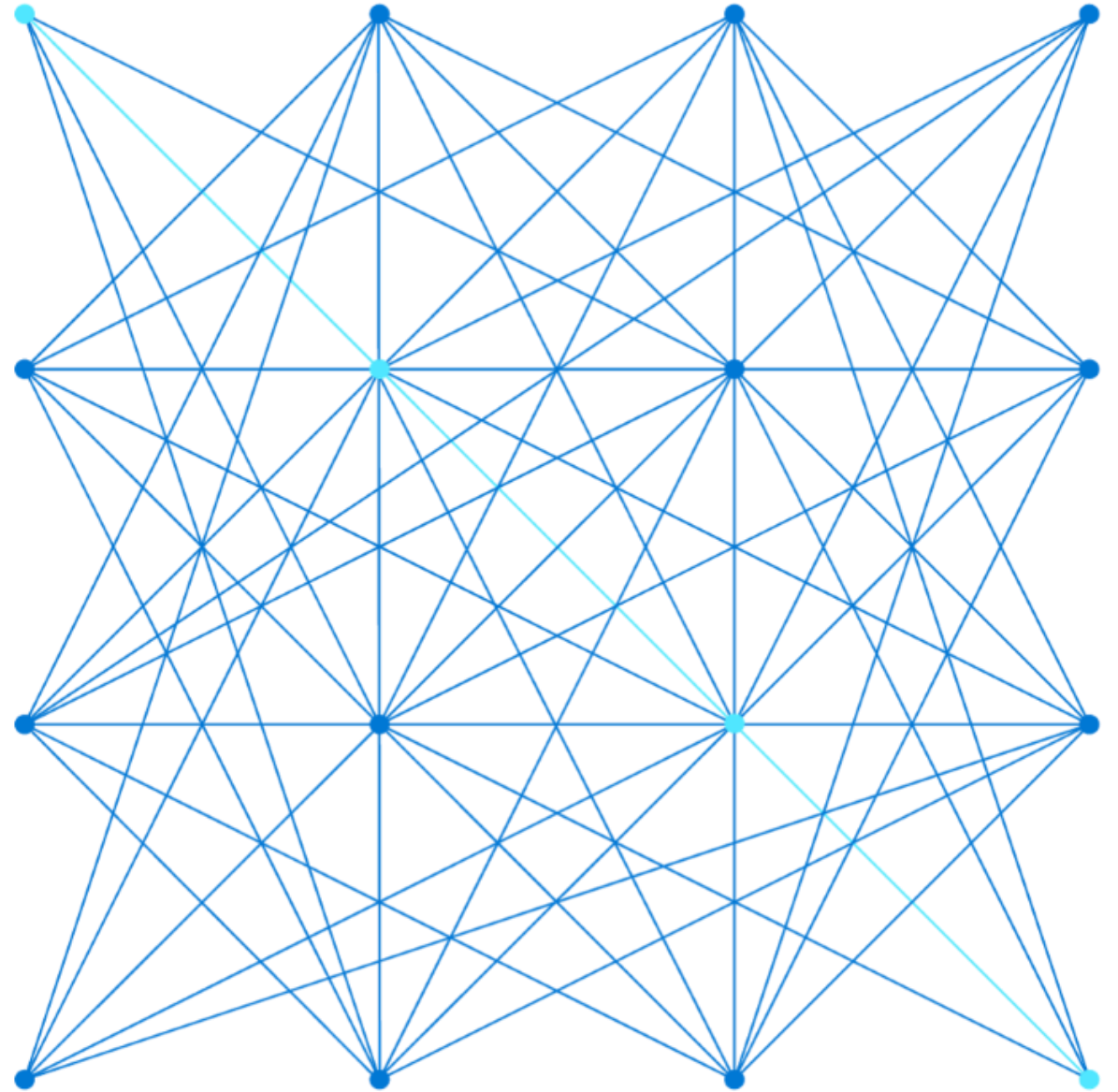
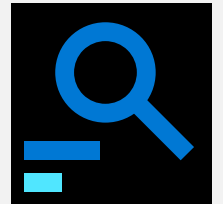


AZ-400.00

Module 19: Implementing Security in DevOps Projects



Lesson 01: Module overview



Module overview



Lesson1: Module overview



Lesson 2: Security in the pipeline



Lesson 3: Azure Security Center



Lesson 4: Lab



Lesson 5: Module review and takeaways

Learning objectives

After completing this module, students will be able to:



Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure



Implement compliance and security in your application infrastructure

Lesson 02: Security in the pipeline

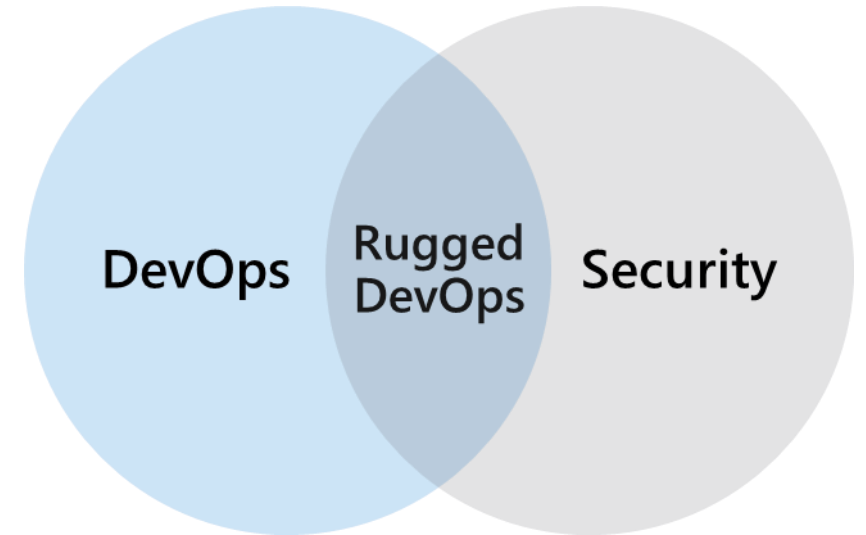


What is rugged DevOps ?

Rugged 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

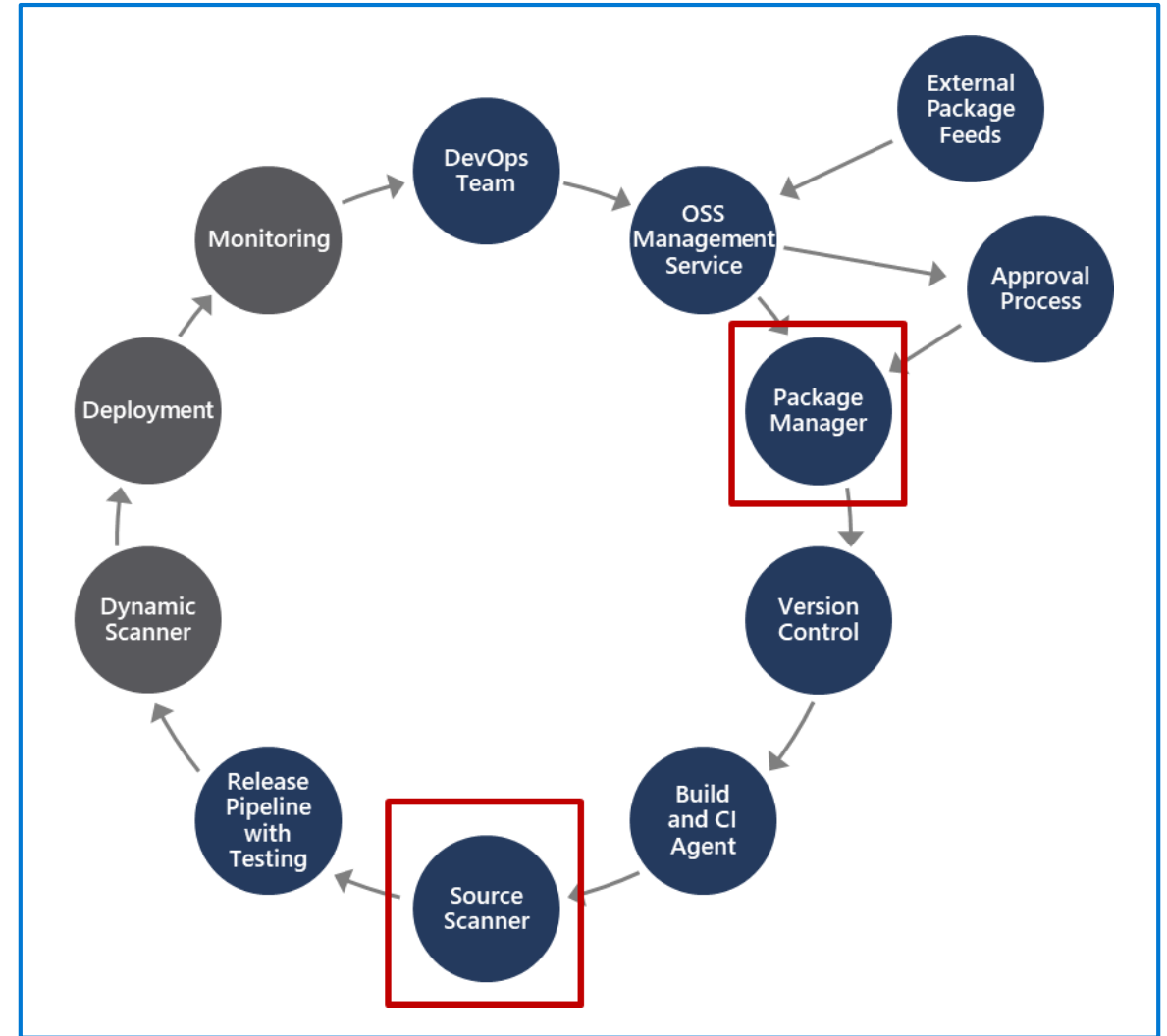


Rugged DevOps is also sometimes referred to as *DevSecOps*

Rugged 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



Software composition analysis (SCA)

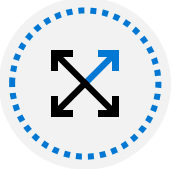
Package Management:

- 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

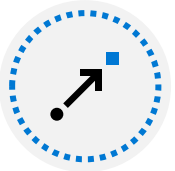
Open-Source Software (OSS) Components:

- Dependencies can have security vulnerabilities
- Ensure you have the latest version
- Check for the correct binaries
- Promptly address vulnerabilities
- Analyze the software to determine its composition can help mitigate potential vulnerabilities
- Scanning tools discussed in the next module

WhiteSource integration with Azure DevOps pipeline



WhiteSource is one such example of an extension available on the Azure DevOps Marketplace



If consuming external packages, the *WhiteSource* 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

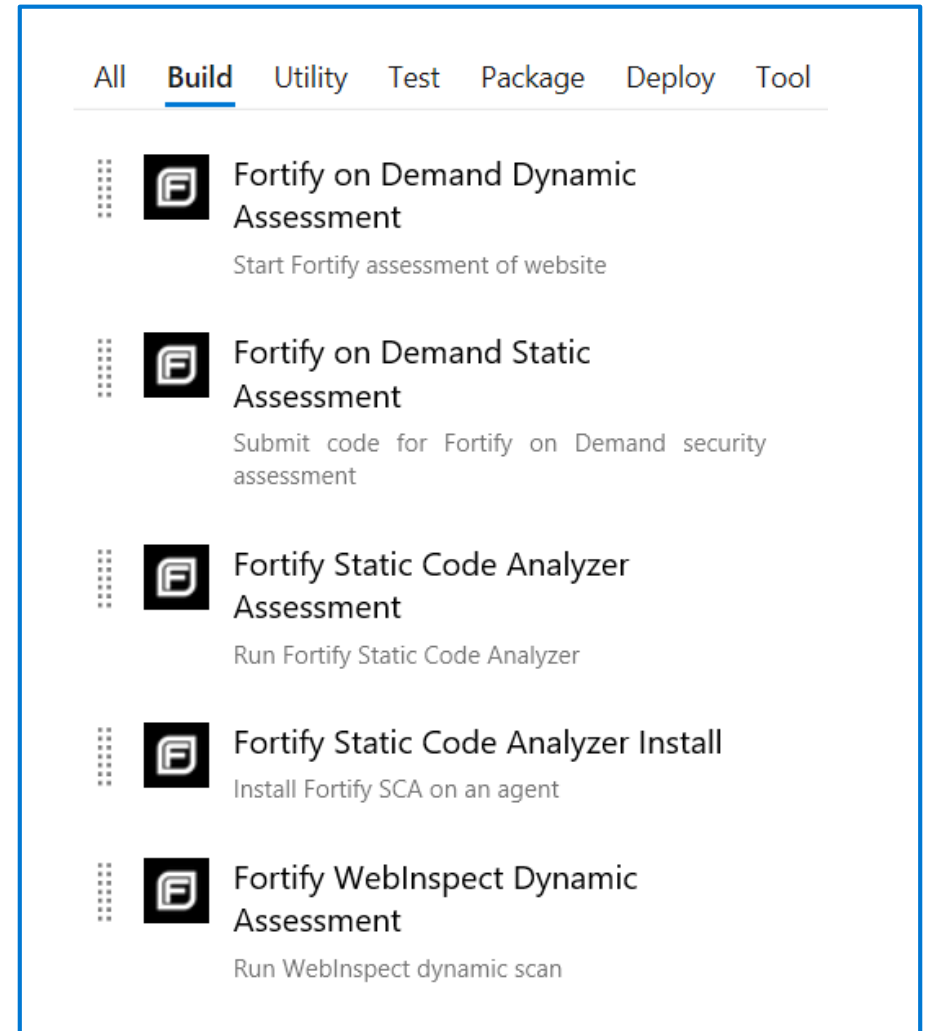
Micro Focus Fortify integration with Azure Pipelines

Can add build tasks to CI/CD pipeline to help identify vulnerabilities in your source code

Provides a comprehensive set of software security analyzers

Fortify Static Code Analyzer: Identifies root causes of software security vulnerabilities

Fortify on Demand: Automatically submits static and dynamic scan requests



The screenshot shows the 'Build' tab selected in the top navigation bar. Below the navigation bar, there is a list of five Fortify build tasks, each with a Fortify logo icon and a description:

- Fortify on Demand Dynamic Assessment**
Start Fortify assessment of website
- Fortify on Demand Static Assessment**
Submit code for Fortify on Demand security assessment
- Fortify Static Code Analyzer Assessment**
Run Fortify Static Code Analyzer
- Fortify Static Code Analyzer Install**
Install Fortify SCA on an agent
- Fortify WebInspect Dynamic Assessment**
Run WebInspect dynamic scan

Checkmarx integration with Azure DevOps

Identifies, tracks, and fixes technical and logical security flaws:

- Best fix location
- Quick and accurate scanning
- Incremental scanning
- Integrates with IDEs, build management servers, bug tracking tools, and source repositories.
- Protects your code and OSS code you use.
- Easy to initiate Open-Source Analysis with *Checkmarx Open-Source Analysis (CxOSA)* and *Checkmarx Static Application Security Testing (CxSAST)*

Add tasks | Refresh

check



Checkmarx CxSAST

Add Secure Static Source Code Analysis inside your build process

Veracode integration with Azure DevOps

Integrate application security into your development tools

Don't stop for false positives.

Align your application security practices with your development practices

Don't just find vulnerabilities, fix them

Designed to get you going quickly. Onboarding process allows for scanning on day one

Add tasks | Refresh

vera



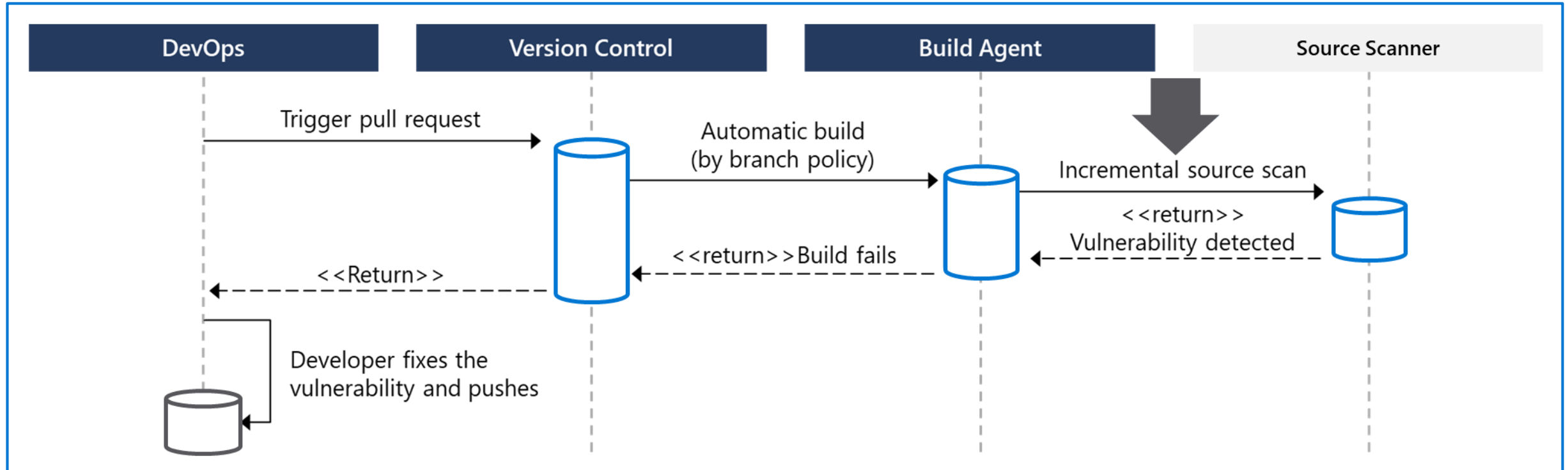
Veracode

Find and fix security defects as part of your Azure DevOps pipeline - v2.4.0

Get it free

#mw-want to try it? Look [here](#)

How to integrate software composition analysis checks into pipelines



Pull requests are the way DevOps teams submit changes

[WhiteSource](#), [Checkmarx](#), [Veracode](#), and [Black Duck by Synopsys](#) can facilitate incremental scans

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

Implementing **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 deploy using your pipeline

Manage Permissions: Secure who can edit pipelines by using RBAC

Dynamic Scanning: Test running apps, such as penetration testing

Monitoring Production: Look for anomalies related to intrusion



(#mw: [dead](#), now [AzTs](#), see [this](#).)

Secure DevOps kit for Azure (AzSK)

AzSK is a collection of scripts, tools, extensions, automations

Helps secure the subscription

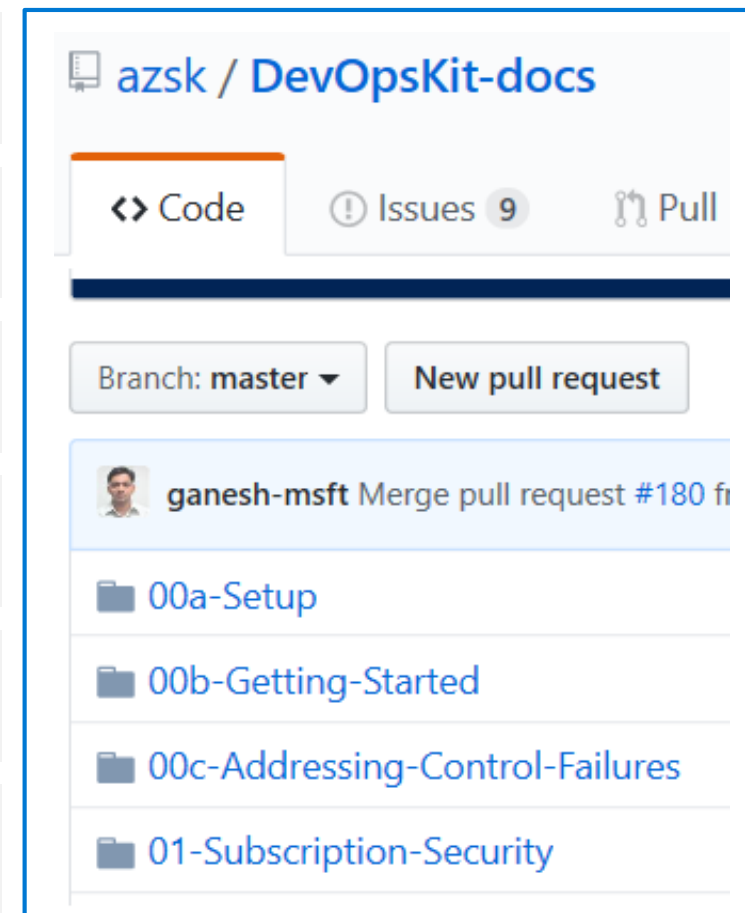
Enables secure development

Integrates security into CI/CD

Provides continuous assurance

Assists with alerts & monitoring

Governing cloud risks



Available at: <https://github.com/azsk/DevOpsKit-docs>

Lesson 03: Azure Security Center



Azure Security Center

Provide security recommendations

Monitor security settings

Monitor all your services

Use Azure Machine Learning

Analyze and identify potential attacks

Provide just-in-time (JIT) access control

Supports Windows and Linux operating systems

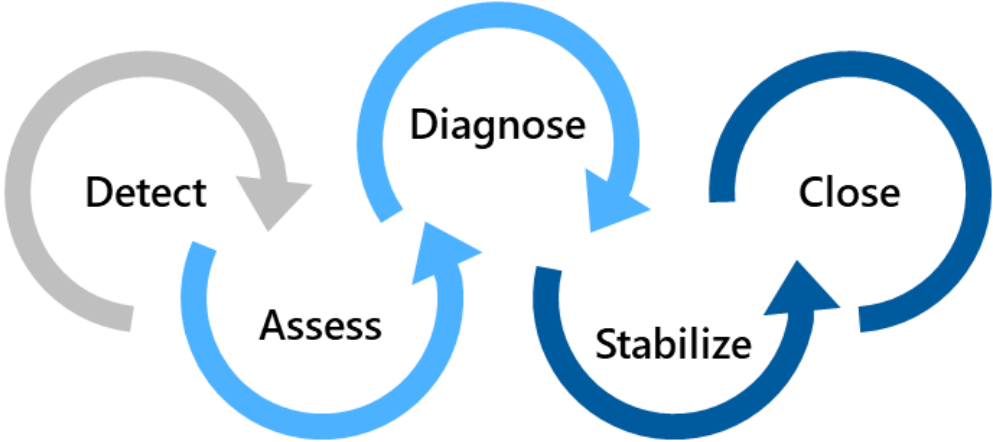
Available in two pricing tiers



Azure Security Center usage scenarios

Scenario 1

Use Security Center for an incident response



The diagram illustrates a five-step incident response cycle: Detect, Assess, Diagnose, Stabilize, and Close. 'Detect' is shown as a grey arrow, while 'Assess', 'Diagnose', 'Stabilize', and 'Close' are shown as blue arrows in a continuous loop.

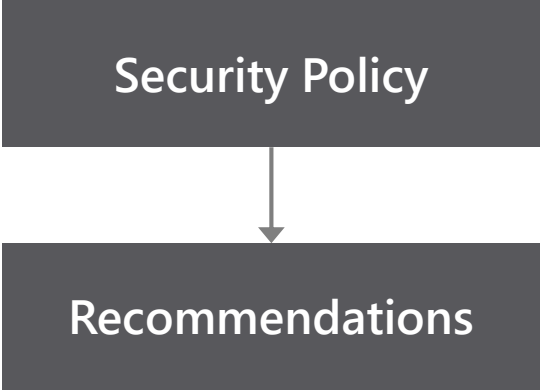
Detect – Verify a high security alert was raised

Assess – Obtain information about the alert

Diagnose – Follow the remediation steps

Scenario 2

Use Security Center recommendations to enhance security



The diagram shows a vertical flow from a 'Security Policy' box to a 'Recommendations' box, connected by a downward-pointing arrow.

Configure a security policy

Implement the recommendations for the security policy

Azure Policy

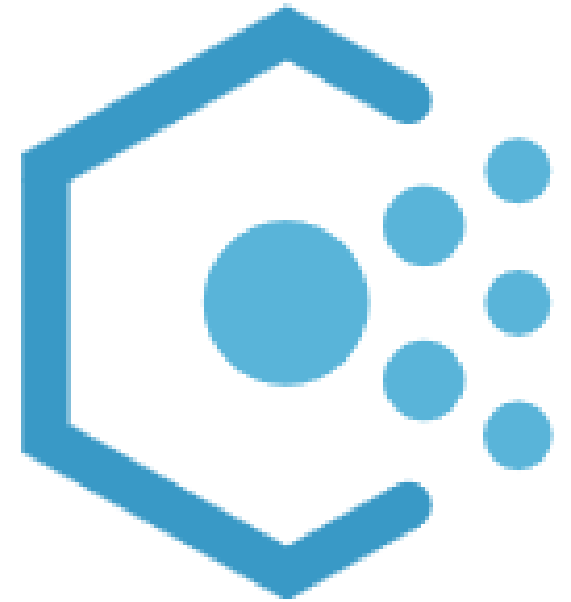
Is a service in Azure that you use to create, assign, and manage policies

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

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"
        }
    }
}
```

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 Azure Security Center.”* This would provide the following policies:

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



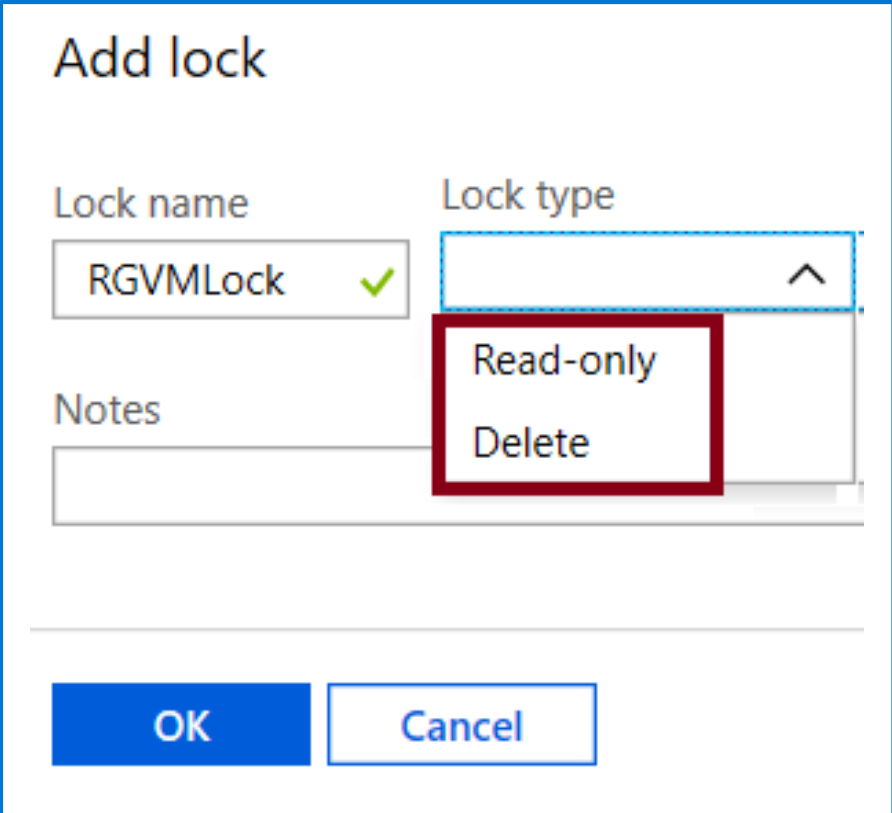
Even if you have a single policy, we recommend using initiatives if you anticipate increasing the number of policies over time

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



Add lock

Lock name: RGVMLock ✓

Lock type: Read-only (selected), Delete

Notes:

OK Cancel

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

Azure Blueprints



Allows for the definition of a repeatable set of Azure resources that implement and adhere to an organization's standards, patterns, and requirements

Can ensure a deployment meets an organization's standards, patterns, and requirements

Is a declarative way to orchestrate deployment for various resource templates and other artifacts

Provides for traceability and auditing as well as adherence



The blueprint definition (what should be deployed) and the blueprint assignment (what is deployed) supports improved deployment tracking and auditing

(#mw-a little out of date)

#mw-Yeah-Nah, see [this](#) instead:

Azure Advanced Threat Protection (ATP)

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

Azure ATP **portal** monitors and responds to suspicious activity

Azure ATP **sensor** monitors domain controller traffic

Azure ATP **cloud service** connects to Microsoft Intelligent Security Graph

The screenshot displays the Azure ATP portal interface for 'contoso-corp'. The top navigation bar includes the title 'Azure Advanced Threat Protection | contoso-corp | Timeline' and a 'PREVIEW' label. A vertical timeline on the left shows three events, each with a smiley face icon. The first event, at 4:04 PM Today, is titled 'Honeytoken activity' with an 'Updated' badge. It details activities performed by 'Bob Minion', including logging into 2 computers via Contoso-DC, authenticating from 2 computers using Kerberos, and authenticating from ITARGOET-T470S using NTLM. The second event, at 3:23 PM Jan 22, 2018, is titled 'Remote execution attempt detected' and shows an attempted remote execution of WMI methods by AdminUser on Contoso-DC from ALICE-DESKTOP. The third event, at 3:06 PM Jan 22, 2018, is titled 'Suspicious service creation' and shows AdminUser creating 10 services on Contoso-DC.

Azure Advanced Threat Protection | contoso-corp | Timeline

PREVIEW

4:04 PM Today

Honeytoken activity **Updated**

The following activities were performed by Bob Minion:

- Logged in to 2 computers via Contoso-DC.
- Authenticated from 2 computers using Kerberos when accessing 5 resources against Contoso-DC.
- Authenticated from ITARGOET-T470S using NTLM against corporate resources via Contoso-DC.

Started at 3:08 PM Jan 22, 2018

3:23 PM Jan 22, 2018

Remote execution attempt detected

The following remote execution attempts were performed on Contoso-DC from ALICE-DESKTOP:

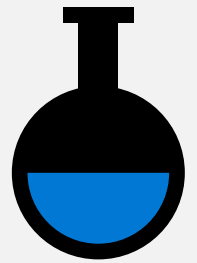
- Attempted remote execution of one or more WMI methods by AdminUser.

3:06 PM Jan 22, 2018

Suspicious service creation

AdminUser created 10 services in order to execute potentially malicious commands on Contoso-DC.

Lesson 04: Lab



Implement security and compliance in Azure DevOps Pipelines



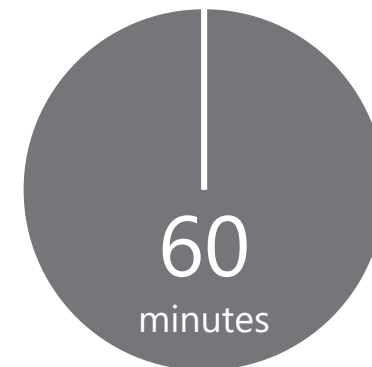
Lab overview:

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

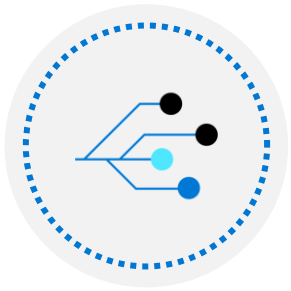
Duration:



Lesson 05: Module review and takeaways



What did you learn?



Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure



Implement compliance and security in your application infrastructure

Module review questions

1 Rugged DevOps combines which two elements?

2 Which term broadly defines what security means in rugged 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?

Module 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 DevOps 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?