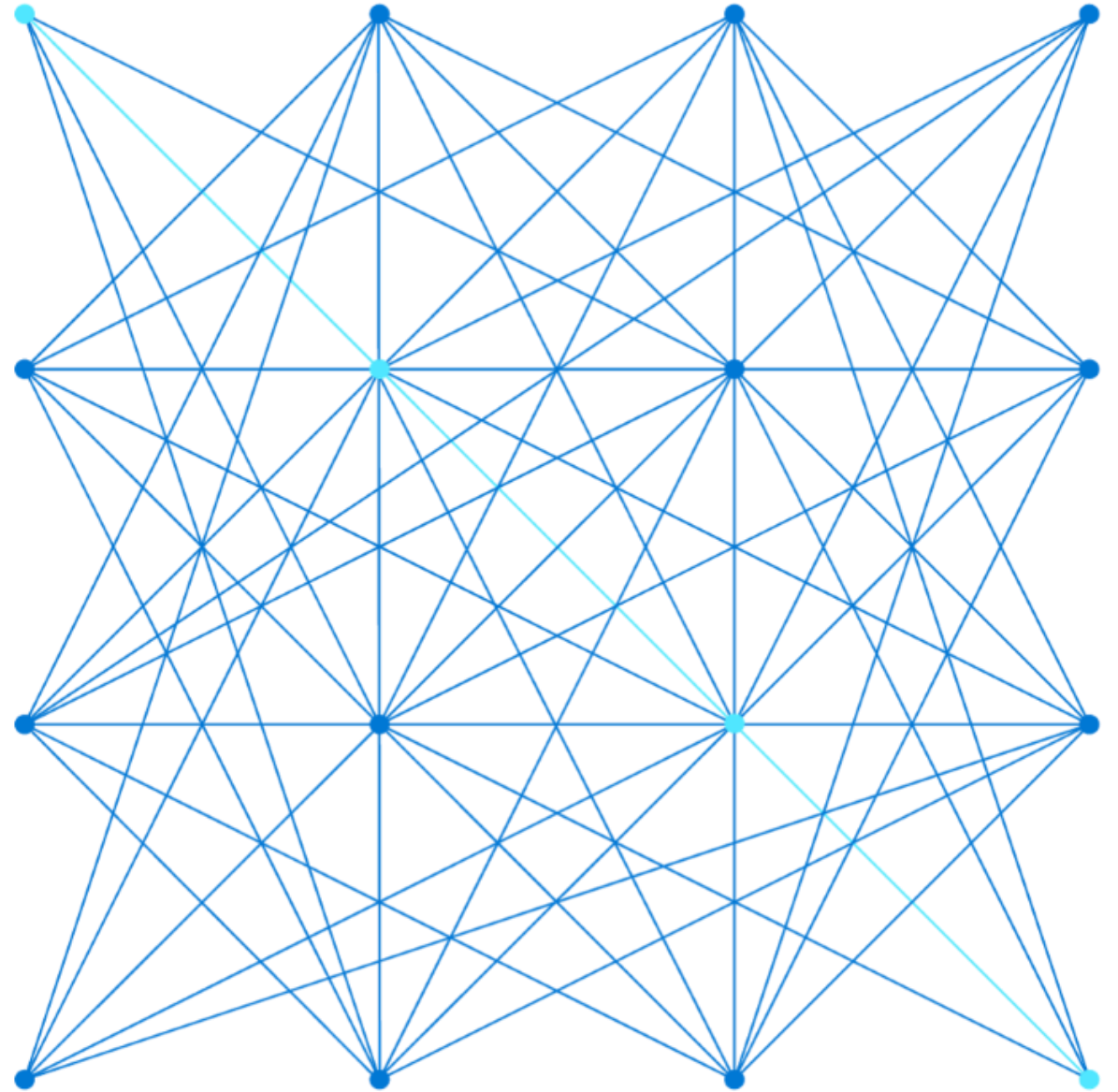


AZ-400.00

Module 20: Validating Code Bases for Compliance



Lesson 01: Module overview



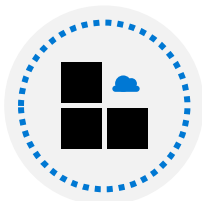
Module overview



Lesson 1: Module overview



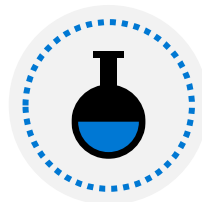
Lesson 2: Open-source software



Lesson 3: Managing security and compliance policies



Lesson 4: Integrating license and vulnerability scans



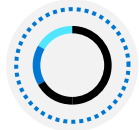
Lesson 5: Lab



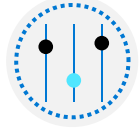
Lesson 6: Module review and takeaways

Learning objectives

After completing this module, students will be able to:



Describe the potential challenges with integrating open-source software



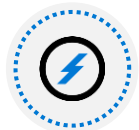
Inspect open-source software packages for security and license compliance



Manage organizational security and compliance policies



Integrate license and vulnerability scans into build and deployment pipelines

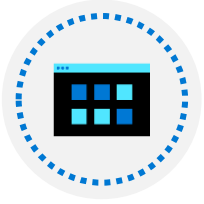


Configure build pipeline to access package security and license ratings

Lesson 02: Open-source software



How software is built

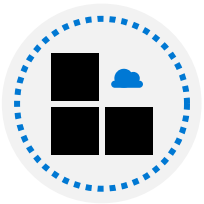


Software based 80% on components:

Internal teams

Commercial 3rd party

Open-source community



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

Corporate concerns with open-source software components



Are of low quality:

This would impact maintainability, reliability, and performance of the overall solution



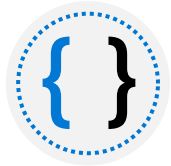
Have no active maintenance:

The code would not evolve over time or be alterable without making a copy of the source code, effectively forking away from the origin



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



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

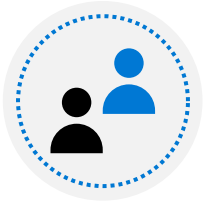


Have unfavorable licensing restrictions:

The effect of a license can affect the entire solution that uses the open-source software

Open-source licenses

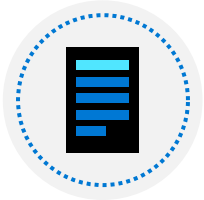
According to the open-source definition of OpenSource.org a license **should not**:



Discriminate against persons or groups



Discriminate against fields of endeavor

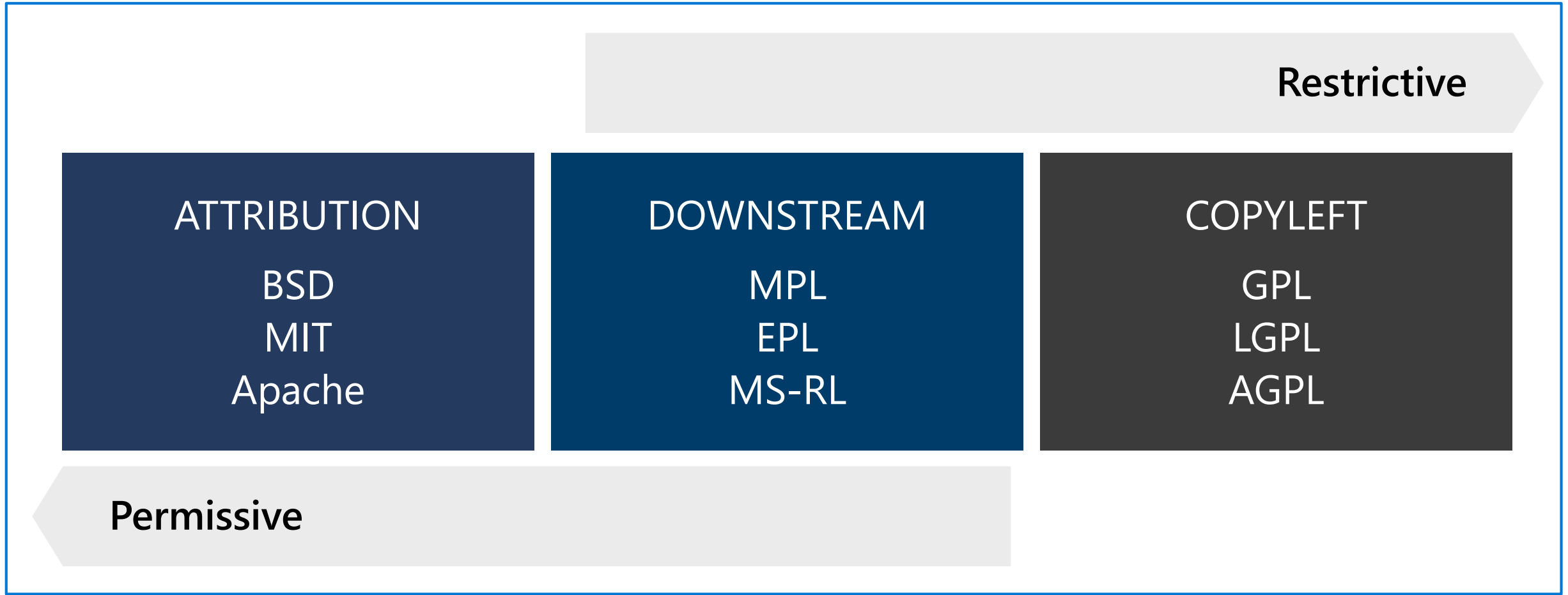


Be specific to a product



Restrict other software

Common open-source licenses



License implications and ratings

Using a package implies following license requirements:

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

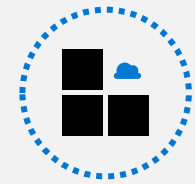
License rating indicates impact of use of packages:

Compliance

Intellectual property

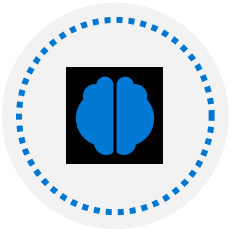
Exclusive rights

Lesson 03: Managing security and compliance policies

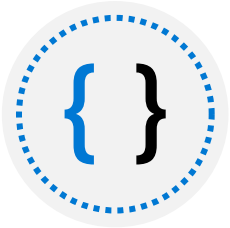


Inspecting and validating code bases for compliance

There are many aspects to building and deploying secure applications



General knowledge problem

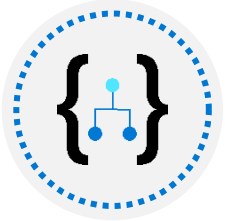


Code is created correctly and securely implements the required features, and we need to make sure that the features were designed with **security in mind in the first place**



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

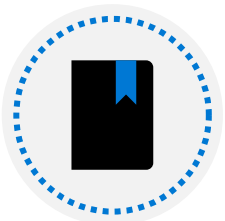
Planning to implement OWASP Secure Coding Practices



Start with secure coding practices



[Open Web Application Security Project \(OWASP\)](https://owasp.org/) – A global charitable organization focused on improving the security of software

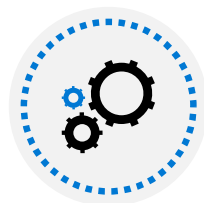


OWASP regularly publish a set of *Secure Coding Practices*

Inspecting and validating code bases for compliance



BinSkim is a static analysis tool that scans binary files



OWASP Zed Attack Proxy Scan is an open-source web app scanner for professional penetration testers

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?

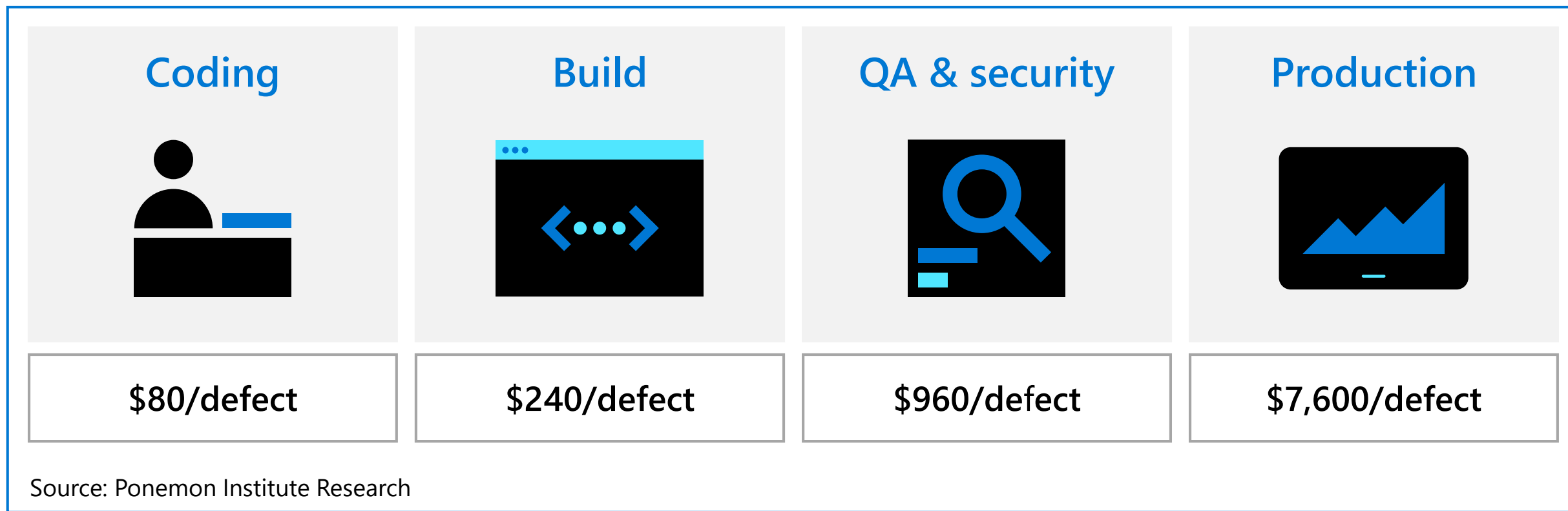


What do you
like or don't like
about the tools?

Lesson 04: Integrating license and vulnerability scans



Implement continuous security validation



Reduce cost by moving DevSecOps to the left

Use automated tooling and processes to identify problems

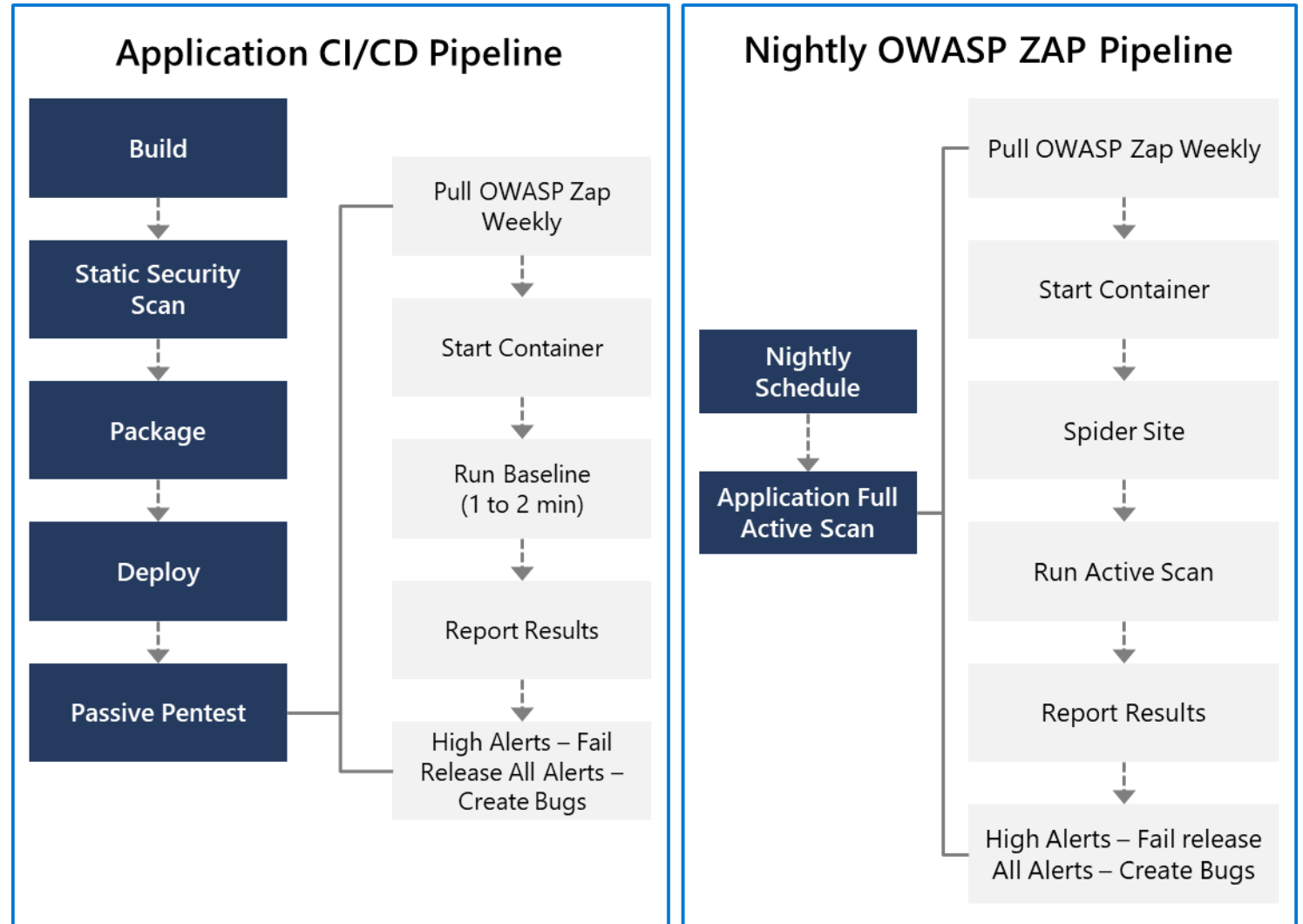
OWASP ZAP penetration testing

OWASP 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



OWASP ZAP results and bugs

OWASP ZAP provides a report with results and bugs

Use a holistic and layered approach to security

The screenshot displays the OWASP ZAP security scan results interface. At the top, there are tabs for 'Backlog', 'Board', and 'Capacity'. Below these, a summary section shows 'OWASP ZAP Nightly / Release-40' with a 'Summary' tab selected. The summary includes a 'Total tests' section with a donut chart showing 6 tests: 0 Passed (0%), 6 Failed (100%), and 0 Others (0%). The 'Pass percentage' is 0% and the 'Run duration' is 0s. Below this, there are buttons for 'Expand all', 'Collapse all', and 'Create bug'. The main section is titled 'Test' and shows a list of failed tests under the heading '0/6 Passed - OWASP ZAP Security Tests'. The failed tests are:

- ✖ Incomplete or No Cache-control and Pragma HTTP Header Set
- ✖ Cookie No HttpOnly Flag
- ✖ Cookie Without Secure Flag
- ✖ Web Browser XSS Protection Not Enabled
- ✖ X-Content-Type-Options Header Missing
- ✖ X-Frame-Options Header Not Set

On the right side, there is a 'Board' view showing a grid of bugs. The bugs are categorized by 'New' (61.5 h) and 'Active' (11 h). The bugs listed are:

- 207810 Incomplete or No Cache-control and Pragma HTTP Header Set (State: New)
- 207811 Cookie No HttpOnly Flag (State: New)
- 207813 Web Browser XSS Protection Not Enabled (State: New)
- 207812 Cookie Without Secure Flag (State: New)
- 207815 X-Frame-Options Header Not Set (State: New)
- 207814 X-Content-Type-Options Header Missing (State: New)

Tools for assessing package security and license rating

Approach 1:
Scan centralized artifact repository

Approach 2:
Tooling during build in pipeline

Tool	Type
Artifactory	Artifact repository
SonarQube	Static code analysis tool
White Source(Bolt)	Build scanning

SonarCloud

SonarQube:

- Open platform for managing code quality
- Joint investments with Microsoft
- Wide range of programming languages supported

Cloud-hosted version: SonarCloud

Interpret alerts from scanning tools

Report Contains:

Security vulnerabilities

License risks and compliancy

Outdated libraries

Aspects to keep in mind:

False positives

Security bug bar

Security

Vulnerability Score

MEDIUM

Vulnerable Libraries



260
No Known
Vulnerabilities



2
Vulnerable
(2 Outdated)

Severity Distribution

High 0

Med 2

Low 0



2 Vulnerable Libraries

Aging Vulnerable Libraries

2

> 90 Days

0

< 90 Days

0

< 30 Days

CodeQL in GitHub

Action:

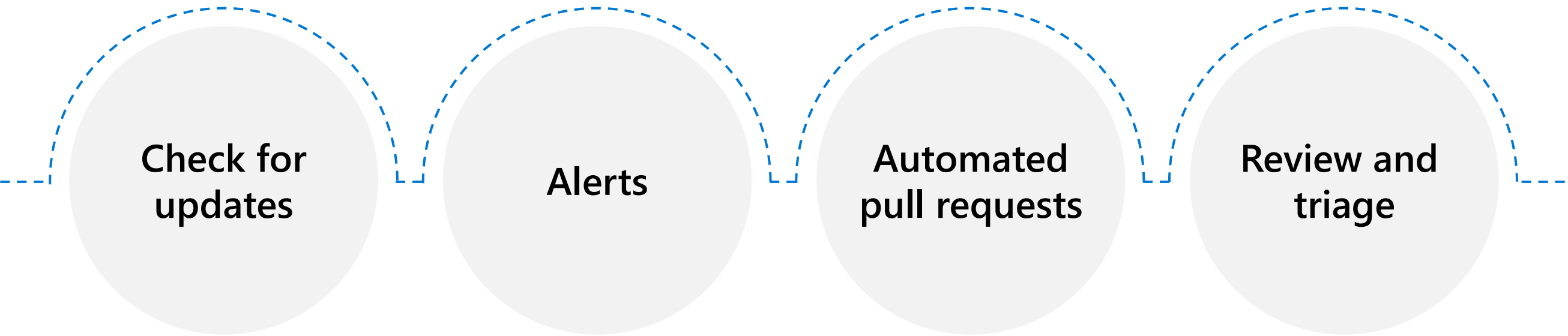
Treats code like queryable data
Standard or custom analysis queries

Steps:

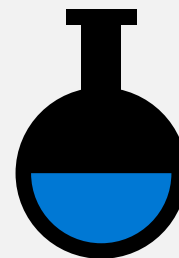
Create a CodeQL database (based upon the code)
Run CodeQL queries against the database
Interpret the results

CodeQL CLI or CodeQL for Visual Studio Code

GitHub Dependabot alerts and security updates



Lesson 05: 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:



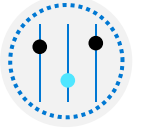
Lesson 06: Module review and takeaways



What did you learn ?



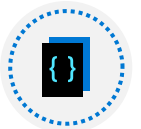
Describe the potential challenges with integrating open-source software



Inspect open-source software packages for security and license compliance to align with corporate standards



Manage organizational security and compliance policies



Integrate license and vulnerability scans into build and deployment pipelines



Configure build pipeline to access package security and license ratings

Module review questions:

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

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

3 What is open-source software?