

Assignment-5

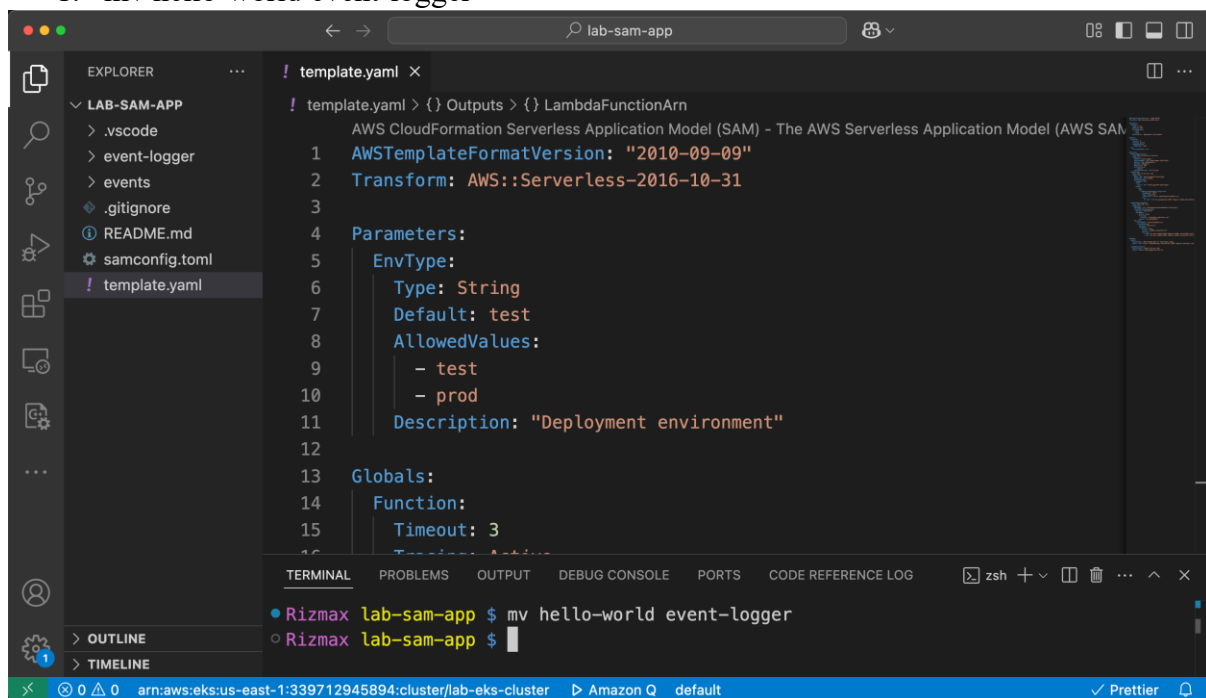
API Gateway

Task 1 - Set Up AWS SAM Project and Define Infrastructure

AWS SAM simplifies the deployment of serverless applications. In this task, you will initialize a new AWS SAM project and define the infrastructure needed for the lab.

Initialize the AWS SAM Project

- Open a terminal and run the following command to create a new AWS SAM project:
 1. `sam init --name lab-sam-app --runtime nodejs22.x`
- Choose:
 - (1) AWS Quick Start Templates
 - (1) Hello World Example
 - (1) Hello World Example as your starter template
 - (y) Enable X-Ray tracing
 - (N) CloudWatch Application Insights
 - (y) Structured logging in JSON format
- Navigate into the project folder and open it VS Code.
 1. `cd lab-sam-app`
 2. `code .`
- Rename the hello-world directory to event-logger.
 1. `mv hello-world event-logger`

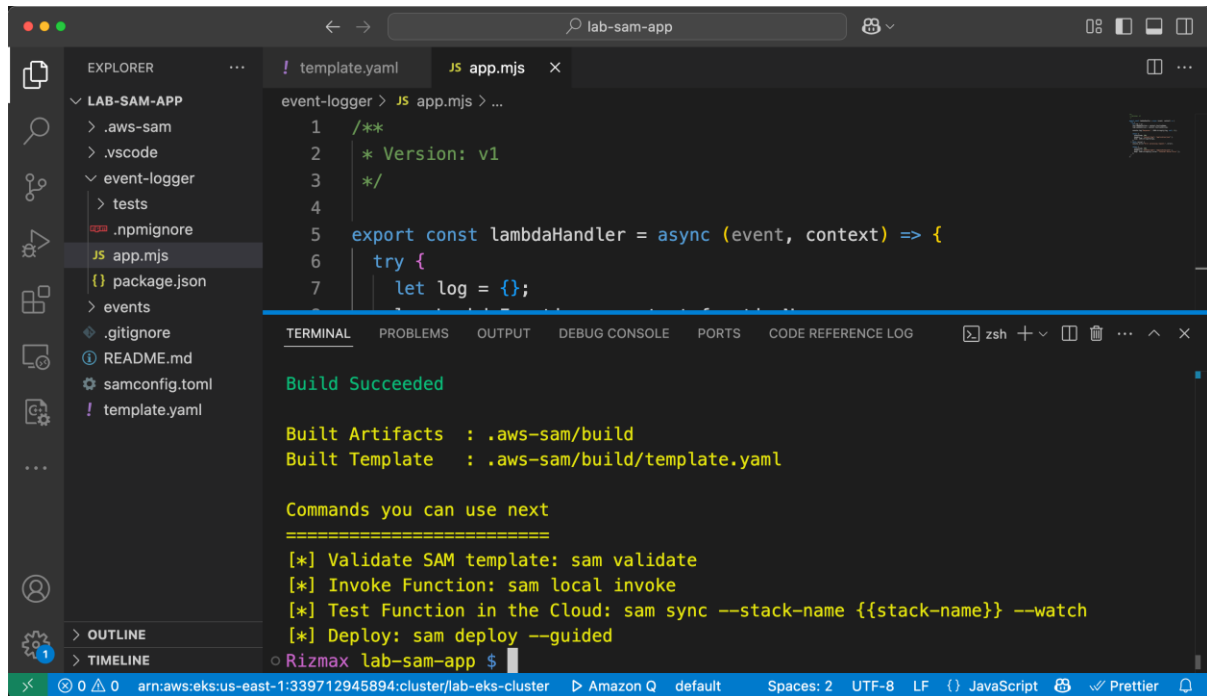


Task 2 - Deploy the AWS SAM Application

Now that you have defined the infrastructure, you will build and deploy it using AWS SAM.

Build and Deploy the Application

- Make sure you have AWS CLI configured or have set AWS temporary credentials in your terminal session.
- Build the application:
 1. `sam build`



The screenshot shows a Visual Studio Code editor with a project named 'lab-sam-app'. The Explorer sidebar on the left shows the file structure: .aws-sam, .vscode, event-logger, tests, .npmignore, JS app.mjs, package.json, events, .gitignore, README.md, samconfig.toml, and template.yaml. The main editor window shows the 'app.mjs' file with the following code:

```
1  /**
2   * Version: v1
3   */
4
5  export const lambdaHandler = async (event, context) => {
6    try {
7      let log = {};
```

The TERMINAL panel at the bottom shows the output of the 'sam build' command:

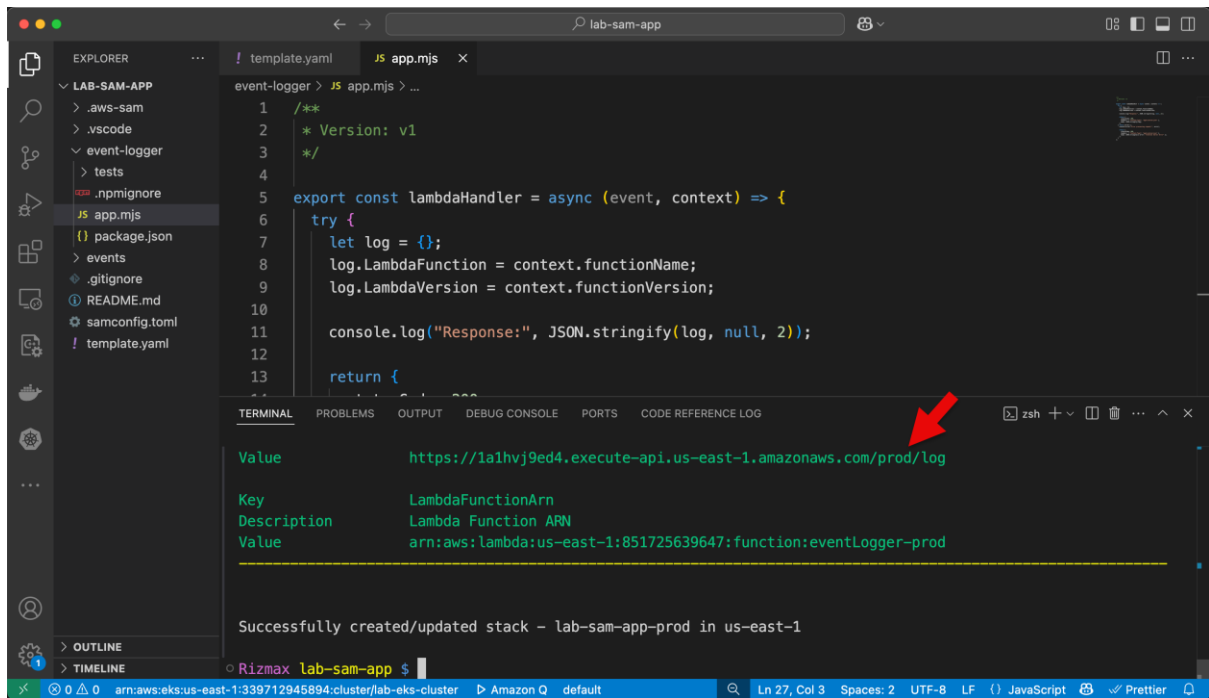
```
Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=====
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {{stack-name}} --watch
[*] Deploy: sam deploy --guided

Rizmax lab-sam-app $
```

- Deploy the infrastructure for test environment. Enter y for any confirmation prompts.
 1. `ENV_TYPE=test`
 2. `sam deploy --stack-name=lab-sam-app-${ENV_TYPE} --capabilities CAPABILITY_NAMED_IAM --parameter-overrides EnvType=${ENV_TYPE}`
- Deploy the infrastructure for prod environment. Enter y for any confirmation prompts.
 1. `ENV_TYPE=prod`
 2. `sam deploy --stack-name=lab-sam-app-${ENV_TYPE} --capabilities CAPABILITY_NAMED_IAM --parameter-overrides EnvType=${ENV_TYPE}`
- Note the API Gateway method endpoint URL from the output.

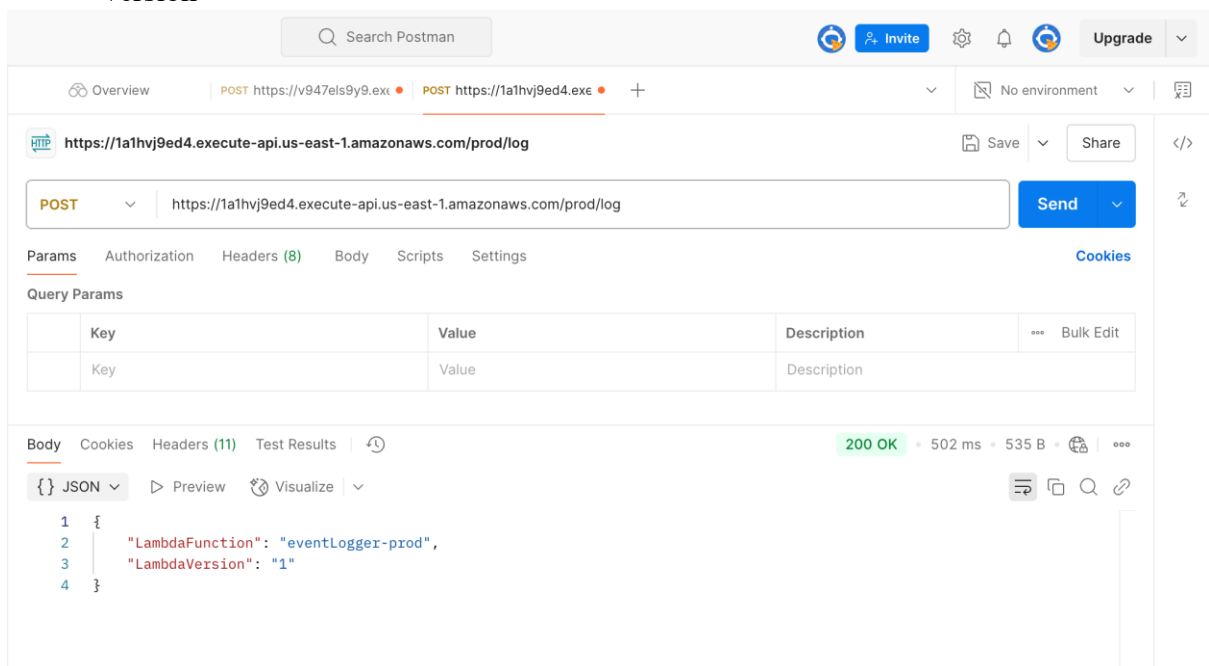


```
event-logger > JS app.mjs > ...
1  /**
2   * Version: v1
3   */
4
5  export const lambdaHandler = async (event, context) => {
6    try {
7      let log = {};
8      log.LambdaFunction = context.functionName;
9      log.LambdaVersion = context.functionVersion;
10
11     console.log("Response:", JSON.stringify(log, null, 2));
12
13     return {
```

```
Value      https://1a1hv9ed4.execute-api.us-east-1.amazonaws.com/prod/log
Key        LambdaFunctionArn
Description Lambda Function ARN
Value      arn:aws:lambda:us-east-1:851725639647:function:eventLogger-prod

Successfully created/updated stack - lab-sam-app-prod in us-east-1
```

- Test using postman to see both test and prod endpoints using the latest Lambda version

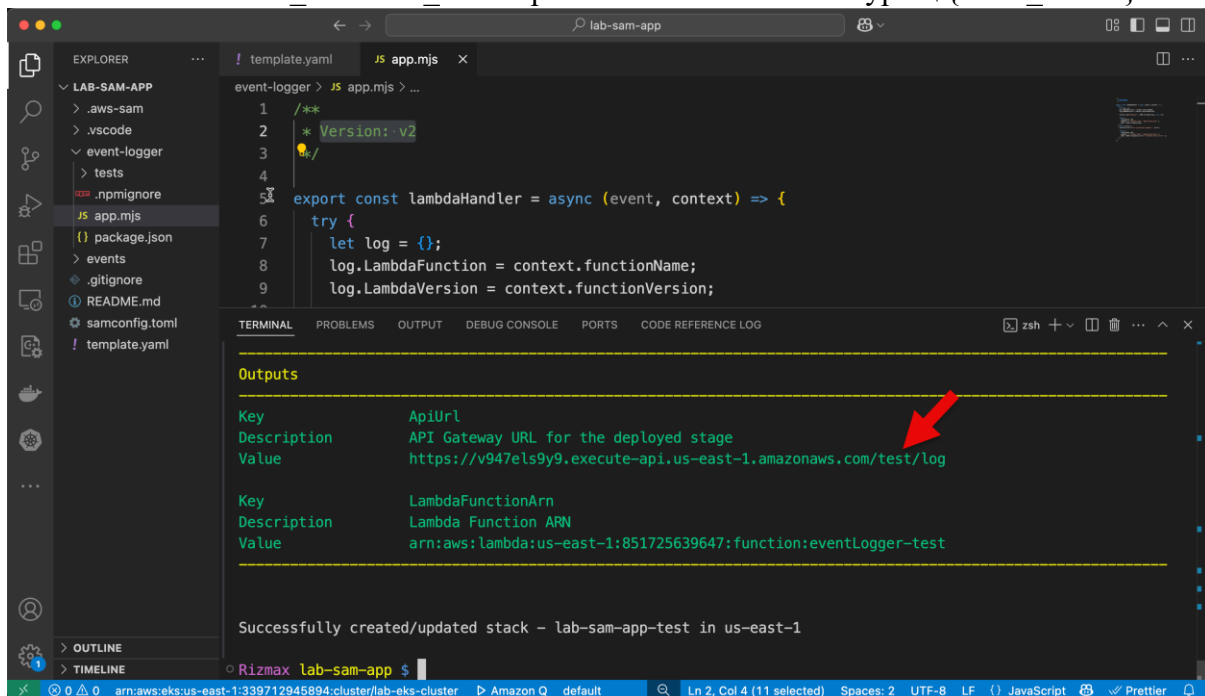


Task 3 - Publish Lambda Versions

Now that the infrastructure is deployed, you will create and test Lambda versioning with traffic shifting.

Publish New Lambda Version

- Make a small change in app.js, such as changing a comment e.g. change v1 to v2:
 1. /**
 2. * Version: v2
 3. */
 4. export const lambdaHandler = async (event, context) => {
- Run the following command to update the Lambda function and deploy again to the test environment.
- Build the application:
 1. sam build
- Deploy the infrastructure for test environment
 1. ENV_TYPE=test
 2. sam deploy --stack-name=lab-sam-app-\${ENV_TYPE} --capabilities CAPABILITY_NAMED_IAM --parameter-overrides EnvType=\${ENV_TYPE}



```
event-logger > JS app.mjs > ...
1  /**
2  * Version: v2
3  */
4
5  export const lambdaHandler = async (event, context) => {
6    try {
7      let log = {};
8      log.LambdaFunction = context.functionName;
9      log.LambdaVersion = context.functionVersion;
```

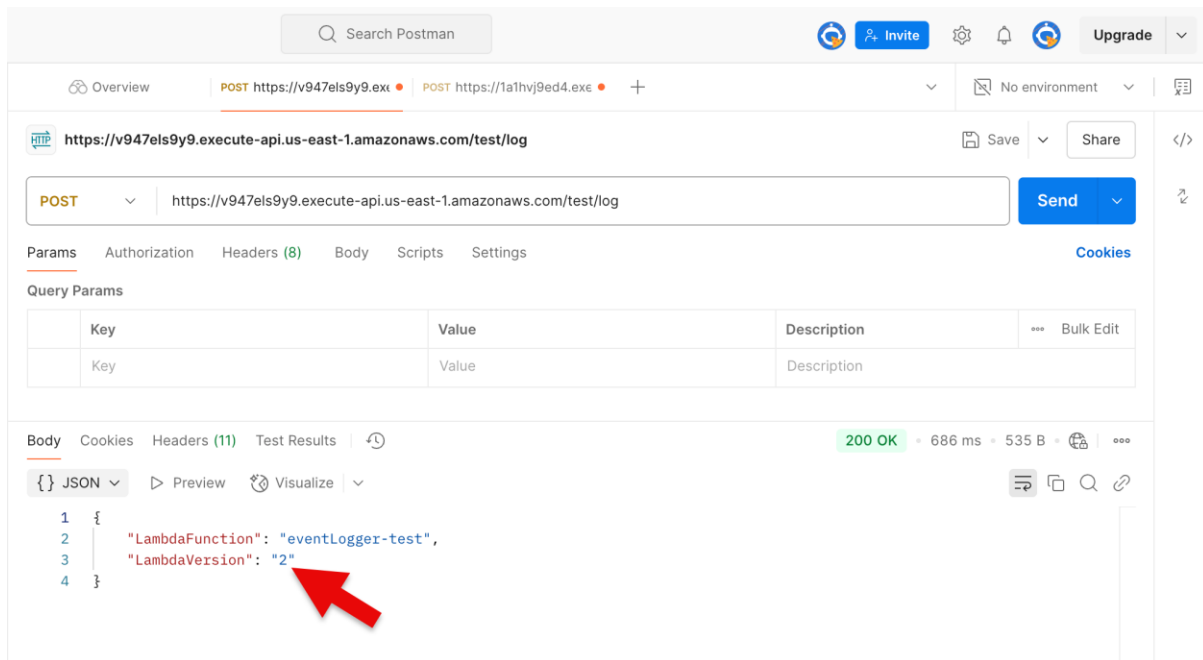
```
Outputs

Key           ApiUrl
Description   API Gateway URL for the deployed stage
Value         https://v947els9y9.execute-api.us-east-1.amazonaws.com/test/log

Key           LambdaFunctionArn
Description   Lambda Function ARN
Value         arn:aws:lambda:us-east-1:851725639647:function:eventLogger-test

Successfully created/updated stack - lab-sam-app-test in us-east-1
```

- Test the API using Postman to verify that the test endpoint now uses the new Lambda version




Task 4 - Test Traffic Shifting with CodeDeploy Deployment

Now, you will test the traffic shifting setup.

- Test the prod API endpoint using Postman.
- Test multiple times. You should see the same LambdaVersion in the response every time.
- Configure DeploymentPreference for the Lambda function, like so:
 1. Resources:
 2. EventLoggerFunction:
 3. Type: AWS::Serverless::Function
 4. Properties:
 5. CodeUri: event-logger/
 6. FunctionName: !Sub "eventLogger-\${EnvType}"
 7. Handler: app.lambdaHandler
 8. Runtime: nodejs22.x
 9. MemorySize: 128
 10. Architectures:
 11. - x86_64
 12. AutoPublishAlias: !Ref EnvType
 13. DeploymentPreference:
 14. Type: Canary10Percent10Minutes

Lambda > Functions > eventLogger-prod

 **eventLogger-prod**
Layers (0)

[+ Add trigger](#)

[+ Add destination](#)

Last modified 5 minutes ago

Function ARN
arn:aws:lambda:us-east-1:851725639647:function:eventLogger-prod

Application
lab-sam-app-prod

Function URL [Info](#)

Code Test Monitor Configuration **Aliases** Versions

Aliases (1) [Info](#)

[Find aliases](#)

[Create alias](#)

Name	Versions	Description
prod	<div>version: 1 (weight=90%) version: 2 (weight=10%)</div>	-

- Now, test the prod API endpoint using Postman.
- Test multiple times. Observe that 10% of responses show new Lambda version, while 90% show the previous version.

Search Postman

Overview POST https://v947els9y9.exe POST https://1a1hvj9ed4.exe +

https://1a1hvj9ed4.execute-api.us-east-1.amazonaws.com/prod/log

POST https://1a1hvj9ed4.execute-api.us-east-1.amazonaws.com/prod/log

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (11) Test Results

200 OK • 117 ms • 535 B

```
{
  "LambdaFunction": "eventLogger-prod",
  "LambdaVersion": "2"
}
```

- Open CodeDeploy console, and open the running deployment to observe traffic shifting progress. If the deployment did not go as expected, it can be stopped or rolled back from here.

Developer Tools **CodeDeploy**

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Deployment**
 - Applications
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings

Go to resource Feedback

Developer Tools > CodeDeploy > Deployments > d-5D73HCIBQ

d-5D73HCIBQ

Stop deployment Stop and roll back deployment

Deployment status

Step 1
Pre-deployment validation
Completed **Succeeded** 100%

Step 2
Traffic shifting
10% complete **In progress** 10%

Step 3
Post-deployment validation
Not started 0%

Traffic shifting progress

Next: the deployment will shift 10% of traffic from the current version to the replacement version at approximately 10 minute(s) after the deployment started.

Original	Replacement
90%	10%

Deployment results info

90% of traffic	10% of traffic
----------------	----------------

Deployment details

Application	Deployment ID	Status
lab-sam-app-prod-ServerlessDeploymentApplication-AOqVOqnFc3rF	d-5D73HCIBQ	In progress
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.LambdaCanary10Percent10Minutes	lab-sam-app-prod-EventLoggerFunctionDeploymentGroup-	User action

- After 10 minutes, traffic will be completely directed to the new Lambda version.

Developer Tools **CodeDeploy**

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Deployment**
 - Applications
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings

Go to resource Feedback

Developer Tools > CodeDeploy > Deployments > d-5D73HCIBQ

d-5D73HCIBQ

Copy deployment Retry deployment

Deployment status

Step 1
Pre-deployment validation
Completed **Succeeded** 100%

Step 2
Traffic shifting
Completed **Succeeded** 100%

Step 3
Post-deployment validation
Completed **Succeeded** 100%

Traffic shifting progress

Next: the deployment will shift 10% of traffic from the current version to the replacement version at approximately 10 minute(s) after the deployment started.

Original	Replacement
0%	100%

Deployment results info

0% of traffic	100% of traffic
---------------	-----------------

Deployment details

Application	Deployment ID	Status
lab-sam-app-prod-ServerlessDeploymentApplication-AOqVOqnFc3rF	d-5D73HCIBQ	Succeeded
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.LambdaCanary10Percent10Minutes	lab-sam-app-prod-EventLoggerFunctionDeploymentGroup-	User action

- Review the Lambda function Aliases tab in Lambda console. You should see traffic being shifted completely to the new Lambda version.

Developer Tools **CodeDeploy**

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Deployment**
 - Applications
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings

Go to resource
Feedback

Developer Tools > CodeDeploy > Deployments > d-5D73HCIQB

d-5D73HCIQB

[Copy deployment](#) [Retry deployment](#)

Deployment status

Step 1
Pre-deployment validation
Completed ✓ Succeeded 100%

Step 2
Traffic shifting
Completed ✓ Succeeded 100%

Step 3
Post-deployment validation
Completed ✓ Succeeded 100%

Traffic shifting progress

Next: the deployment will shift 10% of traffic from the current version to the replacement version at approximately 10 minute(s) after the deployment started.

Original	Replacement
0%	100%

Deployment results info

0% of traffic	100% of traffic
---------------	-----------------

Deployment details

Application lab-sam-app-prod-ServerlessDeploymentApplication-AQoVQnFc3rF	Deployment ID d-5D73HCIQB	Status ✓ Succeeded
Deployment configuration CodeDeployDefault.LambdaCanary10Percent10Minutes	Deployment group lab-sam-app-prod-EventLoggerFunctionDeploymentGroup-	Initiated by User action

- Test the prod API endpoint using Postman, and you should see the new version 100% of the times.

Search Postman

Overview POST https://v947els9y9.exe POST https://1a1hvj9ed4.exe + No environment

https://1a1hvj9ed4.execute-api.us-east-1.amazonaws.com/prod/log

POST https://1a1hvj9ed4.execute-api.us-east-1.amazonaws.com/prod/log Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (11) Test Results 200 OK 106 ms 535 B

JSON Preview Visualize

```

1 {
2   "LambdaFunction": "eventLogger-prod",
3   "LambdaVersion": "2"
4 }
```

- You may also experiment with Linear10PercentEvery1Minute as DeploymentPreference which would shift traffic to the new version in 10% increments every minute.

Task 6 - Clean Up Resources

As a best-practice, you may want to delete the resources created during the lab if no longer needed.

Delete the AWS SAM Stack

- Delete all deployed resources using the following commands. Enter y for any confirmation prompts.
1. `ENV_TYPE=test`
 2. `sam delete --stack-name=lab-sam-app-${ENV_TYPE}`
 1. `ENV_TYPE=prod`
 2. `sam delete --stack-name=lab-sam-app-${ENV_TYPE}`

```
! template.yaml x JS app.mjs
! template.yaml > {} Resources > {} ApiGateway > {} Properties > {} DefinitionBody > {} paths > {} /log
AWS CloudFormation Serverless Application Model (SAM) - The AWS Serverless Application Model (AWS SAM, previously known as Project
1 AWSTemplateFormatVersion: "2010-09-09"
2 Transform: AWS::Serverless-2016-10-31
3
4 Parameters:
5 EnvType:

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS CODE REFERENCE LOG
Rizmax lab-sam-app $ ENV_TYPE=test
sam delete --stack-name=lab-sam-app-${ENV_TYPE}
Are you sure you want to delete the stack lab-sam-app-test in the region us-east-1 ? [y/N]: y
Do you want to delete the template file 16a3c5b002e0bf068252f8be3a5566b1.template in S3? [y/N]: y
- Deleting S3 object with key 9681d99e49170b208e3cf60fcbfadb8b
- Deleting S3 object with key 16a3c5b002e0bf068252f8be3a5566b1.template
- Deleting Cloudformation stack lab-sam-app-test

Deleted successfully
Rizmax lab-sam-app $ ENV_TYPE=prod
sam delete --stack-name=lab-sam-app-${ENV_TYPE}
Are you sure you want to delete the stack lab-sam-app-prod in the region us-east-1 ? [y/N]: y
Do you want to delete the template file d2f2dc8e3ab1321c9239842733836e04.template in S3? [y/N]: y
- Could not find and delete the S3 object with the key 9681d99e49170b208e3cf60fcbfadb8b
- Deleting S3 object with key d2f2dc8e3ab1321c9239842733836e04.template
- Deleting Cloudformation stack lab-sam-app-prod

Deleted successfully
Rizmax lab-sam-app $
```