Amazon Lex

Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions. With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to any developer, enabling you to quickly and easily build sophisticated, natural language, conversational bots (“chatbots”).

Amazon Lex pricing

With Amazon Lex, you pay only for what you use. You are charged based on the number of text or voice requests processed by your bot. Please see the pricing page for current information.

What is Serverless?

Serverless is the native architecture of the cloud that enables you to shift more of your operational responsibilities to AWS, increasing your agility and innovation. Serverless allows you to build and run applications and services without thinking about servers. It eliminates infrastructure management tasks such as server or cluster provisioning, patching, operating system maintenance, and capacity provisioning. You can build them for nearly any type of application or backend service, and everything required to run and scale your application with high availability is handled for you.

In this class, we will build a serverless application that uses Amazon S3, host a static serverless website. Amazon Cloudfront to securily cost-effectively distrubute our content. Amazon Lex, to create serverless chatbot. Amazon API Gateway, to create a test endpoint before we integrate the front and backend of your application. Amazon DynamoDB, will provide a serverless database service behind AWS Lambda, our serverless compute service.

Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application.

Mock Integrations in API Gateway

Amazon API Gateway supports mock integrations for API methods. This feature enables API developers to generate API responses from API Gateway directly, without the need for an integration backend. As an API developer, you can use this feature to unblock dependent teams that need to work with an API before the project development is complete. You can also use this feature to provision a landing page for your API, which can provide an overview of and navigation to your API. For an example of such a landing page, see the integration request and response of the GET method on the root resource of the example API discussed in Build an API Gateway API from an Example.

Pricing

With Amazon API Gateway, you only pay when your APIs are in use. There are no minimum fees or upfront commitments. For HTTP/REST APIs, you pay only for the API calls you receive and the amount of data transferred out. Amazon API Gateway is Free Tier eligable and The Amazon API Gateway free tier includes one million API calls received for HTTP/REST APIs, and one million messages and 750,000 connection minutes for WebSocket APIs per month for up to 12 months. See the Amazon API Gateway pricing page for details. Amazon CloudFront

Amazon CloudFront is is a global content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to your viewers with low latency and high transfer speeds. CloudFront is integrated with AWS – including physical locations that are directly connected to the AWS global infrastructure, as well as software that works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code close to your viewers. The cost to use CloudFront is based upon data transfer costs as well as requests and includes a free usage tier.

Cloudfront Edge Locations as of March 2019:To deliver content to end users with lower latency, Amazon CloudFront, uses a global network of 166 Points of Presence (155 Edge Locations and 11 Regional Edge Caches) in 65 cities across 29 countries. For current information about Edge Locations, see this page.

Origin Access Identity (OAI)

Origin Access Identity provides a method to restrict access to S3 content to only a CloudFront Distribution.

Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications. Pricing for DynamoDB includes a non-expiring free tier allotment. AWS Identity and Access Management

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. We typically use credentials from IAM Users or IAM Roles to authenticate with AWS when making API calls. We control the permissions for which API actions those Users or Roles can perform with IAM Policies.

In this course we focus on IAM roles. An IAM role is an IAM identity that you can create in your account that has specific permissions. Unlike an IAM user, A role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users. Detailed information about AWS IAM can be found in the documentation.

AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running.

Because your code is only running on-demand, in order to troubleshoot, you'll need to take advantage of monitoring. You can troubleshoot Lambda with Amazon Cloudwatch and also troubleshoot Lambda with AWS X-Ray.

With Lambda, there are 2 types of permissions to be aware of. First, your code running in a Lambda function is granted permissions by using an IAM Role. Second, the service triggering your Lambda function can be granted the permissions to do so with a Lambda Function Policy. When configuring triggers using the AWS Management Console, the console will create the necessary Lambda Function Policy for you when you enable the trigger.

If you want more details about what's available to your code as it runs inside AWS Lambda, you can find out more about the Lambda execution environment in the documentation.

Deploying an Amazon Lex Bot on a Messaging Platform

In the video, we explained how to integrate our Lex bot with Facebook messenger. As noted, in Addition to Facebook messenger, you can deploy your Lex bot on Slack and Twilio.

Examples and details about how to do this can be found in the Lex Documentation.