**AWS CodeGuru Security Integration with GitHub: Step-by-Step Lab Guide**

**Lab Overview**

This lab guide will walk you through integrating AWS CodeGuru Security with GitHub using an IAM role and GitHub Actions. By the end of this lab, you will have an automated security scanning workflow that runs every time you push code to the main branch.

**Prerequisites**

* An **AWS account** with administrative access.
* An **AWS IAM user** with permissions to create IAM roles and CloudFormation stacks.
* A **GitHub account** with repository admin permissions.
* AWS CLI installed and configured.
* Git installed on your local machine.

**Step 1: Create an IAM Role**

**Option 1: Using AWS CloudFormation**

1. **Open the AWS Console** and navigate to **AWS CodeGuru Security**.
2. Click on **Integrations** and select **Integrate with GitHub**.
3. Under **Step 1: Create an IAM role**, choose **Use CloudFormation template**.
4. Click **Open template in CloudFormation** to launch the AWS CloudFormation console.
5. In the **Create stack** page:
   * Enter a unique **Stack Name** (e.g., CodeGuruSecurityGitHubStack).
   * Provide your **GitHub repository name**.
   * Check the box to acknowledge that IAM resources may be created.
   * Click **Create stack**.
6. Wait for the stack to be created. AWS will generate a role named CodeGuruSecurityGitHubAccessRole.

**Option 2: Manually Creating an IAM Role**

1. **Navigate to IAM** in the AWS Console.
2. Click **Roles** → **Create role**.
3. Select **Web Identity** and choose **GitHub OIDC provider**.
4. Choose **Allow programmatic access**.
5. Attach the **AWS managed policy** AmazonCodeGuruSecurityGitHubIntegration.
6. Click **Next**, enter a role name (e.g., CodeGuruSecurityGitHubAccessRole), and create the role.
7. Copy the **role ARN** for later use.

**Step 2: Create a Custom GitHub Workflow**

1. **Log in to GitHub** and open the repository you want to scan.
2. Click on the **Actions** tab.
3. Click **New workflow**, then choose **set up a workflow yourself**.
4. Replace the contents of .github/workflows/main.yml with the following code:

name: CodeGuru Security Example

on:

push:

branches:

- 'main'

permissions:

id-token: write

security-events: write

actions: read

contents: read

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout Repository

uses: actions/checkout@v3

with:

fetch-depth: 0

- name: Configure AWS Credentials

uses: aws-actions/configure-aws-credentials@v2

with:

role-to-assume: arn:aws:iam::ACCOUNT\_ID:role/CodeGuruSecurityGitHubAccessRole

aws-region: AWS\_REGION

role-session-name: GitHubActionScript

- name: Run CodeGuru Security Scan

uses: aws-actions/codeguru-security@v1

with:

source\_path: .

aws\_region: AWS\_REGION

fail\_on\_severity: Critical

- name: Print Findings

run: |

ls -l

cat codeguru-security-results.sarif.json

- name: Upload Result

uses: github/codeql-action/upload-sarif@v2

with:

sarif\_file: codeguru-security-results.sarif.json

1. Replace ACCOUNT\_ID with your AWS account ID.
2. Replace AWS\_REGION with your AWS region (e.g., us-east-1).
3. Click **Commit changes** to save the workflow.

**Step 3: Run Scans and Address Findings**

1. Push new code changes to the **main** branch.
2. AWS CodeGuru Security will automatically scan your repository.
3. View findings:
   * **In GitHub**: Go to the **Security** tab → **Code scanning alerts**.
   * **In AWS Console**: Navigate to **AWS CodeGuru Security** → **Findings**.
4. Address any issues based on the suggested remediations.
5. Push the updated code and verify that the vulnerabilities have been resolved.