**Creating a Simple AWS Lambda Function**

**Introduction**

In this hands-on lab, we will create and customize a simple nodejs -based AWS Lambda function using AWS Cloud9.

**Solution**

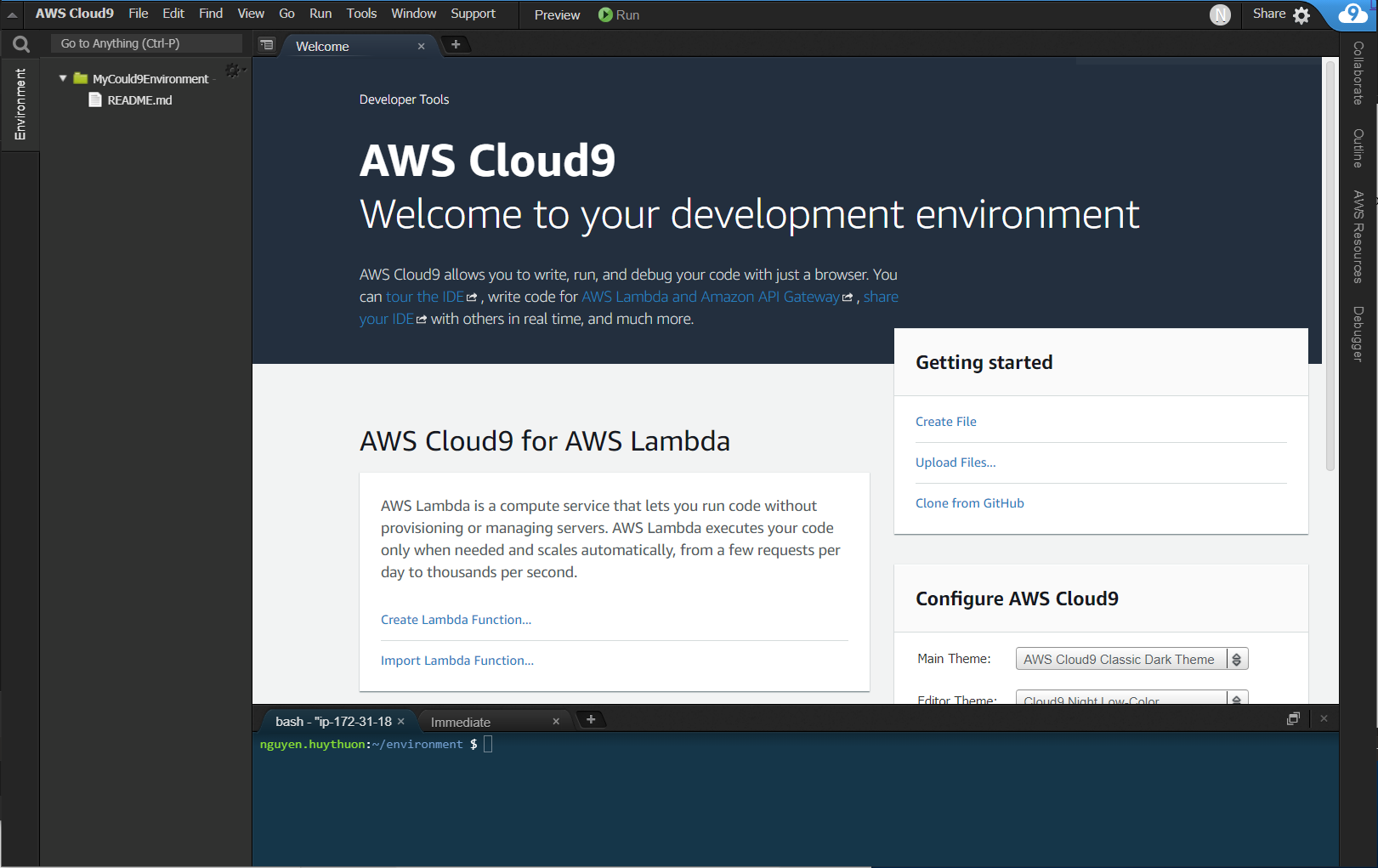
Log in to the live AWS environment using your credentials with free tier subscription. Make sure you're in the South East Asia (ap-southeast-1) region throughout the lab.

**Create a Cloud9 Environment**

1. Navigate to **Cloud9**: <https://ap-southeast-1.console.aws.amazon.com/cloud9>.
2. Click **Create environment**.
3. Enter environment **Name** and **Description**.
4. Click **Next step**.
5. Make sure the following options are selected:
   * *Environment type*: Create a new instance for environment (EC2)
   * *Instance type*: t2.micro (1 GiB RAM + 1 vCPU)
   * *Platform*: Amazon Linux
   * *Cost-saving setting*: After 30 minutes (default)
6. Click **Next step**.
7. Verify all information.
8. Click **Create environment**. (it will take about 1 minute)

**Create a Lambda Function with AWS Cloud9**

After done creating a new Cloud9 environment, the IDE will be automatically opened, your screen should look like this:



1. Click **Create Lambda Function...** link
2. Enter your Function name and your Application name.
   * *Function name*: HelloWorld
   * *Application name*: MyFirstLambdaFunction
3. Click **Next**.
4. Select runtime: **Python 3.6**.
5. Select **empty-python**.
6. Click **Next**.
7. Select Function trigger: **API Gateway**.
8. Trigger Settings:
   * *Resource Path*: **/**
   * *Security*: **NONE**
9. Click **Next**.
10. Make sure the following options are selected:
    * *Memory (MB)*: **128 MB**
    * *Role*: **Automatically generate role**
11. Click **Next**.
12. Verify all information and click **Finish**.
13. Navigate to your function folder in the left panel and open file **lambda\_function.py**.
14. Delete the existing code and replace with this:

import json

print('Loading your function')

def lambda\_handler(event, context):

#print("Received event: " + json.dumps(event, indent=2))

# print statements actually get printed to the logs.

print("message --> " + event['message'])

# Actually returning the value of the 'message' key.

return event['message']

# Raising an exception if something goes wrong...

raise Exception('Something went wrong!')

1. Save your file.

**Test your Lambda Function**

1. While still opening file **lambda\_function.py**, click **Run** button on the top menu.
2. A new window will open up, input this into the Payload section:

{

"message": "Hello World!",

"notmessage": "If this shows, it is broken!"

}

1. Pay attention to the **Execution results** section, if the **Response** is **“Hello World!”** then congratulation, you have done everything right! Otherwise, please repeat the previous steps.
2. Feel free to play around with your code or change the message to see different results.

**Deploy your Lambda Function**

1. Click on **AWS Resources** tab on the right-side panel.
2. Right click on your Application and select **Deploy**.
3. Navigate to Lambda: <https://ap-southeast-1.console.aws.amazon.com/lambda>
4. Click on **Applications** at the left-side menu.
5. Verify that you newly deployed application is there. In this case, it should have the name **cloud9-MyFirstLambdaFunction**.
6. Click on **Functions** at the left-side menu.
7. Verify that you newly deployed Lambda function is there. In this case, it should have the name started with **cloud9-MyFirstLambdaFunction-HelloWorld-…**.

### Create a Test Event and Manually Invoke the Function Using the Test Event

1. While at Functions list in the Lambda console, click on your function name.
2. In the dialog, select **Create new test event**.
3. Select the **Hello World** event template.
4. Give it an event name (e.g., "Test 1").
5. Replace the provided json with this:

{

"message": "Hello World!",

"notmessage": "If this shows, it is broken!"

}

1. Click **Create**.
2. Make sure your newly created test event is selected in the test dialog.
3. Click **Test** and verify if the function is success.

**Verify That CloudWatch Has Captured Function Logs**

1. Navigate to **CloudWatch**: <https://ap-southeast-1.console.aws.amazon.com/cloudwatch>
2. Select **Logs** in the left-hand menu.
3. Select the log group with your function name in it.
4. Select the log stream within the log group.
5. Verify the output is present and correct.

**Conclusion**

Congratulations on completing this hands-on lab!