Labda Lecture and Hands On: https://www.youtube.com/watch?v=J-oyDzrgTFI&list=PLzOJMOiC_x7dkAtC1n8nI8HkT4zw2CK_x

What is Lambda?

- Lambda is a serverless *Function as a Service*
- Lambda is used in infrastructure management tasks like *capacity provisioning and patching*, so you can focus on only writing code that serves your customers.
- With Lambda, you can run code for virtually any type of application or backend service with *zero administration*

Benefits

- No servers to manage
- Continuous scaling
- Cost optimized with millisecond metering
- Consistent performance at any scale

Lambda may be triggered by events from other services

Lambda can read events from:

- *DynamoDB*: NoSQL database used in conjunction with AWS services
- *Kinesis*: A massively scalable and durable real-time data streaming service
- *Simple Queue Service (SQS)*: Used when sending messages via web service applications as a way to communicate over the Internet

Some services that invoke Lambda (a)synchronously:

- *Cognito*: Lets you add user sign-up, sign-in, and access control to your web and mobile apps

- *API Gateway*: Makes it easy for developers to create, publish, maintain, monitor, and secure APIs
- *S3*: An object storage service
- *CloudWatch Logs/Events*: Monitoring and observability service
- *Code Commit*: It makes it easy for teams to securely collaborate on code with contributions encrypted in transit and at rest
- *Code Pipeline*: A fully managed continuous delivery service

---

Synchronous vs. Asynchronous Invocation
Synchronous invocation - Lambda runs the function and *waits for a response*.
Asynchronous invocation - *don't wait for a response* from the function code; instead you hand off the event to Lambda; Lambda automatically places the event in a queue and sends a success response

---

Event and Context

- *Event* is an object passed into a Lambda function that *stores the inputs* that are provided
- *Context* is another object passed into Lambda. It *stores one function and 9 properties*, including:
  - get_remaining_time_in_millis (a function)
  - function_name
  - function_version
  - invoked_function_arn
  - memory_limit_in_mb
  - aws_request_id
  - log_group_name
  - log_stream_name
  - identity (for Cognito; contains identity id and pool id as sub-properties)
  - client_context (for mobile apps; stores many sub-properties)

Push/Pull Sources

Push Sources	Pull Sources
S3 (e.g. ObjectCreated and ObjectRemoved)	DynamoDB – get batches of updates published to stream
API Gateway (HTTP method calls)	Kinesis – Lambda will poll a stream once per second for each stream shard
Amazon SNS (Messages published can be delivered to Lambda)	
CloudFormation (execute custom commands and provide data back to stack during its creation)	
CloudWatch Events (e.g. notify mailing list when Trusted Advisor reports a status change)	
Alexa	

Debugging

To debug your Lambda function, access its logs in CloudWatch