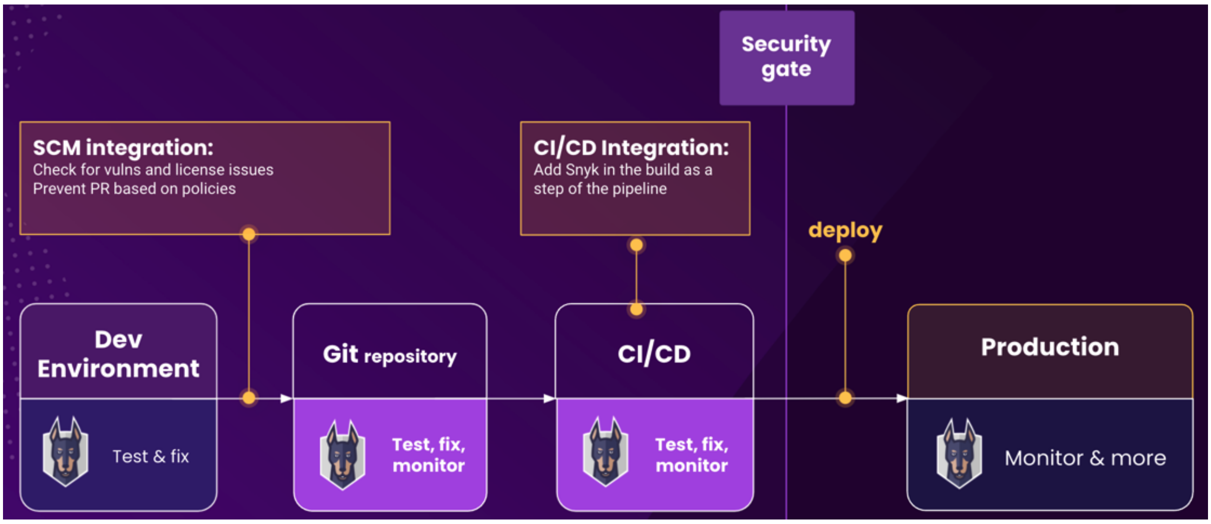
Snyk CI/CD

Snyk can be integrated with AWS CodePipeline to incorporate security testing into your CI/CD pipeline. Snyk is a popular security testing tool that helps identify and fix vulnerabilities in open-source dependencies.



When you decide to use a Snyk CI/CD Integration, typically you will adopt the integration in specific stages. You will also choose a deployment method.

# Typical stages in adopting CI/CD Integration

Stage 1: Expose vulnerabilities (snyk monitor)

⟶ A typical approach is using Snyk results to expose vulnerabilities during the development process. This increases visibility of vulnerabilities among members of your team.

→ When you first implement Snyk in your pipeline, using only the snyk monitor command is recommended. If you use one of the Snyk CI plugins, it is recommended that you configure the plugin to *not* fail the build.

→ This is because all projects have vulnerabilities, and after you set Snyk to fail the build, every build fails because of Snyk. This may cause problems with your team being quickly overwhelmed with failure messages.

→ Using snyk monitor to expose results provides information without disrupting processes.

Stage 2: Use Snyk as a gatekeeper (snyk test)

→ Using Snyk as a gatekeeper prevents the introduction of new vulnerabilities (sometimes known as "stopping the bleeding").
→ After your teams understand the vulnerabilities in their applications, and develop a process for fixing them early in the development cycle, you can configure Snyk to fail your builds, to prevent introducing vulnerabilities into your applications.
→ Add `snyk test` to your build or enable the fail functionality to make Snyk fail your builds, providing the results output to the console. Your developers or DevOps teams can use the results to decide whether to stop or continue the build.

Stage 3: Continuous monitoring (`snyk test` and `monitor`)

→ After you configure Snyk to fail the build when vulnerabilities are detected, you can configure Snyk to send a snapshot of your project's successful builds to Snyk for ongoing monitoring.
→ To do this, configure your pipeline to run `snyk monitor` if your `snyk test` returns a successful exit code.

# Language support for AWS CodePipeline

Snyk integration with AWS CodePipeline is supported for the following languages:

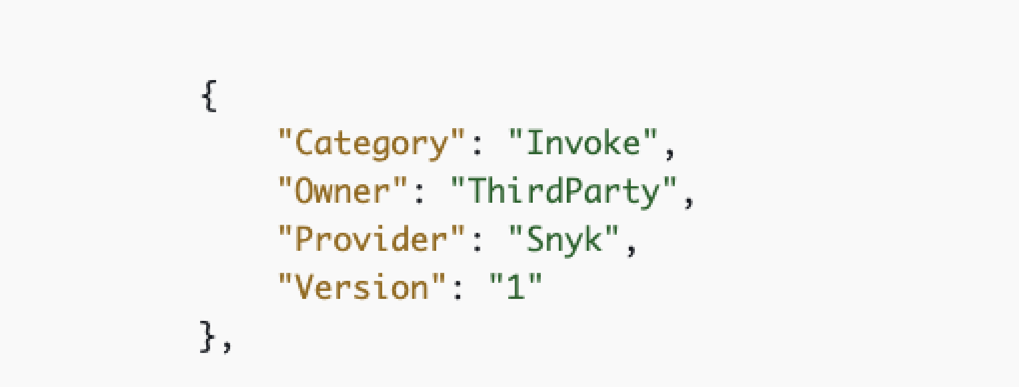
1. JavaScript
2. Java
3. .NET
4. Python
5. Ruby
6. PHP
7. Scala
8. Swift/Objective-C
9. Go

# Snyk action structure reference

The Snyk action in CodePipeline automates detecting and fixing security vulnerabilities in your open source code. You can use Snyk with application source code in your third-party repository, such as GitHub or Bitbucket, or with images for container applications. Your action will scan and report on vulnerability levels and alerts that you configure.

Action type ID

- Category: `Invoke`
- Owner: `ThirdParty`
- Provider: `Snyk`
- Version: `1`



```
{
    "Category": "Invoke",
    "Owner": "ThirdParty",
    "Provider": "Snyk",
    "Version": "1"
},
```