



Microsoft Defender for Cloud

Yuri Diogenes
Tom Janetscheck

foreword by Gilad Elyashar,
Partner Director on Product Management, Microsoft Cloud Security



Microsoft Defender for Cloud

Yuri Diogenes and Tom Janetscheck

Microsoft Defender for Cloud

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

Copyright © 2023 by Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-13-787845-1

ISBN-10: 0-13-787845-1

Library of Congress Control Number: 2022944665

ScoutAutomatedPrintCode

TRADEMARKS

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

PERMISSIONS

Chapter 2, Icons of AWS: Amazon Web Services, Inc, Chapter 6, Figure 06-30: United States Department of Commerce

WARNING AND DISCLAIMER

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

SPECIAL SALES

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

CREDITS

EDITOR-IN-CHIEF

Brett Bartow

EXECUTIVE EDITOR

Loretta Yates

SPONSORING EDITOR

Charvi Arora

DEVELOPMENT EDITOR

Rick Kughen

MANAGING EDITOR

Sandra Schroeder

PROJECT EDITOR

Tracey Croom

COPY EDITOR

Rick Kughen

INDEXER

Tim Wright

PROOFREADER

Jen Hinchliffe

TECHNICAL EDITOR

Liana Tomescu

EDITORIAL ASSISTANT

Cindy Teeters

COVER DESIGNER

Twist Creative, Seattle

COMPOSITOR

codeMantra

GRAPHICS

codeMantra

Contents at a Glance

	<i>Acknowledgments</i>	<i>xiii</i>
	<i>About the authors</i>	<i>xv</i>
	<i>Foreword</i>	<i>xvii</i>
	<i>Introduction</i>	<i>xix</i>
CHAPTER 1	The threat landscape	1
CHAPTER 2	Planning Microsoft Defender for Cloud adoption	27
CHAPTER 3	Onboarding Microsoft Defender for Cloud	45
CHAPTER 4	Policy management	67
CHAPTER 5	Strengthening your security posture	105
CHAPTER 6	Threat detection	155
CHAPTER 7	Better together	189
CHAPTER 8	Enhanced security capabilities	201
CHAPTER 9	Accessing Defender for Cloud from APIs	223
CHAPTER 10	Deploying Microsoft Defender for Cloud at scale	235
APPENDIX	Microsoft Defender for DevOps	245
	<i>Index</i>	<i>259</i>

About the authors

Yuri Diogenes, MsC

Yuri holds a Master of Science in cybersecurity intelligence and forensics investigation from UTICA College and is currently working on his Ph.D. in Cybersecurity Leadership from Capitol Technology University. Yuri has been working at Microsoft since 2006, and currently, he is a Principal PM Manager for the CxE Microsoft Defender for Cloud Team. Yuri has published a total of 26 books, mostly about information security and Microsoft technologies. Yuri is also a professor at EC-Council University, where he teaches in the Bachelor in Cybersecurity Program. Yuri holds an MBA and many IT/Security industry certifications, such as CISSP, MITRE ATT&CK Cyber Threat Intelligence Certified, E|CND, E|CEH, E|CSA, E|CHFI, CompTIA Security+, CySA+, Network+, CASP, and CyberSec First Responder. You can follow Yuri on Twitter at @yuridiogenes.

Tom Janetscheck

Tom is a Senior Program Manager in the CxE Microsoft Defender for Cloud team, where he works with his friend Yuri, helping customers onboard and deploy Microsoft Defender for Cloud. As a former Microsoft MVP, Tom joined the team during COVID-19 in Spring 2020, and he deeply missed in-person conferences, as he loves to speak to audiences all over the world. With almost 20 years of experience in various IT admin and consulting roles, Tom has a deep background in IT infrastructure and security, and he holds various certifications, including MCSE and MCTS. When Tom is not writing a book, preparing a conference or user group session, or helping his customers onboard Defender for Cloud, he is an enthusiastic motorcyclist, scuba diver, and musician. He plays the guitar, bass, and drums. He also volunteers as a firefighter at his local fire department and can usually be met attending rock concerts all over the place. You can follow Tom on Twitter at @azureandbeyond.

Foreword

As customers' path toward the cloud and digital transformation continues, we see increased complexity in our cloud environments, moving from traditional VM workloads to cloud-native applications and leveraging an increasing selection of PaaS services. This introduces new challenges to cloud providers, security vendors, and security teams who have to familiarize themselves with dozens—or even hundreds—of PaaS services and ensure each is secured properly, given the correct context.

Securing these cloud workloads starts with reducing the attack surface by maintaining the security posture and defense-in-depth. This can be quite a challenge given the variety and the sheer number of posture misconfigurations and vulnerabilities found on an average cloud workload. This book goes into detail on how Defender for Cloud can be used to fully visualize the customer's cloud estate. It also helps identify the attack surface across all workload types (prioritizing risks using Secure Score, guiding customers to which threat to address first, and providing the customers with at-scale tooling to build cloud-native applications that are secure from day-1). Lastly, this book helps you enforce the correct set of policies to avoid drift.

While posture management is a must, it must be complemented with threat detection capabilities that can detect sophisticated attackers in a timely manner and assist SOC teams' response by blocking or mitigating these threats. In this book, Yuri and Tom share their knowledge of how Defender for Cloud identifies cyberattacks by leveraging signals from across the cloud workload, including VMs, containers, PaaS access logs, admin activity, networking, and more. And they tell you how this knowledge can be applied in a modern SOC to respond to such attacks.

If you are an IT or Security leader, I highly recommend you share this book with your teams. It is relevant to any organization that needs to protect and defend IT workloads across clouds and hybrid environments.

Gilad Elyashar,
Partner Director on Product Management
Microsoft Cloud Security

Introduction

Welcome to *Microsoft Defender for Cloud*, a book that was developed together with the Microsoft Defender for Cloud product group to provide in-depth information about Microsoft Defender for Cloud and to demonstrate best practices based on real-life experience with the product in different environments.

The purpose of this book is to introduce the wide array of security features and capabilities available in Microsoft Defender for Cloud. After being introduced to all of these security options, you will dig in to see how they can be used in a number of operational security scenarios so that you can get the most out of the protect, detect, and respond skills provided only by Microsoft Defender for Cloud.

Who is this book for?

Microsoft Defender for Cloud is for anyone interested in Azure security or multcloud security: security administrators, support professionals, developers, and engineers.

Microsoft Defender for Cloud is designed to be useful for the entire spectrum of Azure users. You will find this book to be a valuable resource, regardless of whether you have no security experience, some experience, or you are a security expert. This book provides introductory, intermediate, and advanced coverage on a large swath of security issues that are addressed by Microsoft Defender for Cloud.

The approach is a unique mix of didactic, narrative, and experiential instruction. Didactic covers the core introductions to the services. The narrative leverages what you already understand, and we bridge your current understanding with new concepts introduced in the book.

Finally, we share our experiences with Microsoft Defender for Cloud, how to get the most out of it by showing, in a stepwise, guided fashion, how to configure it to gain all the benefits it has to offer.

In this book, you will learn:

- How to secure your Azure assets no matter what your level of security experience
- How to protect resources in AWS and GCP

- How to save hours, days, and weeks of time by removing the need for trial and error
- How to protect, detect, and respond to security threats better than ever by knowing how to get the most out of the different Microsoft Defender for Cloud plans

System requirements

- Anyone with access to a Microsoft Azure subscription can use the information in this book. If you are integrating with AWS and GCP, you will also need an account on each cloud provider.

Errata, updates & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

MicrosoftPressStore.com/DefenderforCloud/errata

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit <http://www.MicrosoftPressStore.com/Support>.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to <http://support.microsoft.com>.

Stay in touch

Let's keep the conversation going! We're on Twitter:
<http://twitter.com/MicrosoftPress>.

Appendix

Microsoft Defender for DevOps

By GEORGE WILBURN,
PRINCIPAL PM
DEFENDER FOR DEVOPS

Attackers know there is a treasure trove of information in source code. This has turned source code management systems into an increasingly high-value attack target. Regardless of whether attackers leverage the source code for initial access or as an exfiltration point, or use it to harvest valuable data such as credentials, the source code and systems that house it must now be on the frontline of the defense. Adversaries can find ways to attack running systems by exfiltrating and examining source code. While the source code itself is valuable—think *ransomware scenarios and corporate infiltration points*—if adversaries can command and control the source code lifecycle, they can achieve a ripple effect, allowing them to infiltrate customers who install compromised software.

Preventing and detecting these attacks is high on the priority list of most CISOs and Security organizations today, and Microsoft is positioned to help its customers protect these source code assets against this modern threat.

In this appendix, you will learn more about the latest addition to Microsoft Defender for Cloud services, Defender for DevOps.

Shift left

Source code management systems, such as GitHub and Azure DevOps, are complicated systems with a lot of moving parts that are difficult to monitor. Developers are constantly pushing and pulling updates to source code. With the distributed nature of Git, source code repositories are forked and cloned continuously to other repositories and local machines. With every software push into source code management systems, builds are initiated. These builds require build definitions, credentials, dependencies, and more to get the software ready to push to production. Once built, the solution runs a company's SaaS offering, is deployed as a cloud workload, or is prepared to be pulled down

as updates or patches to existing systems. All this is to say that the software supply chain comprises a lot of activity that requires monitoring and provides numerous opportunities for attacks in the software lifecycle. Monitoring all this activity can generate false-positives, and distinguishing innocuous behavior from malicious behavior is difficult.

The cybersecurity industry recognizes that source code management systems, source code, and the build and deployment pipelines must be protected. Security operators have frequently been caught off-guard by the latest attacks on source code management systems and have lacked comprehensive knowledge of these assets and systems and the ability to trace them back to a development team or owner. Security organizations need visibility into these systems and the security of the source code throughout the development lifecycle. Microsoft's latest Defender for Cloud product—Defender for DevOps—helps Security Operations gain visibility into the security of source code management systems and source code. This helps customers understand—at enterprise scale—whether their code repositories are staying secure and their software is being developed securely. With this visibility, security teams can take action to harden the security of pre-deployment systems and assets by shifting security activities left to uncover security issues before they make it to production.

Microsoft's CEO, Satya Nadella, has asserted that every company is now a software company. As companies in every industry—from financial services to manufacturing to healthcare—have become more reliant on software to run their business, they must increasingly rely on their developers to create robust and secure code. However, most developers are not security experts and need help solving security issues while they code and when code is checked into source code management systems. Defender for DevOps offers dual-layered protection by looking for security issues as code moves through the development lifecycle—first, on the developer's local machine, and then as code is checked into a repository and moves through the build process. Providing developers with security tools is not new, but it is now a critical part of the defense-in-depth story that enables companies to protect critical software assets and ensure security across the development ecosystem.

As security and development teams work together to address the great DevSecOps divide and connect security, operations, and development, securing the pipeline is where their efforts converge in shifting security to the left (see Figure A-1 later in this chapter). The pipeline is where the security team implements processes and tools to address security posture and compliance requirements, and the development team passes code through their checks to ensure software meets operational deployment requirements as code makes its way to production. Defender for DevOps bridges the gap in this journey toward secure code by helping companies start from a secure position and helping teams operate securely and efficiently throughout the code build and deployment lifecycle.

In this appendix, you will learn how to use Microsoft Defender for Cloud (MDC) to monitor the security posture of your source code management systems, understand new DevOps recommendations in MDC, configure the Microsoft Security DevOps tools, and find vulnerabilities in code.

Understanding Defender for DevOps

Defender for DevOps is a new addition to the Microsoft Defender for Cloud family that helps you discover, monitor, and detect threats to your source code management systems and source code. The service requires a connection to the source code management systems to allow it to discover resources such as repositories, organizations, projects, and code and to initiate assessments of the security posture of these assets.

Defender for DevOps also provides a dedicated dashboard to visualize the discovered assets and configure additional features such as pull request annotations and analytics about the source code management system activities. As illustrated in Figure A-1, Defender for DevOps connects your source code management systems to Defender for Cloud to utilize MDC's rich security capabilities to shift security left and protect your pre-deployment assets.

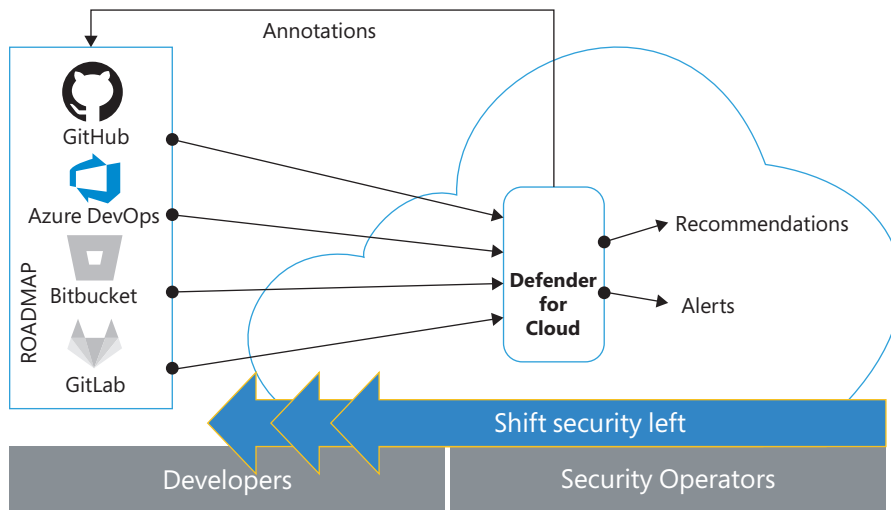


FIGURE A-1 Defender for DevOps connects source code management systems to Microsoft Defender for Cloud

Defender for DevOps brings the security and development teams together and enables collaboration between security operators (SecOps) and developers as code is developed and security issues are discovered and remediated. Let's examine one scenario where Defender for DevOps can help your organization bridge the division between your security and development teams. In MDC, security operators receive security recommendations about issues discovered in repositories and source code. This new information helps SecOps understand, locate, and provide guidance to act on DevOps-related issues, though developers must address these security issues in the code. Now, SecOps can communicate directly with developers by enabling pull request annotations with information about the issue, location, and actionable steps to remediate. Developers, in turn, use the same information provided in the pull request annotations to fix security issues before the code is merged.

Preventing security issues from getting to production is a crucial step in securing the software development lifecycle. Now, when a security incident arises and SecOps and developers come together to review and take action, both teams have the same information from Defender for DevOps, unlocking more meaningful and richer conversations to address identified security risks.

To maintain the security posture of source code management systems, Defender for DevOps provides always-on checks, such as ensuring authorization and authentication settings are configured and ensuring good code hygiene is followed in repositories by assessing whether code, secret, and dependency scanning is enabled. Additionally, assessments are performed on the security configuration of pipelines, service connections, webhooks, and many other configurations to ensure the system is operating securely. These posture assessments ensure that developers work in a secure, hardened environment and that their code is protected from adversaries.

For developers, Defender for DevOps provides the Microsoft Security DevOps (MSDO) tools that can be configured on every repository. These tools help ensure that, as developers check in code, the code is secure and free from common security defects, such as secrets left in code, code security vulnerabilities, Infrastructure as Code (IaC) security risks, and container vulnerabilities. The MSDO tools can also be leveraged from the command-line interface in developers' local environments to help them find and fix security vulnerabilities before pushing code to a repository. If desired, the same tools can even be configured to break a build and not allow code to be deployed to production until developers resolve all identified security issues.

In summary, Defender for DevOps offers new capabilities in three critical areas:

- 1.** Discovery and visibility of source code management systems provide an inventory of DevOps assets—a known blind spot for security organizations—and enables statistics about security vulnerabilities and improvements
- 2.** Continuous posture assessment ensures the SCMS is configured securely and provides recommendations with remediation guidance to address security posture misconfigurations.
- 3.** Code vulnerability management finds and prevents vulnerable source code, IaC templates, containers, and secrets and helps developers find and address security vulnerabilities while code is written.

Defender for DevOps unites security operators and developers like never before. It enables each team to work in the same familiar environments, tools, and experiences that they're used to—SecOps in Microsoft Defender for Cloud and developers in their chosen development environments. It allows each team flexibility in when and where they address discovered security issues—during development, at the pipeline, and in security operations. Teams can communicate seamlessly and collaborate across multiple layers of protection, enabling security organizations to shift left and address security early in the development lifecycle.

Connect your source code management system to Defender for Cloud

To start using Defender for DevOps, you must connect your source code management system (SCMS) to Microsoft Defender for Cloud. By onboarding your SCMS to MDC, you authenticate to your source code management system(s), such as Azure DevOps and GitHub, and then grant authorization to Defender for DevOps to access your repositories.

Start setting up an SCMS connection to Defender for DevOps by logging into the Azure portal, opening Microsoft Defender for Cloud, and clicking **Environment Settings**, where you'll find the **Add Environment** menu. Next, click a source code management system in the menu to begin adding your connector. Connecting to GitHub is shown in Figure A-2.

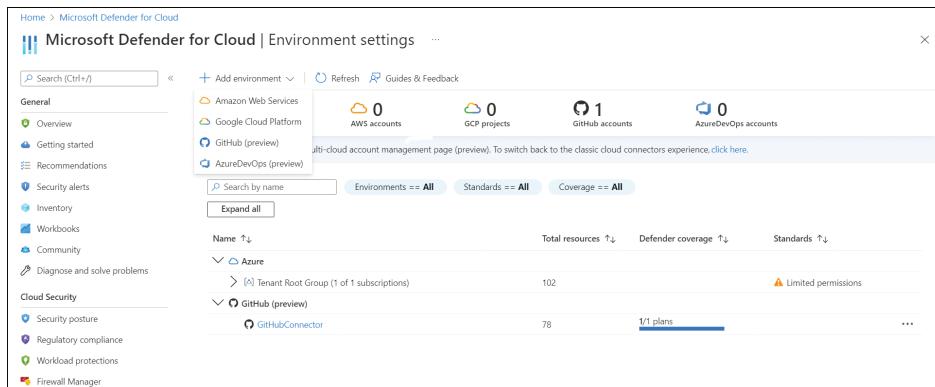


FIGURE A-2 Using the Add environment menu to connect to GitHub

The next onboarding steps require you to give your connector a **Name** and select a **Region**, **Subscription**, and **Resource Group** where the connector will be stored, as shown in Figure A-3.

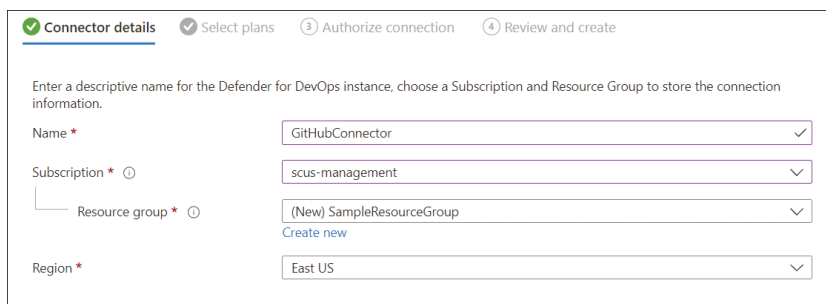


FIGURE A-3 Basic information required to onboard your connector

The steps shown in Figure A-4 authorize your GitHub account. After granting authorization, the installation adds the Defender for DevOps GitHub App to your organization and allows the Defender for DevOps service to access your repositories.

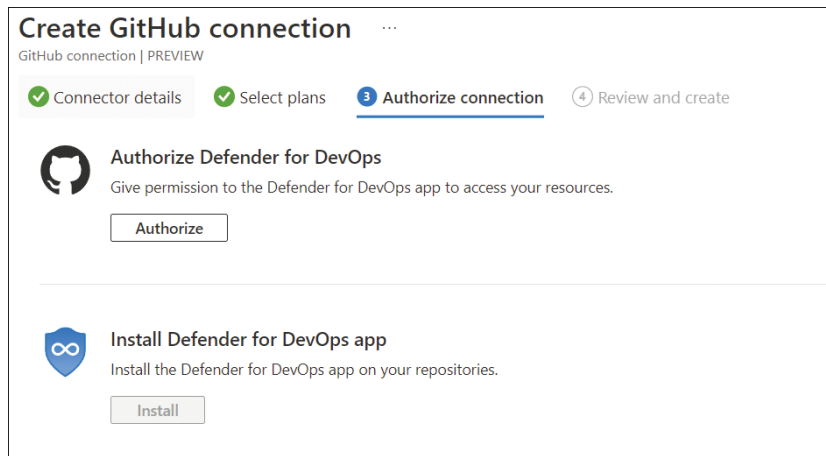


FIGURE A-4 Defender for DevOps authorization and app installation

After onboarding is complete, your source code management system’s connector is displayed as one of the connectors in MDC’s **Environment Settings** blade, as shown in Figure A-5. You can change the repositories you selected during onboarding by clicking the connector.

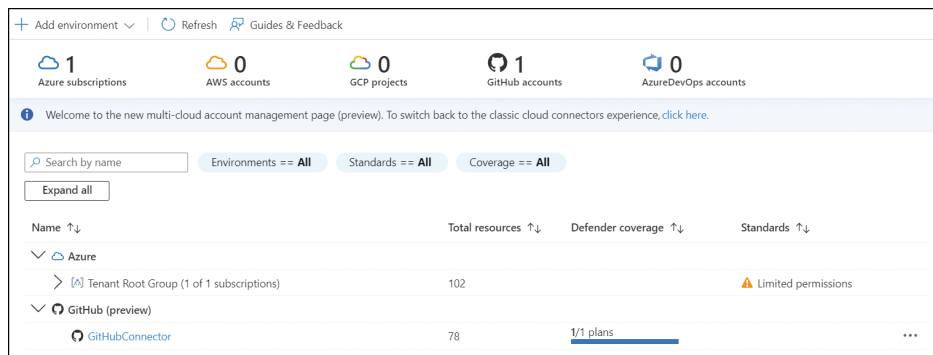


FIGURE A-5 SCMS connector in Environment settings

The Defender for DevOps service now starts discovering your repositories and analyzing them for any security issues. Once discovered, the **Inventory** dashboard shows the repositories, and the **Recommendations** dashboard shows any security issues related to a repository.

Once you have enabled Defender for DevOps, Defender for Cloud continuously scans your source code management system’s resources and provides security recommendations. If any

security issues are found in repositories, a recommendation is created for each type of finding. Remediating these issues reduces your attack surface and increases your Secure Score.

Defender for DevOps analyzes your repositories and creates recommendations that help you harden the security posture of

- Source code management systems, repositories, projects, and organizations
- Source code files, credentials in code, and dependencies in your solutions
- Infrastructure as Code templates
- Container images and Docker files

Defender for DevOps' recommendations seamlessly enhance Defender for Cloud's recommendations by utilizing the same MDC dashboard and the same familiar recommendations experience. Recommendations can be numerous in large and complex environments, so the filtering and exemption capabilities in the **Recommendations** dashboard can help you manage your investigations and remediate findings. For example, to see recommendations specific to GitHub, you can filter by **Resource Type** to show only **GitHub Repositories**, as shown in Figure A-6.

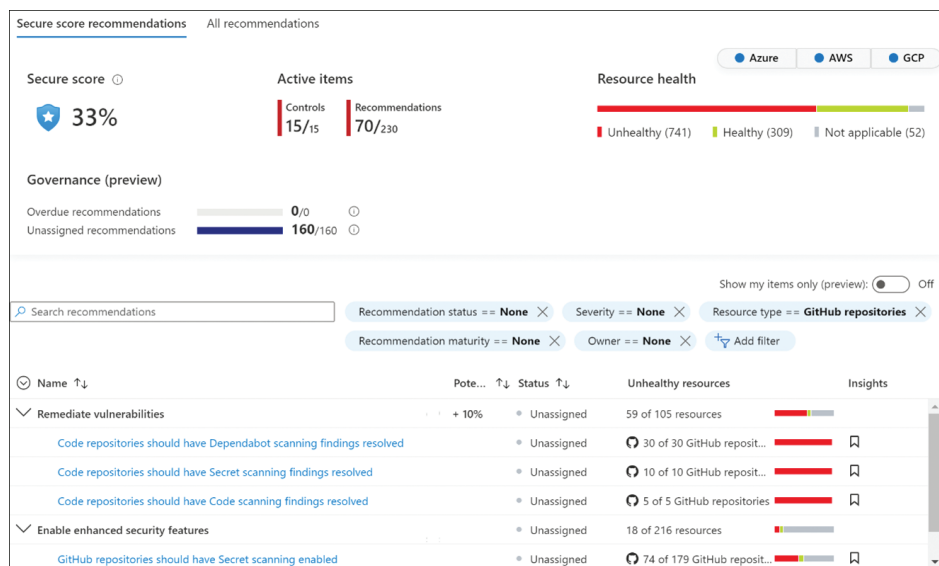


FIGURE A-6 Recommendations filtered by Resource Type: GitHub Repositories

To see additional details about a specific recommendation, simply click it to see more information in the recommendation's context blade. In the example shown in Figure A-7, you can see the result of the **Code Repositories Should Have Secret Scanning Findings Resolved** recommendation.

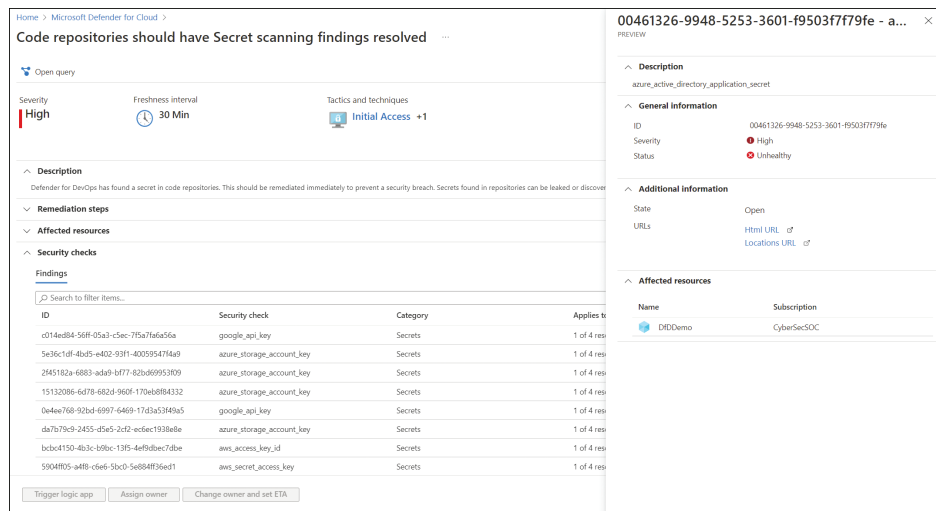


FIGURE A-7 Defender for DevOps Recommendation to resolve secrets found in repositories

When reviewing the findings in this recommendation, you can see that Defender for DevOps found multiple secrets in the code repositories. Secrets found in repositories can be leaked or discovered by adversaries, compromising an application or service. All these credentials should be assumed to be compromised. Clicking each credential displays the context blade where additional information can be found to assist in your investigation, such as a deep link to the location of the file containing the credential. To remediate this recommendation, the credential should be removed from the source code and rotated out of its source system. The code containing this credential should be refactored to use a secure secret store such as Azure Key Vault.

Configure pull request annotations

As you learned earlier, Defender for DevOps has new capabilities like pull-request annotations that help security operators assist developers when adopting security best practices early in the development cycle. Also, these new capabilities help security operators address high-risk vulnerabilities quickly and easily before those vulnerabilities make it into production. Defender for DevOps provides the centralized configuration for pull-request annotations and a way to keep track of the current status of all annotations across the DevOps estate.

Security operators can enable pull-request annotations so that developers receive security findings directly on their pull requests to remediate security issues before merging into the main branch. Developers can then interact with the pull-request annotations to determine when the issue can be prioritized and fixed or explain why an issue cannot be remediated, meaning it needs to be dismissed or suppressed until a later release.

This capability facilitates bidirectional communication between security operators and developers about discovered security issues, recommended guidance to remediate, and which security issues developers have accepted and fixed, have not fixed, or will fix in a later release.

Discover security issues when developers commit code

Defender for DevOps has a set of static analysis tools called Microsoft Security DevOps (MSDO). MSDO contains a combination of Microsoft and open-source tools that scan for security vulnerabilities. These tools scan for credentials left in code files, security vulnerabilities in Infrastructure as Code templates, vulnerable containers and Docker files, and more. You can configure the MSDO tools to break a build—that is, not allow the code to continue moving to the next development step—if any security issues are discovered during the automated scan. This helps prevent developers from merging un-remediated vulnerabilities into a main branch and deploying them to production.

As security teams mature their DevSecOps practices, they can set up the MSDO tools to run automatically when a developer commits code to a repository. This layer of protection helps find and block code vulnerabilities and prevents them from making it to a production resource. To add this layer of protection, you need to configure the MSDO tools in your GitHub workflows or Azure DevOps pipelines.

For example, you can configure the MSDO tools to run in Azure DevOps on a pipeline. Start the configuration by selecting a project and clicking **Pipelines**, as shown in Figure A-8, to create a new pipeline.

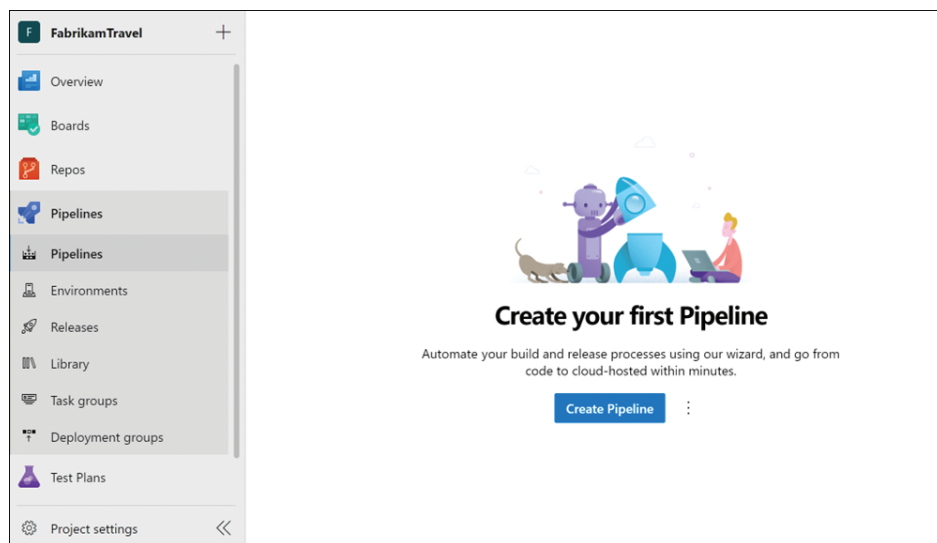


FIGURE A-8 Create a new pipeline in Azure DevOps

The YAML code shown in Figure A-9 is an example of how to create a new pipeline. As shown in the following sample, by including the input parameter `break = true`, if any security issues are found, the build will break until all security issues are resolved, and the scan completes successfully.

After successfully creating the pipeline, you are ready to scan your code for security vulnerabilities.

```

# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
- main

pool:
  vmImage: windows-latest

steps:
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 3.1.x
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 5.0.x
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 6.0.x
- task: MicrosoftSecurityDevOps@1
  displayName: 'Microsoft Security DevOps'
  #Optional to break build
  inputs:
    break: true

```

FIGURE A-9 Create a new pipeline in Azure DevOps

One of the MSDO tools that security operators have most highly anticipated is the Microsoft Credential Scanner tool. During the MSDO scan of Azure DevOps, the Credential Scanner finds secrets that developers have checked in. The increase in attacks on source code management systems helps your organization find and remediate credential leaks and keep them out of source code files. Scan results from the MSDO Credential Scanner are shown in Figure A-10.

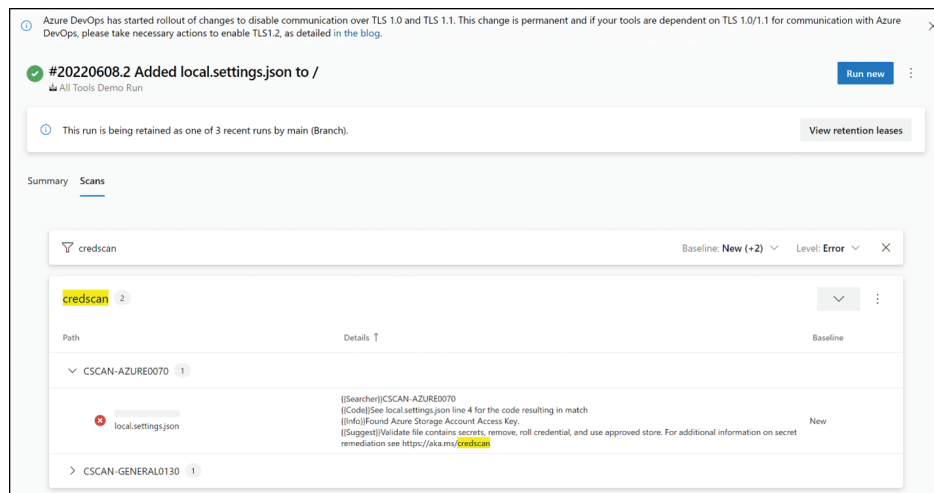


FIGURE A-10 MSDO results filtered for Credential Scanner showing a Storage Account Key found in code

Discover security issues in Infrastructure as Code (IaC)

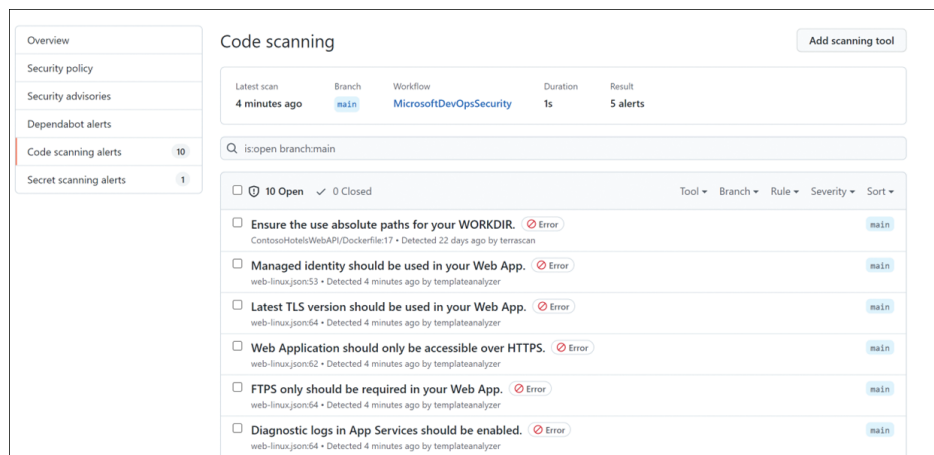
One of the most requested capabilities is the ability to scan Infrastructure as Code templates for security issues and security best practices in your CI/CD pipelines. If you would like to run only IaC scanning and not the other MSDO tools, this can be done by enabling the MSDO tools and configuring them to scan only Infrastructure as Code templates.

The example shown in Figure A-11 has the lines that you should add to your workflow so that it will only perform IaC scanning.

```
26 # Run analyzers
27 - name: Run Microsoft Security DevOps Analysis
28   uses: microsoft/security-devops-action@preview
29   id: msdo
30   with:
31     categories: 'IaC'
```

FIGURE A-11 Sample code to scan only IaC files

To see the IaC scan results in GitHub, after your workflow is configured and runs on your code, click the **Security** tab in your repository and then click **Code Scanning Alerts**, as shown in Figure A-12.



The screenshot shows the GitHub Code scanning dashboard. On the left, there is a sidebar with navigation options: Overview, Security policy, Security advisories, Dependabot alerts, Code scanning alerts (10), and Secret scanning alerts (1). The main content area is titled 'Code scanning' and includes a summary of the latest scan: '4 minutes ago', 'main' branch, 'MicrosoftDevOpsSecurity' workflow, '1s' duration, and '5 alerts'. Below this is a search bar with the text 'is:open branch:main'. A table of alerts is displayed, showing 10 open alerts and 0 closed alerts. The alerts are listed with checkboxes, tool names, branch names, and severity levels. The alerts shown are:

Alert	Tool	Branch	Severity
Ensure the use absolute paths for your WORKDIR. Error	terrascan	main	Error
Managed identity should be used in your Web App. Error	templateanalyzer	main	Error
Latest TLS version should be used in your Web App. Error	templateanalyzer	main	Error
Web Application should only be accessible over HTTPS. Error	templateanalyzer	main	Error
FTPS only should be required in your Web App. Error	templateanalyzer	main	Error
Diagnostic logs in App Services should be enabled. Error	templateanalyzer	main	Error

FIGURE A-12 IaC scan results in the GitHub Code scanning dashboard

To see the IaC scan results in Azure DevOps, click your pipeline's **Summary** tab, as shown in Figure A-13.

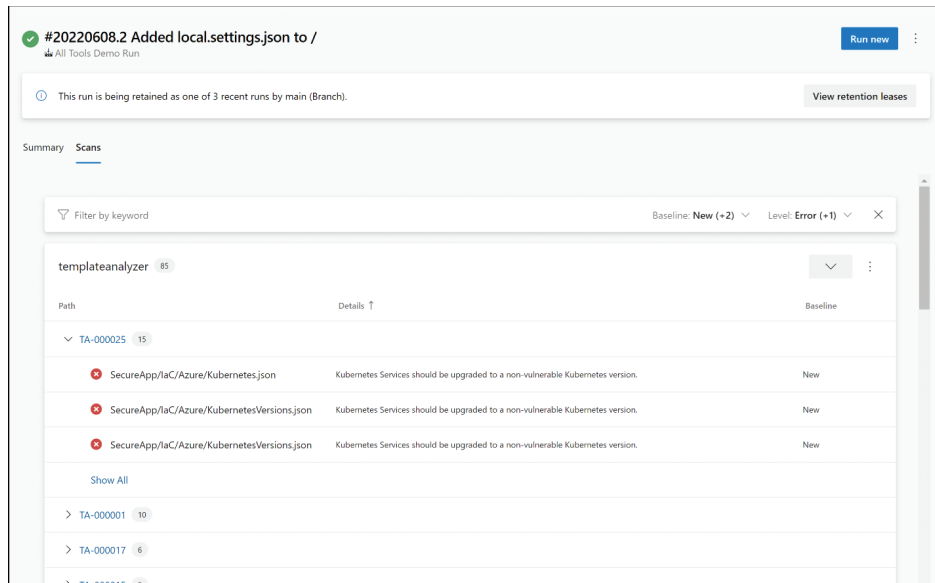


FIGURE A-13 IaC scan results in the Azure DevOps SARIF SAST Scans Tab viewer

Discover security issues during development

The MSDO tools can be configured to run on a pipeline, as explored in a previous section, but also have a command-line option that can be downloaded to a developer’s local machine to help find security issues during development.

Let’s look at one of these MSDO CLI tools—the ARM Template Best Practice Analyzer (BPA)—in more detail. ARM templates are Infrastructure as Code JSON files used to create Azure resources through automation. This tool analyzes ARM templates for best practices and security issues. It also helps developers address these issues by providing recommended steps to easily fix security issues. The Template Analyzer results example in the figure below shows security issues related to the `azuredeploy.json` ARM template. The terminal in the lower half of Figure A-14 shows the Template Analyzer output with these discovered issues from the `azuredeploy.json` ARM template being developed in the screen's top half.

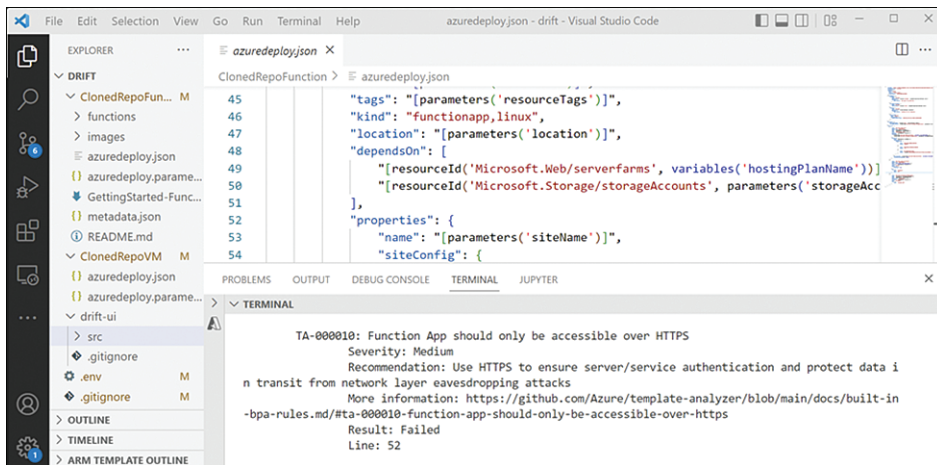


FIGURE A-14 MSDO Template Analyzer

Exploiting leaked credentials is a primary mechanism used to breach cloud resources. Leaked credentials should be remediated before code is pushed into a repository. Another MSDO CLI tool that helps developers find and address this security issue is the Credential Scanner tool. This tool scans all files in your code project and notifies developers when it finds a credential. Figure A-15 shows a code file named `GenerateLicenseFile.cs` that contains a general password.

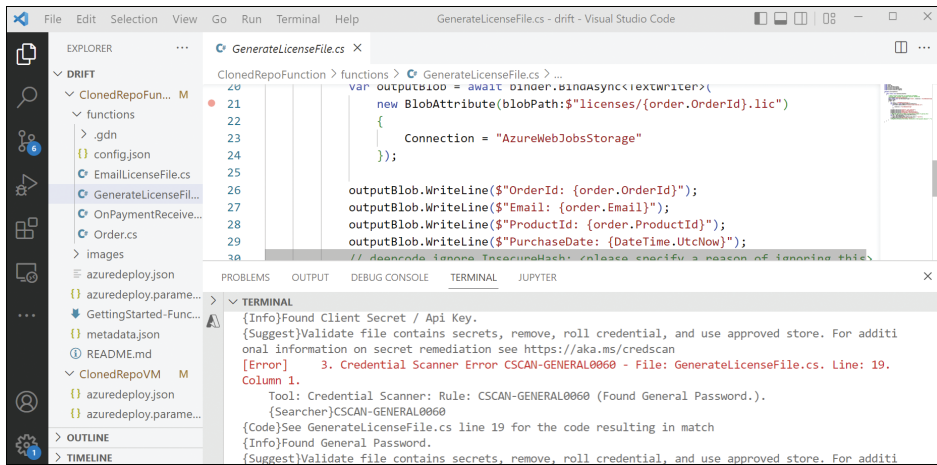


FIGURE A-15 MSDO Credential Scanner

Hear about it first.

Since 1984, Microsoft Press has helped IT professionals, developers, and home office users advance their technical skills and knowledge with books and learning resources.

Sign up today to deliver exclusive offers directly to your inbox.

- New products and announcements
- Free sample chapters
- Special promotions and discounts
- ... and more!

MicrosoftPressStore.com/newsletters

