Exercises for AWS Cloud Native Development

Exercise 1 (S3)

- Write code which does the following
    - Create a bucket
    - List all buckets
    - Delete a bucket
    - Upload file to bucket
    - Download file from bucket

Exercise 2 (S3 and SNS)

- Create SNS topic and subscribe to topic via console
- Using console ensure an S3 upload triggers and event
- Send the event to SNS
- Check if you received mail from SNS
- Not down the S3 event structure

Exercise 3 (S3 Event Structure)

- Copy the S3 event structure from the email to a file or to your program
- In your program, extract the bucket name and file name using the JSON parser
- Print the bucket name and file name

Exercise 4 (SQS)

- Write code that does the following
    - Create a SQS Queue
    - List all Queues
    - Write a message to SQS Queue
    - Read a message from the SQS Queue
    - Delete a message
    - Delete a SQS Queue

Exercise 5 (SQS and S3)

- Modify the event management rule in S3 to point to SQS
- Upload file to S3 using your code
- The S3 event must be sent to SQS
- Read the S3 event from SQS and print out the file name and the bucket name

Exercise 6 (DynamoDB)

- Write a program that does the following

- Create DynamoDB table
- Write an item into DynamoDB table
- Read an item from DynamoDB table
- Delete an item from DynamoDB table
- Update an item in DynamoDB table
- Query a DynamoDB table
- Delete a DynamoDB table

Exercise 7 (Lambda)

- Write and test a Lambda function that prints out a S3 event

Exercise 8 (Lambda and S3)

- Set the event target in S3 to Lambda
- Write a Lambda that prints out only the uploaded file name and bucket name
- Verify if the Lambda is working by uploading file to S3

Exercise 9 (API Gateway)

- Write a Lambda function that returns the string, "Welcome to AWS Cloud Native Development"
- Create a HTTP API
- Integrate the GET method with the newly created Lambda

Exercise 10 (API Gateway, DynamoDB and Lambda)

- Write a Lambda function that does the following
  - Writes an item to a DynamoDB table
  - Reads from a DynamoDB table
  - The item written should be bucket name and file name
- Create a HTTP API, wherein you have two methods
  - GET method. Get bucket and file name details from DynamoDB using Lambda
  - PUT method. Put bucket and file name into DynamoDB using Lambda

Exercise 11 (DynamoDB streams and Lambda)

- Turn on DynamoDB streams
- Ensure the trigger is Lambda
- Print the Stream event to CloudWatch Logs

Grand Design

- Upload a file to S3 keeping Lambda as the event target
- Write a Lambda that will extract the bucket name and file name from the event structure
- Call the API Gateway with the PUT method using these values
- The bucket name and file name must be updated in the DynamoDB table
- (For calling API we can use Postman)

Cognito

- Use the link provided by the instructor.
- Register yourself using this link
- Test if registration is successful
- Then test if the ‘Forgot Password’ workflow works correctly