

Power Up Chatbots with AWS Lambda

Fintech
Lesson 15.2



Class Objectives

By the end of this class, you'll be able to:



Describe how Lambda implements serverless applications.



Apply your Python skills to building a Lambda function.



Test and debug the code of a Lambda function.



Use a Lambda function to add new features to an Amazon Lex bot.



WELCOME



Introduction to AWS Lambda

Introduction to AWS Lambda

With Lambda, we can deploy and run code in the cloud—without concerning ourselves with server configuration or administration.

This type of cloud application is known as a **serverless application**.





**Let's discuss how serverless
applications work and how
Lambda implements them.**

Introduction to AWS Lambda

If you want to create a chatbot to help customers make cryptocurrency trading decisions, you'll need to connect various components:



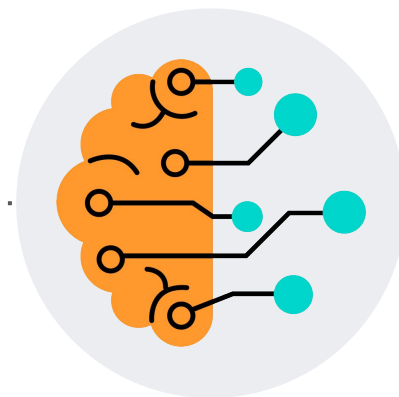
Chatbot development tool

to define the dialogue's interaction



Software application

to fetch both historical and current crypto prices by using an API



Machine learning model

that's deployed to the cloud to aid trading decisions



Database

to store and manage all the information

Introduction to AWS Lambda

Managing all these components poses a challenge. You need to:



Be aware of software updates.



Diagnose hardware for potential failures.



Secure your servers to prevent cyberattacks.

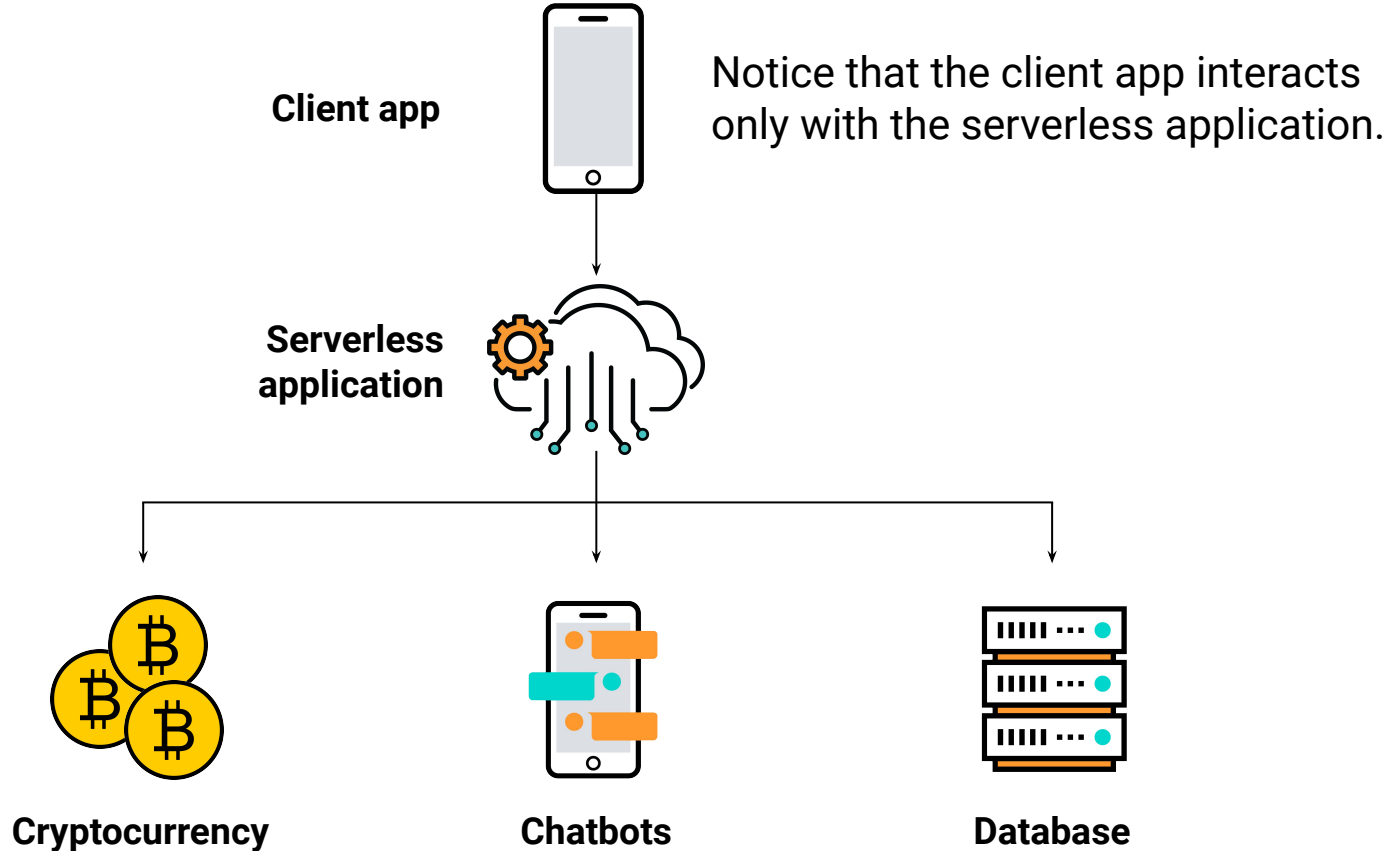


Maintain different servers to host all the components.



This is where the power of serverless applications comes in.

Robo Advisor Architecture



Serverless Application



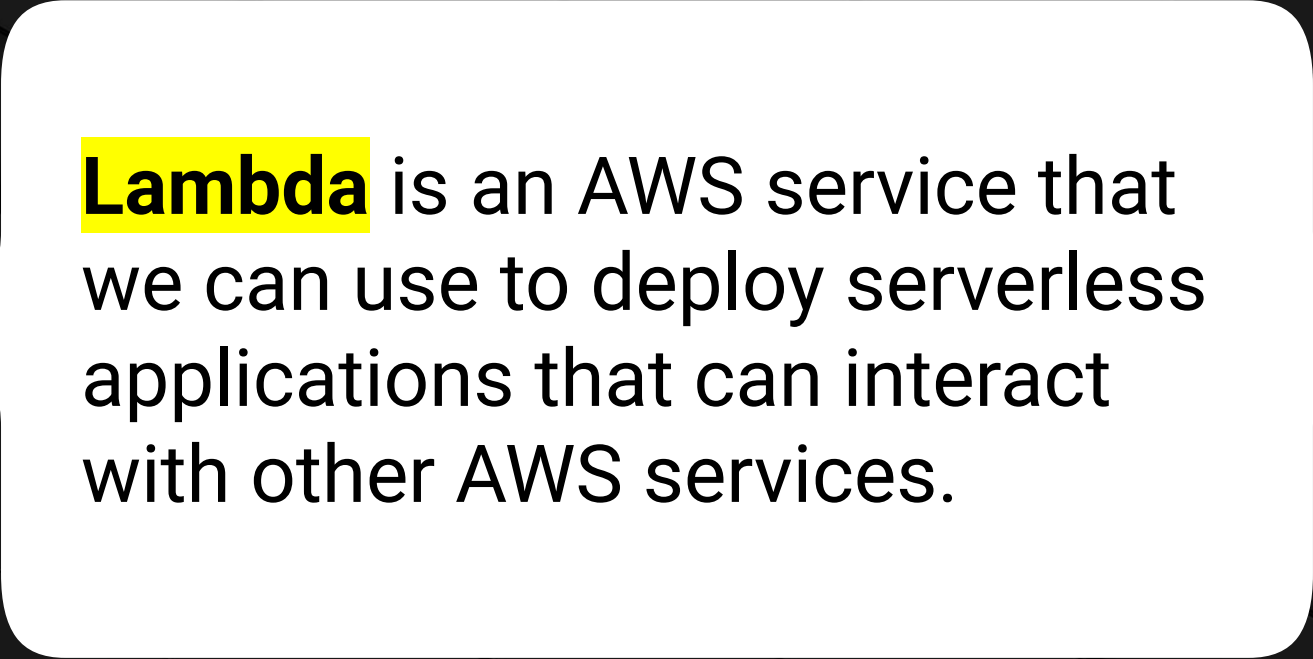
When we deploy a serverless application, it runs in the cloud on one or several servers, depending on the demand.



The cloud services provider that hosts the serverless application will automatically scale the infrastructure depending on the demand.



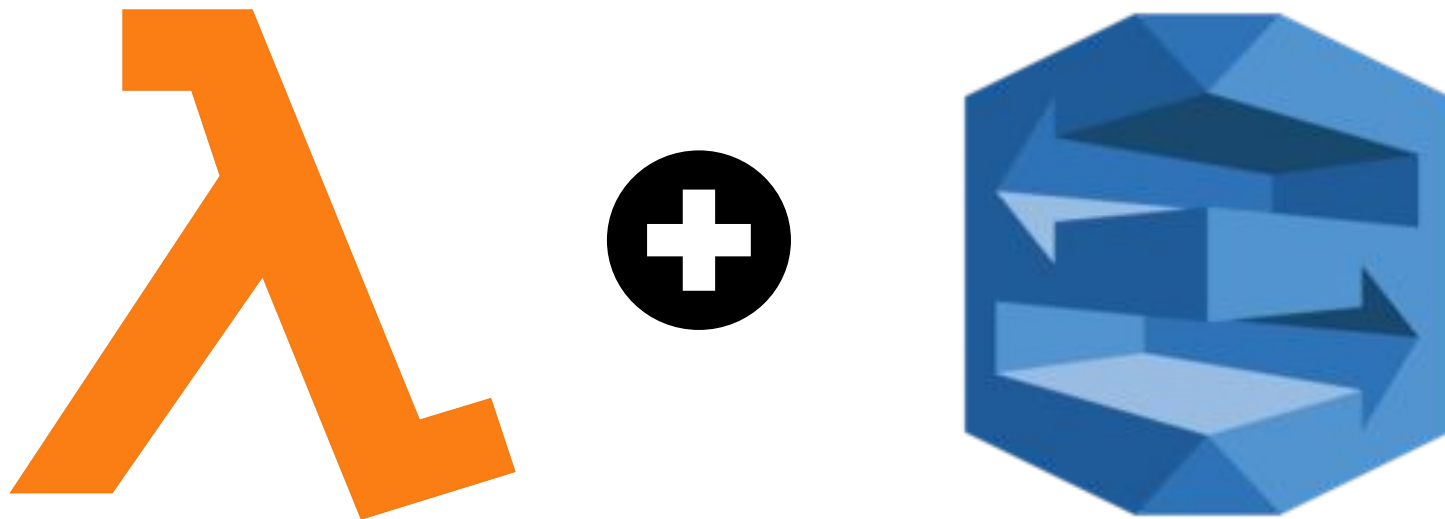
This ensures the optimal functioning of our software.



Lambda is an AWS service that we can use to deploy serverless applications that can interact with other AWS services.

AWS Lambda and Amazon Lex

You'll learn how to use Lambda to create a serverless application that interacts with Amazon Lex.



Questions?





Instructor Demonstration

Creating Python Functions with AWS Lambda

Creating Python Functions with AWS Lambda (1 of 3)

To create a Python function with Lambda, we'll:

01

Open the Lambda console.

02

Create a Lambda function.

03

Open the Lambda function in the online code editor.

04

Organize the Lambda function code.

Creating Python Functions with AWS Lambda (2 of 3)

We can organize a Lambda function into six general building blocks. These will manage the interaction with Amazon Lex and implement the necessary business logic:

01

Required Libraries

02

Functionality Helper Functions

03

Dialog Actions Helper Functions

04

Intents Handlers

05

Intents Dispatcher

06

Main Handler

Creating Python Functions with AWS Lambda (3 of 3)



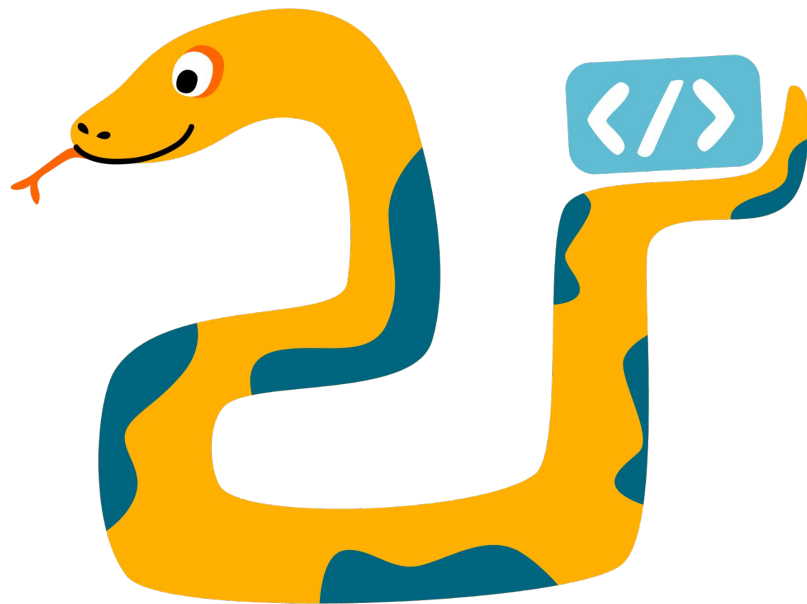
Getting started with coding Lambda functions might seem challenging. But, you already have the required Python skills to succeed!



A Lambda function resembles the Python code you're familiar with.



But, you need to arrange it according to a special structure that allows Amazon Lex to communicate with Lambda.



Anatomy of a Lambda Function

This is how we import the requests library into AWS Lambda

Helper functions implement data validation and business logic support

Dialog helper functions control conversation response events

Intent handlers use helper functions and dialog helper functions to fulfill user intents

The intent dispatcher validates the current intent and dispatches it to the corresponding intent handler

```
### Required Libraries ###
from datetime import datetime
from dateutil.relativedelta import relativedelta
from botocore.vendored import requests

### Functionality Helper Functions ###
def parse_float(n):...

def get_btcprice():...

def build_validation_result(is_valid, violated_slot, message_content):...

def validate_data(birthday, dollars, intent_request):...

### Dialog Actions Helper Functions ###
def get_slots(intent_request):...

def elicit_slot(session_attributes, intent_name, slots, slot_to_elicit, message):...

def delegate(session_attributes, slots):...

def close(session_attributes, fulfillment_state, message):...

### Intents Handlers ###
def convert_dollars(intent_request):...

### Intents Dispatcher ###
def dispatch(intent_request):...

### Main Handler ###
def lambda_handler(event, context):
    """
    Route the incoming request based on intent.
    The JSON body of the request is provided in the event slot.
    """
    return dispatch(event)
```

Questions?





Activity: Enhancing the Bitcoin Fear & Greed Robo Advisor

In this activity, you'll create a Lambda function that adds functionality to the Bitcoin Fear & Greed Robo Advisor that you started in an earlier class.

Suggested Time:

15 Minutes



Time's Up! Let's Review.

Questions?





Instructor Demonstration

Connecting AWS Lambda and Amazon Lex

Questions?





Countdown timer

15:00

(with alarm)



Activity: Adding Business Logic to the Robo Advisor with AWS Lambda

In this activity, you'll connect the Lambda function that you created before to the robo advisor to add business logic to it.

Suggested Time:

25 Minutes



Time's Up! Let's Review.

Questions?



Dealing with Buggy Lambda Functions

Dealing with Buggy Lambda Functions (1 of 2)

In this activity, you'll learn how to test and debug Lambda functions in the [AWS Management Console](#).



Lambda interacts with other AWS services by processing event messages in the JSON format.



JSON is one of the industry standards for sharing data on the internet.



The comma-separated values (CSV) and XML formats are the other standards.

Dealing with Buggy Lambda Functions (2 of 2)

Each AWS service uses a specific JSON format.

```
1  {
2    "messageVersion": "1.0",
3    "invocationSource": "DialogCodeHook",
4    "userId": "John",
5    "sessionAttributes": {},
6    "bot": {
7      "name": "CryptoConverter",
8      "alias": "$LATEST",
9      "version": "$LATEST"
10   },
11   "outputDialogMode": "Text",
12   "currentIntent": {
13     "name": "convertUSD",
14     "slots": {
15       "birthday": "1978-12-16",
16       "dollars": "0"
17     },
18     "confirmationStatus": "None"
19   }
20 }
21
```



Instructor Demonstration

Dealing with Buggy Lambda Functions

Questions?





Activity: Debugging Lambda Functions

In this activity, you'll test and debug the `getFGIndex` Lambda function that you created before.

Suggested Time:

15 Minutes



Time's Up! Let's Review.

Questions?





Recap

In today's class, you learned how to:



Build complex and automatic dialogues with just a few clicks by using Lambda functions.



Use Lambda functions to improve the NLP experience for users by both debugging and adding customized business logic.



Create and deploy a serverless application by using Lambda.



Add customized business logic and data validation by making a connection to Amazon Lex.

Recap

CUIs offer several opportunities to engage with customers in a new way.

Apple Siri



Amazon Alexa



Google Assistant

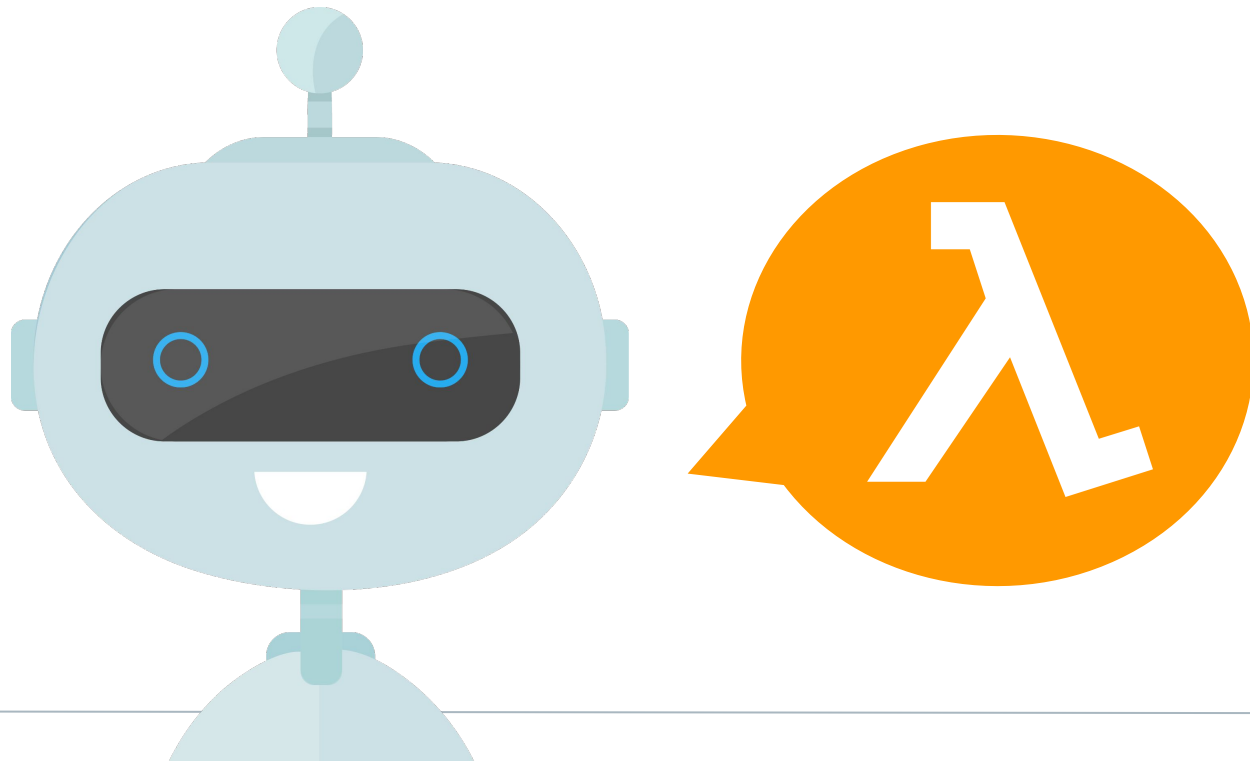


A chatbot offers a virtual agent that can support customers of financial organizations 24–7.



Recap

By adding rules with Lambda, you can now create a bot that will help customers by answering questions in the form of natural language.





Next Class

**You'll learn to deploy machine learning models
that can connect with other AWS services
to create more robust applications.**

Questions?



*The
End*