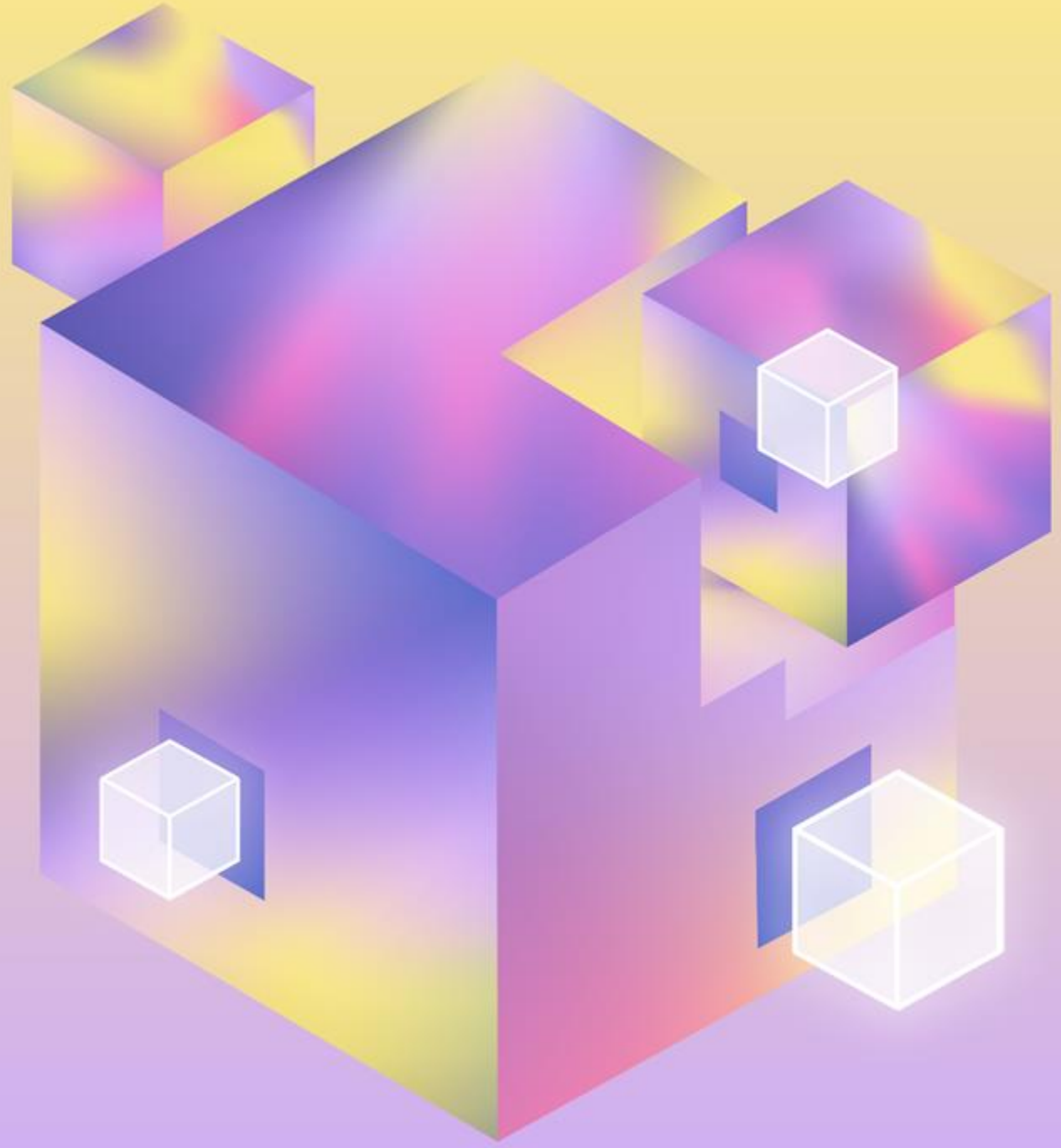
aws

Builders Online Series

31 JULY, 2025 | APJ



AWS Builders Online Series

Secure by design: Building AWS applications with security practices

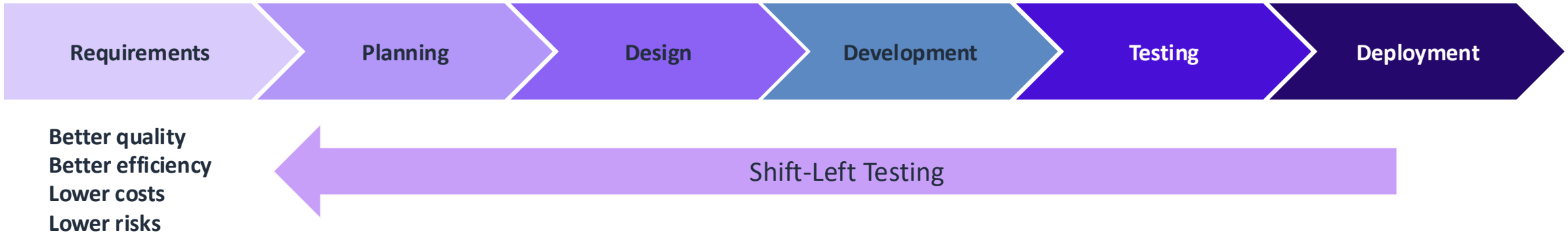
Sam Zhang
Security Specialist Technical Account Manager
AWS

Agenda

- Introduction of DevSecOps
- Serverless Security Essentials
- Security in Infrastructure as Code (AWS CDK)
- DevSecOps implementation powered by gen AI

Introduction of DevSecOps

A new approach is required



Governance changes

- Responsibilities shifted left
- Fast decision making
- Self-enabled teams
- Switch to continuous assurance
- Real-time Observability

People, Process, Technology changes

- Cultural changes
- Sec training and education
- High degree of automation
- Seamless CI/CD pipeline integration
- Security and compliance as code

Business benefits of DevSecOps

- Increased revenue
- Cost savings
- Improved brand reputation
- Improved compliance
- Competitive advantage
- Customer trust
- Speed and agility

45%
Faster Vulnerability Remediation

20-50%
Operational Cost Savings

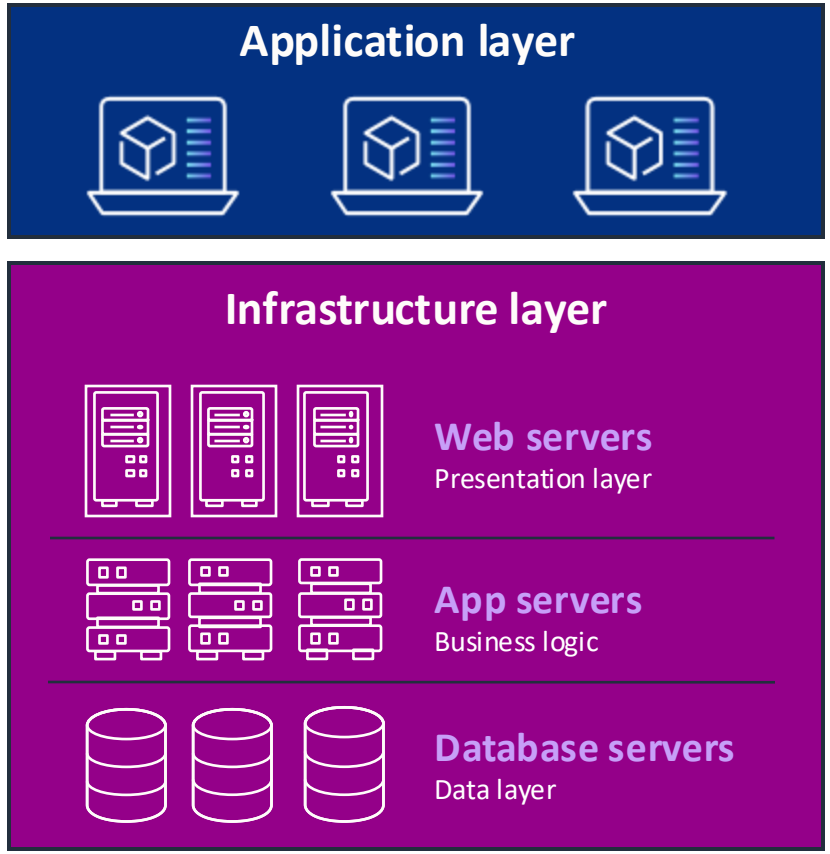
2.5x
Likelihood to Exceed Goals

2X
Recruitment Retention

Source: [Accelerate: Building and Scaling High Performing Technology Organizations]

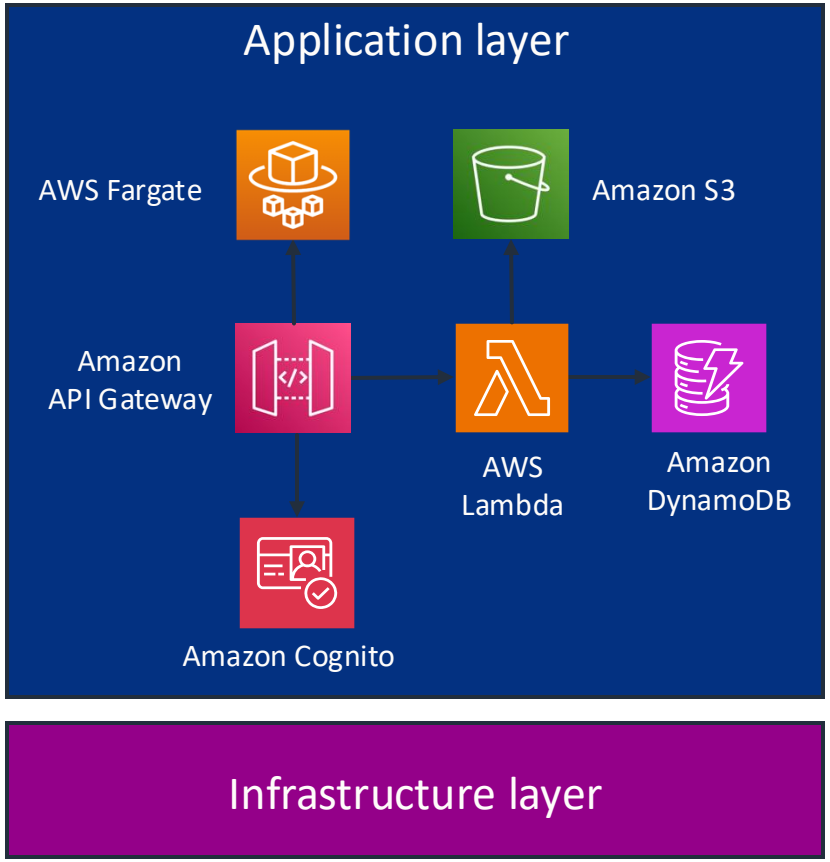
Serverless Security Essentials

Traditional Applications



- **Rigid separation** between infrastructure and application teams, tools, processes
- Frequently **coupled, manual workflows** for release and quality control
- **Long cycles** - need a new database? Open a ticket, we'll get back to you…

Serverless Applications



- **“Infrastructure” is redefined.** A function, an event-source mapping, an event routing rule are **app resources** owned by the app team.
- Updating app resources is commonly a part of **application developer’s responsibilities**.
- Guided **shift-left responsibilities** and trust between teams promote operational agility.

Serverless shifts security . . .

responsibility toward AWS

and

left in your organization

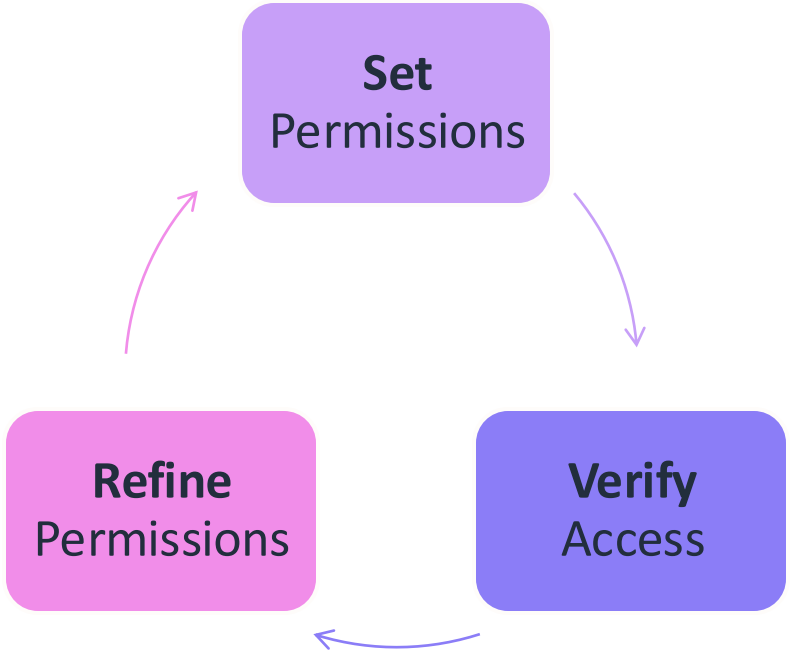
toward builders

Key priorities for Lambda function security

- Least Privilege
- Managing Credentials
- Dependency Vulnerabilities
- Code Quality Control

Grant least privilege permissions

- Least Privilege is the set of essential privileges needed to perform intended work
  - Evolves over time
- Attach via IAM Execution role
  - Prefer unique role per resource
  - Assign fine-grained permissions



Secure Amazon Lambda functions with IAM

Resource-based policy

- Defines how function can be **invoked**
- Supports cross-account access
- Used for synchronous and asynchronous invocations

Execution role

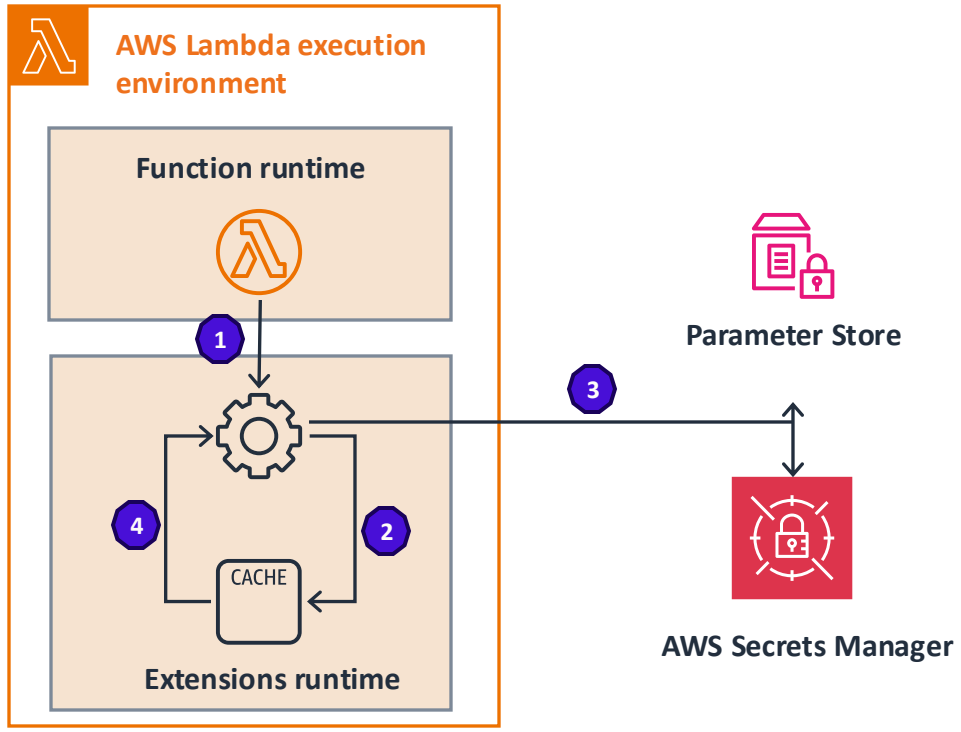
- Defines **which AWS resources** can access via IAM
- Used for poll-based invocations (Lambda polling)

"Actions on API Gateway A can invoke Lambda function B"

"Lambda function A can put an object in S3 bucket B"

Retrieve and cache secrets using Amazon Lambda Extension

- **No hard-coded credentials**
- **Lambda Extensions**: Use Secrets Manager/Parameter Store extensions to cache credentials locally



Validate event payloads in Amazon Lambda

- Validate input before processing, before parsing
- Use strict typing
  - Parameters should not accept more than one type of data primitive
  - Apply constraints, additional validation to ambiguous types (e.g., String)
- Consider for all event sources
  - Particularly important for APIs

Noncompliant

```
1 def verify_file_path_noncompliant():
2     from flask import request
3     file_path = request.args["file"]
4     # Noncompliant: user input file path is not sanitized.
5     file = open(file_path)
6     file.close()
```

Compliant

```
1 def verify_file_path_compliant():
2     from flask import request
3     base_path = "/var/data/images/"
4     file_path = request.args["file"]
5     allowed_path = ["example_path1", "example_path2"]
6     # Compliant: user input file path is sanitized.
7     if file_path in allowed_path:
8         file = open(base_path + file_path)
9         file.close()
```

Dependency management

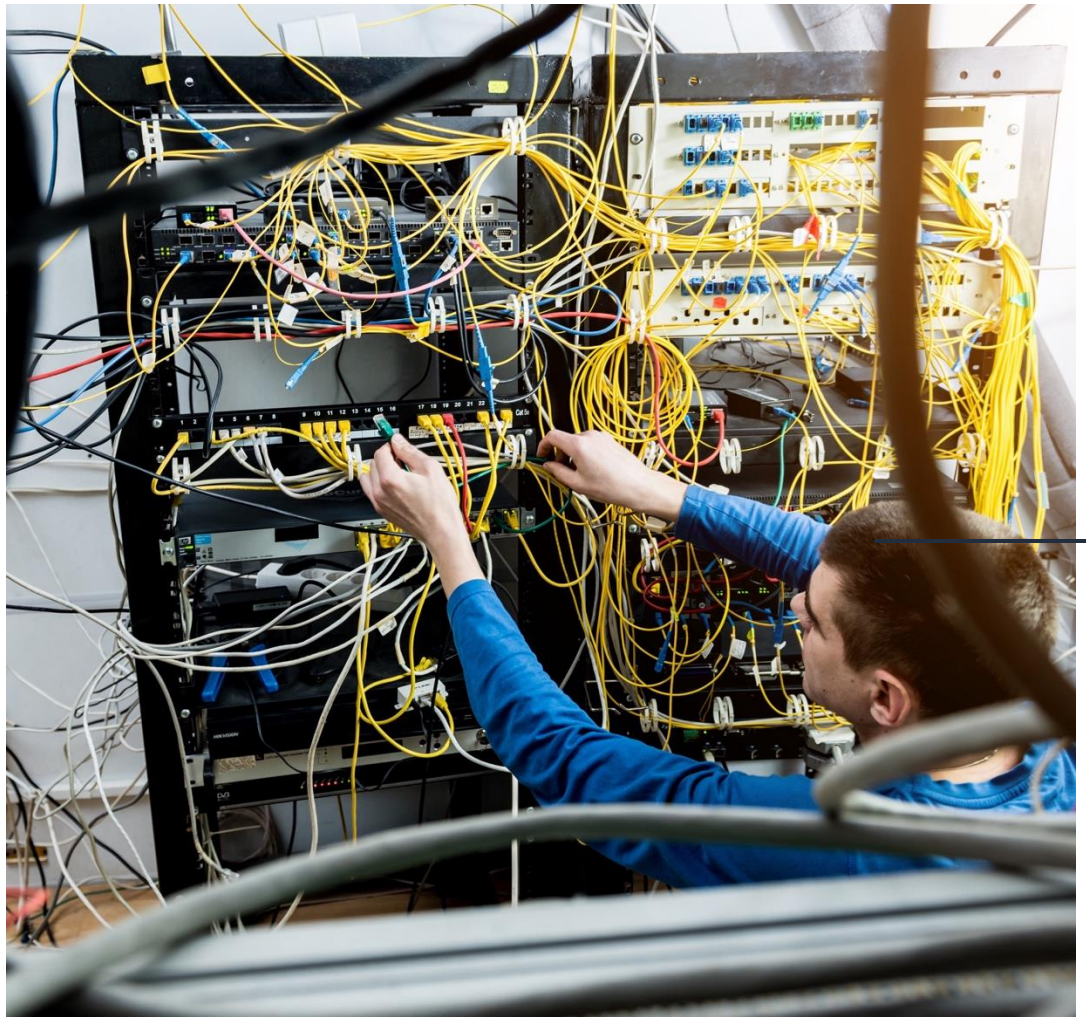
- Security Vulnerabilities
- Runtime Compatibility Issues
- Compliance Violations

Package.json:

```
{
  "name": "vulnerable-lambda-example",
  "version": "1.0.0",
  "description": "Lambda with vulnerable dependency",
  "main": "index.js",
  "dependencies": {
    "request": "^2.88.2"
  }
}
```

Security in Infrastructure as Code (CDK)

Infrastructure as Code (IaC)

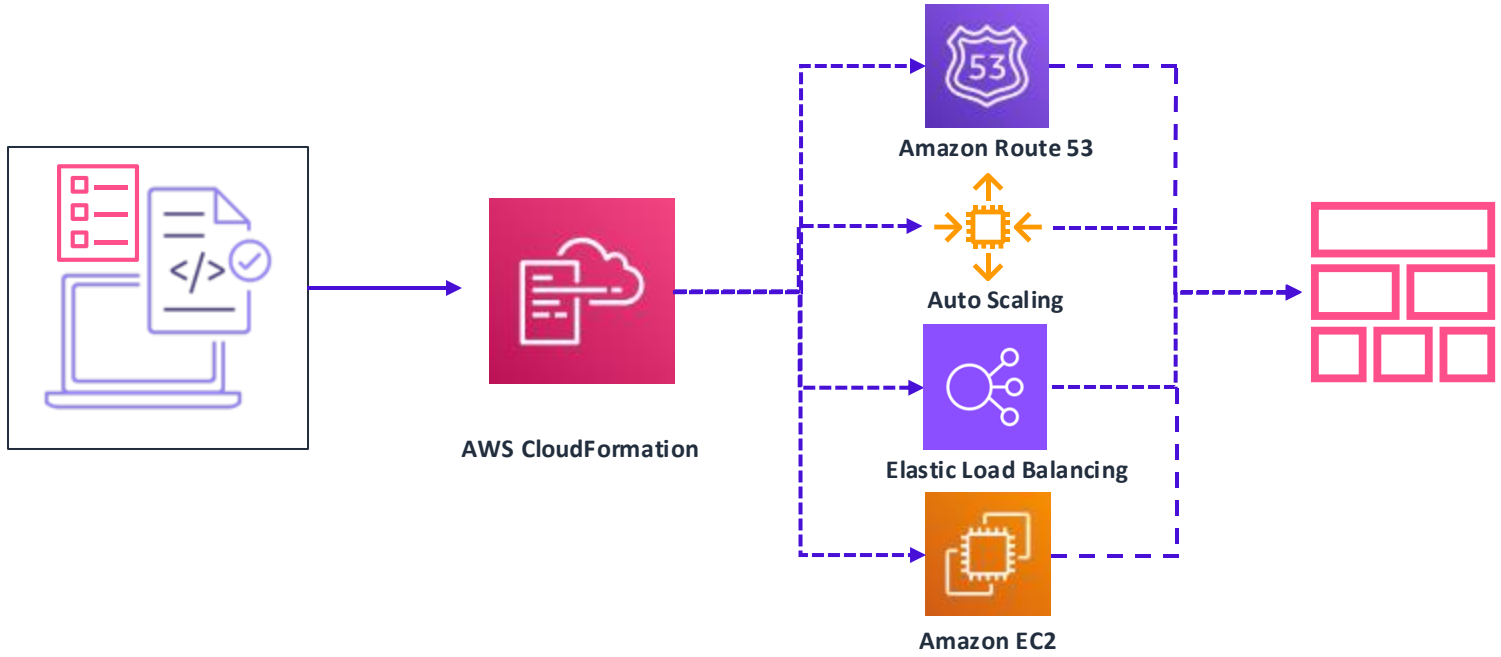


```yaml
AWSTemplateFormatVersion: "2010-09-09"

Parameters:
  VPCName:
    Description: The name of the VPC being created.
    Type: String
    Default: "VPC Public and Private with NAT"
Mappings:
  SubnetConfig:
    VPC:
      CIDR: "10.0.0.0/16"

Resources:
  VPC:
    Type: "AWS::EC2::VPC"
    Properties:
      EnableDnsSupport: "true"
      EnableDnsHostnames: "true"
      CidrBlock:
        Fn::FindInMap:
          - "SubnetConfig"
          - "VPC"
          - "CIDR"
      Tags:
        -
          Key: "Network"
          Value: "Public"

```

Developer tools

```
permissions.go | arns.go | ! forecast-lambda.yaml 5, M | permissions_
plates > ! forecast-lambda.yaml > {} Resources > {} MyFunctionRole > {} Properties > {} AssumeRolePolicyDocur
30                - logs:PutLogEvents
31              Effect: Allow
32              Resource: "*"
33          Roles:
34
35
36      MyFu
37        Ty
38        Pr
39        AssumeRolePolicyDocumen:
40          Statement:
41            - Action:
42                - sts:AssumeRole
43              Effect: Allow
44              Principal:
45                Service:
46                  - lambda.amazonaws.com
47          Version: "2012-10-17"
48          ManagedPolicyArns:

[cfn-lint] E3002: Invalid Property
Resources/MyFunctionRole/Properties/AssumeRolePolicyDocumen. Did
you mean AssumeRolePolicyDocument?
View Problem (⌥F8)   No quick fixes available

PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   zsh

Check: CKV_AWS_116: "Ensure that AWS Lambda function is configured for a Dead Letter Queue(DLQ)"
        FAILED for resource: AWS::Lambda::Function.MyFunction
        File: /test/templates/forecast-lambda.yaml:3-12
        Guide: https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-code-security-policy-reference/aws-policies/aws-general-policies/ensure-that-aws-lambda-function-is-configured-for-a-dead-letter-queue-dlq.html

                3 |   MyFunction:
                4 |     Type: AWS::Lambda::Function
                5 |     Properties:

0   Live Share   AWS: IAM Identity Center (d-9067925563)   CodeWhisperer   58 LOC, 0 Comment   Pr
```

Infrastructure as Code, AWS CloudFormation



- JSON/YAML format template
- Presents template to AWS CloudFormation
- AWS CloudFormation translates it to an API request
- Forms a stack of resources

- FREE – you only pay for resources
- All regions
- APIs are called in parallel
- Manages dependencies/relationships

AWS Cloud Development Kit (AWS CDK)

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    // Setup AutoScaling policy
    const scaling = service.service.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling',
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

domainName
domainZone

(property) patterns.NetworkLoadBala
ncedServiceBaseProps.domainName?: s
tring | undefined

The domain name for the service, e.g.
"api.example.com."

@default

- No domain name.

Your language
Just code

Tool support
Autocomplete
Inline documentation

Abstraction
Sane defaults
Reusable classes

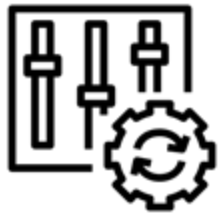
cdk-nag Overview

Supports

AWS CDK

Command

cdk synth

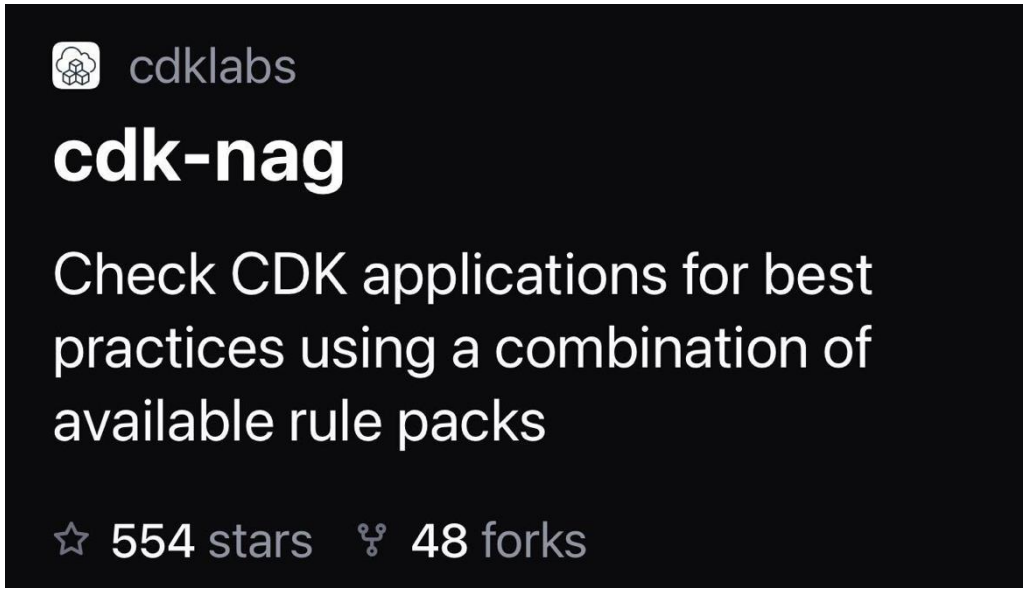


Configuration

Rule suppression
Custom rules

Rules/checks

NagPacks



https://shorturl.at/pG16Y

cdk-nag

- Bundled NagPacks
  - AWS solutions
  - HIPAA security
  - PCI DSS 3.2.1
  - NIST 800-53 rev 4
  - NIST 800-53 rev 5
- Create your own with custom NagPacks

```python
#!/usr/bin/env python3
import os

import aws_cdk as cdk

from cdk.cdk_stack import CdkStack
from aws_cdk import Aspects
from cdk_nag import AwsSolutionsChecks


app = cdk.App()
Aspects.of(app).add(AwsSolutionsChecks())
CdkStack(app, "CdkStack")

app.synth()
```

cdk-nag

An AWS cdk application

```python
from aws_cdk import (
    Stack,
    aws_s3 as s3,
)
from constructs import Construct

class CdkStack(Stack):

    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        s3.Bucket(self, "CdkBucket",
            bucket_name="my-bucket",
            block_public_access=s3.BlockPublicAccess.BLOCK_ALL,
```

cdk-nag

- synthesize

```
(DOP209-demv) → cdk git:(main) ✗ cdk synth
[Error at /CdkStack/CdkBucket/Resource] AwsSolutions-S1: The
S3 Bucket has server access logs disabled.

[Error at /CdkStack/CdkBucket/Resource] AwsSolutions-S10: The
 S3 Bucket or bucket policy does not require requests to use
SSL.


Found errors
```

DevSecOps implementation powered by gen AI

Amazon Q Developer

- Helps developers and IT professionals build faster across the entire software development lifecycle (SDLC)
- Most accurate coding recommendations
- Agents can autonomously help you implement features, refactor code, perform software upgrades and more
- Amazon Q is an expert on AWS and optimizing AWS environment
- Best-in-class security vulnerability scanning and remediation

Amazon Q is built with security and privacy in mind from the start, making it easier for organizations to use generative AI safely.

Build faster across the SDLC

Explore and plan

- Onboard to new projects faster
- Plan new features
- Understand how to use AWS APIs
- Ask questions about your internal code bases

Create

- Inline coding companion in IDE and CLI
- Software development
- Conversational coding

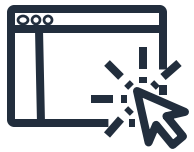
Test and Secure

- Improve test coverage with unit test generation
- Scan and remediate security vulnerabilities

Review and Deploy

- Automate code reviews
- Assess deployment risk
- Generate documentation

Available where you work



AWS consoles

IDEs

AWS documentation

AWS Console
Mobile Application

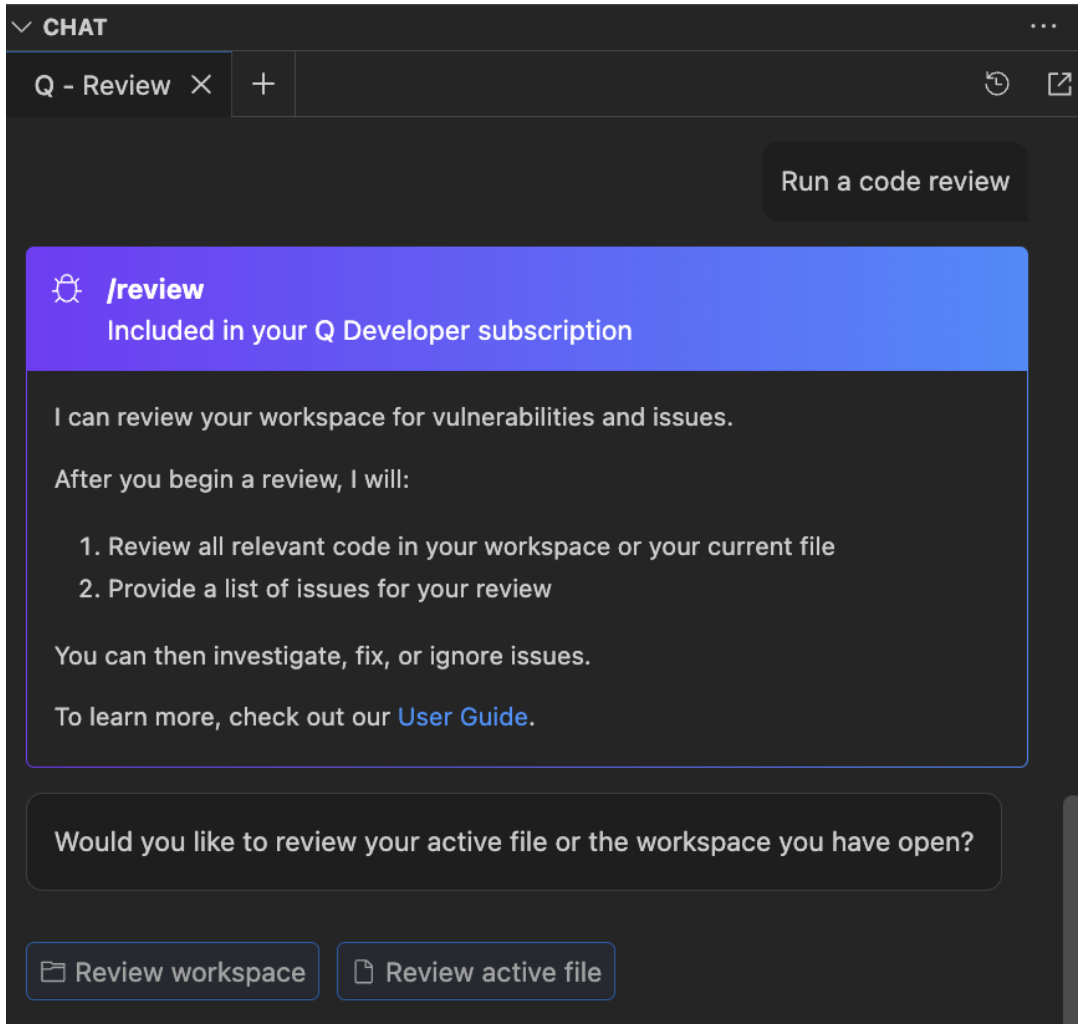
Slack and Teams
(through AWS Chatbot)

GitLab

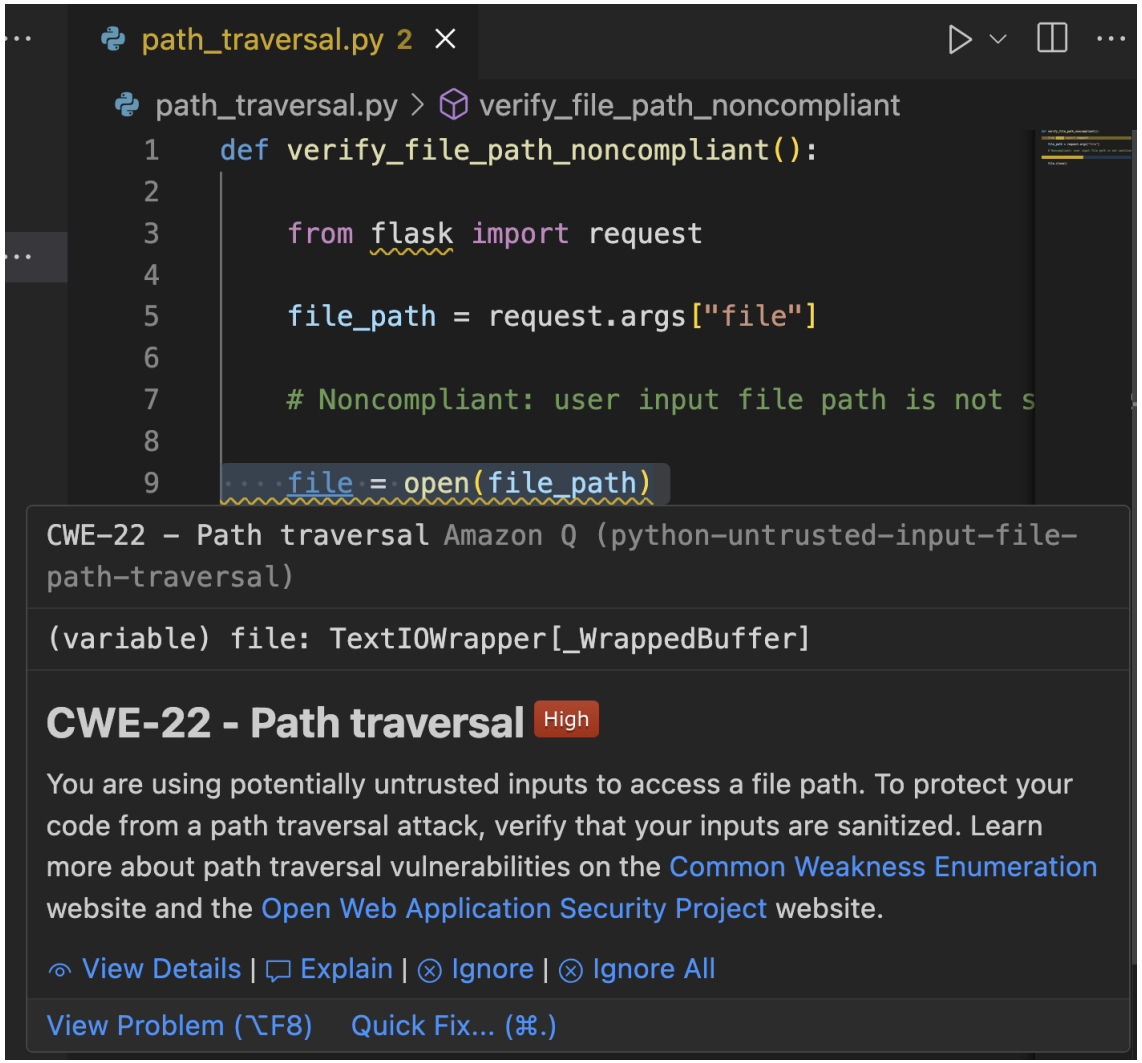
Gitlab Duo with
Amazon Q

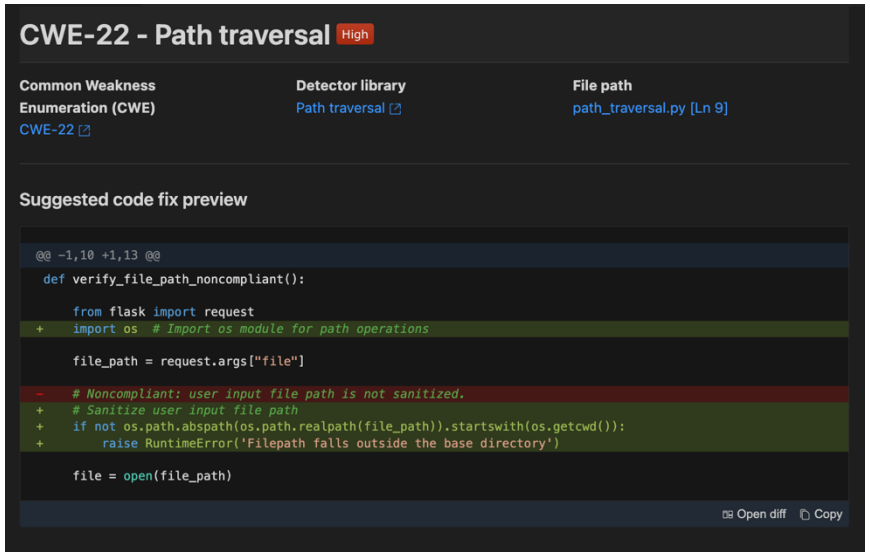
Amazon Q Developer /review

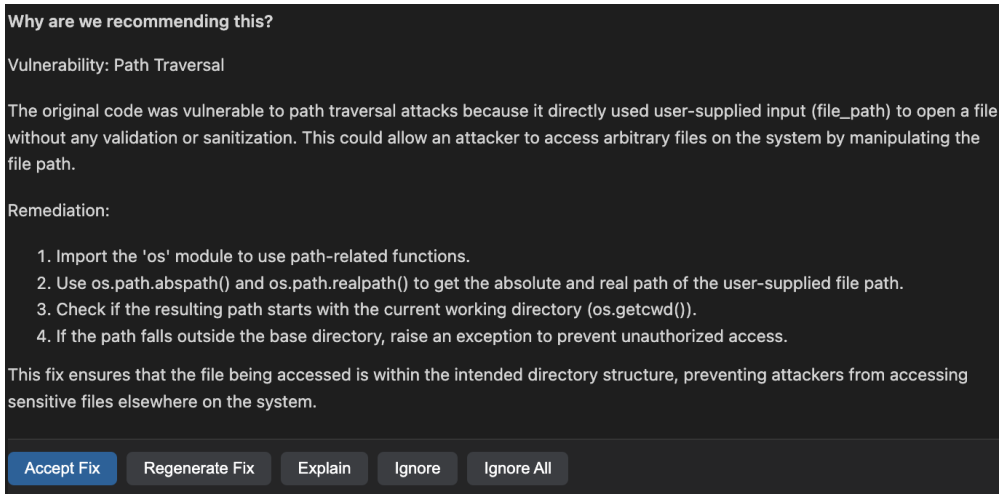
- SAST scanning
- Secrets detection
- IaC issues
- Code Quality issues
- Code deployment risks
- Software composition analysis (SCA)



Security Review for application code







Vibe Coding

Vibe Coding Warnings

Never **blindly** trust code generated by AI assistants. Always:

- Thoroughly **review** and understand the generated code
- Verify all dependencies
- Perform necessary **security checks**
- **Test** the code in a controlled environment

Model Context Protocol (MCP)

MCP Introduction

What

1. Integrates LLMs with **external** data and tools.
2. Standardized client-server architecture for LLM capabilities.
3. Provides contextual information to enhance LLM outputs.

Why

1. Improves LLM output quality and accuracy.
2. Keeps LLMs up-to-date with latest information.
3. Enables specialized workflows and domain knowledge.

https://modelcontextprotocol.io/introduction

AWS MCP Servers

AWS MCP Server	Description
Core MCP Server	Manages and coordinates the other AWS MCP Servers.
AWS Documentation MCP Server	Provides access to AWS documentation and best practices.
Amazon Bedrock Knowledge Bases Retrieval MCP Server	Retrieves and queries data from Amazon Bedrock knowledge bases.
AWS CDK MCP Server	Provides assistance with AWS CDK best practices.
Cost Analysis MCP Server	Analyzes and visualizes AWS costs.
Amazon Nova Canvas MCP Server	Generates images using Amazon Nova Canvas.
AWS Diagram MCP Server	Creates diagrams using the Python Diagrams package.
AWS Lambda MCP Server	Runs AWS Lambda functions as MCP tools.
AWS Terraform MCP Server	Provides best practices for AWS Terraform development.

https://shorturl.at/vSVRX

Demo

- Roo Code
- Configure AWS MCP Servers in Roo Code
- Generate a simple serverless application using Roo Code
- AWS CDKv2 in Python
- Use AWS CDK MCP Server to implement security best practices
- Amazon Q Developer to perform SAST review

Recap

- DevSecOps and Shift-Left mindset
- Choice of Serverless
- Maintain your applications and infrastructure using IaC
- Vibe Coding with Security
- Review Code

Additional resources

- Best practices for working with AWS Lambda functions
- Best practices for accelerating development with serverless blueprints
- AWS MCP Servers
- Amazon Q Detector Library
- Amazon Q Developer CLI
- What is DevSecOps

Dive deeper with AWS migrate and modernize resources

Unlock the full potential of the cloud with AWS—learn how to seamlessly migrate your workloads and modernize with tools, best practices, and expert support.

Visit the Resource Center via event platform homepage »

AWS Training & Certification

Access 600+ free digital courses with AWS Skill Builder

Focus on the cloud skills and services that are most relevant to you across 30+ AWS solutions, including digital self-paced learning plans and ramp-up guides

- Build your future in the AWS Cloud at your own pace
- Advance your skills and knowledge with learning plans
- Validate your cloud expertise with AWS Certification

https://shorturl.at/22v8J

Learn your way, explore skillbuilder.aws >>

Thank you for attending **AWS Builders Online Series**

We hope you found it interesting! A kind reminder to complete the survey.
Let us know what you thought of today's event and how we can improve the event experience for you in the future.


aws-apj-marketing@amazon.com


twitter.com/AWSCloud


facebook.com/AmazonWebServices


youtube.com/user/AmazonWebServices


linkedin.com/company/amazon-web-services


twitch.tv/aws

Thank you!

Sam Zhang

Security Specialist Technical Account Manager
AWS