

Microsoft Reactor Lab Guide

By Modus Create

moduscreate.com



Reactor



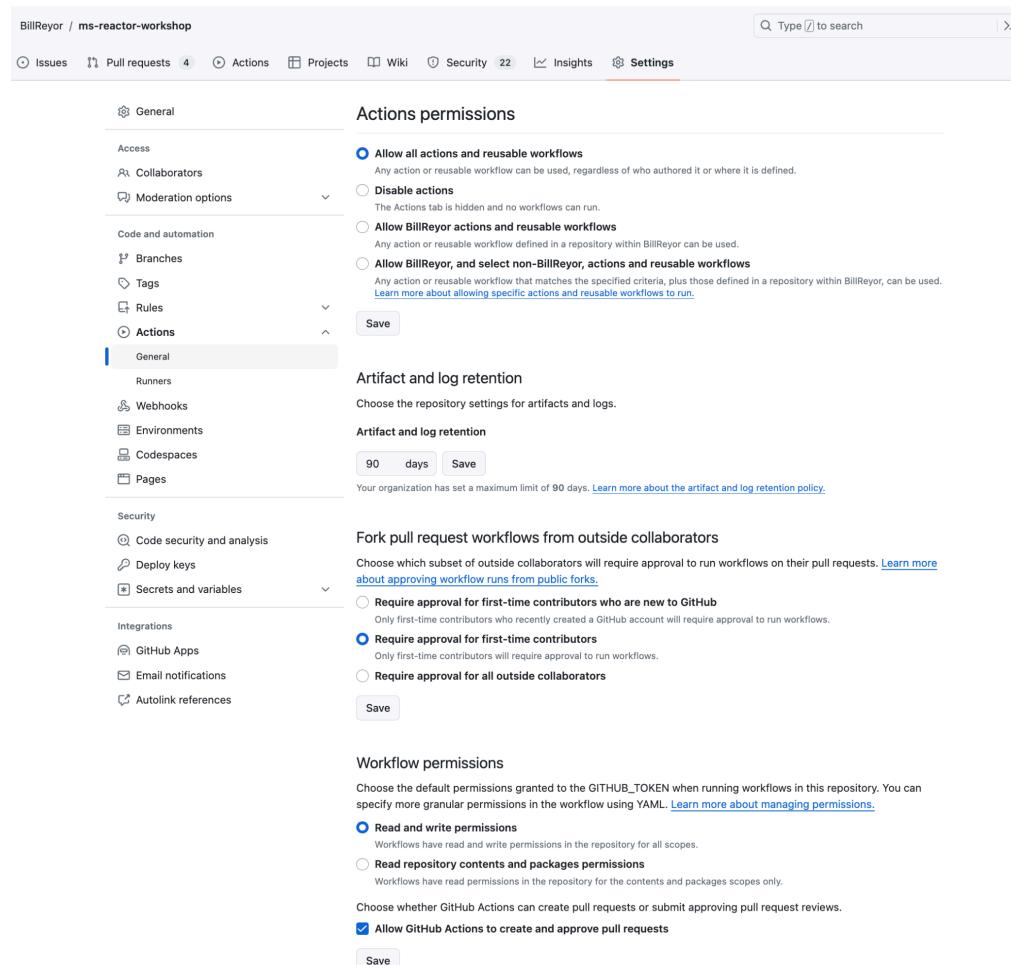
About this lab

This Microsoft Reactor Lab Guide, designed by Modus Create, is designed to be presented by an instructor who explains the concepts and demonstrates each step live. After each demonstration, participants are encouraged to replicate the actions, using the provided screenshots and detailed instructions as a guide. This method ensures that attendees not only understand the theoretical aspects of the tools and practices being taught but also gain practical experience by applying what they've learned in real time.

About this lab.....	2
Setup.....	3
Module 1 - Dependabot.....	4
Module 2 - CodeQL.....	8
Module 3 - AI-driven Secret Scanning.....	11
Module 4 - Setup Branch Protection Rules.....	15
Module 5 - Branch Protection Demonstration.....	17
Module 6 - Setup Copilot Security Lab.....	20
Module 7 - Copilot Security lab.....	22
Module 8 - Github Autofix Demo.....	24
Module 9 - Code-to-Cloud with GitHub and Azure.....	27

Setup

1. Fork the provided workshop repository from
<https://github.com/tweag/ms-reactor-workshop>
2. Check Workflow Permissions (See Figure 0)
 - a. In your repository, go to "Settings."
 - b. Click "Actions" > “General” in the left sidebar.
 - c. Under “Actions Permissions” Ensure that "Allow all actions and reusable workflows" is enabled.
 - d. Under "Workflow permissions," make sure "Read and write permissions" are selected.
3. If you receive the message “GitHub Actions is disabled on this repository because it is a fork. To use code scanning, please enable it”,
 - a. Click “enable it” > I understand my workflows, go ahead and enable them.



(Figure 0 - Workflow configuration)

Module 1 - Dependabot

1. Enabling Dependabot Alerts
 - Go to the forked repository's "Settings" page.
 - Click on "Code Security & Analysis."
 - Locate the "Dependabot alerts" option and click "Enable."
 - This will notify you of vulnerabilities in your dependencies.
2. Enable Dependabot Security Updates (See Figure 1)
 - Stay on the "Code Security & Analysis" page.
 - Find the "Dependabot security updates" option and click "Enable."
 - Dependabot will attempt to automatically create pull requests to fix detected vulnerabilities with available patches.
3. Navigate to the Pull Requests tab and evaluate the results. (See Figure 2)
 - Open Any finding
 - Navigate to the "Files Changed" tab (See Figure 3)
 - Navigate back to the bottom of the conversation tab & Merge the Pull Request (PR) > Confirm Merge (See Figure 4)
4. Review Dependabot alerts
 - In the workshops forked repository, navigate to Security -> Dependabot (See Figure 5)

Summary

This module explored activating Dependabot alerts and updates in GitHub repositories. By enabling these features, your teams are promptly notified about vulnerabilities in project dependencies. This is crucial as it allows for quick identification and resolution of security issues, reducing the risk of exploits in software projects.

BillReyor / ms-reactor-workshop

Issues Pull requests Actions Projects Wiki Security 6 Insights Settings

Code security and analysis

General Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions Webhooks Environments Codespaces Pages

Security

Code security and analysis

Deploy keys Secrets and variables

Integrations GitHub Apps Email notifications Autolink references

Code security and analysis

Dependabot alerts Dependabot security updates Grouped security updates Dependabot version updates Dependabot on Actions runners

Enable

Disable

Enable

Enable

Enable

Enable

Enable

(Figure 1 - Enabling Dependabot)

BillReyor / ms-reactor-workshop

Code Issues Pull requests 4 Actions Projects Wiki Security 26 Insights Settings

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Dismiss

Filters is:pr is:open Labels 10 Milestones 0 New pull request

4 Open 0 Closed

- Bump marked from 0.3.6 to 4.0.10 in /workshop/VulnerableAppTwo dependencies #4 opened 5 minutes ago by dependabot bot
- Bump mongoose from 5.0.16 to 5.13.20 in /workshop/VulnerableAppTwo dependencies #3 opened 5 minutes ago by dependabot bot
- Bump express from 4.16.0 to 4.19.2 in /workshop/VulnerableAppTwo dependencies #2 opened 5 minutes ago by dependabot bot
- Bump lodash from 4.17.10 to 4.17.21 in /workshop/VulnerableAppTwo dependencies #1 opened 5 minutes ago by dependabot bot

(Figure 2 - Dependabot PR results)

Bump marked from 0.3.6 to 4.0.10 in /workshop/VulnerableAppTwo #4

Merging this pull request will resolve 4 Dependabot alerts on marked including a high severity alert.

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾

0 / 1 files viewed Review in codespace Review changes

`@@ -9,7 +9,7 @@`

```

9   "dependencies": {
10     "express": "4.16.0", // Known to have vulnerabilities in this version
11     "todash": "4.17.10", // Vulnerable version
12 -   "marked": "0.3.6", // Vulnerable version
13 +   "marked": "4.0.10", // Vulnerable version
14     "mongoose": "5.0.16", // Known vulnerabilities in this version
15     "request": "2.81.0" // Deprecated and has known vulnerabilities
16   },

```

+1 -1 0

(Figure 3 - Dependabot PR results)

Bump marked from 0.3.6 to 4.0.10 in /workshop/VulnerableAppTwo #4

Merging this pull request will resolve 4 Dependabot alerts on marked including a high severity alert.

Conversation 0 Commits 1 Checks 0 Files changed 1

dependabot bot commented on behalf of github 15 minutes ago

- Bumps [marked](#) from 0.3.6 to 4.0.10.
- ▶ Release notes
- ▶ Commits
- ▶ Maintainer changes

compatibility unknown

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting @dependabot rebase .

▶ Dependabot commands and options

Bump marked from 0.3.6 to 4.0.10 in /workshop/VulnerableAppTwo Verified d9c1268

dependabot bot added the [dependencies](#) label 15 minutes ago

BillReyor approved these changes now View reviewed changes

Add more commits by pushing to the [dependabot/npm_and_yarn/workshop/VulnerableAppTwo/marked-4.0.10](#) branch on [BillReyor/ms-reactor-workshop](#).

Changes approved 1 approving review [Learn more about pull request reviews](#).

1 approval

Require approval from specific reviewers before merging Rulesets ensure specific people approve pull requests before they're merged. Add rule

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers BillReyor Still in progress? Convert to draft

Assignees No one—assign yourself

Labels [dependencies](#)

Projects None yet

Milestone No milestone

Development Successfully merging this pull request may close these issues.
None yet

Notifications Customize [Unsubscribe](#)
You're receiving notifications because you're watching this repository.

1 participant BillReyor

Lock conversation

(Figure 4 - Dependabot PR Merge)

Dependabot alerts

[Configure ▾](#)

Auto-triage your alerts (Beta)

Control how Dependabot opens pull requests, ignores false positives and snoozes alerts. Rules can be enforced at the organization level. Free for open source and available for private repos through [GitHub Advanced Security](#).

[Learn more about auto-triage](#)

Q is:open

□ **16 Open** ✓ 4 Closed

Package ▾ Ecosystem ▾ Manifest ▾ Severity ▾ Sort ▾

Alert Details	Actions
Mongoose Vulnerable to Prototype Pollution in Schema Object (Critical) #20 opened 28 minutes ago • Detected in mongoose (npm) • workshop/VulnerableAppTwo/package.json	Generate fix for custom pattern (Green)
Mongoose Prototype Pollution vulnerability (Critical) #17 opened 28 minutes ago • Detected in mongoose (npm) • workshop/VulnerableAppTwo/package.json	Generate fix for ecosystem (Purple)
Improper Input Validation in Automattic Mongoose (Critical) #14 opened 28 minutes ago • Detected in mongoose (npm) • workshop/VulnerableAppTwo/package.json	Ignore for manifest (Red)
Prototype Pollution in lodash (Critical) #8 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	Open for patch (Green)
Remote code injection in Log4j (Critical) #2 opened 28 minutes ago • Detected in org.apache.logging.log4j:log4j-core (Maven) • workshop/VulnerableApp/pom.xml	
Incomplete fix for Apache Log4j vulnerability (Critical) #1 opened 29 minutes ago • Detected in org.apache.logging.log4j:log4j-core (Maven) • workshop/VulnerableApp/pom.xml	
Prototype Pollution in lodash (High) #18 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	
automattic/mongoose vulnerable to Prototype pollution via Schema.path (High) #15 opened 28 minutes ago • Detected in mongoose (npm) • workshop/VulnerableAppTwo/package.json	
Command Injection in lodash (High) #11 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	
Prototype Pollution in lodash (High) #7 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	
Apache Log4j2 vulnerable to Improper Input Validation and Uncontrolled Recursion (High) #3 opened 28 minutes ago • Detected in org.apache.logging.log4j:log4j-core (Maven) • workshop/VulnerableApp/pom.xml	
Express.js Open Redirect in malformed URLs (Moderate) #19 opened 28 minutes ago • Detected in express (npm) • workshop/VulnerableAppTwo/package.json	
Server-Side Request Forgery in Request (Moderate) #16 opened 28 minutes ago • Detected in request (npm) • workshop/VulnerableAppTwo/package.json	
Regular Expression Denial of Service (ReDoS) in lodash (Moderate) #10 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	
Regular Expression Denial of Service (ReDoS) in lodash (Moderate) #9 opened 28 minutes ago • Detected in lodash (npm) • workshop/VulnerableAppTwo/package.json	
Improper Input Validation and Injection in Apache Log4j2 (Moderate)	

(Figure 5 - Dependabot Alerts)

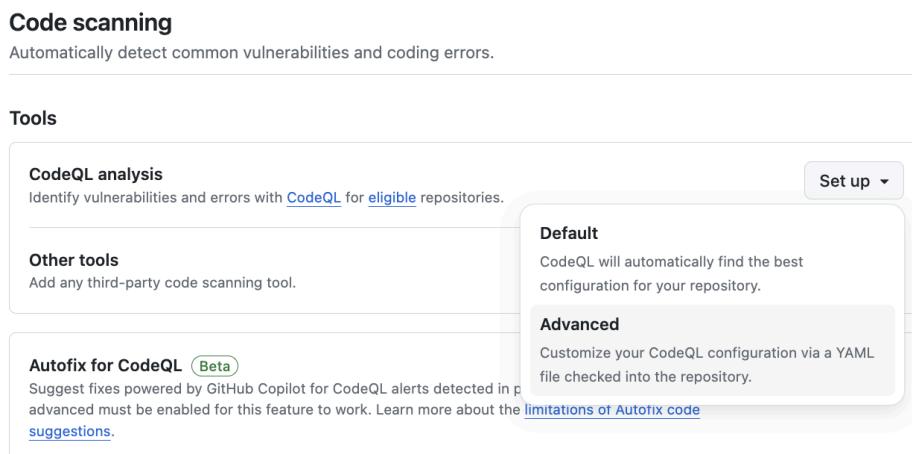
Module 2 - CodeQL

Setting Up CodeQL

- Navigate back to your repository's Settings > "Code Security and Analysis"
- Under the Code Scanning section CodeQL Analysis > Set up > Advanced > (See Figure 6)
- From CodeQL YML > Commit Changes > Commit Changes (See Figures 7)
- You will now see under Actions the CodeQL scan workflow has been triggered, let's give it a minute to finish.
- After the analysis is completed, navigate to security -> Code Scanning, where you should see CodeQL results. (Figure 8)

Summary

This module focused on setting up CodeQL for security analysis in GitHub repositories. By enabling GitHub Advanced Security and configuring CodeQL, your teams can automatically perform deep security checks on their codebases, helping to maintain the integrity and security of software projects.



(Figure 6 - CodeQL Setup)

The screenshot shows the GitHub Actions CodeQL workflow editor. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a search bar with the query "ms-reactor-workshop/.github/workflows/" and a dropdown showing "in main". To the right of the search bar are buttons for "Cancel changes" and "Commit changes...". The main area contains an "Edit" button and a "Preview" button. The code editor displays a YAML configuration file:

```
1 # For most projects, this workflow file will not need changing; you simply need
2 # to commit it to your repository.
3 #
4 # You may wish to alter this file to override the set of languages analyzed,
5 # or to provide custom queries or build logic.
6 #
7 # ***** NOTE *****
8 # We have attempted to detect the languages in your repository. Please check
9 # the `language` matrix defined below to confirm you have the correct set of
10 # supported CodeQL languages.
11 #
12 name: "CodeQL"
13
14 on:
15   push:
16     branches: [ "main" ]
17   pull_request:
18     branches: [ "main" ]
19   schedule:
20     - cron: '43 23 * * 6'
```

Below the code editor are buttons for "Spaces", "2", "No wrap", and a copy icon. To the right, there's a sidebar titled "Marketplace" with a search bar and a "Documentation" link. The sidebar lists featured actions:

- Cache** By actions ★ 4.3k: Cache artifacts like dependencies and build outputs to improve workflow execution time.
- Setup Node.js environment** By actions ★ 3.6k: Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH.
- Setup Java JDK** By actions ★ 1.4k: Set up a specific version of the Java JDK and add the command-line tools to the PATH.

(Figure 7 - Commit the Default CodeQL workflow to main)

A screenshot of a 'Commit changes' dialog box. At the top, it says 'Commit changes' and has a close button 'X'. Below that is a 'Commit message' field containing 'Create codeql.yml'. Underneath is an 'Extended description' field with the placeholder 'Add an optional extended description..'. At the bottom, there are two radio button options: 'Commit directly to the main branch' (selected) and 'Create a new branch for this commit and start a pull request'. A link 'Learn more about pull requests' is also present. At the very bottom are 'Cancel' and 'Commit changes' buttons, with 'Commit changes' being highlighted by a red box.

(Commit the Default CodeQL workflow to main)

The screenshot shows the GitHub Code scanning interface for the repository 'ms-reactor-workshop' owned by 'BillReyor'. The sidebar on the left includes links for Overview, Reporting, Policy, Advisories, Vulnerability alerts, Dependabot, Code scanning (which is selected), and Secret scanning. The main area is titled 'Code scanning' and displays a summary: 'All tools are working as expected' with 6 open issues. A search bar shows the query 'is:open branch:main'. Below this, a table lists six vulnerabilities:

Issue Type	Description	Severity	Branch
Uncontrolled command line	#4 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableApp.java:31	Critical	main
Uncontrolled data used in path expression	#6 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableApp.java:40	High	main
Query built from user-controlled sources	#5 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableApp.java:19	High	main
Missing rate limiting	#3 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableAppTwo.js:30	High	main
Database query built from user-controlled sources	#2 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableAppTwo.js:33	High	main
Reflected cross-site scripting	#1 opened 4 minutes ago - Detected by CodeQL in workshop/.../src/VulnerableAppTwo.js:46	High	main

A note at the bottom says: 'ProTip! You can run CodeQL locally using Visual Studio Code. [Learn more](#)'.

(Figure 8 - Code Scanning Results)

Module 3 - AI-driven Secret Scanning

Secret scanning is enabled by default for all public repositories on GitHub. However, those holding an enterprise license have access to enhanced features. For most attendees, simply follow along with the demonstration and slides for this segment, as the steps require a license that may not be available to everyone.

1. Navigate to Settings > Code Security and Analysis to configure secret scanning options.
2. Select the New Pattern button to begin creating a custom pattern.
3. Use the following regex to test for the presence of a GitHub Personal Access Token (PAT)
`(ghp|gho|ghu|ghs|ghr)_[a-zA-Z0-9]{36}`
4. Test this regex by pasting a dummy example key from vuln.ini and perform a dry run to ensure it works as expected.
5. After testing, click the Publish pattern button to save and activate your custom secrets detection pattern.
6. Let's create one more new pattern to highlight how AI can help us write this regex
 - a. Go back to the secret scanning options
 - b. Click - Generate with AI
 - c. For a prompt enter "Find instances that match an SSN" and provide an **fake** ssn example like nnn-nn-nnnn
 - d. Save and perform a dry run

Reviewing Detected Secrets

1. To view the secrets detected by your configurations, navigate to Security > Secret scanning on your GitHub repository settings. Here, you can review and manage any findings.

Summary

In this module, we focused on AI-driven secret scanning, which is automatically enabled for public repositories. Enterprise users enjoy enhanced capabilities. We explored how to configure custom secret scanning patterns, using regular expressions to detect GitHub Personal Access Tokens. This helps secure your code by identifying sensitive information inadvertently committed to your repositories.

Secret scanning alerts / #1

PAT detection rule

Open GitHub detected a [custom pattern](#) 15 seconds ago

Possibly active secret

```
ghp_abcdefghijklmnopqrstuvwxyzABCD012345
```

Remediation steps

Follow the steps below before you close this alert.

- 1 Rotate the secret if it's in use to prevent breaking workflows.
- 2 Revoke this PAT detection rule through the provider to prevent unauthorized access.
- 3 Check security logs for potential breaches.
- 4 Close the alert as revoked.

Detected in 2 locations

workshop/VulnerableAppTwo/vuln.ini

```
42
43 [third_party_services]
44 ; Example PAT token
45 ghp_pat_token="ghp_abcdefghijklmnopqrstuvwxyzABCD012345"
```

Added PAT token example 982b7b4 8 minutes ago

workshop/README.md

```
60
61 for example:
62
63 ghp_abcdefghijklmnopqrstuvwxyzABCD012345
64 ...
65
66 Lets' test this regex out. You can do this by pasting in our dummy example key from `vuln.ini`, and creating a dry run.
```

Added PAT token example 982b7b4 17 minutes ago

(Secret Scanning - Figure 9 - Pat Detection)

Security & analysis / New custom pattern

Generate with AI

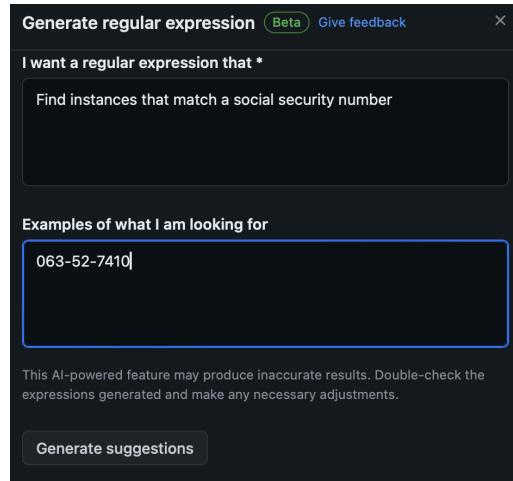
Pattern name *

This cannot be edited after saving.

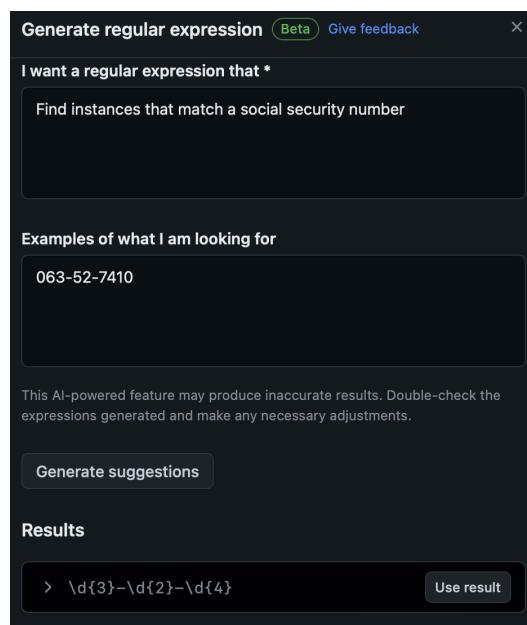
Secret format (specified as a regular expression) *

The pattern for the secret, specified as a regular expression. [Learn more about defining custom patterns.](#)

(Secret Scanning - With GenAI)



(Secret Scanning - With GenAI for an SSN)



(Secret Scanning - With GenAI created regex)

Security & analysis / New custom pattern Generate with AI

Pattern name *
findssn
This cannot be edited after saving.

Secret format (specified as a regular expression) *
\d{3}-\d{2}-\d{4}
The pattern for the secret, specified as a regular expression. [Learn more about defining custom patterns.](#)

> More options

Test string * - 1 match
063-52-7418

Provide a sample test string to make sure your configuration matches the patterns you expect.

Save and dry run

(Secret Scanning - With GenAI created regex save and dry run)

<input type="checkbox"/>		11 Open	<input checked="" type="checkbox"/>	0 Closed	Bypassed	Validity	Secret type	Provider	Sort
<input type="checkbox"/>		findssn 987-65-4321			#11 opened 19 seconds ago	• Detected custom pattern in workshop/VulnerableAppTwo/fakeSecrets.js:3			
<input type="checkbox"/>		findssn 123-45-6789			#10 opened 19 seconds ago	• Detected custom pattern in workshop/VulnerableAppTwo/fakeSecrets.js:2			
<input type="checkbox"/>		findssn 963-52-7410			#9 opened 19 seconds ago	• Detected custom pattern in workshop/VulnerableAppTwo/fakeSecrets.js:11			
<input type="checkbox"/>		findssn 852-74-1963			#8 opened 19 seconds ago	• Detected custom pattern in workshop/VulnerableAppTwo/fakeSecrets.js:10			

(Secret Scanning - With GenAI created regex results)

Module 4 - Setup Branch Protection Rules

Cool, we detected the code problems, but how do we prevent them in the first place?

1. Navigate to Branch Protection Settings:

- From your forked repository, click the "Settings" tab.
- In the left sidebar, select "Branches."
- In the "Branch protection rules" section, click "Add Branch Protection Rule" to create a new rule or edit an existing one for the main branch.

2. Require Passing Status Checks

- Enter main as the branch pattern to which the rule should apply.
- Check "Require a pull request before merging"
- Check "Require status checks to pass before merging."
- Within the "Search for status checks in the last week for this repository" section, select the following checks:
 - Search for and select "CodeQL"
 - Now scroll to the bottom of the page and Save by clicking the "Create" or "Save changes" button.
 - See Figure 10 - Branch Protection Rules for illustration.
- Navigate back to Settings > Branches > Edit our "main" branch protection rule set > Validate that "Require approvals" is unchecked (for our example only) > Save Changes.

The screenshot shows the GitHub settings interface for a repository named 'ms-reactor-workshop'. The left sidebar contains navigation links for General, Access, Code and automation, Security, and Integrations. Under 'Code and automation', the 'Branches' link is selected. The main content area is titled 'Branch protection rule' and includes a form for defining a branch name pattern ('main'), which applies to one branch. The 'Protect matching branches' section contains several configuration options, all of which are currently checked:

- Require a pull request before merging**: When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.
- Require approvals**: When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.
- Dismiss stale pull request approvals when new commits are pushed**: New reviewable commits pushed to a matching branch will dismiss pull request review approvals.
- Require review from Code Owners**: Requires an approved review in pull requests including files with a designated code owner.
- Require approval of the most recent reviewable push**: Whether the most recent reviewable push must be approved by someone other than the person who pushed it.
- Require status checks to pass before merging**: Choose which **status checks** must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.
- Require branches to be up to date before merging**: This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

Below these options is a search bar for status checks and a list of required status checks, which currently includes 'CodeQL'. There is also a 'GitHub Advanced Security' dropdown menu.

(Figure 10 - Branch Protection Rule)

Module 5 - Branch Protection Demonstration

1. To Test the Branch Protection Rule

- Navigate to the Code tab and edit the file directly via the editor at GitHub.com /workshop/VulnerableApp/src/VulnerableApp.java
- We will now add an insecure deserialization vulnerability by appending the code below to the end of our project between lines 50 and 51.
- Commit the change (*Figure 12*)
 - Select the option to create a new branch for this commit and start a pull request.
 - Click “Propose the change”
 - Wait for status checks to finish
- If everything has gone to plan, our changes are blocked because they failed the CodeQL scan (See Figure 13)

```
// Insecure Deserialization Vulnerability
try {
    // Simulating deserialization of an untrusted object
    String serializedObject = request.getParameter("serializedData");
    byte[] data = serializedObject.getBytes("ISO-8859-1");
    ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(data));
    Object deserializedObject = ois.readObject(); // This line is the vulnerable spot
    ois.close();

    response.getWriter().println("Deserialized Object: " + deserializedObject.toString());
} catch (Exception e) {
    e.printStackTrace();
}
```

Summary

In this module, we tested branch protection rules by deliberately introducing an insecure deserialization vulnerability into our project's codebase. This exercise involved appending specific vulnerable code to **VulnerableApp.java** and submitting a pull request. If the branch protection rules are set correctly, introducing such vulnerabilities triggers a failure in the CodeQL scan, effectively blocking the merge of potentially harmful changes.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ms-reactor-workshop / workshop / VulnerableApp / src / VulnerableApp.java in main

Cancel changes Commit changes...

Edit Preview

Spaces 4 No wrap

```

5 import java.sql.Statement;
6 import javax.servlet.*;
7 import javax.servlet.http.*;
8
9 public class VulnerableApp extends HttpServlet {
10
11     public void doGet(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException {
13         // Vulnerability 1: SQL Injection
14         try {
15             String user = request.getParameter("user");
16             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test?user=mysql&password=mysqlpassword");
17             Statement stmt = conn.createStatement();
18             // Unsafe query construction
19             ResultSet rs = stmt.executeQuery("SELECT * FROM users WHERE username = '" + user + "'");
20             while (rs.next()) {
21                 response.getWriter().println("User found: " + rs.getString("username"));
22             }
23         } catch (Exception e) {
24             e.printStackTrace();
25         }
26
27         // Vulnerability 2: Command Injection
28         try {
29             String data = request.getParameter("data");
30             // Unsafe command execution
31             Runtime.getRuntime().exec("echo " + data);
32         } catch (IOException e) {
33             e.printStackTrace();
34         }
35
36         // Vulnerability 3: Path Traversal
37         try {
38             String filePath = request.getParameter("filePath");
39             // Unsafe file access
40             FileInputStream fis = new FileInputStream("/var/www/data/" + filePath);
41             int ch;
42             PrintWriter pw = response.getWriter();
43             while ((ch = fis.read()) != -1) {
44                 pw.print((char)ch);
45             }
46         } catch (FileNotFoundException e) {
47             e.printStackTrace();
48         } catch (IOException e) {
49             e.printStackTrace();
50         }
51         // Insecure Deserialization Vulnerability
52         try {
53             // Simulating deserialization of an untrusted object
54             String serializedObject = request.getParameter("serializedData");
55             byte[] data = serializedObject.getBytes("ISO-8859-1");
56             ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(data));
57             Object deserializedObject = ois.readObject(); // This line is the vulnerable spot
58             ois.close();
59
60             response.getWriter().println("Deserialized Object: " + deserializedObject.toString());
61         } catch (Exception e) {
62             e.printStackTrace();
63         }
64     }
65 }
66 
```

(Figure 11 - Bad code in PR)

Propose changes

Commit message
Update VulnerableApp.java

Extended description
Add an optional extended description..

Commit directly to the main branch
 Create a new branch for this commit and start a pull request
[Learn more about pull requests](#)

BillReyor-patch-6

Cancel Propose changes

(Figure 12 - Create a new branch for the commit and start a Pull Request)

Update VulnerableApp.java #11

[Open](#) BillReyor wants to merge 1 commit into `main` from `BillReyor-patch-7`

Conversation 1 · Commits 1 · Checks 3 · Files changed 1

BillReyor commented 4 minutes ago

No description provided.

Update VulnerableApp.java

github-advanced-security (bot) found potential problems 3 minutes ago

View reviewed changes

workshop/VulnerableApp/src/VulnerableApp.java

```
54 +     String serializedObject = request.getParameter("serializedData");
55 +     byte[] data = serializedObject.getBytes("ISO-8859-1");
56 +     ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(data));
57 +     Object deserializedObject = ois.readObject(); // This line is the vulnerable spot
```

Check failure

Code scanning / CodeQL

Deserialization of user-controlled data Critical

Unsafe deserialization depends on a user-provided value.

Show more details

Show paths Dismiss alert ▾

Reply...

Add more commits by pushing to the [BillReyor-patch-7](#) branch on [BillReyor/ms-reactor-workshop](#).

[Some checks were not successful](#) Hide all checks

2 successful and 1 failing checks

- ✓ [CodeQL / Analyze \(java-kotlin\) \(pull_request\)](#) Successful in 1m
- ✓ [CodeQL / Analyze \(javascript-typescript\) \(pull_request\)](#) Successful in 1m
- ✗ [Code scanning results / CodeQL](#) Failing after 3s — 1 new alert including 1 critical severity security... Required

[Required statuses must pass before merging](#)

All required [statuses](#) and check runs on this pull request must run successfully to enable automatic merging.

[Merge without waiting for requirements to be met \(bypass branch protections\)](#)

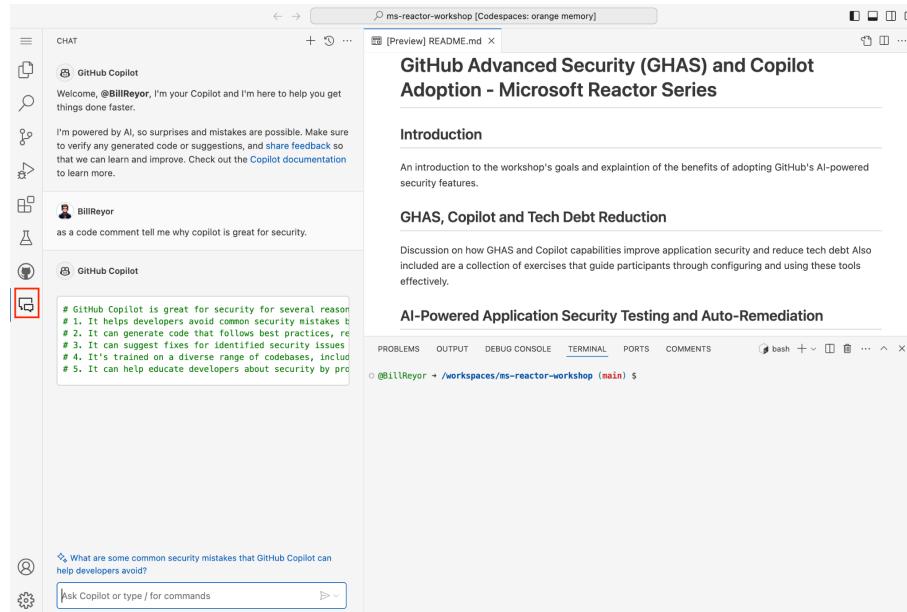
Merge pull request ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add a comment

(Figure 13 - CodeQL checks fail, and insecure code is blocked from being merged)

Module 6 - Setup Copilot Security Lab

1. Accessing the Codespace
 - Navigate to the forked repository on GitHub.
 - Click the "Code" button and select tab "Codespaces" > "Create codespace on main".
 - Choose the appropriate development container configuration to initiate the environment if not autodetected.
2. Verifying GitHub Copilot Activation (Figure 14)
 - Wait about 2 minutes for the container to deploy
 - Once the codespace is loaded, verify that GitHub Copilot is enabled.
 - You will see on the left pane an icon that looks like a chat window, this is CoPilot chat > Click on the Chat button and validate Copilot chat responds by providing any example question initiating a chat.
 - If Copilot is inactive, go to the Extensions view, install if necessary, and ensure the GitHub Copilot extension is enabled.
 - **Remember**, you will need a Copilot license for this workshop portion.
3. From our code space terminal, enter the following in the web terminal (figure 15&16)
 - cd /workspaces/ms-reactor-workshop/workshop/VulnerableAppThree
 - Npm install
 - Npm start
 - When prompted, click "open in browser" > Once validated running "Hello!" we can close this tab and return to our CodeSpaces workshop tab.



(Figure 14 - Validating Copilot is working in our codespace)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 COMMENTS
Updating 8d745dc..6f71184
Fast-forward
workshop/VulnerableAppThree/package.json | 16 ++++++=====
workshop/VulnerableAppThree/vulnerable-server.js | 33 ++++++=====+
2 files changed, 49 insertions(+)
create mode 100644 workshop/VulnerableAppThree/vulnerable-server.js
@billReyor ~ /workspaces/ms-reactor-workshop (main) $ ^C
@billReyor ~ /workspaces/ms-reactor-workshop (main) $ cd workshop/VulnerableAppThree
@billReyor ~ /workspaces/ms-reactor-workshop/workshop/VulnerableAppThree (main) $ npm install
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs

added 189 packages, and audited 190 packages in 11s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
@billReyor ~ /workspaces/ms-reactor-workshop/workshop/VulnerableAppThree (main) $ npm start

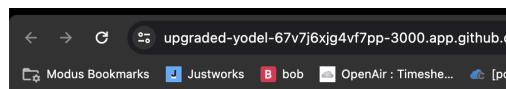
> vulnerable-node-app@1.0.0 start
> node vulnerable-server.js

Server running on http://localhost:3000
  
```

Your application running on port 3000 is available. See all forwarded ports

[Open in Browser](#) [Make Public](#)

(Figure 15 - Validating VulnerableAppThree is functional in our codespace)



Hello!

(Figure 16 - Validating VulnerableAppThree is functional in our codespace)

Module 7 - Copilot Security lab

1. Identifying Vulnerabilities with Copilot (*Figure 17*)

- Now navigate to our Explorer icon on the left pane and open the file /workshop/VulnerableAppThree/vulnerable-server.js
- Select all the code's content, switch to the Copilot chat on the left pane, and prompt, “Does my code contain vulnerabilities?” We can see it has found several vulnerabilities and has provided more insight into the issue.
- Navigate back to the code editor window on the right. Within the code editor, find and highlight the line CoPilot chat mentioned earlier in the chat (“Here’s the problematic line”), then right-click and from the contextual menu select Copilot -> Fix this, and when prompted, simply enter “the sql injection vulnerability”
- Accept the suggestion
- As you type or modify code, observe GitHub Copilot’s suggestions, focusing on identifying insecure coding patterns.
- Copilot may offer alternative coding suggestions that enhance security.
- After you apply the fix, Ask Copilot Chat if the code still contains vulnerabilities. (Notice that Copilot identifies additional vulnerabilities..)

Summary

In this module, we demonstrated that Copilot can identify and provide useful suggestions to remediate vulnerabilities; however, prompting is key and we can’t expect Copilot to uniformly detect all code vulnerabilities.

```
app.get('/add-user', (req, res) => {
  const { username, email } = req.query;

  /fix the SQL injection vulnerability
```

/fix

GitHub Copilot used /fix
To fix the issue, I would add the missing import statement for the `express` module at the top of the code block.

BillReyor
/fix the SQL injection vulnerability

GitHub Copilot used /fix
To fix the SQL injection vulnerability, you need to use parameterized queries instead of directly concatenating user input into the SQL query string.

Accept Discard Accept or Discard 1 change.

```
const sql = "INSERT INTO users (username, email) VALUES (?, ?)";
db.run(sql, [username, email], function(err) {
```

(Figure 17 - Validating VulnerableAppThree is functional in our codespace)

Module 8 - Github Autofix Demo

Autofix (Currently Beta) is an expansion of code scanning that provides targeted recommendations to fix code scanning alerts found by CodeQL in pull requests. It's available to GitHub Enterprise Cloud users with GitHub Advanced Security.

In this demo, we'll introduce an uncontrolled command-line vulnerability by appending the following code snippet to the end of our application, defined at `vulnerable-server.js` within `/workshop/VulnerableAppThree` within our repository.

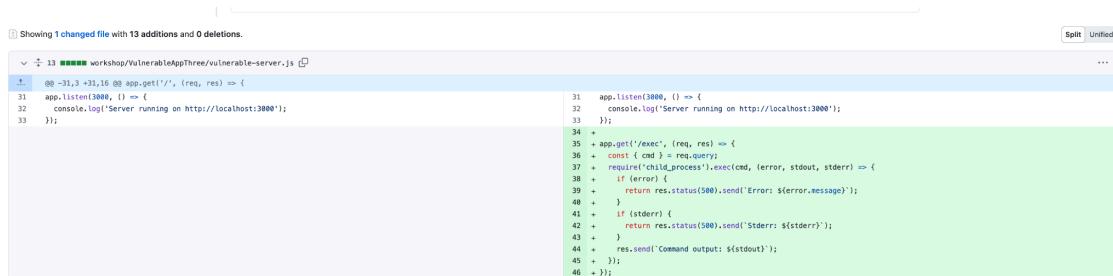
```
app.get('/exec', (req, res) => {
  const { cmd } = req.query;
  require('child_process').exec(cmd, (error, stdout, stderr) => {
    if (error) {
      return res.status(500).send(`Error: ${error.message}`);
    }
    if (stderr) {
      return res.status(500).send(`Stderr: ${stderr}`);
    }
    res.send(`Command output: ${stdout}`);
  });
});
```

Testing Github Autofix (Figures 18 & 19)

1. Go to your repository settings, navigate to the "Security & analysis" tab, and enable Autofix for CodeQL analysis.
2. From Github.com, Navigate to `/workshop/VulnerableAppThree/vulnerable-server.js` and edit the file in place.
3. Pasting in the vulnerable code snippet at the end of the file
4. Commit the changes to a new branch and start a pull request, then click "Create Pull request"
5. Wait for CodeQL scans to complete
6. Review and accept the Autofix suggestions by clicking "Accept suggestion" to automatically commit the fix to your pull request branch.
7. CodeQL will re-run and validate the revised PR (this may take multiple iterations as multiple vulnerabilities are introduced)
8. Merge the pull request into the main branch after all the checks pass, indicating that the vulnerability has been fixed.

Summary

In this module, we explored GitHub Autofix, an advanced feature that extends code scanning capabilities by offering actionable fixes directly in pull requests. Autofix helps rapidly fix code vulnerabilities to help your team ensure code is vulnerability-free while offering actionable suggestions for fixes to help keep your team productive and focused on creating.



```

  1  08-31-2 23:16 08 app.get('/', (req, res) => {
  2
  3     app.listen(3000, () => {
  4         console.log('Server running on http://localhost:3000');
  5     });
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31     app.listen(3000, () => {
 32         console.log('Server running on http://localhost:3000');
 33     });
 34
 35     + app.get('/exec', (req, res) => {
 36         +   const cmd = req.query.cmd;
 37         +   require('child_process').exec(cmd, (error, stdout, stderr) => {
 38             +     if (error) {
 39                 +       return res.status(500).send(`Error: ${error.message}`);
 40             }
 41             +           if (stderr) {
 42                 +               return res.status(500).send(`Stderr: ${stderr}`);
 43             }
 44             +               res.send(`Command output: ${stdout}`);
 45         });
 46     });

```

(Figure 18 - Code Commit highlighting our introduction of the vulnerable code)

Update vulnerable-server.js #7

Open BillReyor wants to merge 1 commit into `main` from `BillReyor-patch-3`

Conversation 0 Commits 1 Checks 0 Files changed 1

BillReyor commented 4 minutes ago

No description provided.

Update vulnerable-server.js

github-advanced-security (bot) found potential problems 2 minutes ago

[View reviewed changes](#)

`workshop/VulnerableAppThree/vulnerable-server.js`

```
... ... @@ -31,3 +31,15 @@
31   31   app.listen(3000, () => {
32   32     console.log('Server running on http://localhost:3000');
33   33   });
34 + app.get('/exec', (req, res) => {
35 +   const { cmd } = req.query;
36 +   require('child_process').exec(cmd, (error, stdout, stderr) => {

```

Check failure

Code scanning / CodeQL

Uncontrolled command line Critical

This command line depends on a **user-provided value**.

Show more details

Show paths Dismiss alert ▾

github-advanced-security (bot)

Give us feedback Beta

The problem with the code is that it directly uses user-provided input in a command that is executed by the server. This is a serious security vulnerability as it allows for command injection attacks, where a malicious user can execute arbitrary commands on the server.

The best way to fix this problem is to avoid executing shell commands with user-provided input altogether. If it's absolutely necessary to execute a command, we should use a safer method that doesn't spawn a shell, such as `child_process.execFile` or `child_process.execFileSync`, and provide the command arguments as an array rather than a single string.

In this case, we can replace the `child_process.exec` call on line 36 with a `child_process.execFile` call. We'll need to parse the `cmd` query parameter into a command and an array of arguments. We can use the `string.split` method to split the `cmd` string into an array of strings, using a space as the delimiter. The first element of the array will be the command, and the rest of the elements will be the arguments.

Suggested fixes for CodeQL alerts, powered by GitHub Copilot, may produce inaccurate results. Review carefully before use.

`workshop/VulnerableAppThree/vulnerable-server.js`

```
... ... @@ -35,3 +35,5 @@
35   35   const { cmd } = req.query;
36 -   require('child_process').exec(cmd, (error, stdout, stderr) => {
36 +   const args = cmd.split(' ');
37 +   const command = args.shift();
38 +   require('child_process').execFile(command, args, (error, stdout, stderr) => {
37   39     if (error) {

```

Dismiss Edit ▾ Commit fix ▾

Reply...

(Figure 19 - GitHub Autofix suggestions find and fix vulnerabilities)

Module 9 - Code-to-Cloud with GitHub and Azure

This demo will show how to use GitHub webhooks with Azure Function Apps to automate workflows and enhance project security monitoring.

1. Create a Function App

- Log into your Azure Portal.
- Search for and select 'Function App,' then click 'Create'
- Follow the prompts to configure your Function App (select a resource group, choose a unique name, select a runtime like Node.js, and ensure it's hosted on Windows).
- Click 'Review + Create' and then 'Create' once validation passes.

2. Develop Your Function

- Once your Function App is deployed, go to the resource, click Overview in the sidebar (left), and then click 'Create function'.
- Choose 'HTTP trigger', provide a name, and set the authorization level to 'Function'.
- Click 'Create' and wait for the deployment to complete.
- Navigate to 'Code + Test' in the function settings, enter a function that logs the incoming HTTP request data then save - We're replacing the default content in the function with our custom function from our forked repository See **logWebhookData.js** in the **Azure** directory in the workshop repository
- Click "Get function URL" and note the default (Function key)

3. Add Webhook to Repository

- Go back to your GitHub repository settings.
- Click 'Webhooks', then 'Add webhook'.
- For the 'Payload URL', enter the URL of the Azure Function you just created (you can get this from the 'Get function URL' button in your function's overview but remember you want the **Function key URL**).
- Choose '**application/json**' for the content type.
- Select which events you'd like to trigger this webhook, for our example, select "Send me everything" to receive all event types'.
- Click 'Add webhook'.

4. Trigger the Webhook

- Make a change in your repository, such as pushing a small commit.

- This action should trigger the webhook, sending data to your Azure Function. (There may be a 1-2 minute delay in ingestion)
- Check the invocations & logs in your Azure Function App to see if it received the webhook data. (We provide a KQL Query below to illustrate the power of this)
- Back on the left pane of your function app, select Monitoring > Logs. Here, we will run our demo query to see the results
- Ensure the data matches the actions you performed in GitHub.

Summary

In this module, we explored the integration of GitHub webhooks with Azure Function Apps, a powerful toolset that extends automation capabilities across cloud environments. This setup automatically triggers functions based on GitHub events, helping teams enhance security monitoring and manage workflows efficiently. The Azure Function App acts as a dynamic responder to GitHub events, logging each activity and ensuring that all project changes are tracked and secured in real time. We also highlighted how by importing log data into Azure it becomes queryable with KQL, providing teams with powerful insights into repository or organizational activity.

KQL Query to show all activity ordered by time

```

traces
| where message startswith "{" and message has "timestamp"
| extend logEntry = parse_json(message)
| project
    Timestamp = todatetime(logEntry.timestamp),
    EventType = tostring(logEntry.eventType),
    RepositoryName = tostring(logEntry.repositoryName),
    PusherName = tostring(logEntry.pusherName),
    CommitMessage = tostring(logEntry.commitMessage),
    IPAddress = tostring(logEntry.ipAddress),
    Branch = tostring(logEntry.branch),
    WorkflowName = tostring(logEntry.workflowName),
    WorkflowStatus = tostring(logEntry.workflowStatus),
    JobName = tostring(logEntry.jobName),
    JobStatus = tostring(logEntry.jobStatus),
    RawLog = tostring(logEntry.RAW_LOG)
| order by Timestamp desc

```

Home >

Function App ...

MOC HOL 100292 (mochol100292.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query | Assign tags Start Restart

Filter for any field... Subscription equals all X Resource group equals all X Location equals all X + Add filter

Showing 0 to 0 of 0 records.

Name ↑↓ Status ↑↓ Location ↑↓ Pricing Tier ↑↓ App Service Plan ↑↓



No function apps match your filters

Try changing or clearing your filters.

Create Function App

Clear filters

[Learn more about App Service](#)

(Code to Cloud - Figure 1 - Create function all)

☰ Microsoft Azure Search resources, services, and docs

[Home](#) > [Function App](#) >

Create Function App ...

Select a hosting option

These options determine how your app scales, resources available per instance, and pricing

Hosting plans	<input checked="" type="radio"/> Consumption Pay for compute resources when your functions are running (pay-as-you-go).
Scale to zero	
Scale behavior	Event-driven
Virtual networking	-
Dedicated compute and prevent cold start	-
Max scale out	200

Select

(Code to Cloud - Figure 2 - Select Consumption hosting plan)

Microsoft Azure Search resources, services, and docs (G+/)

Home > Function App > Create Function App >

Create Function App (Consumption) ...

Basics Storage Networking Monitoring Deployment Tags Review + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	MOC HOL 100292
Resource Group *	(New) function-webhooks-NYC-Reactor_group
	Create new

Instance Details

Enter a global unique name below
(append numbers as needed)

Function App name *	function-webhooks-NYC-Reactor.azurewebsites.net
---------------------	---

Runtime stack *

Node.js

Version *

20 LTS

Region *

East US

Operating System *

<input type="radio"/> Linux <input checked="" type="radio"/> Windows
--

Review + create [< Previous](#) [Next : Storage >](#)

(Code to Cloud - Figure 3 - Function app configuration)

Create Function App (Consumption)

...



Basic authentication for this app is currently disabled and may impact deployments. Click to learn more.

Details

Subscription	7fc993b4-a6fe-42a7-a5fa-e158b67fe74e
Resource Group	function-webhooks-NYC-Reactor_group
Name	function-webhooks-NYC-Reactor
Runtime stack	Node.js 20 LTS

Hosting

Storage (New)

Storage account	functionwebhooksnyc89c0
-----------------	-------------------------

Plan (New)

Hosting options and plans	Consumption
Name	ASP-functionwebhooksNYCReactorgroup-a430
Operating System	Windows
Region	East US
SKU	Dynamic

Monitoring (New)

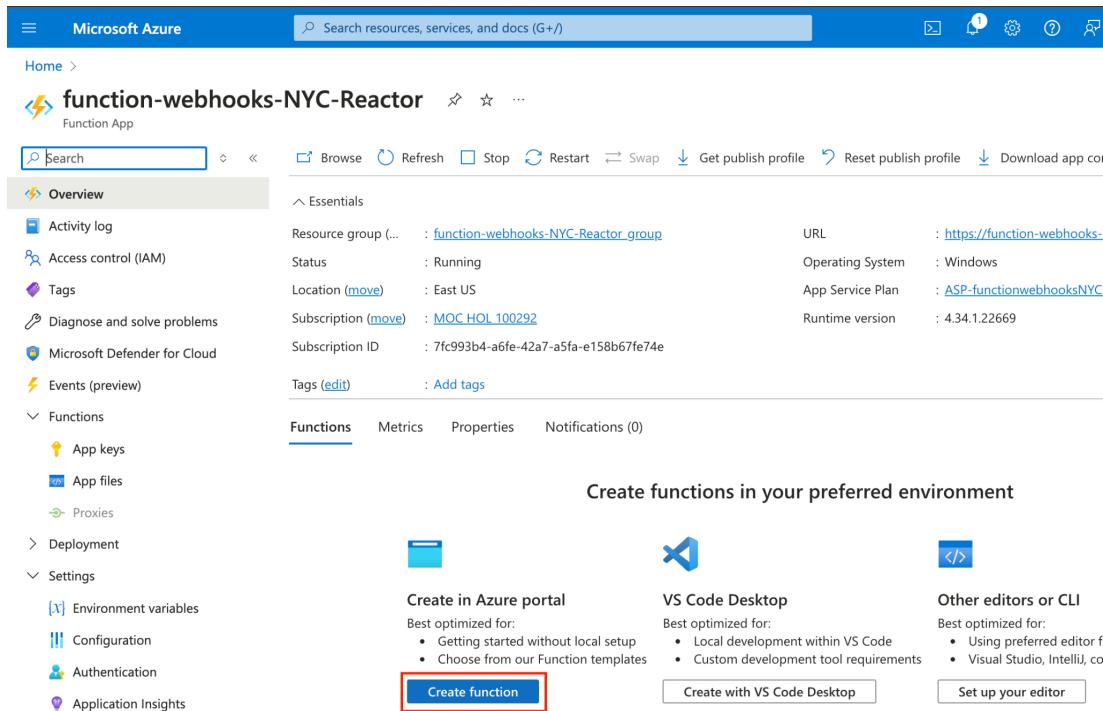
Application Insights	Enabled
Name	function-webhooks-NYC-Reactor
Region	East US

Deployment

Basic authentication	Disabled
Continuous deployment	Not enabled / Set up after app creation

[Create](#)[< Previous](#)[Next >](#)[Download a template for automation](#)

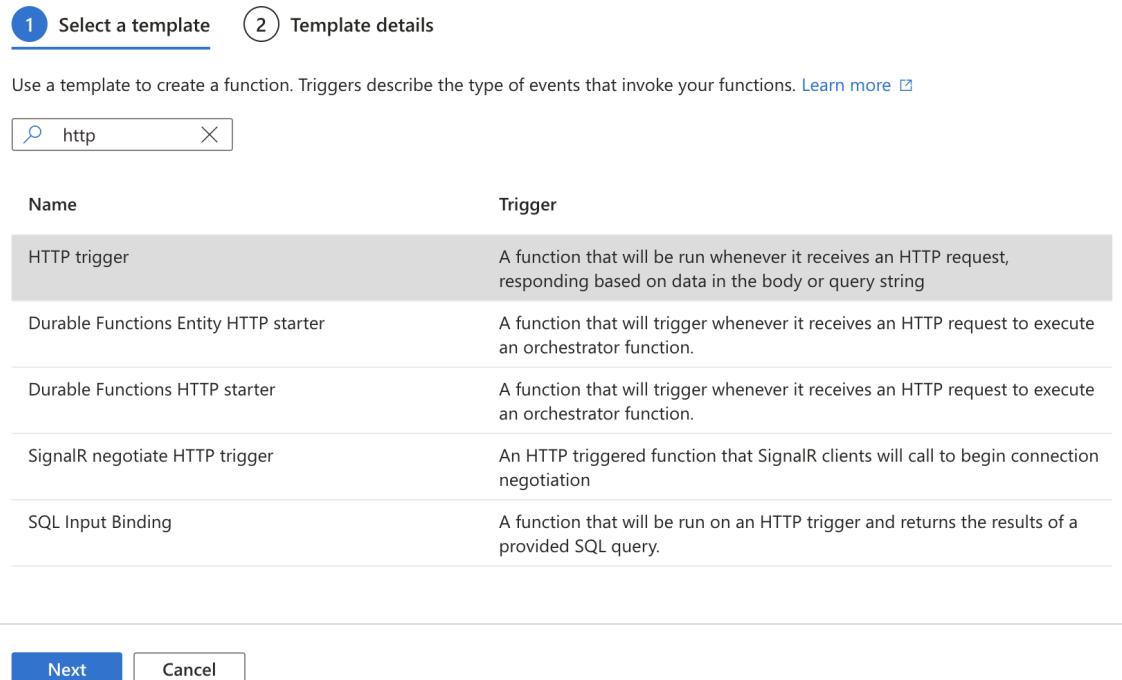
(Code to Cloud - Figure 4 - Function app create)



The screenshot shows the Microsoft Azure portal interface for a Function App named "function-webhooks-NYC-Reactor". The top navigation bar includes "Microsoft Azure", a search bar, and various icons for account management. The main content area displays the app's configuration under the "Overview" tab, including resource group, status, location, and subscription details. A sidebar on the left provides links for activity log, access control, tags, and other settings. At the bottom, there is a section titled "Create functions in your preferred environment" with three options: "Create in Azure portal", "VS Code Desktop", and "Other editors or CLI". The "Create in Azure portal" option has a "Create function" button, which is highlighted with a red box.

(Code to Cloud - Figure 5 - Function app Create function)

Create function



The screenshot shows the "Create function" wizard, step 1: "Select a template". It features a search bar with "http" typed in. Below the search bar is a table with two columns: "Name" and "Trigger". The "HTTP trigger" row is selected and highlighted with a gray background. Other rows include "Durable Functions Entity HTTP starter", "Durable Functions HTTP starter", "SignalR negotiate HTTP trigger", and "SQL Input Binding". At the bottom of the screen are "Next" and "Cancel" buttons.

Name	Trigger
HTTP trigger	A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string
Durable Functions Entity HTTP starter	A function that will trigger whenever it receives an HTTP request to execute an orchestrator function.
Durable Functions HTTP starter	A function that will trigger whenever it receives an HTTP request to execute an orchestrator function.
SignalR negotiate HTTP trigger	An HTTP triggered function that SignalR clients will call to begin connection negotiation
SQL Input Binding	A function that will be run on an HTTP trigger and returns the results of a provided SQL query.

(Code to Cloud - Figure 6 - Create a function from HTTP trigger)

Create function



- 1 Select a template 2 Template details

We need more information to create the function. [Learn more](#)

Function template

HTTP trigger

Function name *

HttpTrigger1

Authorization level * ⓘ

Function

[Create](#)

[Cancel](#)

(Code to Cloud - Figure 7 - Create a function from HTTP trigger - set auth & create)

Microsoft Azure

Home > function-webhooks-NYC-Reactor >

HttpTrigger1 | Code + Test

function-webhooks-NYC-Reactor

Code + Test 1
Integration Function Keys Invocations Logs Metrics

Save 2 Discard Refresh Test/Run Get function URL 3 Disable Delete Upload Send us your feedback

Paste code from repository

```

1 module.exports = async function (context, req) {
2   context.log('JavaScript HTTP trigger function processed a request.');
3
4   try {
5     // Extract specific data from the payload
6     const repositoryName = req.body.repository ? req.body.repository.name : 'Unknown';
7     const pusherName = req.body.sender ? req.body.sender.login : 'Unknown';
8     const ipAddress = req.headers['X-Forwarded-For'] || req.headers['x-forwarded-for'] || 'Unknown IP';
9     const eventType = req.headers['x-github-event'] || 'Unknown event type';
10
11     let commitMessage = 'Unknown';
12     let branch = '';
13     let workflowName = '';
14     let workflowStatus = '';
15     let jobName = '';
16     let jobStatus = '';
17
18     if (req.body.head_commit && req.body.head_commit.message) {
19       commitMessage = req.body.head_commit.message;
20     } else if (req.body.workflow_run && req.body.workflow_run.head_commit && req.body.workflow_run.head_commit.message) {
21       commitMessage = req.body.workflow_run.head_commit.message;
22       workflowName = req.body.workflow_run.name;
23       workflowStatus = req.body.workflow_run.status;
24     }

```

Logs App Insights Logs Log Level Stop Copy Clear M:

Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, please Function menu.

(Code to Cloud - Figure 8 - Create a function from HTTP trigger - set auth & create)

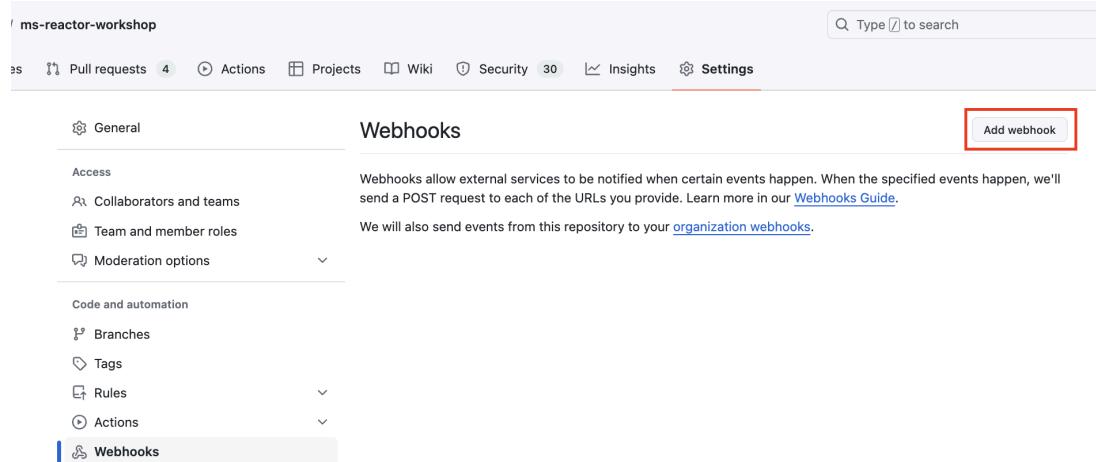
Get Function URL

master (Host key)
<https://function-webhooks-nyc-reactor.azurewebsites.net/api/HttpTrigger1?code=XUiF...>

default (Function key)
<https://function-webhooks-nyc-reactor.azurewebsites.net/api/HttpTrigger1?code=DS9...>

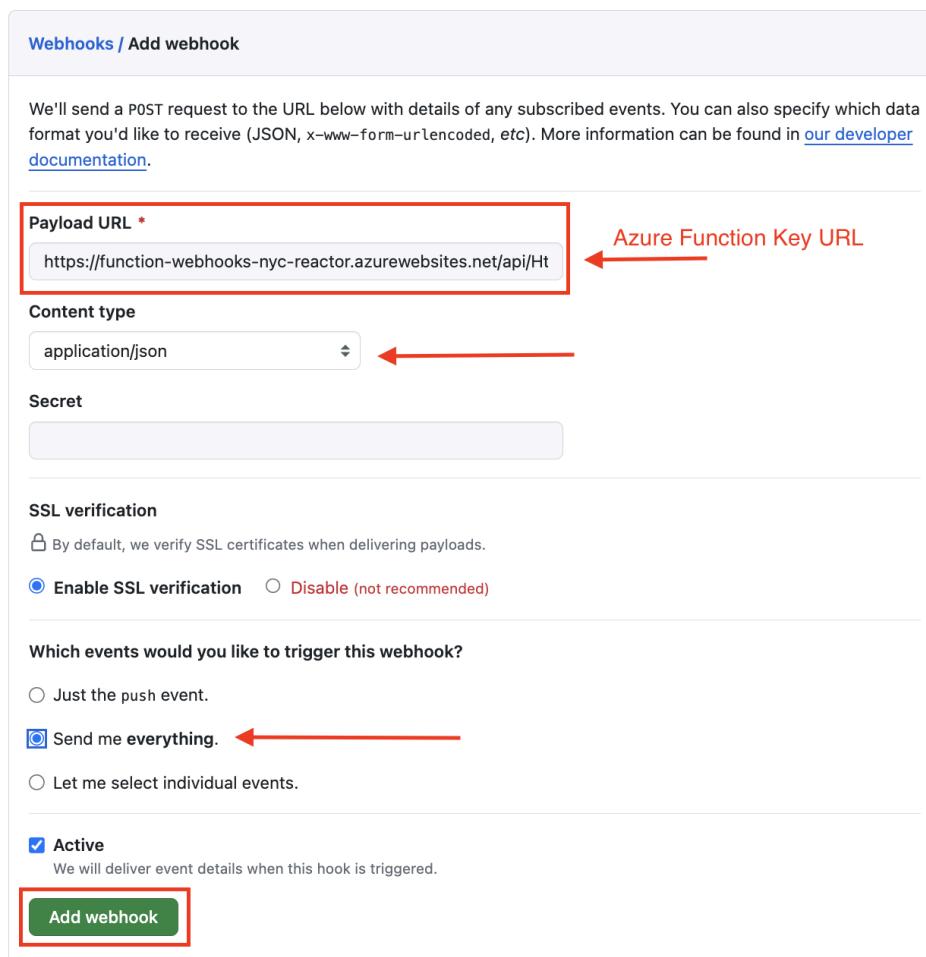
default (Host key)
<https://function-webhooks-nyc-reactor.azurewebsites.net/api/HttpTrigger1?code=3yJ-...>

(Code to Cloud - Figure 9 - Copy the **function key URL**)



The screenshot shows the GitHub repository settings for 'ms-reactor-workshop'. The 'Webhooks' tab is selected. A red box highlights the 'Add webhook' button in the top right corner. On the left sidebar, the 'Webhooks' option is also highlighted with a blue bar.

(Code to Cloud - Figure 10 - Add a webhook in GitHub)



The screenshot shows the 'Webhooks / Add webhook' configuration page. Several fields are highlighted with red boxes and arrows pointing to specific configuration details:

- Payload URL ***: The URL `https://function-webhooks-nyc-reactor.azurewebsites.net/api/H` is highlighted with a red box. An arrow points from this box to the text "Azure Function Key URL".
- Content type**: The dropdown menu showing "application/json" is highlighted with a red box. An arrow points from this box to the text "Azure Function Key URL".
- Secret**: A text input field for a secret key is shown.
- SSL verification**: The "Enable SSL verification" radio button is selected and highlighted with a red box. An arrow points from this box to the text "Azure Function Key URL".
- Which events would you like to trigger this webhook?**: The "Send me everything" radio button is selected and highlighted with a red box. An arrow points from this box to the text "Azure Function Key URL".
- Active**: The "Active" checkbox is checked and highlighted with a red box. An arrow points from this box to the text "Azure Function Key URL".

At the bottom, a green "Add webhook" button is highlighted with a red box.

(Code to Cloud - Figure 11 - Configure webhook in GitHub)

Commit changes

Commit message

Update README.md

Extended description

Add an optional extended description..

Commit directly to the main branch
Some rules will be bypassed by committing directly

Create a new branch for this commit and start a pull request
[Learn more about pull requests](#)

[Cancel](#) [Bypass rules and commit changes](#)

(Code to Cloud - Figure 12 - We make a minor edit to the repo readme and commit)

HttpTrigger1 | Invocations ...

function-webhooks-NYC-Reactor

Code + Test Integration Function Keys Invocations Logs Metrics

[Open in Application Insights](#) [Refresh](#) [Send us your feedback](#)

Query
Up to 20 of the most recent function invocation traces. For more advanced analysis, run the query in Application Insights.

Success count 11 Last 30 days	Error count 0 Last 30 days
--	---

[Search](#)

Date (UTC)	Status	Result Code	Duration (ms)	Operation ID
05/12/2024, 13:51:57.317	Success	200	7	5e204a64054c53b27fa18ff43a3ecfef
05/12/2024, 13:51:57.158	Success	200	10	368fc8fc499ddd825a7a7fc36c1a49ca

(Code to Cloud - Figure 13 - Invocations show successful triggering)

Home >

function-webhooks-NYC-Reactor

Function App

Search

Browse Refresh Stop Restart Swap Get publish profile Reset publish profile Download app content Delete ...

Backups Custom domains Certificates Networking Scale up (App Service plan) Scale out Service Connector Properties Locks

App Service plan Development Tools API Monitoring

Alerts Metrics Advisor recommendations Health check Logs

Essentials

Resource group (...): [function-webhooks-NYC-Reactor_group](#) URL: <https://function-webhooks-nyc-reactor.azurewebsites.net>
 Status: Running Operating System: Windows
 Location (move): East US App Service Plan: [ASP-functionwebhooksNYCReactorgroup-a584 \(Y1-0\)](#)
 Subscription (move): [MOC HOL 100292](#) Runtime version: 4.34.1.22669
 Subscription ID: 7fc993b4-a6fe-42a7-a5fa-e158b67fe74e
 Tags (edit): Add tags

Functions Metrics Properties Notifications (0)

+ Create { Set up local environment Refresh

Filter by name...

Name	Trigger	Status	Monitor
HttpTrigger1	HTTP	Enabled	Invocations and more ...

(Code to Cloud - Figure 14 - Navigate back to the function app -> Logs)

Run Time range: Last 24 hours Save Share + New alert rule Export Pin to Format query

```

1 traces
2 | where message.startswith "(" and message has "timestamp"
3 | extend logEntry = parse_json(message)
4 | project
5   Timestamp = todatetime(logEntry.timestamp),
6   EventType = tostring(logEntry.eventType),
7   RepositoryName = tostring(logEntry.repositoryName),
8   PusherName = tostring(logEntry.pusherName),
9   CommitMessage = tostring(logEntry.commitMessage),
10  IPAddress = tostring(logEntry.ipAddress),
11  Branch = tostring(logEntry.branch),
12  WorkflowName = tostring(logEntry.workflowName),
13  WorkflowStatus = tostring(logEntry.workflowStatus),
14  JobName = tostring(logEntry.jobName),
15  JobStatus = tostring(logEntry.jobStatus),
16  RawLog = tostring(logEntry.RAW_LOG)
17 | order by Timestamp desc
18

```

Results Chart

Timestamp [UTC]	EventType	RepositoryName	PusherName	CommitMessage	IPAddress	WorkflowName	WorkflowStatus	JobName
> 5/12/2024, 1:53:43.783 PM	workflow_run	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.173:62104	CodeQL	completed	
> 5/12/2024, 1:53:43.773 PM	check_suite	ms-reactor-workshop	BillReyor	Unknown	140.82.115.152:59110			
> 5/12/2024, 1:53:43.472 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.26:49156		Analyze (java-kotlin)	
> 5/12/2024, 1:53:43.452 PM	check_run	ms-reactor-workshop	BillReyor	Unknown	140.82.115.117:46048			
> 5/12/2024, 1:53:18.248 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.80:46720		Analyze (javascript-typescript)	
> 5/12/2024, 1:53:18.099 PM	check_run	ms-reactor-workshop	BillReyor	Unknown	140.82.115.110:62946			
> 5/12/2024, 1:51:57.320 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.34:51354		Analyze (java-kotlin)	
> 5/12/2024, 1:51:57.164 PM	workflow_run	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.39:51420	CodeQL	in_progress	
> 5/12/2024, 1:51:56.813 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.254:47430		Analyze (javascript-typescript)	
> 5/12/2024, 1:51:49.965 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.152:568526		Analyze (javascript-typescript)	
> 5/12/2024, 1:51:40.902 PM	workflow_job	ms-reactor-workshop	BillReyor	Unknown	140.82.115.62:47560		Analyze (java-kotlin)	

(Code to Cloud - Figure 15 - Query our log data using the provided KQL)

Run Time range : Last 24 hours Save Share New alert rule Export

```

1 traces
2 | where message startswith "{" and message has "timestamp"
3 | extend logEntry = parse_json(message)
4 | project
5     Timestamp = todatetime(logEntry.timestamp),
6     EventType = tostring(logEntry.eventType),
7     RepositoryName = tostring(logEntry.repositoryName),
8     PusherName = tostring(logEntry.pusherName),
9     CommitMessage = tostring(logEntry.commitMessage),
10    IPAddress = tostring(logEntry.ipAddress),
11    Branch = tostring(logEntry.branch),
12    WorkflowName = tostring(logEntry.workflowName),
13    WorkflowStatus = tostring(logEntry.workflowStatus),
14    JobName = tostring(logEntry.jobName),
15    JobStatus = tostring(logEntry.jobStatus),
16    RawLog = tostring(logEntry.RAW_LOG)
17 | order by Timestamp desc
18

```

Results Chart

Timestamp [UTC]	EventType	RepositoryName	PusherName	CommitMessage
5/12/2024, 1:53:43.783 ...	workflow_run	ms-reactor-worksh...	BillReyor	Update README.mc
	Timestamp [UTC]	2024-05-12T13:53:43.783Z		
	EventType	workflow_run		
	RepositoryName	ms-reactor-workshop		
	PusherName	BillReyor		
	CommitMessage	Update	 Copy	
	IPAddress	140.82	 Include "BillReyor"	

(Code to Cloud - Figure 16 - Open a log entry and add a filter to our KQL)

Run Time range : Last 24 hours Save Share New alert rule Export Pin to Format query

```

2 | where message startswith "{" and message has "timestamp"
3 | extend logEntry = parse_json(message)
4 | project
5     Timestamp = todatetime(logEntry.timestamp),
6     EventType = tostring(logEntry.eventType),
7     RepositoryName = tostring(logEntry.repositoryName),
8     PusherName = tostring(logEntry.pusherName),
9     CommitMessage = tostring(logEntry.commitMessage),
10    IPAddress = tostring(logEntry.ipAddress),
11    Branch = tostring(logEntry.branch),
12    WorkflowName = tostring(logEntry.workflowName),
13    WorkflowStatus = tostring(logEntry.workflowStatus),
14    JobName = tostring(logEntry.jobName),
15    JobStatus = tostring(logEntry.jobStatus),
16    RawLog = tostring(logEntry.RAW_LOG)
17 | order by Timestamp desc
18 | where PusherName == "BillReyor"
19 | where CommitMessage == "Update README.md"

```

Results Chart

Timestamp [UTC]	EventType	RepositoryName	PusherName	CommitMessage	IPAddress
> 5/12/2024, 1:53:43.783 PM	workflow_run	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.173:62104
> 5/12/2024, 1:51:57.164 PM	workflow_run	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.39:51420
> 5/12/2024, 1:51:48.078 PM	workflow_run	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.255:27664
> 5/12/2024, 1:51:46.595 PM	push	ms-reactor-workshop	BillReyor	Update README.md	140.82.115.168:33034

(Code to Cloud - Figure 17 - Observe how having pivotable log data is powerful)