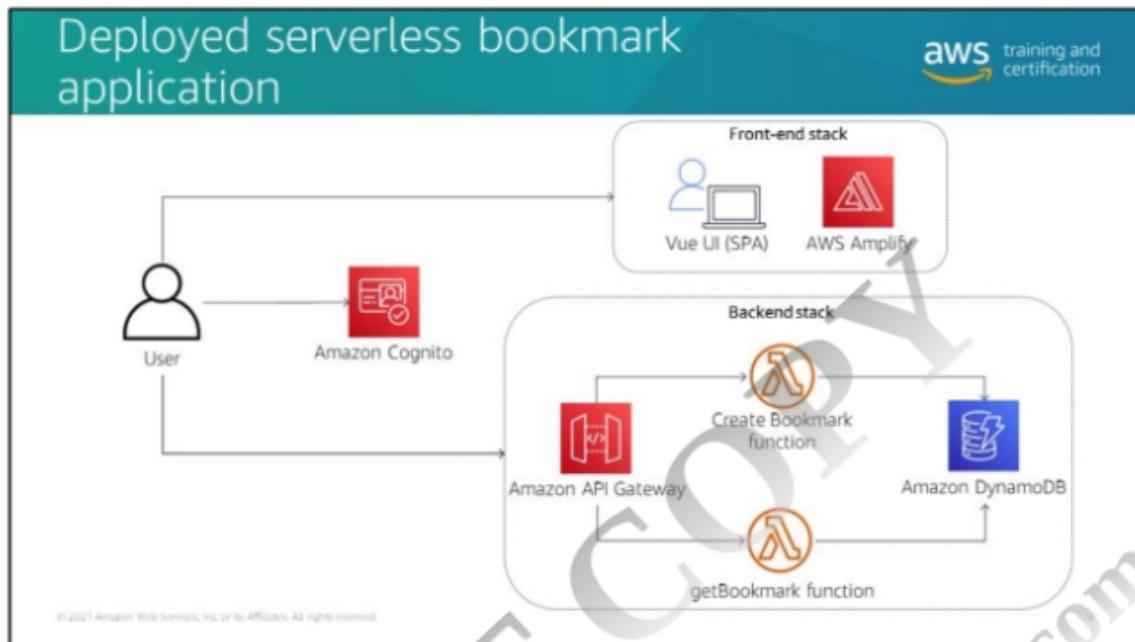


© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.



In your first lab, you deployed a serverless application that lets users sign in, save a bookmark, and view their saved bookmarks in the UI. This slide illustrates what you built in the lab.

Amazon Cognito is used to authenticate users, and Amazon API Gateway proxies incoming requests to the `createBookmark` and `getBookmark` functions. Bookmarks are saved to an Amazon DynamoDB table.

The tasks you completed in lab 1 included the following:

- Installing the latest version of the AWS Serverless Application Model (AWS SAM) command line interface (CLI) in your AWS Cloud9 integrated development environment (IDE)
- Configuring authentication through an Amazon Cognito user pool
- Deploying your backend code using AWS SAM
- Updating your front-end configuration and running the build through AWS Cloud9 to test it prior to deployment
- Deploying your front-end application using AWS Amplify

# Lab 2: Message Fan-Out with Amazon EventBridge

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.





Now that you've verified your serverless proof of concept with the first iteration, you need to add the capability for sharing bookmarks with the team.

To get the managers on board with your new application, you need align to the existing knowledge base process.

To get the team on board, your product manager has authorized a contest so that each new submission through the application enters the submitter into a drawing to win a prize.

## Lab 2 application requirements

**Functional requirements**

- Allow users to select a bookmark to be shared in the team's knowledge base
- Forward information about submissions to managers for review
- Register a contest entry for each new user submission

**Technical requirements**

- Use the AWS serverless stack
- Use an event-driven approach for processing submissions
- Use fan-out to perform parallel activities

**Developer tools**

- AWS Cloud9 IDE
- AWS SAM and AWS SAM CLI

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

## Lab 2 objectives



- Enable DynamoDB Streams as an event source for a Lambda function that is triggered when new items are added to your DynamoDB table
- Configure an EventBridge event bus with a Lambda function as its event source and Lambda, Amazon SNS, and CloudWatch as targets
- Configure EventBridge rules that route events to your targets based on criteria you specify
- Configure an Amazon SNS topic that notifies an email subscriber
- Use AWS SAM to deploy your updated application

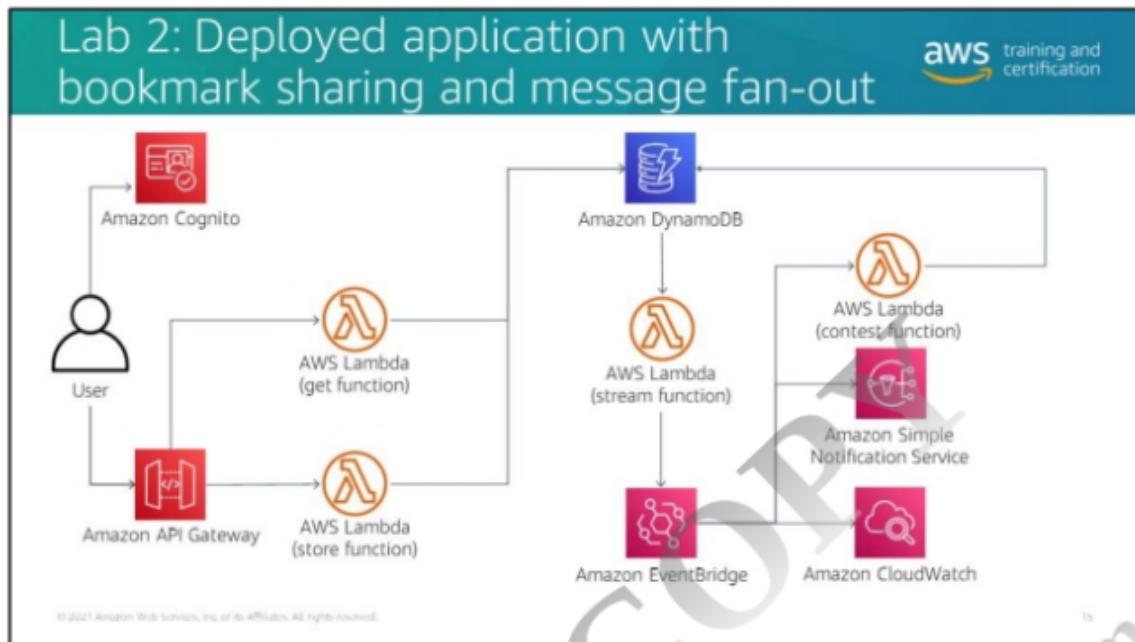
© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

13



© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

14



In your second lab, you added features to your bookmarks application to support sharing bookmarks, notifying managers, and entering submissions into a contest.

This slide illustrates what you built in the lab. Incoming requests from the storeFunction to DynamoDB trigger DynamoDB Streams on your DynamoDB table. The DynamoDB stream triggers the AWS Lambda stream function, which in turn is an event source for Amazon EventBridge. EventBridge routes messages to the Lambda contest function, to Amazon Simple Notification Service (Amazon SNS), and to Amazon CloudWatch based on rules.

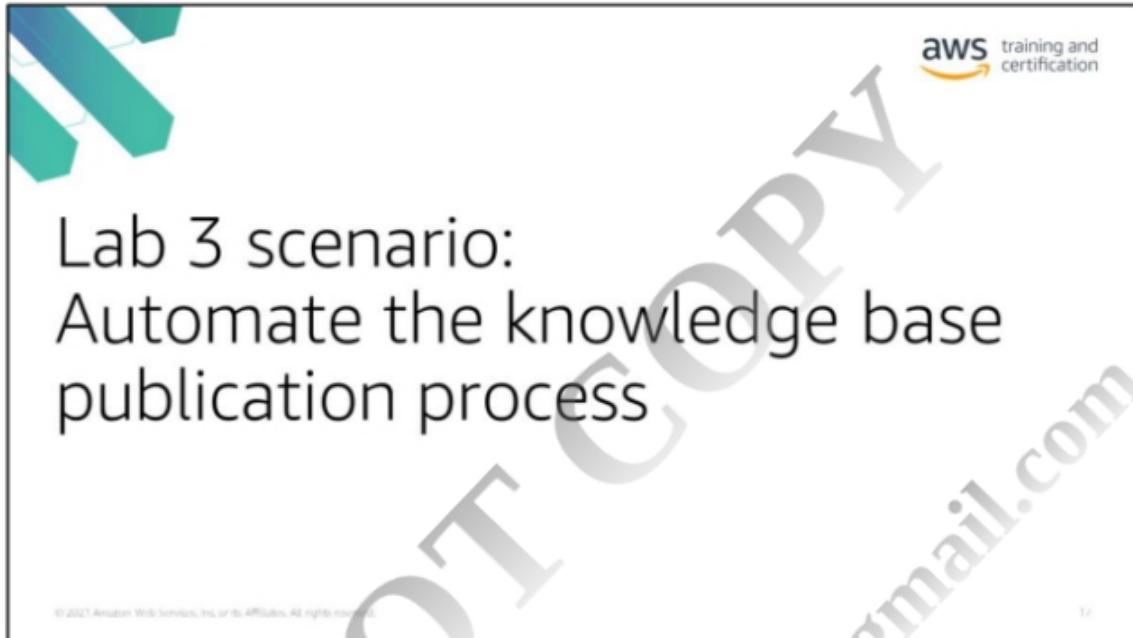
The tasks you completed in lab 2 include the following:

- Enabling DynamoDB Streams with a Lambda trigger
- Configuring an EventBridge bus with a Lambda event source and rules that route incoming messages to CloudWatch, Lambda, and Amazon SNS
- Configuring an SNS topic with an email subscriber
- Testing your updated application
- Using AWS SAM to deploy your updated application

# Lab 3: Workflow Orchestration Using AWS Step Functions

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.





Yesterday, you deployed your serverless bookmark application and then updated it to support bookmark sharing, routing of request to the managers' mailbox, and support for a contest feature.

The managers who review the knowledge base submissions recognize that submissions are going to increase a lot with the application, so they want you to scale their submissions process by automating as much of it as possible. As part of their manual review, the managers check to make sure the submission isn't a duplicate, they validate the URL and make sure it meets eligibility requirements to be shared, and then they publish the resource. Next, they have to let the submitter know whether or not the item has been added. When the item is a duplicate or is invalid, the submitter does not get a contest entry for that item, so the managers need to inform the contest administrator of duplicates and bad submissions.

Given the variety of sources and type of materials that might be shared, the managers still want to maintain a manual approval step so they can review the material and make sure it follows practices and technologies that they want to promote.

## Lab 3 application requirements

**Functional requirements**

- Check for duplicate submissions
- Validate the URL provided
- Provide a method for pausing the process for manual approval
- Remove the contest submission for invalid or duplicate entries

**Technical requirements**

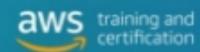
- Use the AWS serverless stack
- Keep orchestration logic out of Lambda functions

**Developer tools**

- AWS Cloud9 IDE
- AWS SAM and AWS SAM CLI

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

## Lab 3 objectives



- Configure EventBridge to target a Step Functions workflow
- Use a Step Functions standard workflow to orchestrate tasks
- Use Lambda for tasks within a Step Functions state machine
- Include a manual approval step in the workflow
- Select states and use Amazon State Language for state machine logic

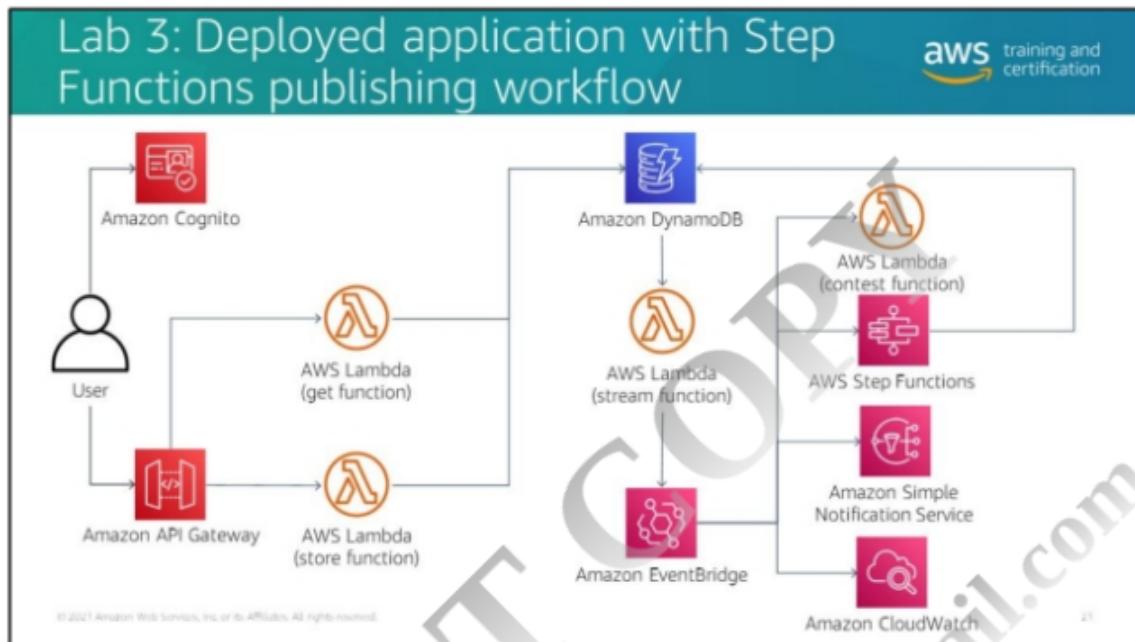
© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

19



© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

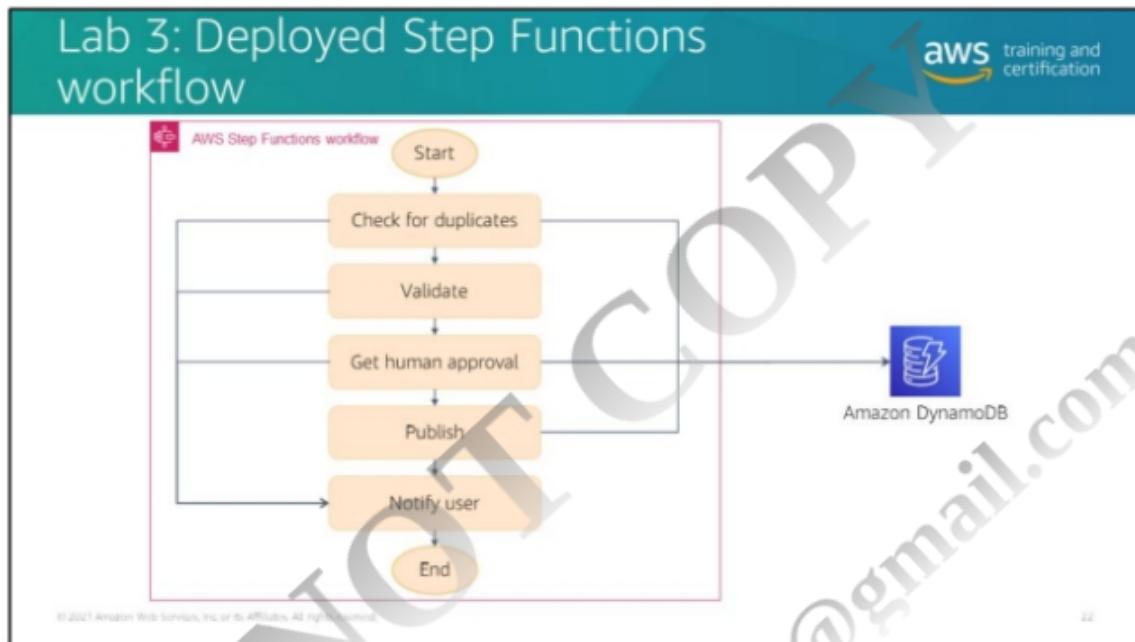
20



In your third lab, you updated your serverless application to incorporate a Step Functions workflow that orchestrates steps in the publishing process. This slide illustrates what your application looks like after completing the third lab.

Step Functions has been added as a target of your EventBridge bus.

The next slide describes the detailed steps of the Step Functions workflow you created.



This slide depicts the Step Functions workflow you deployed.

The workflow checks the DynamoDB table for duplicates to the submitted link, validates its format, and then sends the request for manager approval.

When the approval is provided, the workflow adds the approved submission to the knowledge base DynamoDB table and notifies the submitter regarding their entry (whether it was a duplicate or invalid or whether it has been added).

The publish step also removes the contest attribute on the submission if the submission were invalid or a duplicate of an item already published in the knowledge base.

The tasks you completed in lab 3 included the following:

- Configuring EventBridge to target a Step Functions workflow
- Using a Step Functions standard workflow to orchestrate tasks
- Using Lambda for tasks within a Step Functions state machine
- Using the “feature” to implement manual approval in the workflow

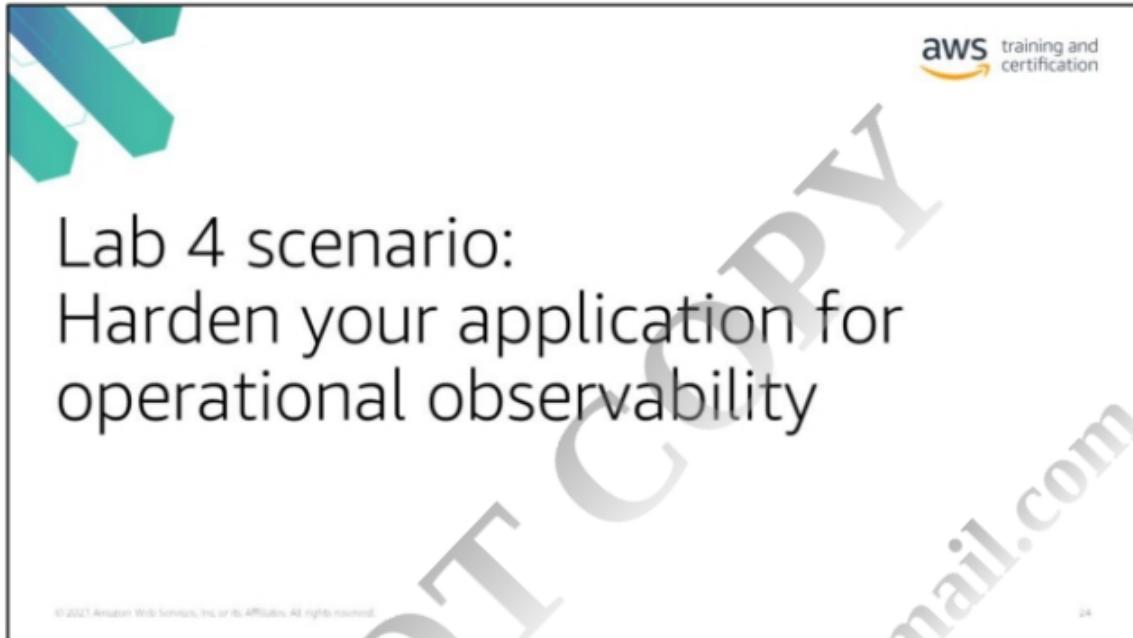
- Selecting states and using Amazon State Language for state machine logic

DO NOT COPY  
faroqahmad.dev@gmail.com

# Lab 4: Observability and Monitoring

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.





You now have a stable bookmarking applications with the features you need to roll it out to the larger team. Before it becomes your production version, you need to think about the operational data you'll need in order to know if it's working as expected and to understand how users are accessing it. You need to build observability into your application..

## Lab 4 application requirements

**aws training and certification**

### Operational requirements

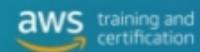
- Provide observability for end-to-end transactions
- Log event details that will support troubleshooting
- Alarm on key metrics

### Developer tools

- X-Ray
- CloudWatch
- AWS Cloud9 IDE
- AWS SAM and AWS SAM CLI

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

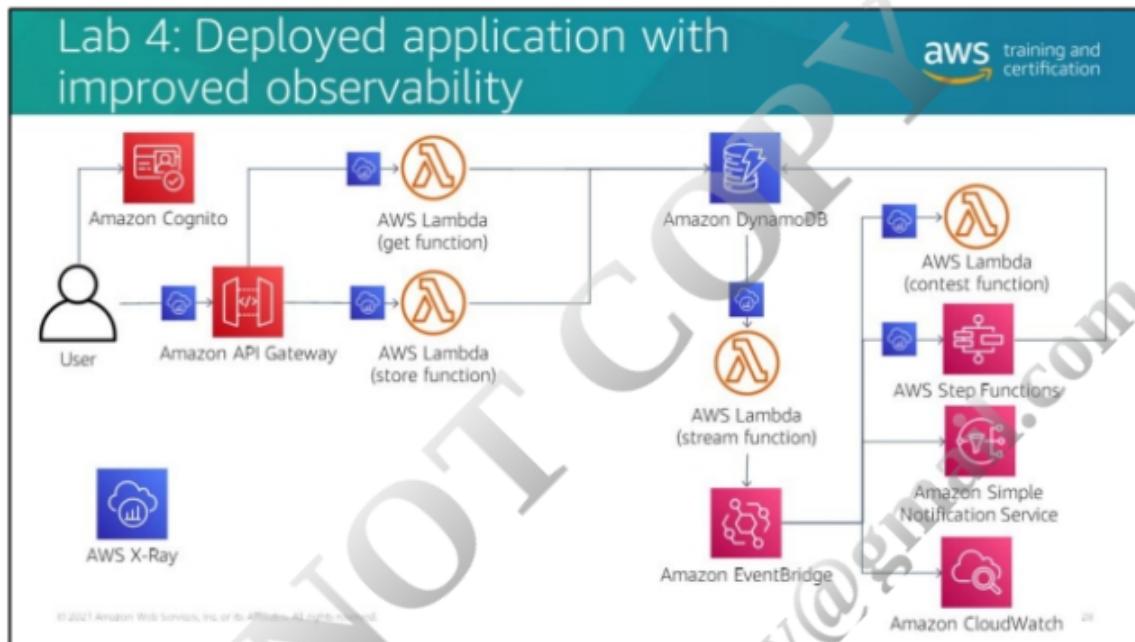
## Lab 4 objectives



- Use pre-built code to generate application errors
- Update your code for better logs, applying logging best practices
- Use CloudWatch Logs and CloudWatch metrics to monitor application operations
- Troubleshoot application issues using X-Ray and CloudWatch ServiceLens

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.





In this lab, you introduced AWS X-Ray for tracing requests, improved the usefulness of your logs, and used CloudWatch features to review operational data about your application.

This diagram illustrates the same architecture of the application you deployed in lab 3 with callouts to the places where you directly enabled X-Ray for Lambda, API Gateway, and Step Functions.

The tasks you completed in lab 4 included the following:

- Updating your Lambda code to provide additional troubleshooting data
- Monitoring your applications and infrastructure resources using CloudWatch and CloudWatch Metrics
- Analyzing your applications and troubleshooting and fixing issues with X-Ray and CloudWatch ServiceLens

# Lab 5: Securing Serverless Applications

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.





In the labs on day 1 you deployed the first iteration of your serverless bookmark application that lets a user sign in and add or review bookmarked resources. You used Amazon Cognito for user authentication and Amazon API Gateway, Lambda, and DynamoDB for your backend.

In the second lab, you added bookmark sharing and contest features to your application. You introduced event-driven design with DynamoDB Streams and messaging fan-out using EventBridge.

In the first lab on day 2, you updated your application with an automated workflow to simplify the publishing step for shared bookmarks using Step Functions.

Starting with the second lab yesterday, you froze the applications features and began working to harden your application for production use. In lab 4, you introduced X-Ray and worked with CloudWatch to monitor and troubleshoot your application.

Before you'll be able to make this application available outside your development team, you need to review security best practices for securing access and protecting resources and data.

In this lab, you continue hardening your application for production by addressing application security.

DO NOT COPY  
farooqahmad.dev@gmail.com

## Lab 5 application requirements



### Operational requirements

- Increase permission limits on accessing your API
- Protect the application from common external attacks
- Secure Lambda functions and backend services
- Protect credential data used in the application

### Developer tools

- X-Ray
- CloudWatch
- AWS Cloud9 IDE
- AWS SAM and AWS SAM CLI

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

DO NOT COPY  
faroogahmad.dev@gmail.com

## Lab 5 objectives

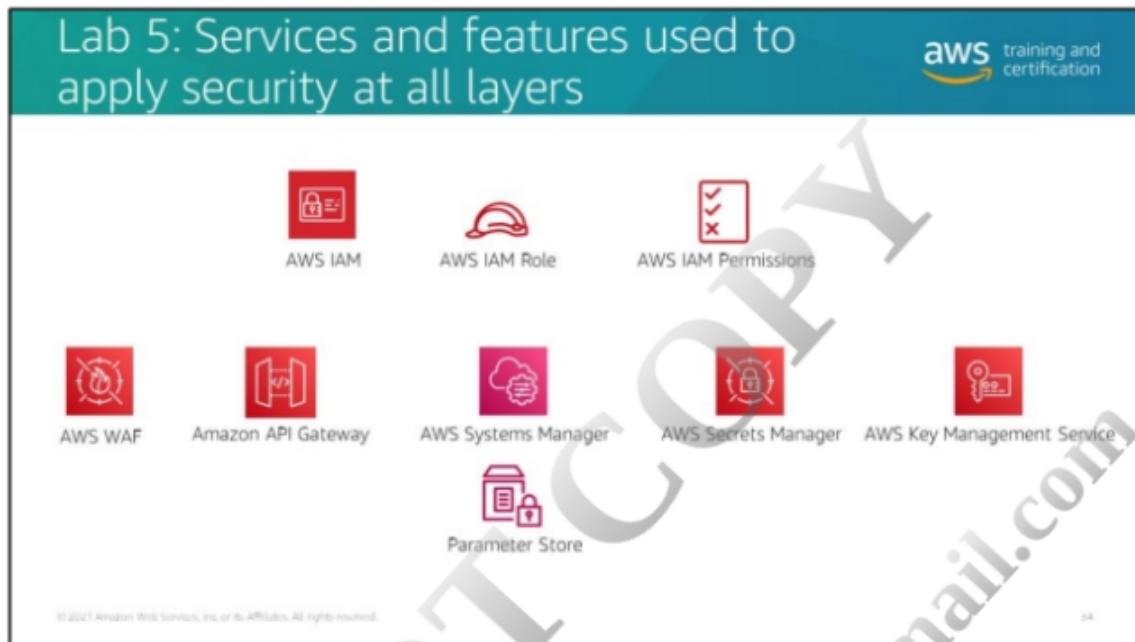


- Secure your application with AWS WAF
- Secure your access to your API with an API Gateway resource policy
- Secure your Lambda functions and other backend services
- Handle secrets with AWS KMS, Systems Manager Parameter Store, and Secrets Manager

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

52





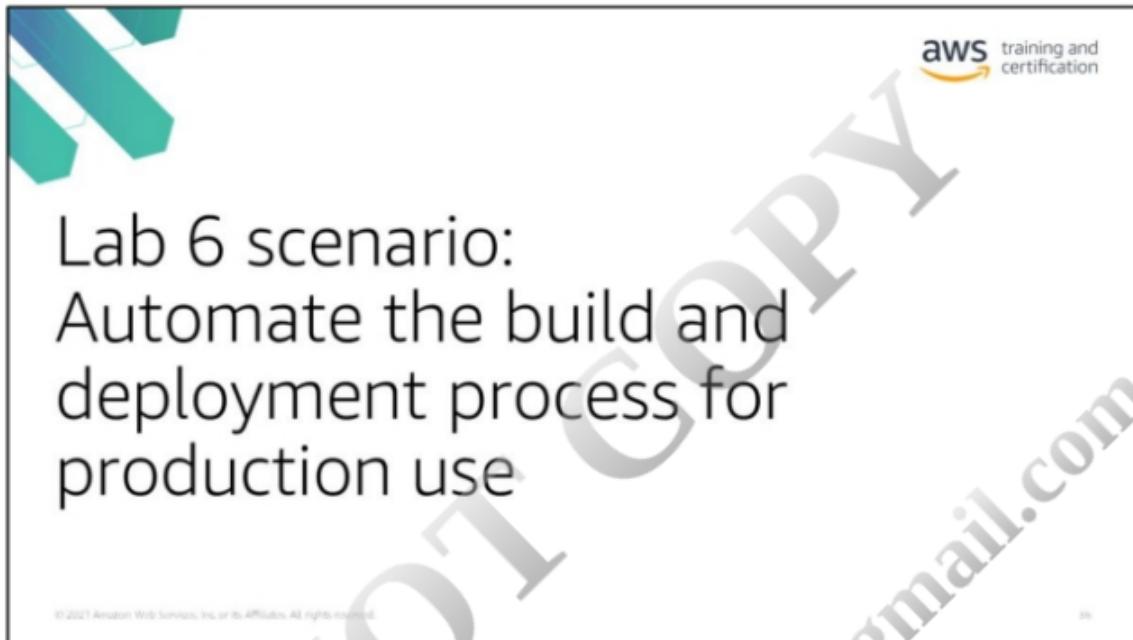
In this lab, you updated your application to align with best practices for securing your application at all layers. The architecture illustration did not change, but this slide illustrates the AWS services you used to secure your application.

The tasks you completed in lab 5 included the following:

- Configuring AWS WAF to protect against external attacks
- Securing your API using an API Gateway resource policy
- Securing your Lambda functions with AWS Key Management Service (AWS KMS), AWS Systems Manager Parameter Store, and AWS Secrets Manager



You have developed and tested your application with support from your development team. During this journey, you would like to automate the build and deployment processes so that the development team continues to focus more on developing the application. In this lab, you will learn how to set up a Continuous integration/continuous deployment (CI/CD) pipeline and do canary deployments on AWS.



Now you have a **fully functional serverless application** that allows users to sign in, save bookmarks for their own use, and submit bookmarks to be shared in the team knowledge base. Several tasks happen in parallel when a user submits a new bookmark for the knowledge base: The submitter's entry is entered into a contest, notifications are sent, and an automated publishing workflow is triggered. You have also incorporated logging and tracing that will give you operational visibility into the production application. Finally, you incorporated security at all layers to protect your application.

As you have learned, a key part of developing serverless applications is to use your visibility into how the application is used in production to determine where future modifications should be made. You should continue to monitor the production application to understand access patterns, resolve operational errors, and iteratively optimize your application to minimize costs and continually improve the user experience. Because your application uses small decoupled components, you can more easily modify components independently.

To support these types of small, frequent updates, you need to automate the build and deployment processes for your application.

## Lab 6 application requirements



### Operational requirements

- Automate the build process for application components
- Automate the deployment of new builds
- Provide mechanisms for automatically rolling back failed builds and deployments

### Developer tools

- AWS Cloud9 IDE
- AWS SAM and AWS SAM CLI
- AWS CDK
- CodeCommit, CodeBuild, and CodePipeline
- CodeDeploy (through AWS SAM template)

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

DO NOT COPY  
farooqahmad.dev@gmail.com

## Lab 6 objectives

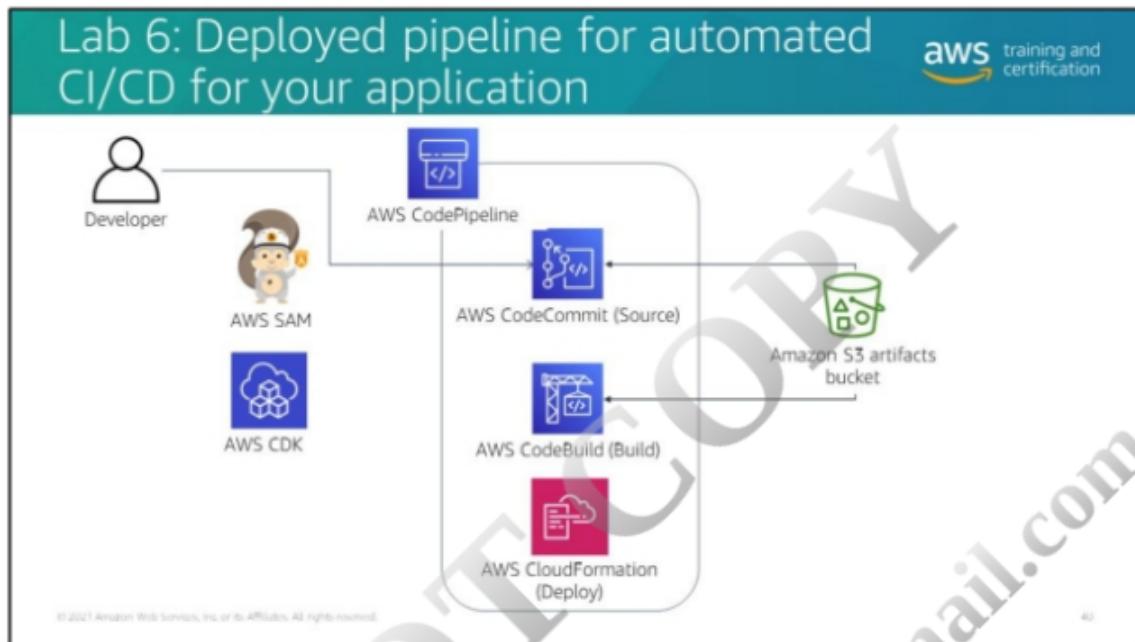


- Build a CI/CD pipeline using AWS tools
- Use AWS SAM and AWS CDK to automate build and deployment tasks.
- Implement canary deployments using AWS SAM template attributes

© 2021 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

31





In this lab, you automated the build and deployment process for your application. This diagram illustrates the developer tools that you used and the flow of the pipeline.

You used AWS SAM to define the resources your application needs and AWS Cloud Development Kit (AWS CDK) to define the resources for the deployment infrastructure.

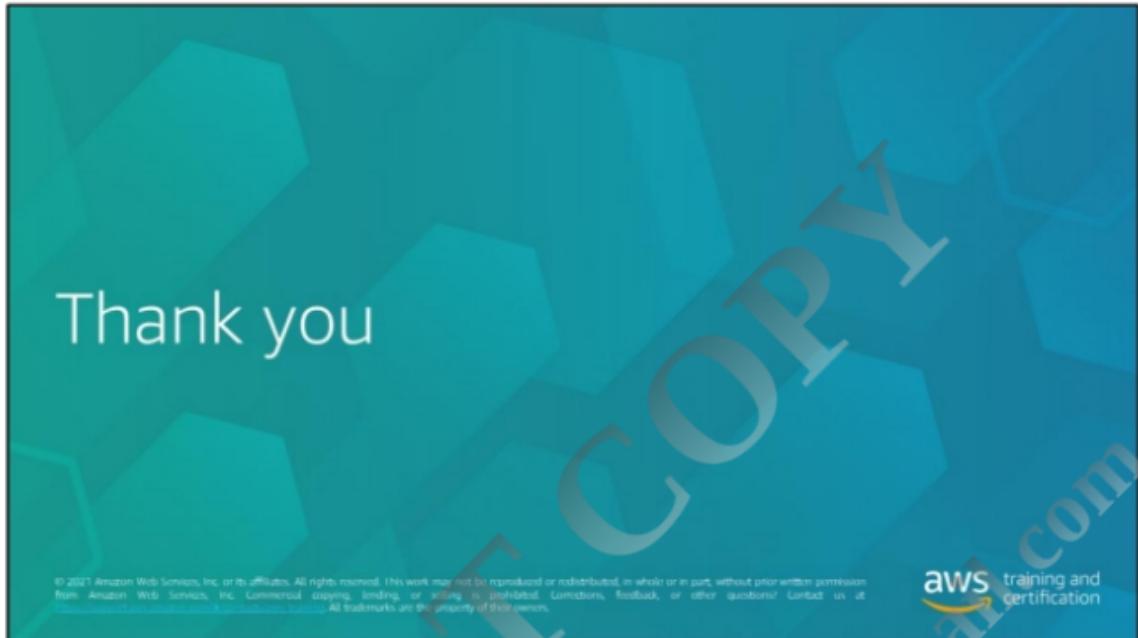
AWS CodePipeline orchestrates your CI/CD process. AWS CodeCommit is your source repository. Changes in CodeCommit trigger AWS CodeBuild to produce a new build. CodeBuild completes the build, and a successful build triggers a deployment. Commands within your AWS SAM template include guidance for an AWS CodeDeploy application that CloudFormation will build as part of your infrastructure.

The tasks you completed in lab 6 included the following:

- Configuring a CI/CD pipeline for your application
- Using AWS SAM templates to define your application resources
- Using AWS CDK to define your infrastructure resources for your pipeline

- Using the AWS SAM template DeploymentPreference section to define pre- and post-deployment hooks and canary deployment criteria
- Monitoring your deployment

DO NOT COPY  
faroogahmad.dev@gmail.com



© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at [feedback@aws.training](mailto:feedback@aws.training). All trademarks are the property of their owners.

**aws** training and certification