

# セキュリティのシフトレフト

吉田 裕貴 (Yuki Yoshida)

Solutions Architect  
Amazon Web Services Japan G.K.



# 自己紹介

名前: 吉田 裕貴 (よしだ ゆうき)

所属: アマゾンウェブサービスジャパン合同会社  
ISV/SaaS Solutions Architect

好きな技術領域: セキュリティ、運用の効率化

趣味: 筋トレ、バイク、旅、狩猟



# 本日お話すこと

日々の開発の中でよく「セキュリティは重要です」と言われていて、理解もしています。けれど「セキュリティ対策は後回し」になりがちです。

しかし今、生成AIの登場によって状況は一変しています。

本日は、Amazon Q Developer や AWS の AI サービスを活用することで、  
**セキュリティの専門知識が少ない開発者でも、開発のライフサイクルにセキュリティを効果的に組み込む方法**をご紹介します。

# 令和7年上半期 サイバーセキュリティ動向

## ランサムウェアの被害に関する統計

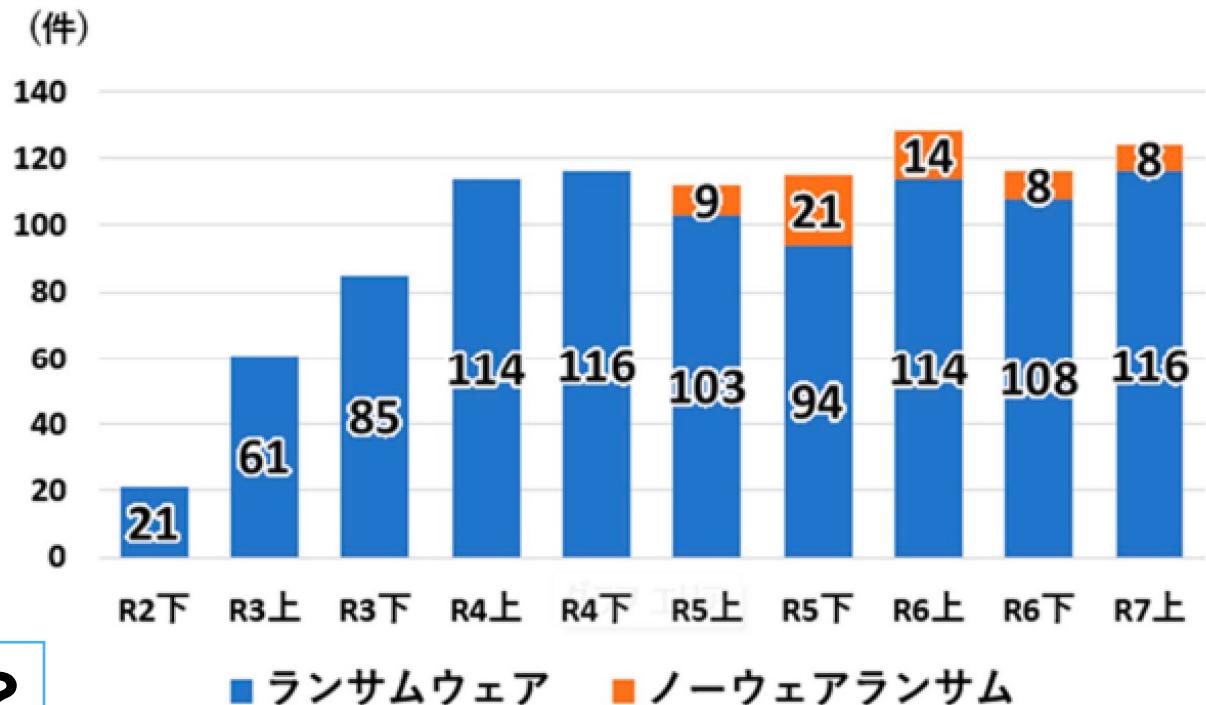
### 1 企業・団体等における被害の報告件数の推移

※ノーウェアランサムの被害については、令和5年上半期から集計。

令和7年上半期のランサムウェア報告件数は124件となっています。

この数値の推移からもセキュリティ対策の継続的な強化が求められる状況が見えてきます。

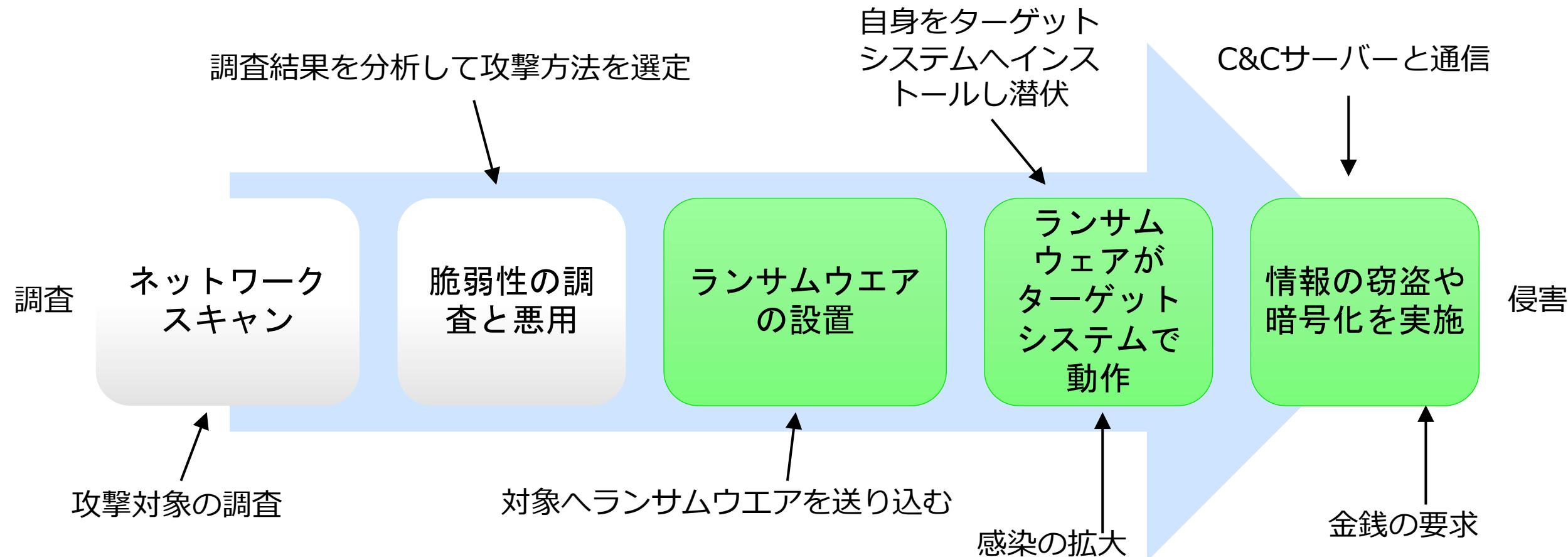
具体的に「何」を「どうやって」？



警視庁 令和7年上半期におけるサイバー空間をめぐる脅威の情勢等について より抜粋

[https://www.npa.go.jp/publications/statistics/cybersecurity/data/R7kami/R07\\_kami\\_cyber\\_jyosei.pdf](https://www.npa.go.jp/publications/statistics/cybersecurity/data/R7kami/R07_kami_cyber_jyosei.pdf)

## 例) ランサムウェア（マルウェア）の感染経路



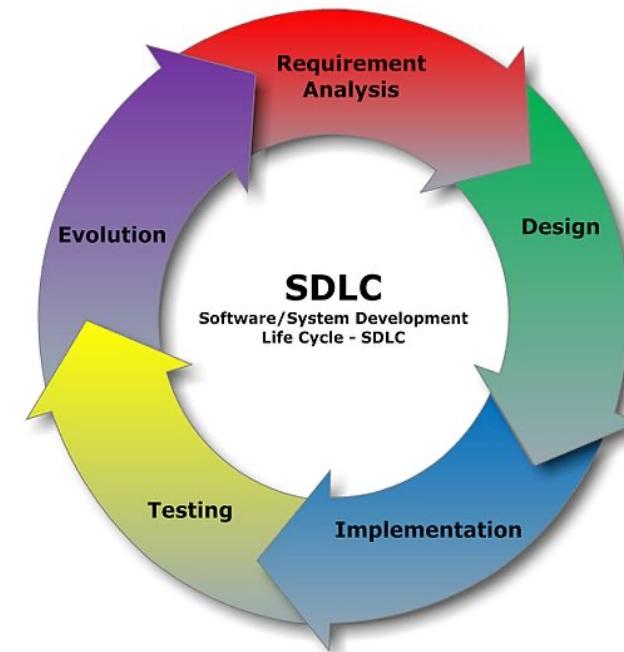
対象を調査し、脆弱な部分を利用して攻撃を仕掛けます。  
つまり、これは**標準的なサイバー攻撃と同様の感染経路**

# SHIFT LEFT - DevSecOps

シフトレフトとは、ソフトウェア開発ライフサイクル（SDLC）の早期段階でセキュリティ上の懸念に対処することを意味します

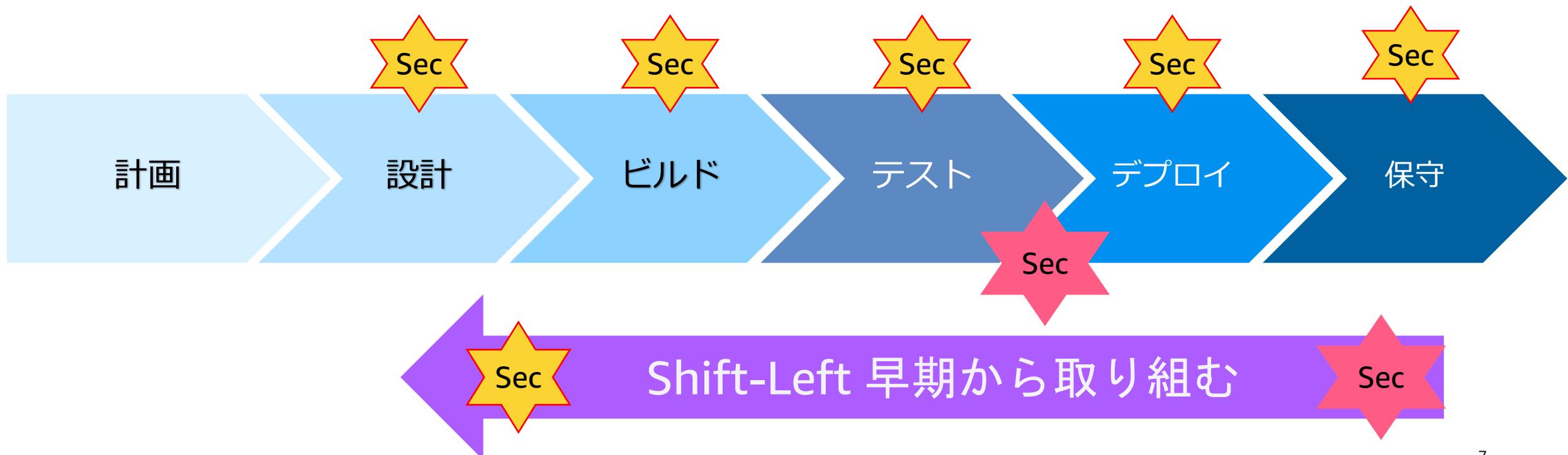
## SDLC の各フェーズ

- Requirement analysis
- Planning
- Software design
- Software development
- Testing
- Deployment



# SHIFT LEFT の価値

開発の早い段階でセキュリティを組み込むことで手戻りにかかる工数を削減する効果やセキュリティホールが見逃されて製品がリリースされるリスクを低減させる効果が期待できます。



# DevSecOps を取り巻く状況の変化

## DevSecOps 登場当時

開発者にセキュリティの専門知識を求める  
ツールが出力する大量のアラートを手動で選別  
セキュアコーディングの学習コストが必要  
**対応難易度が高い**

## 2025年現在

AIと対話しながらリスクの本質を理解  
Coding Agentが脆弱性を検出・修正案を提示  
AI Agent が膨大なログから重要な情報を自動抽出  
専門知識がなくても高品質なセキュリティ対応が可能に

# Amazon Q Developer



開発者と IT 専門家がソフトウェア開発ライフサイクル (SDLC) 全体を通してより迅速に構築できるよう支援

最も精度の高いコーディング推奨事項を提供

エージェントが自律的に機能実装、コードリファクタリング、ソフトウェアアップグレードなどを支援

Amazon Q は AWS の専門家であり、AWS 環境の最適化に精通

最高水準のセキュリティ脆弱性スキャンと修復機能を提供

Amazon Q はセキュリティとプライバシーを最初から考慮して構築されており、組織が生成 AI を安全に利用できるようにします

# SDLC 全体を通した Amazon Q Developer の活用



## 調査

- 新規プロジェクトへより迅速に参加
- 新機能の計画立案
- AWS API の使用方法の理解
- 社内コードベースに関する質問



## ビルド

- 対話型コーディングアシスタント
- ソフトウェア開発
- 対話型コーディング



## テスト

- ユニットテスト生成によるテストカバレッジの向上
- セキュリティ脆弱性のスキャンと修復



## デプロイ

- コードレビューの自動化
- デプロイリスクの評価
- ドキュメントの生成

# SDLC 全体を通した ツールの活用

設計

ビルド

テスト

デプロイ&保守

- SAST
- Secrets Detection
- Code Quality

- Test Code
- Document Maintenance



Amazon Q Developers

- SCA



AWS CodeBuild  
+ 3<sup>rd</sup> party



Amazon Inspector

- Container scans



Elastic Container Service (ECR)

- Monitoring & logging
- Synthetic tests



Amazon CloudWatch



AWS CodeBuild  
+ 3<sup>rd</sup> party

- DAST



AWS CodeBuild  
+ 3<sup>rd</sup> party

- Monitoring & logging
- Synthetic tests



Amazon CloudWatch



AWS CodeBuild  
+ 3<sup>rd</sup> party



Amazon Inspector



# 参考: セキュリティ用語説明

## SAST

Static Application  
Security Testing

ソースコードを静的に解析してセキュリティ脆弱性を検出する手法

## DAST

Dynamic Application  
Security Testing

実行中のアプリケーションに対して外部からテストを行い、セキュリティ脆弱性を検出する手法

## SCA

Software Composition  
Analysis

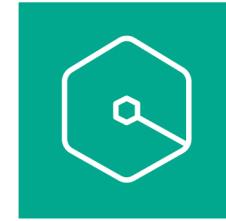
アプリケーションで使用されているオープンソースコンポーネントやサードパーティライブラリを分析し、セキュリティ脆弱性やライセンス問題を検出する手法

# セキュリティトレーニング

AIを利用してすることで開発者にセキュリティを取り込んだ開発体験を



# 調査・設計を手助けする



Amazon Q Developers

- 利用するOSSのモジュールはセキュリティ的に安全か
  - すべてのモジュールのコード調査は人力で可能か
  - 生成AIによる静的なコード解析
  - 脅威モデリングの実施

「ビルダーのための脅威モデリング」で  
ご紹介した内容になります



# 設計

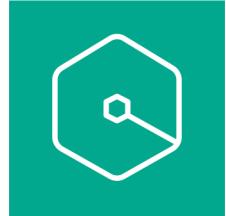
The screenshot shows the AWS Console Home page with a dark theme. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar, and account information for 'Oregon'. Below the navigation is a 'Recently visited' section with links to IAM, Lambda, and DynamoDB. To the right is a 'Welcome to AWS' section featuring 'Getting started with AWS' (with a rocket icon), 'What's new with AWS?' (with a lightbulb icon), and 'Training and certification' (with a diploma icon). Further down is an 'Explore AWS' section with links to 'Build Web and Mobile Apps Faster', 'Amazon Bedrock is Now Gener...', 'AWS Support' (with a shield icon), and 'Calling All Java and Python De...'. On the far right, there are sections for 'Build a solution' (with icons for launching a virtual machine, migrating to AWS, registering a domain, and hosting a static web app) and 'Start a development project' (with icons for starting a project and building a SQL server). The bottom of the page includes links for CloudShell, Feedback, and various legal notices.

AWS の 17 年分  
の知識、ベストプ  
ラクティス、お  
よび Well-  
Architected のア  
ドバイスに基づい  
て質問とガイダン  
スを取得

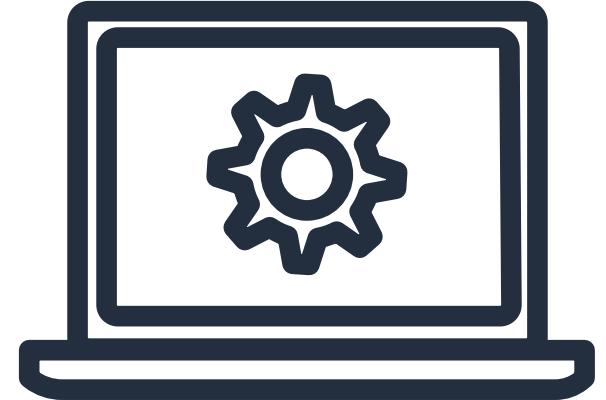
お客様のデータや  
システムなどを活  
用して、ビジネス  
に特化したアプリ  
ケーションを作成



# ビルドを手助けする

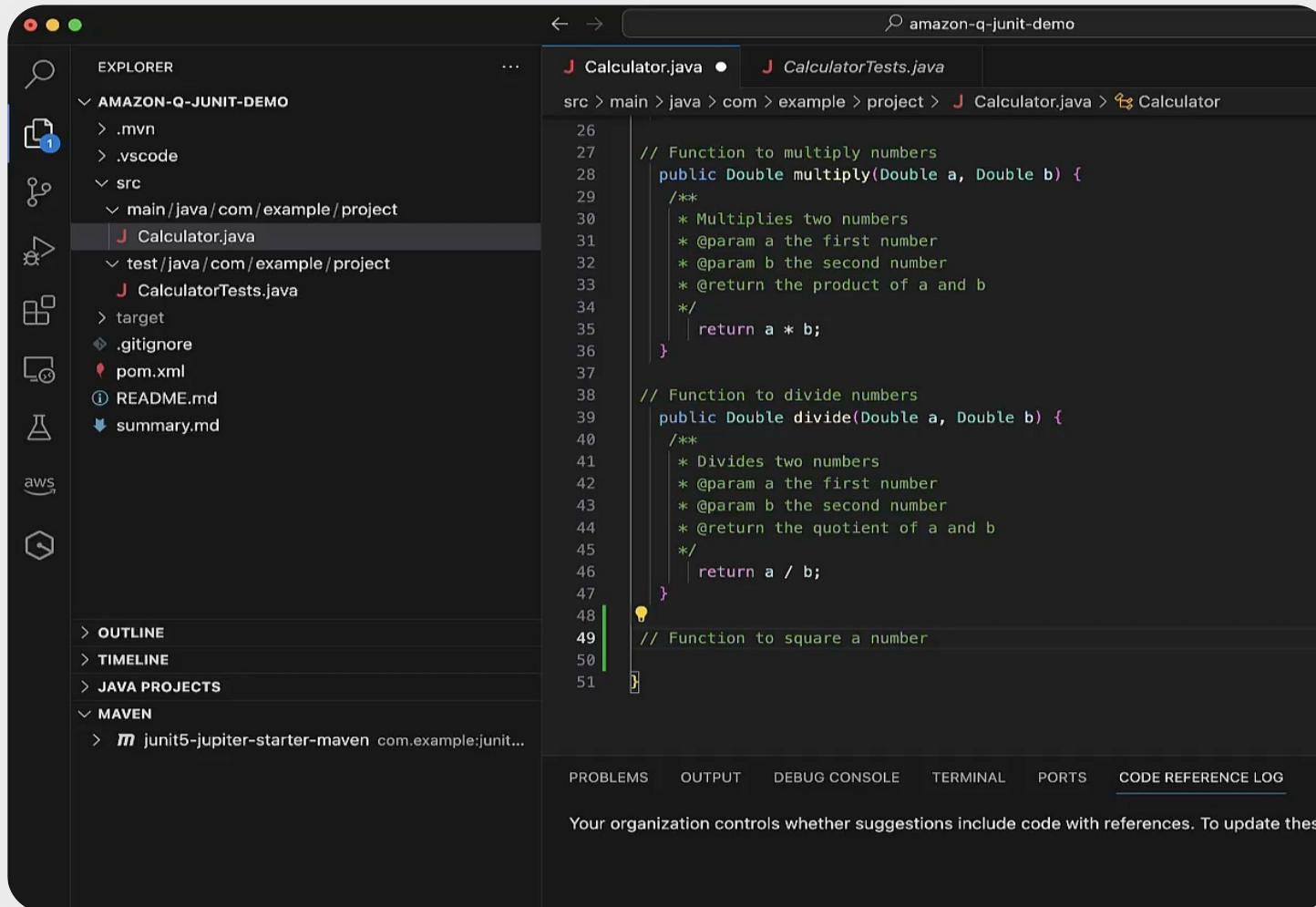


Amazon Q Developers



- セキュアコーディングの手助け
  - Ex.ハードコードされたパスワードの検出
  - プロジェクト及びコード記述時のスキャンと修正の提案

# ビルレド



The screenshot shows the AWS Toolkit for VS Code interface. The Explorer sidebar on the left displays the project structure for 'AMAZON-Q-JUNIT-DEMO', including '.mvn', '.vscode', 'src' (containing 'Calculator.java' and 'CalculatorTests.java'), 'target', '.gitignore', 'pom.xml', 'README.md', and 'summary.md'. The 'OUTLINE' and 'TIMELINE' sections are also visible. The main editor area shows Java code for a calculator project:

```
26 // Function to multiply numbers
27 public Double multiply(Double a, Double b) {
28     /**
29      * Multiplies two numbers
30      * @param a the first number
31      * @param b the second number
32      * @return the product of a and b
33      */
34     return a * b;
35 }
36
37 // Function to divide numbers
38 public Double divide(Double a, Double b) {
39     /**
40      * Divides two numbers
41      * @param a the first number
42      * @param b the second number
43      * @return the quotient of a and b
44      */
45     return a / b;
46 }
47
48 // Function to square a number
49
50 }
```

The bottom navigation bar includes 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'CODE REFERENCE LOG'. A note at the bottom states: 'Your organization controls whether suggestions include code with references. To update these'.

コードの生成

コードの説明

インラインチャットでの対話

コードベースの理解支援

コードに合わせたカスタマイズ



# テストを手助けする



Amazon Q Developers



- 生成 AI によるユニットテスト作成
- 生成AIによるテスト用モックデータの準備
- テスト結果のサマリやセキュリティとコード品質を向上させる修復策の生成

# テスト

The screenshot shows a dark-themed IDE interface with the following details:

- EXPLORER** panel on the left lists the project structure:
  - Java-Security-Findings
  - demo
  - src
  - main
  - java/com/example
  - Main.java (selected)
  - resources
  - test
  - target
  - pom.xml
- CODE REFERENCE LOG** tab is selected in the bottom navigation bar.
- PROBLEMS**, **OUTPUT**, **DEBUG CONSOLE**, **TERMINAL**, **PORTS** tabs are also present in the bottom navigation bar.
- MAIN.JAVA** tab is active in the top center.
- The code editor contains the following Java code:

```
1 package com.example;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import javax.servlet.http.HttpServletRequest;
6
7 public class Main {
8
9     public void createSqlConnection(String url) throws Exception {
10         final Connection connection = DriverManager.getConnection(url, "username", "password");
11         connection.close();
12     }
13
14     public void executeSqlStatement(HttpServletRequest request, java.sql.Connection connection) {
15         final String favoriteColor = request.getParameter("favoriteColor");
16         try {
17             String sql = "SELECT * FROM people WHERE favorite_color='" + favoriteColor + "'";
18             java.sql.Statement statement = connection.createStatement();
19             statement.execute(sql);
20         } catch (java.sql.SQLException e) {
21             throw new RuntimeException(e);
22         }
23     }
24 }
25 }
```

- The status bar at the bottom shows: AWS: 2 Connections, CodeWhisperer, Java: Ready, Ln 25, Col 2, Spaces: 4, UTF-8, LF, Java.

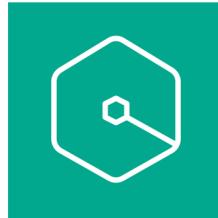
生成 AI によるユニット  
テスト作成

セキュリティ脆弱性の  
プロジェクト全体ス  
キャン

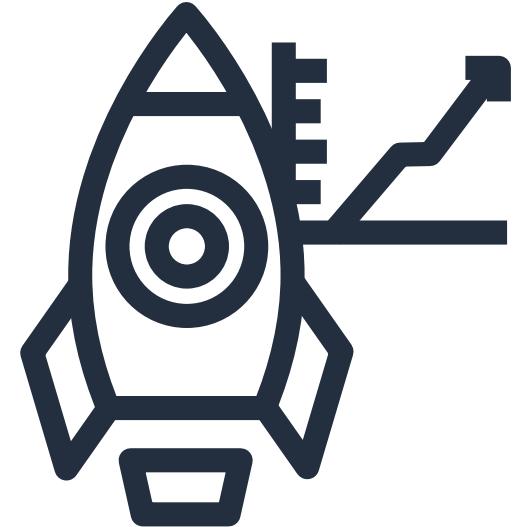
セキュリティとコード  
品質を向上させる修復  
策の生成



デプロイを手助けする

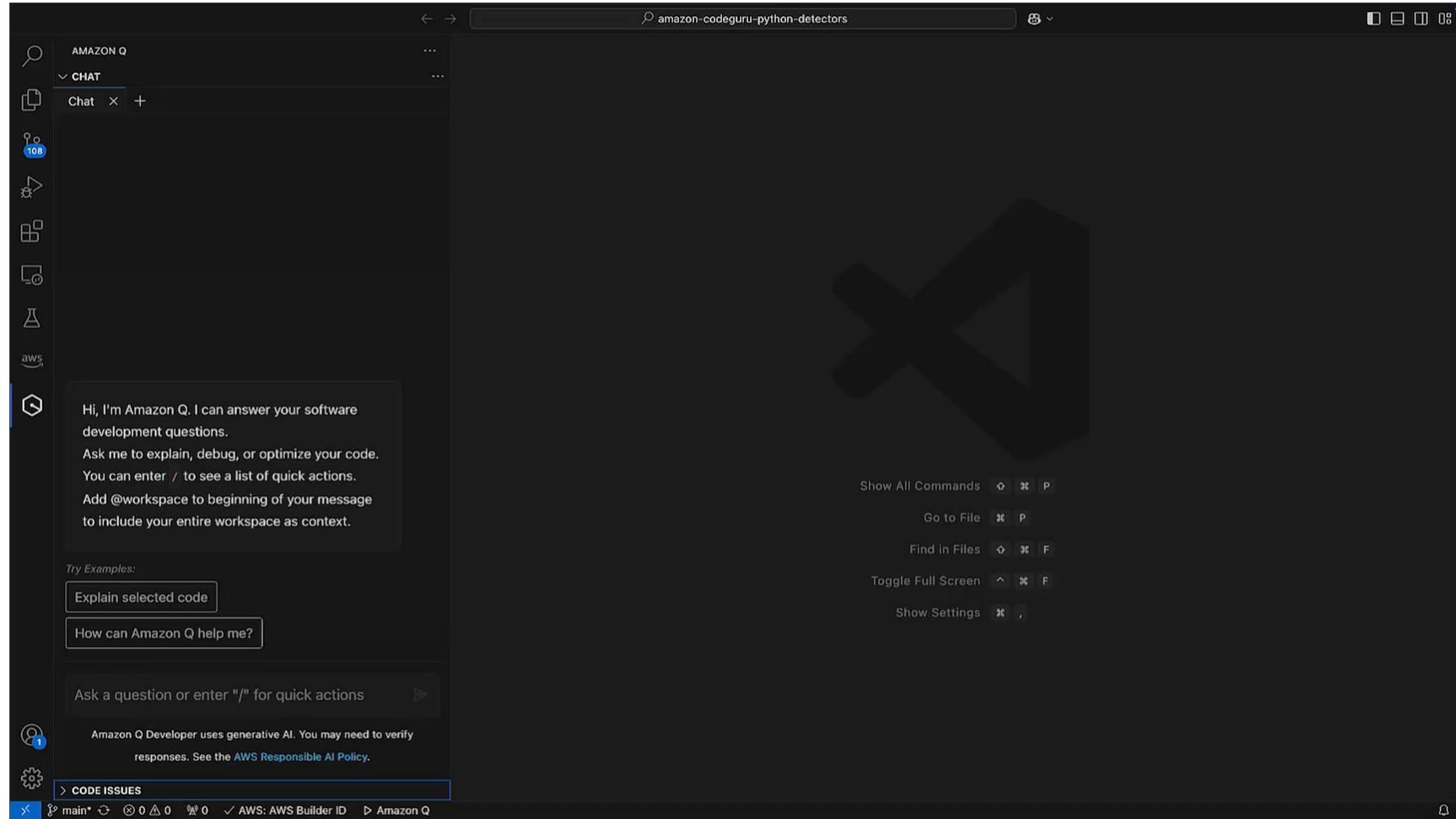


Amazon Q Developers



- デプロイ時にAIによるコードレビュー実施
- 属人化しない一貫したコードレビュー品質の担保

# デプロイ



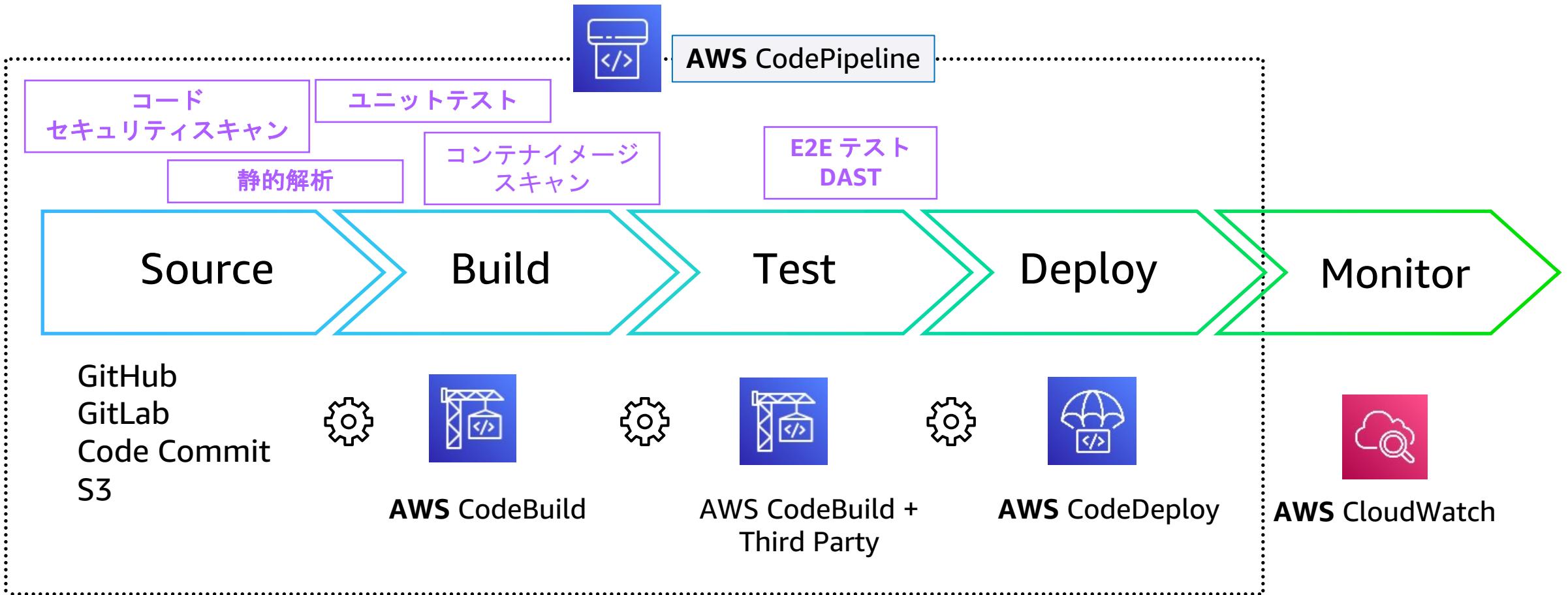
るコードレビューの自動化

一般的なコード品質の問題を数時間ではなく数分で自動的に検出し解決

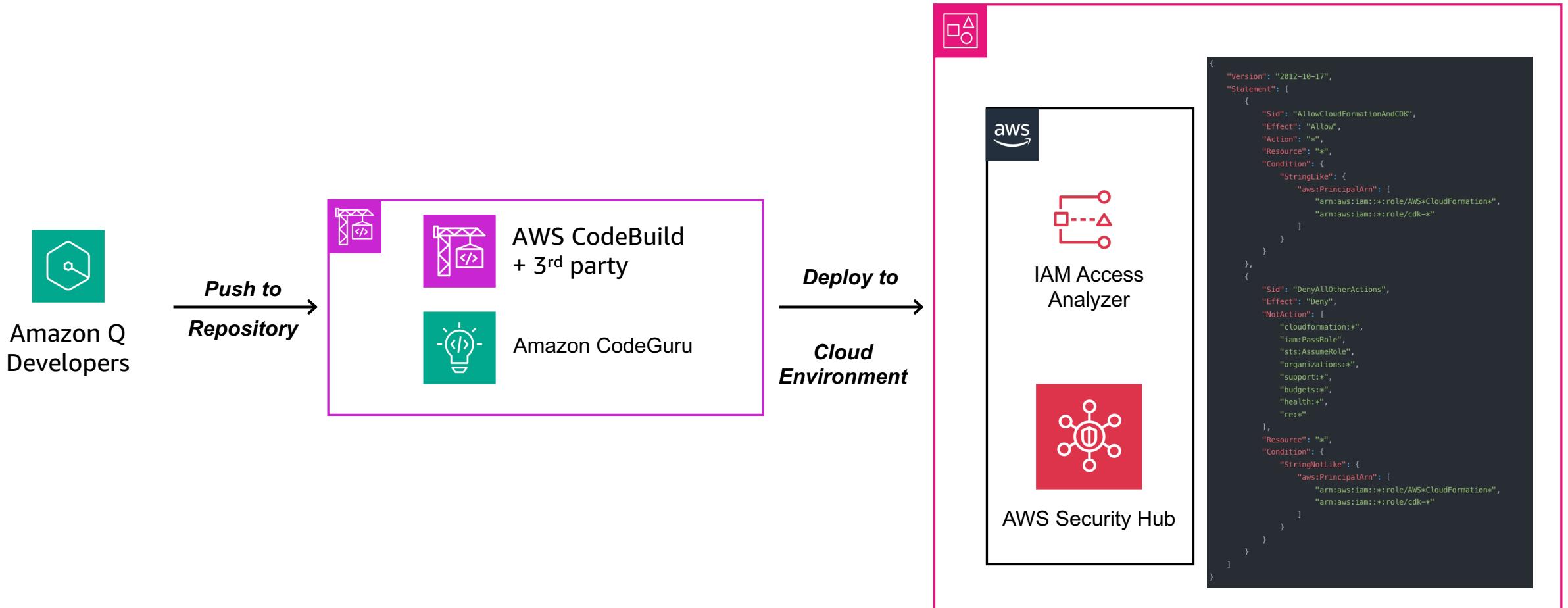
一貫したコードレビューの確保



# デプロイパイプラインとの統合



# デプロイパイプラインを通して IAM の権限を統制する



# Thank you!

