Microsoft

AZ-400.00 Learning Path 07: Implement security and validate code bases for compliance

Agenda

- Module 01: Introduction to Secure DevOps.
- Module 02: Implement open-source software.
- Module 03: Software Composition Analysis.
- Module 04: Static analyzers.
- Module 05: OWASP and Dynamic Analyzers.
- Module 06: Security Monitoring and Governance.
- Labs & Learning Path review and takeaways.

Learning Path overview

Learning objectives

After completing this Learning Path, students will be able to:

1. Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
2. Implement compliance and security in your application infrastructure
3. Describe the potential challenges with integrating open-source software
4. Inspect open-source software packages for security and license compliance
5. Manage organizational security and compliance policies
6. Integrate license and vulnerability scans into build and deployment pipelines
7. Configure build pipeline to access package security and license ratings

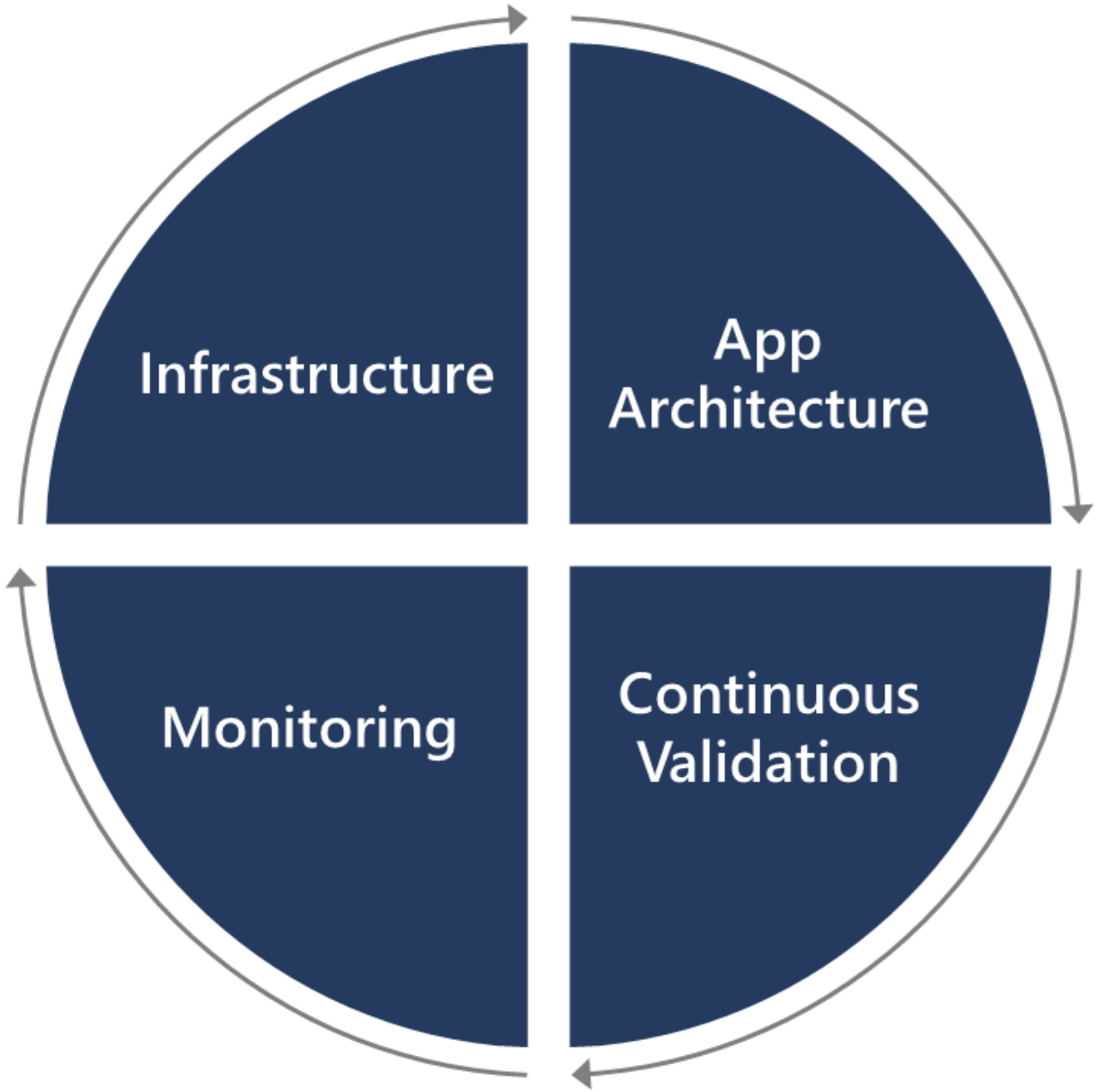
Module 01: Introduction to Secure DevOps

Introduction

Security is everyone's responsibility.

Securing applications is a continuous process that encompasses the following:

- Secure infrastructure
- Designing an architecture with layered security
- Continuous security validation
- Monitoring for attacks

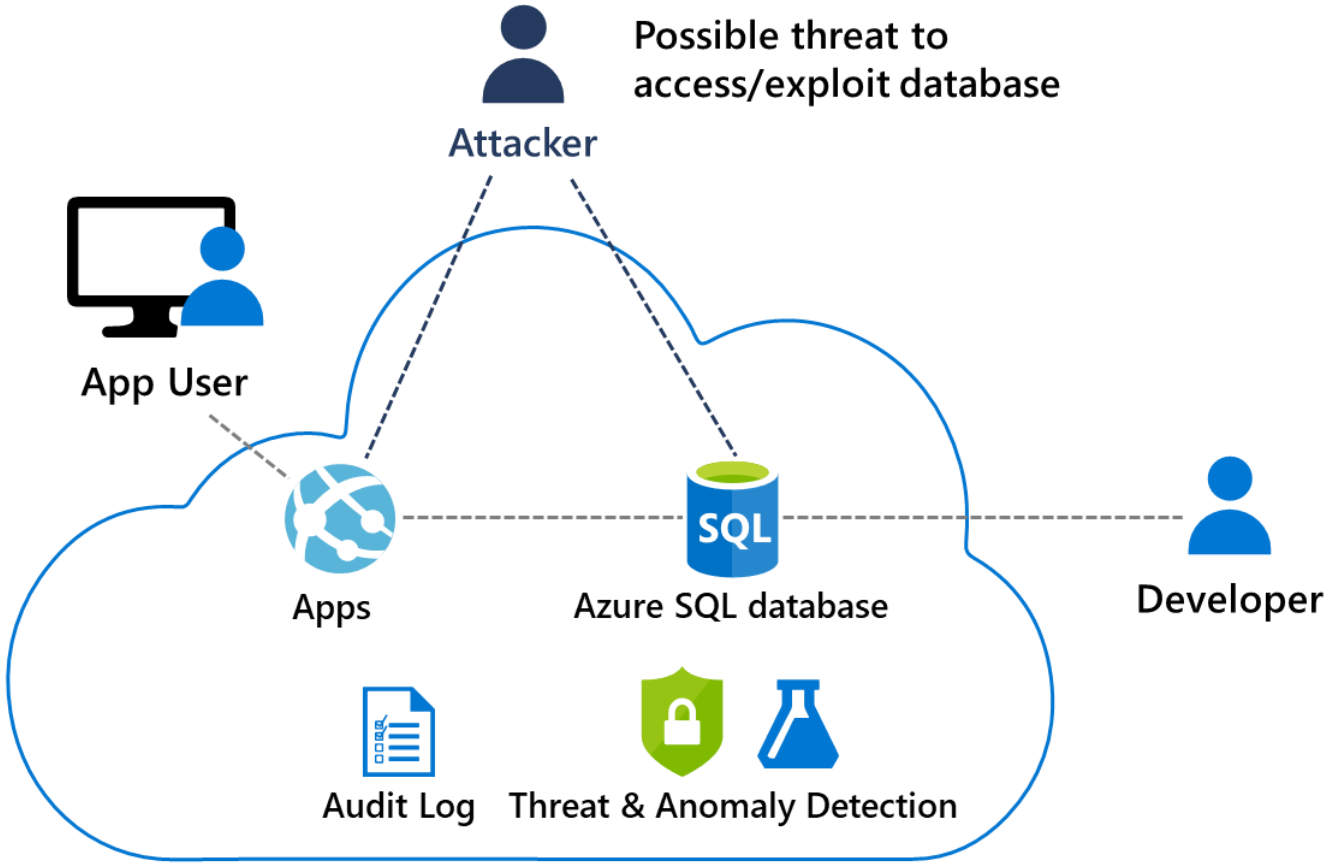


Describe SQL injection attack

A simple web app with SQL database

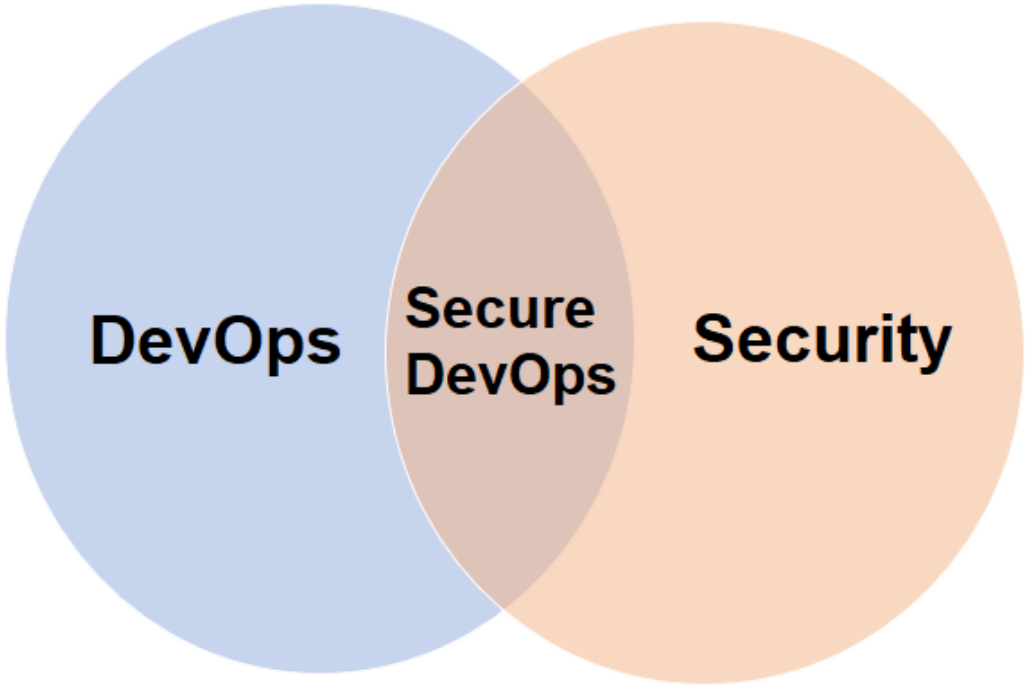
Heard this one before?

Hey John, a business user needs to see the customer data that's stored in the database... Why don't you expose the data on a web page in a table that the user can directly interact with?



Understand DevSecOps

- Secure DevOps is a set of practices designed to integrate DevOps and security
- The goal is to enable development teams to work fast without introducing unwanted vulnerabilities
- Security strategy includes access control, environment hardening, perimeter protection, and more

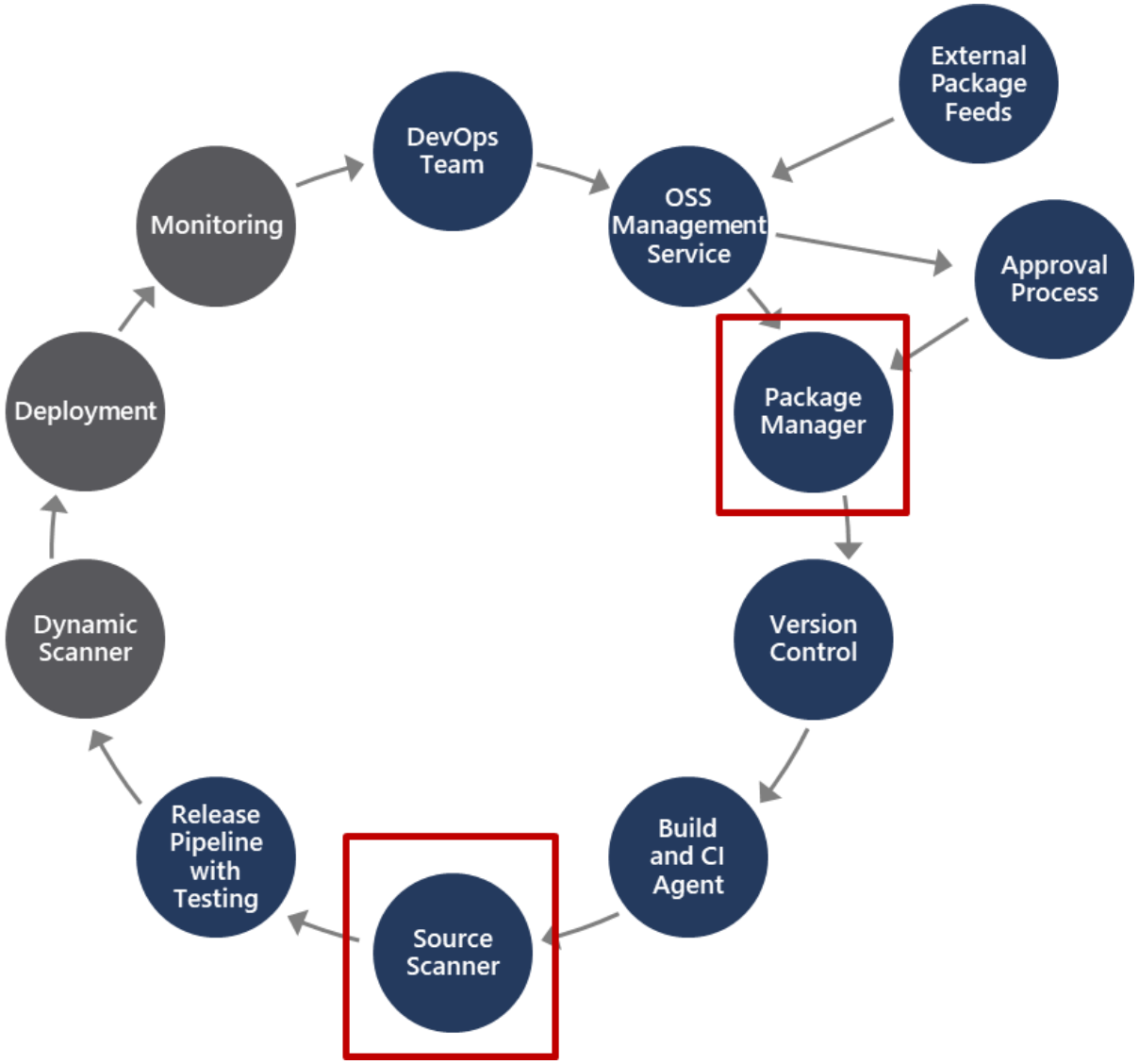


Secure DevOps is also sometimes referred to as *DevSecOps*

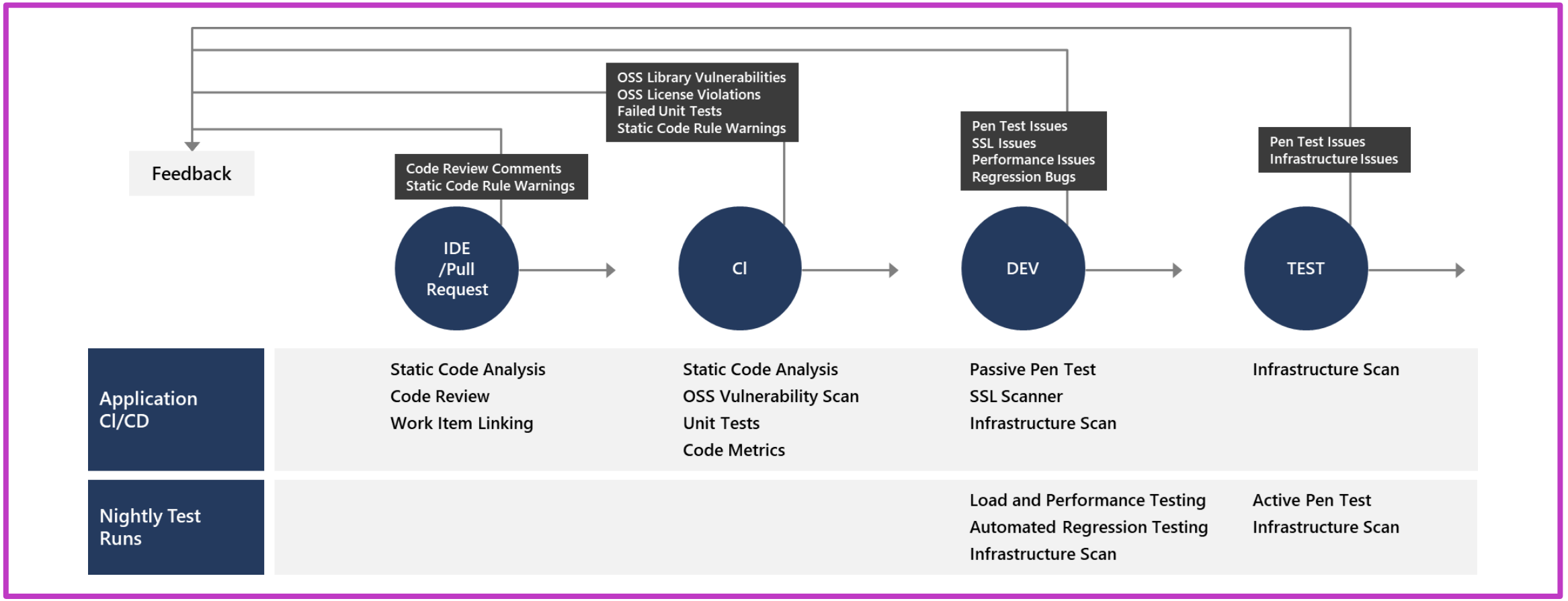
Explore Secure DevOps pipeline

Package management, and the approval process associated with it, accounts for how software packages are added to the pipeline, and the approval process they need to go through

Source scanner is for performing a security scan to verify certain security vulnerabilities are not present in our application source code

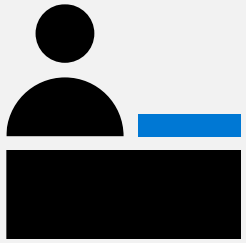
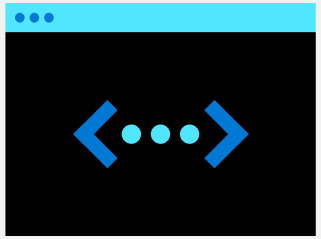


Explore key validation points



Continuous security validation should be added at each step from development through production

Explore continuous security validation

Coding	Build	QA & security	Production
			
$80/defect	$240/defect	$960/defect	$7,600/defect

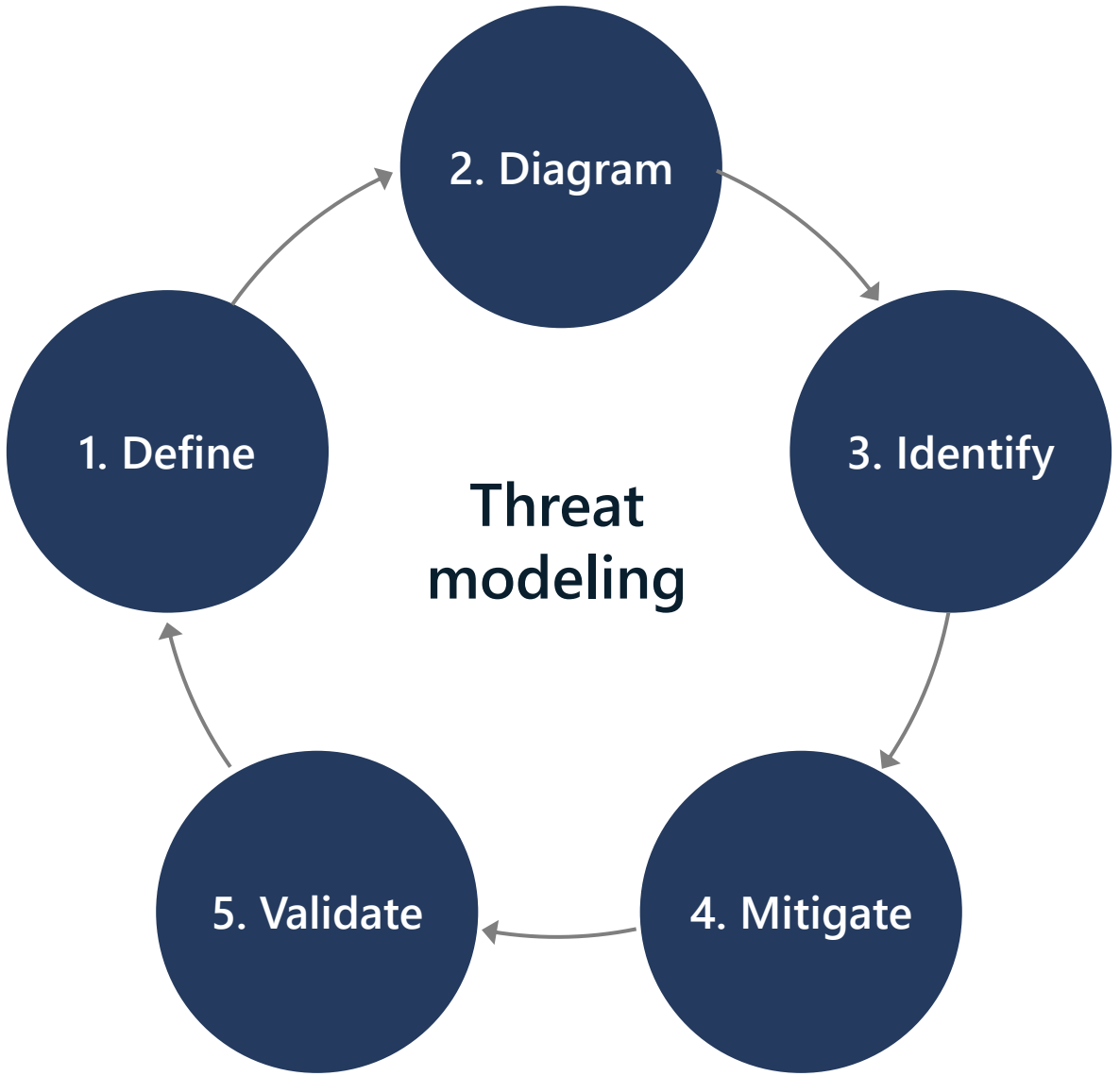
Source: Ponemon Institute Research

Reduce cost by moving Secure DevOps to the left

Use automated tooling and processes to identify problems

Understand threat modeling

- Define security requirements
- Create an application diagram
- Identify threats
- Mitigate threats
- Validate that threats have been mitigated



Threat modeling

DEMO

Module 02: Implement open-source software

Explore how software is built

1. **Software based 80% on components:**
   - Internal teams
   - Commercial 3rd party
   - Open-source community

2. **Almost all software nowadays uses open-source software in some way, shape, or form**

What is open-source software?

"Open-source software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose."

Enable

open-source use cases in the cloud

Integrate

open-source technologies in our platform

Release

innovative technologies as open source

Draft
R Server
.NET Core
CNTK
Brigade
TypeScript
F#
PowerShell

ACS Engine
Graph Engine
Visual Studio Code
Office UI Fabric
SONiC

Contribute

to the open-source ecosystem

ECOSYSTEM

https://en.wikipedia.org/wiki/Open-source_software

Explore corporate concerns with open-source software components

1. **It may have low quality:**
   This would impact maintainability, reliability, and performance of the overall solution

2. **It may have no active maintenance:**
   The code would not evolve over time or be alterable without making a copy of the source code, effectively forking away from the origin

3. **It could contain malicious code:**
   The entire system that includes and uses the code will be compromised. Potentially the entire company's IT and infrastructure is affected

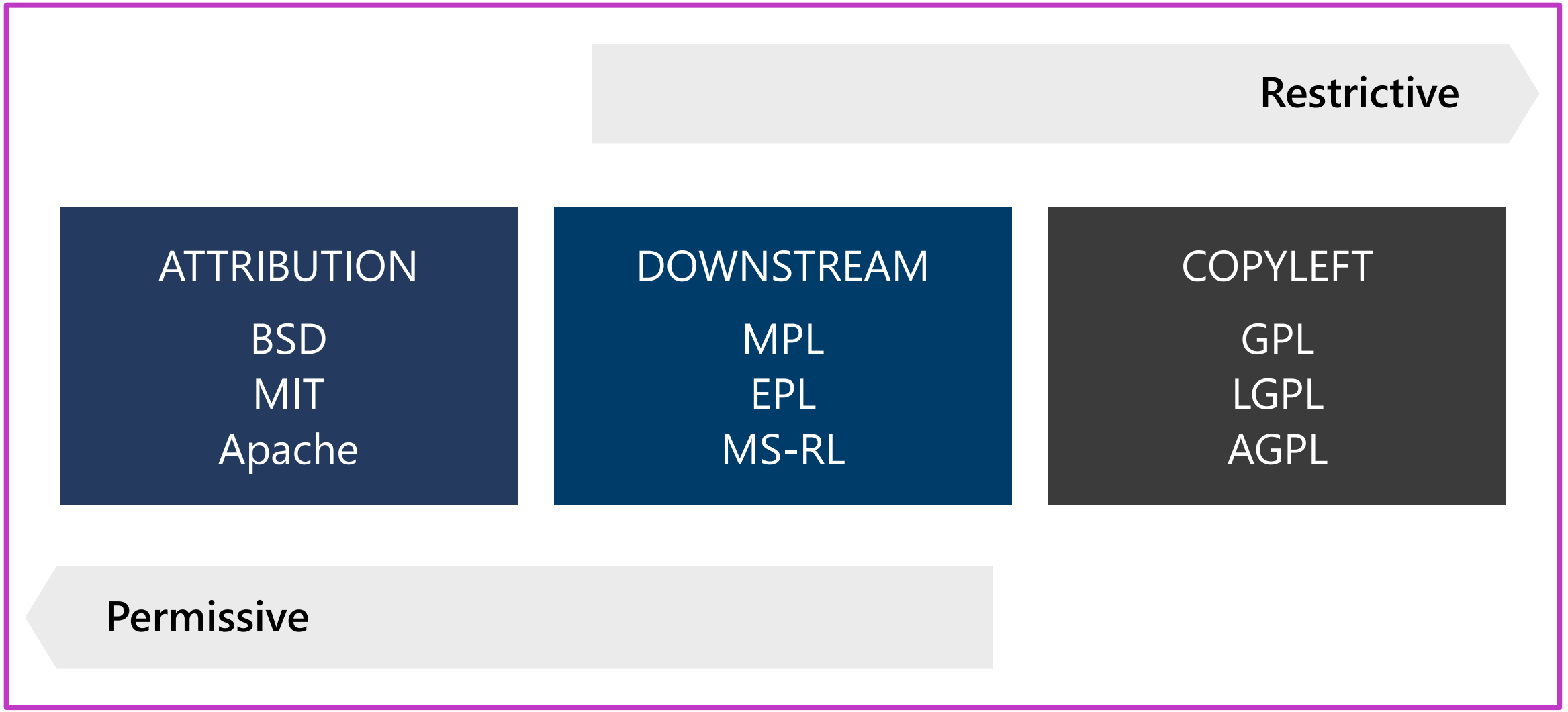
4. **It may have security vulnerabilities:**
   The security of a software system is as good as its weakest part. Using source code with vulnerabilities makes the entire system susceptible to attack by hackers and misuse

5. **It may have unfavorable licensing restrictions:**
   The effect of a license can affect the entire solution that uses the open-source software

The Open-Source Definition

1. Free Redistribution
2. Provide Full Access to Source Code
3. Allow for modifications and derived works of the original Source Code
4. Allow the author to protect the integrity of the source code through patches
5. No discrimination against persons or groups for any reason
6. No discrimination against fields of endeavor, such as business or research
7. License must not be specific to a product, and must not restrict other software

Explore common open-source licenses



Examine license implications and ratings

Using a package implies following license requirements:

1. **The license type may have a high, medium or low impact on distributed software including it**

2. **License rating indicates impact of use of packages:**
   - Compliancy
   - Intellectual property
   - Exclusive rights

Module 03: Software Composition Analysis

Inspect and validate code bases for compliance

There are many aspects to building and deploying secure applications

1. General knowledge problem

2. Code is created correctly and securely implements the required features, and we need to make sure that the features were designed with **security in mind in the first place**

3. Complies with the rules and regulations that it is required to meet

Explore software composition analysis (SCA)

Package Management:

- Unique source of binary components
- Create a local cache of approved components
- Use Azure Artifacts to share and organize your packages
- Package types: NuGet, npm, Maven, Gradle, Universal, and Python

Open-Source Software (OSS) Components:

- Reused dependencies can have security vulnerabilities
- Ensure you have the latest version
- Check for the correct binaries
- Promptly address vulnerabilities
- Analyzing the software to determine its composition can help mitigate potential vulnerabilities
- Scanning tools discussed in the next Learning Path

Integrate Mend with Azure Pipelines

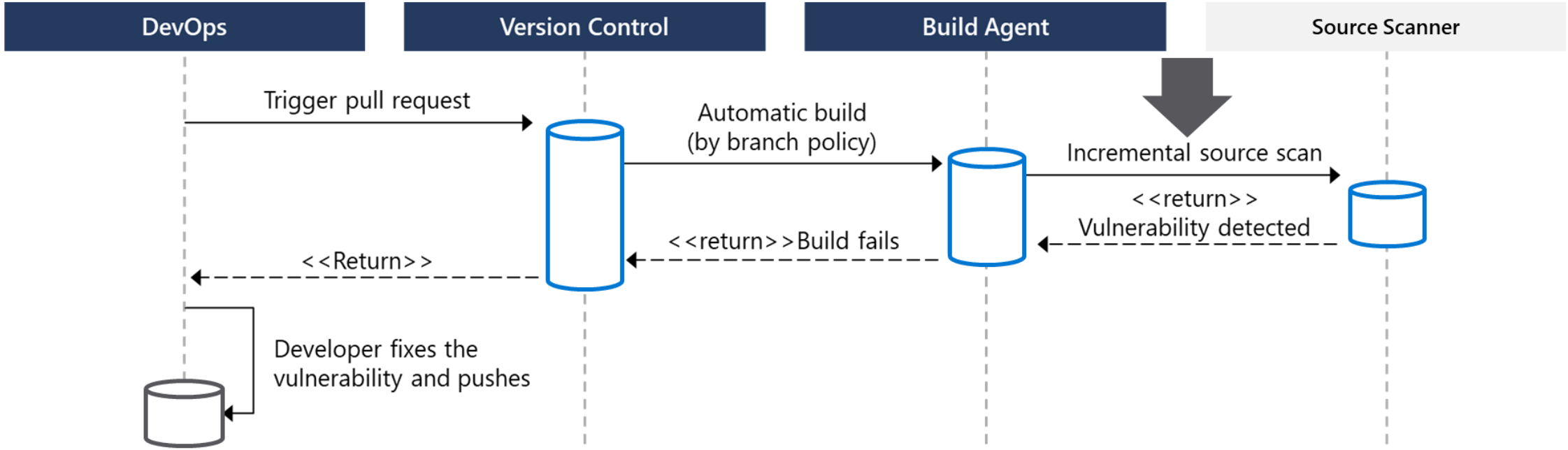
1. *Mend* is one such example of an extension available on the Azure DevOps Marketplace

2. **If consuming external packages, the *Mend* extension specifically addresses the questions of open-source security, quality, and license compliance.**
   - Continuously detect all open-source components in your software
   - Receive alerts on open-source security vulnerabilities
   - Automatically enforce open-source security and license compliance

Implement GitHub Dependabot alerts and security updates

1. Check for updates
2. Alerts
3. Automated pull requests
4. Review and triage

Integrate software composition analysis checks into pipelines



Pull requests are the way DevOps teams submit changes

Mend, Checkmarx, Veracode, and Black Duck by Synopsis can facilitate incremental scans

Integrate scanning into a team's workflow at multiple points along the path

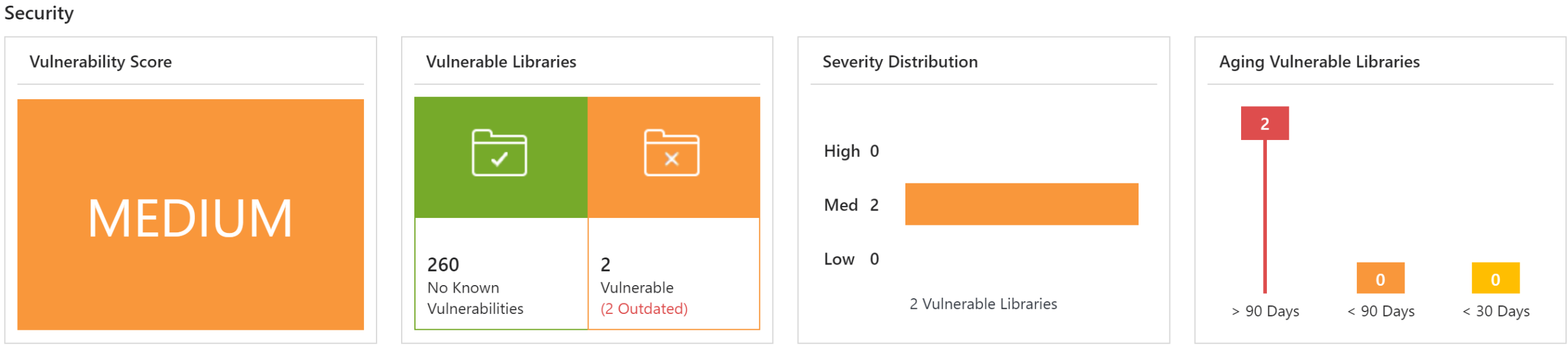
Examine tools for assess package security and license rate

Approach 1:
Scan centralized artifact repository

Approach 2:
Tooling during build in pipeline

Tool	Type
Artifactory	Artifact repository
SonarQube	Static code analysis tool
Mend Bolt	Build scanning

Interpret alerts from scanner tools



Report Contains:

Security vulnerabilities

License risks and compliancy

Outdated libraries

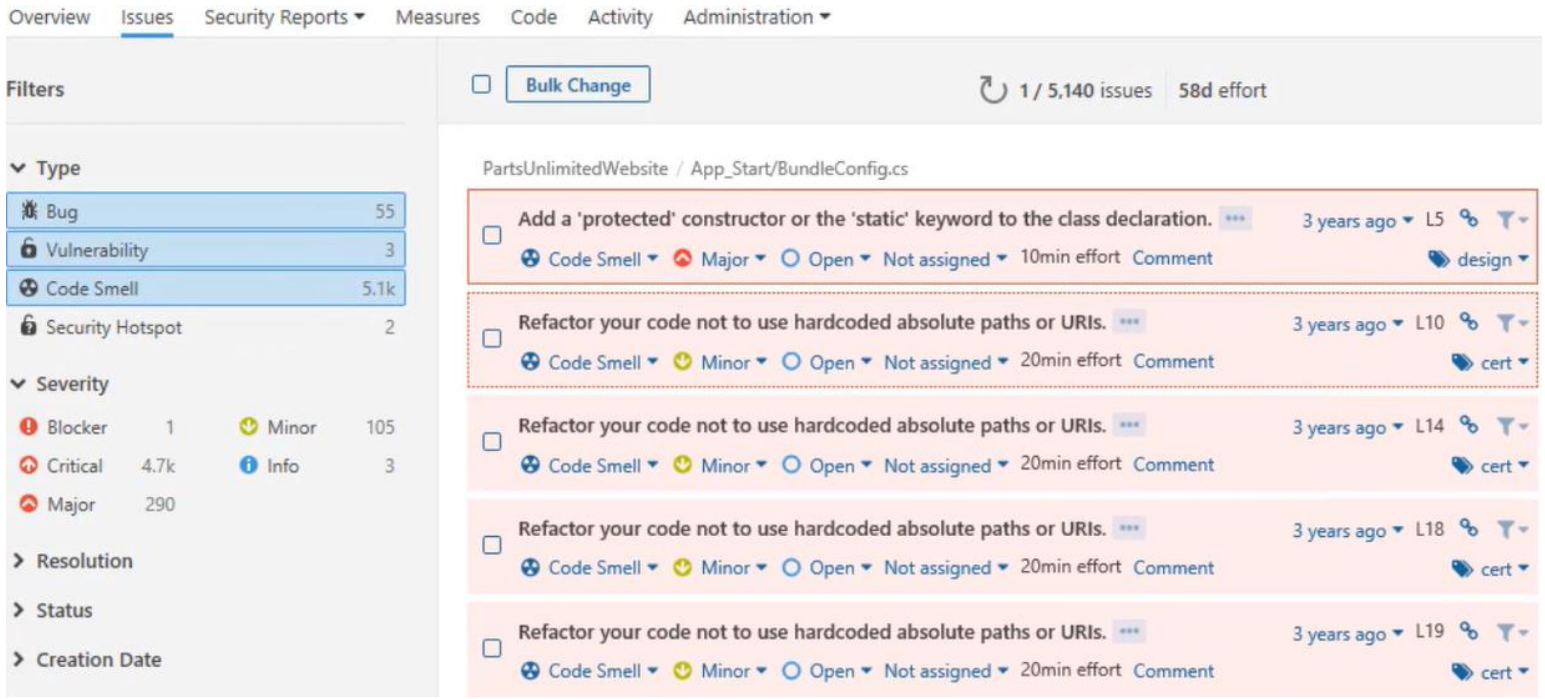
Aspects to keep in mind:

False positives

Security bug bar

Module 04: Static analyzers

Explore SonarCloud



- Open platform for managing code quality
- Joint investments with Microsoft
- Wide range of programming languages supported
- Cloud-hosted version: SonarCloud

Explore CodeQL in GitHub

Action:

- Treats code like queryable data
- Standard or custom analysis queries

Steps:

- Create a CodeQL database (based upon the code)
- Run CodeQL queries against the database
- Interpret the results

CodeQL CLI or CodeQL for Visual Studio Code

Module 05: OWASP and Dynamic Analyzers

Plan to Implement OWASP Secure Coding Practices

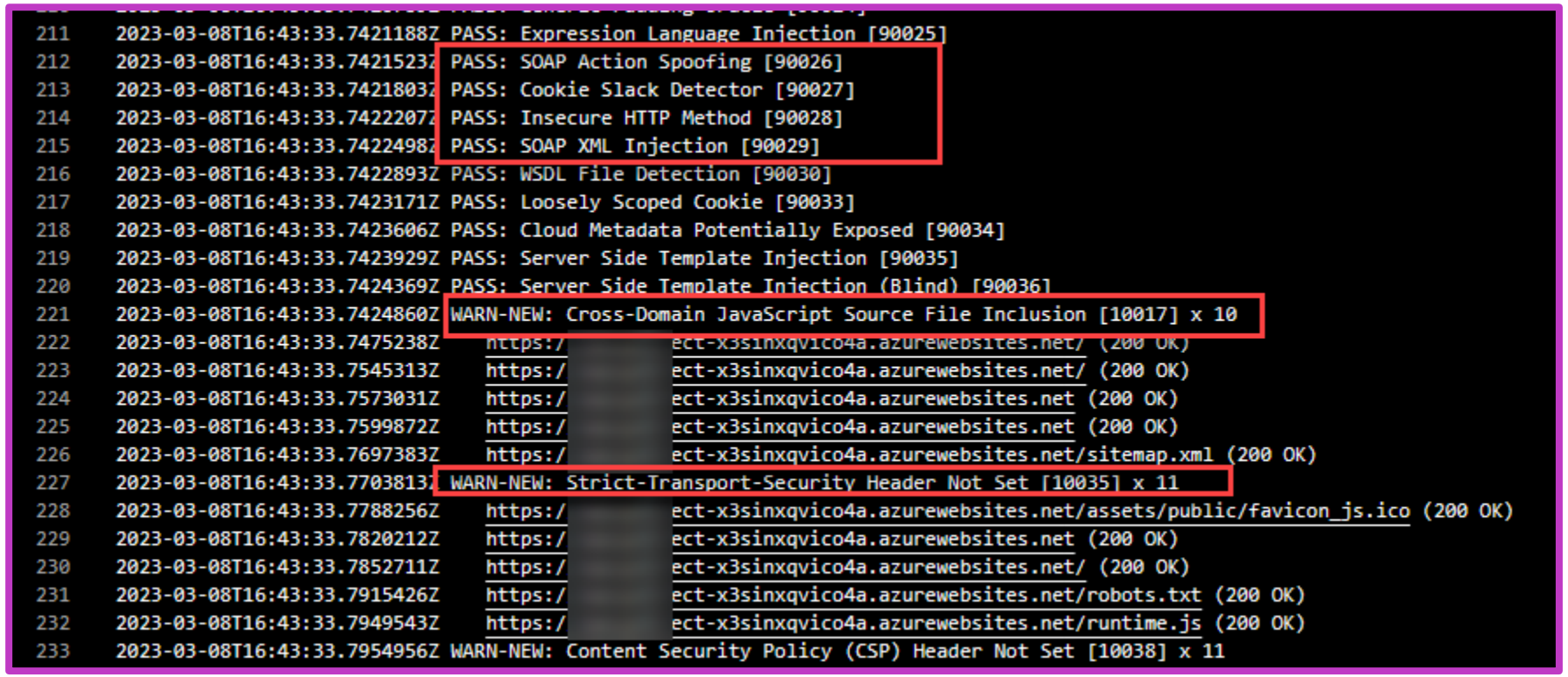
1. Start with secure coding practices
2. Open Web Application Security Project (OWASP) – A global charitable organization focused on improving the security of software
3. OWASP regularly publish a set of *Secure Coding Practices*

Explore ZAP (Zed Attack Proxy) penetration test

- ZAP can be used for penetration testing
- Testing can be active or passive
- Conduct a quick baseline scan to identify vulnerabilities
- Conduct nightly more intensive scans

Explore ZAP (Zed Attack Proxy) results and bugs

- ZAP provides a report with results and bugs
- Use a holistic and layered approach to security



Discussion: Security policy tooling

Azure DevOps can be integrated with a wide range of existing tooling that is used for checking security policy during builds

Which security policy tools do you currently use?	What do you like or don't like about the tools?

Module 06: Security Monitoring and Governance

Implement pipeline security

Use traditional operational methods for protecting identities and assets

- **Authentication and Authorization:** Use Multi-Factor Authentication and Just enough Admin (JEA)
- **Use the CI/CD release pipeline**: Use immutable infrastructure and only using your pipeline
- **Manage Permissions**: Secure the pipeline using RBAC
- **Dynamic Scanning**: Test running apps, such as penetration testing

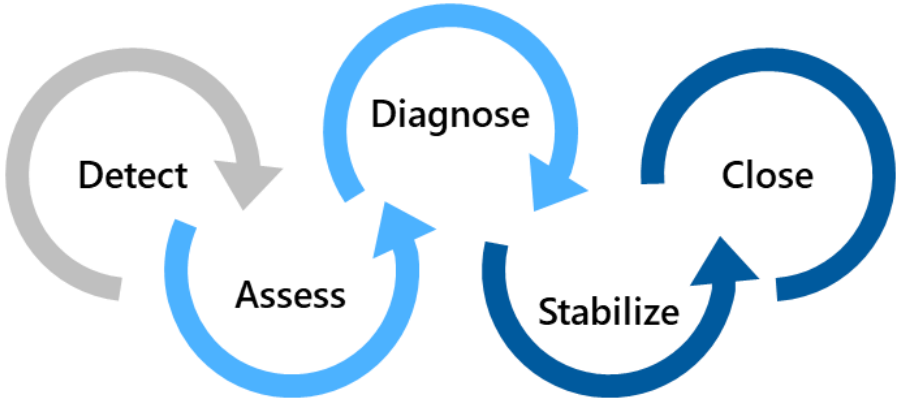
Implement pipeline security

- Provide security recommendations
- Monitor all your services
- Analyze and identify potential attacks
- Supports Windows and Linux operating systems
- Monitor security settings
- Use Azure Machine Learning
- Provide just-in-time (JIT) access control
- Available in two pricing tiers

Examine Microsoft Defender for Cloud usage scenarios

Scenario 1

Use Microsoft Defender for Cloud for an incident response



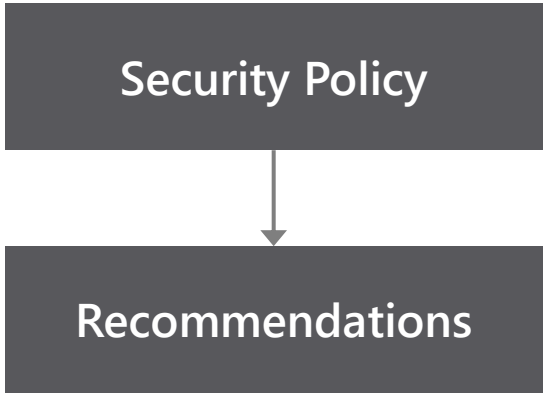
Detect – Verify a high security alert was raised

Access – Obtain information about the alert

Diagnose – Follow the remediation steps

Scenario 2

Use Microsoft Defender for Cloud recommendations to enhance security

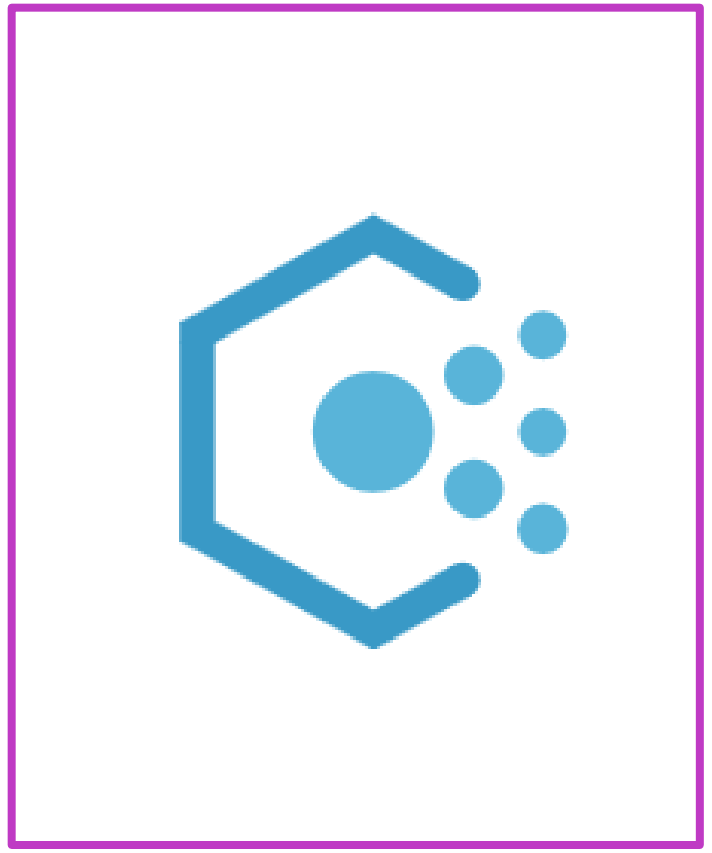


Configure a security policy

Implement the recommendations for the security policy

Explore Azure Policy

- Is a service in Azure that you use to create, assign, and manage policies
- Provides enforcement by using policies and initiatives
- Runs evaluations on your resources and scans for those not compliant
- Comes with several built-in policy and initiative definitions
- Integrates with Azure DevOps by applying any continuous integration (CI) and continuous deployment (CD) pipeline policies



✓ **An example of an Azure policy that you can integrate with your DevOps pipeline is the Check Gate task**

Understand policies

A *policy definition* expresses what to evaluate and what action to take

Policies are defined in JSON

Assign policies using Azure Portal, Azure CLI, or Azure PowerShell

Use remediation for non-compliant resources

The screenshot is a section of a policy file defining allowed locations

```
        "displayName": "Allowed locations",
        "description": "This policy enables you to
restrict the locations your organization can specify
when deploying resources.",
        "policyRule": {
            "if": {
                "not": {
                    "field": "location",
                    "in": "
[parameters('allowedLocations')]"
                }
            },
            "then": {
                "effect": "deny"
            }
        }
    }
}
```

Explore initiatives

An *initiative definition* is a set of policy definitions

Helps track your compliance state for a larger goal

Assigned to a specific scope

Reduces the need for individual scope assignments

Initiative Example:

Create an Initiative named *"Enable Monitoring In Microsoft Defender for Cloud."* This would provide the following policies:

- Monitor unencrypted SQL Database policy definition
- Monitor OS vulnerabilities policy definition
- Monitor missing Endpoint Protection policy definition

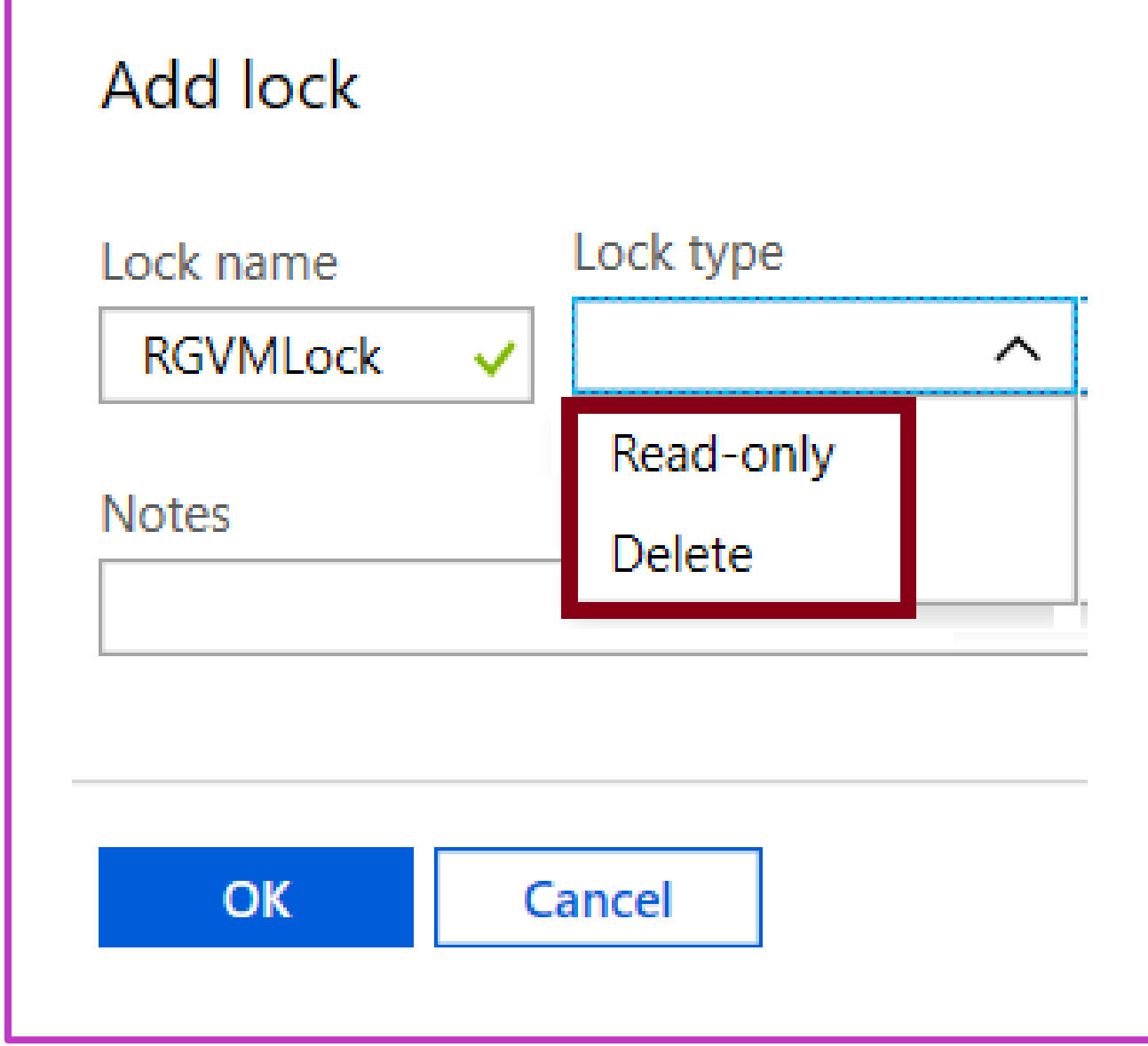
✓ **Even if you have a single policy, we recommend using initiatives if you anticipate increasing the number of policies over time**

Explore resource locks

Prevent accidental deletion or modification of your Azure resources:

CanNotDelete – Means authorized users can still read and modify a resource, but they can't *delete* the resource

ReadOnly – Means authorized users can read a resource, but they can't delete or update it. Applying this lock is like restricting all authorized users to the permissions granted by the Reader role



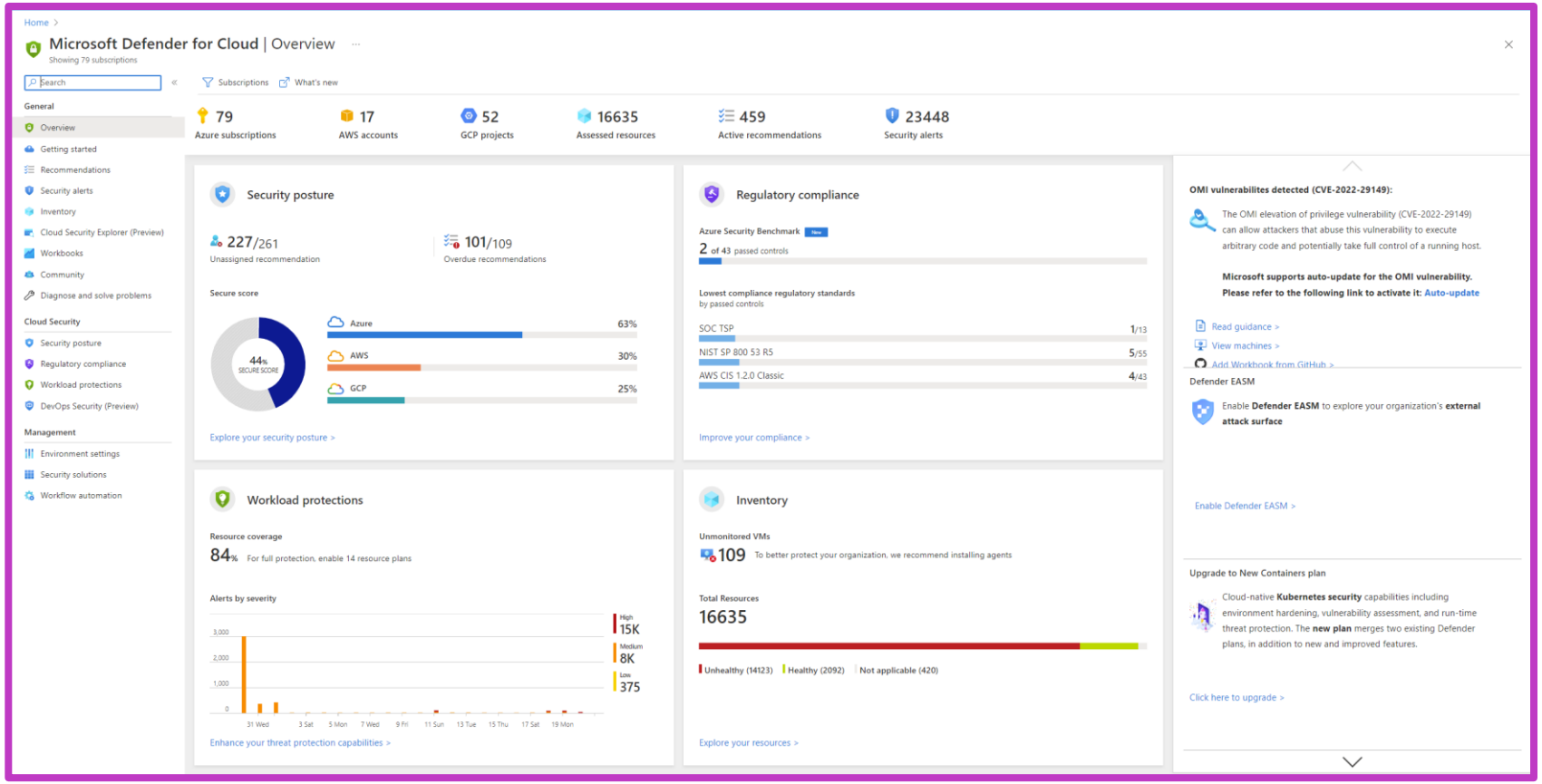
In the Azure portal, the locks are called ***Delete*** and ***Read-only*** respectively

Understand Microsoft Defender for Identity

Microsoft Defender for Cloud:

- **Portal** monitors and responds to suspicious activity
- **Sensor** monitors domain controller traffic
- **Service** connects to Microsoft Intelligent Security Graph

Identifies, detects, and helps you investigate advanced threats, compromised identities, and malicious insider actions



Labs

Lab: Implement security and compliance in Azure Pipelines

Lab overview:

In this lab, we will create a new Azure DevOps project, populate the project repository with a sample application code, create a build pipeline. Next, we will install Mend Bolt from the Azure DevOps Marketplace to make it available as a build task, activate it, add it to the build pipeline, use it to scan the project code for security vulnerabilities and licensing compliance issues, and finally view the resulting report.

Objectives:

- Create a Build pipeline
- Install Mend Bolt from the Azure DevOps marketplace and activate it
- Add Mend Bolt as a build task in a build pipeline
- Run build pipeline and view Mend security and compliance report

Duration:

Lab: Managing technical debt with SonarQube and Azure DevOps

Lab overview:

In this lab, you will learn how to setup SonarQube on Azure and integrate it with Azure DevOps.

Objectives:

- Provision SonarQube server as an Azure Container Instance from the SonarQube Docker image
- Set up a SonarQube project
- Provision an Azure DevOps Project and configure CI pipeline to integrate with SonarQube
- Analyze SonarQube reports

Duration:

Learning Path review and takeaways

What did you learn?

1. Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
2. Implement compliance and security in your application infrastructure
3. Describe the potential challenges with integrating open-source software
4. Inspect open-source software packages for security and license compliance
5. Manage organizational security and compliance policies
6. Integrate license and vulnerability scans into build and deployment pipelines
7. Configure build pipeline to access package security and license ratings

Learning Path review questions

1. Secure DevOps combines which two elements?
2. Which term broadly defines what security means in Secure DevOps?
3. What component in Azure DevOps can you use to store, organize and share access to packages, and integrate those packages them with your continuous integration and continuous delivery pipeline?
4. What describes the term software composition analysis?
5. From where can extensions be sourced from, to be integrated into Azure DevOps CI/CD pipelines and help provide security composition analysis?

Learning Path review questions (continued)

6. Which products are available as extensions in Azure DevOps Marketplace, and can provide either OSS or source code scanning as part of an Azure Pipeline?

7. Which Azure service is a monitoring service that can provide threat protection and security recommendations across all your services both in Azure and on-premises?

8. Which Azure service should you use from the below list to monitor all unencrypted SQL databases in your organization?

9. Which facility allows you to prevent accidental deletion of resources in Azure?

Learning Path review questions (continued)

10 What issues are often associated with the use of open-source libraries?

11 How can an open-source library cause licensing issues if it is free to download?

12 What is open-source software?