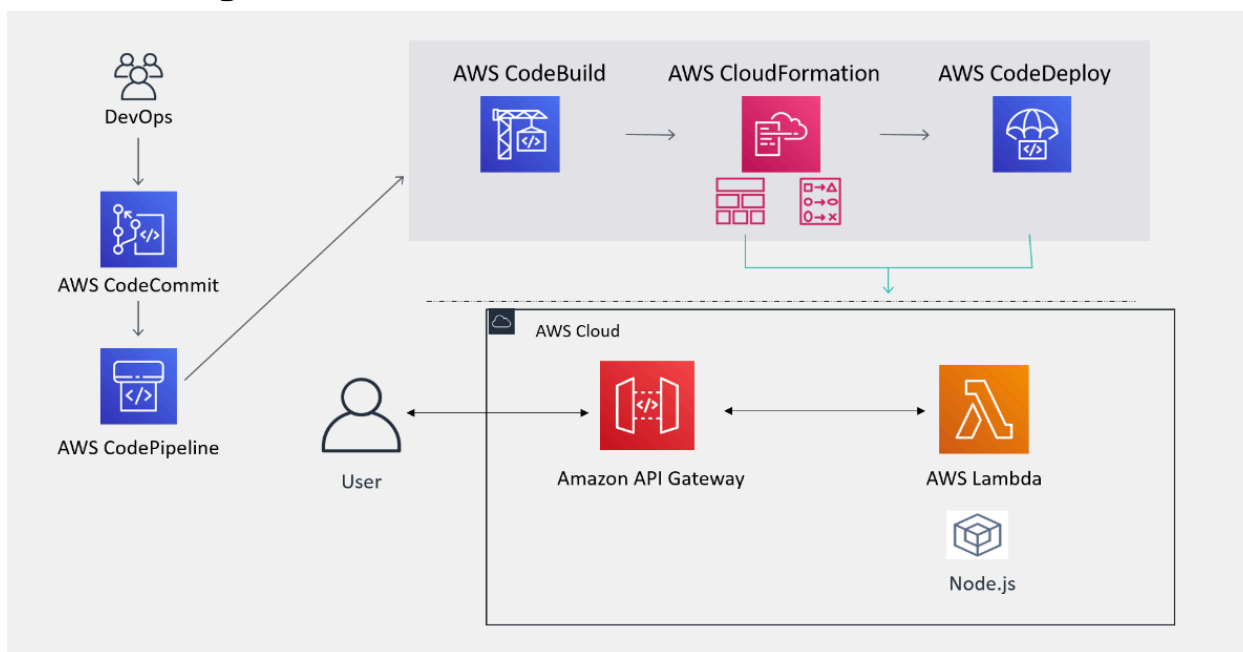


Lab 4: Deploying a serverless application using AWS SAM and a CI/CD pipeline

Objectives:

1. Build and locally test your application.
2. Use AWS SAM to publish a serverless application.
3. Automate deployment with AWS CodeDeploy and AWS CodePipeline.
4. Implement traffic shifting deployment to gradually roll out updates.

Overview Diagram:



Explanation:

1. **Code Commit:** The DevOps team pushes code changes to the AWS CodeCommit repository.
2. **Pipeline Trigger:** AWS CodePipeline detects the change in the CodeCommit repository and starts the pipeline.
3. **Build Stage:** CodePipeline triggers AWS CodeBuild to compile the code, run tests, and prepare the deployment package.

4. Infrastructure Provisioning: After a successful build, AWS CodePipeline uses AWS CloudFormation to provision or update the necessary infrastructure, ensuring all resources (API Gateway, Lambda functions) are correctly set up.
5. Deployment: AWS CodePipeline then triggers AWS CodeDeploy to deploy the new version of the application.
6. AWS CodeDeploy
A service that automates application deployments to various compute services, including AWS Lambda. It can implement a **traffic shifting strategy**, gradually shifting traffic to the new version to ensure stability and allow for quick rollback if needed.
7. User Interaction: The end user interacts with the application via the Amazon API Gateway, which routes the request to the AWS Lambda function.
8. The Lambda function processes the request and returns a response to the user.

Objective 1 - AWS SAM

- 1.1 Open Cloud9 IDE and run the following command to initialize a SAM project



```
sam init --runtime python3.9
```

- 1.2 Choose the options to get a hello world example project from AWS quick start templates, and choose the name - 'sam-app'

```
Which template source would you like to use?
  1 - AWS Quick Start Templates
  2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
  1 - Hello World Example
  2 - Hello World Example with Powertools for AWS Lambda
  3 - Infrastructure event management
  4 - Multi-step workflow
  5 - Lambda EFS example
  6 - Serverless Connector Hello World Example
  7 - Multi-step workflow with Connectors
Template: 1
```

1.3 The app will be cloned from git

```
Project name [sam-app]:

Cloning from https://github.com/aws/aws-sam-cli-app-templates (process may take a moment)

-----
Generating application:
-----
Name: sam-app
Runtime: python3.9
Architectures: x86_64
Dependency Manager: pip
Application Template: hello-world
Output Directory: .
Configuration file: sam-app/samconfig.toml

Next steps can be found in the README file at sam-app/README.md
```

1.4 Go to the app directory and perform the build (of the app)

```
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment $ cd ./sam-app
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam build
Starting Build use cache
Manifest file is changed (new hash: 3298f13049d19cffaa37ca931dd4d421) or dependency folder
(.aws-sam/deps/619eae1-857e-4f33-a52c-f3b6e6484d64) is missing for (HelloWorldFunction), downloading dependencies and
copying/building source
Building codeuri: /home/ec2-user/environment/sam-app/hello_world runtime: python3.9 metadata: {} architecture: x86_64
functions: HelloWorldFunction
Running PythonPipBuilder:CleanUp
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
Running PythonPipBuilder:CopySource
```

1.5 Test your application locally. Use AWS SAM to test your application, when reviewing the test output, look for a status code 200

```

AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam local invoke HelloWorldFunction --event events/event.json
Invoking app.lambda_handler (python3.9)
Local image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.9
Building image.....
.....
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /home/ec2-user/environment/sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated, inside runtime container
START RequestId: 203a9430-6154-420e-918d-7c07cbac4f0e Version: $LATEST
END RequestId: 824f3060-f951-4acb-89a7-fc1262e3aa1d
REPORT RequestId: 824f3060-f951-4acb-89a7-fc1262e3aa1d Init Duration: 0.04 ms Duration: 48.36 ms Billed Duration: 49 ms Memory Size: 128 MB
Max Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $

```

Meaning of SC 200-

HTTP Status Code 200

Status Code 200: OK

- **Class:** Successful responses (2xx)
- **Meaning:** The request has succeeded.

1.6 Run the following command to test the application

```

AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam local invoke HelloWorldFunction
Invoking app.lambda_handler (python3.9)
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /home/ec2-user/environment/sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated, inside runtime container
START RequestId: 071c0d52-44ed-455b-80f1-93240d531321 Version: $LATEST
END RequestId: 9e061d25-ba4b-40aa-ab22-de0e7f0fb1cd
REPORT RequestId: 9e061d25-ba4b-40aa-ab22-de0e7f0fb1cd Init Duration: 0.04 ms Duration: 48.36 ms Billed Duration: 49 ms Memory Size: 128 MB
Max Memory Used: 128 MB
{"statusCode": 200, "body": "{\"message\": \"hello world\"}"}

```

It again produces SC200

Meaning test has been successful

1.7 Start the application

```

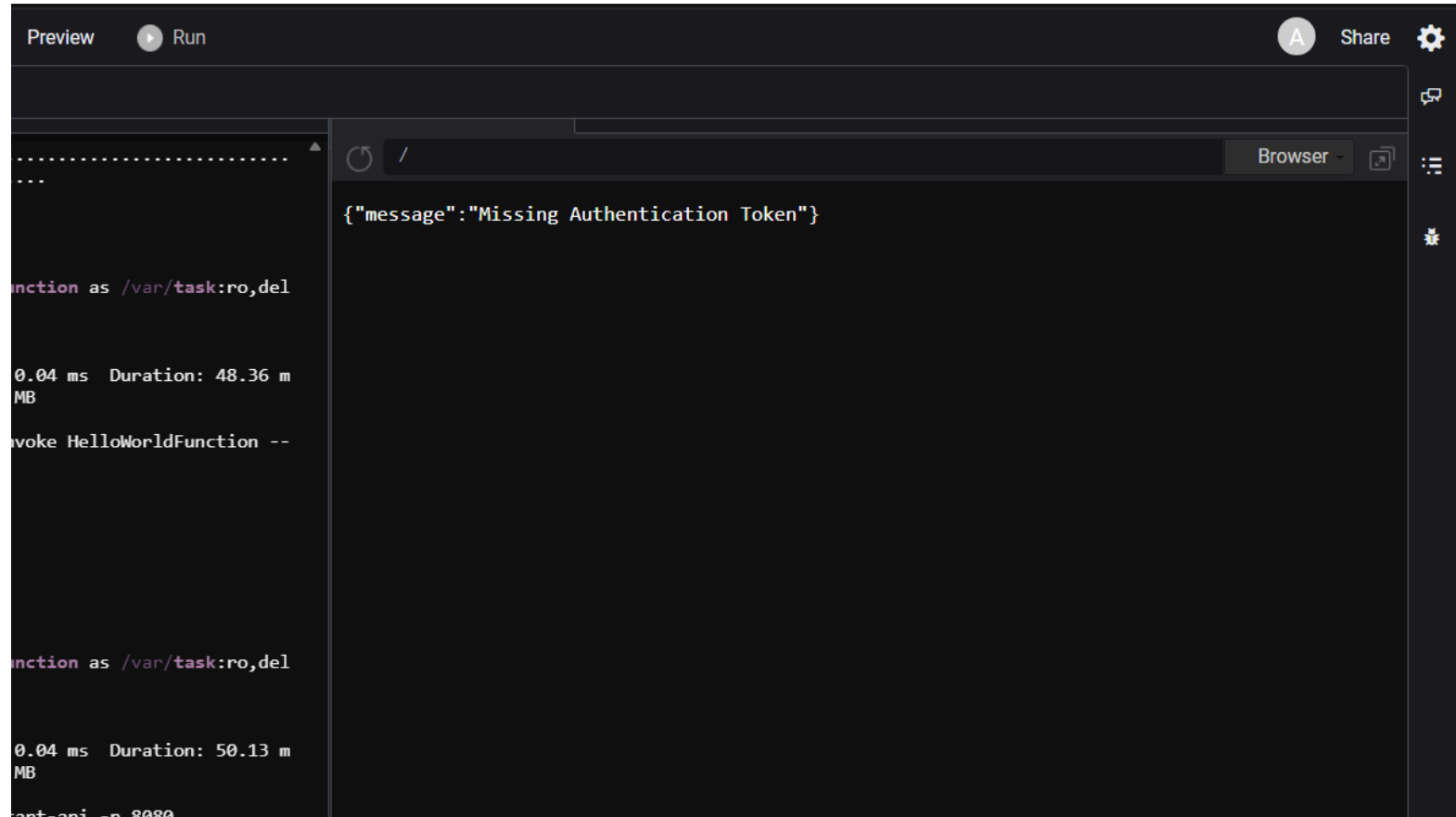
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam local start-api -p 8080
Initializing the lambda functions containers.
Local image is up-to-date
Using local image: public.ecr.aws/lambda/python:3.9-rapid-x86_64.

Mounting /home/ec2-user/environment/sam-app/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated, inside runtime container
Containers Initialization is done.
Mounting HelloWorldFunction at http://127.0.0.1:8080/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do not need to restart/reload SAM CLI while working on your functions,
changes will be reflected instantly/automatically. If you used sam build before running local commands, you will need to re-run sam build for the
changes to be picked up. You only need to restart SAM CLI if you update your AWS SAM template
2024-05-20 21:18:36 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8080
2024-05-20 21:18:36 Press CTRL+C to quit

```

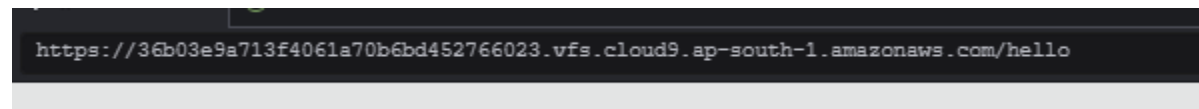
Now: The start-api command directs you to browse to port 8080 on the local system loopback address (127.0.0.1). AWS Cloud9 does not allow this and you receive an error if you use the address included in the command output. AWS Cloud9 does offer a function to securely view your application through the Preview feature.

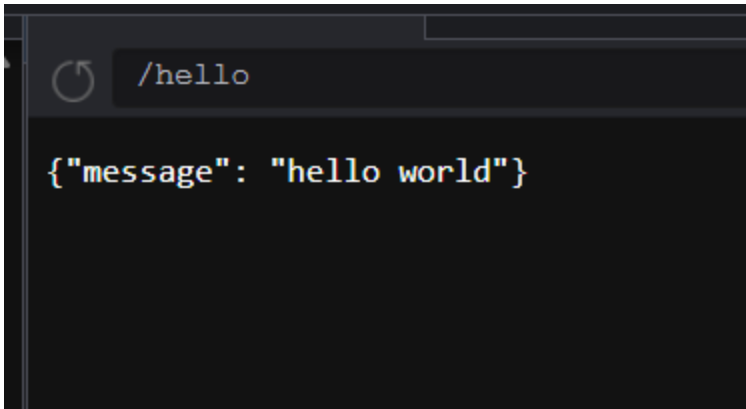
1.8 Preview the app, by clicking on preview option on top of Cloud9 UI



This is the expected output as per Lab instructions

1.9 Enter “hello” at the end of address bar





App run Test - successful

Objective 2 - USING AWS SAM TO DEPLOY THE APPLICATION

2.1 To deploy the hello world application the first step is to create an (Amazon S3) bucket to host your application.

```
Y:~/environment/sam-app $ labBucket=lab4-sam-aggoyal-400001
```

2.2 Use the variable to create a globally unique bucket

```
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ aws s3 mb s3://$labBucket  
make_bucket: lab4-sam-aggoyal-400001
```

2.3 Run the following command to package your application and push it to the S3 bucket you created

```
make_bucket: lab4-sam-aggoyal-400001  
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam package --output-template-file packaged.yaml --s3-bucket $labBucket  
Uploading to 2f56c9887400c8719a3177a4f222bd7d 557854 / 557854 (100.00%)  
  
Successfully packaged artifacts and wrote output template to file packaged.yaml.  
Execute the following command to deploy the packaged template  
sam deploy --template-file /home/ec2-user/environment/sam-app/packaged.yaml --stack-name <YOUR STACK NAME>
```

2.4 Run the command below to deploy your application package

```
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/sam-app $ sam deploy --template-file packaged.yaml --stack-name sam-app --capabilities CAPABILITY_IAM
Creating the required resources...
Successfully created!

Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-0e58wbo9yc1d
A different default S3 bucket can be set in samconfig.toml
Or by specifying --s3-bucket explicitly.

Deploying with following values
=====
Stack name           : sam-app
Region              : ap-south-1
Confirm changeset    : True
Disable rollback     : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-0e58wbo9yc1d
Capabilities         : ["CAPABILITY_IAM"]
Parameter overrides  : {}
Signing Profiles     : {}

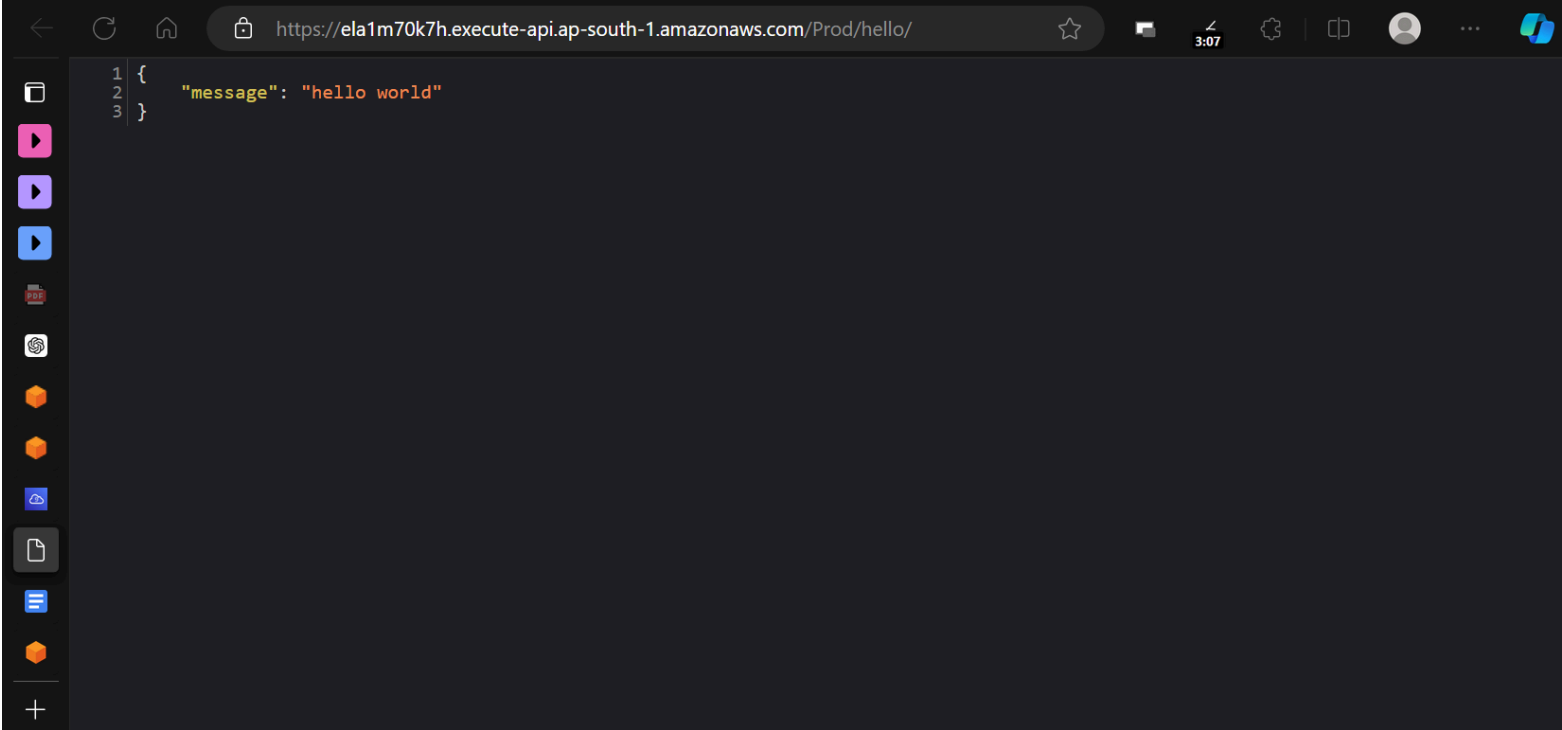
Initiating deployment
=====

Uploading to c205beb0912ff0c6472c93f662b78539.template 1155 / 1155 (100.00%)
```

Successfully created/updated stack - sam-app in ap-south-1

```
dApi`]' --output table
-----
|                                     DescribeStacks                                     |
|-----|-----|-----|
| Description                        | OutputKey | OutputValue |
|-----|-----|-----|
| API Gateway endpoint URL for Prod stage for Hello World function | HelloWorldApi | https://ela1m70k7h.execute-api.ap-south-1.amazonaws.com/Prod/hello/ |
|-----|-----|-----|
```

2.5 Click on the link, in a new tab

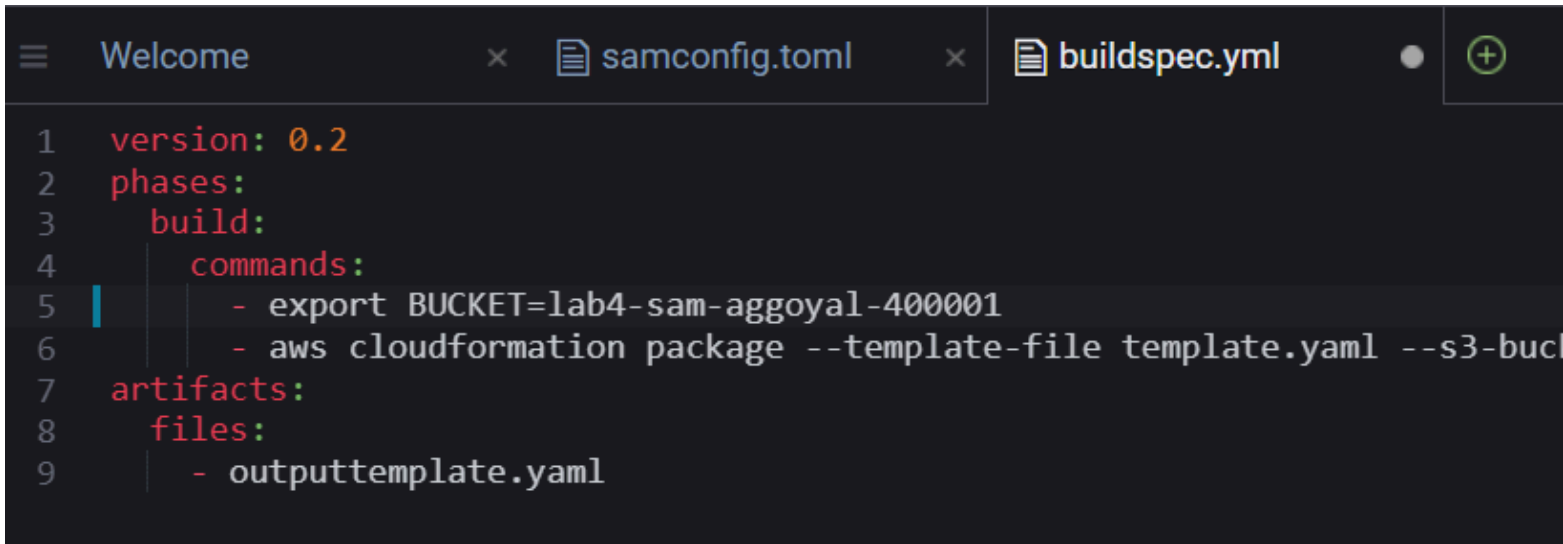


Deployed, successfully

Objective 3: Automate application deployment with AWS CodePipeline

Note - For this task, you create a CI/CD pipeline to automate the deployment of an application. You create a three-stage pipeline that uses AWS CodeCommit, AWS CodeBuild, and AWS CloudFormation. Your initial HelloWorld application was initialized using Python as the language but you've chosen to change to Node.js for the deployment. You overwrite your existing SAM deployment with this new one. The new SAM project is in the Cloud9 environment. You need to add a new bucket for your deployment and make the appropriate code changes before you can automate the deployment.

3.1 Go to the lab4 directory, verify the bucket name and put the bucket name in buildspec.yml file

A screenshot of a code editor interface with a dark theme. At the top, there are three tabs: 'Welcome', 'samconfig.toml', and 'buildspec.yml'. The 'buildspec.yml' tab is active, showing a YAML configuration. The code is as follows:

```
1 version: 0.2
2 phases:
3   build:
4     commands:
5       - export BUCKET=lab4-sam-aggoyal-400001
6       - aws cloudformation package --template-file template.yaml --s3-buc
7 artifacts:
8   files:
9     - outputtemplate.yaml
```

3.2 Update the remote AWS CodeCommit repository by running the following commands


```
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/lab4-app (main) $ git add .
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/lab4-app (main) $ git commit -m "Updated buildspec file with bucket name"
[main 33a52f8] Updated buildspec file with bucket name
Committer: EC2 Default User <ec2-user@ip-10-187-10-217.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

1.64KB git config --global user.name "Your Name"
git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/lab4-app (main) $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 348 bytes | 348.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/lab4-app
4a2683b..33a52f8 main -> main
AWSLabsUser-hou7z9uipSGc4sUddFZL8Y:~/environment/lab4-app (main) $
```

Objective 4 - CREATE THE CONTINUOUS DELIVERY PIPELINE FOR YOUR LAB4 APPLICATION

4.1 Open AWSCodePipeline for this

4.2 Create Pipeline, using the following SSs:

Developer Tools > CodePipeline > Pipelines

Pipelines Info

Notify

View history

Release change

Delete pipeline

Create pipeline

< 1 >

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
------	-------------------------	-------------------------	--------------------------	------------------------



Let all values remain the same

Click next

4.3 choose AWS CodeCommit from the drop-down menu for Source provider.

Step 2 of 5

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit
Amazon ECR AWS CodeCommit
Amazon S3
Bitbucket
GitHub (Version 1)
GitHub (Version 2)
GitHub Enterprise Server
GitLab
GitLab self-managed

PreviousNext

4.4 Update repo and branch name

Repository name

Choose a repository that you have already created where you have pushed your source code.

Branch name

Choose a branch of the repository

4.5 Choose AWS CodeBuild in build stage

Build - optional

Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

Add Jenkins


AWS CodeBuild


Skip build stage

Next

4.6 Do the following:

56. For **Project name** , choose [Create project](#) .

 **Note:** A new browser window opens for creating a CodeBuild project.

57. For **Project name** , enter  `lab4-sam-build`

58. In the **Environment** card, configure the following values:

- For **Environment image** , choose **Managed image**
- For **Operating system** , choose **Ubuntu** from the drop-down menu
- For **Runtime(s)** , choose **Standard** from the drop-down menu
- For **Image** , choose **aws/codebuild/standard: 7.0** from the drop-down menu
- For **Image version** , choose **Always use the latest image for this runtime version** from the drop-down menu
- For **Service role** , select **Existing service role**
- For **Role ARN** , choose (it auto-updates with the ARN)
- For **Build specifications** choose ☒ **Use a buildspec file** .

59. In the **Logs** sections, keep the default values.


60. Choose [Continue to CodePipeline](#) .

Output -

Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

 or [Create project](#)

 Successfully created lab4-sam-build in CodeBuild.

4.7 For Deploy provider, choose AWS CloudFormation from the drop-down menu.

Info

Step 4 of 5

Deploy - optional

Deploy provider

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

A screenshot of the AWS IAM console search results. At the top, there is a search bar with a magnifying glass icon. Below the search bar, a list of search results is displayed. The results are: 'AWS AppConfig', 'AWS CloudFormation' (which is highlighted with a light gray background), 'AWS Cloud' (with a dark gray tooltip box containing the text 'AWS CloudFormation'), 'AWS CodeDeploy', 'AWS Elastic Beanstalk', 'AWS OpsWorks Stacks', 'AWS Service Catalog', 'Amazon ECS', 'Amazon ECS (Blue/Green)', and 'Amazon S3'. The list is scrollable, as indicated by a vertical scrollbar on the right side.

skip deploy stage

Next

Note -

Your Deploy stage still needs some work but you have to fix that after the pipeline has been created. When you create the pipeline, it immediately starts running the steps. If the pipeline starts running, it wouldn't hurt anything, it would just lead to unnecessary waiting. Please continue directly to the **next step** after you choose **Create pipeline**.

4.8 Review and create the pipeline

The pipeline lab4-sam-pipeline has been created.

4.9 Choose stop execution

Success

Congratulations! The pipeline lab4-sam-pipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > lab4-sam-pipeline

lab4-sam-pipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: **V2** Execution mode: **QUEUED**

Source

In progress

Source

[AWS CodeCommit](#)

In progress - 1 minute from now

View details

Disable transition

4.10 Select the default values

Stop execution ✕

Select execution

Choose the pipeline execution you want to stop.

f964ca49-7bdd-4060-b48d-302b7fbc2d1f

In progress ▼

33a52f89: Updated buildspec file with bucket name
Stage: Build

Choose a stop mode for the execution

If you choose to stop and wait, and you change your mind while your execution is still in a stopping state, you can choose to abandon.

☐ **Stop and wait**
Wait until all in-progress actions are complete.

☒ **Stop and abandon**
Don't wait until the in-progress actions are complete.
Warning: This option can lead to failed actions.

Comments - optional ☐ Preview markdown [Learn more](#) 

Cancel

Stop

4.11

✔ Success

Create a notification rule for this pipeline ✕

Congratulations! The pipeline lab4-sam-pipeline has been created.

✔ Success ✕

Stop execution f964ca49-7bdd-4060-b48d-302b7fbc2d1f successfully

[Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > lab4-sam-pipeline

lab4-sam-pipeline

4.12 Now you can complete your deploy stage. The current configuration creates an AWS CloudFormation change set but you still need to run it to shift traffic to the new version of your application.

4.13 Select the edit button, and choose edit stage

+

Add stage

Edit: Deploy

Edit stage

Deploy

AWS CloudFormation

☐

Configure automatic rollback on stage failure

+

Add stage

4.14 Select add action group and select the following values

Edit action

Action name

Choose a name for your action

deploy-changeset

No more than 100 characters

Action provider

AWS CloudFormation

Region

Asia Pacific (Mumbai)

4.15 Enter the following details

Input artifacts

Choose an input artifact for this action. [Learn more](#)

BuildArtifact

Add

No more than 100 characters

Action mode

When you update an existing stack, the update is permanent. When you use a change set, the result provides a diff of the updated stack and the original stack before you choose to execute the change.

Execute a change set

Stack name

If you are updating an existing stack, choose the stack name.

Q sam-app

Change set name

If you are updating an existing change set, choose the change set name.

Q lab4-sam-changeset

4.16 Click on Done->Done->Save->Save

✓ Success

Pipeline was saved successfully.

4.17 Choose the Release Change option

lab4-sam-pipeline

🔔 Notify ▼

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2 Execution mode: QUEUED

4.18

✓ Success

The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

4.19 All steps of the pipeline have been completed

lab4-sam-pipeline

Pipeline type: **V2** Execution mode: **QUEUED**

✔ **Source** Succeeded

Pipeline execution ID: [e2b4cece-84d0-413a-bd54-2d453bf70b5c](#)

Source

[AWS CodeCommit](#)

✔ Succeeded - Just now

[33a52f89](#)

View details

[33a52f89](#) Source: Updated buildspec file with bucket name

Disable transition

✔ **Build** ⚠ Succeeded

Pipeline execution ID: [e2b4cece-84d0-413a-bd54-2d453bf70b5c](#)

Build

[AWS CodeBuild](#)

✔ Succeeded - Just now

View details

[33a52f89](#) Source: Updated buildspec file with bucket name

Disable transition

✔ **Deploy** ⚠ Succeeded

Pipeline execution ID: [e2b4cece-84d0-413a-bd54-2d453bf70b5c](#)

Deploy

[AWS CloudFormation](#)

✔ Succeeded - Just now

View details



deploy-changeset

[AWS CloudFormation](#)

✔ Succeeded - 1 minute from now

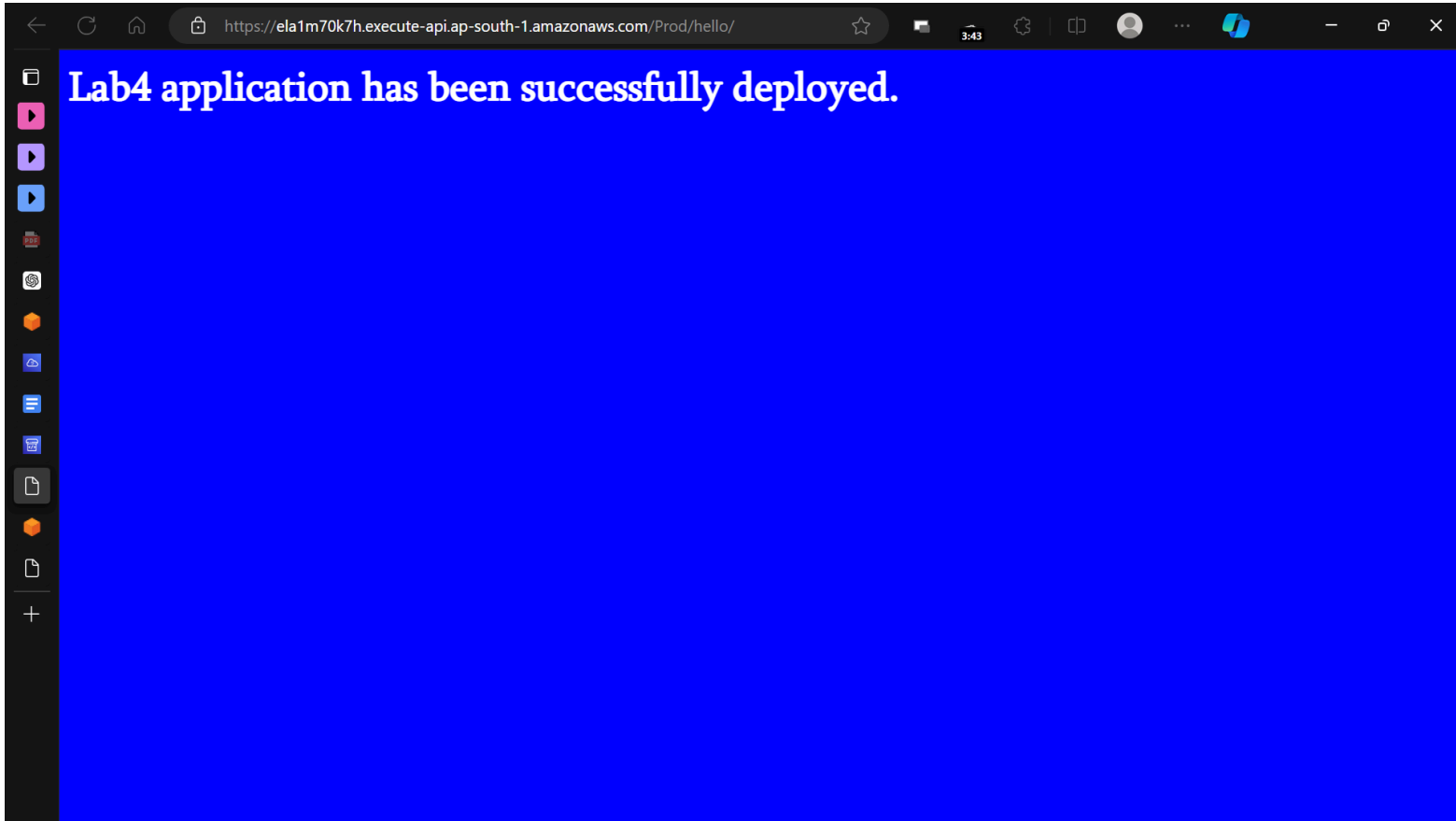
View details

[33a52f89](#) Source: Updated buildspec file with bucket name

DOUBT - Where did we implement logic for Shifting Traffic here?

Objective 5 - Verify the deployment, by clicking on the link in deploy-changeset card in above step.

Output-



Objective 6 - Make a new change to the application and re-deploy, to check for shifting traffic

6.1 For making the changes, I'll just do the steps given -

Which includes making some change in code for app, and pushing the change in git, and not show them here, as that is not our main motive, our main motive is the deploy the app with those changes, and the observe the traffic shift

6.2 Open the pipeline window, and notice that the pipeline has observed changes in the app, and automatically, it initiates a re-build of the lab4 application.

Success Pipeline was saved successfully.

Success The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Developer Tools > CodePipeline > Pipelines > lab4-sam-pipeline

lab4-sam-pipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: [c3fc359c-bbe8-4d95-8d00-46f86f3fc8f4](#)

Source

[AWS CodeCommit](#)

Succeeded - Just now
[b509f8d0](#)

View details

[b509f8d0](#) Source: Updated app.js file to display lab4b html page

Disable transition

Deploy In progress

Pipeline execution ID: [c3fc359c-bbe8-4d95-8d00-46f86f3fc8f4](#)

Deploy

[AWS CloudFormation](#)

Succeeded - Just now

View details



deploy-changeset

[AWS CloudFormation](#)

In progress - Just now

View details

[b509f8d0](#) Source: Updated app.js file to display lab4b html page

6.3 View deployment history from left pane

[Developer Tools](#) > [CodeDeploy](#) > Deployments

Deployment history

🔄

View details

Actions ▾

Copy deployment

Retry deployment

< 1 > ⚙️

	Deployment Id	Status	Deploym...	Compute ...	Application	Deploym...	Revision L...	Initiating ...	Start ti
○	d-0VBXGD035	✔ Succeeded	Blue/green	AWS Lam...	sam-app-...	sam-app-...	08ac8a7d...	User action	May 21

6.4 Clicking on the deployment ID will show the following

d-0VBXGD035

🔄

Copy deployment

Retry deployment

Deployment status

Step 1
Pre-deployment validation

100%

Completed ✔ Succeeded

Step 2
Traffic shifting

100%

Completed ✔ Succeeded

Step 3
Post-deployment validation

100%

Completed ✔ Succeeded

Traffic shifting progress

The deployment will shift 10% of traffic from the current version to the replacement version every 1 minute(s) until all of the traffic is routed to the new version.

Original

0%

Deployment results Info

0% of traffic

Replacement

100%

100% of traffic