Week 6 Notes and Resources

Key Concepts

AWS Messaging Services

In modern cloud architecture, applications are decoupled into smaller, independent building blocks that are easier to develop, deploy and maintain.

In this course, part of the application runs on EC2, part of it on AWS Lambda, and another part could run on-premises. Message brokers make it easier to decouple these parts of the application and process events asynchronously.

AWS offers several messaging services and these include:

- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Message Broker for AWS IoT
- Amazon MQ

For more on developing distributed applications that use messaging services, see Building Scalable Applictions and Microservices: Adding Messaging to Your Toolbox on the AWS Blog.

Publish and Subscribe with Amazon SNS

Amazon Simple Notification Service (SNS) is a fully managed Publish Subscribe (pub/sub) messaging service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications. With SNS, you can use topics to decouple message publishers from subscribers and to fan-out messages to multiple recipients at once.

The Publish Subscribe model allows messages to be broadcast to different parts of a system asynchronously. A message topic provides a lightweight mechanism to broadcast asynchronous event notifications.

In the class project, the publisher is Amazon S3 which is publishing events as new objects appear in the Bucket. There are 2 subscribers - an AWS Lambda function and an Amazon SQS Queue.

You can review the SNS Reference docs for Boto3.

Message Queues with with Amazon SQS

Amazon Simple Queue Service (SQS) is a fully managed message queuing service. SQS makes it simple and cost-effective to decouple and coordinate the components of a cloud application.

Message queues allow different parts of a system to communicate and process operations asynchronously. A message queue provides a lightweight buffer which temporarily stores messages. To send a message, a component called a producer adds a message to the queue. The message is stored on the queue until another component called a consumer retrieves the message and does something with it.

In the class project, the publisher is an Amazon SNS topic. The subscriber is a server polling for images to print. This architecture gives us the option of running the printing server on-premises while the main application and the Lambda function continue to run in AWS.

You can review the SQS Sample tutorial or the Boto3 SQS Reference docs.

What you accomplished this week

- You refactored the application to include an SNS topic to enable additional services to subscribe to new image upload events.

- You added a queue to the application to allow for a new service to poll for incoming photos.
- YOU COMPLETED THE COURSE!! :)
- Don't forget to register for our next course: Deploying on AWS