Activity 1: Serverless using API Gateway and Lambda

Lambda Function

1. Create a lambda function from template
   a. Use a blueprint
   b. Blueprint name: Hello world function
   c. Runtime: python3.10
   d. Lambda name: `{name}-hello-world`
2. Invoke the lambda function using the 'test' button
   a. Hit 'Invoke' instead of 'Save'
   b. Success would be a execution log that prints `Loading function` message
3. Remove the lambda_handler function from the code
   a. Invoke the lambda function
   b. Success would be a execution log that indicates the function is missing
      Note: Do not proceed until this is complete
4. Using the cloudwatch console, attempt to
   a. find the execution log results
5. Replace the code with the following

```
import json

print('Loading function')

def lambda_handler(event, context):
  payload = json.dumps(event, indent=2)
  print("Received event: " + payload)
  return {
    'statusCode' : 200,
    'body': 'hello world'
  }
```

   a. Invoke the function
   b. Success would be a execution log that prints the hello world message
      Note: Do not proceed until this is complete
6. Explore the tabs in your lambda function and find the following:
   a. Timestamp of the failed invocations
   b. Logs of your recent invocations
   c. Most expensive invocations
   d. What are the IAM permissions given to the lambda function
   e. What is the memory and timeout configured for the function

API Gateway

7. Create a API Gateway
    a. Use REST API > New API
    b. API name: `{name}-rest-api`
8. Create `hello` resource
9. Create `GET` method against the resource
    a. Integration type: Lambda function
    b. Check against Lambda proxy integration
    c. Select your lambda function
10. Under the `hello` resource `GET` method, execute a `test`
    a. Success would be returned with a `200` status message
    Note: Do not proceed until this is complete
11. Create a new stage
    a. On the resource page, hit the `Deploy API` button
    b. Stage: New stage
    c. Name: dev
12. Get the Invoke URL under the stage
    a. Prepare a `curl` command to hit the url

Activity 2: Securing the API Gateway

1. Under the `hello` resource `GET` method, set API key required to true
   a. Is the Invoke URL still working?
   b. Success would be a `Forbidden` message
   Note: Do not proceed until this is complete
2. Create an API key
3. Create a Usage Plan
   a. Name: `{name}-usage-plan`
   b. Enable throttling
   c. Rate: 1
   d. Burst: 1
   e. Disable Quota
4. Associate your stage to the usage plan
5. Use a `curl` command to hit your invoke url
   a. Hint: HTTP Header (name: `x-api-key`)
   b. Success would be the hello world message
   Note: Do not proceed until this is complete
6. Enable the Quota in your Usage Plan
   a. 5 request per day
7. Use a `curl` command to hit your invoke url
   a. Attempt to the limits using the curl command
   b. Success would be error code `429` with `Limit Exceeded` message
   Note: Do not proceed until this is complete

Activity 3: Clean Up

Ensure
1. Delete the Policy associate with the Lambda Role
2. Delete the Lambda Role associated with the Lambda
3. Delete the Lambda function
4. Delete the API Key
5. Delete the Usage Plans
6. Delete the API Gateway

Bonus Activity: Replicate the Changes into Terraform

Using the terraform configuration files provided on Thursday, enhance it to replicate the changes introduced in Activity 2.

Hint:
https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/api_gateway_usage_plan_key#example-usage