AWS

# Effective Vulnerability Management Strategy

**Purnaresa Yuliartanto**

Sr Security Solutions Architect
AWS ASEAN

**Kovan Chandra**

Technical Account Manager – Security Field
AWS Indonesia

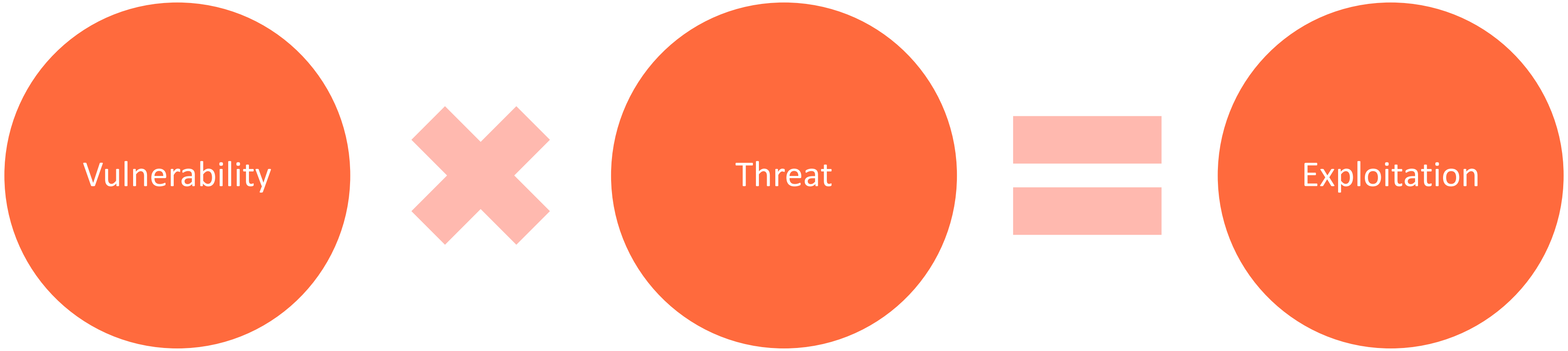
# Component of Cybersecurity Incident

Vulnerability

- Weakness or flaw in a system, network, or application

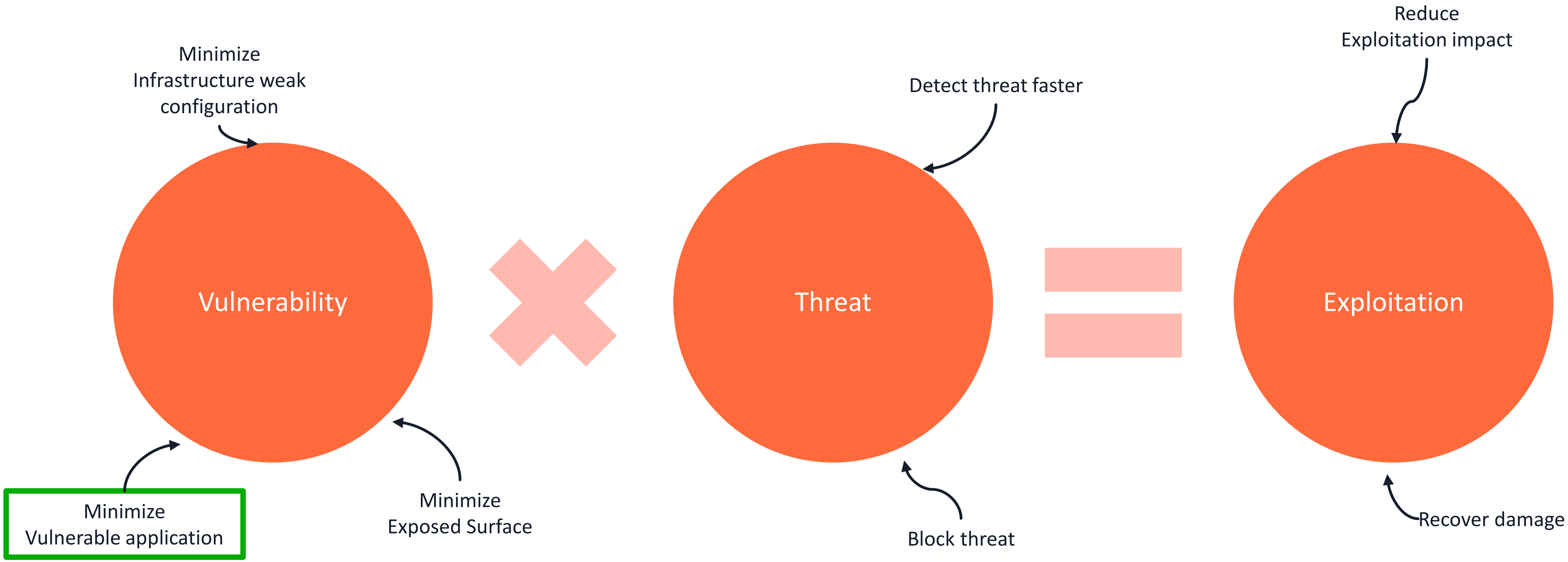
Threat

- Any potential danger to an organization's assets, data, or systems



Vulnerability × Threat = Exploitation

# Component of Cybersecurity Incident



Minimize
Infrastructure weak
configuration

Vulnerability

Minimize
Vulnerable application

Minimize
Exposed Surface

Detect threat faster

Threat

Block threat

Reduce
Exploitation impact

Exploitation

Recover damage

# Security Capabilities on AWS

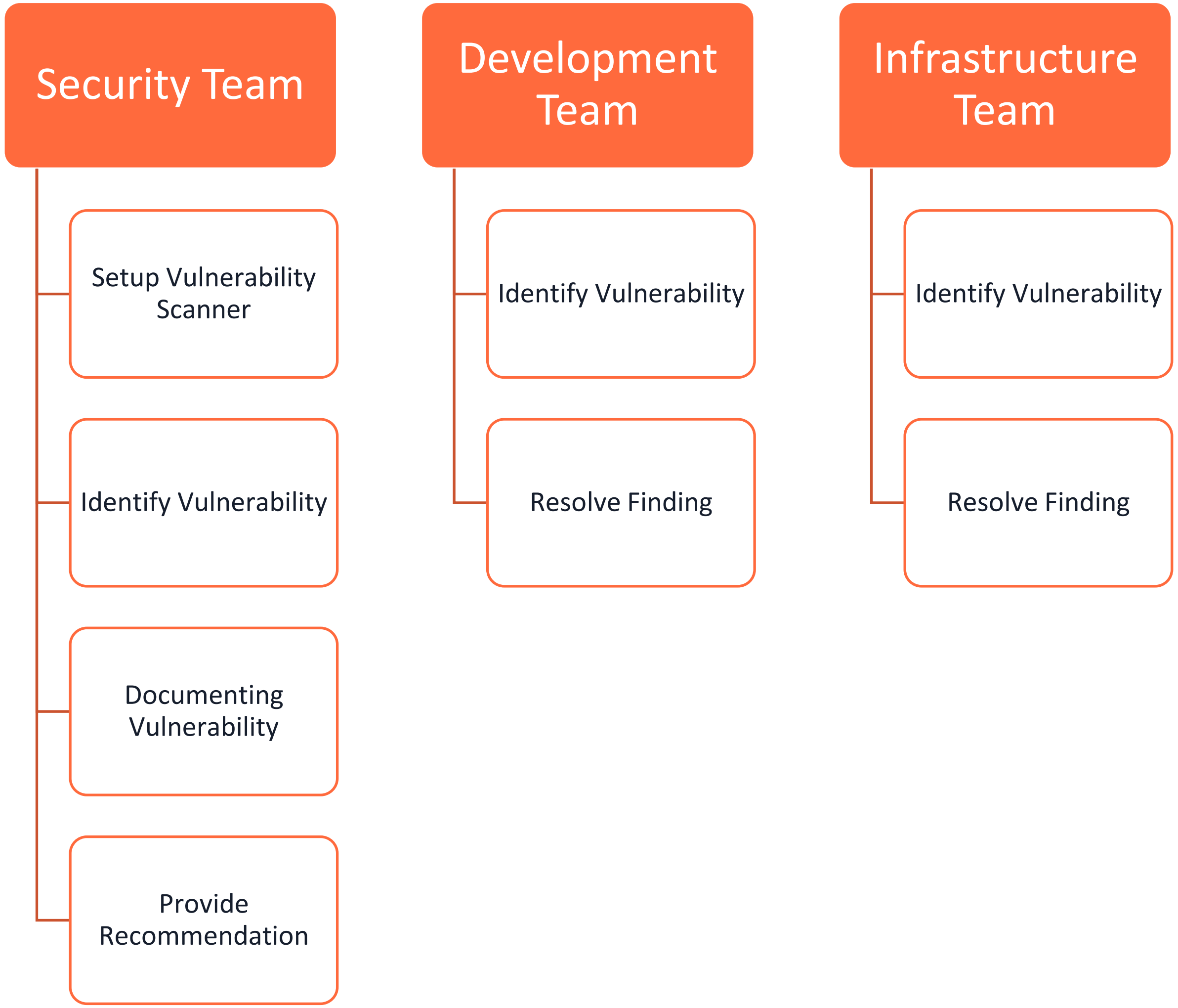
Based on the AWS Cloud Adoption Framework (CAF), the following capabilities can help you achieve confidentiality, integrity, and availability for your data and workloads:

| | | |
|---|---|---|
| Security governance | Security assurance | Identity and access management |
| Vulnerability management | Infrastructure protection | Data protection |
| Application security | Threat detection | Incident response |

Whitepaper: https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-caf-security-perspective/aws-caf-security-perspective.pdf

# Vulnerability Management Goal

Collectively work to reduce the vulnerability that could be targeted by threat actor


Security Team

Setup Vulnerability Scanner

Identify Vulnerability

Documenting Vulnerability

Provide Recommendation

Development Team

Identify Vulnerability

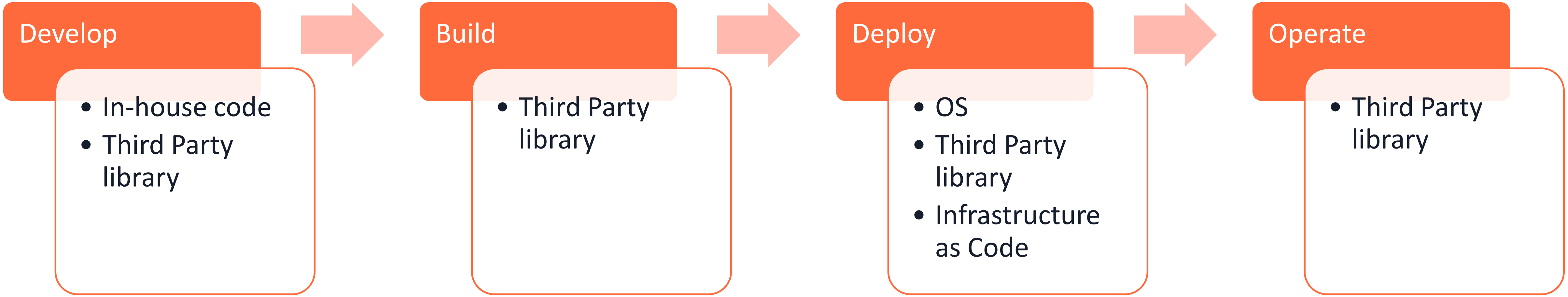
Resolve Finding

Infrastructure Team

Identify Vulnerability

Resolve Finding

# Software Lifecycle Components



Develop

- In-house code
- Third Party library

Build

- Third Party library

Deploy

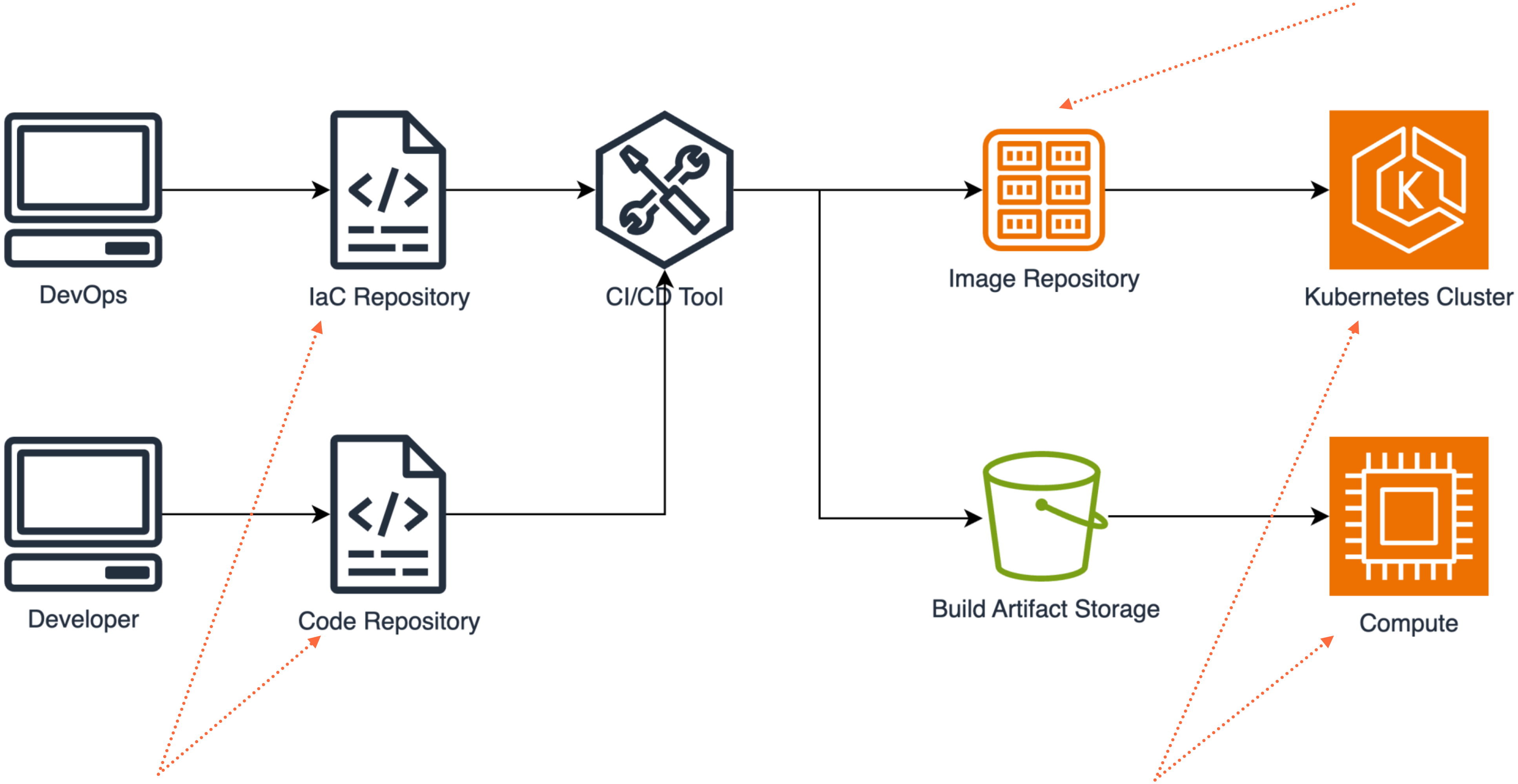
- OS
- Third Party library
- Infrastructure as Code

Operate

- Third Party library

# Where we should conduct vulnerability scan

2. Image Scan

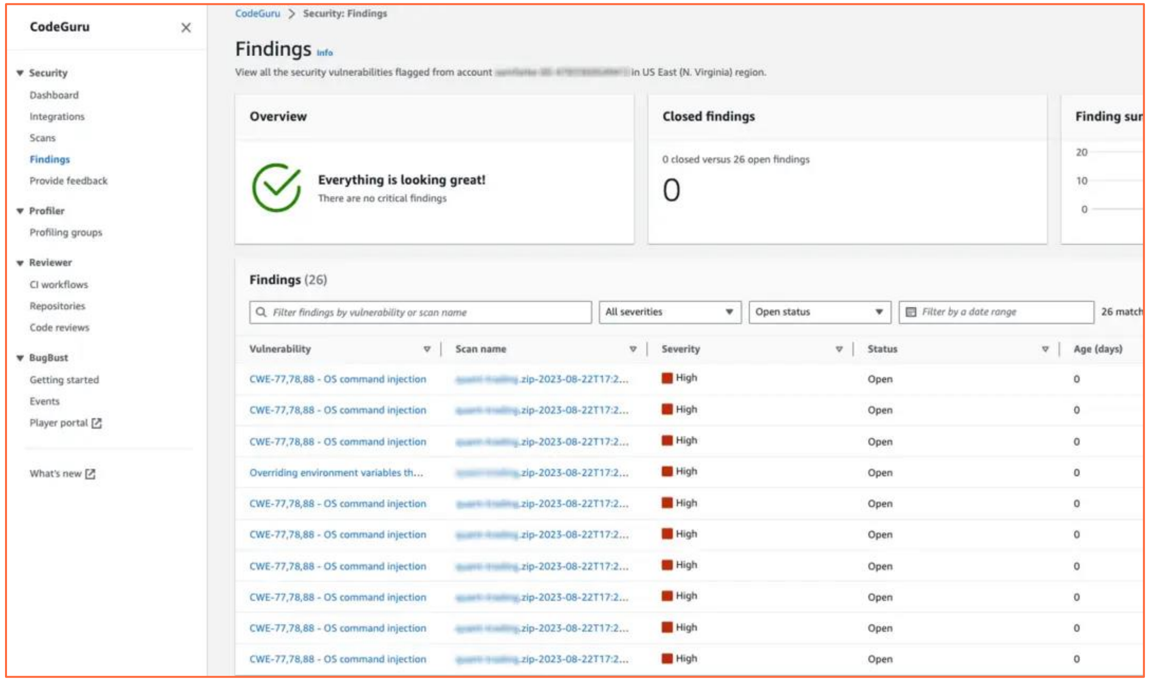


1. Source Code Scan

3. Runtime Scan

# 1. Code Scanning Best Practice

- Automated Scanning on Pull/Merge Requests
- Mandatory Security Gates for Code Merging
- Secrets and Credentials Scanning
- Tool Selection Based on Technology Stack
  - AWS Service: CodeGuru Security
  - AWS Partner: CheckMarx, GitLab, Veracode, …
  - OpenSource: AWSLabs/ASH, SonarQube, …



Findings

View all the security vulnerabilities flagged from account [redacted] in US East (N. Virginia) region.

Overview

Everything is looking great!

There are no critical findings

Closed findings

0 closed versus 26 open findings

0

Finding sur

Findings (26)

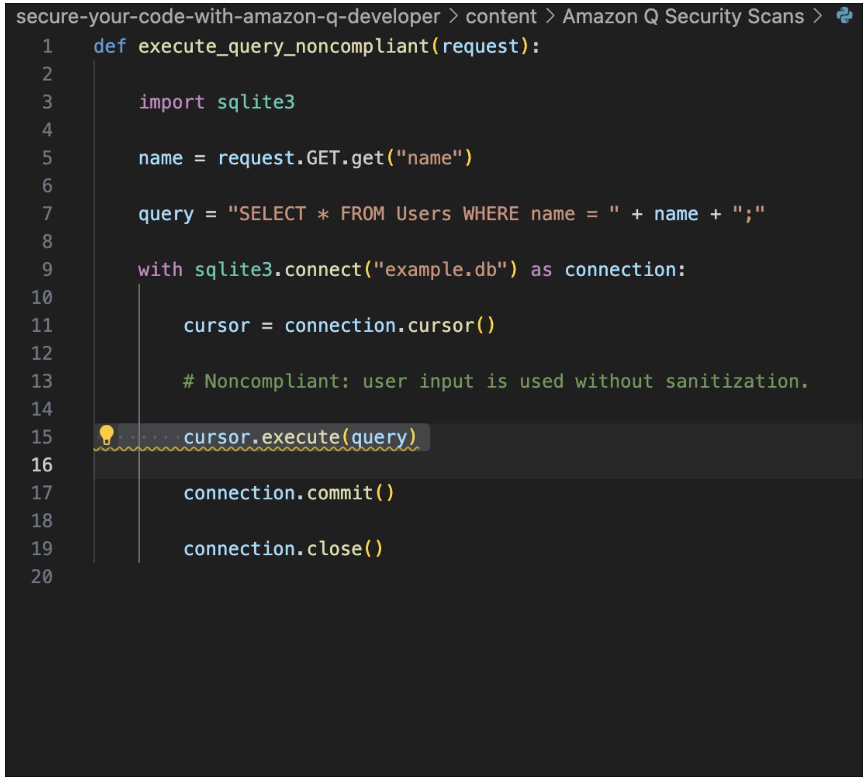
Filter findings by vulnerability or scan name

26 match

| Vulnerability | Scan name | Severity | Status | Age (days) |
|---|---|---|---|---|
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| Overriding environment variables th... | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |
| CWE-77,78,88 - OS command injection | [redacted].zip-2023-08-22T17:2... | 🟥 High | Open | 0 |

# Detect Vulnerability Earlier

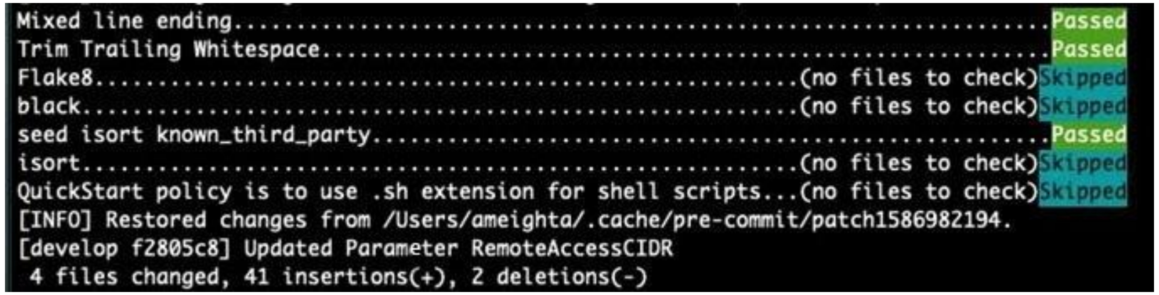
- Implement scanning in developer IDEs
  - Utilize Gen-AI based code development



secure-your-code-with-amazon-q-developer > content > Amazon Q Security Scans >

```python
def execute_query_noncompliant(request):

    import sqlite3

    name = request.GET.get("name")

    query = "SELECT * FROM Users WHERE name = " + name + ";"

    with sqlite3.connect("example.db") as connection:

        cursor = connection.cursor()

        # Noncompliant: user input is used without sanitization.

        cursor.execute(query)

        connection.commit()

        connection.close()

```

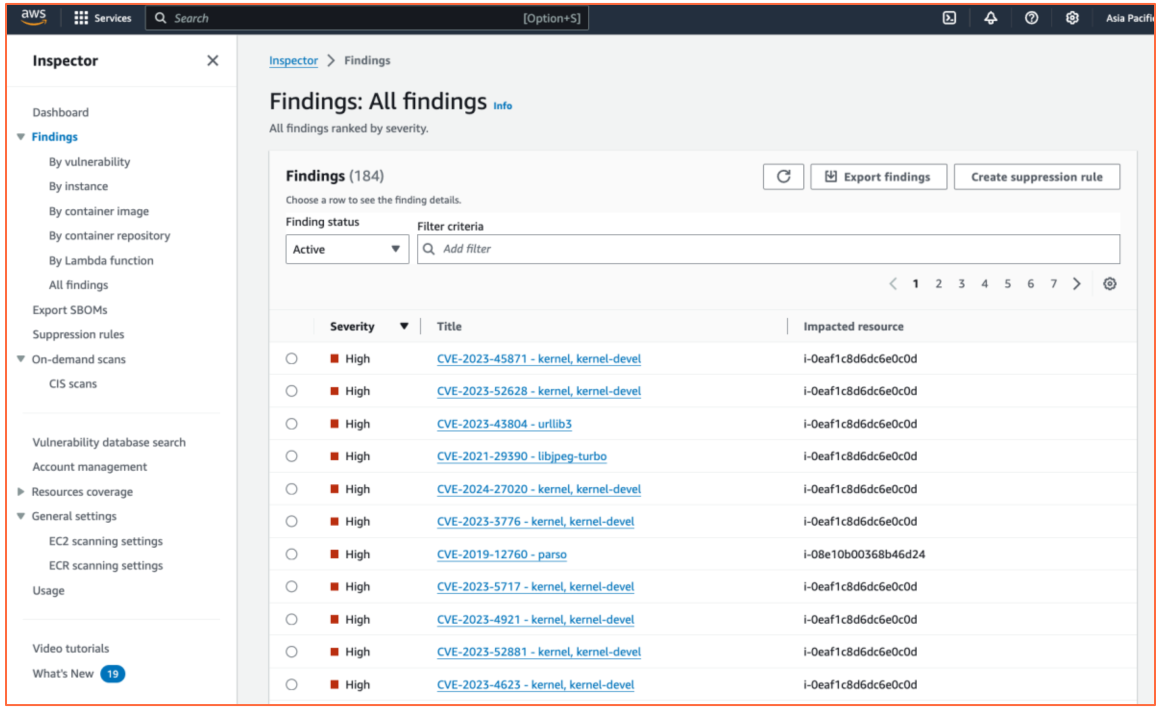
- Set up pre-commit hooks
  - Scan before code uploaded into code Repository



```
Mixed line ending......................................................Passed
Trim Trailing Whitespace...............................................Passed
Flake8.........................................(no files to check)Skipped
black..........................................(no files to check)Skipped
seed isort known_third_party...........................................Passed
isort..........................................(no files to check)Skipped
QuickStart policy is to use .sh extension for shell scripts...(no files to check)Skipped
[INFO] Restored changes from /Users/ameighta/.cache/pre-commit/patch1586982194.
[develop f2805c8] Updated Parameter RemoteAccessCIDR
 4 files changed, 41 insertions(+), 2 deletions(-)
```

# 2. Image Scanning Best Practice

- Automate Scanning on New Image
- Secrets and Credentials Scanning
- Mandatory Security Gates for Image Deployment
  - Speed concern – filter based on environment
- Tool Selection Based on Technology Stack
  - AWS Service: Amazon ECR, Amazon Inspector
  - AWS Partner: Crowdstrike, Trend Mikro, ...
  - OpenSource: Trivy, ...



Findings: All findings

All findings ranked by severity.

Findings (184)

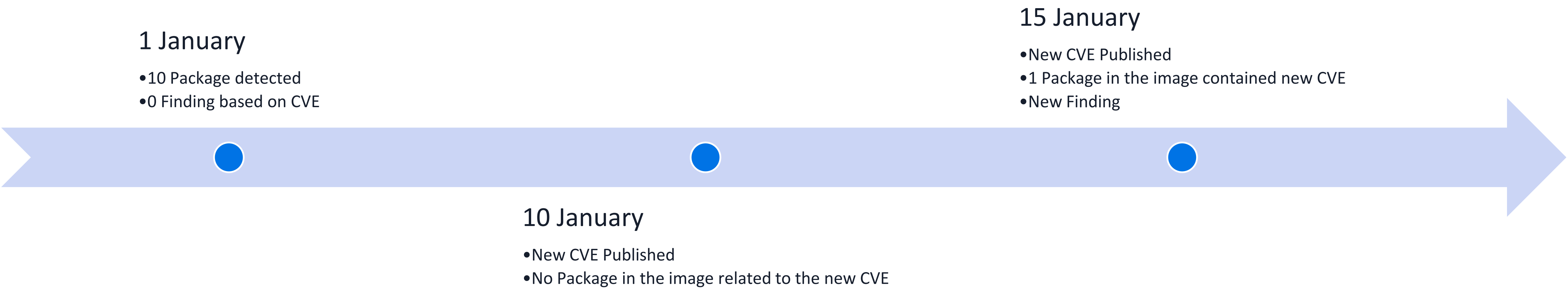
Choose a row to see the finding details.

Finding status: Active

Filter criteria: Add filter

| | Severity | Title | Impacted resource |
|---|---|---|---|
| ○ | 🟥 High | CVE-2023-45871 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-52628 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-43804 - urllib3 | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2021-29390 - libjpeg-turbo | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2024-27020 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-3776 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2019-12760 - parso | i-08e10b00368b46d24 |
| ○ | 🟥 High | CVE-2023-5717 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-4921 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-52881 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |
| ○ | 🟥 High | CVE-2023-4623 - kernel, kernel-devel | i-0eaf1c8d6dc6e0c0d |

# Why Continuously Scan?



1 January

- 10 Package detected
- 0 Finding based on CVE

10 January

- New CVE Published
- No Package in the image related to the new CVE

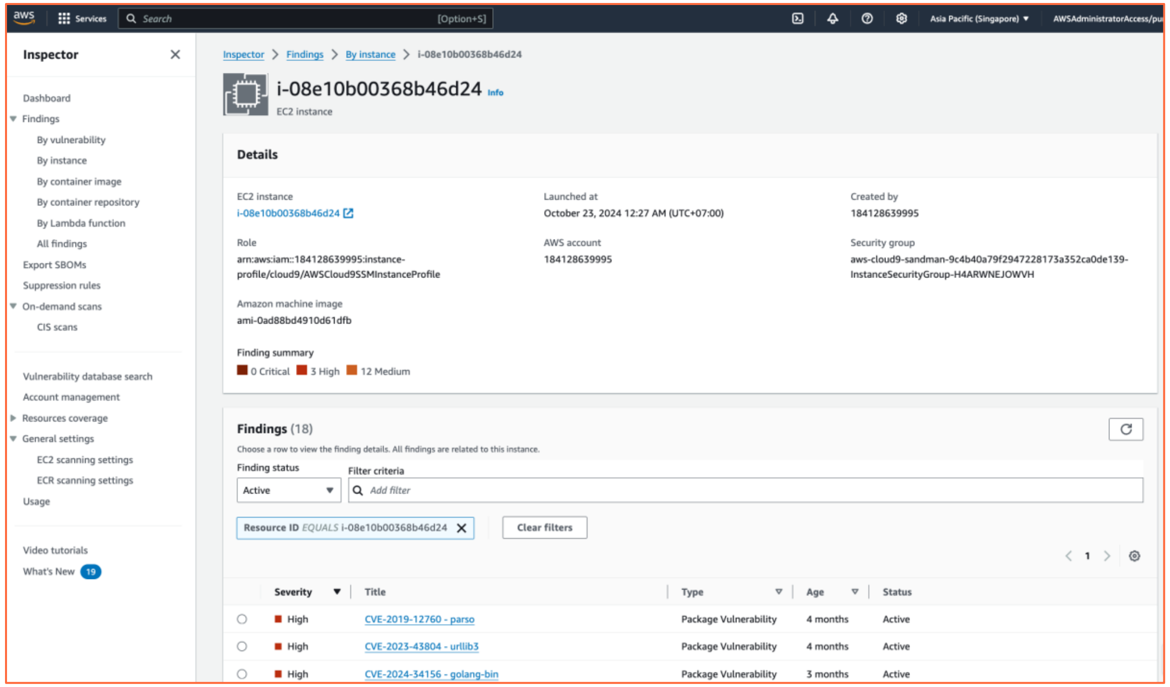
15 January

- New CVE Published
- 1 Package in the image contained new CVE
- New Finding

Ensuring your deployments remain secure against
emerging threats and reducing the risk of deploying vulnerable containers into production.

# 3. Runtime Scan Best Practice

- Apply on all type of workload
  - Compute on EC2
  - Compute on Lambda
  - Container on EKS
- Focus on Scan coverage to avoid unmonitored workload
- Tool Selection Based on Technology Stack
  - AWS Service: Amazon Inspector
  - AWS Partner: Rapid7, Tenable, ...
  - OpenSource: Faraday, ...



i-08e10b00368b46d24

EC2 instance

**Details**

EC2 instance: i-08e10b00368b46d24

Launched at: October 23, 2024 12:27 AM (UTC+07:00)

Created by: 184128639995

Role: arn:aws:iam::184128639995:instance-profile/cloud9/AWSCloud9SSMInstanceProfile

AWS account: 184128639995

Security group: aws-cloud9-sandman-9c4b40a79f2947228173a352ca0de139-InstanceSecurityGroup-H4ARWNEJOWVH

Amazon machine image: ami-0ad88bd4910d61dfb

Finding summary: 0 Critical, 3 High, 12 Medium

**Findings** (18)

Choose a row to view the finding details. All findings are related to this instance.

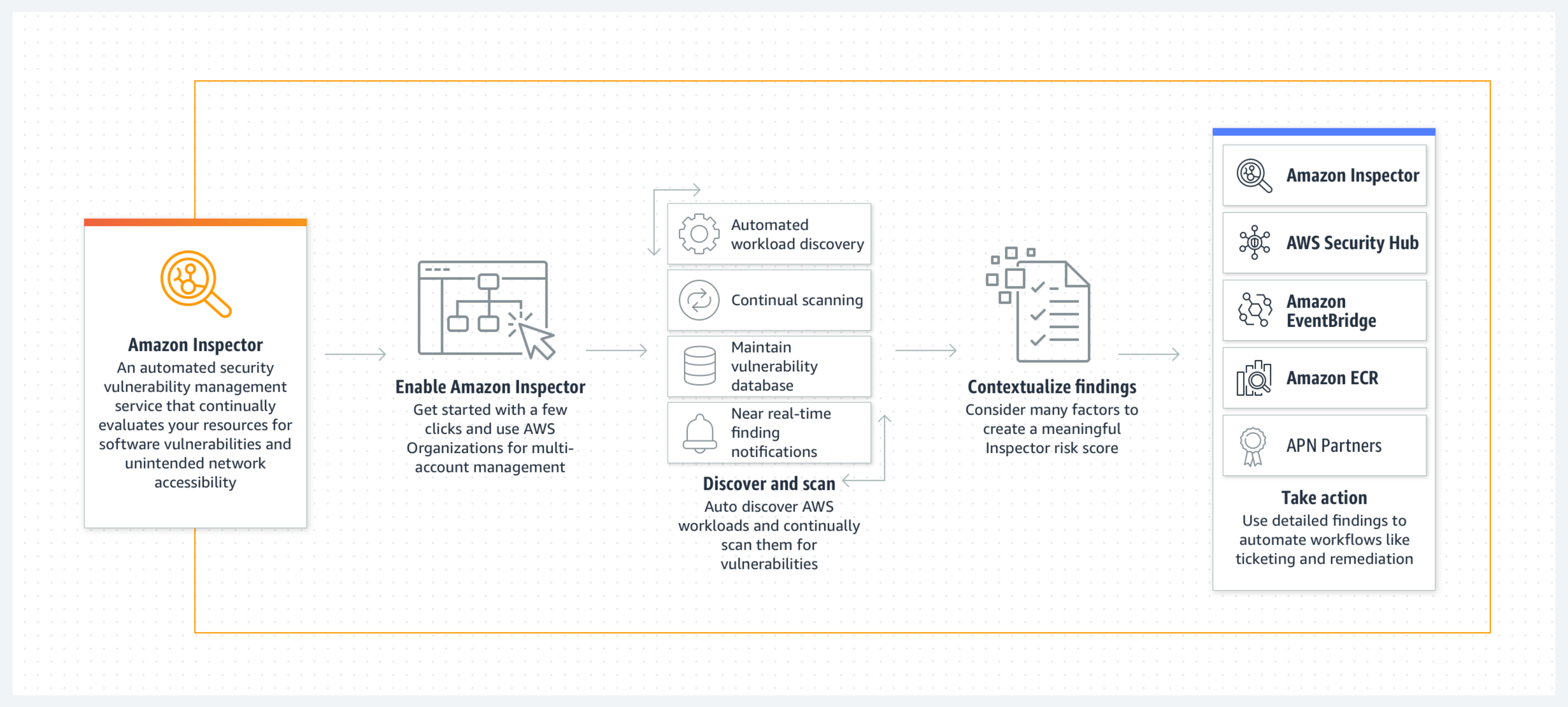
Finding status: Active

Filter criteria: Add filter

Resource ID *EQUALS* i-08e10b00368b46d24

| | Severity | Title | Type | Age | Status |
|---|---|---|---|---|---|
| ○ | High | CVE-2019-12760 - parso | Package Vulnerability | 4 months | Active |
| ○ | High | CVE-2023-43804 - urllib3 | Package Vulnerability | 4 months | Active |
| ○ | High | CVE-2024-34156 - golang-bin | Package Vulnerability | 3 months | Active |

# Overview of Amazon Inspector



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

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

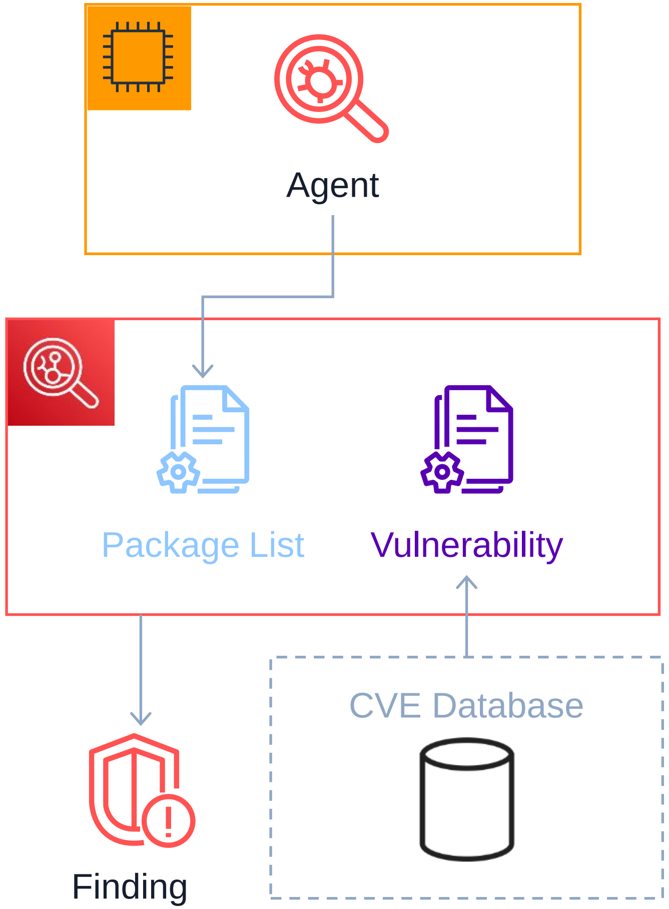
Automated workload discovery
Continual scanning
Maintain vulnerability database
Near real-time finding notifications

**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

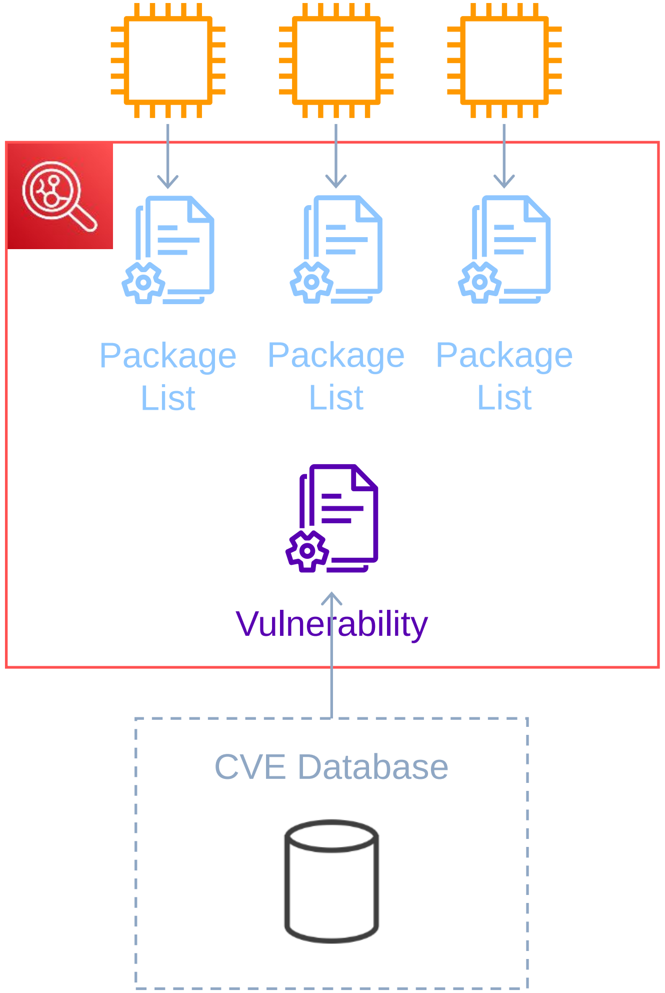
**Take action**
Use detailed findings to automate workflows like ticketing and remediation

# EC2 Scanning - Package Vulnerability



- ✓ Inspector uses inventory data gathered from Systems Manager to determine what is and isn't installed on an instance
- ✓ Inspector correlates individual packages and their versions to known associated CVE's to report a finding
- ✓ When packages are installed or updated on an instance, a new review of the packages is triggered.
- ✓ This kind of scanning happens even when the instance is down, providing visibility into a stopped instance's security posture.

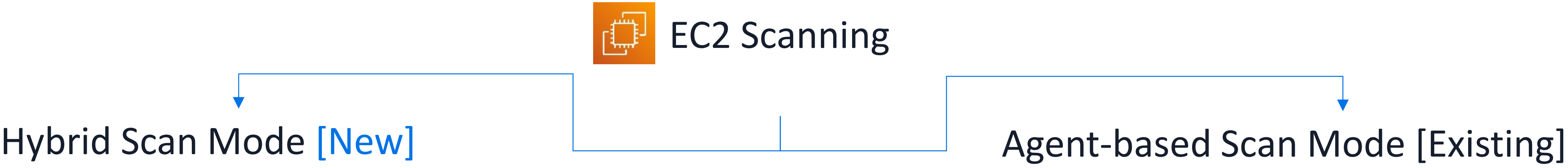
# EC2 Scanning - Package Vulnerability



Similarly, when a new CVE is discovered in one of the many CVE databases Inspector sources its vulnerability data from, a new review of installed packages is triggered on all applicable instances, comparing the data against the refreshed list of vulnerabilities, even if the instance(s) are down.

# Agentless Scanning of EC2 Instances

Continuously monitor your EC2 instances for software vulnerabilities (CVEs)  without installing an agent or additional software



EC2 Scanning

Hybrid Scan Mode [New]

Agent-based Scan Mode [Existing]

If the account is configured to Hybrid scan mode, Inspector relies on SSM agents to perform assessments for instances managed by SSM, but automatically switches to agentless scanning for EC2 instances that do not have SSM agents installed or configured

If the account is configured to Agent-based scan mode, Inspector will only assess instances managed by SSM by leveraging SSM agents

- ➢ For agentless scans, Inspector snapshots EBS volumes to access filesystem data using EBS Direct APIs, but snapshots are never copied outside of your account!

# Container image scanning within CI/CD Tools

Proactively assess your container images during build time within your CI/CD tool before pushing to your container registry or deploying it to production

## Scan images in CI/CD using native plugins [New]

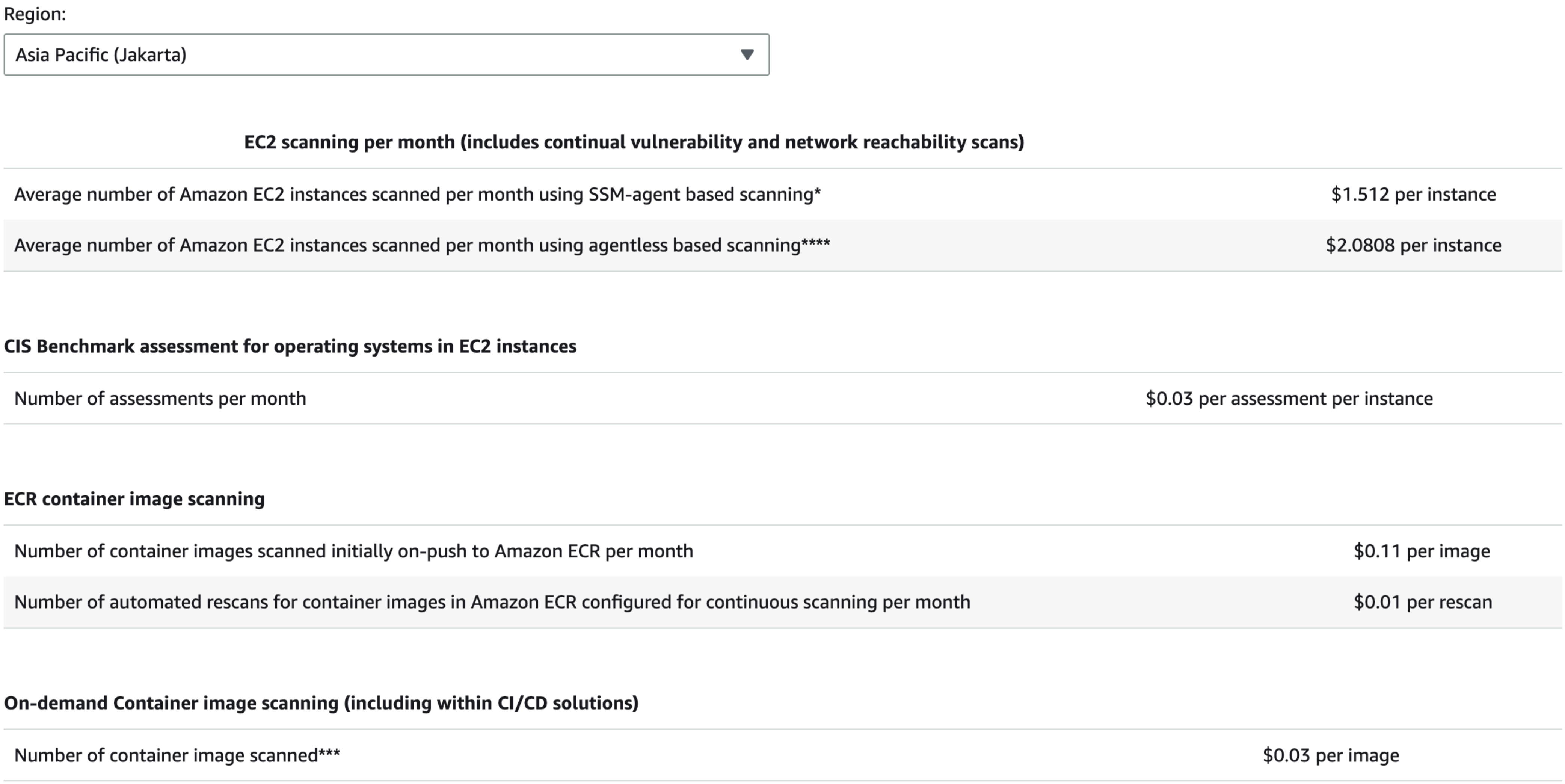
- ✓ Native plugins for Jenkins and TeamCity supported at launch
- ✓ Follow 3 simple steps to make it work
- ✓ Plugins orchestrates the scan workflow

## Continuously monitor your images in Elastic Container registry (ECR) [Existing]

- ✓ Use continuous scanning to monitor your images for zero-day vulnerabilities after pushing to ECR
- ✓ Use on-push scanning to scan images only once upon push to ECR

✓ Your CI/CD solution can be hosted in AWS, hybrid clouds, or on-premises hosts

# Pricing



Region:

Asia Pacific (Jakarta) ▼

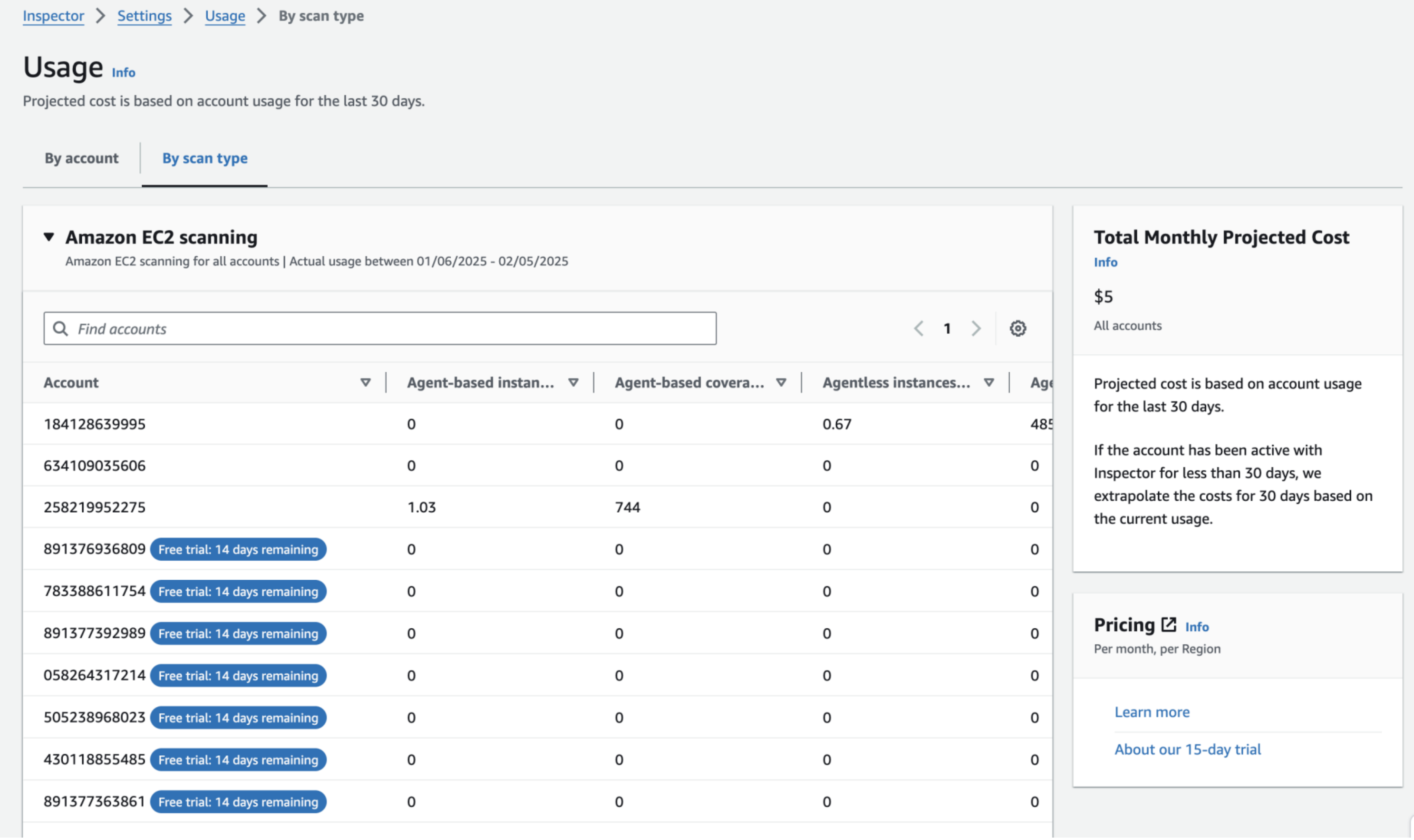
| **EC2 scanning per month (includes continual vulnerability and network reachability scans)** | |
|---|---|
| Average number of Amazon EC2 instances scanned per month using SSM-agent based scanning* | $1.512 per instance |
| Average number of Amazon EC2 instances scanned per month using agentless based scanning**** | $2.0808 per instance |

| **CIS Benchmark assessment for operating systems in EC2 instances** | |
|---|---|
| Number of assessments per month | $0.03 per assessment per instance |

| **ECR container image scanning** | |
|---|---|
| Number of container images scanned initially on-push to Amazon ECR per month | $0.11 per image |
| Number of automated rescans for container images in Amazon ECR configured for continuous scanning per month | $0.01 per rescan |

| **On-demand Container image scanning (including within CI/CD solutions)** | |
|---|---|
| Number of container image scanned*** | $0.03 per image |

# Usage Monitoring



Inspector > Settings > Usage > By scan type

## Usage Info

Projected cost is based on account usage for the last 30 days.

By account | By scan type

### ▼ Amazon EC2 scanning

Amazon EC2 scanning for all accounts | Actual usage between 01/06/2025 - 02/05/2025

| Account | Agent-based instan... | Agent-based covera... | Agentless instances... | Age |
|---|---|---|---|---|
| 184128639995 | 0 | 0 | 0.67 | 485 |
| 634109035606 | 0 | 0 | 0 | 0 |
| 258219952275 | 1.03 | 744 | 0 | 0 |
| 891376936809 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 783388611754 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 891377392989 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 058264317214 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 505238968023 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 430118855485 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |
| 891377363861 Free trial: 14 days remaining | 0 | 0 | 0 | 0 |

### Total Monthly Projected Cost

Info

$5

All accounts

Projected cost is based on account usage for the last 30 days.

If the account has been active with Inspector for less than 30 days, we extrapolate the costs for 30 days based on the current usage.

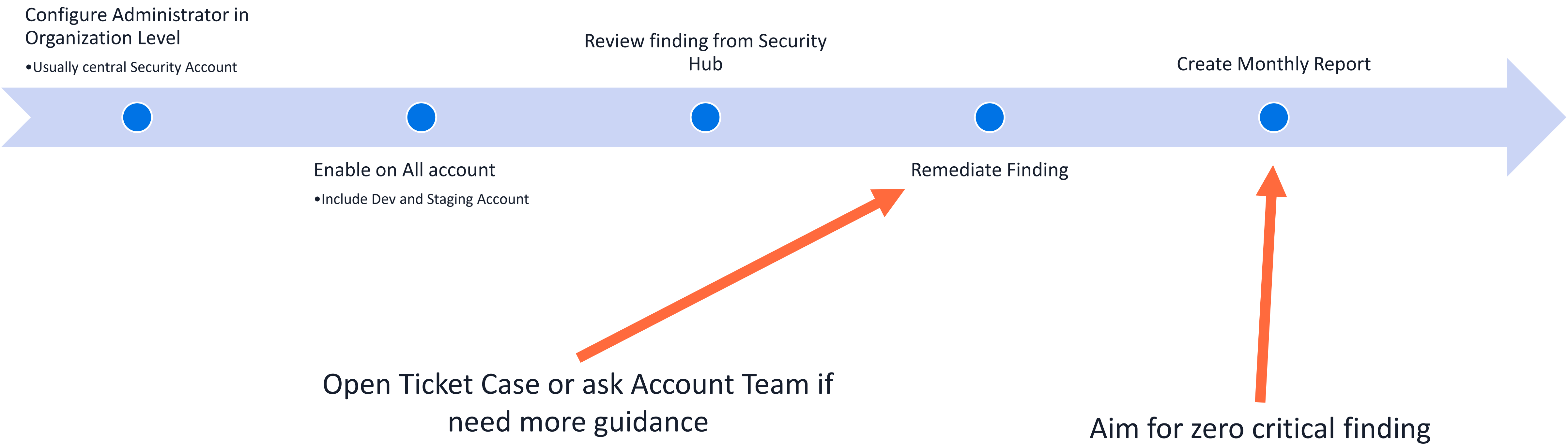
### Pricing Info

Per month, per Region

Learn more

About our 15-day trial

# Implementation Strategy



Configure Administrator in Organization Level

- Usually central Security Account

Enable on All account

- Include Dev and Staging Account

Review finding from Security Hub

Remediate Finding

Create Monthly Report

Open Ticket Case or ask Account Team if need more guidance

Aim for zero critical finding

# Demo

# Take Away

- Detect vulnerabilities across multiple layers of your infrastructure
- If your security team has limited manpower, prioritize scanning your runtime workloads
  - While shifting left is generally more cost-effective, you may not always have control over application development
- For organizations primarily running containerized workloads, explore serverless container
  - Focus on managing vulnerabilities in your container images
- Set realistic goals for your organization
  - A good starting point is to ensure comprehensive vulnerability scan coverage and use the identified findings to gain executive sponsorship for security initiatives

# Thank you