Developing secure software with GitHub

Laura Kokkarinen & Thomas Vochten

Who are we?

Technology Evangelist, MVP

Thomas Vochten

De Cronos Groep, Belgium

@ThomasVochten

Software Architect, MVP

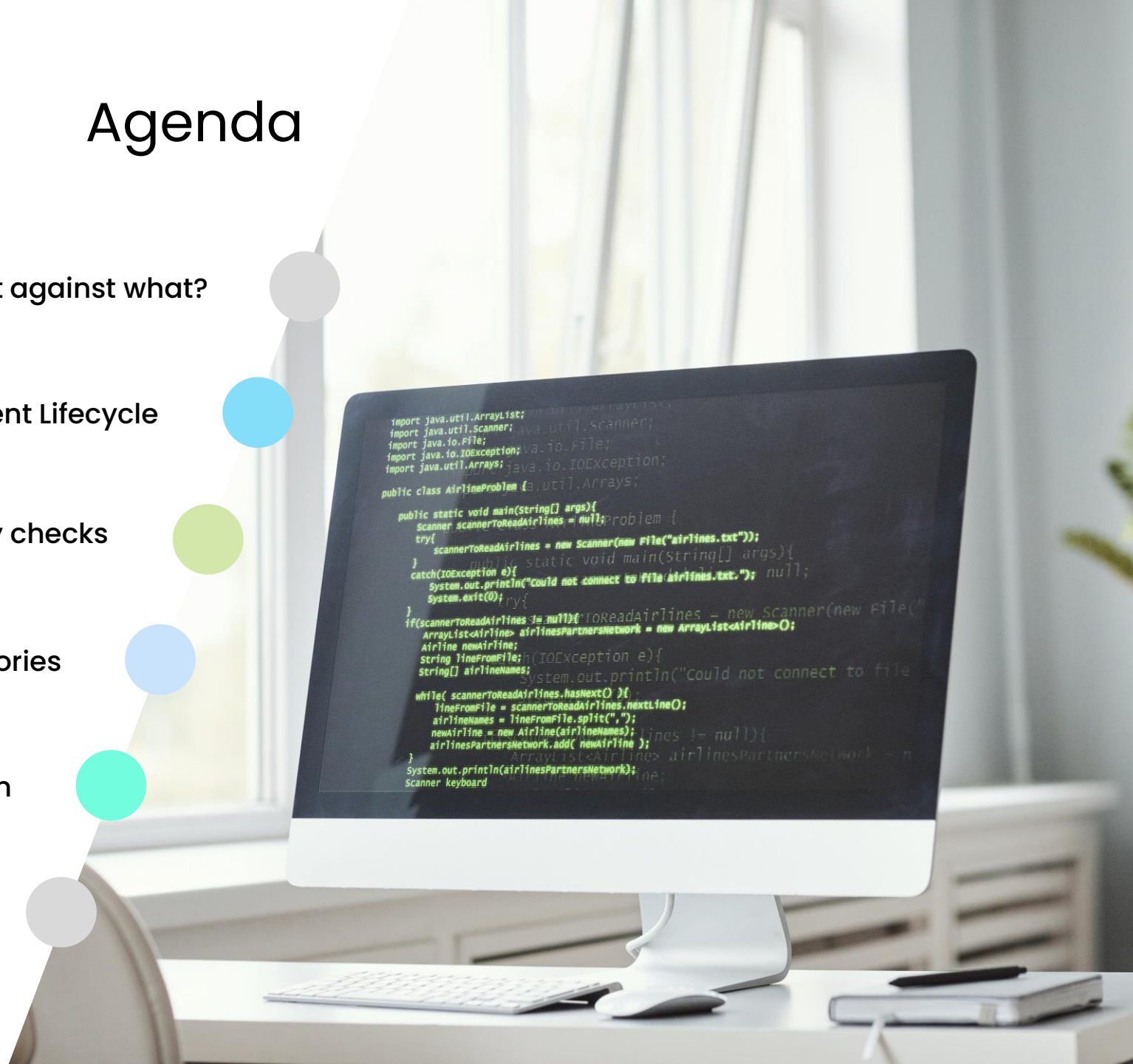
Laura Kokkarinen

Sulava, Finland

@LauraKokkarinen

Agenda

- Protect against what?
- Secure Software Development Lifecycle
- Automating code security checks
- Protecting source code and repositories
- Defender for Cloud integration
- What about Azure DevOps?

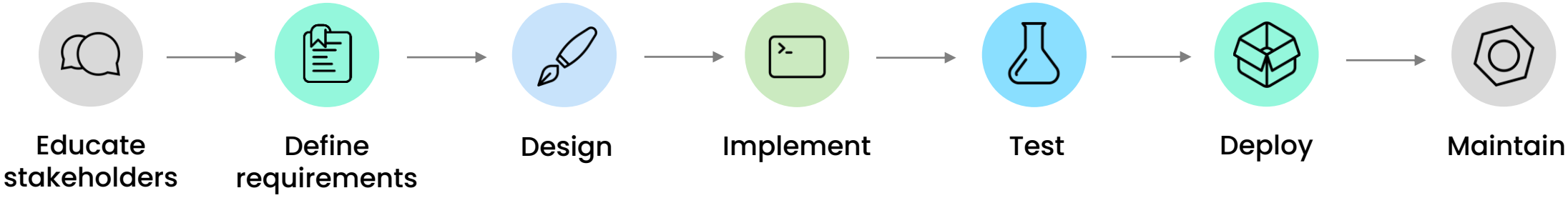


OWASP Top 10

- Broken Access Control
- Vulnerable and Outdated Components
- Cryptographic Failures
- Identification and Authentication Failures
- Injection
- Software and Data Integrity Failures
- Insecure Design
- Security Logging and Monitoring Failures
- Security Misconfiguration
- Server-Side Request Forgery (SSRF)

Secure Software Development Lifecycle (S-SDLC)

- Software development process that prioritizes security at every stage
- Framework



Microsoft Security Development Lifecycle (SDL)

1. Provide security training

4. Perform Threat Modeling
5. Establish Design Requirements

9. Perform Static Analysis Security Testing
10. Perform Dynamic Analysis Security Testing
11. Perform Penetration Testing

Educate stakeholders | **Defined requirements** | **Design** | **Implement** | **Test** | **Deploy** | **Maintain**

2. Define Security Requirements
3. Define Metrics and Compliance Reporting

6. Define and Use Cryptography Standards
7. Manage the Security Risk of Using Third-Party Components
8. Use Approved Tools

12. Establish a Standard Incident Response Process

Why is it important and what will you gain?

- Attacks targeting apps become ever more prevalent and sophisticated
- Initially requires additional resources but has ROI over long term

Reduced number of security vulnerabilities

Improved overall quality

Compliance with regulations and standards

Reduced development costs

Competitive advantage through reputation and customer trust

Improved customer satisfaction

When is it feasible to implement?

- Viable for all projects
- Ideally adopted from the very beginning

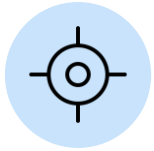
Business, enterprise or infrastructure environment

Sensitive information

Communicates over a network

What steps can be automated?

- Often some manual work is combined with automation



Threat modeling

Code analysis

Security testing

Configuration scanning

Continuous integration and deployment

Incident response

UP NEXT

How can GitHub help us automate steps during SDLC?

Automated testing is key

- Provide a baseline quality check
- Avoid common mistakes or anti-patterns
- Awareness of vulnerabilities
- Consistency through CI/CD integration

Manual tests are still valuable

- Reviewing pull requests
- Validation of design decisions
- Applying common sense

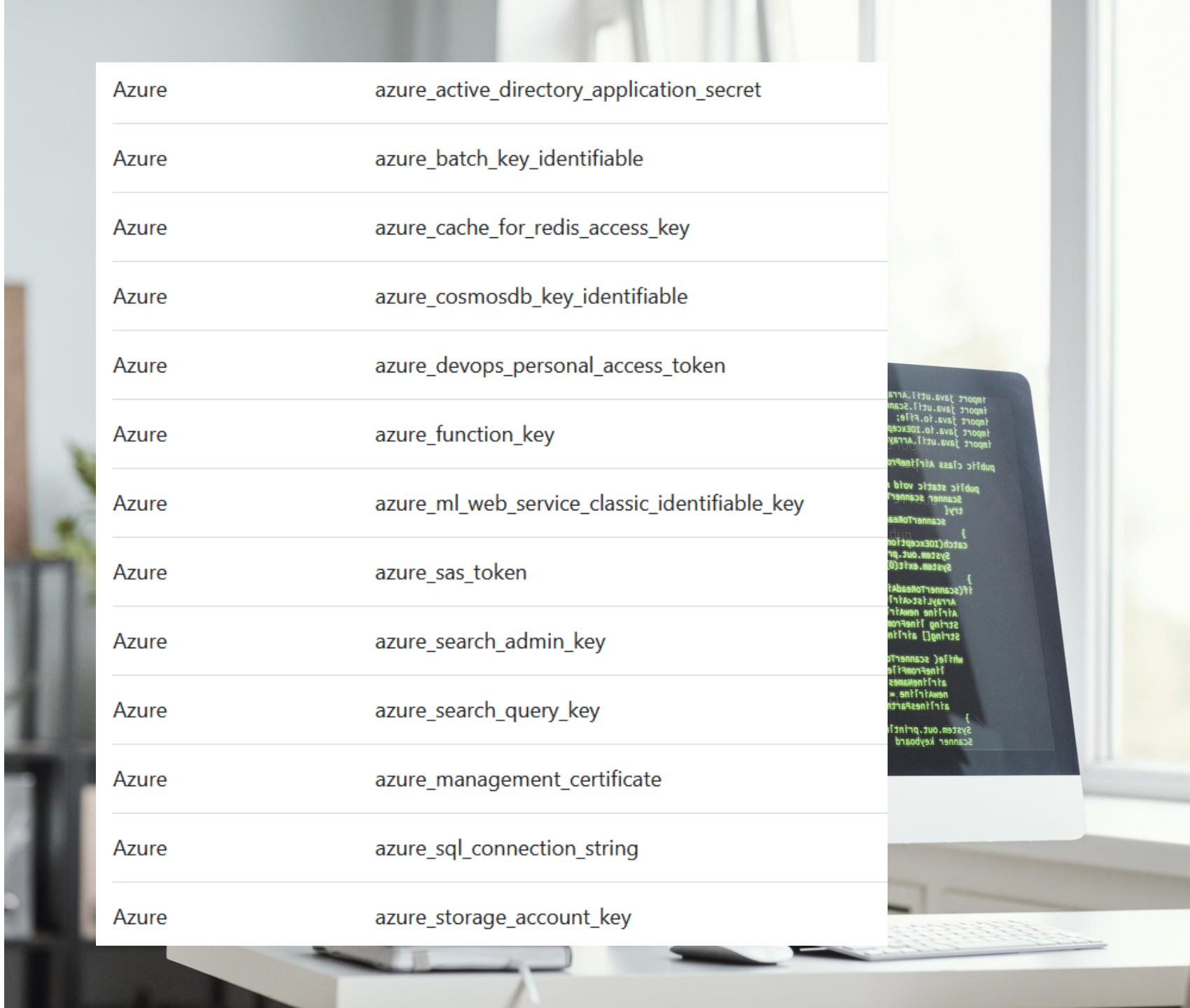
GitHub to the rescue!

- Secret Scanning
- Code Scanning
- Dependency Scanning

Secret Scanning – What?

- Prevents exposing tokens, private keys or other secrets
- Scanning across all branches and git history
- Looks for patterns provided by the vendors
- Reported as alerts in the repository’s Security tab
- Free for public repositories

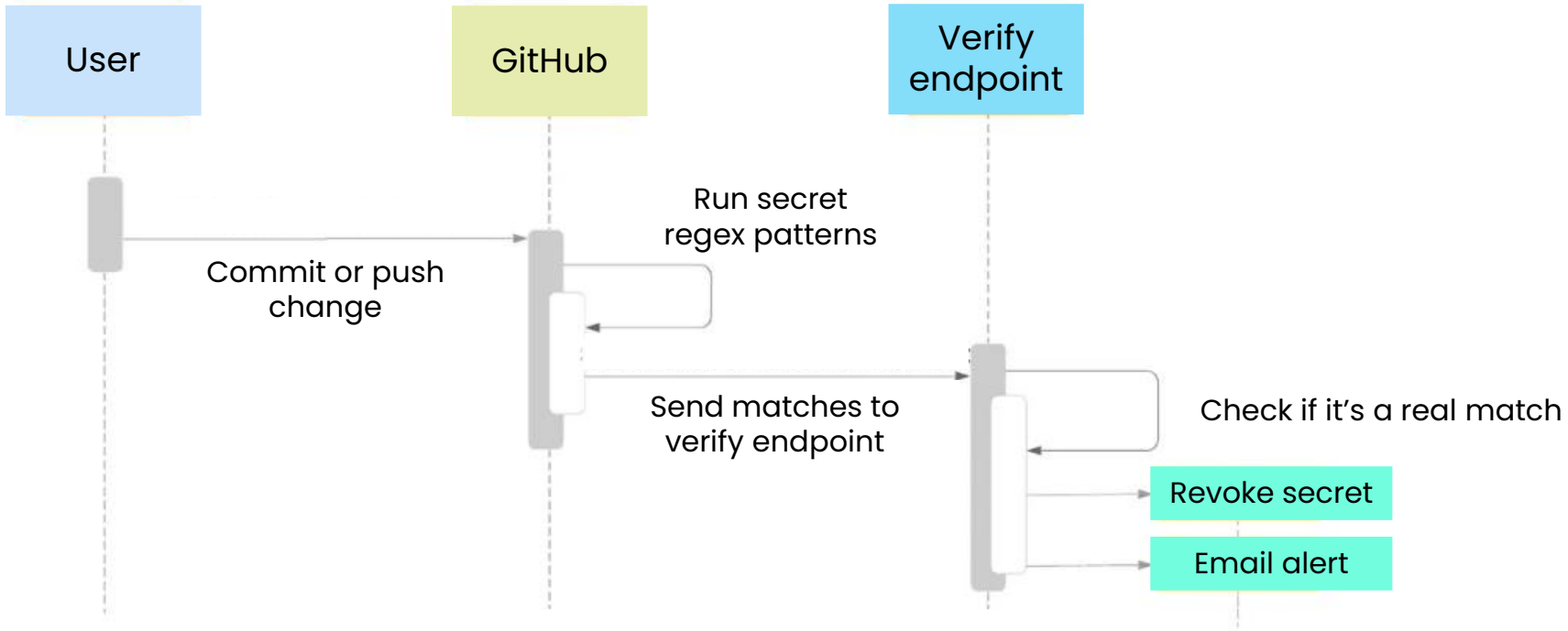
Secret Scanning Patterns



Azure	azure_active_directory_application_secret
Azure	azure_batch_key_identifiable
Azure	azure_cache_for_redis_access_key
Azure	azure_cosmosdb_key_identifiable
Azure	azure_devops_personal_access_token
Azure	azure_function_key
Azure	azure_ml_web_service_classic_identifiable_key
Azure	azure_sas_token
Azure	azure_search_admin_key
Azure	azure_search_query_key
Azure	azure_management_certificate
Azure	azure_sql_connection_string
Azure	azure_storage_account_key

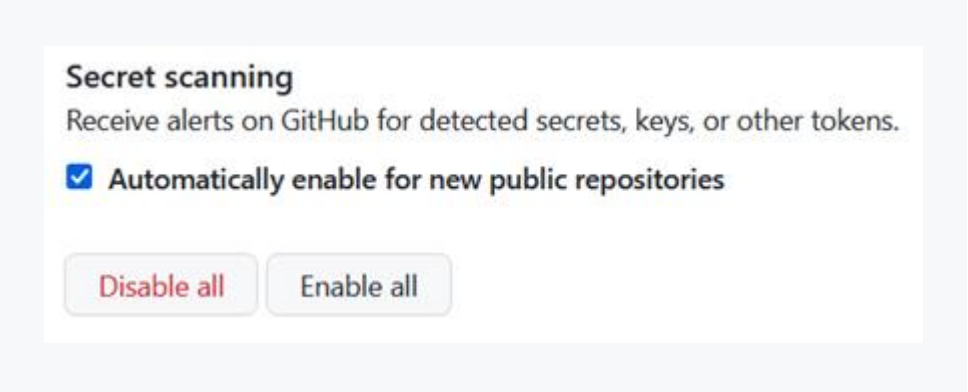
Secret Scanning – Partner Program

- Contributed by partners such as Microsoft
- Offers additional intelligence and automatic revoking of secrets (!)



Secret Scanning - Configuration

- **Scope**
  Enable secret scanning for a single repository, or on an account level

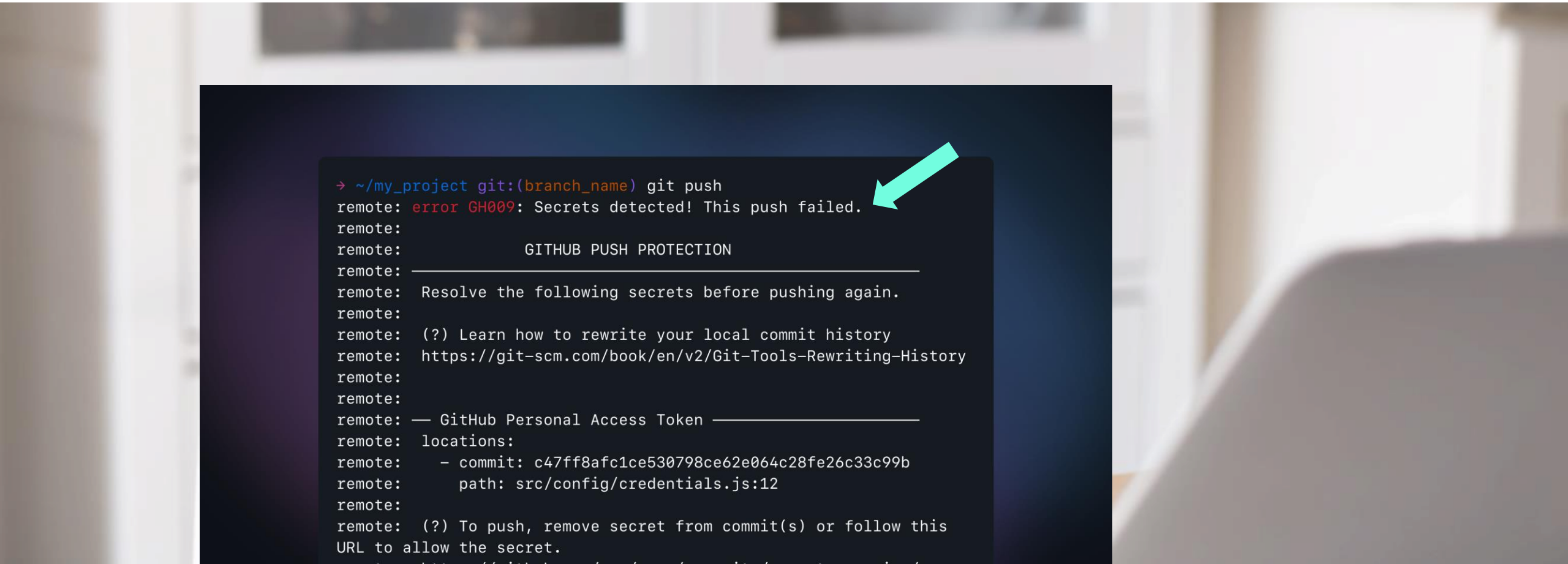


- **Exclusions**
  You can exclude directories through a .github/secret_scanning .yml file

```
paths-ignore:
  - "foo/bar/*.js"
```

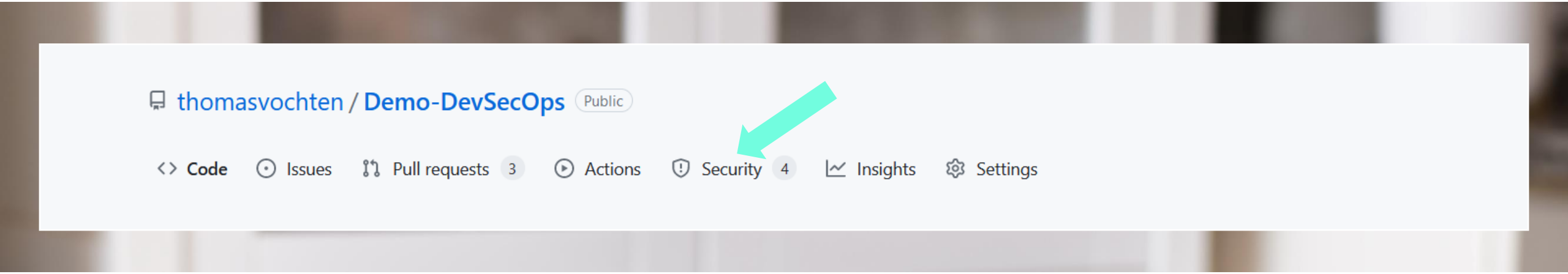
Secret Scanning – Push Protection

- Free for public repositories

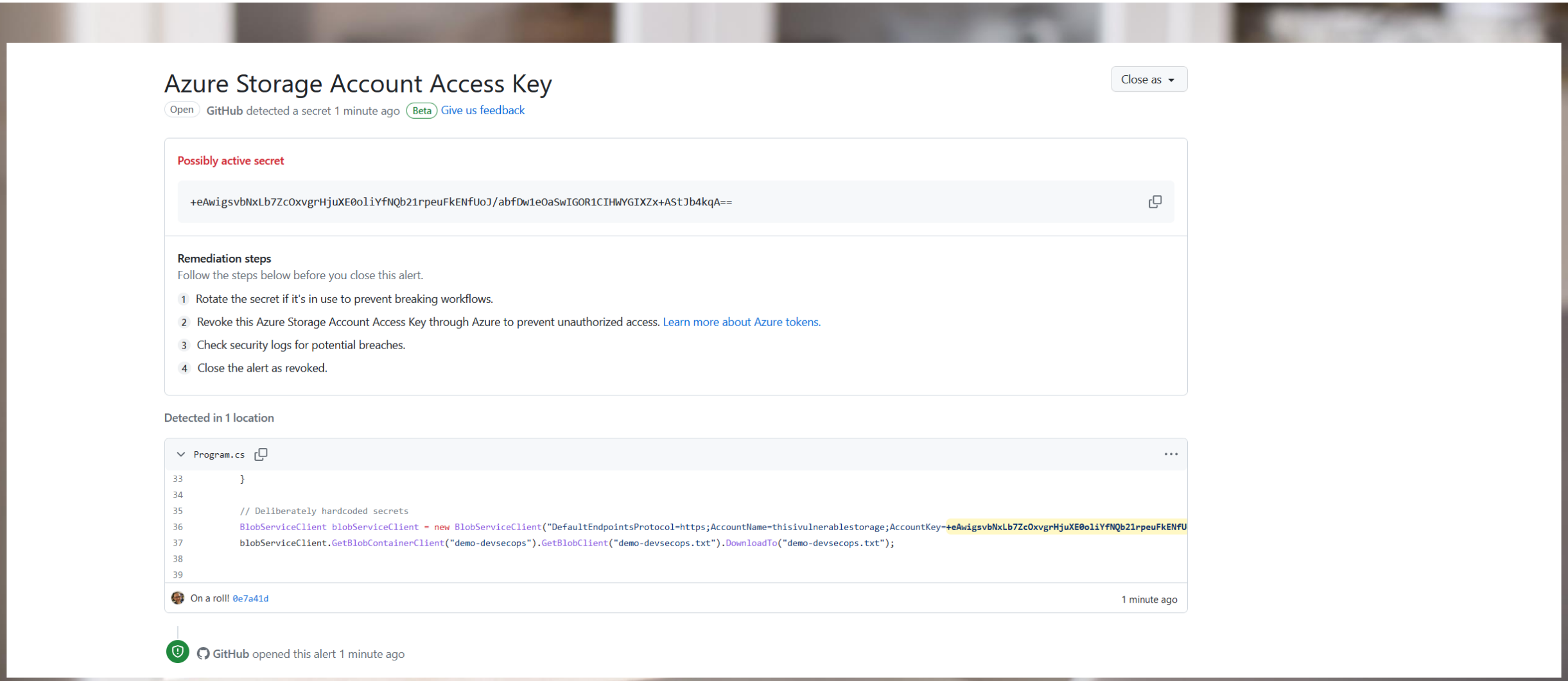


Secret Scanning – Alerting

- Alerts are sent to
  - Contributor who committed the secret
  - Repository administrators
  - Organization owners
- Only admins can dismiss secret scanning alerts



Secret Scanning – Alerting



Secret Scanning – What now?

- How do I remove secrets from git history?
- Examples:

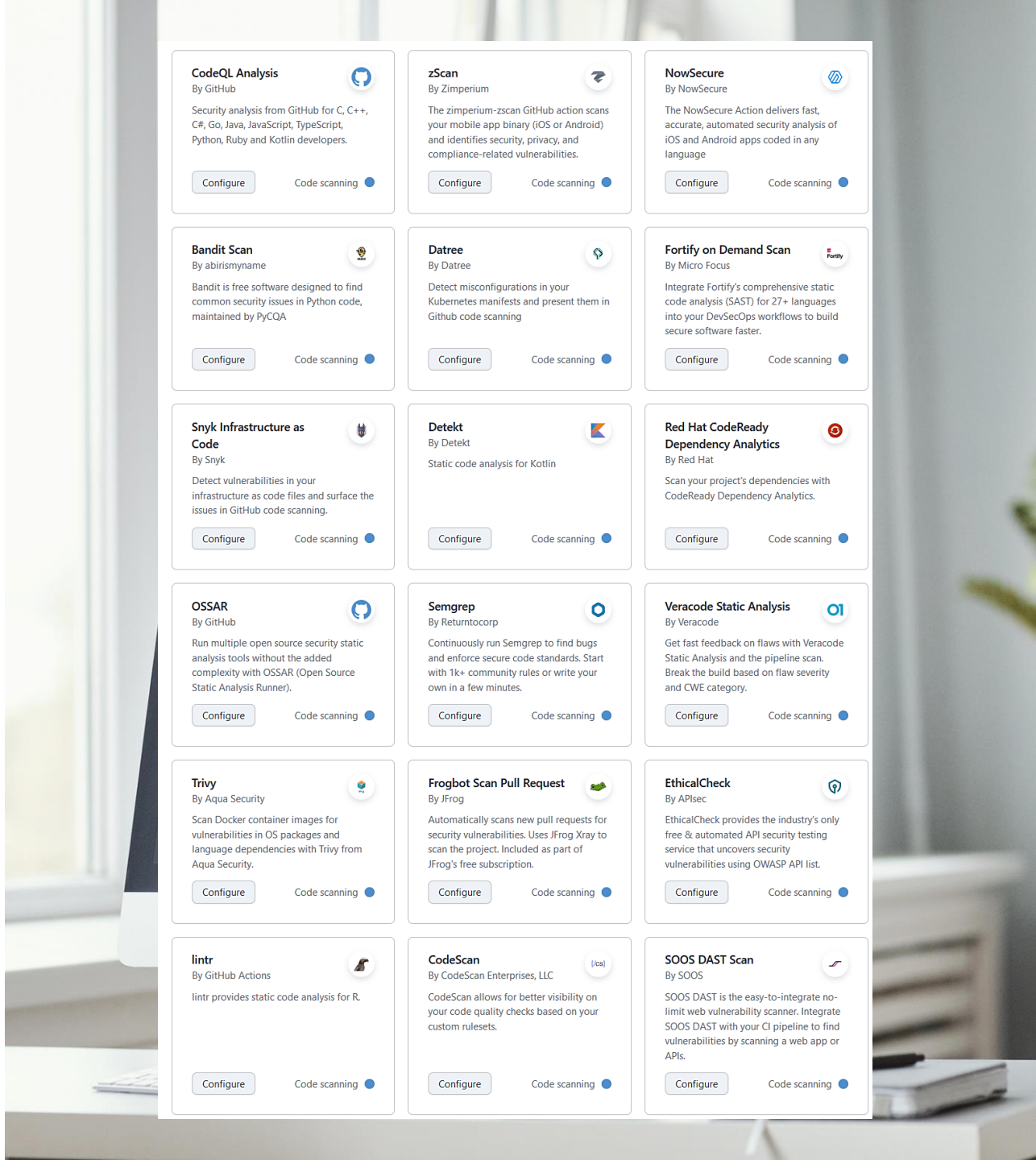
```
$ bfg --delete-files YOUR-FILE-WITH-SENSITIVE-DATA
$ bfg --replace-text passwords.txt

$ git push --force
```

Code Scanning – What?

- Identify and fix security vulnerabilities and coding errors
- Scheduled scans or trigger on certain events (push)
- Creates an alert (and closes it automatically)
- Uses GitHub Actions
- Free for public repositories

Code Scanning – How?

- CodeQL
- Or through a 3rd party tool that supports the Static Analysis Results Interchange Format (SARIF)



Code Scanning – CodeQL

CodeQL is a code analysis engine to automate security checks.

Database

You generate a CodeQL database to represent your codebase

Queries

You run CodeQL queries on that database to identify problems in the codebase

Alerts

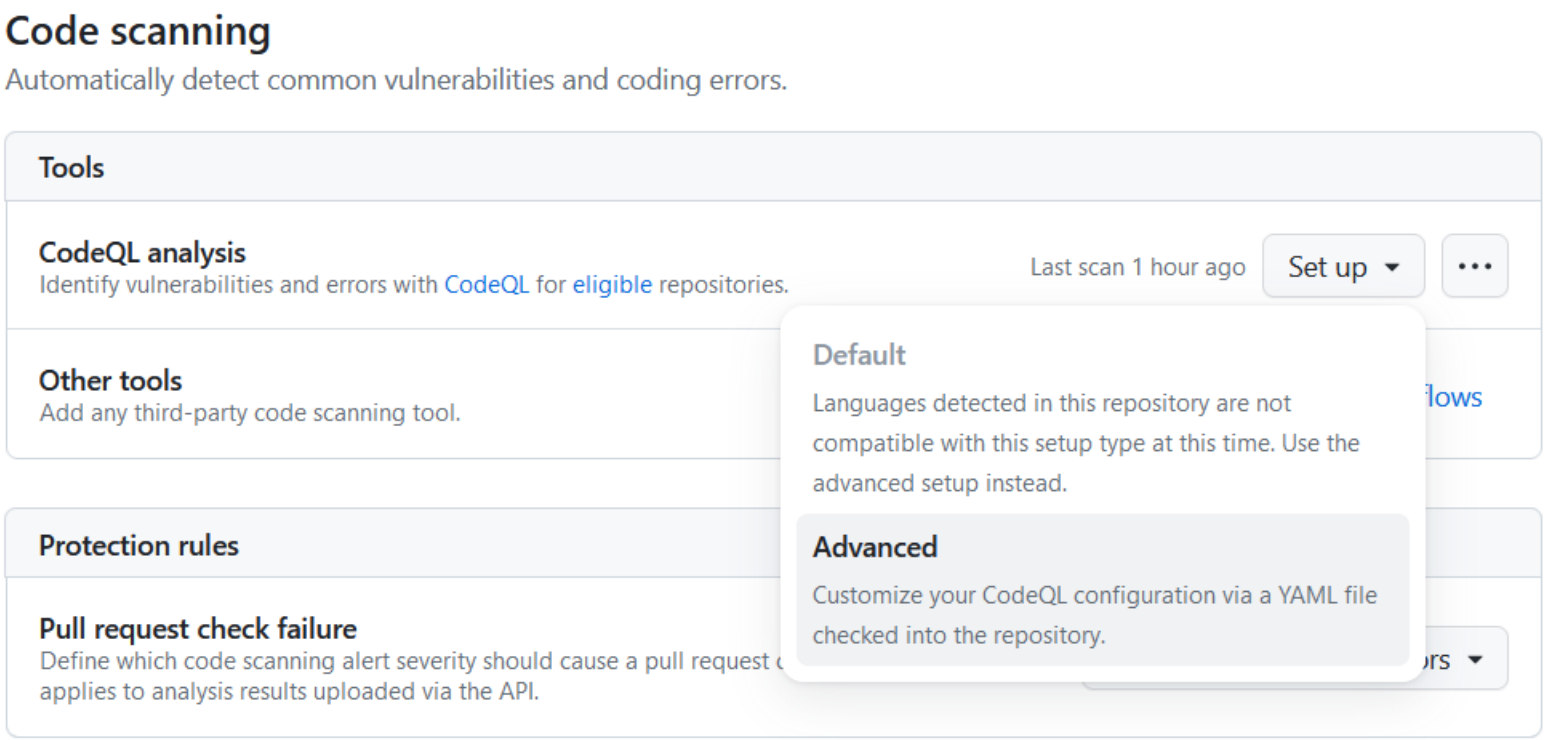
The query results are shown as code scanning alerts in GitHub when you use CodeQL with code scanning.

Code Scanning – CodeQL

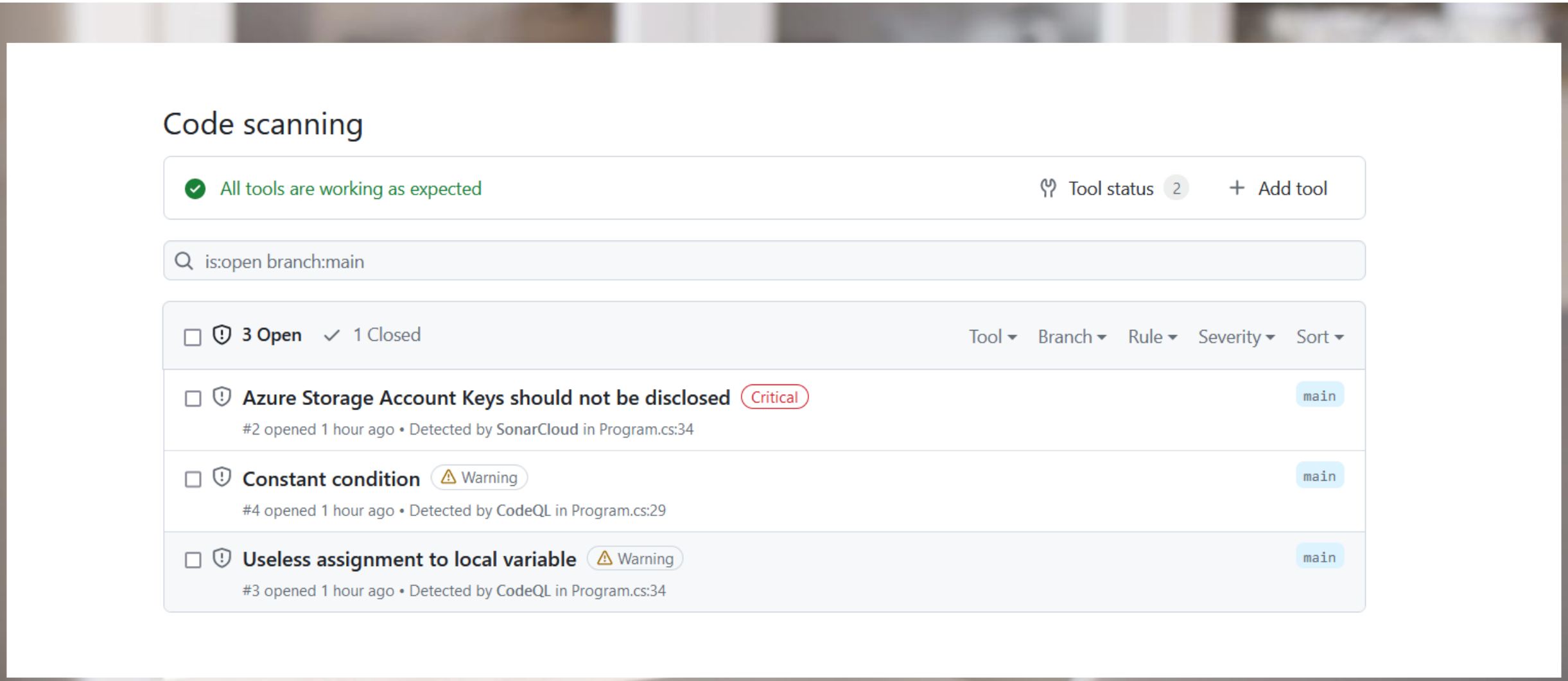
- CodeQL code scanning automatically detects code written in the supported languages:

Language	Variants	Compilers	Extensions
C/C++	C89, C99, C11, C18, C++98, C++03, C++11, C++14, C++17, C++20 [1]	Clang (and clang-cl [2]) extensions (up to Clang 12.0), GNU extensions (up to GCC 11.1), Microsoft extensions (up to VS 2019), Arm Compiler 5 [3]	`.cpp`, `.c++`, `.cxx`, `.hpp`, `.hh`, `.h++`, `.hxx`, `.c`, `.cc`, `.h`
C#	C# up to 10.0	Microsoft Visual Studio up to 2019 with .NET up to 4.8, .NET Core up to 3.1 .NET 5, .NET 6	`.sln`, `.csproj`, `.cs`, `.cshtml`, `.xaml`
Go (aka Golang)	Go up to 1.20	Go 1.11 or more recent	`.go`
Java	Java 7 to 20 [4]	javac (OpenJDK and Oracle JDK), Eclipse compiler for Java (ECJ) [5]	`.java`
Kotlin [6]	Kotlin 1.5.0 to 1.8.20	kotlinc	`.kt`
JavaScript	ECMAScript 2022 or lower	Not applicable	`.js`, `.jsx`, `.mjs`, `.es`, `.es6`, `.htm`, `.html`, `.xhtm`, `.xhtml`, `.vue`, `.hbs`, `.ejs`, `.njk`, `.json`, `.yaml`, `.yml`, `.raml`, `.xml` [7]
Python [8]	2.7, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11	Not applicable	`.py`
Ruby [9]	up to 3.2	Not applicable	`.rb`, `.erb`, `.gemspec`, `Gemfile`
TypeScript [10]	2.6-4.9	Standard TypeScript compiler	`.ts`, `.tsx`, `.mts`, `.cts`

Code Scanning – CodeQL

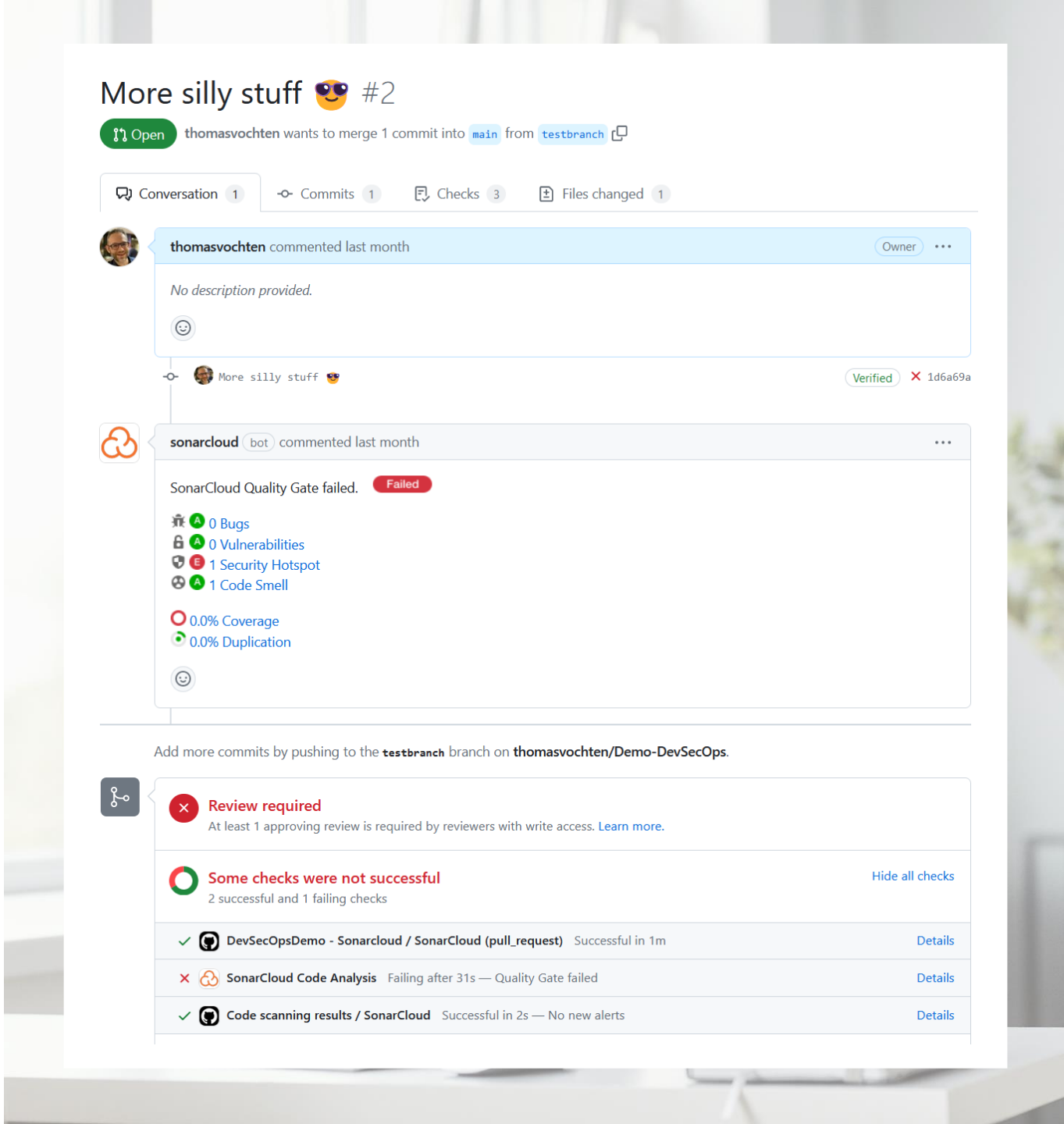


Code Scanning – Alerting



Code Scanning

Pull request integration



Dependabot – What?

Alerts

GitHub creates alerts when a vulnerable dependency or malware is detected

Scans

Scans when a new advisory is published or when you change the dependencies of your project

Fixes

Dependabot can fix vulnerable dependencies for you by raising pull requests with security updates.

Updates

You can use Dependabot to keep the packages you use updated to the latest versions.

PR integration

Can also integrate as a pull request (PR) check

Free

Free for all repositories

Dependabot

GitHub security alert digest

thomasvochten's repository security updates from the week of **Nov 21 - Nov 28**

thomasvochten's personal account

thomasvochten / Demo-DevSecOps

Known security vulnerabilities detected

Dependency `System.Data.SqlClient` Version `<= 4.8.4` Upgrade to `~> 4.8.5`

Defined in **DevSecOpsDemo.csproj**

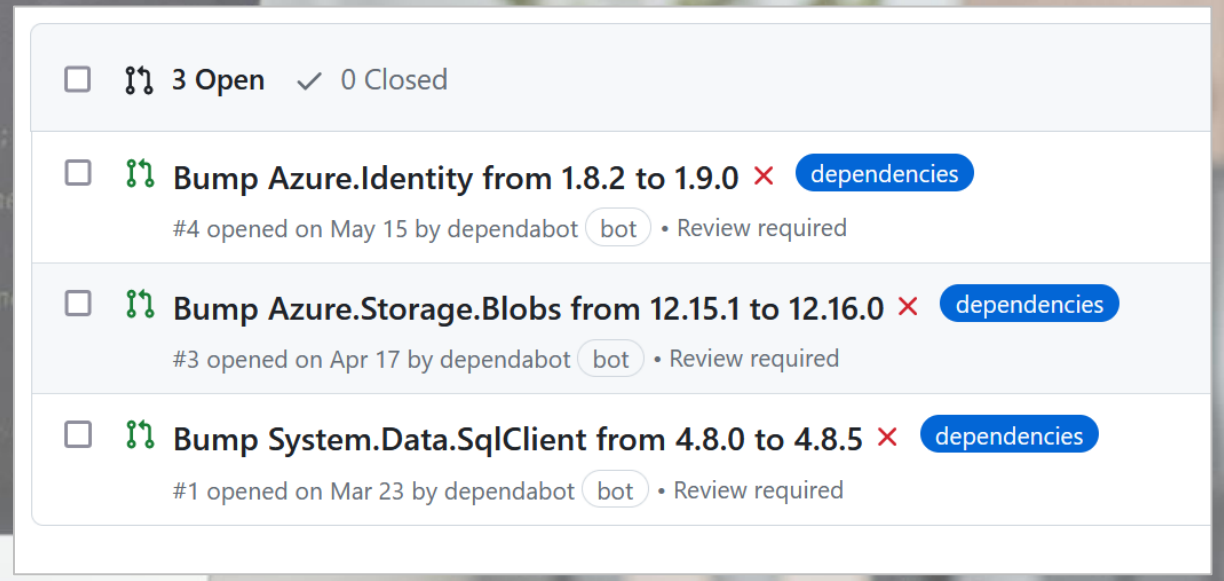
Suggested update #1

Vulnerabilities
CVE-2022-41064 Moderate severity

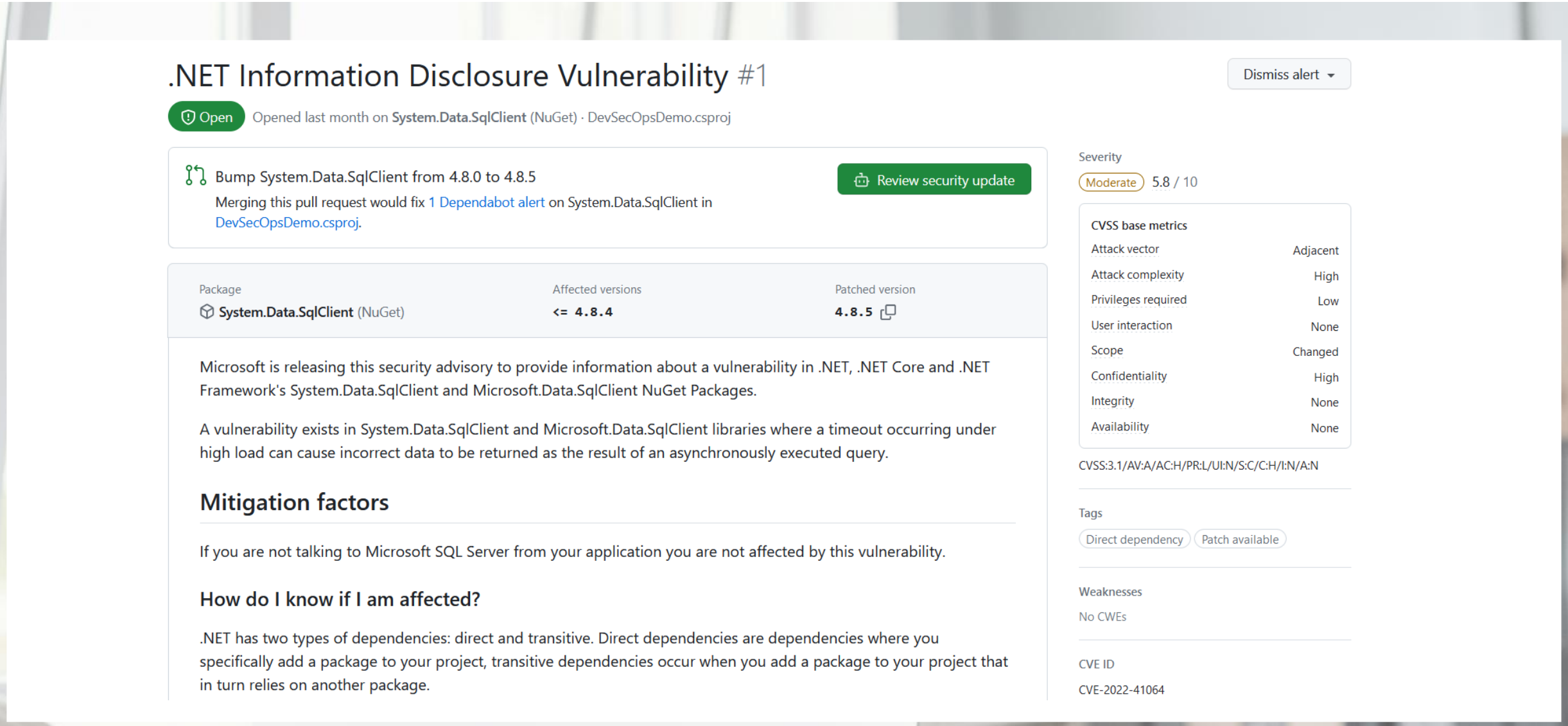
Dependency `Azure.Identity` Version `< 1.10.2` Upgrade to `~> 1.10.2`

Defined in **DevSecOpsDemo.csproj**

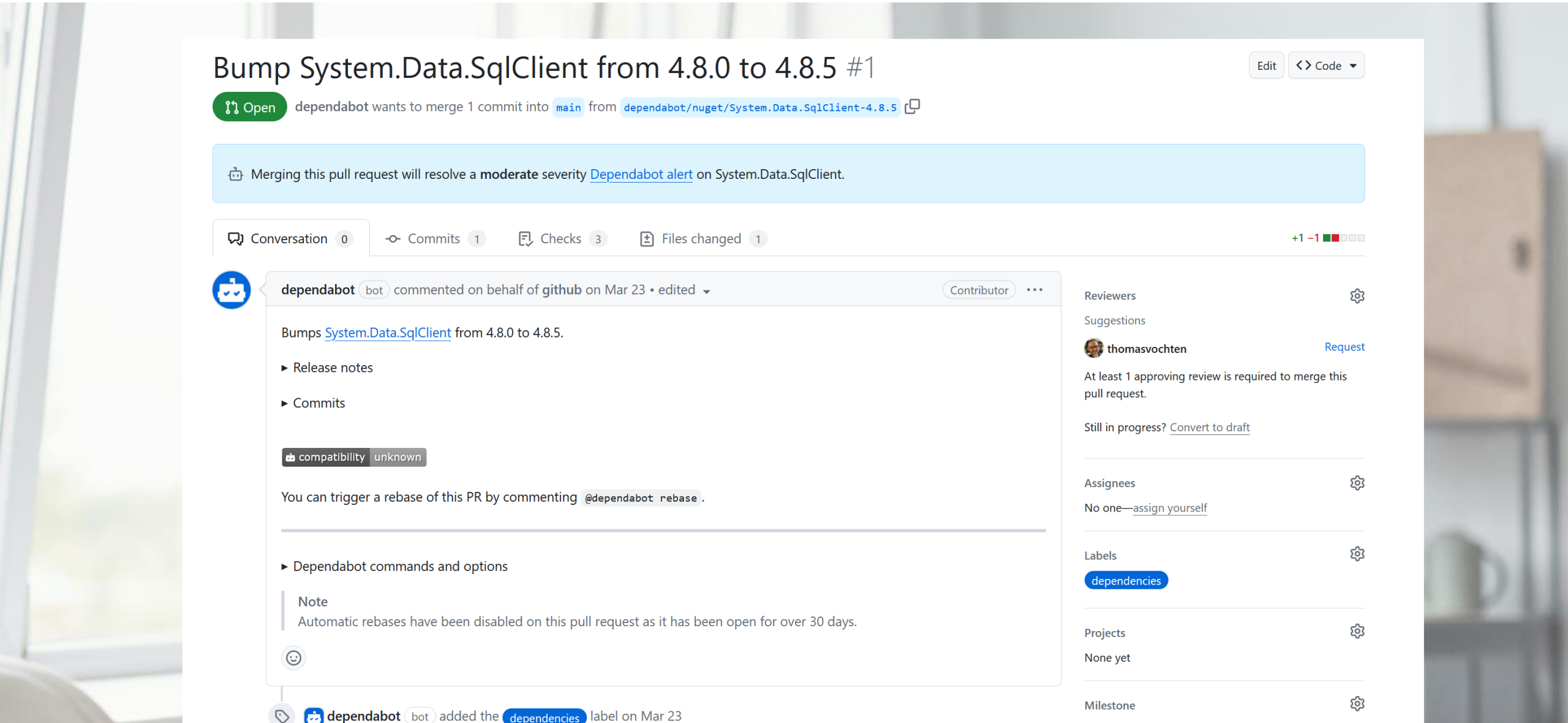
Vulnerabilities
CVE-2023-36414 High severity



Dependabot – Alerting



Dependabot – Pull Request



GitHub to the rescue!

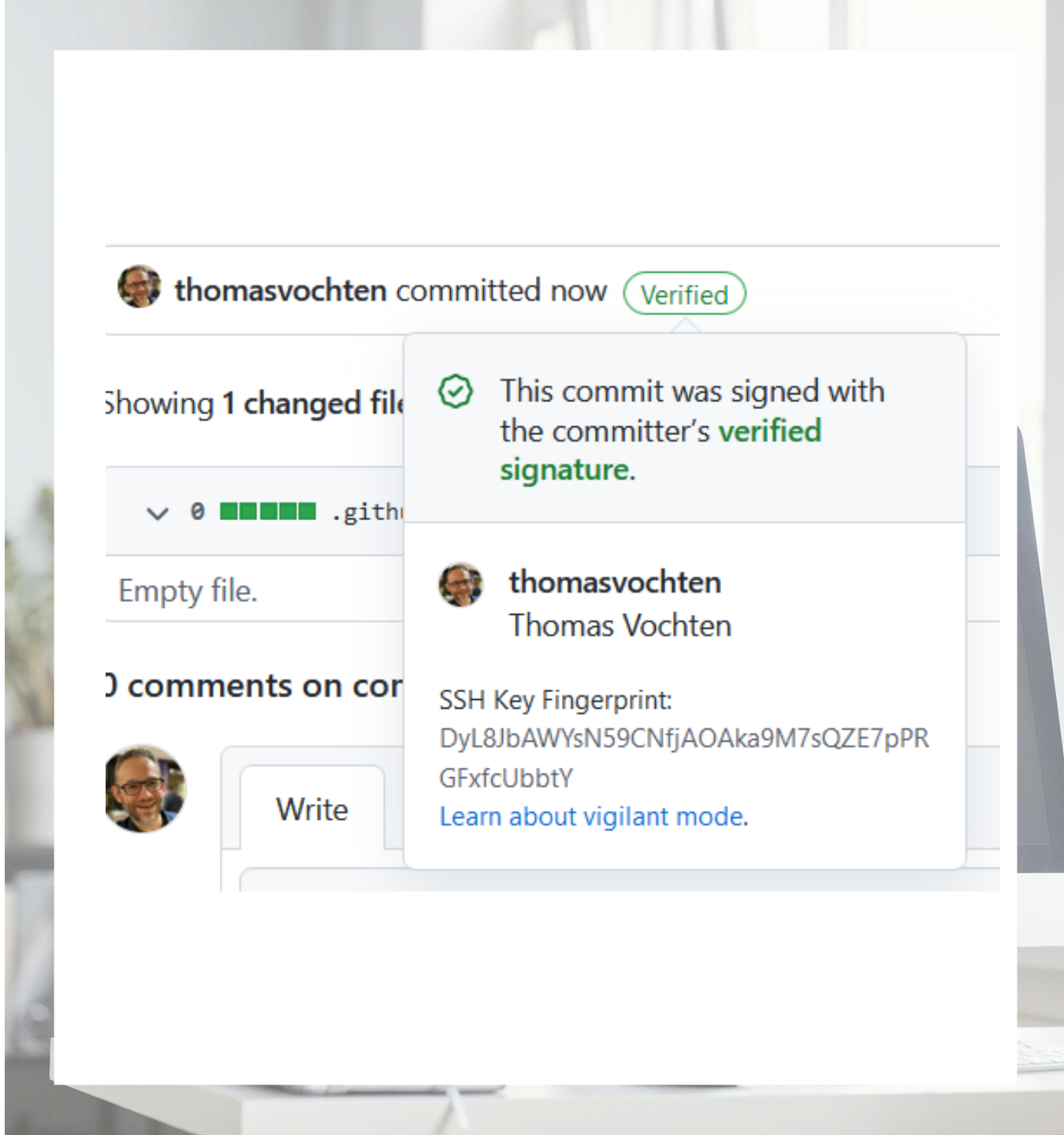
- Secret Scanning
- Code Scanning
- Dependency Scanning

GitHub Account Security

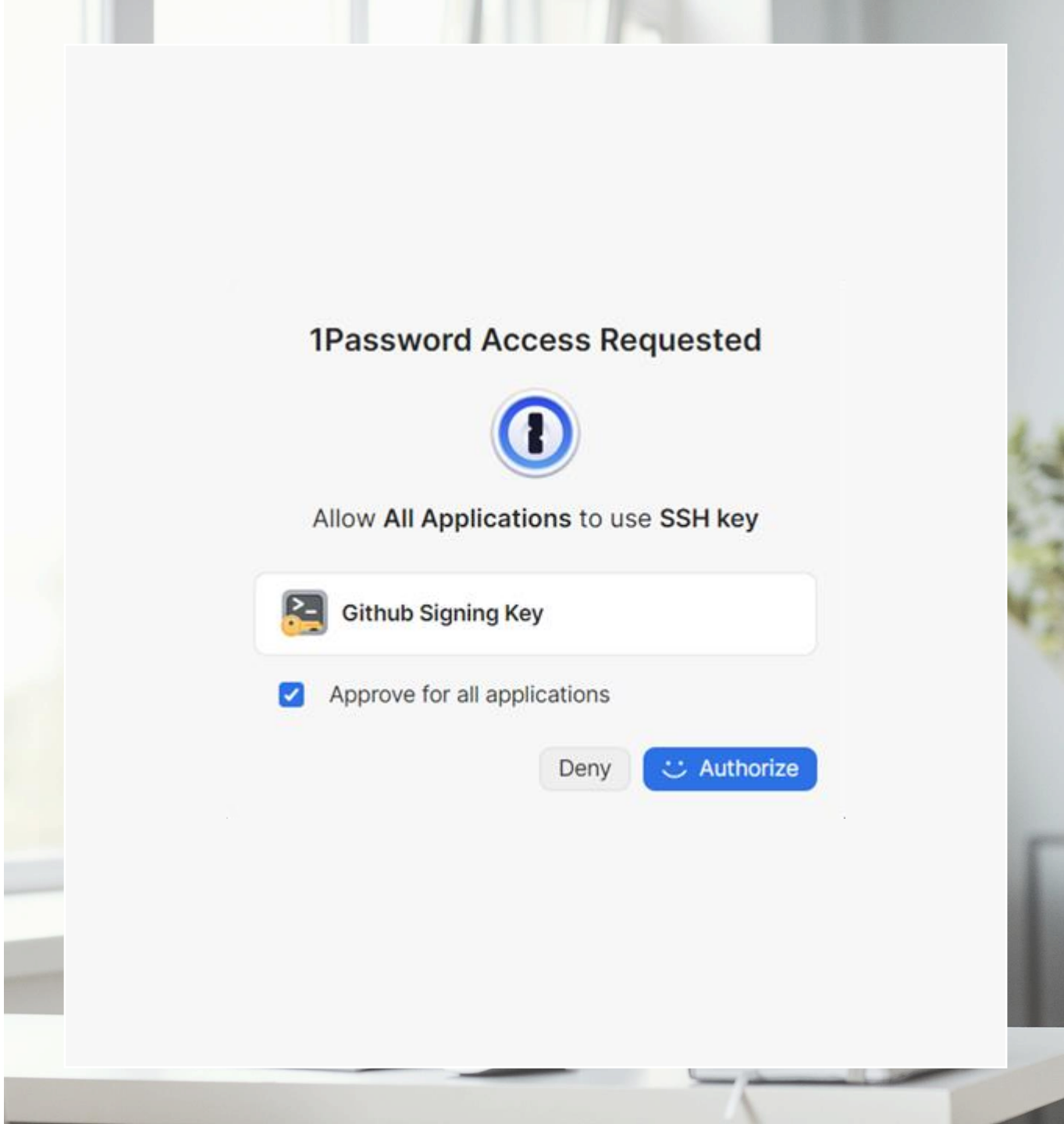
- Username/Password with MFA
  - Time-based one-time password (TOTP)
  - GitHub Mobile
  - Security Key
  - Passkey
- Personal Access Token (PAT)
- SSH Key

Verified Commits

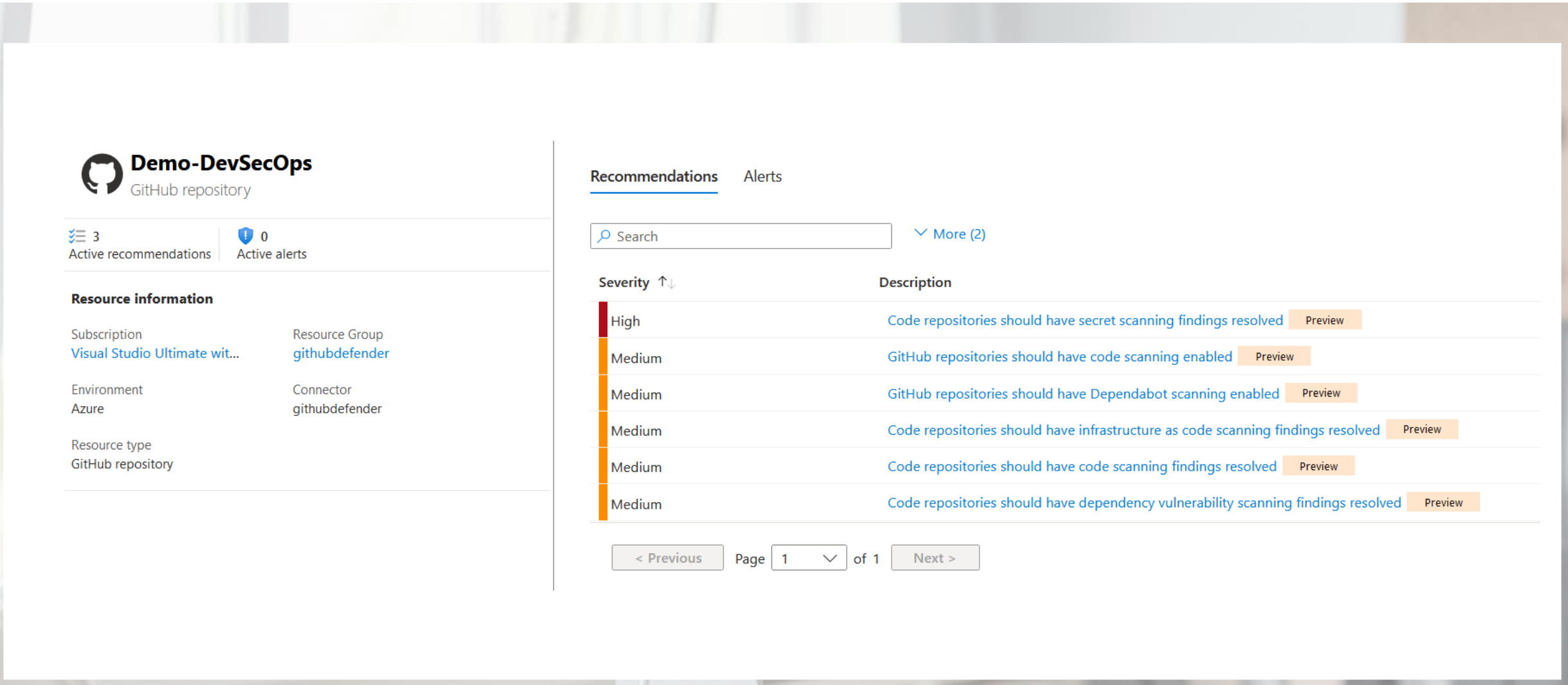
Using GPG, SSH, or S/MIME, you can sign tags and commits locally. These tags or commits are marked as verified on GitHub so other people can be confident that the changes come from a trusted source.



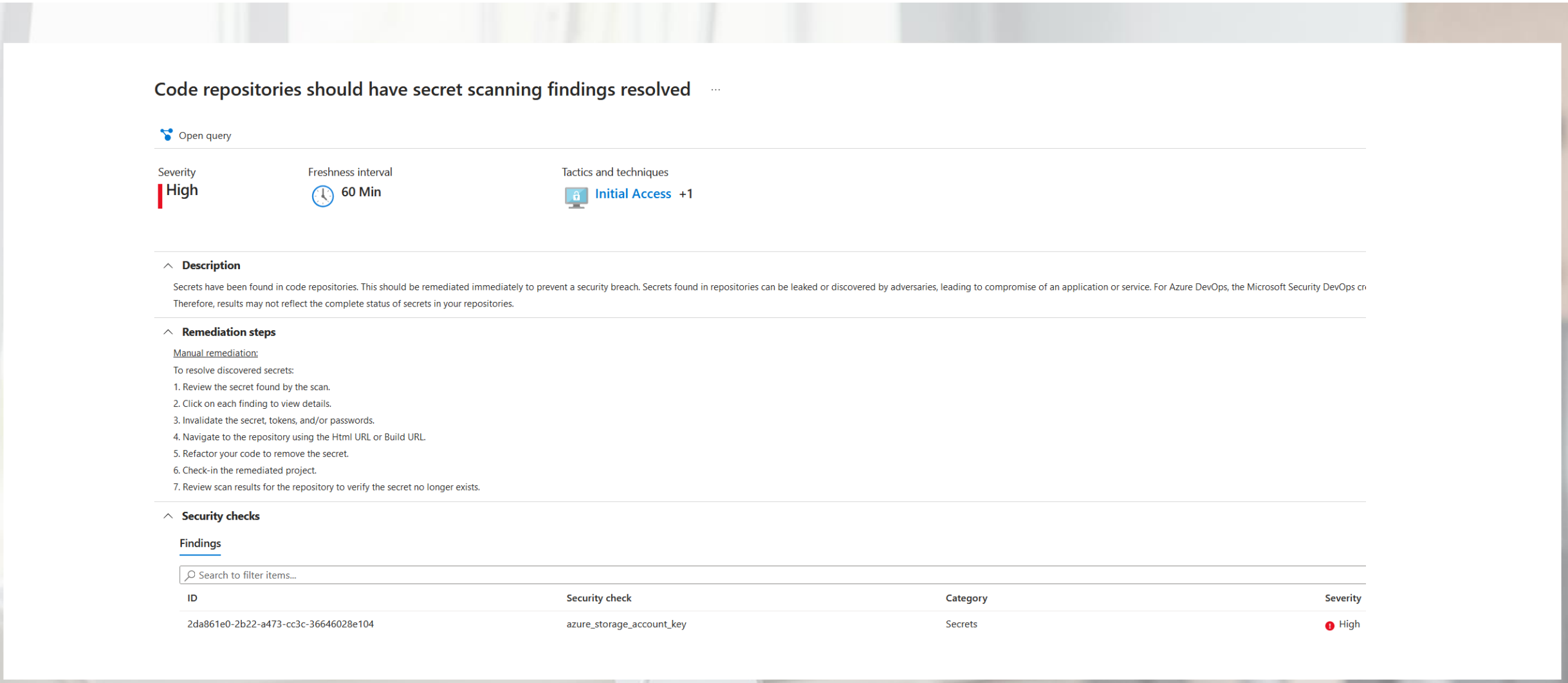
Verified commits



Defender for Cloud integration



Defender for Cloud integration



What about Azure DevOps?

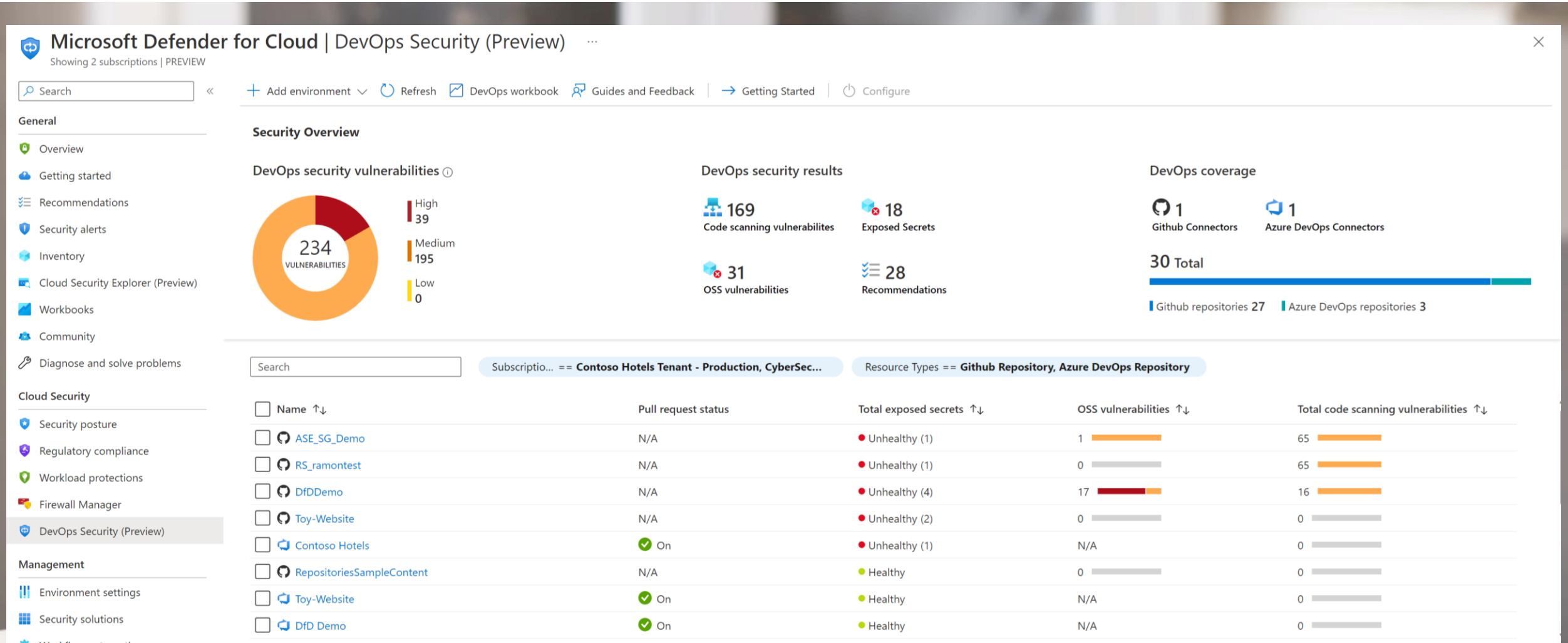
Similar features?

Something missing?

Something more?

Secret scanning

- Microsoft Defender for DevOps (preview; currently free, future cost unknown)

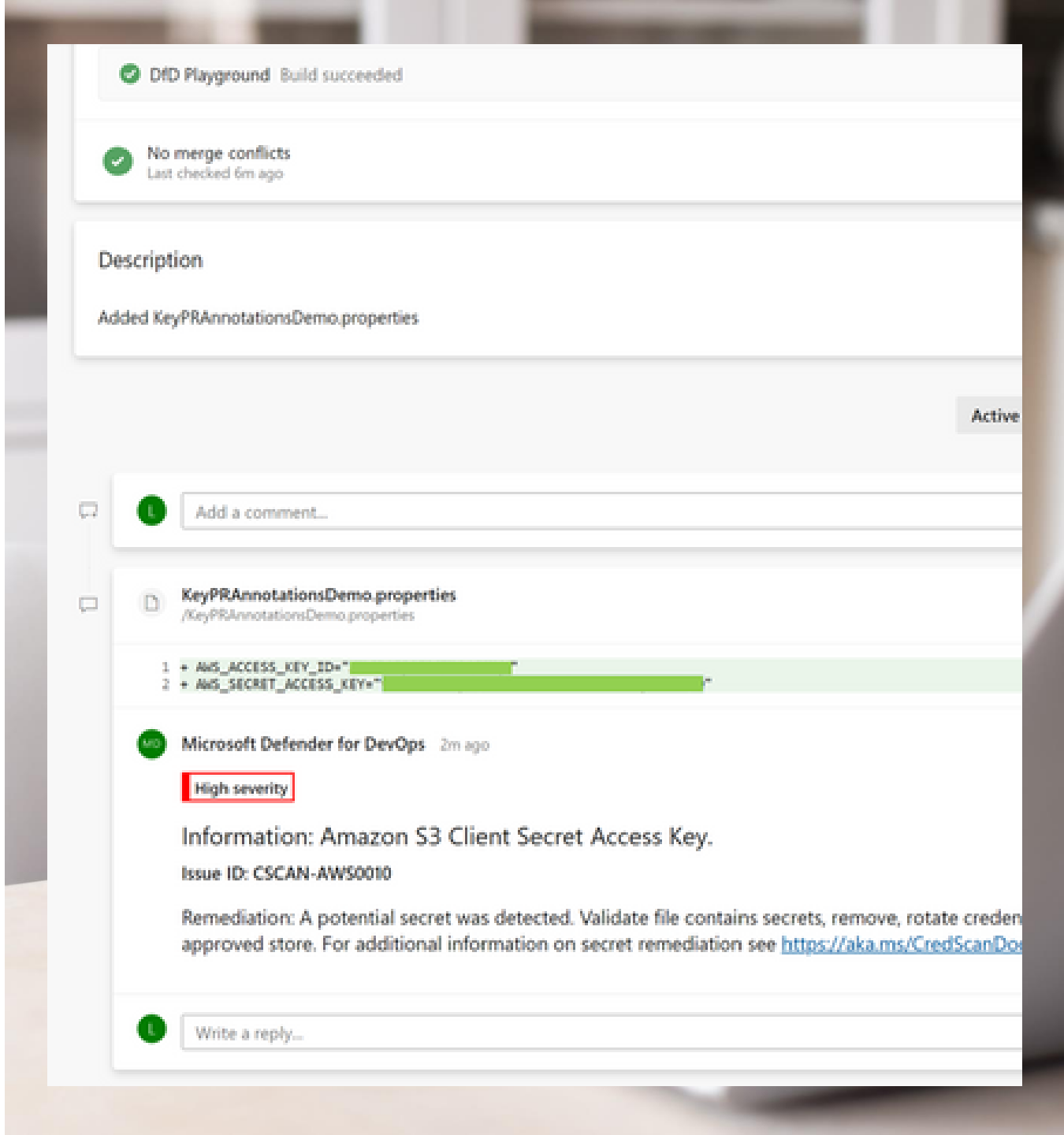


Secret scanning

- Enable Microsoft Defender for DevOps on Azure Portal
- Add Microsoft Security DevOps task to pipeline
- Needs to be run for every repo/branch you want to scan

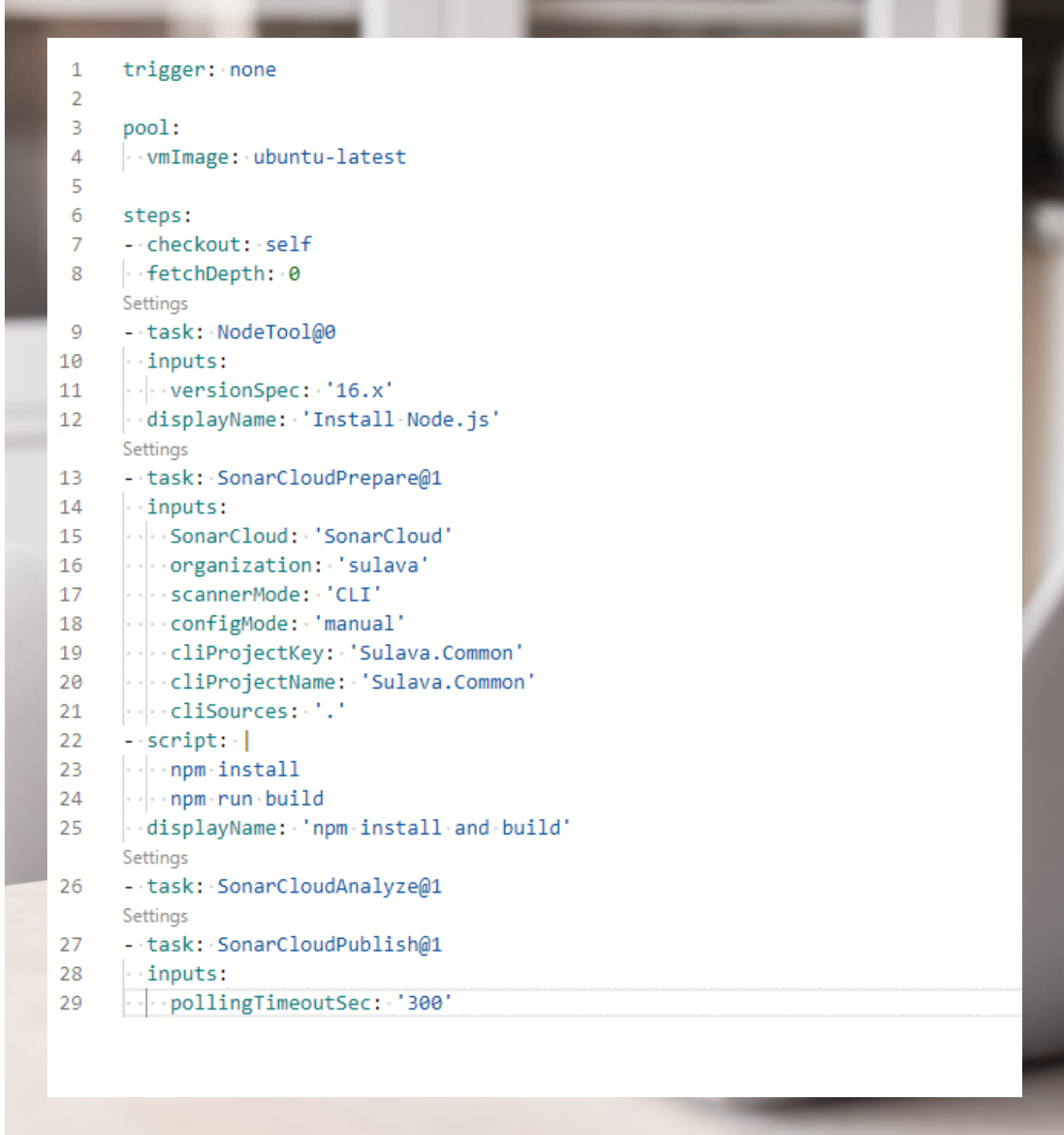
Secret scanning

- Results show in Azure Portal, build results, pull requests (if enabled)
- Combine pull request annotations with a branch policy that requires resolving all comments on a pull request
- Promote the usage of git pre-commit hooks



Code scanning

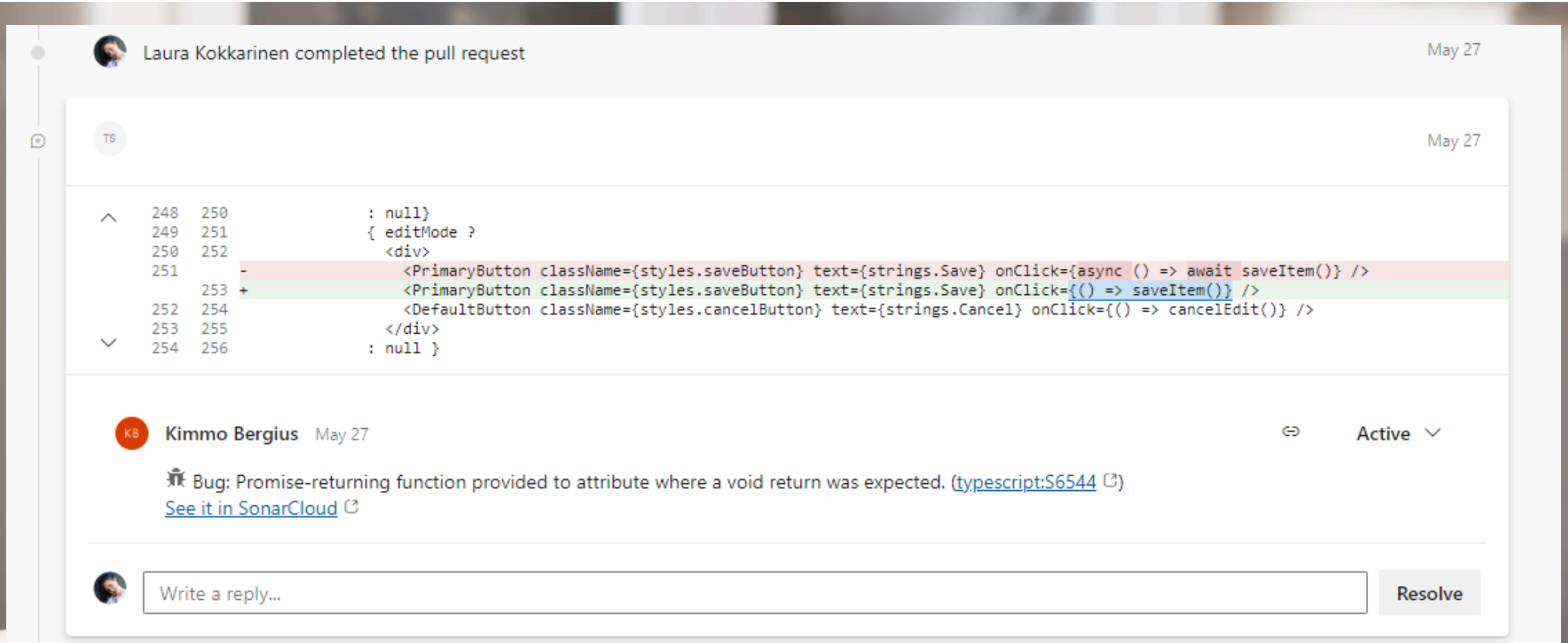
- Many third-party tools available
- SonarCloud
- 10$+/month
- Requires pull request build pipeline configuration



```yaml
trigger: none

pool:
  vmImage: ubuntu-latest

steps:
- checkout: self
  fetchDepth: 0
Settings
- task: NodeTool@0
  inputs:
    versionSpec: '16.x'
  displayName: 'Install Node.js'
Settings
- task: SonarCloudPrepare@1
  inputs:
    SonarCloud: 'SonarCloud'
    organization: 'sulava'
    scannerMode: 'CLI'
    configMode: 'manual'
    cliProjectKey: 'Sulava.Common'
    cliProjectName: 'Sulava.Common'
    cliSources: '.'
- script: |
    npm install
    npm run build
  displayName: 'npm install and build'
Settings
- task: SonarCloudAnalyze@1
Settings
- task: SonarCloudPublish@1
  inputs:
    pollingTimeoutSec: '300'
```

Code scanning

- Require build pipeline execution and resolving comments via branch policies



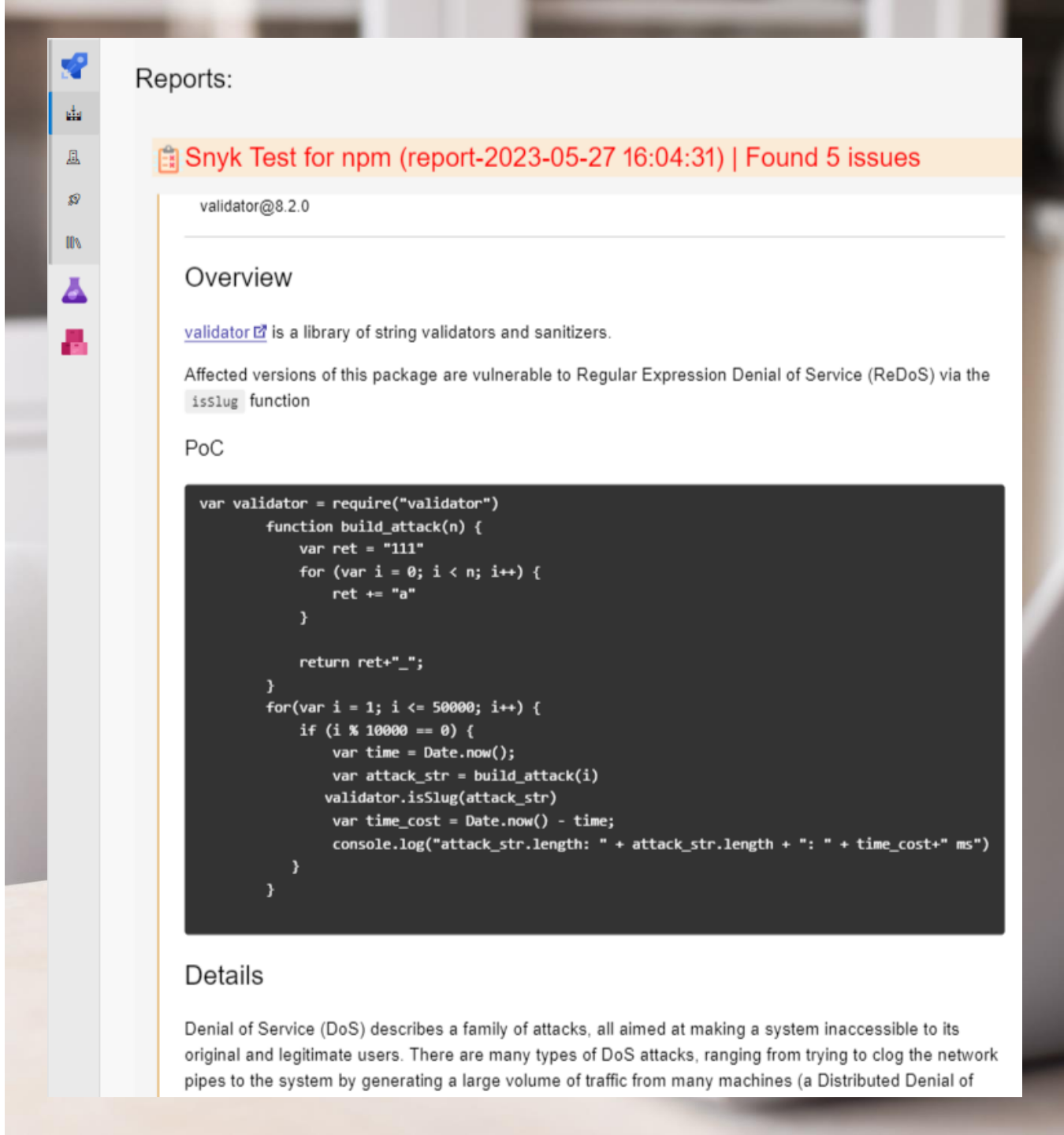
Dependency scanning

- Many third-party tools available
- Snyk (free)
- Requires build pipeline configuration
- Config when to fail the build

```yaml
trigger:
- main

pool:
  vmImage: ubuntu-latest

steps:
Settings
- task: NodeTool@0
  inputs:
    versionSpec: '16.x'
  displayName: 'Install Node.js'

- script: |
    npm install
    npm run build
  displayName: 'npm install and build'

Settings
- task: SnykSecurityScan@1
  inputs:
    serviceConnectionEndpoint: 'Snyk'
    testType: 'app'
    monitorWhen: 'always'
    failOnIssues: true
    additionalArguments: '--fail-on=all'
    ......
Settings
- task: Npm@1
  displayName: 'Publish package to feed'
  continueOnError: true
  inputs:
    command: 'publish'
    publishRegistry: 'useFeed'
    publishFeed: '3236f8cd-1ab2-47cd-9e49-f395927f206b/13954e39-73a7-4799
```

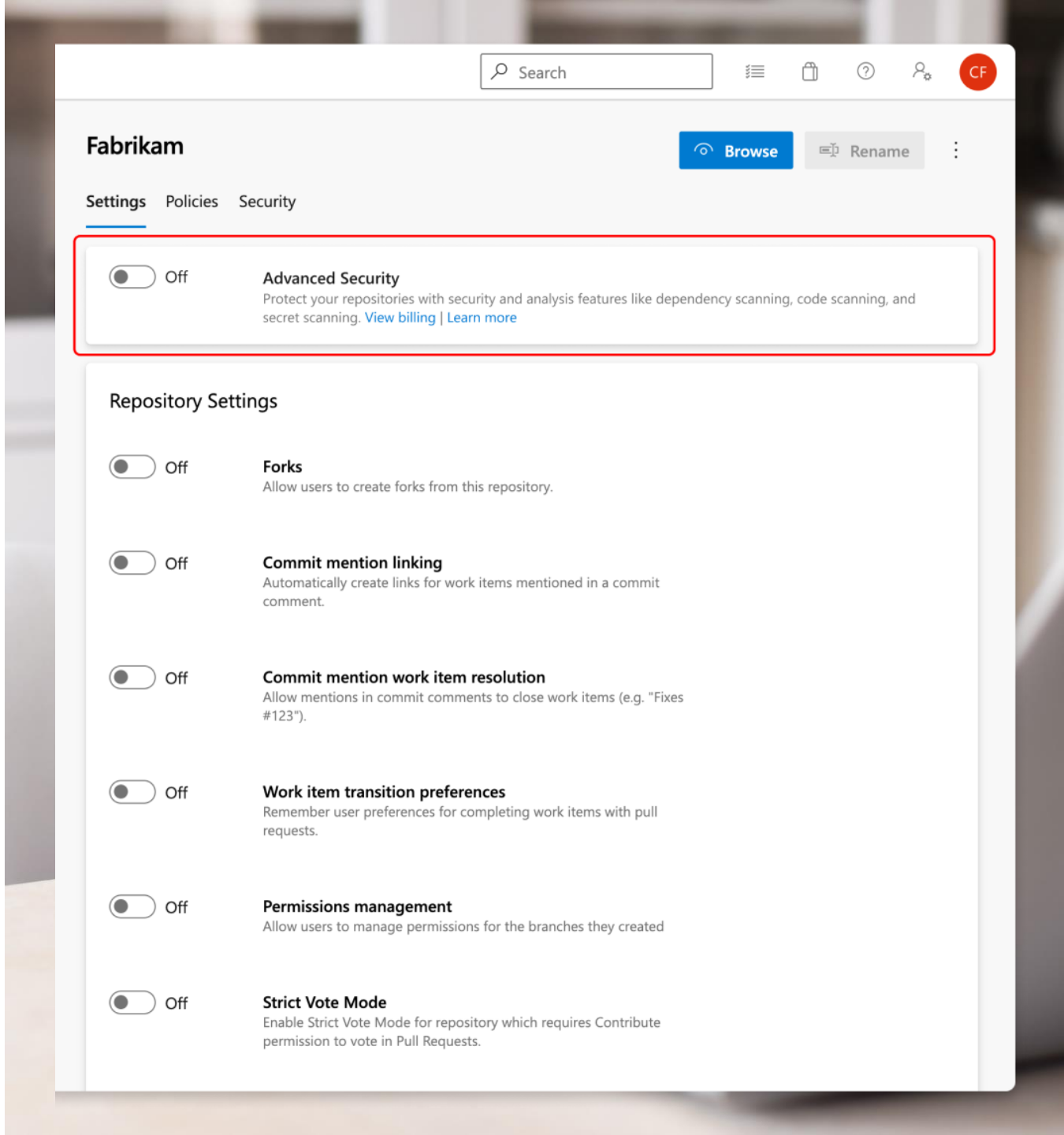
Dependency scanning

- Direct and transitive
- Detailed report
- Sends out email alerts about discovered vulnerabilities



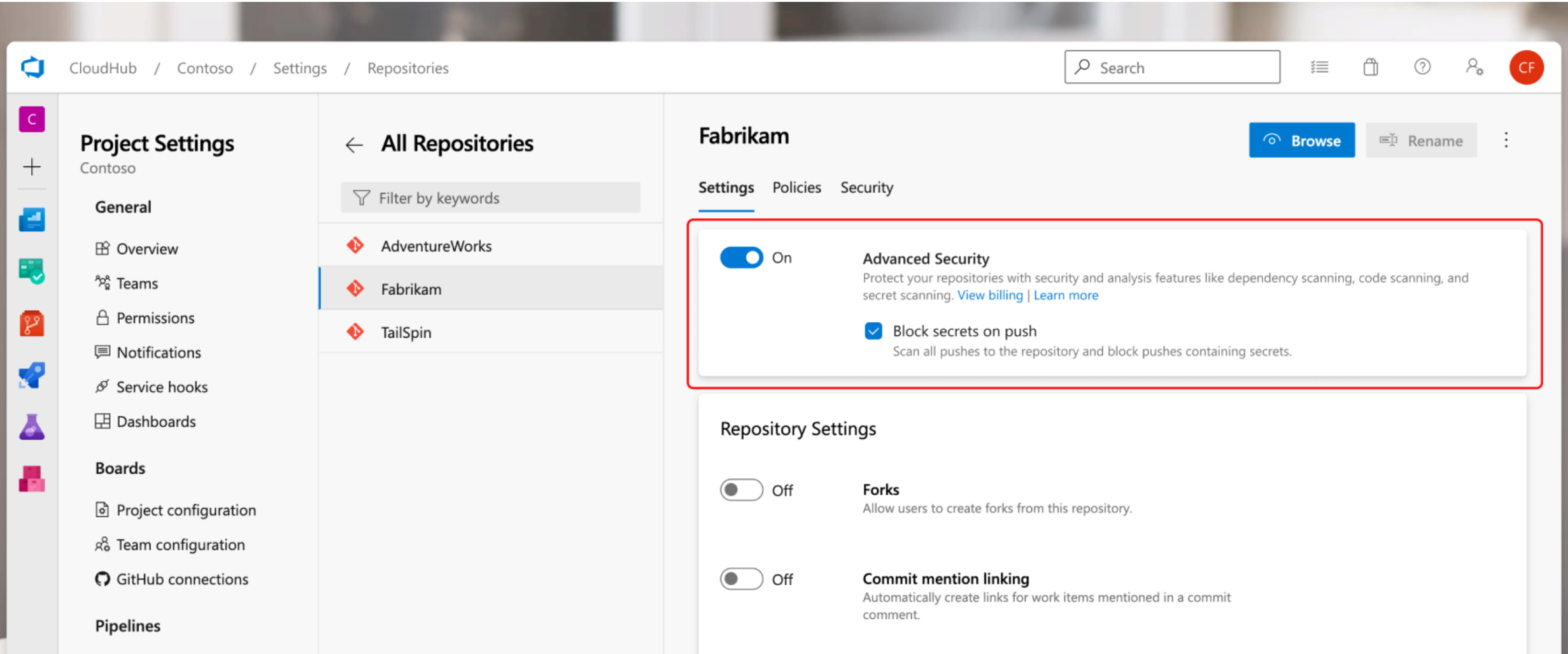
GitHub Advanced Security for Azure DevOps

- GA on September 20, 2023
- Secret, code and dependency scanning
- Enable for individual repositories under Project Settings -> Repositories
- Project Collection Admin



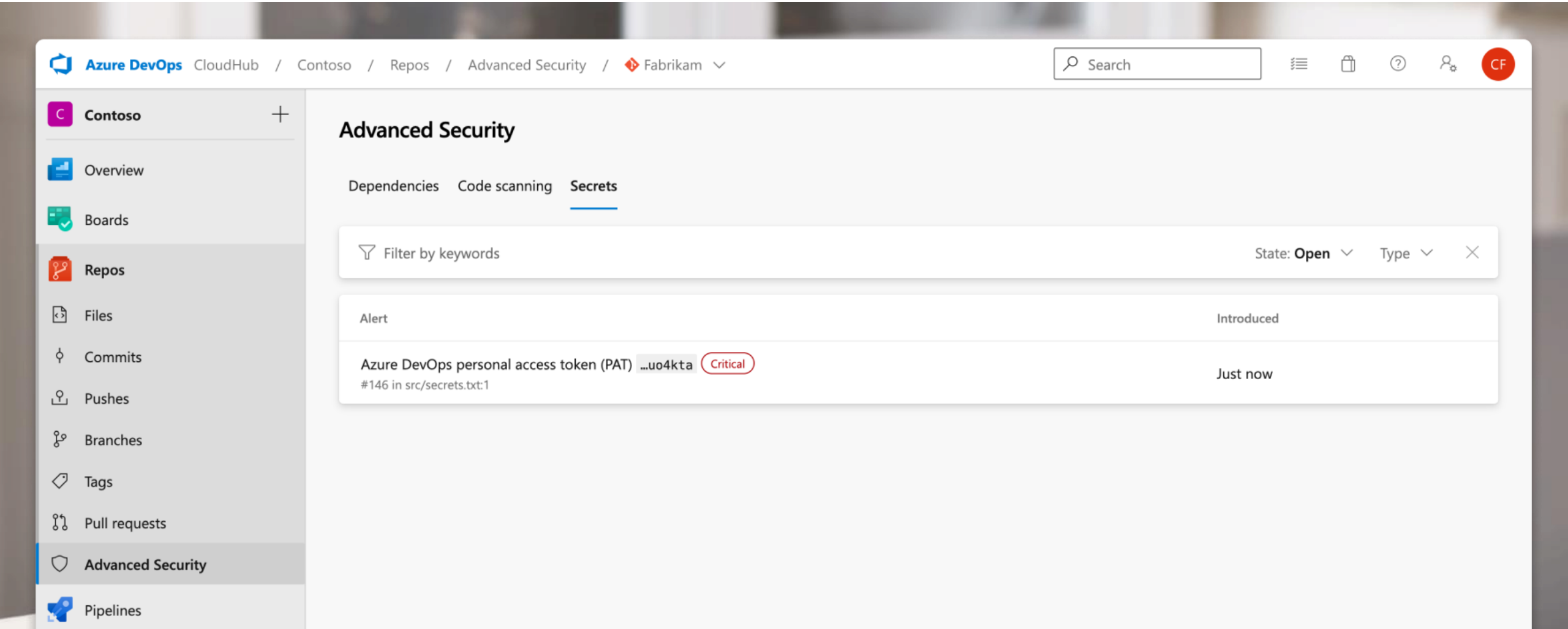
Secret scanning

- Automatically enabled with push protection(!)



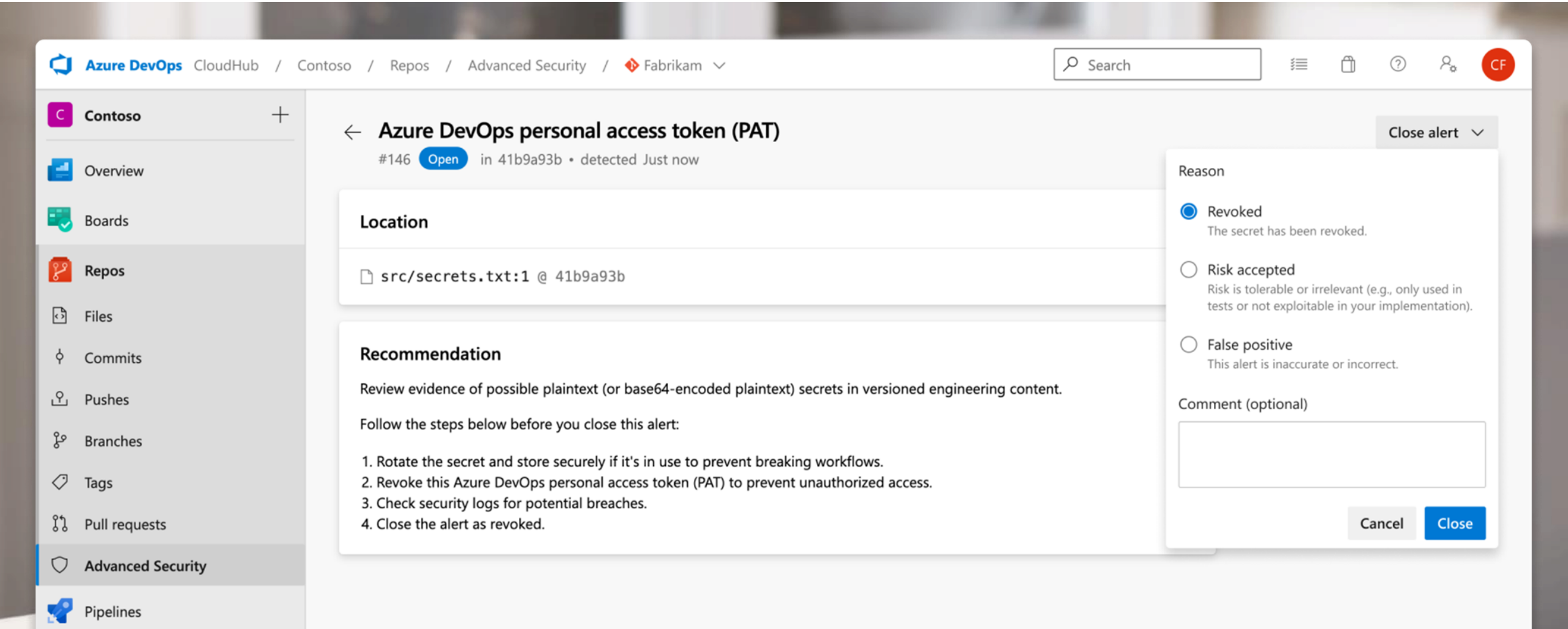
Secret scanning

- GitHub Advanced Security team maintains the default secret scanning patterns



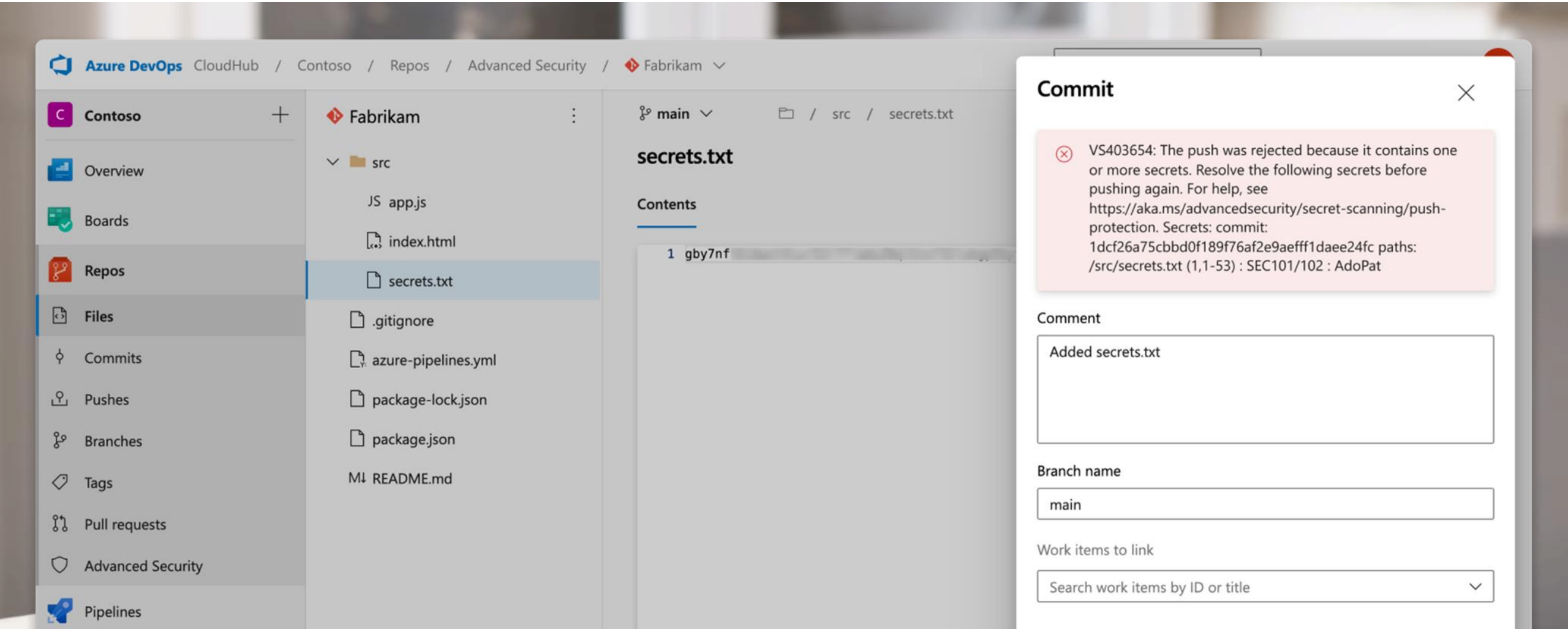
Secret scanning

- Close the alert after revoking the secret, accepting the risk or if false positive



Push protection

- Push protection is available on both the command line and the web interface



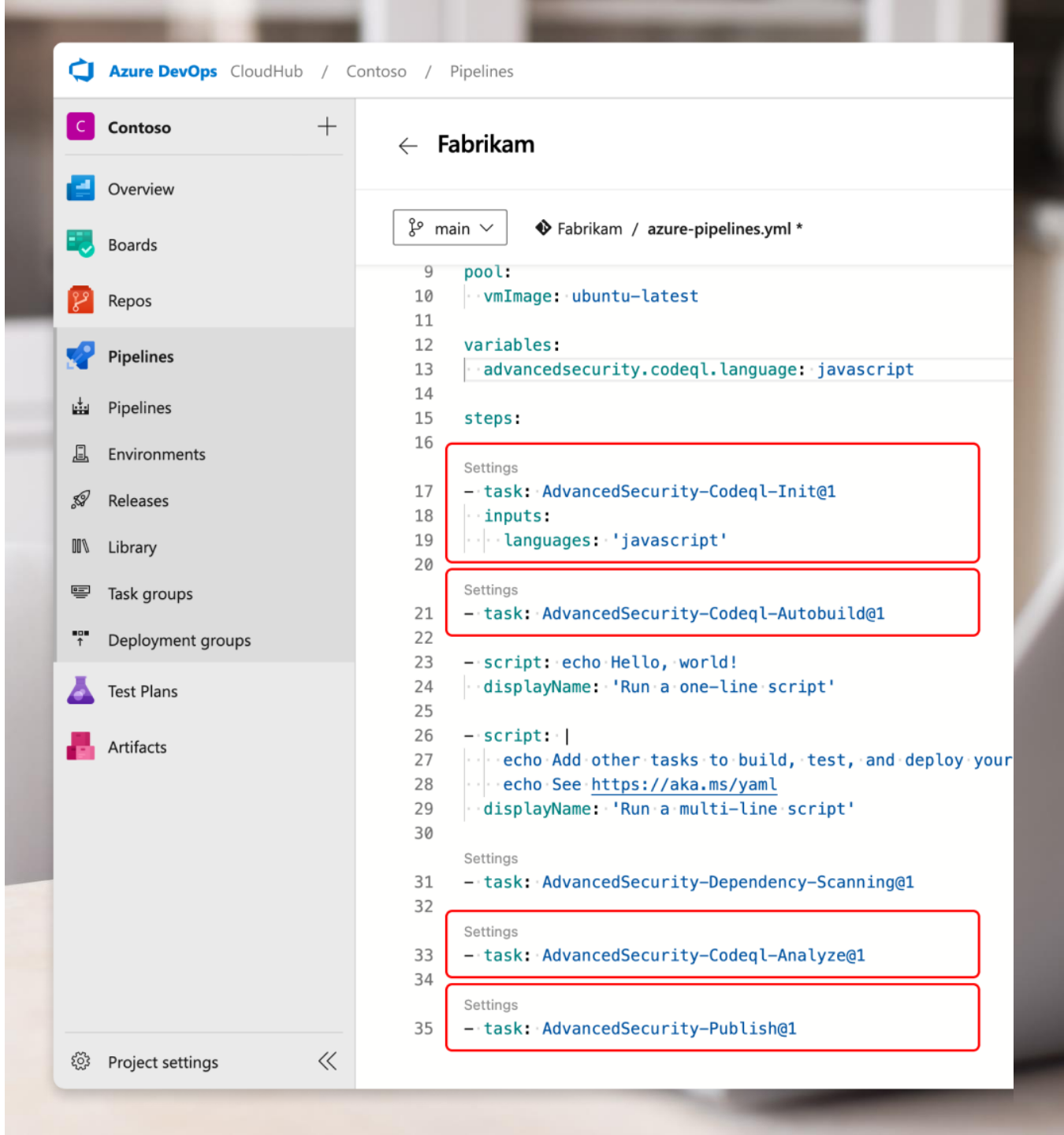
Code scanning

- CodeQL identifies security vulnerabilities
- Code analysis engine
- Queries are open source

- C/C++
- C#
- Go
- Java,
- JavaScript/TypeScript
- Kotlin (beta)
- Python
- Ruby

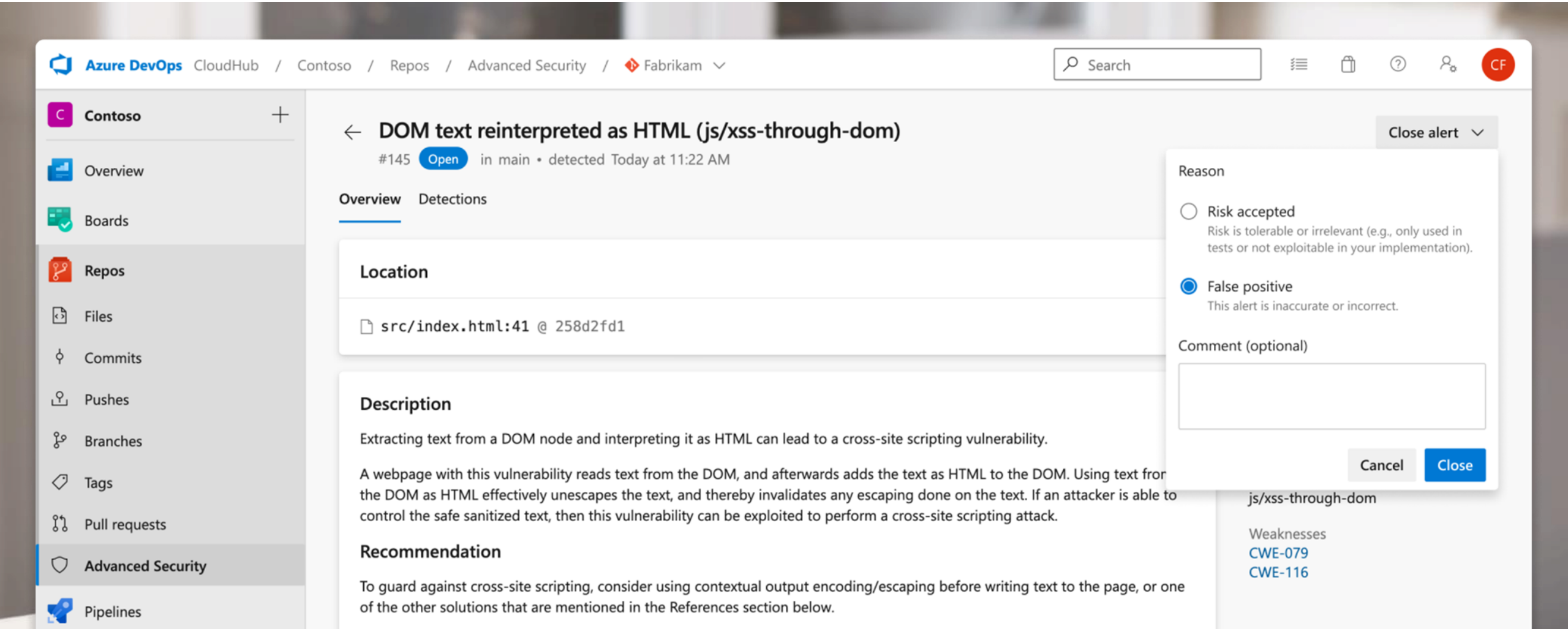
Code scanning

- Add the required actions to the build pipeline
- Pros:
  - Easier to setup than SonarCloud
- Cons:
  - Does not find as many issues (esp. quality)



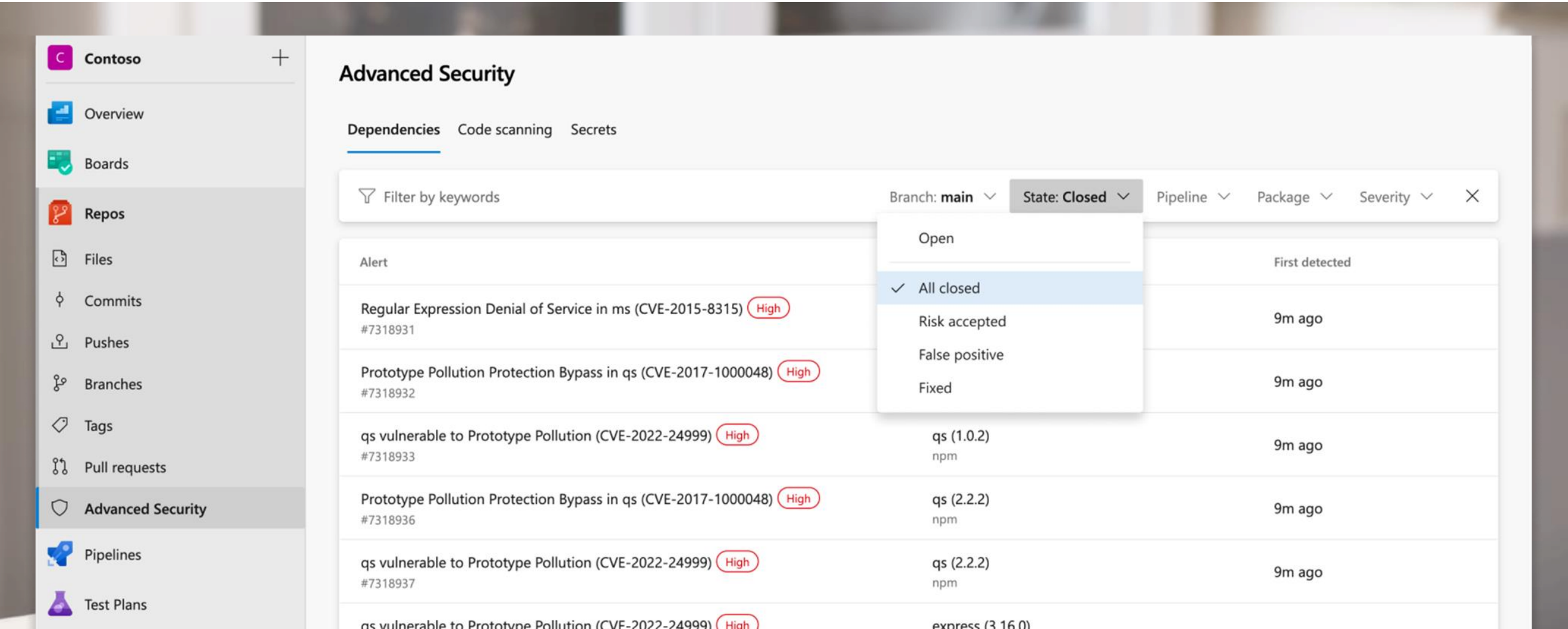
Code scanning

- Automatically closed when no longer detected; can also be accepted or marked as FP



Code scanning

- Closed alerts can be viewed via the filtering options

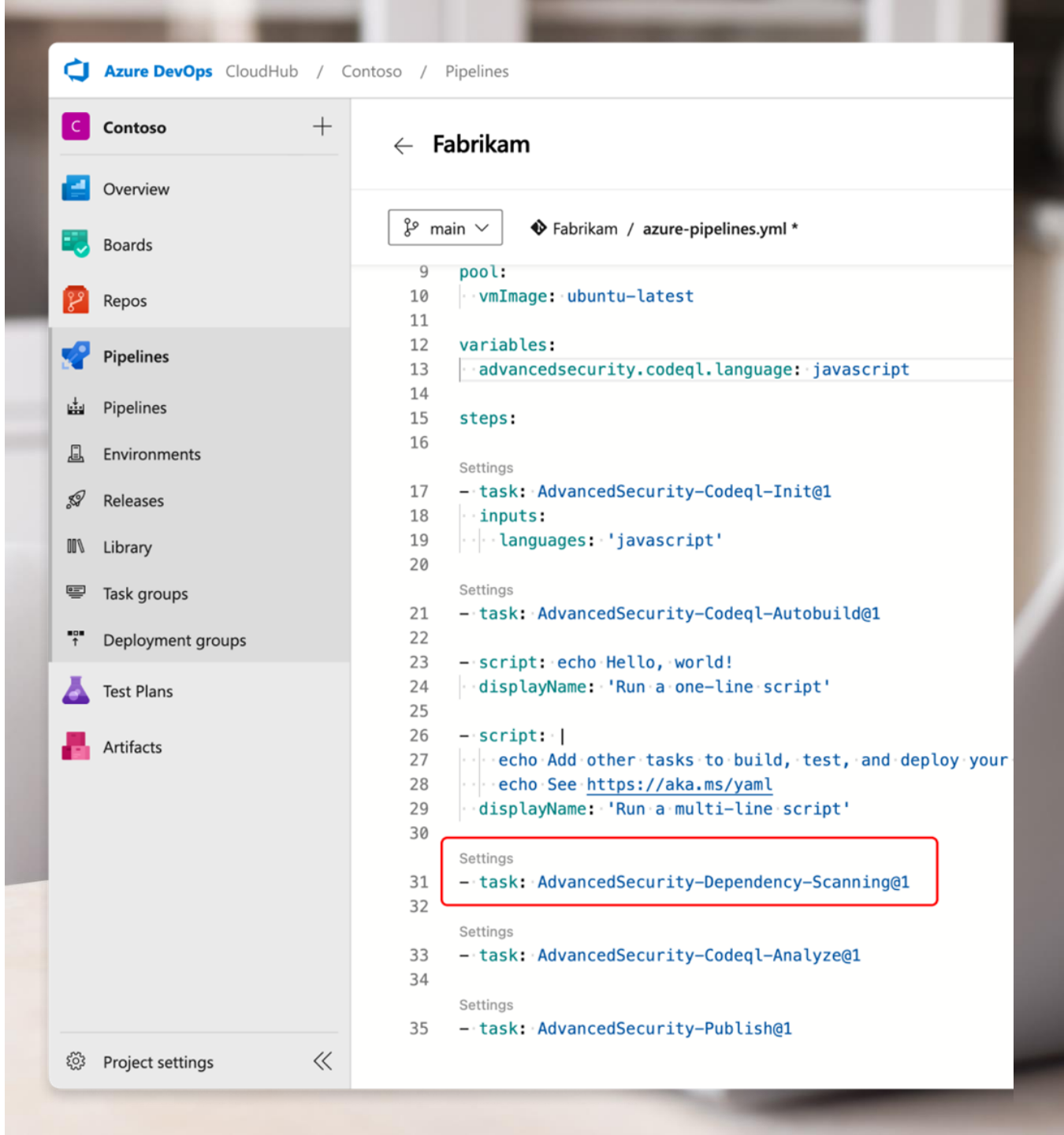


Dependency scanning

- Checks if open-source components used in your source code have associated vulnerabilities
- Direct and transitive
- GitHub Advisory Database

Dependency scanning

- Add the required action to the build pipeline
- Pros:
  - Finds more issues than Snyk
- Cons:
  - No inheritance path
  - No email alerts
  - No break build option



Dependency scanning

- How to override transitive vulnerable “acorn” package in a Node project:

```
{
  "name": "npm-overrides",
  "version": "1.0.0",
  "license": "MIT",
  "dependencies": {
    "axios": "0.19.2",
    "eslint": "5.16.0"
  },
   "overrides":{
      "eslint": {
        "espree": {
          "acorn": "6.4.1“ // patched, non-vulnerable version
        }
    }
  }
}
```

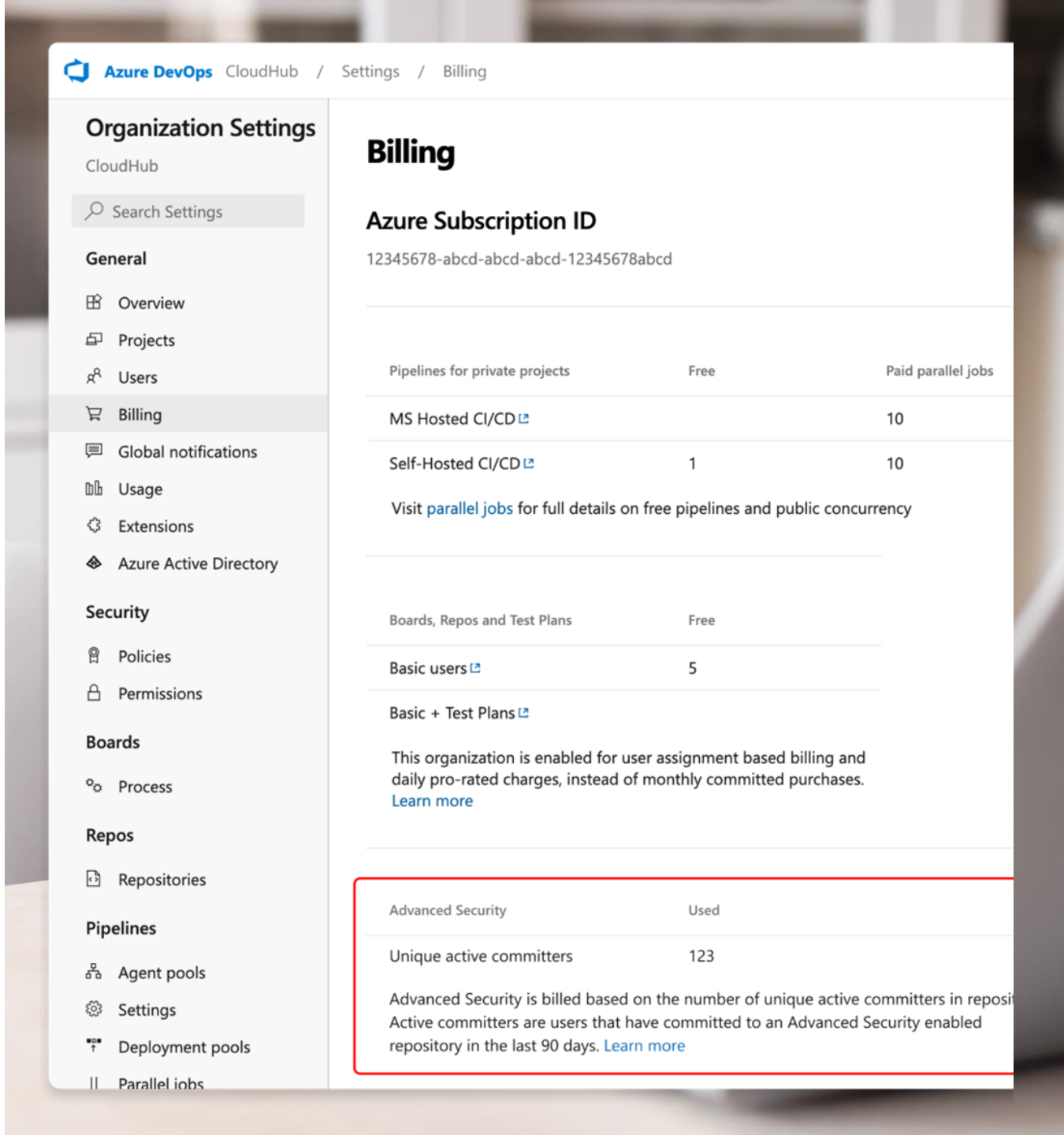
Dependency scanning

- How to override transitive vulnerable “Http.Connections” package in a .NET project:

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>netcoreapp3.1</TargetFramework>
    <RootNamespace>NuGet.Dependencies</RootNamespace>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.App" Version="2.2.8" />
  </ItemGroup>

  <ItemGroup Label="Dependency Resolutions">
    <!-- Microsoft.AspNetCore.App -->
    <PackageReference Include="Microsoft.AspNetCore.Http.Connections"
Version="[1.0.15,1.1.0)" />
  </ItemGroup>
</Project>
```

Price and billing

- 49$ per *active committer* per month
- Billed via the Azure sub connected to the ADO org
- If the same sub has multiple orgs, committers are deduplicated



Which way to go?

- Currently the different third-party tools seem more capable
- Hopefully, GitHub Advanced Security for Azure DevOps will improve
- Ideally, use both
- Costs

Azure DevOps vs GitHub

- GitHub for open source, Azure DevOps for the enterprise
- What about GitHub Enterprise? What is keeping people on Azure DevOps?

Project management and collaboration

- Planning
- Collaboration
- Analytics and reports

Tracing and auditing

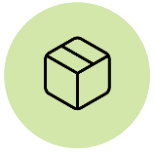
- Work item links

More granular access control management

- Levels
- Features
- Permissions

Flexible licensing

- Stakeholder
- Basic
- Basic + Test Plans



More mature pipelines

- Microsoft
- Release pipeline features

Other features

- Test Plans
- Printable Wikis
- Etc.

Future

- What is the point of having two products for the same purpose?
- Migration from Azure DevOps to GitHub Enterprise – eventually

Recap

Protect against what?

Secure Software Development Lifecycle

Automating code security checks

Protecting source code and repositories

Defender for Cloud integration

What about Azure DevOps?

Questions?