

- Create two Lambda Functions
- First Lambda function returns 200 Response as {"Hello": "Default"}
- Second Lambda function returns 200 Response as {"Hello": "{Dynamic route name}"}
- Configure API Gateway with that hits first lambda function on / and the second lambda function on /*

Two lambda functions were created for this assignment:

First one is bijaykandel-hello

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

 Python 3.9

Architecture Info
Choose the instruction set architecture you want for your function code.

x86_64
 arm64

And its test event is configured as default.

Configure test event X

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

Create new test event
 Edit saved test events

Event template

Event name

```
1 -> []
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 ]
```

The second lambda function is named as bijaykandel-helloDynamic

Basic information

Function name
Enter a name that describes the purpose of your function.
bijaykandel-helloDynamic

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.9

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

And its test event is configured as follows: (“rawPath” option is added in the existing default document)

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

Create new test event
 Edit saved test events

Saved Test Events

bijay-helldynamic

```
1 ↵ []
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3",
5   "rawPath": "path"
6 ]
```

First lambda function is written as:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps({'Hello': 'General'})
8     }
```

And the second lambda function is written as:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps(event['rawPath'])
8     }
```

Now API gateway is created: **API Gateway >> Create API gateway:**

We added two integrations for two lambda functions as follows:

The screenshot shows the 'Integrations' section of the AWS API Gateway console. It displays two entries for Lambda functions:

- Lambda**: AWS Region: us-east-1, Lambda function: bijaykandel-hello, Version: 2.0.
- Lambda**: AWS Region: us-east-1, Lambda function: bijaykandel-helloDynamic, Version: 2.0.

At the bottom left is a button labeled 'Add integration'.

And Routes are configured so that accessing the URL triggers first lambda function and accessing URL/* triggers the second lambda function.

The screenshot shows the 'Configure routes' section of the AWS API Gateway console. It includes a descriptive text about routes and two route configurations:

Method	Resource path	Integration target
ANY	/	bijaykandel-hello
-	\$default	bijaykandel-helloDynamic

At the bottom left is a button labeled 'Add route'.

After creating the API gateway, we can click on the link inside our API gateway. With no path specified, the url returns the following value:

The screenshot shows a browser window with the following details:

- Title bar: API Gateway
- Address bar: https://rd8sa3iymc.execute-api.us-east-1.amazonaws.com
- Content area:

```
{"Hello": "General"}
```

And if path is provided with url, it returns the given path as shown below:

The screenshot shows a browser window with the following details:

- Title bar: API Gateway
- Address bar: https://rd8sa3iymc.execute-api.us-east-1.amazonaws.com/abd
- Content area:

```
"/abd"
```

- Create a bash script to deploy your lambda functions
- Create a bash script to deploy your react app to S3
- Integrate both these scripts with one of Jenkins, Github Actions, CircleCI or TravisCI

To deploy react app to s3, following bash script file is created:

Sudo nano cicd.sh

```
GNU nano 4.8                                         cicd.sh
#!/bin/bash
# Create a new React app named reactapp
npx create-react-app reactapp
# Change directory to the newly created app
cd reactapp/
# Inside the app directory, build the static files
echo 'inside reactapp building static files'
npm run build
echo 'Build completed'

# Upload the static files to AWS S3 via CLI
echo 'uploading static files of build folder to aws via cli'
aws s3 cp build s3://intern-bijaykandel37 --recursive
echo 'uploading static files completed'

# Provide instructions for viewing the deployed app
echo 'Now you can go to s3 bucket and see the static url link in index.js object'
echo 'OR'
echo 'click on the link below to see the hosted app'
aws s3 presign s3://intern-bijaykandel37/index.html
```

Making it executable: ***sudo chmod +x cicd.sh***

Running the script file with command: ***./cicd.sh***

```
The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:
  https://cra.link/deployment

Build completed
uploading static files of build folder to aws via cli
upload: build/static/css/main.a617e044.chunk.css.map to s3://intern-bijaykandel37/static/css/main.a617e044.chunk.css.map
upload: build/main.css to s3://intern-bijaykandel37/main.css
```

And we can get the link of index.html file as shown:

```
upload: build/static/js/2.6bdd5a89.chunk.js.map to s3://intern-bijaykandel37/static/js/2.6bdd5a89.chunk.js.map
uploading static files completed
Now you can go to s3 bucket and see the static url link in index.js object
OR
click on the link below to see the hosted app
https://intern-bijaykandel37.s3.us-east-1.amazonaws.com/index.html?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA52BEGI3BCCP6PEYVC%2F20211215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20211215T182738Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=dda4a875bee1dad71388cf51e0c86d5
bj@batman:~/react/react-bijay$ 
```

When we go to the link, we can see our page being hosted statically from s3.

