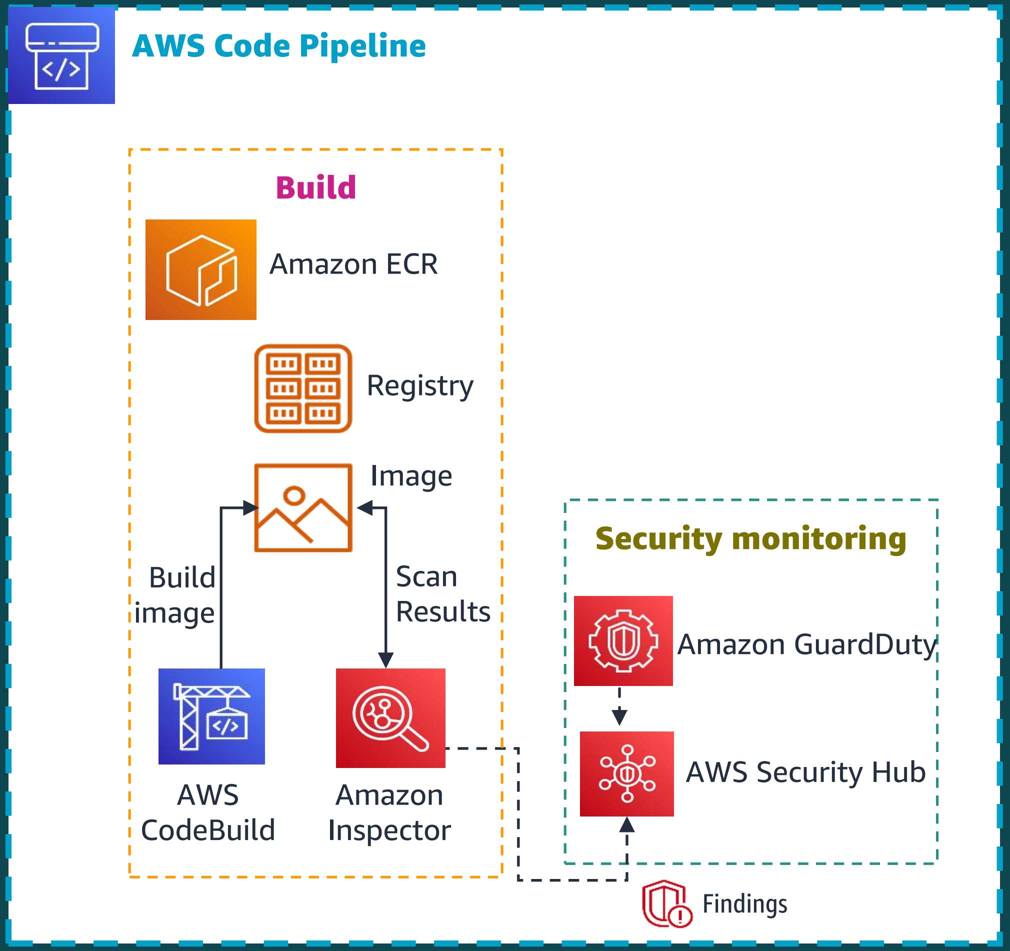
Delete the image in the ECR repository

- Closes associated findings

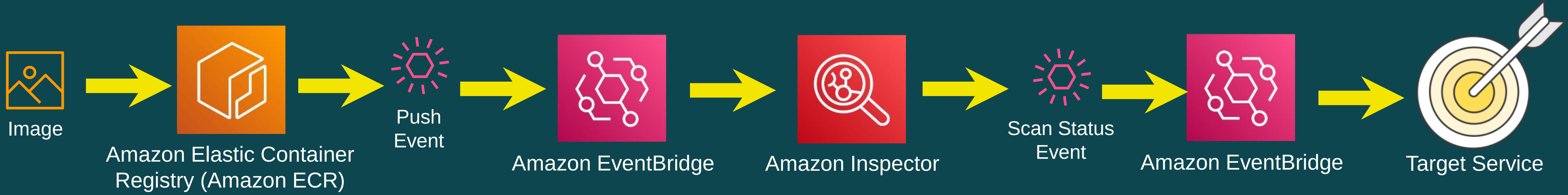
Publish an updated image

# Using Inspector and container image scans

## Build Stage



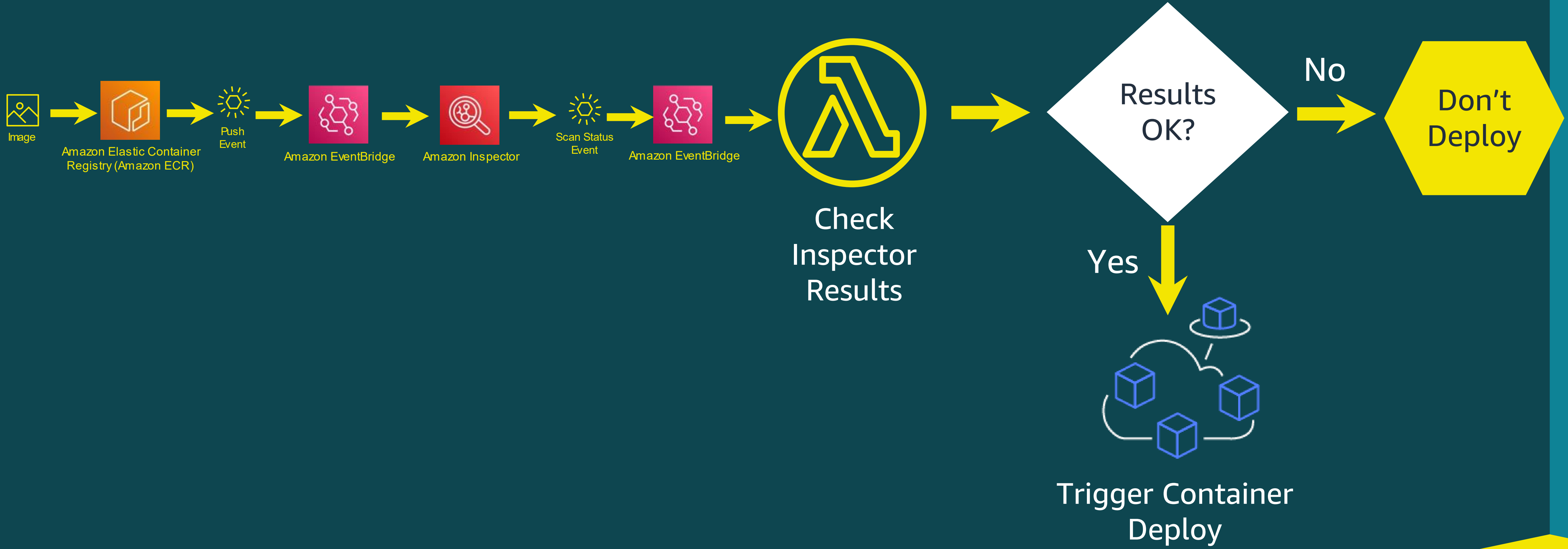
# Inspector notifications for container image scans



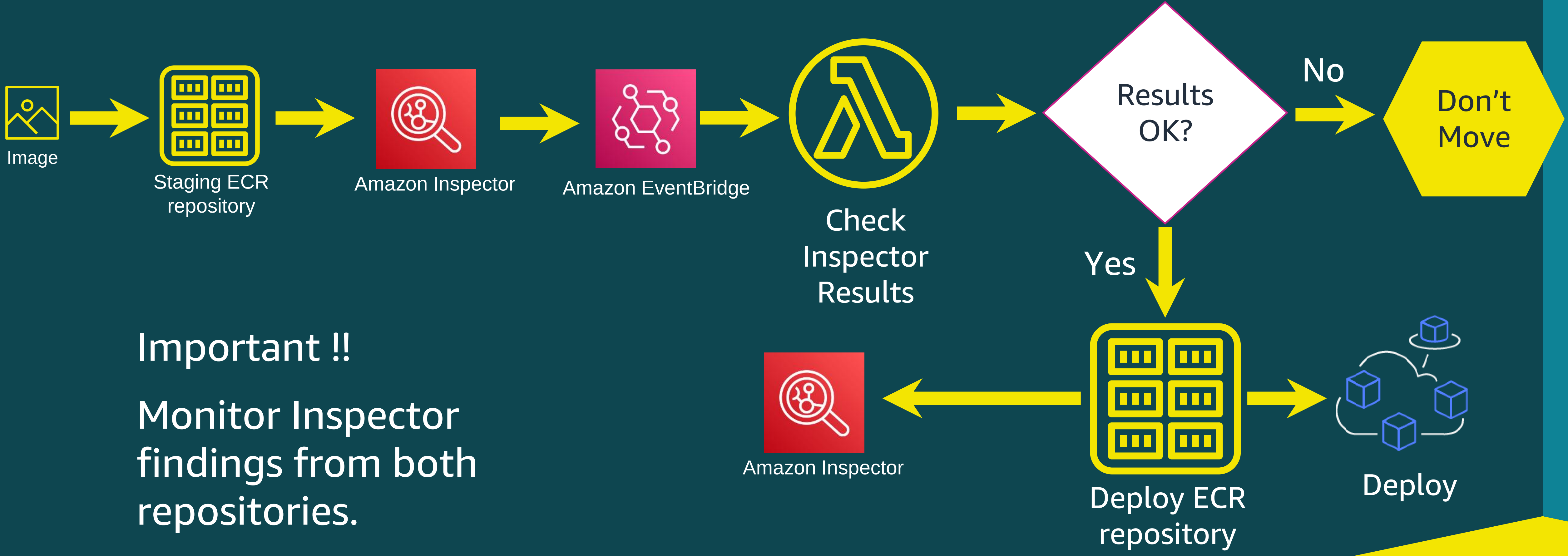
# Inspector container scan events

```
{
    "version": "0",
    "id": "20d26d65-e3a7-14d5-9e27-e71a63f889ad",
    "detail-type": "Inspector2 Coverage",
    "source": "aws.inspector2",
    "account": "<account ID>",
    "time": "2022-01-21T21:14:49Z",
    "region": "us-east-1",
    "resources":
    [
        "arn:aws:ecr:us-east-1:<account ID>:repository/ictu/sha256:0298122deacefd0cxxx"
    ],
    "detail":
    {
        "scanStatus":
        {
            "reason": "SUCCESSFUL",
            "statusCodeValue": "ACTIVE"
        },
        "eventTimestamp": "2022-01-21T21:14:44.588013Z"
    }
}
```

# Deploying containers post Inspector scan – verify and deploy



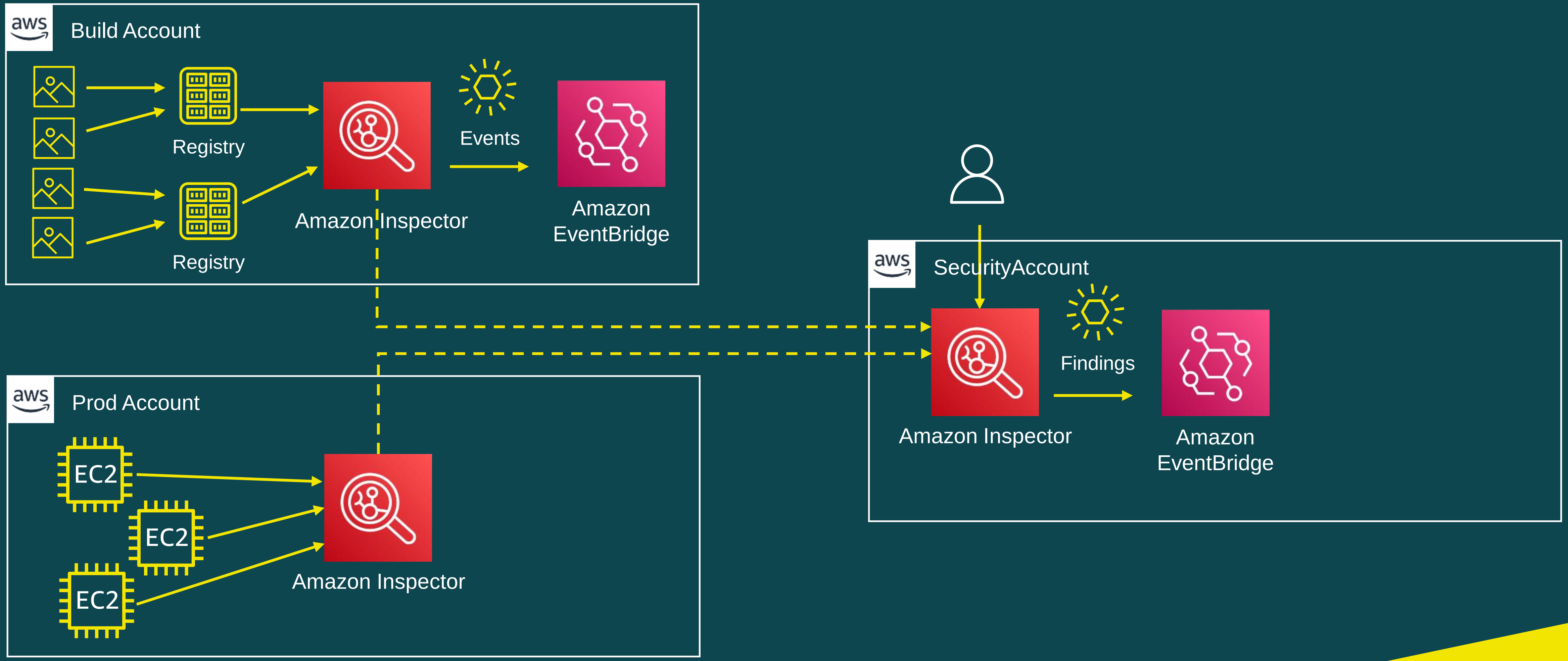
# Inspector container scanning – stage to deploy



Important !!

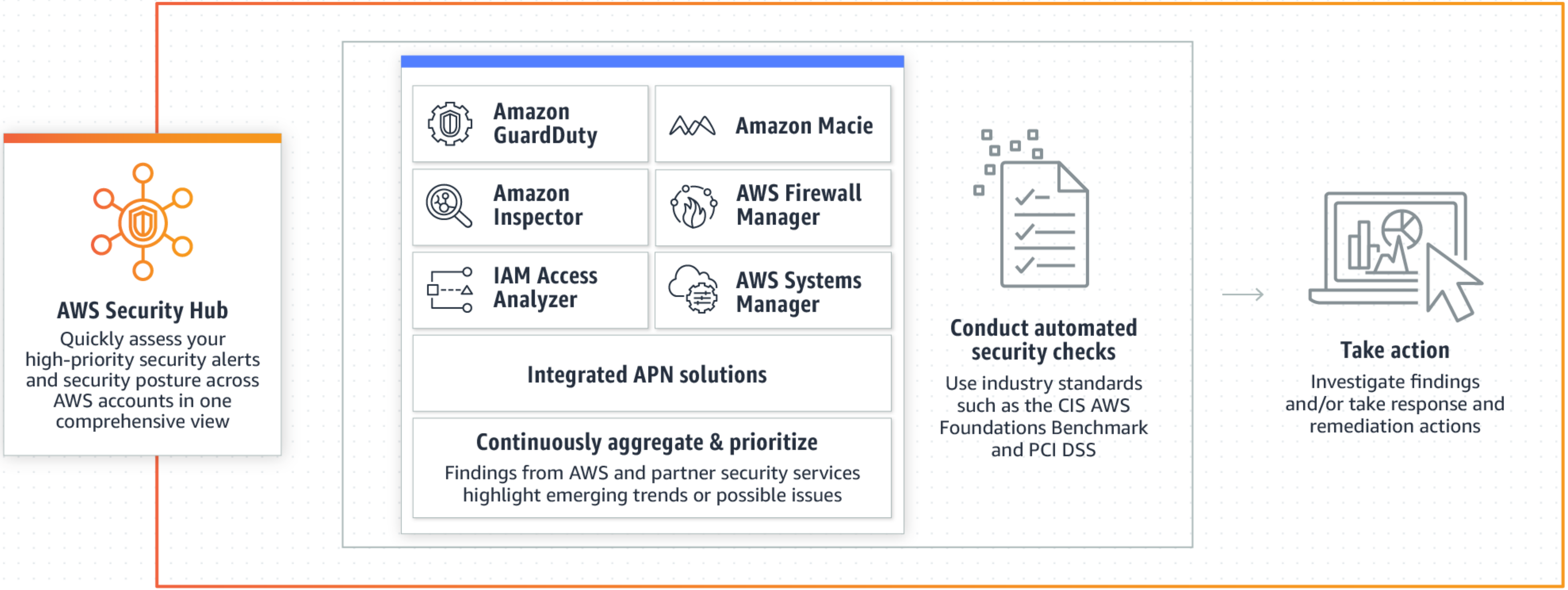
Monitor Inspector findings from both repositories.

# How can you use Inspector and container image scans – Multi Account



# AWS Security Hub

# Security Hub with GuardDuty and Inspector



**AWS Security Hub**

Quickly assess your high-priority security alerts and security posture across AWS accounts in one comprehensive view

Amazon GuardDuty

Amazon Macie

Amazon Inspector

AWS Firewall Manager

IAM Access Analyzer

AWS Systems Manager

Integrated APN solutions

**Continuously aggregate & prioritize**

Findings from AWS and partner security services highlight emerging trends or possible issues

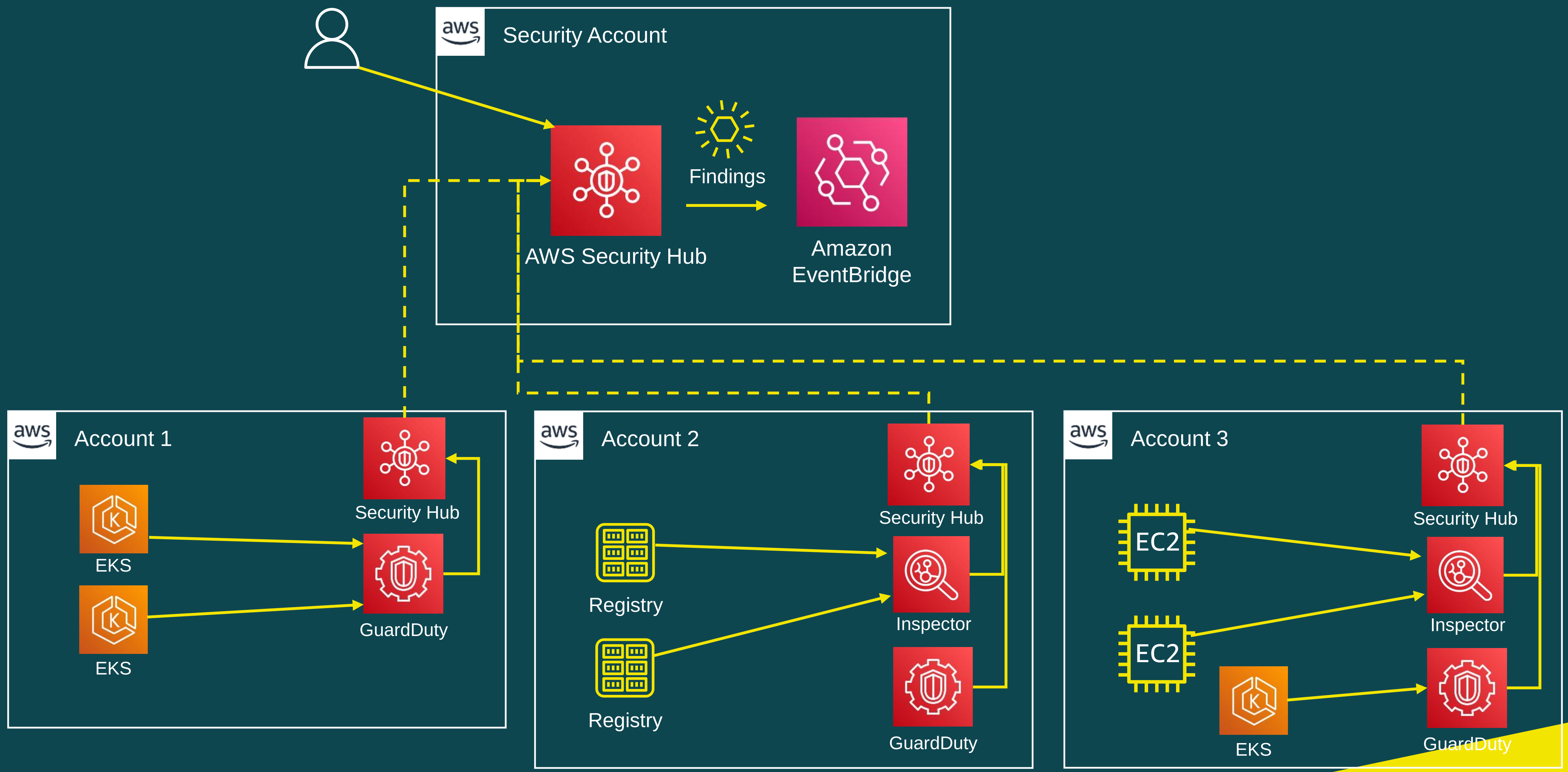
**Conduct automated security checks**

Use industry standards such as the CIS AWS Foundations Benchmark and PCI DSS

**Take action**

Investigate findings and/or take response and remediation actions

# Security Hub, Inspector, & GuardDuty – Multiple Accounts



# Using Security Hub to aggregate and prioritize

- View findings across multiple accounts and regions
- View findings for one resource from multiple sources
- Include findings from 3rd party products and your custom checks
- Security standards to help confirm best practice compliance

# Thank you!